ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Machine Learning for Computer Vision

3D RECONSTRUCTION AND ANALYSIS OF LITHIUM-ION BATTERIES FROM X-RAY IMAGES

CANDIDATE
Marco Solimè

SUPERVISOR
Prof. Samuele Salti

CO-SUPERVISOR
Prof. Imari Sato

Academic year 2023-2024

Session 1st

To my family and my home university, that allowed me to embark in this unprecedented (and life-changing) journey.

Contents

List of Figures

List of Tables

Abstract

The three-dimensional reconstruction of lithium cobalt oxide (LCO) batteries in the X-ray domain is crucial for analyzing their internal structure and degradation processes. Traditional reconstruction techniques require acquiring and aligning thousands of X-ray projections, making the process computationally expensive and impractical for certain applications. In this work, we explore state-of-the-art rendering techniques to significantly reduce the number of required projections while maintaining high reconstruction quality. Our primary focus is on 3D Gaussian Splatting (3DGS), which we demonstrate to be a faster and more efficient alternative compared to Neural Radiance Fields (NeRF).

We begin by introducing the problem, detailing the X-ray acquisition system, and discussing the role of X-ray absorption fine structure (XAFS) technology in LCO battery analysis. Through systematic experimentation, we compare 3DGS and NeRF, showing that 3DGS achieves superior reconstruction quality with significantly lower computational cost and training time. Additionally, we investigate strategies for optimizing reconstruction quality by carefully selecting, averaging, and filtering projections to minimize acquisition and processing overhead. Our findings confirm that modern rendering techniques, particularly 3DGS, enable high-fidelity 3D reconstructions with an order-of-magnitude reduction in required projections, making them a promising solution for efficient battery analysis.

Chapter 1

Introduction

1.1 Problem Definition

3D object reconstruction from X-ray images involves recovering the internal structure of an object as a 3D volume from a series of 2D X-ray projections. This process is fundamental in fields such as medical imaging, material science, industrial inspection, and cultural heritage preservation.

The reconstruction is typically achieved using techniques like computed tomography (CT), where a complete dataset of X-ray projections taken at multiple angles is used to reconstruct the 3D volume. However, acquiring many projections is time-consuming, expensive, and exposes objects (or patients) to higher radiation doses, making it critical to achieve high-quality reconstruction with a reduced number of input projections.

The goal of this thesis is to:

- Minimize the number of input X-ray projections while still achieving high-quality 3D reconstructions. Volumes must be accurate, i.e., faithfully represent the original object; small details should be preserved; and artifacts and noise should be minimized
- Adapt and train SoTA reconstruction methods that compensate for the reduced projection data

- Balance the trade-offs between computational cost, training time, reconstruction quality, and practical constraints (e.g., data acquisition and radiation exposure)
- Define metrics to evaluate reconstructed volumes with and without ground truth data

1.2 Metrics

When it comes to evaluating a density volume, we need to establish whether ground truth (GT) data is available, the accuracy and reliability of such data, or whether reference data is not available. For this reason, we can categorize our metrics according to the presence of a ground truth volume.

With GT volume In this scenario, a direct comparison of a reconstructed volume and a reference volume is possible. A GT volume can be obtained in different ways according to the application of interest. **Synthetic Volume** A 3D object can be designed and modeled with a 3D Software, e.g., Blender[8], and can be printed using a 3D printer. In this case, the shape of the reference volume is known because it has been manually designed by a modeler. To evaluate the capacity of the model to distinguish between low and high density regions, we may print objects using composite materials (e.g., a mixture of plastic, metals, resins, ceramic), or calibrate the infill percentage for volumetric region. **Traditional Volume** If a high and accurate number of projections are captured around the object (e.g., 1800 images), a high-fidelity volume can be reconstructed using traditional algorithms (e.g., Filtered Back Projection [16]) and can be treated as GT. This assumes a qualitative analysis by experts to certify the accuracy and fidelity of the volume.

Once the GT volume is available, we can transfer metrics usually used to compare 2D images to the world of 3D. Metrics include the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

Peak Signal-to-Noise Ratio is defined as

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right), \tag{1.1}$$

where MAX is the maximum possible pixel value of the volume and MSE is the Mean Squared Error, namely

$$\text{MSE} = \frac{1}{N_x N_y N_z} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{z=1}^{N_z} \left(I_{\text{GT}}(x,y,z) - I_{\text{pred}}(x,y,z) \right)^2 . \tag{1.2}$$

In general, a high PSNR (>30 dB) means good reconstruction, whereas a low PSNR indicates greater noise or deviation from the ground truth. SSIM measures the perceptual quality of the reconstructed volume according to the ground truth, focusing on the structural similarity.

$$SSIM_{3D} = \frac{1}{3} \left(SSIM_D + SSIM_H + SSIM_W \right) \tag{1.3}$$

$$SSIM(X,Y) = \frac{(2\mu_X \mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{1.4}$$

In Eq. 1.3, $SSIM_{3D}$ is computed as the mean of the 2D SSIM for each volume dimension. In Eq. 1.4, X and Y are two given slices of the predicted and reference volume. $\mu_{\text{X}}, \mu_{\text{Y}}$ are the mean intensities of the slices; $\sigma_{\text{X}}^2, \sigma_{\text{Y}}^2$ are the variances; $\sigma_{\text{X, Y}}$ is the covariance; C_1 and C_2 are stabilization constants to avoid division by zero.

An SSIM $= 1$ means perfect similarity; an SSIM ≈ 0 means no perceptual similarity.

Without GT Volume If the GT volume is unavailable, we need to engineer metrics to assess the quality of the reconstructed volumes without the presence of a reference volume. In general, a visual assessment of the volumes suggests the outcome of a successful reconstruction. Given a small amount

of prior knowledge about the object, the presence of artifacts or unwanted floaters provides a first quality assessment. Noise and abrupt density changes also signal a poor reconstruction.

Following [15], we can segment active regions from empty regions in the volumes. Regions are labeled as active if we are confident enough that there is high-density material. Such labeling process is usually supervised by an expert that is able to interpret the predicted volume and assess their quality; empty regions, corresponding to low-density material, are labeled as background. On such segmented volumes, signal-to-noise ratio (SNR) and contrast-to-noise ratio (CNR) can be calculated as follows:

$$\text{SNR} = 20 \log_{10} \left(\frac{\mu_i}{\sigma_b} \right) \tag{1.5}$$

$$\text{CNR} = 20 \log_{10} \left(\frac{|\mu_i - \mu_b|}{\sigma_b} \right) \tag{1.6}$$

μ_i and μ_b represent the mean values of the foreground and background regions, respectively, while σ_b is the standard deviation of values in the background regions. The factor of 20 in the SNR and CNR formulas arises because these ratios are typically defined in terms of power, which is proportional to the square of the amplitude. Using the logarithm property $\log_{10}(x^2) = 2 \log_{10}(x)$, the squared term introduces a factor of 2, resulting in $10 \cdot 2 = 20$ as the multiplicative factor in the decibel calculation.

Binary mask Labeling 3D regions on a volume is a cumbersome, error-prone, and dataset-specific process. An alternative approach is to use an unsupervised learning algorithm (e.g., K-Means with $K = 2$) on a good-enough reconstruction and classify each voxel between high or low density region. We will use this method in our experiments to get a quantitative evaluation of the results in little time while avoiding manual annotations of active regions. Since we are interested in the active material (stone-like features, see Figure

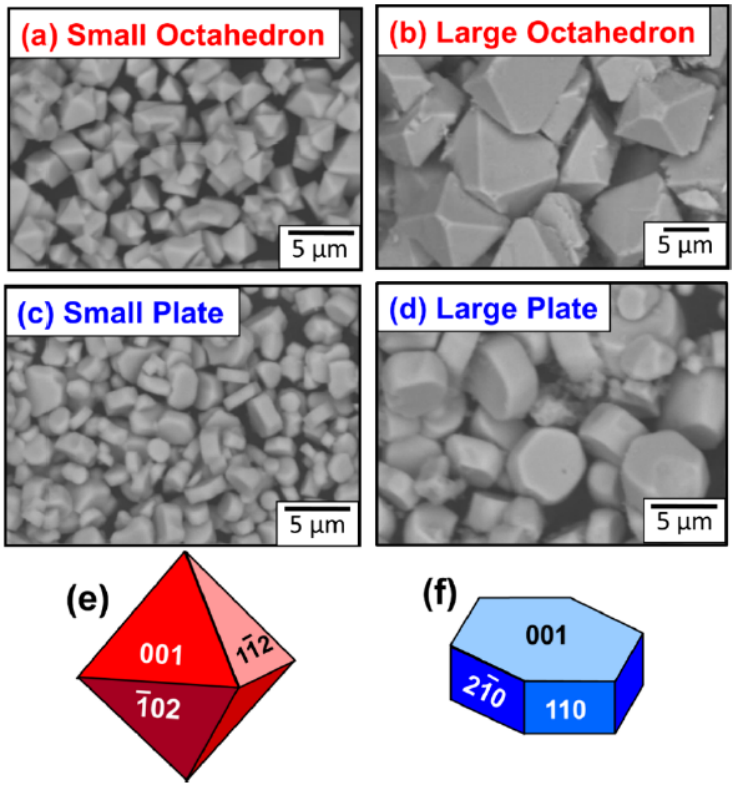

Figure 1.1: Illustration from Ariyoshi et al. [2], depicting SEM images (Scanning Electron Microscope images) of NCM (Lithium Nickel Cobalt Manganese Oxide) particles. (a-b) octahedral and (c-d) plate-like morphologies are synthesized using different fluxes and temperatures, illustrating the influence of synthesis conditions on particle shape and size. (e) and (f) depict the morphologies of octahedral and plate-like features, respectively.

1.1) in the inside of the object, a central vertical crop is performed to exclude boundaries that may alter evaluation.

To better isolate the most salient features, we crop the cuboid inscribed in the circumference of the battery and add some margin.

1.3 RGB vs X-ray imaging

The crucial difference between RGB and X-ray 3D reconstruction lies in the physics of light interaction with the objects and the type of information captured during the imaging process.

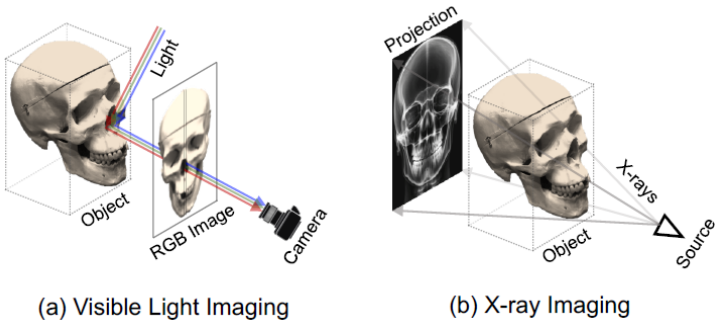

Figure 1.2: Visible light vs. X-ray. Visible light imaging relies on reflection. X-ray imaging is based on penetration and attenuation. Illustration taken from Cai et al. [6].

Physics of RGB and X-ray imaging See Fig. 1.2 for a visual comparison between RGB and X-ray imaging. RGB imaging relies on reflection and scattering of visible light off the surface of an object. Cameras are passive sensors that capture intensity and color or reflected light. What is captured is the surface-level information, as visible light cannot penetrate most materials.

On the other hand, X-rays operate in a completely different setting, and interact with materials in different ways. Absorption is the capacity of denser of thicker materials to absorb more X-rays, leaving fewer photons to reach the detector. Some X-rays may be scattered out of the beam path. The remaining X-rays pass through the material and are captured as intensities in the detector.

Indeed, X-ray imaging captures internal structures by measuring how materials attenuate the X-ray beam. This provides a cross-sectional density map of the object, enabling volumetric reconstruction.

Difference in the data RGB imaging relies on high-resolution color data to reconstruct fine surface details. Reconstruction quality depends on feature visibility and lighting conditions. There is no information about internal structure. Instead, X-ray imaging provides intensity data proportional to material density and thickness along the ray. It enables 3D reconstructions of internal structures, but lacks color and texture information. It suffers from artifacts, like noise and ghosting, due to scattering or limited projections.

