

Simulating and Reconstructing X-Ray CT Projections

Reece Garthwaite

Abstract

I have chosen to simulate x-ray transmission through digital phantoms, using energy dependent attenuation from NIST data, to generate sinograms that I then reconstruct using back projection methods.

1 Introduction

Computed tomography (CT) scans are a form of medical imagery that uses x-rays to create images of the body. CT scans reconstruct cross sectional images of the body from multiple projections of x-ray scans at a range of angles. This project aims to deepen my understanding of medical imaging techniques through the simulation and reconstruction of CT projections.

This work particularly focuses on incorporating energy-dependent attenuation coefficients, derived from NIST data, to accurately model materials like bone and soft tissue, thereby allowing for the investigation of phenomena such as beam hardening.

2 Methods

2.1 Background Physics

2.1.1 X-ray generation

X-rays are produced when high kinetic energy electrons are accelerated towards a positive anode in a x-ray tube. Tungsten is a common choice for the anode (due to its high melting point and high atomic number). Electrons come close to nuclei of the target, causing a deceleration and change in direction, converting kinetic energy into a spectrum of electromagnetic radiation. Incident electrons can ionise the material, removing an electron from the anode. As the electron orbit vacancy gets filled by a orbital shell electron in a further out shell a photon is emitted. As orbital energies and their differences are unique in atoms, this leads to what we call "characteristic radiation". [1]

Figure 1 [2] shows a diagram of a x-ray tube . Which helps to visualise the origin of the x-ray spectrum as electrons are emitted via thermionic emission

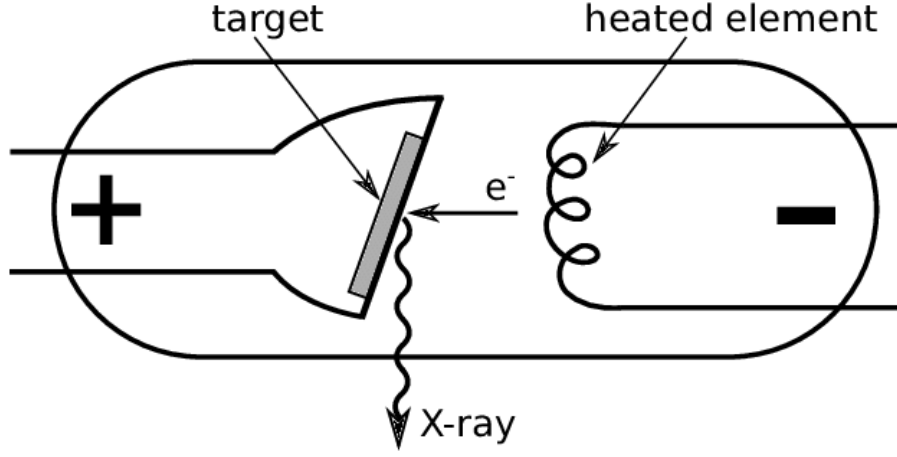


Figure 1: Schematic of a X-ray tube

and then accelerated toward a positively charged anode, where their interactions with the target produce both bremsstrahlung and characteristic radiation.

2.1.2 Attenuation of X-rays

As x-rays pass through matter, they are attenuated according to the Beer-Lambert law:

$$I(E) = I_0 \cdot e^{-\mu(E)x} \quad (1)$$

Where:

- I = transmitted intensity of radiation after passing through the material
- I_0 : Initial intensity
- μ : Linear attenuation coefficient
- x : Thickness of material

I have included μ to be energy dependent as this is what is observed in experiments, and allows us to look for an interesting phenomenon known as beam hardening. I calculate μ using:

$$\mu(E) = \left(\frac{\mu}{\rho} \right) \cdot \rho \quad (2)$$

Where $\left(\frac{\mu}{\rho} \right)$ is the mass attenuation coefficient from NIST [3]

