Inverse Problems in Imaging: Radon Transform and its Inverse

1 Part A: Core Project

1.1 Calculate the Radon transform of an image and test the back-projection method.

We first load the 128x128 Shepp-Logan phantom image(f_{true}) and show it in Fig .1.
We retrieve the original image's dimensions f_true, which are the height and width (v and h). Then, we use these dimensions to create a volume geometry object vol_geom. This object defines the spatial dimensions of the image, which ASTRA uses to set up the projection and reconstruction geometries for subsequent operations.
We use `angles = np.linspace(0, np.pi, 180, endpoint=False)` to set the projection angles ensuring that it was taken from 0° to 179° at 1° intervals during the Radon transform process. We than generate the Radon transform `sinogram`(g) in Fig .2.
The size of the sinogram generated by the Radon transform is determined by two main factors:

1. **Number of Angles (Angular Resolution)**:

    - The number of rows in the sinogram corresponds to the number of angles at which projections are taken. In this case, we perform the Radon transform at 1-degree intervals from 0 to 179 degrees, then have 180 angles. Hence, there will be 180 rows in the sinogram.

2. **Number of Projection Samples (Detector Resolution)**:

    - The number of columns in the sinogram corresponds to the number of detector elements that capture the projection data at each angle. This is often related to the size of the image being transformed. If the original image is 128×128 pixels, a common choice for the number of detector elements is to use the diagonal length of the image since this ensures that all parts of the image are captured in the projection no matter the angle. For a 128×128 image, the diagonal can be calculated using the Pythagorean theorem, approximately $\sqrt{128^2 + 128^2} \approx 181$. Hence, there will be 181 columns in our sinogram.

`proj_geom` uses those angles and the detector count previously calculated to define a parallel beam projection geometry.
The size of the back-projected image in our Radon transform computation using the ASTRA toolbox will match the original image dimensions we used for the Radon transform. This is because back-projection reconstructs the image to its original dimensions from

the sinogram data. Hence, it is 128 × 128.
We create a data object to store the reconstruction with `astra.data2d.create(’-vol’, vol_geom)`. And configure the unfiltered back-projection using `astra.astra_dict(’BP’)`. To compute the filtered back-projection, we modify the existing configuration `cfg` by changing its type to ’FBP’, indicating that the reconstruction algorithm should use filtered back-projection. We then show the reconstructed image in Fig .3b. Aside from visualization, we verify the reconstruction quality by calculating the mean squared error between the original image and the reconstructed image. This metric quantitatively evaluates the accuracy of the reconstruction, with lower values indicating a closer match to the original. We get the MSE of 0.003, a quite low value, showing that the FBP is a good method.
To add noise to the data g, we use synthetic noise to the sinogram data at varying levels. The function add_noise_to_sino uses Poisson noise. The noise levels ranged from 1 to 100 in increments of 10. The Background intensity parameter is important in simulating noise; lower values result in higher noise levels. Fig .4 shows a generally increasing trend in MSE as the noise level increases from 0 to 70. We would generally expect this trend because higher noise levels usually degrade the quality of the image reconstruction, leading to greater discrepancies between the reconstructed image and the original.

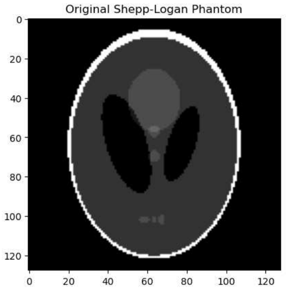


Figure 1: Original Shepp-Logan Phantom

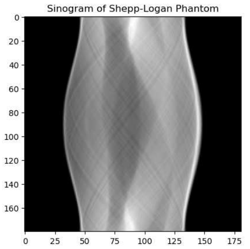


Figure 2: Radon transform of the Shepp-Logan phantom

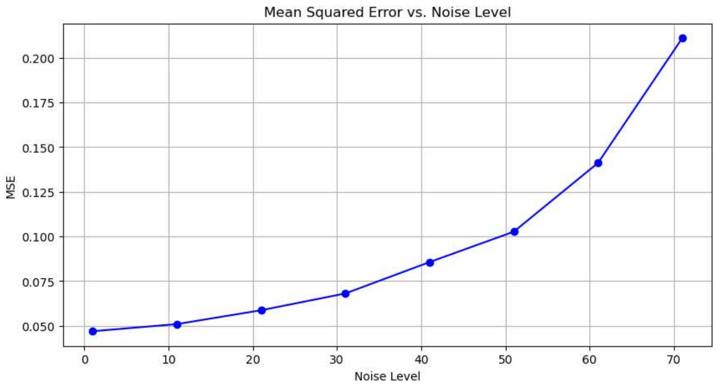
(a) Unfiltered back-projection

(b) Filtered back-projection

Figure 3: Comparison of unfiltered and filtered back-projection results

1.2 Calculate an explicit matrix form of the Radon transform and investigate its SVD

The explicit matrix form of the Radon transform, denoted by $\mathbf{A}$, is constructed by mapping the Radon transform of each pixel in an image onto the columns of $\mathbf{A}$. The matrix has dimensions (num_angles × det_count) × (img_size^2), where det_count is set to



3.1

Figure 4: Plot of MSE versus noise level

$\lceil \sqrt{2} \times \text{img\_size} \rceil$. Each column of $\mathbf{A}$ corresponds to the Radon transform of a single-pixel image, where only one pixel is set to one, and all others are set to zero. This approach simulates how each pixel contributes to the overall Radon transform.

To populate $\mathbf{A}$, we loop through each pixel index j within the image, constructing a basis image for each index. This basis image is then projected using ASTRA to compute its sinogram, which is reshaped and stored in $\mathbf{A}$.

To explain how to calculate the explicit matrix form of the Radon transform and investigate its singular value decomposition (SVD), we need to do the following steps:

1. **Create the Zero Matrix** A: We create a zero matrix A with the number of rows equal to the number of angles × the number of projection samples, and the number of columns equal to the number of pixels in the image. Suppose we have M angles, and each angle has P projection samples, then the size of matrix A is $(M \times P) \times (N \times N)$.

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{N^2} \end{bmatrix} \tag{1}$$

   where A_j is the column vector corresponding to the j-th pixel.

2. **Filling the Matrix** A: For each pixel j in the image (traversing in column-major order), create an image of the same size with a 1 at that pixel and 0 elsewhere. For each pixel j:

   (a) Create the image f_j:

$$f_j = \text{reshape}(\mathbf{e}_j, (N, N)) \tag{2}$$

   where $\mathbf{e}_j$ is the j-th unit vector.