Consequently, methods that traditionally work with RGB data must be adjusted to work with X-ray data. Differently from RGB images, X-ray imaging captures projections, namely integrated density values along straight lines (ray paths). X-ray projections mix information from all materials along the path.

Chapter 2

Background

2.1 3D Reconstruction for X-ray imaging

2.1.1 Traditional Approaches

Analytical Approaches Filtered back projection (FBP) works well for dense, evenly distributed projections but struggle with sparse data. Its evolution, FDK[10], produces results almost immediately ($<$ 1 second), solving the Radon transform and its inverse [23]. However, both methods tend to introduce serious streak artifacts in sparse-view scenarios.

Iterative Methods These optimize the reconstruction iteratively by solving the system of equations formed by projections. Examples include ART [13], SIRT [12], SART [1], and other iterative algorithms [28, 20, 26]. They leverage regularization techniques (e.g., total variation) to reduce noise and artifacts. The main drawbacks of these techniques are that they take longer time ($<$ 10 minutes) and lose structure details.

2.1.2 NeRFs

Neural Radiance Fields (NeRFs) [21] are neural-based models that learn a volumetric scene representation and synthesize novel views by approximating

the radiance emitted from points in a 3D scene.

They represent the 3D scene as a continuous volumetric function parametrized by a neural network. They output a density and radiance (RGB color) at any queried point in 3D space.

$$F_\Theta : (x, y, z, \theta, \phi) \rightarrow (R, G, B, \sigma). \tag{2.1}$$

In Eq. 2.1, an MLP with weights Θ is employed to learn a mapping function F_Θ from the point position $(x, y, z) \in \mathcal{R}^3$ and view direction (θ, ϕ) to the color $(R, G, B) \in \mathcal{R}^3$ and volume density $\sigma \in \mathcal{R}$.

Rendering is performed via volume rendering, accumulating densities and colors along rays. In particular, for each pixel in the desired view, a ray is cast from the camera's position into the scene. A ray is parametrized as $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$, where $\mathbf{o}$ is the ray origin, $\mathbf{d}$ is the ray direction, t is the distance along the ray. To compute the color for a pixel, NeRF integrates the contribution of color and density along the ray $\mathbf{r}(t)$. The volume rendering equation is

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot c(\mathbf{r}(t))\, dt, \tag{2.2}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right)$ is the transmittance, representing the fraction of light that reaches t without being absorbed; $\sigma(\mathbf{r}(t))$ is the volume density at point $\mathbf{r}(t)$; $c(\mathbf{r}(t)) = (R, G, B)$ is the radiance (color) at point $\mathbf{r}(t)$; and t_n, t_f are the near and far bounds of the ray.

NeRF discretizes Eq. 2.2 by samplingthe ray at N points:

$$C(\mathbf{r}) \approx \sum_{i=1}^{N} T_i \cdot \alpha_i \cdot c_i, \tag{2.3}$$

where $T_i = \prod_{j=1}^{i-1} \exp(-\sigma_j \Delta_j)$ is the approximation of the transmittance up to sample i; $\alpha_i = 1 - \exp(-\sigma_i \Delta_i)$ is the probability of light being scattered at sample i; $\mathbf{c}_i$ is the color at sample i, Δ_i is the distance between consecutive sample points.

NeRF optimizes the parameters Θ of the function F_Θ by minimizing the reconstruction error between rendered and ground-truth images using a loss function:

$$\mathcal{L} = \sum_{\mathbf{r}\in\text{rays}} ||C_{\text{rendered}}(\mathbf{r}) - C_{\text{ground-truth}}(\mathbf{r})||^2. \tag{2.4}$$

MipNeRF [3] extends NeRF to render anti-aliased conical frustums instead of rays, improving representation and achieving faster multiscale scene rendering. Mip-NeRF 360 [4] extends mip-NeRF to handle unbounded scenes. Evolutions of NeRF have been applied to broader fields [7, 29].

In general, applying existing RBG NeRF methods for X-ray rendering (e.g., MedNeRF [9], NeAT [24]) may achieve suboptimal results due to the differences between visible light and X-ray imaging. For instance, NAF [31] follows NeRF to employ an MLP model for medical X-ray neural rendering, showing limitations in capturing complex structures of imaged objects in 3D space.

Although NeRF-based methods excel in per-case reconstruction, they are time-consuming (> 30 minutes) due to the extensive point sampling in volume rendering.

2.1.3 3D Gaussian Splatting

3D Gaussian splatting (3DGS) [17] has been a breakthrough in the world of 3D reconstruction because of its significantly faster rendering phase with respect to NeRF approaches. The code idea of 3DGS is to represent objects with a set of trainable 3D Gaussians primitives. During training, properties for each Gaussian like position, size, and color are optimized to best represent the scene.

3DGS takes as input a set of images with corresponding cameras calibrated with structure-from-motion (SfM [27]); as side effect, SfM produces a sparse point cloud of the scene. Then, 3D Gaussians are initialized on these

initial points with initial position (mean), covariance matrix, color (spherical harmonics), and opacity α. Optimization steps affecting parameters of each Gaussian are interleaved with operations for adaptive control of the Gaussian density. After "splatting" 3D Gaussians into the image plane, a tile-based rasterizer performs α-blending according to the order of each Gaussian.

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}\right) \tag{2.5}$$

As shown in Eq. 2.5, Gaussians are defined by a 3D covariance matrix Σ centered at point μ.

To project the 3D Gaussian to 2D for rendering, authors use a viewing matrix W and a Jacobian matrix J; the transformed 2D covariance matrix hence becomes $\Sigma' = JW\Sigma W^T J^T$. To avoid a covariance matrix with no physical meaning, authors parametrize it with a scaling matrix S and rotation matrix R: $\Sigma = RSS^T R^T$.

During optimization, $\mathcal{L}_1$ loss is computed between rendered images and reference images, and an additional *D-SSIM* loss is added to compensate for structural quality. D-SSIM loss is defined as:

$$\text{D-SSIM} = 1 - \text{SSIM}(I_{\text{rendered}}, I_{\text{ground truth}}).$$

The adaptive control mechanism controls the number of Gaussians and density over unit volume. For example, we talk about "under-reconstruction" in those regions missing geometric features, and "over-reconstruction" in areas covered by large Gaussians and that must be split in two.

A tile-based rasterizer sorts Gaussian splats for an entire image at a time. Screen is split into $16x16$ tiles, and 3D Gaussians are culled against the view frustum for each tile. Radix sort algorithm is used to sort the Gaussians according to the depth. Colors and α values are accumulated by traversing lists front-to-back

Some works have extended 3DGS to X-ray imaging. X-Gaussian [5] modify 3DGS to synthesize novel-view X-ray projections. DDGS-CT [11] improve X-Gaussian by considering complex noise-inducing physical effects. However, they cannot extract 3D density volumes from trained Gaussians.

2.2 LCO Cathode Batteries

In this section, we focus on the object of our investigation–the active material of a battery–, and briefly describe the chemical reactions that happen during charging/discharging cycles inside a Lithium-ion battery.

We are analyzing the active material contained in the cathode of a high-energy battery. The atomic formula is $LiCoO_2$, a compound rich of Lithium atoms that serves as the source of energy during the charging/discharging cycles. The chemical composition of the battery is very similar to Kimura et al. [18]; in this study, they show how solid-state-batteries (SSB) can be damaged by some chemical reaction that take place in the electrodes, visualizing the reaction during charge and discharge (*operando imaging*). The size and shape of the active material is studied in Ariyoshi et al. [2], where the electrochemical properties are studied by analyzing the morphology (shape and size) of the NCM active material. The morphology of our LCO crystals is referenced in [2] as "Large Oct", as illustrated in Figure 1.1.

During the discharging phase, lithium ions move from the anode to the cathode, and electrons flow through the external circuit, providing power to the device. In the anode, typically made of graphite, lithium ions are extracted and move toward the cathode; at the same time, electrons are released and flow through the external circuit.

$$\mathrm{Li}_x\mathrm{C}_6 \rightarrow x\mathrm{Li}^+ + xe^- + \mathrm{C}_6 \tag{2.6}$$

In Eq. 2.6, we denote with x the percentage of Lithium currently present in the anode. In the cathode ($LiCoO_2$), lithium ions and electrons are inserted

into the atomic structure.

$$\mathrm{Li}_{1-x}\mathrm{CoO}_2 + x\mathrm{Li}^+ + x\mathrm{e}^- \rightarrow \mathrm{LiCoO}_2 \tag{2.7}$$

During charging, lithium ions move from the cathode back to the anode, and electrons flow through the external circuit to the anode. In the cathode, lithium ions are extracted and navigate towards the anode through the electrolyte; electrons are released and flow through the external circuit to the anode (see Eq. 2.8).

$$\mathrm{LiCoO}_2 \rightarrow \mathrm{Li}_{1-x}\mathrm{CoO}_2 + x\mathrm{Li}^+ + x\mathrm{e}^- \tag{2.8}$$

In the anode, lithium ions and electrons are inserted into the graphite (see Eq. 2.9).

$$x\mathrm{Li}^+ + xe^- + \mathrm{C}_6 \rightarrow \mathrm{Li}_x\mathrm{C}_6 \tag{2.9}$$

To sum up, during charging and discharging phases, the Li^+ ion movement takes place: Li^+ ions move from the anode to the cathode; during charging, they move back to the anode. Meanwhile, electrons flow though the external circuit providing electrical energy during discharging.

2.3 XAFS

XAFS (X-ray Absorption Fine Structure) is a powerful tool for the structural analysis of materials [14]. In our study, we use this technology to study cathode materials in Lithium Cobalt Oxide (LCO) batteries.

In the typical scenario, an X-ray source generates X-rays at different energy wavelengths that interact with the imaged object. At the atomic level, ejected photoelectrons interact with nearby atoms, and based on this interference, X-rays are absorbed by the material. By studying the oscillations of the absorption coefficient $\mu(E)$, where E is the initial energy of the X-rays,

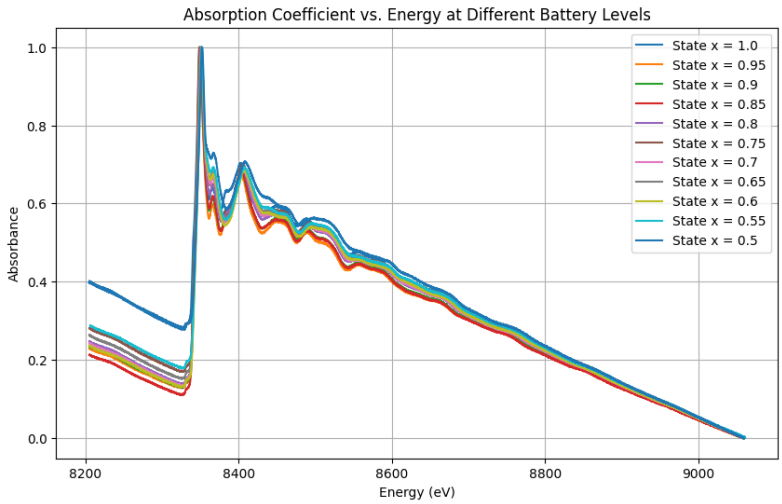

Figure 2.1: XAS diagram showing the absorption coefficient at different energies and charging levels

distances between atoms can be inferred [14].

Fig. 2.1 shows the X-ray Absorption Spectrum (XAS) of our LCO active material inside the cathode. On the x-axis there is the energy level measured in electron-volts (eV); on the y-axis there is the absorption coefficient $\mu(E)$, measuring how strongly the material absorbs X-rays at a given energy.

We can identify several regions that are based on the ionization threshold of the material:

- **Pre-edge** This is the lower energy region before the edge. Provides details about electronic structure and chemical environment
- **XANES** (X-ray absorption near edge structure): this is near the absorption edge where features arise due to electronic transitions and local atomic arrangements. Features give information about the oxidation state, coordination environment, bond type
- **EXAFS** (Extended X-ray absorption fine structure): energies are beyond the edge, and oscillations occur due to interference between the outgoing photoelectron wave and scattered waves from neighboring atoms.

This region provides structural information about distances and arrangements of atoms around the absorbing atom.