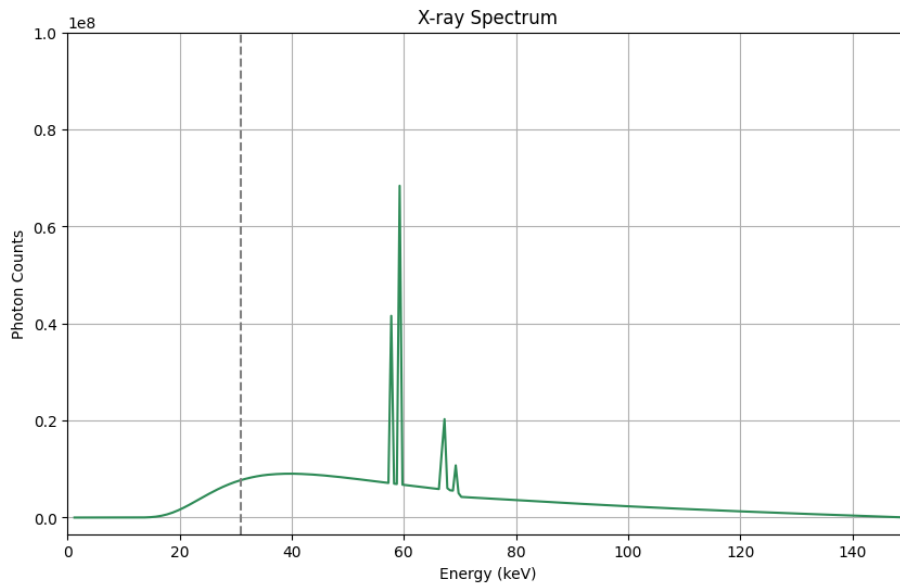


Figure 2: X-ray Spectrum from a x-ray tube with a Tungsten target

2.2 Investigating Beam Hardening

Beam hardening occurs during x-ray attenuation as higher energy (hard) X-rays penetrate dense materials more effectively than the low energy (soft) X-rays. This has the effect of shifting the peak of the bremsstrahlung radiation towards higher energies. Beam hardening cannot occur for monochromatic X-rays, as the average energy will not change, so we can predict that the characteristic radiation will not harden. I used *spekpy*, a dedicated x-ray spectrum simulation toolkit, and *matplotlib* to generate a typical spectra from a Tungsten x-ray tube and plot for energies upto 150 keV. I used a 2mm Aluminium filter to remove low energy, non-diagnostic x-rays. We can use *spekpy*'s built in filter function for this but will also write our own filtering code shortly.

In Figure 2 we can see the sharp characteristic peaks that correspond to Tungsten. Writing code for filtering using NIST data and interpolation to fit the number of data points required.,

```
import pandas as pd
from scipy.interpolate import interp1d

# Load csv
bone_data = pd.read_csv("bone.csv", names=["Energy_MeV", "Mu_Rho"],
                        header=None)#

# Convert energy to KeV
bone_data["Energy_KeV"] = bone_data["Energy_MeV"] * 1000

# Average out duplicates by grouping
```

```

bone_data = bone_data.groupby("Energy_KeV", as_index=False).mean()

# Convert to linear attenuation ( $\mu = \mu/\rho * \rho$ )
bone_density = 1.85 # g/cm3
bone_data["Mu"] = bone_data["Mu_Rho"] * bone_density # [cm-1]
mu = bone_data["Mu"].to_numpy()

# Create interpolation function for attenuation coefficients
mu_interp = interp1d(bone_data["Energy_KeV"], bone_data["Mu"],
                     bounds_error=False, fill_value="extrapolate")

mu_at_spectrum_energies = mu_interp(energies)

thickness_cm = 0.5
I0 = s.get_spk()

I_filtered = I0 * np.exp(-mu_at_spectrum_energies * thickness_cm)