   (b) Compute the Radon transform Rf_j:

$$g_j = Rf_j \tag{3}$$

   (c) Reshape the result into a column vector and assign it to the j-th column of matrix A:

$$A(:, j) = \text{reshape}(g_j, (M \times P, 1)) \tag{4}$$

3. **Investigating the SVD**: Computing the SVD of the Radon Transform Matrix We need to perform the singular value decomposition (SVD) on the constructed matrix A. The SVD decomposes the matrix into the product of three matrices:

$$[U, \Sigma, V] = \text{SVD}(A) \tag{5}$$

where U and V are orthogonal matrices, and Σ is a diagonal matrix with singular values on the diagonal.

1.2.1 Different Numbers of Projections

Fixed Angle Range from 0 → 180°, Changing the Number of Projections

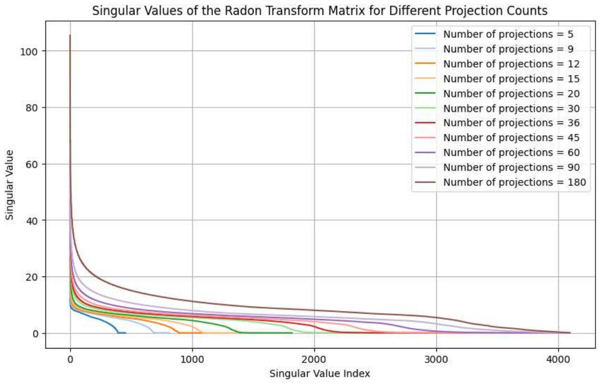


Figure 5: Plot of the singular values of the Radon Transform matrix for varying counts of projections.

1. **Singular Value Index Range**: As the number of projections increases, the singular value index range expands. This is because more projections provide more information, leading to a more complete representation of the image in the Radon transform domain, resulting in more non-zero singular values.
2. **Maximum Singular Value**: The maximum singular value increases with the number of projections because more projections provide more information, leading to a more complete representation of the image in the Radon transform domain and a stronger signal.
3. **Singular Value Decrease**: The singular values decrease along their index because the first few singular vectors contribute the most to the image, while the later singular vectors contribute less. As the number of projections increases, the contributions of the first few singular vectors become more significant.

The experiment result is showed in Fig .5. The graph illustrates how the magnitude of the singular values decreases as the number of singular value indices increases. Each line represents a different number of projection counts, ranging from 5 to 180, highlighting the impact of the number of projections on the decay rate of the singular values.

1.2.2 Different Numbers of Angle Ranges

Fixed Number of Projections, Changing the Angle Range Less Than 0 → 180°

1. **Consistency of Singular Values**: The index range and maximum value of singular values remain constant as the angle range increases due to a fixed number of projection samples.

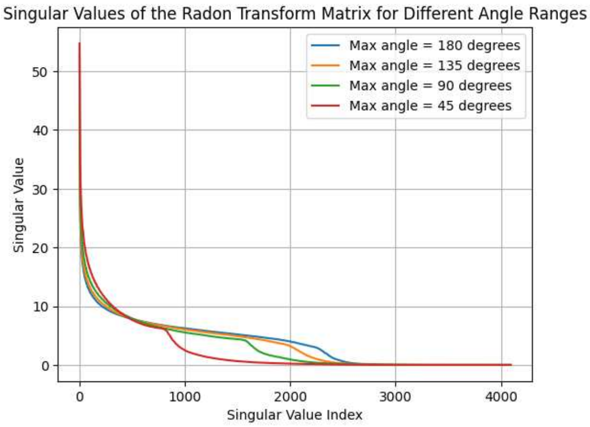


Figure 6: Plot of the singular values of the Radon Transform matrix for different maximum angle ranges.

2. **Non-proportional Singular Value Distribution**: The distribution of singular values does not proportionally change with an increase in angle range because limited-angle projection data contains less information compared to full-angle projection data.

3. **Impact of Limited-angle Projection Data**: Limited-angle projection data may lead to an incomplete and uneven representation of the image in the Radon transform domain, affecting the shape and distribution of singular values.

The experiment result is showed in Fig .6. The graph displays the decay of singular values for different maximum angles of projection, including 45 degrees, 90 degrees, 135 degrees, and 180 degrees. Each line represents the magnitude of the singular values as a function of their index, showing how the completeness of angular coverage affects the values' decline.

1.3 Implement a matrix-free regularised least-squares solver for the Radon Transform.

Implementing a matrix-free regularized least-squares solver for the Radon Transform is useful when dealing with large datasets where explicitly forming matrices such as A^TA is computationally impractical.
The task involves solving the regularized equation:

$$(A^TA + \alpha L)f^* = A^Tg$$

where:

- A represents the Radon transform operator.
- A^T is the adjoint (or transpose) of the Radon transform.
- g is the sinogram data obtained from the Radon transform of the image.
- L is a regularization matrix typically used to impose smoothness or other constraints.

- α is the regularization parameter that balances fidelity to the data against the smoothness enforced by the regularization term.
- f^* is the solution vector, representing the reconstructed image.

To implement our algorithm, we first define the following operators:
In a matrix-free approach, we do not explicitly form the matrices A and A^T. Instead, we define functions that can compute the results of Ax and A^Ty for any vector x or y.

1. **Forward Radon Operator(Ax):** The ASTRA toolbox is used to perform the Radon transform, converting 2D images into sinograms.
2. **Adjoint Radon Operator(A^Ty):** Utilizing the ASTRA toolbox, this function executes the backprojection, reconstructing an image from sinogram data.
3. **Regularized Least Squares Solver:** This solver integrates the previous operators in a matrix-free approach, applying regularization techniques to enhance image reconstruction quality.

1.3.1 Integrated Overview of `regularized_least_squares_solver`

Inputs:

- **Image**: The original image to be reconstructed.
- **Alpha**: The regularization parameter that balances the accuracy of the data against the smoothness or other constraints imposed by matrix L.
- **Angles**: Specifies the angles at which the Radon transform is computed, typically covering a full or partial circle.
- **L**: The regularization matrix applied to enforce properties in the reconstructed image, such as smoothness or edge preservation.

Matrix-Free Operations: The function uses matrix-free techniques to perform transformations, reducing memory usage and computational overhead. It simulates the effect of matrix operations using `LinearOperator`:

- **LinearOperator**: Defines a custom operator for $A^TA+\alpha L$, enabling matrix-vector multiplication without explicitly forming the matrix.