We can clearly identify the absorption edge at ~ 8348.519 eV. The edge is a sharp rise in absorption, and it is where atoms absorb X-rays and eject core electrons. We also notice that the edge remains almost constant at different energy states of the battery. In our experiments, we perform one-energy level analysis considering the energy level that corresponds to the maximum peak in the absorption coefficient, analyzing a full-charged battery (in Fig. 2.1, state $x = 1$).

XAFS analysis is particularly useful because it allows us to clearly identify the edge band on the x-axis (see Fig. 2.1), corresponding to the highest peak in absorption. Indeed, to better isolate the high-density regions in the volumes, we aim to capture the highest contrast in the images, that is achieved exactly in the edge-band region. On the other hand, hitting the imaged object using wavelengths outside the edge-band regions may produce less high-contrast images due to lower absorption. Essentially, by studying the XAFS spectrum, we can clearly identify the most suitable x-ray frequency to produce high-contrast images and then 3D reconstruct the model.

Chapter 3

Methodology

3.1 Acquisition setup

3.1.1 Optical Path

In this section, we describe the mechanism for converting X-rays into pixel intensities. Fig. 3.1 shows the optical path diagram.

Initially, X-rays are generated and interact with the imaged object; transmitted X-rays reach the fluorescent screen, that converts them into visible light. Then, an objective lens, an imaging lens, and a mirror, focus the light beam and direct it towards the detection system. A highly-sensitive camera captures the visible light and produce high-resolution X-ray images.

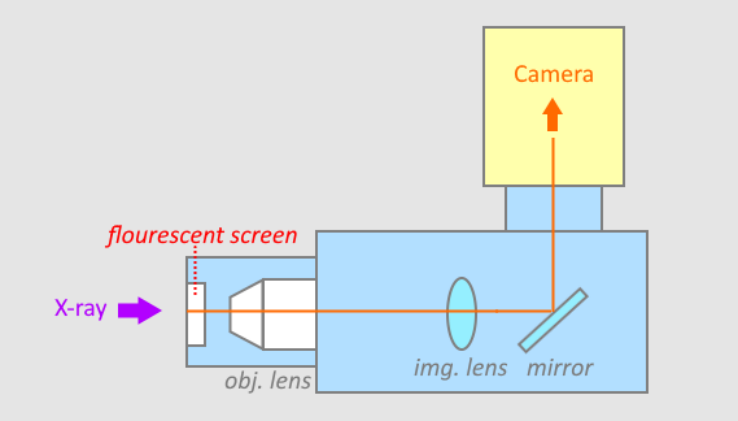

Figure 3.1: Optical Path Diagram

Figure 3.2: ORCA-Quest qCMOS (Hamamatsu Photonics, C15550-22UP), a high-resolution, ultra-low noise, high-QE qCMOS camera. Image from Hamamatsu Photonics website [1].

3.1.2 qCMOS camera

An ultra-sensitive, low-noise qCMOS camera (ORCA-Quest, Hamamatsu Photonics, C15550-22UP), Fig. 3.2, was used to capture high-resolution images. A quantitative CMOS (qCMOS) image sensor is capable of detecting multiple photoelectrons and distinguishing their numbers (photon counting) even in low-light conditions. The camera was used in "standard" mode, with a root-mean-square (RMS) noise of 0.43 electrons at 120 fps and a resolution of 4094×2304 pixels (~ 9.4 megapixels).

The peak quantum efficiency (QE) is 85%. QE is the ability to convert incoming photons into electrons and it varies according to the wavelength of the incident light. More formally, QE can be measured as follows:

$$\mathrm{QE} = \frac{\#\,\text{photoelectrons generated}}{\#\,\text{incident photons}} \times 100 \tag{3.1}$$

ORCA-Quest features photon number resolving; it is an advanced method of measuring light by counting photoelectrons. In order to provide accurate measurements, the camera noise must be sufficiently smaller than the amount of photoelectron signal. RMS noise indicates how much uncertainty or random fluctuations exists in the signal, and it is measured in terms of the smallest detectable unit, the electron.

[1] Source: `https://camera.hamamatsu.com/jp/ja/product/camera/C15550-22UP.html`

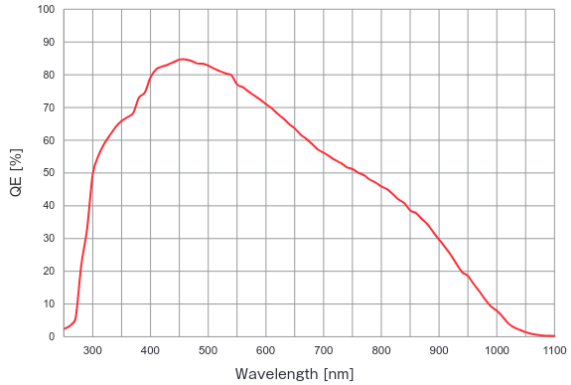

Figure 3.3: Quantum efficiency (QE) is plotted spanning over different wavelengths. A Peak QE of 85% is reaches at $\sim$ 460nm (blue spectrum). Figure from the camera catalog on Hamamatsu Photonics website.

In Figure 3.3, we see the quantum efficiency of our camera spanning over different wavelengths.

3.2 X-ray Novel Datasets

A series of acquisitions are performed and X-ray projections are captured. We see the acquisition phase as an incremental process toward high-quality projections. After carefully inspecting images of early acquisitions, we identify issues that cause image degradation, obfuscate the main features, give rise to artifacts and noise, and use such knowledge to perform better acquisition the subsequent times.

Since we want to prioritize high-speed acquisition, the camera is left with shutter open for most of the time, and at every fixed interval, an external trigger is activated and the image is captured.

We notice that a reduction in number of captures increases the exposure time for each acquisition, leading to more motion blur. On the contrary, a more frequent acquisition rate naturally reduces the exposure time and blurriness.

3.3 Preprocessing

Early inspections Raw data is stored as a single *tif* file of approximately 30GB of memory. Such file contains 1800 projections (grayscale images) of the imaged object, each of size 2048×2048. Values are store as 32 bit floating point numbers.

To start with, for all our datasets, we draw some statistics among the images, including min/max/average values, variance, sum of the absolute values, the presence of non-data regions, and the entropy.

We then spend time visualizing data and inspecting the presence of noise, blurriness, occlusions, undesired effects. We indeed notice visual anomalies especially in early acquisition datasets.

One main issue includes the presence of floaters appearing in the foreground for a limited number of consecutive projections and disappearing in the others. This effect limits the capacity of our reconstruction methods, since only few projections can be used to identify the location of the floater.

We notice also a recurrent problem in early acquisitions. The field-of-view of the acquisition system seems too narrow and focused in the middle of the object, limiting ourselves to a partial reconstruction; on the other hand, borders and regions close to the edges are not captured during acquisition.

We observe also a weird-looking effect and we call it "ghost effect". While inspecting the images as the object rotates, a fog-like material emerges from the left and quickly disappears on the right in a few frames. This suggests us that there may be an object occluding the battery that alters the intensities captured by the detector. We note the effect and investigate the causes. Eventually, it turns out that there is indeed an occluding object during acquisition and, after removing it, we notice how it is absent in the subsequent recording sessions.

In early acquisitions, to mitigate the presence of highly noisy images, we treat as noise low intensity values and we set to zero all the pixels below a

threshold τ. This way, we try to retain the most important features, e.g., stone-like objects and discard minor features. Additionally, we try to find a smaller angle range so that the object is clearly visible (e.g., 30-40 degrees), and treat these projections as training images.

After a comprehensive visual and statistical inspection, we prepare data for training. Images are resized to 256×256 using bicubic interpolation and normalized in desired range (e.g., 0-1).

Training projections One of the main objectives of our study is to achieve high-quality reconstruction while minimizing the number of training images required. We try different strategies to select and preprocess input projections and compare the reconstructed volumes. We assess which input configurations provide high-quality reconstruction without excessive computational cost.

3.4 SoTA 3D Reconstruction Methods

3.4.1 SaxNeRF

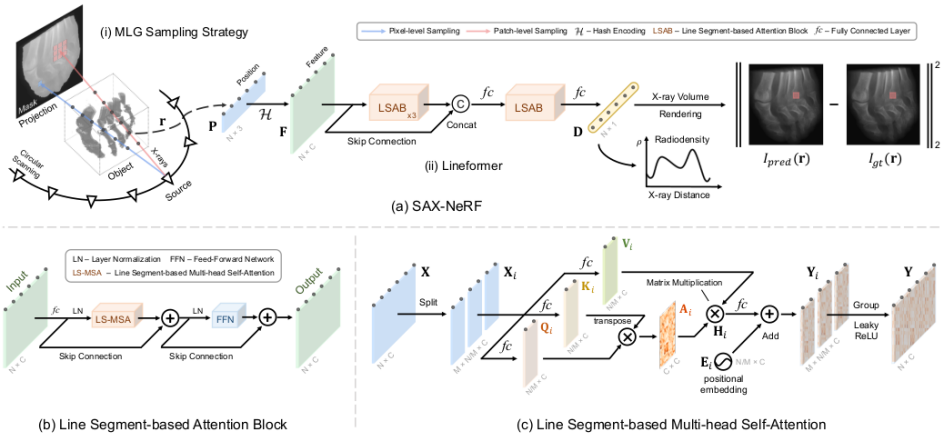

Figure 3.4: Training pipeline of SaxNeRF and neural layers. Figure reproduced from Cai et al. [6].

Fig. 3.4 depicts the training pipeline and the configuration of the neural layers. A sampling technique called Masked Local-Global (MLG) sampling is

used to sample batches $\mathcal{R}$ of X-rays hitting the imaged object during training. On each of these rays $\mathbf{r} \in \mathcal{R}$, $\mathbf{N}$ point positions $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\} \in \mathcal{R}^{N\times 3}$ are sampled and fed into a neural block called Lineformer. It is based on a basic unit named Line Segment-based Attention Block (LSAB).

The radiodensity field is modeled as

$$F_{\Theta_L} : (x, y, z) \rightarrow \rho, \tag{3.2}$$

where F_{Θ_L} is the mapping function of the neural network with weights Θ_L. Differently from the original NeRF formulation, the color information is not included in the output; input viewing direction is also excluded as the radiodensity on the point position only depends on the 3D location in the scene.

The ground truth intensity value $I_{gt}(\mathbf{r})$ for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \in \mathcal{R}^3$ can be modeled via the Beer-Lambert law.

$$I_{gt}(\mathbf{r}) = I_0 \cdot \exp\left(-\int_{t_n}^{t_f} \rho(\mathbf{r}(t))\, dt\right) \tag{3.3}$$

where I_0 is the initial density, and t_n and t_f are the near and far bounds. After discretizing Eq. 3.3, we derive the predicted intensity $I_{pred}(\mathbf{r}) \in R$.

$$I_{pred}(\mathbf{r}) = I_0 \cdot \exp\left(-\sum_{i=1}^{N} \rho_i \delta_i\right) \tag{3.4}$$

In Eq. 3.4, ρ_i is the density predicted by the model for sample i and $\delta_i = ||\mathbf{p}_{i+1} - \mathbf{p}_i||$ is the distance between adjacent points.

The training loss is simply the minimization of the total squared error $\mathcal{L}$ between the predicted and ground-truth intensities in the training X-ray batch $\mathcal{R}$.

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} ||I_{pred}(\mathbf{r}) - I_{gt}(\mathbf{r})||_2^2 \tag{3.5}$$

Line Segment-based Transformer As shown in Fig. 3.4, the position $\mathbf{P}$ of sampled rays goes through a hash encoding layer $\mathcal{H}$ to produce point feature

$\mathbf{F} \in \mathcal{R}^{N \times C}$. $\mathbf{F}$ is passed through four LSABs with a skip connection and two *fc* layers to derive the point radiodensity $\mathbf{D} \in \mathcal{R}^{N}$.