```

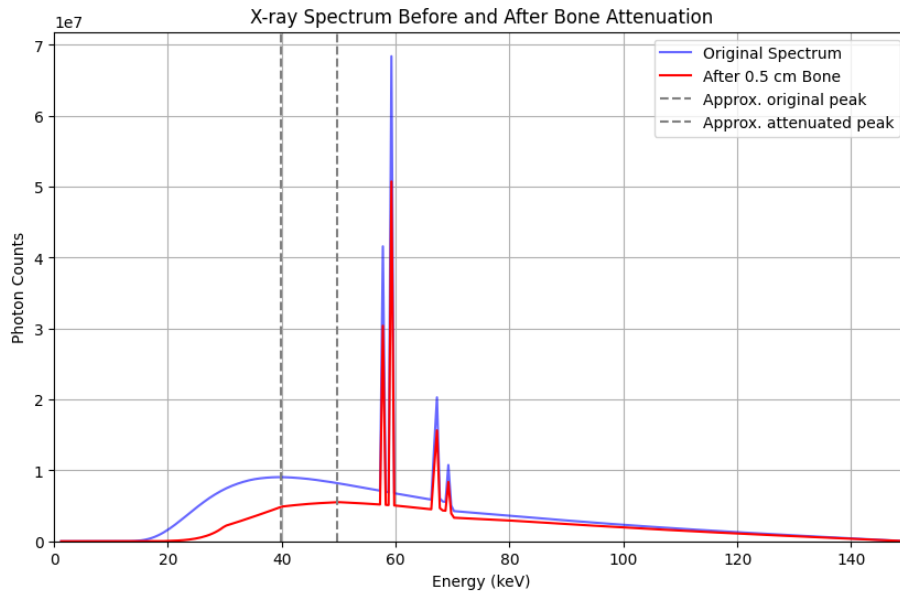


Figure 3: Comparing X-ray spectra before and after passing through 0.5cm of bone

Figure 3 shows beam hardening on the X-ray spectrum. The lower-energy (soft) x-rays are preferentially attenuated, resulting in a spectrum that is skewed toward higher energies. Lower-energy photons are absorbed more readily than high-energy ones. As a result, the transmitted spectrum becomes "hardened," with its average energy increasing. We also don't see any hardening for the monochromatic peaks, as expected.

Beam hardening is significant in image reconstruction, as it may result in


```

def circlemask(radii, centre_x, centre_y, width, height):
    x, y = np.ogrid[:width, :height]
    mask = (x-centre_x)**2 + (y-centre_y)**2 <= radii**2
    return mask

width, height = 180, 180
s = sp.Spek(kvp=150, th=12)
energies = s.get_k()
# Load each material from a csv file
mu_bone = load_mu_data(r"code\bone.csv", density=1.85,
                      energy_range_KeV=energies)
mu_tissue = load_mu_data(r"code\soft_tissue.csv", density=1.06,
                        energy_range_KeV=energies)
E=len(energies)

phantom = np.zeros((height, width, E))
bone_mask = circlemask(10, width//2, height//2, width, height)
thigh_mask = circlemask(40, width//2, height//2, width, height)

# Background
phantom[:, :, :] = 0 # broadcast over all pixels

# Thigh region
phantom[thigh_mask, :] = mu_tissue

# Bone region
phantom[bone_mask, :] = mu_bone

```

Using matplotlib to display these phantoms:

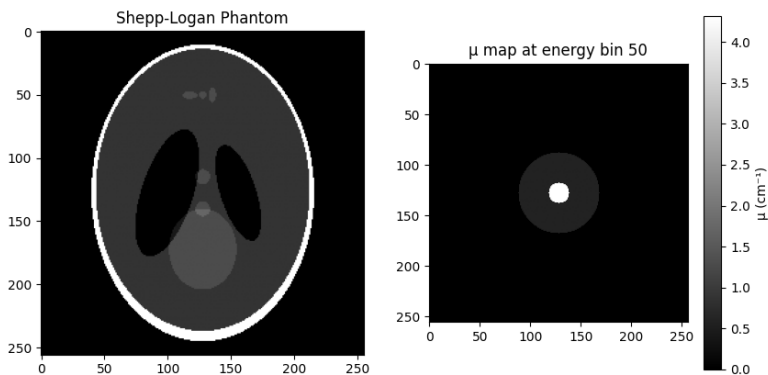


Figure 5: Left: Shepp-Logan phantom. Right: Circular phantom with central bone inclusion (μ map at energy bin 50).

2.3.2 Sinograms

A sinogram is a 2D representation of the projection data acquired in a CT scan, where each row corresponds to a 1D projection taken at a specific angle. Before we can reconstruct the cross-sectional image of the scanned object, the initial x-ray attenuation measurements are compiled into this sinogram format.

To understand image reconstruction, we need to define projections. A projection in CT scans represents the total attenuation of x-rays along a specific path of the object, mathematically this is represented as a line integral. At an angle θ , the parallel beam geometry represents a new coordinate system (r, s) in which the intensity profile $I_\theta(r)$ is measured. Other beam geometries such as fan or cone geometries require different methodology.

$$\begin{pmatrix} r \\ s \end{pmatrix} = R \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Conversely,

$$\begin{pmatrix} x \\ y \end{pmatrix} = R^T \begin{pmatrix} r \\ s \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} r \\ s \end{pmatrix}$$