Core Computational Steps:

- **Radon Transform**: Projects the image into sinogram space using specified angles via `forward_radon_operator`.
- **Noise Addition**: Adds Gaussian noise to the sinogram, simulating real-world imaging artifacts.
- **Backprojection**: Reconstructs an initial approximation of the original image from the noisy sinogram using `adjoint_radon_operator`.
- **Regularization**: Applies the regularization matrix L scaled by α directly in the matrix-vector multiplication within the `LinearOperator`.

Solving the Equation:

- **Conjugate Gradient Solver (`cg`)**: Solves the linear system defined by the `LinearOperator`. This method is efficient for large, sparse, or symmetric systems and is particularly effective for iterative refinement of image reconstruction.

Output:

- Returns the reconstructed image along with the exit code from the solver, indicating the success or failure of the iterative process.

1.3.2 Find optimal alpha

The function `find_optimal_alpha` in Python utilizes the discrepancy principle to determine the optimal regularization parameter, α, for image reconstruction problems. It defines an objective function that computes the residual norm between the Radon transform of the original and reconstructed images, reduced by the square of the noise level σ^2. The objective is to find the α that minimizes the discrepancy between this residual and the expected noise level. This is realized using the `minimize` function from `scipy.optimize`, which iteratively adjusts α within specified bounds to minimize the objective function.

1.3.3 Comparison of Reconstruction methods

We compare reconstruction from sinograms under two setups. Each setup tests the efficacy of traditional filtered backprojection and Tikhonov regularization (zero-order and first-order) under different conditions.

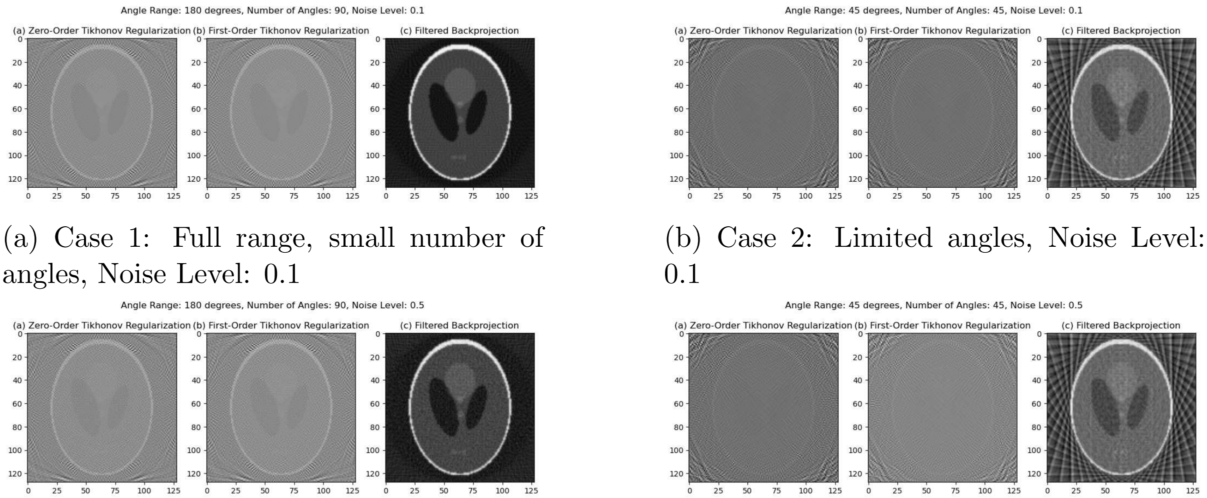
1. **Case 1: Full Range but Small Number of Angles** - This setup tests the reconstruction capabilities when the angle range is complete (180 degrees) but with a reduced number of projection angles. This scenario simulates the condition where time constraints or dose considerations limit the number of projections.

2. **Case 2: Limited Angles** - This case involves reconstructing images from projections taken over a limited angle range, typical in situations where the object blocks the path of the x-ray beam for certain angles or equipment constraints limit the angular range.

Method For both cases, we generate sinograms from a standard phantom image, introduce Gaussian noise to simulate realistic measurement conditions, and then apply the three reconstruction methods. The `perform_backprojection` function implements filtered backprojection using the ASTRA Toolbox, providing a baseline for comparison.

To optimize the regularization parameter α, we use the `find_optimal_alpha` function. This function iterates over a range of α values, solving the regularized least squares problem via a conjugate gradient solver, and selects the α that minimizes the reconstruction error.

Results and Visualization The reconstructed images for both cases under two different noise levels are presented in Figure 7. The results showcase the outcomes from filtered backprojection, zero-order Tikhonov regularization, and first-order Tikhonov regularization for full angular range with a small number of angles (Case 1) and limited angles (Case 2). Each subplot illustrates the impact of noise level on the reconstruction quality, emphasizing the differences and effectiveness of each method under varying conditions.


Angle Range: 180 degrees, Number of Angles: 90, Noise Level: 0.1
Angle Range: 45 degrees, Number of Angles: 45, Noise Level: 0.1
Angle Range: 180 degrees, Number of Angles: 90, Noise Level: 0.5
Angle Range: 45 degrees, Number of Angles: 45, Noise Level: 0.5
(a) Zero-Order Tikhonov Regularization
(b) First-Order Tikhonov Regularization
(c) Filtered Backprojection
0
20
40
60
80
100
120
0
25
50
75
100
125

(a) Case 1: Full range, small number of angles, Noise Level: 0.1

(b) Case 2: Limited angles, Noise Level: 0.1

(c) Case 1: Full range, small number of angles, Noise Level: 0.5

(d) Case 2: Limited angles, Noise Level: 0.5

Figure 7: Comparison of reconstruction results for different cases and noise levels.

Conclusions As expected, the noise impacts the image quality in all methods. With increasing noise levels, the resultant images from both zero-order and first-order Tikhonov regularization show more artefacts and blurring. The noise appears to introduce more significant distortions in lower noise levels, with the reconstructions at higher noise levels exhibiting profound blurring, which obscures details.

Zero-Order Tikhonov Regularization: This method tends to produce smoother reconstructions but at the cost of blurring sharp features and edges. In the presence of higher noise (e.g., 0.5), this method fails to recover finer details, leading to overly smooth images where even prominent structures are not clearly discernible.

First-Order Tikhonov Regularization: Comparatively, first-order regularization maintains more edge details than zero-order. This is visible in the slightly sharper boundaries and less smoothing effect in the images. However, this method also introduces some directional artifacts, particularly visible at higher noise levels, which can distort the true geometry of the objects.