In the LSAB blocks, self-attention within each line segment is computed. The first step is to partition the point feature into M segments as

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M]^T \tag{3.6}$$

where $\mathbf{X}_i \in \mathcal{R}^{\frac{N}{M} \times C}$ and $i = 1, 2, \ldots, M$. Each $\mathbf{X}_i$ is linearly projected into query $\mathbf{Q}_i \in \mathcal{R}^{\frac{N}{M} \times C}$, key $\mathbf{K}_i \in \mathcal{R}^{\frac{N}{M} \times C}$, and value $\mathbf{V}_i \in \mathcal{R}^{\frac{N}{M} \times C}$ by three *fc* layers. Multi-head self attention is employed by using k heads along the channel dimension. The self-attention within each head $\mathbf{H}_i^j$ thus becomes

$$\mathbf{H}_i^j = \text{Attn}(\mathbf{Q}_i^j, \mathbf{K}_i^j, \mathbf{V}_i^j) = \mathbf{V}_i^j \text{softmax}\left(\frac{\mathbf{K}_i^{jT}\mathbf{Q}_i^j}{\alpha_i^j}\right). \tag{3.7}$$

$\alpha_i^j \in \mathcal{R}$ is a learnable parameter that scales the inner product before the softmax. Then, the k heads are concatenated along the channel dimension and pass through an *fc* layer; only then are summed with the positional embeddings $\mathbf{E_i} \in \mathcal{R}^{\frac{N}{M} \times C}$. The *i*-th output $\mathbf{Y_i} \in \mathcal{R}^{\frac{N}{M} \times C}$ thus becomes

$$\mathbf{Y}_i = [\mathbf{H}_i^1, \mathbf{H}_i^2, \ldots, \mathbf{H}_i^k]\mathbf{W}_i + \mathbf{E}_i, \tag{3.8}$$

$\mathbf{W}_i \in \mathcal{R}^{C \times C}$ is the learnable matrix of the *fc* layer. The outputs are then grouped to get the final feature $\mathbf{Y}_i \in \mathcal{R}^{N \times C}$ as

$$\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_M]^T. \tag{3.9}$$

Complexity Analysis Authors analyze the computational complexity of the LS-MSA and derive the following formula:

$$\mathcal{O}(\text{LS-MSA}) = \frac{2NC^2}{k}. \tag{3.10}$$

Differently from the vanilla Transformer, LS-MSA is linear to N, significantly reducing computation cost. C is the channel length and k is the number of heads.

Masked Local-Global Ray Sampling The idea is to sample rays in informative regions that characterize the radiodensity property of the object, excluding rays hitting the background or uninformative regions.

A mask $\mathbf{M} \in \mathcal{R}^{H \times W}$ is used to segment the foreground regions. It is used a threshold $T \in \mathcal{R}$ on the original projection $\mathbf{I} \in \mathcal{R}^{H \times W}$ to obtain $\mathbf{M} = \mathbf{1}_{\mathbf{I} > T}$. Then, $\mathbf{M}$ is partitioned into a set $\mathbf{W}$ of non-overlapping windows of size $S \times S$. Thereby, a window is selected if it is entirely contained in the foreground masked region.

$$\mathcal{W}_f = \{\mathbf{w} \in \mathcal{W} \mid \mathbf{w} = \mathbf{1}_{s \times s}\} \tag{3.11}$$

$\mathcal{W}_f$ denotes the set of regions that are available for sampling. Then, N_l windows $\mathcal{W}_l = \{\mathbf{w_1}, \ldots, \mathbf{w}_{N_l}\}$ are randomly selected from $\mathbf{W}_f$, and rays are projected.

$$\mathcal{R}_l = \bigcup_{i=1}^{N_l} \bigcup_{p \in \mathbf{W}_i} \text{Ray}(p) \tag{3.12}$$

In Eq. 3.12, $\mathcal{R}_l$ denotes the set of rays that are selected for sampling, where Ray(p) maps a pixel p to its corresponding ray.

On top of that, pixel-level sampling is performed by randomly selecting N_g pixels from the foreground regions excluding the area of $\mathcal{W}_l$, to avoid repeated sampling.

$$\mathcal{R}_g = \bigcup_{p \in (\mathbf{M} - \mathcal{W}_l)} \text{Ray}(p). \tag{3.13}$$

$\mathcal{R}_g$ denotes the set of rays that are selected via global sampling. Overall, the rays selected for training are the union between $\mathcal{R}_l$ and $\mathcal{R}_g$.

$$\mathcal{R} = \mathcal{R}_l \bigcup \mathcal{R}_g. \tag{3.14}$$

3.4.2 R^2-Gaussian

The main idea of R2Gaussian [30] is to represent the target object with a group learnable 3D Gaussians $\mathcal{G}^3 = \{G_i^3\}_{i=1,\dots,M}$. Each Gaussian G_i^3 defines a local Gaussian-shaped density field, i.e.,

$$G_i^3(\mathbf{x} \mid \rho_i, \boldsymbol{\mu}_i, \Sigma_i) = \rho_i \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right), \tag{3.15}$$

where $\rho_i \in \mathcal{R}$, $\boldsymbol{\mu}_i \in \mathcal{R}^3$ and $\boldsymbol{\Sigma}_i \in \mathcal{R}^{3\times3}$ are learnable parameters representing respectively the central density, mean, and covariance. For optimization purposes, $\boldsymbol{\Sigma}_i$ is parametrized with the rotation matrix $\mathbf{R_i}$ and scale matrix $\mathbf{S}_i$:

$$\boldsymbol{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^\top \mathbf{R}_i^\top. \tag{3.16}$$

To find the density value σ at position $\mathbf{x} \in \mathbf{R}^3$, the density contribution of each Gaussian at location $\mathbf{x}$ is summed:

$$\sigma(\mathbf{x}) = \sum_{i=1}^{M} G_i^3(\mathbf{x} \mid \rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{3.17}$$

Initialization Initially, FDK[10] is used to reconstruct a low-quality volume. With a density threshold τ, empty spaces are excluded and M points are randomly sampled as starting Gaussian positions. The scale of each Gaussian is set as the nearest neighbor distances and there is no initial rotation. The central densities are queried from the FDK volume and empirically scaled down to compensate for the overlay between Gaussians.

Training Fig. 3.5 depicts the overall training pipeline. Initially, Gaussians are initialized from an FDK volume. Projections are rasterized for photometric

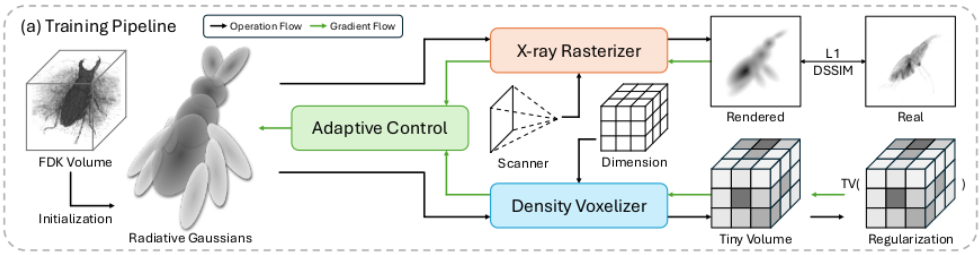

Figure 3.5: Overall training pipeline of R^2-Gaussian. Figure from the original work by Zha et al. [30].

losses and small volumes are voxelized for 3D regularization. Throughout the whole process, adaptive control is used to densify Gaussians for better representation. After training, density volumes are voxelized at target size.

Projections and composition To calculate the final intensity value $I_r(\mathbf{r})$ for a pixel along ray $\mathbf{r}$, authors suggest how an X-ray projection can be rendered by simply summing 2D Gaussians starting from the Beer-Lambert integration law and converting 3D Gaussians to 2D Gaussians by integrating along an axis.

$$I_r(\mathbf{r}) \approx \sum_{i=1}^{M} G_i^2(\hat{\mathbf{x}}|\sqrt{\frac{2\pi|\tilde{\Sigma}_i|}{|\hat{\Sigma}_i|}}\rho_i, \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) \tag{3.18}$$

In Eq. 3.18 the intensity pixel of ray $\mathbf{r}$ is computed as an approximation of the summation of M 2D Gaussians, where $\hat{\mathbf{x}} \in \mathbf{R}^2$, $\hat{\boldsymbol{\mu}}_i \in \mathbf{R}^2$, $\hat{\boldsymbol{\Sigma}}_i \in \mathbf{R}^{2\times2}$, $\tilde{\boldsymbol{\Sigma}}_i \in \mathbf{R}^{3\times3}$.

Voxelization Once the optimized set of Gaussians $\mathcal{G}^3$ are obtained, a voxelizer $\mathcal{V}$ maps them to a density volume $\mathbf{V} \in R^{X\times Y\times Z}$. The voxelizer first partitions the target space into multiple $8\times8\times8$ tiles. It then culls Gaussians, retaining those with a 99% confidence of intersecting the tile. In each 3D tile, voxel values are parallelly computed by summing the contributions of nearby Gaussians. The voxelizer is implemented in CUDA for fast computation.

Optimization Stochastic gradient descent (SGD) is used to optimize Gaussians. Photometric losses include L1 loss $\mathcal{L}_1$ and D-SSIM loss $\mathcal{L}_{ssim}$. A 3D total variation (TV)[25] regularization loss $\mathcal{L}_{tv}$ is included as a homogeneity prior. During each iteration, a small density volume is queried and its variation is minimized.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1(\mathbf{I}_r, \mathbf{I}_m) + \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}}(\mathbf{I}_r, \mathbf{I}_m) + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}}(\mathbf{V}_{\text{tv}}). \tag{3.19}$$

Eq. 3.19 represents the overall loss, where $\mathbf{I}_r$, $\mathbf{I}_m$, λ_{ssim} and λ_{tv} are respectively the rendered projection, measured projection, D-SSIM weight, and TV weight. Adaptive control removes empty Gaussians and densify (clone or split) those with large loss gradients. Large Gaussians are not pruned since they may be useful to represent large homogeneous areas like human organs. The densification process halves the densities of both the original and replicated Gaussians.

3.4.3 Postprocessing

After visualizing the 3D volume with Napari [22], we identify region of interests and crop cuboids that encapsulate them. To grasp an overall view of the battery, we remove the borders since they occlude the internal structures. We identify the top-view circumference of the battery and perform a vertical crop following the inscribed squared with some additional margin. The side of the inscribed square can be formulated as follows: $side = 2 \cdot r \cdot cos\theta$, where r is the radius of the circumference, and $\theta = 45°$.

Then, we perform a series of post-processing steps (see Fig. 3.6) to enhance the visualization and analysis of the stone-like particles. We wish to isolate specific regions of interest and improve the clarity of the internal structure of the battery.

First, normalization on the raw volume is applied, as voxels values are

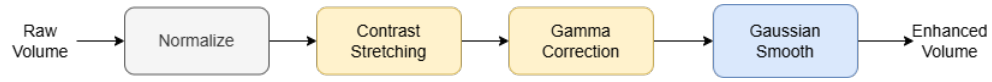

Figure 3.6: Postprocessing pipeline to enhance the voxelized volume and better visualize regions of interest.

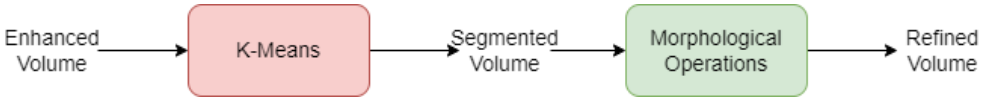

Figure 3.7: Clustering pipeline to obtain a refined segmented volume.

rescaled between 0 and 1. Then, we exclude outliers that may compromise the visualization: we apply left and right percentile thresholds to eliminate extreme values, ensuring that the majority of the data is properly scaled. The rescaling step is performed, and voxels values are rescaled within the adjusted range. Gamma correction is then applied to adjust the brightness and contrast, enhancing the visibility of finer details. Finally, a Gaussian filter is applied to the final volume to smooth out surfaces and get rid of noise and tiny artifacts.

Clustering To further isolate regions of interest, we perform unsupervised learning on the enhanced volumes using the K-Means [19] clustering algorithm (see Fig. 3.7). This step enables the separation of clusters corresponding to different density regions. Using the "elbow" rule we identify the optimal number of clusters to best capture the density variations.