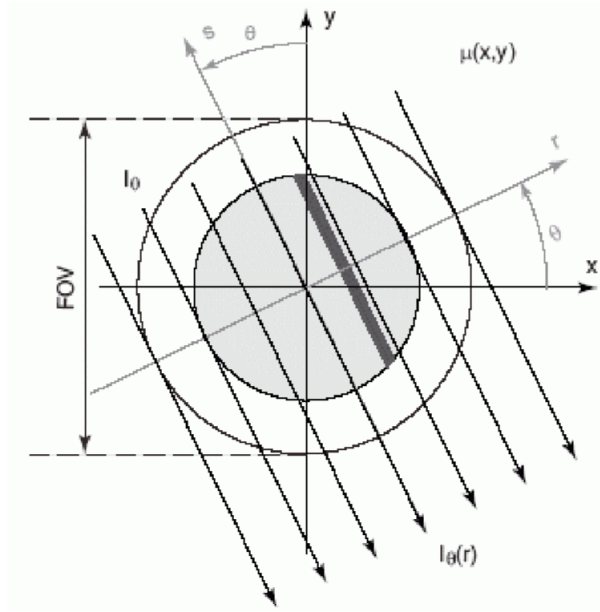


Figure 6: The object is defined in the Cartesian coordinate system (x, y) . For a projection at angle θ , a new coordinate system (r, s) is defined, where r is the perpendicular distance from the origin to the x-ray ray, and s is the coordinate along the ray path

With parallel beams at a fixed angle θ , the measured intensity at position r along $L_{(r,\theta)}$ is given by the Beer-Lambert law:

$$I_{\theta}(r) = I_0 \cdot \exp\left(- \int_{L_{(r,\theta)}} \mu(x,y) ds\right) \quad (3)$$

$$= I_0 \cdot \exp\left(- \int_{L_{(r,\theta)}} \mu(r \cos(\theta) - s \sin(\theta), r \sin(\theta) + s \cos(\theta)) ds\right) \quad (4)$$

Each Intensity profile $I_{\theta}(r)$ is transformed into an attenuation profile $p_{\theta}(r)$, by taking the negative natural logarithm of the ratio of transmitted intensity to incident intensity:

$$p_{\theta}(r) = -\ln \frac{I_{\theta}(r)}{I_0} = - \int_{L_{(r,\theta)}} \mu(r \cos(\theta) - s \sin(\theta), r \sin(\theta) + s \cos(\theta)) ds \quad (5)$$

This attenuation profile is what our detector measures at each angle. Going back to the circle phantom example, we can write some code to simulate this:

```
# Sum attenuation coefficients along the ray path (top to bottom)
# for each (width, energy).
# 'mu_total' is 2D (width, energy), representing total attenuation
# per ray per energy.
mu_total = phantom.sum(axis=0)

I = I0 * np.exp(-mu_total * pixel_thickness_cm)

# Sum transmitted intensities across all energies for each detector
# position.
# 'incident_intensity' is 1D (width,), representing the final
# polychromatic projection profile.
incident_intensity = I.sum(axis=1)

# Visualize detector readout (log form)
detector_readout = -np.log(incident_intensity + 1e-6) # add epsilon
# to avoid log(0)
```

Plotting our result we get:

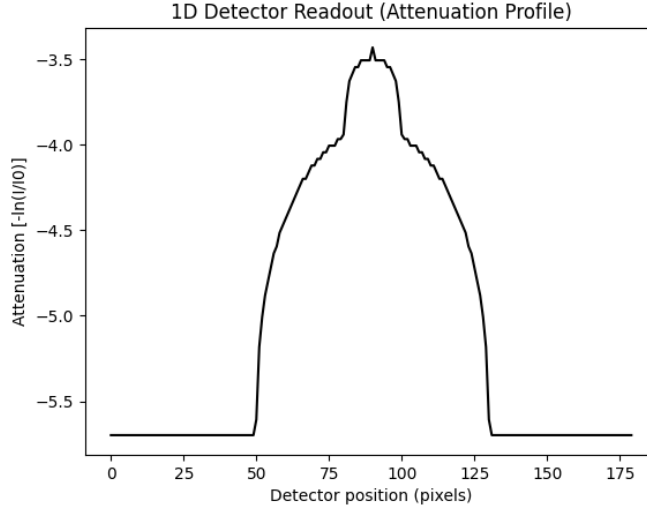


Figure 7: 1D detector readout for our circular phantom, we see the increase in attenuation in the middle as we go through the bone.

If we arrange projections from all angles (only $(0, \pi)$ as $(\pi, 2\pi)$ are redundant for parallel beams) side by side as a 2D image, we get a sinogram. We can view this 2D dataset $p(r, \theta)$ for our circular phantom in Figure 8, this sinogram was generated using `scipys ndimage.rotate` to simulate the projections. Another, more interesting sinogram can be found from our more complex Shepp-Logan phantom, which we can see in Figure 9. We can notice sinusoidal curves throughout the image, motivating the name sinogram.