Without regularization, filtered backprojection clearly shows the structure of the Shepp-Logan phantom but with significant artifacts due to noise and incomplete data coverage. The circular ringing artifacts, characteristic of this method, are pronounced, especially at lower noise levels (0.1), and they obscure fine details.

The performance of each method varies significantly with the angle range. Narrower angle ranges (45 degrees) lead to more limited information, which exacerbates the noise effects and regularization artifacts. For instance, at 45 degrees, both the zero-order and first-order regularizations struggle to reconstruct accurate geometries and mostly yield diffused and distorted images.

1.4 Write a Haar wavelet denoiser.

The Haar wavelet transformation was applied to a monochrome image using the PyWavelets library in Python. The discrete wavelet transform was computed up to seven levels, decomposing the image into approximation and detailed coefficients.

We analyse these coefficients visually to understand an image's structural and frequency characteristics. The approximation coefficients at level 7, which represent low-frequency components, capture the general outline and broader shapes within the image, as shown in Fig .8. This level is mainly concerned with the overall structure rather than fine details.

Furthermore, the detail coefficients expose the finer structures and textures of the image at different horizontal, vertical, and diagonal directions. These coefficients, illustrated in the second figure, highlight edges and transitions at multiple scales, showcasing the image's finer details as one moves to lower levels of decomposition.

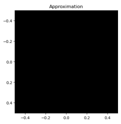


Figure 8: Approximation coefficients at the highest level of the Haar wavelet decomposition, capturing the low-frequency components of the image.

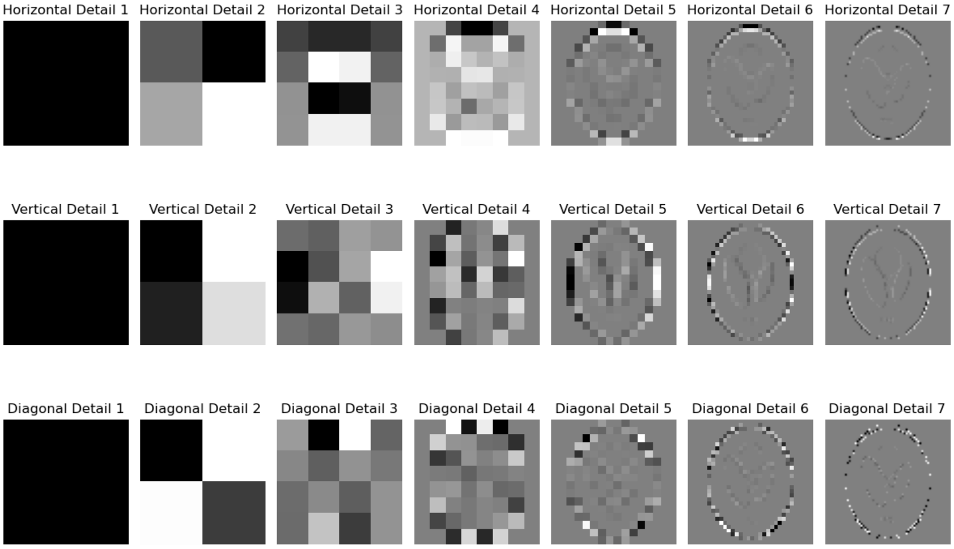


Figure 9: Detail coefficients for horizontal, vertical, and diagonal directions from level 1 to level 7, illustrating the multi-scale edge and texture information captured by the Haar wavelet.

1.4.1 Reconstruction from Wavelet Coefficients

We reconstructed the image from its Haar wavelet coefficients using an inverse wavelet transform to verify the integrity of the wavelet decomposition and reconstruction process. This step was essential to confirm that the reconstructed image faithfully represents the original image, which is crucial for any subsequent image processing operations like denoising that rely on accurate wavelet transformations.

The results demonstrate that the wavelet-based reconstruction process can restore the original image with high accuracy. The visual comparison in Figure 10 shows no apparent differences between the original and reconstructed images, supporting the lossless nature of the Haar wavelet transform when all coefficients are used for reconstruction. The quantitative analysis confirms this, as the maximum pixel difference between the

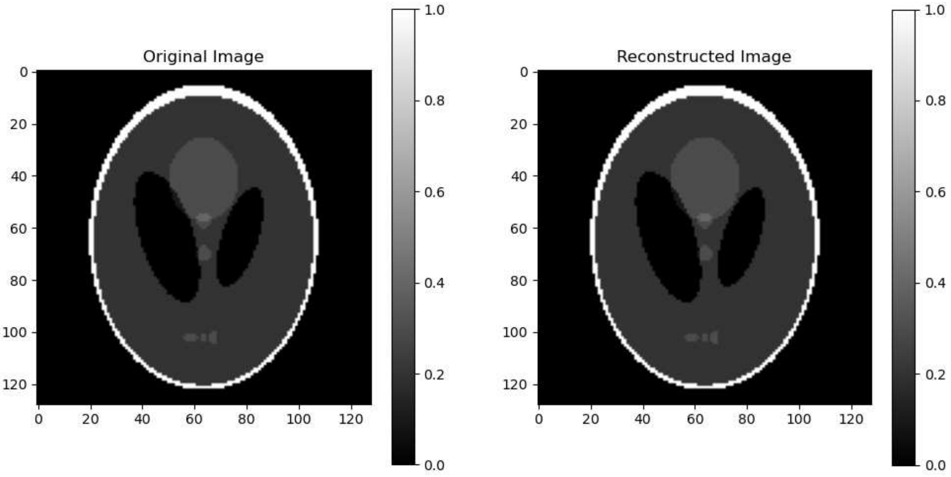


Figure 10: Original vs Reconstructed Image: The left side shows the original image and the right side shows the reconstructed image after performing an inverse wavelet transformation. The grayscale intensity values are preserved with high fidelity, indicating a successful reconstruction.

original and reconstructed images is negligible, effectively zero within the precision limits of the computing environment. This confirms the suitability of the Haar wavelet transform for tasks that require reversible transformations, such as noise reduction and data compression. Additionally, to quantitatively assess the exactness of the wavelet-based reconstruction, we computed the difference between the original and the reconstructed images. The difference image, shown in Figure 11, highlights the minor discrepancies between the two images.

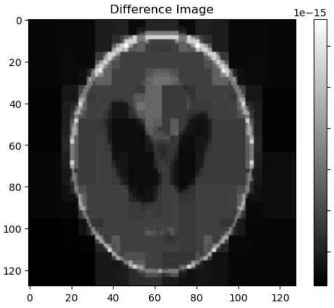


Figure 11: Difference Image: This image visualizes the absolute differences between the original and reconstructed images on a pixel-by-pixel basis. The maximum difference recorded is 1.33×10^{-15}, which is zero, indicating that the reconstruction has achieved near-perfect accuracy.