To refine the clustered results, morphological operations are applied, including binary open and binary close operations. These operations are based on the erosion and dilation of the segmented volume. In this way, we fill small gaps and voids within the segmented objects via hole filling, creating a continuous representation of the stone-like structures. This also naturally includes the removal of small and irrelevant features.

As demonstrated in our experiments, these post-processing steps significantly improved the quality and interpretability of the reconstructed volumes, making it possible to analyze the internal structure of the objects with greater precisions.

Chapter 4

Experiments

4.1 Dataset Analysis

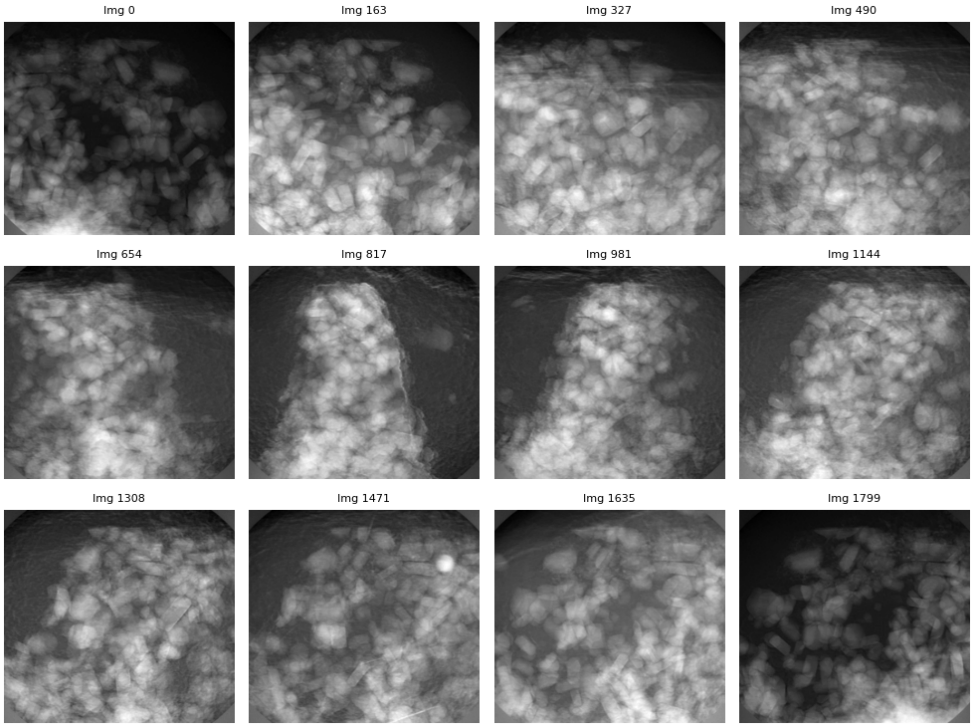

Figure 4.1: An insight of our battery dataset, depicting different projections from all the 1800 views.

In Fig. 4.1 we report some images from our battery dataset. These are gray scale images with an original dimension of 2048x2048, that we rescale

to 256x256 for efficiency. White regions denote dense material (active material) and black regions denote low-density material. Since we are particularly interested in characterizing the shape and size of the active material, we aim to reconstruct the stone-like object as faithfully as possible. Views are captured along a semi-circumference around the battery using the acquisition machinery described in the Acquisition setup section.

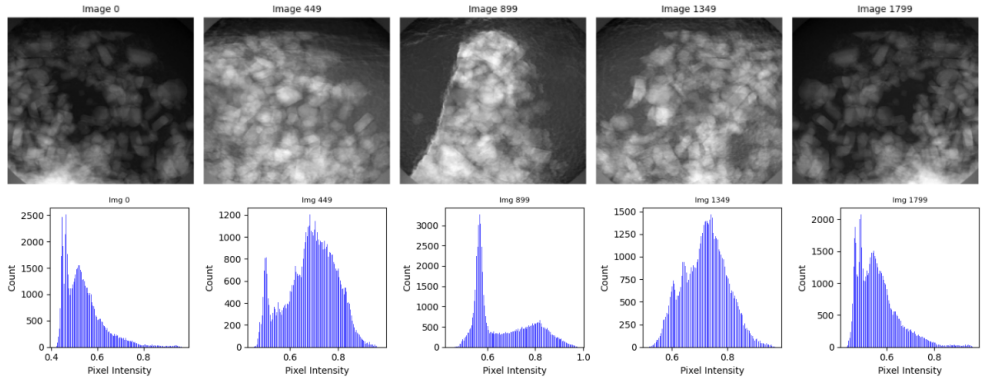

Figure 4.2: Top raw: some samples of views from the battery dataset; bottom: their histogram distributions.

In Fig. 4.2 we report some views and their relative pixel value histograms. From certain viewpoints, less dense material may be captured, resulting in a histogram shifted to low values; on the other hand, capturing denser material results in a distribution shifted to middle-to-high values.

While all the dense material tend to lay in the middle of the projections, there is material that quickly enters the field-of-view and leaves, and such effect is present in a limited consecutive amount of projections. After visual inspections, we notice that such kind of material appears only on the opposite side of the rotation angle. Consequently, there is less chance for the model to reconstruct this material due to a limited amount of projections capturing this phenomenon.

Previous acquisitions In previous acquisitions, we noticed anomalies that we successfully removed recalibrating the acquisition machinery and hardware setup. Anomalies included:

- a black narrow band on the left side of the images, limiting the field-of-view and occluding active materials
- a "ghost" effect in a series of images, appearing from right to left, that caused out-of-distributions images with higher pixel intensities

Both issues were solved and the "battery" dataset we used for our experiments is free from these defects.

In previous acquisitions, noisy artifacts were present in views and limited the reconstruction capabilities of our models. For this reason, we used filtering methods (e.g., Gaussian filter) to smooth out the projections and thresholding schemes, setting to zero lower-intensity pixels.

4.2 Experiment 1 - Fixed Sampling

We compare the performance of R2Gaussian and SaxNeRF while gradually reducing the number of training projections from each experiment. In fact, the ability to reconstruct a sufficiently-accurate 3D volume with a low number of input images is one of our main objectives. Given the original 1800 projections, we pick equally-spaced projections while increasing the step angle for each experiment. The training time for each experiment is kept constant for fairness among methods and we pick projections from 0 to 180 degrees. We then perform a qualitative assessment inspecting the internal slices side-by-side and compute the relative metrics according to ground-truth data.

For R2Gaussian, we train for 30000 iterations, as it was performed in the original paper; regarding SaxNeRF we fix a train time of 3 hours as we saw little improvement after this training time in previous experiments.

Qualitative Analysis In Fig. 4.3 we show the 3D visualization of the reconstructed volumes for both R2Gaussian and SaNeRF. We notice how all 45, 90, and 180 versions can successfully reconstruct all the active material (red

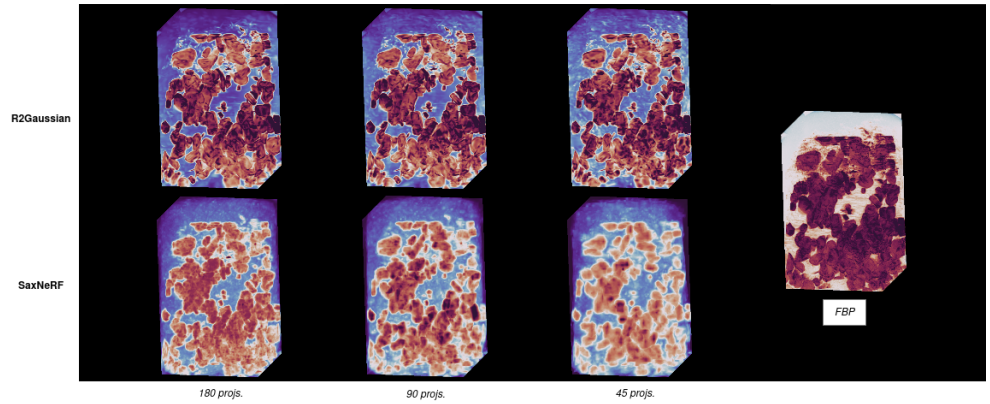

Figure 4.3: 3D visualization of GT and reconstructed volumes by R2Gaussian and SaxNeRF among different numbers of training projections (Napari, "twilight shifted" color map); on the right, the volume reconstructed with FBP using 1800 projections.

regions). Regarding R2Gaussian (top row), we observe that the 45 version suffers from noisy artifacts in the background regions, whereas the 90 projections alternative is more robust. When it comes to SaxNeRF (bottom row), when we reduce the number of projections, the active regions becomes much blurrier and quickly lose sharp contours. Unwanted material appears also in the void region. For visualization purposes, we used color enhancement so we could clearly inspect the stone-like material.

In Figure 4.4, we see a comparison between volumes produced by FBP and R2Gaussian. We notice how the GT volume is darker, as FBP tends to produce smoother and more uniform reconstructions. Moreover, volumes reconstructed with R2Gaussian are typically in a different intensity range compared to the GT. Since volumes are then normalized, they may appear brighter or clearer. The different in brightness does not inherently indicate a problem: it is indeed the result of different reconstruction algorithms, methods, or intensity scaling. The key here is evaluate the structural accuracy and feature preservation rather than absolute intensity. Since the GT volume clearly shows the internal structure and features, the difference in brightness is not a significant issue.

Commenting Fig. 4.4, we notice a progressive degradation in the quality of

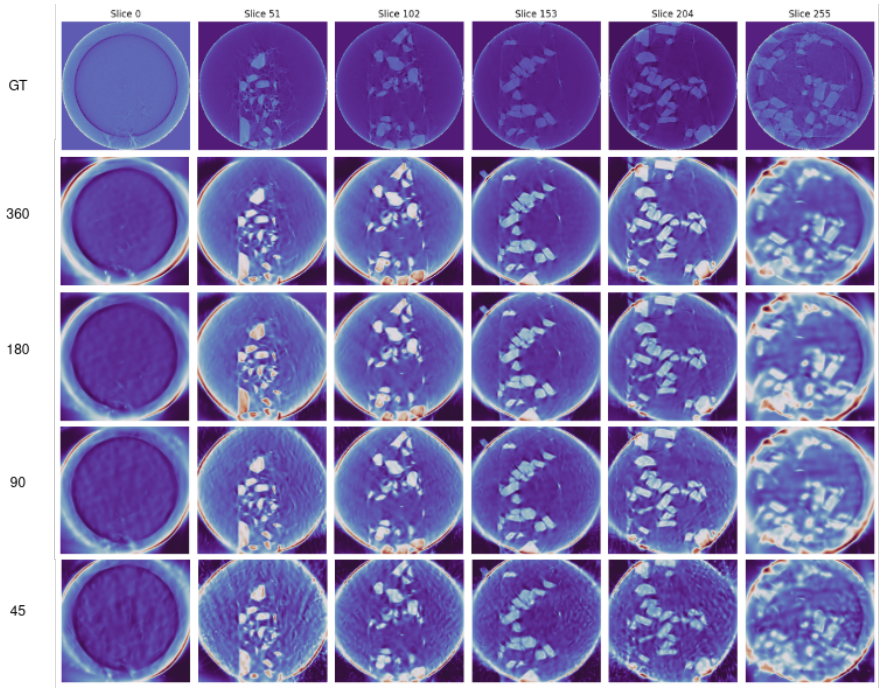

Figure 4.4: Comparison of internal slices produced by R2Gaussian (top-bottom) among different number of training projections (shown on left); GT is the ground-truth volume reconstructed with FBP with 1800 projections.

the reconstruction when we reduce the number of projections. Noise consists of misplaced Gaussians (a.k.a. *floaters*) that appear in the void regions. We notice that the model trained with 360 projections very closely approximate the reference volume; active material is well reconstructed and isolated from the background, with sharp edges, and the circular boundaries of the battery are also reconstructed.

When we compare the 360 and 180 versions, we do not notice a significant degradation in quality reconstruction, suggesting that 180 may be a good candidate number for further reconstructions. Intensities inside the active regions and in the boundaries are comparable with the 360 alternative.

When it comes to the 90 and 45 versions, *floaters* start appearing in the background regions, yet the active material is still well-reconstructed. Also in the 45 alternative, all the stone-like features are preserved and have sharp edges.