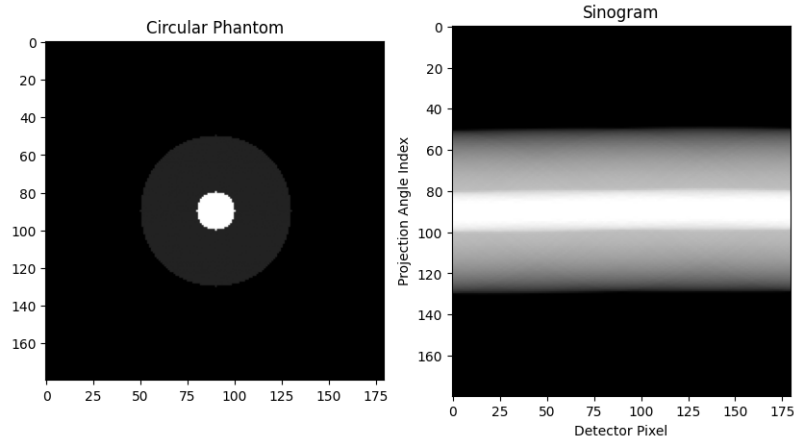


Figure 8: Sinogram for our circular phantom, note how it is constant in θ due to the rotational symmetry of the circle.

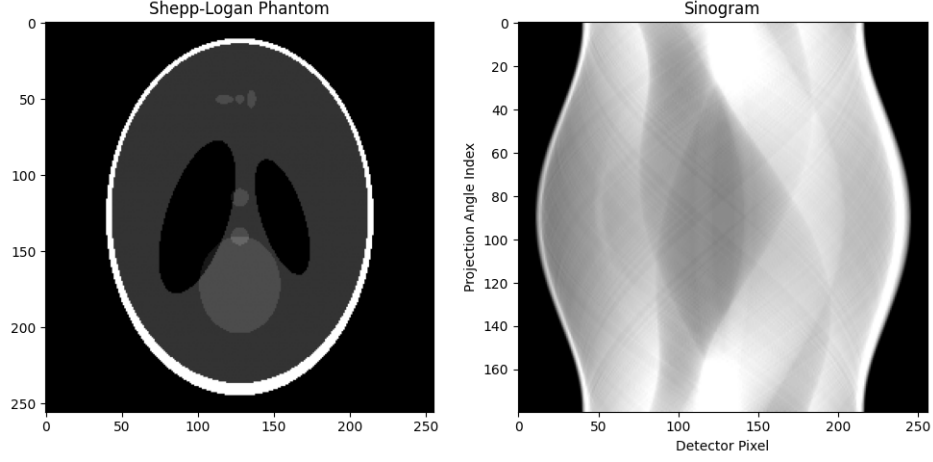


Figure 9: Sinogram for the Shepp-Logan phantom.

The transformation of a function $\mu(x, y)$ into $p(r, \theta)$, is known as the Radon transform defined here as:

Definition 2.1. For a function $\mu(x, y)$ defined on \mathbb{R}^2 , the Radon transform of μ , denoted as $R\{\mu\}$, is defined as

$$\begin{aligned} p(r, \theta) &= R\{\mu(x, y)\} \\ &= \int_{-\infty}^{\infty} \mu(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds \end{aligned}$$

We can therefore phrase the problem of CT reconstruction as finding an inversion formula for the Radon transform [7].

2.4 Inverse Radon Transform and Back Projection

2.4.1 Simple Back Projection

One intuitive way we can think of image reconstruction is with 'Simple Back Projection'. If we take each projection $p_\theta(r)$ of our sinogram, and smear the value along lines perpendicular to the angle θ and add these up for all θ before normalising we should get back to our result. Imagine each projection as a shadow cast from a specific direction; by stacking up these shadows, we can approximate the object that cast them. Mathematically, we define the back projection as:

Definition 2.2. Let $p = p(r, \theta)$. We define the back projection, $\mathcal{B}p$, at a point (x, y) as

$$\mathcal{B}p(x, y) = \frac{1}{\pi} \int_0^\pi p(x \cos(\theta) + y \sin(\theta), \theta) d\theta$$

In practice, we implement this by:

- Taking each row of the sinogram (a projection)
- Replicating it across a 2D grid (smearing)
- Rotating it back in place
- Summing the results across all angles