The maximum difference observed, 1.33×10^{-15}, is infinitesimally small, practically denoting no difference and confirming the lossless nature of the Haar wavelet transform in this context. This result supports the earlier visual assessment and establishes the Haar wavelet method as exceptionally reliable for image-processing tasks requiring exact reversibility.

1.4.2 Thresholding Wavelet Coefficients

To denoise the image, we implemented a function named `threshold_wavelet_coeffs` to selectively modify the Haar wavelet coefficients by applying a soft threshold. This function is designed to operate on specified scales, allowing us to fine-tune the level of detail affected by the thresholding process.

Initially, the function copies the original set of coefficients to ensure the original data remains unchanged. It then applies a soft threshold to the coefficients of the specified scales, skipping the approximation coefficient at the lowest scale. This method suppresses noise by reducing coefficients that are deemed insignificant, defined by a threshold value.

For the demonstration, we used the Shepp-Logan phantom image to compute its wavelet transform up to seven levels. We then applied the thresholding function to all scales except the approximation, with a threshold value set to 20. This approach targets and diminishes smaller coefficients representing noise or less important details.

Figure 12 compares the two images, illustrating the effect of the thresholding where noise elements are reduced.

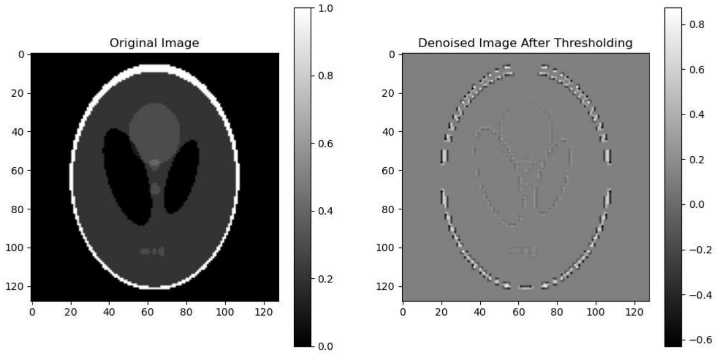


Figure 12: Comparison of Original and Denoised Images: The left image shows the original Shepp-Logan phantom, while the right image displays the outcome after applying the wavelet thresholding. The noise reduction is visually significant, highlighting the effectiveness of the thresholding approach in enhancing image clarity.

It shows the utility of wavelet-based denoising in scenarios where preservation of major structural details is crucial, while effectively diminishing undesired noise. The parameterization of the threshold value and the choice of scales offer flexibility, making this method adaptable to different noise levels and image characteristics.

1.4.3 Noise Addition and Wavelet Denoising (a noisy version)

We explored the strength of wavelet denoising techniques by first adding Gaussian noise to the original Shepp-Logan phantom image and then applying a denoising process through wavelet thresholding. The objective was to evaluate how effectively the wavelet-based method can mitigate noise and restore the original image's clarity.

The Gaussian noise was introduced using a custom function `add_gaussian_noise`, parameterized by a sigma value of 0.05, indicating the noise level. This function generates a noise array following a normal distribution and adds it directly to the original image.

Following the noise addition, we computed the wavelet transform of the noisy image using the Haar wavelet up to seven decomposition levels. The threshold for the wavelet coefficients was determined based on the 95th percentile of the absolute values of all detail coefficients across all levels, ensuring that only the most significant coefficients were retained. This percentile-based thresholding effectively preserves essential image features while reducing noise.

After thresholding the coefficients across all scales, the inverse wavelet transform was applied to reconstruct the image from the modified coefficients. We compared the original and the noisy images to assess the effectiveness of the denoising process.

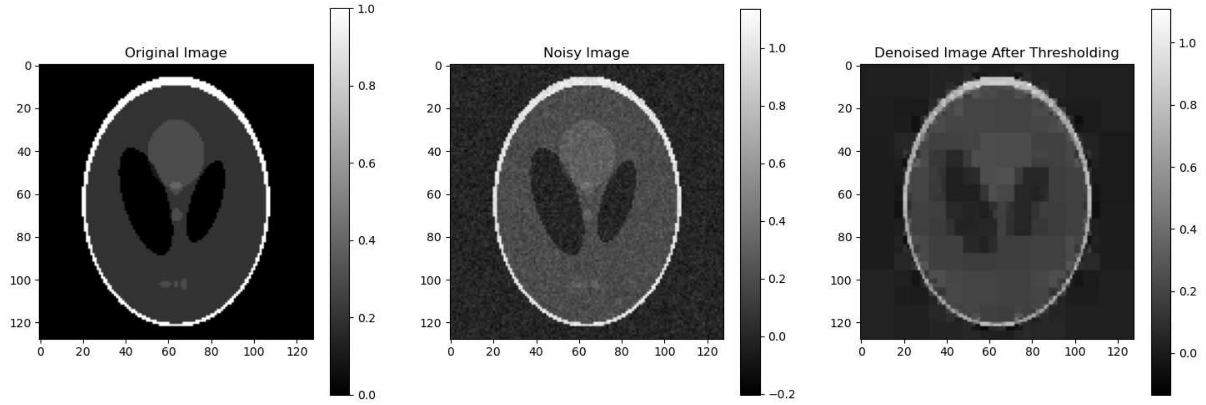


Figure 13: The image on the left shows the original Shepp-Logan phantom, the middle image represents the noisy version with a Gaussian noise level of 0.05, and the right image illustrates the effectiveness of wavelet denoising.

1.4.4 Investigating the Effects of Threshold and Range Parameters

We adjust these parameters and observe their impact on the Mean Squared Error (MSE) between the denoised and original images. Our analysis focused on understanding how different thresholds and numbers of scales influenced the denoising quality.

Function, `threshold_wavelet_coeffs`, was created to apply thresholding to the wavelet coefficients selectively. This function was crucial for experimenting with various thresholds and identifying the most effective range of scales for denoising.

The functions `plot_mse_vs_threshold` and `plot_mse_vs_scales` were utilized to visualize the effects of these parameters on denoising performance. The former plotted MSE against different threshold values for a fixed number of scales, while the latter illustrated how the MSE changed as the number of scales used in the thresholding varied. These visualizations are crucial for identifying optimal denoising settings that minimize MSE, thus enhancing image quality.

We then Visualize the original, noisy, and denoised images side by side.