Regarding slice 255 (top slice), all models struggle during reconstruction

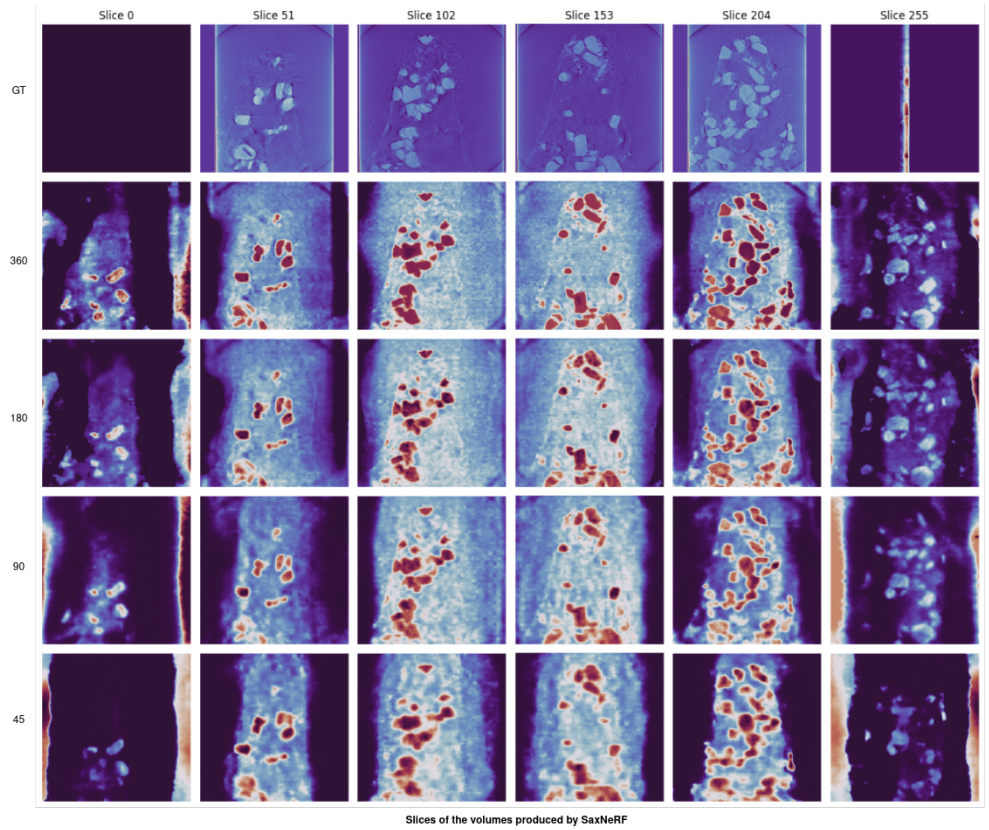

Figure 4.5: Comparison of internal slices produced by SaxNeRF (left-right) among different number of training projections (shown on left); GT is the ground-truth volume reconstructed with FBP with 1800 projections.

and we can see blurriness and lose details. This is probably caused by the acquisition setup, where projections well-depict the whole central region of the battery while culling the bottom and high edges.

In Fig. 4.5 we report a slice comparison of the volumes produced by SaxNeRF with different number of training projections as input. As expected, when we feed a higher number of projections (360), active material tends to be clustered in high-defined regions, with high contrast on the background. When we reduce the number of training images, the active regions becomes blurrier and lose well-defined shape. We notice, however, that there is very little difference between the 180 and 360 version. When it comes to the 90 alternative, most of the high-density regions are also preserved. We start losing details and consistency only in the 45 projections case.

Model	N. projs	Step angle	PSNR	SSIM	SNR	CNR
R2Gaus	360	0.5	**22.097**	**0.692**	**20.243**	**15.445**
SaxNeRF	360	0.5	19.374	0.531	18.936	13.345
R2Gaus	180	1	21.557	0.671	20.145	15.363
SaxNeRF	180	1	18.755	0.520	18.675	12.660
R2Gaus	90	2	21.786	0.669	19.889	15.008
SaxNeRF	90	2	17.803	0.459	16.723	10.733
R2Gaus	45	4	22.094	0.638	18.951	13.988
SaxNeRF	45	4	17.482	0.406	15.000	9.160

Table 4.1: Performance of R2Gaussian and SaxNeRF with different number of input views.

Quantitative Analysis In Table 4.1 we show the performance of R2Gaussian and SaxNeRF among different input settings. Regarding R2Gaussian, we obtain higher performance when we provide in input more projections (360). However, we do not notice a significant degradation in performance when we start reducing the number of training views, as demonstrated by the 180 and 90 settings. While changing the number of input projections, R2Gaussian always outperforms SaxNeRF according to all our metrics. On the contrary, SaxNeRF loses consistency especially in the case of low number of input images (45).

4.3 Experiment 2 - Narrowing Angles

In this experiment we gradually narrow down the angle of projection; from a full 0-180 degrees, we restrict to 30-150, 45-135, 60-120. While doing this, we distinguish between two cases: in (A) we decrease also the number of projections; in (B) we keep constant the number of projections.

Case A As depicted in Fig. 4.6, we gradually decrease the number of projections and feed them to R2Gaussian. We assess how narrow-wide angles affect the reconstruction of the volume.

In Fig. 4.7, we show a visual comparison of the internal slices among different ranging sectors. We observe how the angle range particularly affect the

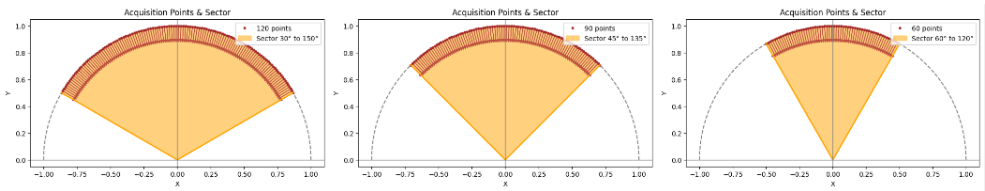

Figure 4.6: Training angles used during training. We narrow down the angle sector and reduce the number of projections.

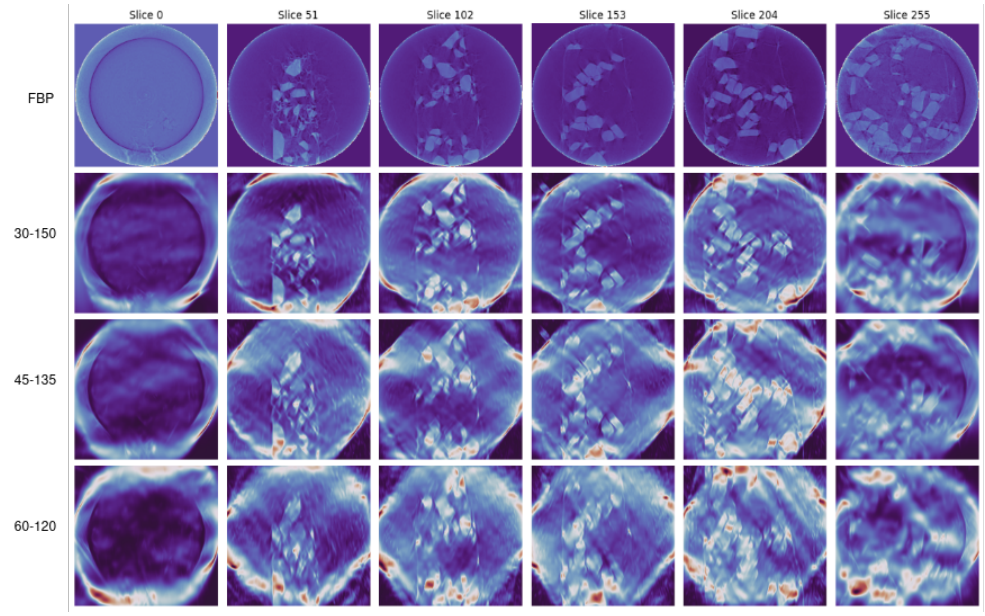

Figure 4.7: Slice comparisons of volumes while narrowing the sector angle; on the left we report the sector range in degrees. (Training setting: 120 projs. were used in the 30-150 deg. experiment; 90 projs. were used in the 45-135 deg. experiment; 60 projs. were used in the 60-120 deg. experiment.)

reconstruction performance. In this setup, we trained with 120, 90, and 60 projections as we narrowed down the sector range. We notice that while reducing down the angle view, the quality of the reconstruction volume significantly drops. The active material gradually gets blurrier and *floaters* start appearing in the background regions; circular boundaries are also progressively lost.

This qualitative insight suggests that the reconstruction procedure is highly affected by the sector angle. While the number of projections is not a crucial factor, as evidenced in experiment 1, reducing the sector angle highly affect the reconstruction, loosing quality and consistency.

In Table 4.2 we show the performance of R2Gaussian with different angle

Model	N. projs	Angle range	PSNR	SSIM	SNR	CNR
R2Gaus	120	30-150	**21.340**	**0.660**	**16.608**	**10.944**
R2Gaus	90	45-135	20.881	0.607	14.118	7.485
R2Gaus	60	60-120	20.603	0.601	12.034	4.407

Table 4.2: Performance of R2Gaussian with different angle ranges and number of projections.

ranges and number of projections. As expected, while narrowing the angle range and reducing the number of projections, performance tends to decrease.

Case B We fix the number of projections and gradually reduce the angle range. This setting evaluates the impact of the sector angle on the reconstruction process.

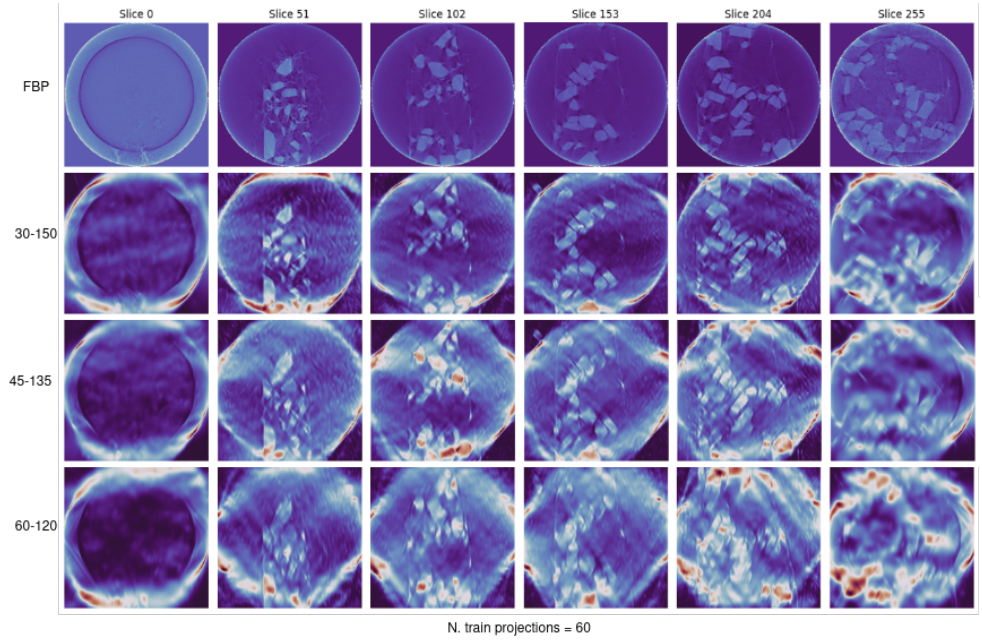

Figure 4.8: Slice comparisons with 60 train projections and different angle segment.

We notice that while keeping the number of training images constant (in Fig. 4.8, the number of projections is 60), and narrowing down the sector angle range, we notice a significant impact on quality reconstruction, the volumes gradually becoming blurrier.

This insight further suggests us that a wide angle of capturing is crucial for a correct and high-quality reconstruction.