Or in python code:

```
# Initialize reconstruction
reconstruction = np.zeros((width, height))

# Backproject each angle's projection
for i, angle in enumerate(angles):
    # Expand the 1D projection into a 2D image by smearing along
    # the angle
    projection = sinogram[i]
    expanded = np.tile(projection, (width, 1)) # Repeat along y-
    # axis

    # Rotate back to original angle and accumulate
    rotated_back = ndimage.rotate(expanded, -angle, reshape=False,
    order=1)

    reconstruction += rotated_back

# Normalize
reconstruction /= len(angles)
```

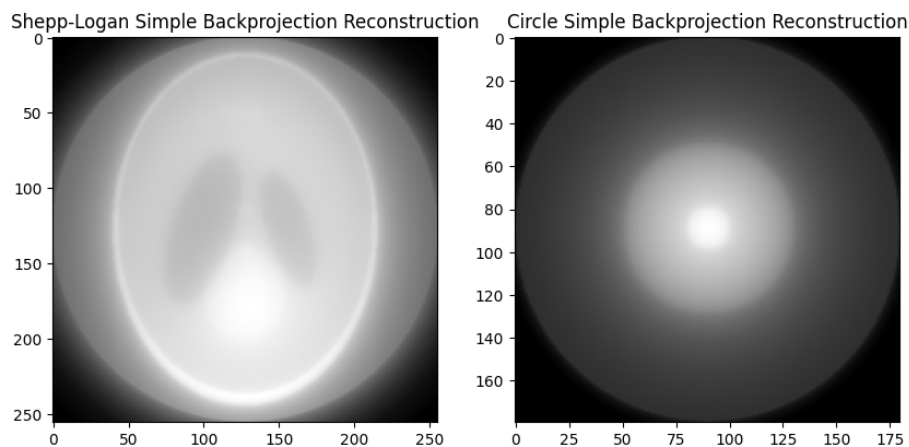


Figure 10: Simple Back Projection Reconstructions for our 2 phantoms.

This roughly reconstructs the structure of the phantoms and we can make out some of the internal structure in both of our phantoms. Both of our phantoms are blurred and have pronounced dark corners. The dark corners are a

known artefact, as they receive fewer ray contributions when we rotate around but to understand the blurring we need to requires further mathematical understanding.

Doing more quantitative analysis on how our Shepp-Logan reconstruction compares to the original we can compare Mean Square Error (MSE) and Structural Similarity Index Measure (SSIM) [8], for our Shepp-Logan:

- MSE: 0.3778
- SSIM: 0.3130

In order to get these figures I ensured that both the original image and reconstruction was normalised, and applied a circle masking function to ensure we were only comparing the significant parts of the image and not the darkened corners.

MSE measures the average squared difference between actual and ideal pixel values. Since the normalized image ranges from 0 to 1, an MSE of 0.3778 is relatively high, indicating a low-fidelity reconstruction.

SSIM evaluates image quality by considering luminance, contrast and structural information. A value close to 1 would imply the 2 images are similar, so 0.3130 suggests significant degradation in these areas (which matches what we observe).

Overall quantitative image comparison is a tricky topic and both of these metrics have issues [9], but they still clearly indicate the need for a more sophisticated reconstruction algorithm.

References

- [1] D. Tafti and C. V. Maani, “Physics, x-ray production,” 2023. Accessed: 2025-06-06.
- [2] J. Mason, *Quantitative cone-beam computed tomography reconstruction for radiotherapy planning*. PhD thesis, 11 2018.
- [3] “X-Ray Mass Attenuation Coefficients — nist.gov.” <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients>. [Accessed 09-06-2025] at <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients>.
- [4] A. Murphy, “Beam hardening — radiology reference article — radiopaedia.org,” Oct 2016.
- [5] L. A. Shepp and B. F. Logan, “The fourier reconstruction of a head section,” *IEEE Transactions on Nuclear Science*, vol. 21, p. 2143, Jun 1974.
- [6] J. Hrachovec, “phantom.py,” n.d. Accessed: 2025-06-09 at <https://jenda.hrach.eu/f2/cat-py/phantom.py>.

- [7] J. Beatty, “The radon transform and the mathematics of medical imaging,” honours thesis, Colby College, 2012.
- [8] U. Sara, M. Akter, and M. Uddin, “Image quality assessment through fsim, ssim, mse and psnra comparative study,” *Journal of Computer and Communications*, vol. 07, no. 03, p. 818, 2019.
- [9] J. Nilsson and T. Akenine-Mller, “Understanding ssim,” 2020.