We also use a function utilized in a systematic analysis where the threshold percentiles ranged from 95% down to 75%, and the range limits varied from 1 to 7 levels of wavelet decomposition. This approach allowed us to determine the optimal settings for denoising under different noise conditions and wavelet scales.

Each subplot provides insights into how adjusting the threshold and range affects the restored image quality, particularly focusing on how well the denoising process can reduce noise while preserving important image details. Range limit 7 shows a better performance.

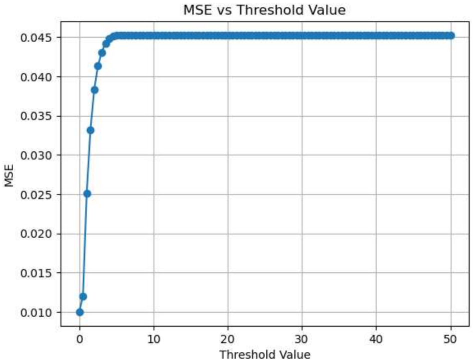


Figure 14: MSE versus threshold values, showing the relationship between the threshold setting and denoising performance.

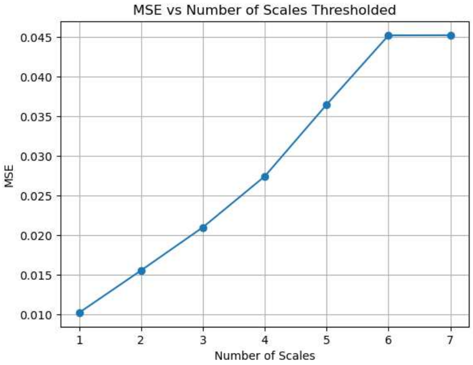


Figure 15: MSE as a function of the number of scales used in thresholding. This graph helps determine the optimal number of scales to use for effective denoising.

1.5 Iterative soft-thresholding for X-ray tomography.

1.5.1 Implementation and Execution of Iterative Soft-Thresholding Algorithm (ISTA)

The Iterative Soft-Thresholding Algorithm (ISTA) is employed for sparse image reconstruction in X-ray tomography, using the ASTRA toolbox for efficient projection and back-projection operations and utilizing Haar wavelet transform for regularization. The algorithm initiates by adapting a Shepp-Logan phantom to predefined dimensions, where necessary adjustments are made either by cropping or padding to ensure consistency in size. The imaging setup involves establishing projection and volume geometries through ASTRA's configuration functions, followed by generating a sinogram from the phantom.

Each iteration of ISTA calculates the projection of the current estimate, subtracts it from the observed data to determine the residual, and back-projects this residual to update the estimate. Soft-thresholding is applied within the wavelet domain using the Haar wavelet to promote sparsity. This process iterates, refining the image estimate by minimizing the error between the projected estimate and the observed data while adjusting for sparsity via wavelet thresholding.

The algorithm is executed with different initializations—backprojection of the noisy sinogram, a random noise matrix, and a zero matrix—to assess the impact of initial values on convergence behavior and effectiveness of soft-thresholding in achieving sparse solutions. The iterative update is controlled by a pre-set tolerance level, terminating the

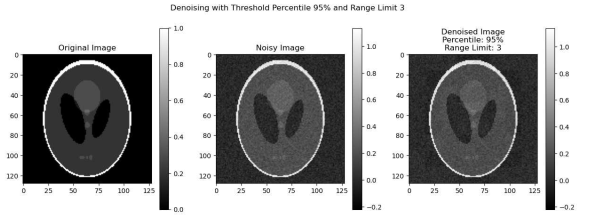


Figure 16: Denoising results for percentile 95% and range limit 3.

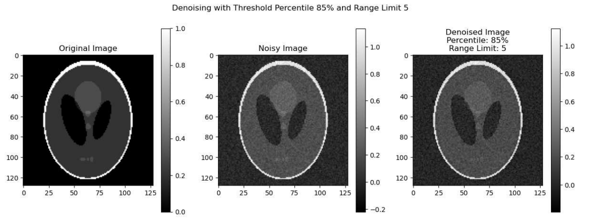


Figure 17: Denoising results for percentile 85% and range limit 5.

iterations upon satisfaction to prevent oscillations or divergence.

Visualizations post-reconstruction compare the effects of various initializations on the reconstruction quality, illustrating differences in image quality and feature preservation. Figure 20 illustrates the results of running the Iterative Soft-Thresholding Algorithm (ISTA) with three distinct initializations: UBP Initialization, Random Noise Initialization, and Zero Initialization. Each method reveals significant differences in how the algorithm converges to a solution, reflecting the influence of initial values on the reconstruction quality.

The UBP Initialization (a) demonstrates the most reliable reconstruction, preserving the core features of the Shepp-Logan phantom with minimal artifacts. This approach utilizes the prior knowledge embedded in the unfiltered backprojection of the noisy sinogram, providing a beneficial starting point that closely aligns with the true distribution of X-ray attenuation.

Conversely, the Random Noise Initialization (b) introduces considerable noise and artifacts, significantly degrading the image quality. This initialization tests the algorithm's ability to recover from a highly perturbed state, emphasizing the challenge of achieving convergence starting from a randomly generated image matrix.

The Zero Initialization (c) method demonstrates the algorithm's capability to progress gradually through the reconstruction from a blank slate. Although it starts with no prior information, this method still manages to reconstruct the basic features of the phantom, showcasing the ISTA's effectiveness in building up image details iteratively. However, convergence is slower than initialization at UBP.

These results underline the importance of choosing an appropriate initialization for iterative reconstruction algorithms in medical imaging, directly impacting the speed of convergence and the fidelity of the reconstructed image.

1.5.2 Optimization of Step Size λ

We explore the optimization of the step size, λ, in the iterative soft-thresholding algorithm for X-ray tomography. The objective was to ensure the algorithm's stable and accelerated convergence under varying noise conditions and angular configurations.

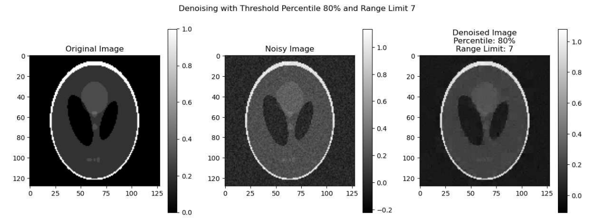


Figure 18: Denoising results for percentile 80% and range limit 7.