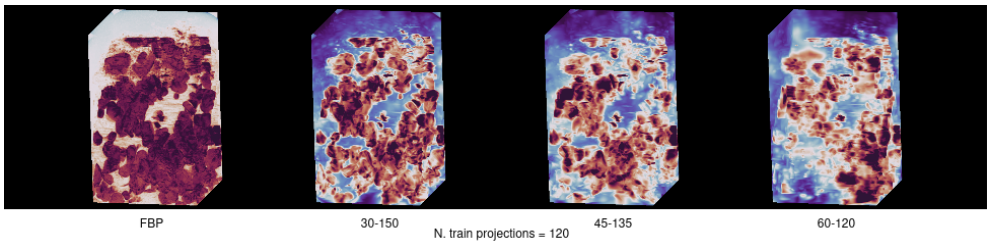

Figure 4.9: Here we report a 3D visual comparison of an internal crop of the volumes reconstructed by R2Gaussian. Here we used 120 train projections for each volume and varying sector angle ranges (shown at the bottom of each image).

Model	N. projs	Angle Range	PSNR	SSIM	SNR	CNR
R2Gaus	60	30-150	21.351	0.628	16.261	10.927
R2Gaus	60	45-135	20.579	0.590	14.301	7.650
R2Gaus	60	60-120	21.583	0.600	12.011	4.318
R2Gaus	120	30-150	**22.026**	**0.673**	**17.155**	**11.541**
R2Gaus	120	45-135	21.537	0.636	14.665	7.980
R2Gaus	120	60-120	21.007	0.608	11.972	4.445

Table 4.3: Performance of R2Gaussian with different angle ranges and number of projections.

In Table 4.3 we show the performance of R2Gaussian with different angle ranges and number of projections. The angle range seems to play the most significant impact on performance, hindering the reconstruction quality. It is worth noting that while the number of projections is the same, just having them all close together or far apart plays a crucial role during reconstruction.

4.4 Experiment 3 - Random Sampling

In the third experiment, we consider a full 0-180 projection angle but pick training views at random angle intervals. We also reduce the number of projections and directly compare the results with experiment 1, that assumes a fixed step angle between projections.

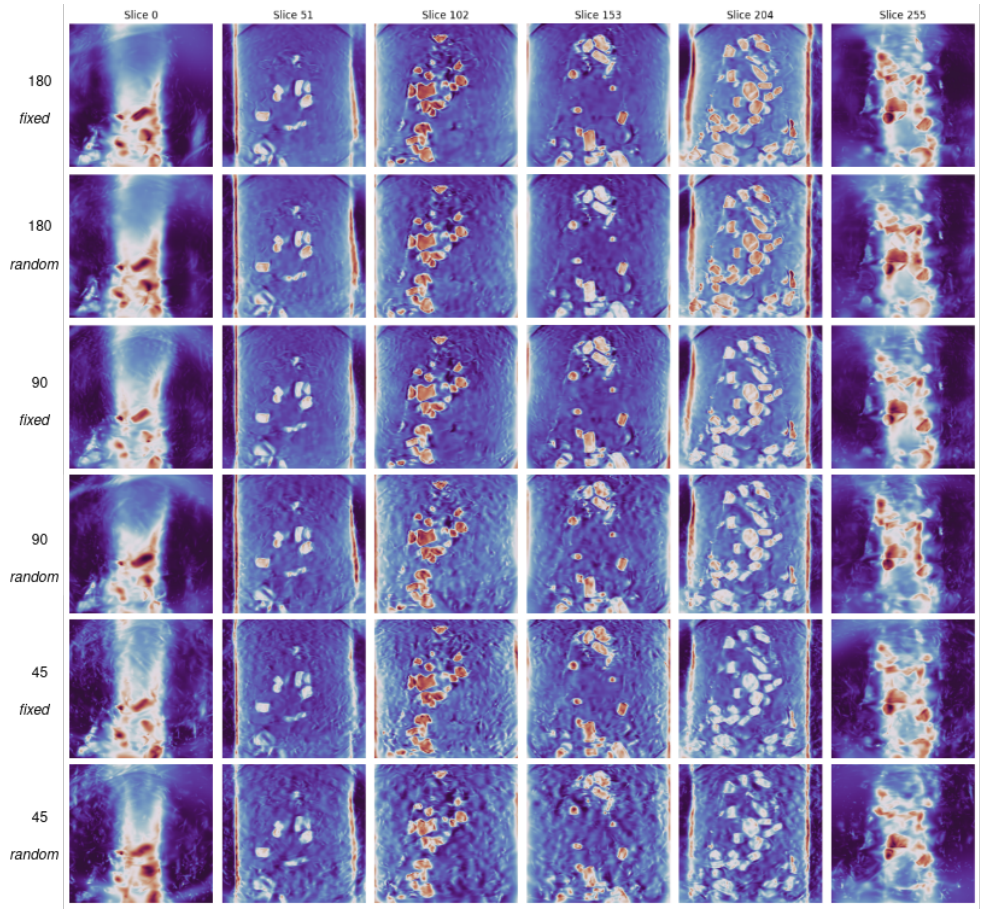

Figure 4.10: We show a slice comparison among each training setting. On the left side we note the number of projections used (45, 90, 180) and the angle scheme (fixed or random).

Qualitative Analysis In Fig. 4.10 we show a slice comparison side-to-side of the battery. We can compare how fixed and random angles perform while reducing the number of projections. As expected, unwanted Gaussians start appearing in both methods when we reduce the number of training images. Overall, it is difficult to tell whether fixed or random sampling is better than the other. In the 45 projections setting, we notice that random sampling tends to produce larger Gaussians in the void regions and loose shapes of the active materials. However, in the 180 and 90 settings, both method successfully reconstruct the main stone-like objects. Borders are also visible as vertical red lines at the side of the slices and are present in both methods.

Model	Sampling	N. projs	PSNR	SSIM	SNR	CNR
R2Gaussian	Random	360	22.360	**0.697**	20.014	15.157
R2Gaussian	Fixed	360	22.097	0.692	20.243	**15.445**
R2Gaussian	Random	180	**22.544**	0.694	19.791	14.847
R2Gaussian	Fixed	180	21.557	0.671	**20.145**	15.363
R2Gaussian	Random	90	21.856	0.653	18.903	13.949
R2Gaussian	Fixed	90	21.786	0.669	19.889	15.008
R2Gaussian	Random	45	21.265	0.608	17.724	12.633
R2Gaussian	Fixed	45	22.097	0.638	18.951	13.988

Table 4.4: Performance of R2Gaussian with both random and fixed projections in input.

Quantitative Analysis In Table 4.4 we report the performance of R2Gaussian among different sampling schemes and number of projections. We find that fixed sampling outperforms random sampling when the number of input projection is reduced (e.g., 45). However, when when the number of projections is sufficiently high (180, 360), random sampling may be preferable than fixed sampling.

4.5 Experiment 4 - Window Averaging

In this experiment, we average the input 1800 projections calibrating a window size and compare the results. In practice, we take batches of consecutive images and average them at pixel level, and pass the final averaged images to the model. In the first case (A), we set different averaging groups and reduce the number of training projections; in the second scenario (B), we set the averaging window but keep the total number of projections constant among experiments. Regarding the latter aspect, overlapping between projections is necessary to achieve a constant number of views for each experiment.

Case A In this scenario, we calibrate the window size and take averages among consecutive projections. According to Table 4.5, a window size of 10 gives a strong SNR and CNR on the active regions. On the other hand,

Model	N. projs	Average group	PSNR	SSIM	SNR	CNR
R2Gaussian	180	10	21.993	0.692	**20.050**	**15.202**
R2Gaussian	90	20	22.175	**0.694**	19.590	14.660
R2Gaussian	60	30	**22.290**	0.693	19.320	14.264

Table 4.5: Performance metrics for the R2Gaussian model with different average windows.

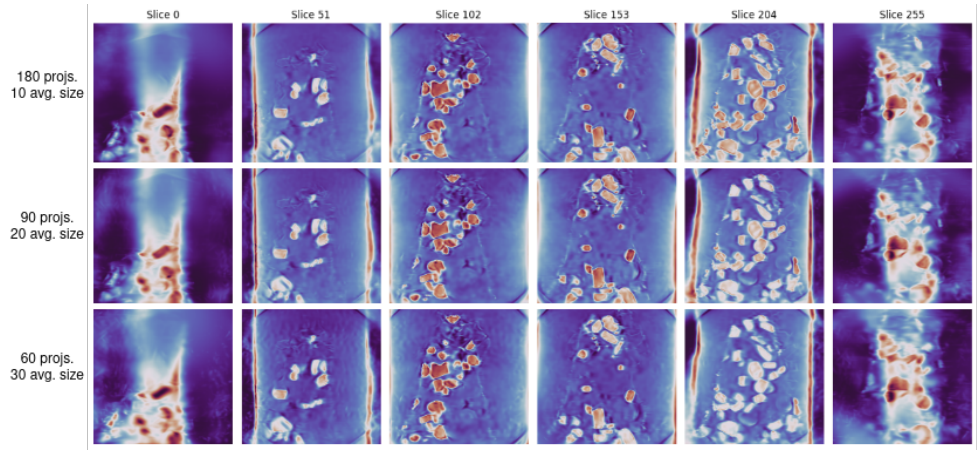

Figure 4.11: Slice comparison among different projection numbers and average windows.

widening the window size provides a higher PSNR and SSIM, suggesting that averaging is a good strategy to remove noise.

Visually inspecting the internal slices in Fig. 4.11, we notice that a too wide average window may cause noise and artifacts.

Model	N. projs	Average group	PSNR	SSIM	SNR	CNR
R2Gaussian	180	20	22.001	0.692	**19.880**	**15.000**
R2Gaussian	180	30	22.530	0.699	19.308	14.288
R2Gaussian	180	40	**22.760**	**0.704**	18.953	13.760
R2Gaussian	180	50	22.621	0.694	18.411	13.039
R2Gaussian	180	60	22.488	0.693	18.114	12.519

Table 4.6: Performance metrics for the R2Gaussian model with varying average group sizes.

Case B Commenting Table 4.6, an average group of 40 seems to lead to the highest PSNR and SSIM among the other sizes. Widening the average size, however, has the drawback of lowering the SNR and CNR in the active

regions.

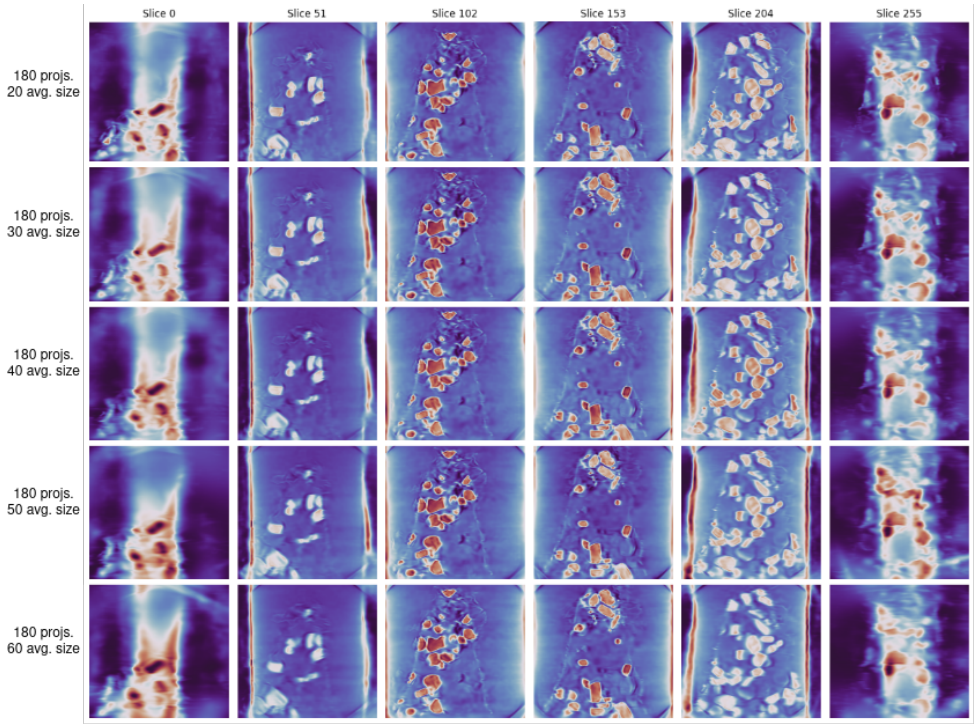

Figure 4.12: Slice comparison among different averaging methods and constant number of projections.