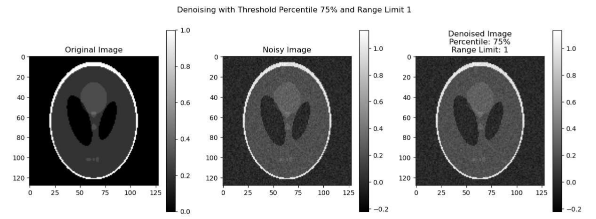


Figure 19: Denoising results for percentile 75% and range limit 1.

We developed a function, `evaluate_ista`, iteratively reconstructs the Shepp-Logan phantom, each time varying the noise level and the number of projection angles, while maintaining the fixed values for λ and the regularization parameter, α.

The test involves:

- Initializing the volume geometry and loading the ground truth phantom image.
- Generating noisy sinograms by adding Gaussian noise to clean sinograms at specified noise levels: low, medium, and high.
- Performing ISTA with the given λ and α over a fixed number of iterations, allowing us to observe the convergence behavior.
- Computing the Mean Squared Error (MSE) and the Structural Similarity Index (SSIM) for each reconstruction to quantitatively assess image quality.
- Plotting these metrics against the number of angles used in the reconstructions to evaluate the impact of projection density and noise on reconstruction accuracy.

This analysis aims to determine how λ influences the reconstruction quality in different settings and to establish a criterion for selecting an optimal step size that balances the speed of convergence with the accuracy of the reconstructed images.

In conclusion, the choice of λ plays a critical role in the iterative reconstruction's success, affecting both the process's quality and computational efficiency. The results suggest that a smaller λ may be beneficial in scenarios with higher noise and fewer angles, promoting gradual and stable convergence.

1.5.3 Incorporation of Non-Negativity Constraint

In the iterative soft-thresholding algorithm for X-ray tomography, the non-negativity constraint is integrated to ensure that the reconstructed images adhere to physical reality, where image intensities, representing physical densities, cannot be negative. This constraint is applied in the update step of the algorithm as follows:

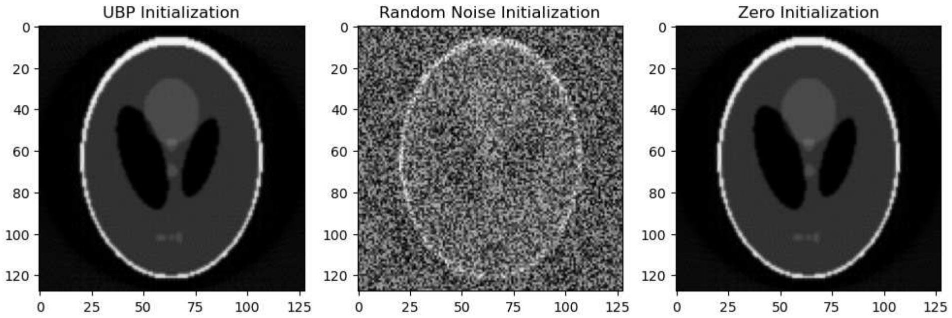


Figure 20: Comparative results of ISTA with different initializations: (a) UBP Initialization shows clear feature preservation, (b) Random Noise Initialization results in significant artifacts and noise presence, and (c) Zero Initialization gradually builds up image features from a blank state.

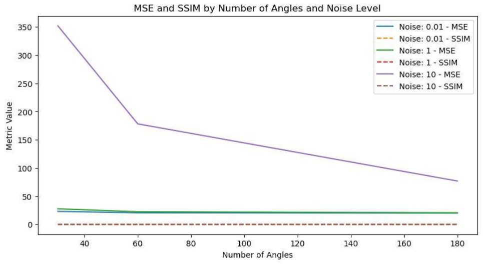


Figure 21: Performance evaluation of the ISTA across varying noise levels and number of angles. The graphs show MSE and SSIM for each configuration, highlighting the impact of λ and α in stabilizing and expediting the convergence of the reconstruction process.

$$f_{k+1} = \max(W^{-1}S_\mu(W(f_k + \lambda A^T(g - Af_k))), 0),$$

where W denotes the wavelet transform, S_μ represents the soft-thresholding operator with threshold μ, A is the projection operator, g is the observed data, and λ is the step size. The $\max(\cdot, 0)$ operation enforces non-negativity by setting all negative values to zero, ensuring that each iteration produces a physically plausible image. This implementation not only adheres to the expected physical properties of the images but also enhances the overall quality and realism of the reconstruction.

2 Part B:

For the advanced topic selection in part B of our project, we have chosen to focus on ”Advanced Topic 1: Inpainting in Sinogram Space.”

2.1 Sinogram Data Generation

In this section, we focus on generating sinogram data for two specific cases: full-angle coverage with undersampling and limited-angle scenarios. We start by loading the Shepp-Logan phantom with 128×128 pixels and setting up the detector configuration to adequately capture all projections. The steps are as follows:

- **Full-Angle Sinogram:** We generate a complete sinogram using 180 projections evenly spaced across 180°. This sinogram captures the full range of angles without any undersampling.

- **Undersampled Sinogram:** By selecting every third angle from the full-angle projection set, we simulate an undersampled scenario. This method effectively reduces the number of projections by a factor of three, mimicking conditions where time or radiation exposure must be minimized.

- **Limited-Angle Sinogram:** We generate a sinogram where data between 60 and 120 degrees are artificially set to zero, representing a limited-angle acquisition. This simulates practical situations where certain angles are obstructed or not scanned.

Each sinogram is computed using the ASTRA Toolbox to simulate the parallel beam projection geometry. The following figure illustrates the generated sinograms, highlighting each configuration's differences and challenges.

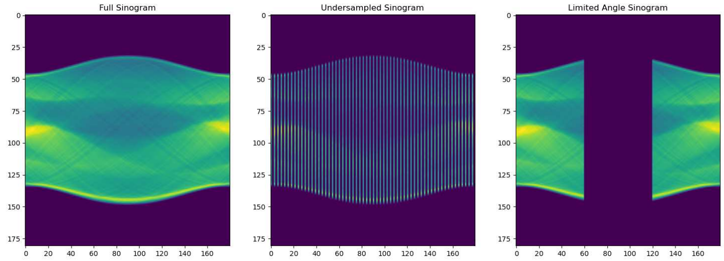


Figure 22: Sinograms generated for the Shepp-Logan phantom: Full, Undersampled, and Limited Angle scenarios.

We first addressed incomplete data in sinograms through inpainting, filling missing measurements with zeros, followed by image reconstruction using filtered backprojection via the ASTRA Toolbox. We construct the projection geometry and perform the reconstruction from the manipulated sinograms. The visualization in Fig 23 compares the results of reconstructions from undersampled and limited angle sinograms, illustrating the impact of inpainting on image quality.