In Figure 4.12 we show a visual inspection of the different averaging methods. We find that visually spotting the differences between each variant is quite challenging. Indeed, the overlapping approach due to averaging windows turned out to be a good strategy to keep the number of projections constant and provide a good reconstruction quality.

Chapter 5

Discussion

R2Gaussian vs SaxNeRF In Experiment 1 we demonstrated that while changing the number of input images, R2Gaussian regularly surpass SaxNeRF in all test cases. This suggests that not only R2Gaussian is faster, but it produces high quality and less noisy volumes than SaxNeRF. On top of that, R2Gaussian is free from many hyper-parameters that SaxNeRF has, e.g., the near/far planes, the number of coarse and fine points to sample, the window dimension, the segment length, etc.

Low number of projections We demonstrated that 3D reconstruction is possible with sparse input, namely using a number of projections that is less than one order of magnitude of the original training data. In fact, while traditional algorithms typically require 1000+ training images to achieve a good reconstruction, our methods requires significantly less data and still achieves comparable, if not better, reconstructions.

Angle range We experimented with the sector ranges of acquisition, narrowing down the capturing angle and potentially speeding up the acquisition phase. We showed that while good reconstruction is still possible with an angle sector of less than 180 degrees, performance dramatically decreases if the angle range becomes too narrow. This insight suggests that to best reconstruct

an object, we need to have it in many views with different angles; reconstruction may be hindered by too aligned views or with small angle difference.

Angle step We compared fixed and random angles between each projections and compared the performances. We showed that there is not a predominant approach, and in most cases, both methods show similar performance.

Averaging With Experiment 4, we found that averaging projection is indeed an interesting approach to smooth out the volume, reduce noise, while keeping the active regions intact. We experimented with different averaging windows and assess that a medium-sized window is the best compromise.

We also noticed how the overlapping strategy among averaging windows is a good strategy to keep the number of projections constant and provide a good reconstruction quality.

Chapter 6

Conclusion

6.1 Summary

The goal of this thesis was to explore the effectiveness of 3D reconstruction from X-rays for inspecting lithium-ion batteries. By leveraging state-of-the-art (SoTA) methods, we addressed this application problem and demonstrated that 3D reconstruction in X-ray imaging is not only feasible but also achieves high precision using modern approaches, even with a limited number of input images.

We started by defining our task, that is 3D reconstruction of LCO batteries from multiple views around the object. We then defined several metrics to assess the quality of the reconstructed volumes with and without a reference volume. When a GT volume is provided, metrics traditionally applied to images (SSIM and PSNR) can be extended in the 3D domain; when a GT volume is not available, we engineered ad-hoc metrics proposed in previous works.

We then described the crucial difference between RBG and X-ray imaging, providing physics insights from the optical domain. These observations helped us to better comprehend the nature of X-ray data and guided our following experiments.

In the Background section, we briefly touched on traditional approached

for 3D reconstruction, including analytical and iterative methods. We then continued with modern SoTA models, covering NeRF and 3D Gaussian Splatting.

We then described the acquisition setup we used during our X-ray acquisitions, since we worked on novel X-ray datasets. Reconstruction methods were also illustrated in details, namely SaxNeRF and R2Gaussian Splatting. A postprocessing pipeline was also introduced to segment active material from the background.

We then proceeded with our experiments, using our models under different input settings and comparing the reconstruction results. We assessed results both qualitatively and quantitatively. Indeed, we visually inspected 3D volumes and internal slices with a 3D modern visualization software and compared the differences. Results were also provided in tables according to the metrics we defined in the Metrics section. For each experiment, we discussed the results and tried to understand the reason and behavior of each model.

In our experiments, we compared SaxNeRF and R2Gaussian using a fixed number of projections, and assessed that the latter is superior in all cases. When it comes to lowering the number of projections, R2Gaussian is still able to produce acceptable results, differently from traditional methods that require 1000+ train images. Regarding the capturing angle of the imaged object, we saw how it plays a crucial role in reconstruction and how a wide angle (~ 180 degrees) is necessary for a high-quality reconstruction. When dealing with the angle step, we found that there is no dominant approach between fixed-step or random-step angles between each projection. Finally, we discovered how averaging projections is a promising way to reduce noise in the empty regions and create sharp borders on the surfaces of the active regions.

6.2 Future Work

In this work, we performed 3D reconstruction in X-ray imaging relying on a single energy level; this approach limits the capacity to differentiate between similar structures and material contrast. A natural extension may be to leverage multiple energy levels, enhancing the characterization of materials with different absorption properties.

Indeed, different materials have different absorption behaviors depending on the energy input level. Combining multiple energies can help mitigate noise, artifacts, and ambiguities present in single-energy reconstructions. Consequently, we may achieve more precise estimations of density and composition, useful not only in battery analysis but also in other domains.

Bibliography

[1] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (sart): a superior implementation of the art algorithm. *Ultrasonic Imaging*, 6(1):81–94, 1984. DOI: 10.1177/016173468400600107.

[2] K. Ariyoshi and T. Tanaka. Single-crystal growth of lini1/3co1/3mn1/3o2 via the flux method and its electrochemical properties. *Crystal Growth & Design*, 24(9):3771–3776, 2024. DOI: 10.1021/acs.cgd.4c00101. eprint: https://doi.org/10.1021/acs.cgd.4c00101. URL: https://doi.org/10.1021/acs.cgd.4c00101.

[3] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: a multiscale representation for anti-aliasing neural radiance fields, 2021. arXiv: 2103.13415 [cs.CV]. URL: https://arxiv.org/abs/2103.13415.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: unbounded anti-aliased neural radiance fields, 2022. arXiv: 2111.12077 [cs.CV]. URL: https://arxiv.org/abs/2111.12077.

[5] Y. Cai, Y. Liang, J. Wang, A. Wang, Y. Zhang, X. Yang, Z. Zhou, and A. Yuille. Radiative gaussian splatting for efficient x-ray novel view synthesis, 2024. arXiv: 2403.04116 [eess.IV]. URL: https://arxiv.org/abs/2403.04116.

[6] Y. Cai, J. Wang, A. Yuille, Z. Zhou, and A. Wang. Structure-aware sparse-view x-ray 3d reconstruction, 2024. arXiv: 2311.10959 [eess.IV]. URL: https://arxiv.org/abs/2311.10959.

[7] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang. Hallucinated neural radiance fields in the wild, 2022. arXiv: 2111.15246 [cs.CV]. URL: https://arxiv.org/abs/2111.15246.

[8] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[9] A. Corona-Figueroa, J. Frawley, S. Bond-Taylor, S. Bethapudi, H. P. H. Shum, and C. G. Willcocks. Mednerf: medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray, 2022. arXiv: 2202.01020 [eess.IV]. URL: https://arxiv.org/abs/2202.01020.

[10] L. Feldkamp, L. C. Davis, and J. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1:612–619, June 1984. DOI: 10.1364/JOSAA.1.000612.

[11] Z. Gao, B. Planche, M. Zheng, X. Chen, T. Chen, and Z. Wu. Ddgs-ct: direction-disentangled gaussian splatting for realistic volume rendering, 2024. arXiv: 2406.02518 [cs.CV]. URL: https://arxiv.org/abs/2406.02518.

[12] P. Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of Theoretical Biology*, 36(1):105–117, January 1972. ISSN: 0022-5193. DOI: 10.1016/0022-5193(72)90180-4. URL: https://doi.org/10.1016/0022-5193(72)90180-4.

[13] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray

photography. *Journal of Theoretical Biology*, 29(3):471–481, December 1970. ISSN: 0022-5193. DOI: 10.1016/0022-5193(70)90109-8. URL: https://doi.org/10.1016/0022-5193(70)90109-8.

[14] W. Grünert and K. Klementiev. X-ray absorption spectroscopy principles and practical use in materials analysis. *Physical Sciences Reviews*, 0(0), 2020. DOI: 10.1515/psr-2017-0181.

[15] D. He, J. Zhou, X. Shang, X. Tang, J. Luo, and S.-L. Chen. De-noising of photoacoustic microscopy images by attentive generative adversarial network. *IEEE Transactions on Medical Imaging*, 42(5):1349–1362, 2023. DOI: 10.1109/TMI.2022.3227105.

[16] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001. DOI: 10.1137/1.9780898719277. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898719277. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898719277.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. arXiv: 2308.04079 [cs.GR]. URL: https://arxiv.org/abs/2308.04079.

[18] Y. Kimura, A. Tomura, M. Fakkao, T. Nakamura, N. Ishiguro, O. Sekizawa, K. Nitta, T. Uruga, T. Okumura, M. Tada, Y. Uchimoto, and K. Amezawa. 3d operando imaging and quantification of inhomogeneous electrochemical reactions in composite battery electrodes. *The Journal of Physical Chemistry Letters*, 11(9):3629–3636, 2020. DOI: 10.1021/acs.jpclett.0c00876. eprint: https://doi.org/10.1021/acs.jpclett.0c00876. URL: https://doi.org/10.1021/acs.jpclett.0c00876. PMID: 32315194.

[19] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[20] S. H. Manglos, G. M. Gagne, A. Krol, F. D. Thomas, and R. Narayanaswamy. Transmission maximum-likelihood reconstruction with ordered subsets for cone beam ct. *Physics in Medicine and Biology*, 40(7):1225–1241, July 1995. ISSN: 0031-9155. DOI: 10.1088/0031-9155/40/7/006. URL: https://doi.org/10.1088/0031-9155/40/7/006.

[21] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: representing scenes as neural radiance fields for view synthesis, 2020. arXiv: 2003.08934 [cs.CV]. URL: https://arxiv.org/abs/2003.08934.

[22] napari contributors. Napari: a multi-dimensional image viewer for python. https://github.com/napari/napari, 2019. DOI: 10.5281/zenodo.3555620.

[23] J. Radon, P. Parks, and C. Clark. On the determination of functions from their integral values along certain manifolds, April 2018.

[24] D. Rückert, Y. Wang, R. Li, R. Idoughi, and W. Heidrich. Neat: neural adaptive tomography, 2022. arXiv: 2202.02171 [cs.CV]. URL: https://arxiv.org/abs/2202.02171.

[25] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, November 1992. DOI: 10.1016/0167-2789(92)90242-F.

[26] K. Sauer and C. Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, 41(2):534–548, 1993. DOI: 10.1109/78.193196.

[27] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] E. Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53(17):4777–4807, September 2008. ISSN: 0031-9155. DOI: 10.1088/0031-9155/53/17/021. URL: https://doi.org/10.1088/0031-9155/53/17/021.

[29] M. Suhail, C. Esteves, L. Sigal, and A. Makadia. Light field neural rendering, 2022. arXiv: 2112.09687 [cs.CV]. URL: https://arxiv.org/abs/2112.09687.

[30] R. Zha, T. J. Lin, Y. Cai, J. Cao, Y. Zhang, and H. Li. R^2-gaussian: rectifying radiative gaussian splatting for tomographic reconstruction, 2024. arXiv: 2405.20693 [eess.IV]. URL: https://arxiv.org/abs/2405.20693.

[31] R. Zha, Y. Zhang, and H. Li. *Naf: neural attenuation fields for sparse-view cbct reconstruction*. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. Springer Nature Switzerland, 2022, pages 442–452. ISBN: 9783031164460. DOI: 10.1007/978-3-031-16446-0_42. URL: http://dx.doi.org/10.1007/978-3-031-16446-0_42.

Acknowledgements

I am grateful to my home university, the University of Bologna, for allowing me to participate in the 1st Call of 2024 for the NII International Internship Program in Tokyo. I would also like to sincerely thank my supervisor at NII, Prof. Imari Sato, for her invaluable guidance, support, and motivation throughout this journey.

I would also like to thank my advisor, Prof. Samuele Salti, for his continuous support and feedback throughout the writing of this thesis.

A special thanks goes to my colleague and friend, Ming-Yang Ho, for his essential help, knowledge, and inspiration.