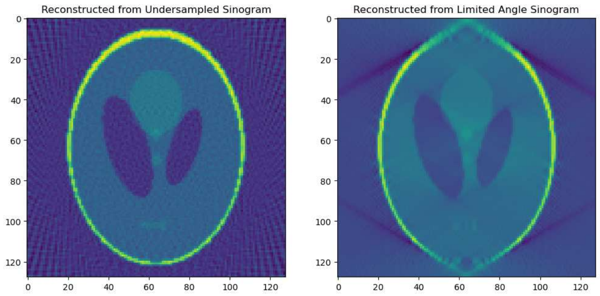


Figure 23: Reconstructed images from inpainted sinograms by filling with zeros: (a) Undersampled and (b) Limited angle.

2.2 Inpainting Sinograms Using Isotropic Laplacian

To address the missing data in sinograms resulting from undersampling and limited angle measurements, we employed an inpainting strategy using an isotropic Laplacian matrix. This approach is based on the diffusion process described by the equation:

$$\frac{\partial g}{\partial t} = -\alpha \mathcal{L}(g), \quad g \in \Omega$$

where $\mathcal{L} = -\nabla^2$ represents the Laplacian operator, α is a scaling factor, and g denotes the sinogram data. The domain Ω includes regions of the sinogram with missing data, and the boundary conditions are set such that $g = g_{\text{obs}}$ on $\partial\Omega$, where g_{obs} is the observed data.

The Laplacian matrix $\mathcal{L}$ is configured for each pixel in the sinogram to connect with its immediate neighbors, ensuring smooth transitions across the inpainted areas. This matrix setup propagates known information into the regions with missing data, iteratively filling in these areas to create a complete dataset.

The inpainting process adjusts the matrix diagonal to enforce known values directly from the sinogram where data exists and iteratively solves for unknowns where data is missing. This results in reconstructed sinograms that are subsequently used for image reconstruction via filtered backprojection.

Figure 24 shows the original and inpainted sinograms for undersampled and limited angle cases by using the isotropic Laplacian. The effects of the inpainting process are evident in the continuity and clarity of the sinogram structures, promising improved reconstruction accuracy.

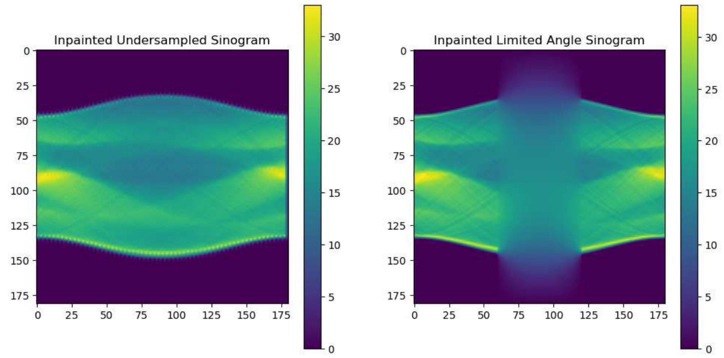


Figure 24: visualization of inpainted sinograms by isotropic Laplacian. Left: inpainted undersampled, Right: Inpainted limited angle.

2.3 Reconstruction from Inpainted Sinograms

Following the inpainting of undersampled and limited angle sinograms, we reconstruct the original images using filtered backprojection implemented via the ASTRA toolbox. The `filtered_backprojection` function employs the ASTRA library's capabilities to handle sinograms directly and reconstruct images based on the provided geometry and projection data.

Both inpainted sinograms, one from the undersampled scenario and the other from the limited angle condition, were used to recover the images. This process highlights the potential of inpainting in mitigating artifacts and improving image quality in scenarios where traditional reconstruction would result in significant loss of detail or introduce artifacts.

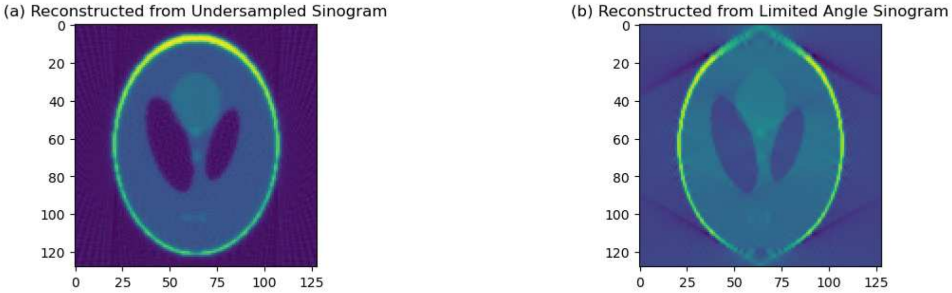


Figure 25: (a) Reconstructed image from the inpainted undersampled sinogram showcasing enhanced clarity and detail. (b) Reconstructed image from the inpainted limited angle sinogram illustrating the improvements in image quality due to inpainting techniques applied to limited data regions.

Fig .25 showcases the improvements in reconstruction when using inpainted data, compared to reconstructions from directly sampled sinogram data, which often result in artifacts or incomplete image recovery.

2.4 Image Domain Denoising

Post-reconstruction denoising was applied to images to enhance quality further. We utilize wavelet denoising, which reduces noise while preserving edges and details. We tested two types of wavelets: Haar and Symlet-5, each known for their properties in handling image details.

1. **Haar Wavelet Denoising:** The choice of Haar wavelets is due to their ability to capture abrupt changes effectively, which is beneficial for images with distinct, sharp edges. The specific parameters used were:

    - Wavelet decomposition level set to 3 to balance detail capture and computational efficiency.
    - A soft thresholding approach, with the threshold set at 20% of the maximum absolute detail coefficient, was utilized to reduce noise while preserving edges.
    - The image was then reconstructed from these thresholded coefficients to produce the denoised output.

    This configuration was selected after a visual assessment of the denoising results, ensuring it effectively reduced noise while maintaining important image details.

2. **Symlet-5 Wavelet Denoising:** Symlet-5 wavelets, chosen for their symmetry and smoother characteristics compared to Haar wavelets, were used to achieve a less aggressive and more visually pleasing denoising effect. The denoising steps included:

    - Using the same decomposition level of 3 to maintain consistency with the Haar setup.

- Employing a soft thresholding method with the threshold set to 20% of the highest detail coefficient, aimed at smoothing while preserving as much of the image structure as possible.
- The image was reconstructed from these modified coefficients to yield the denoised result.

Similar to the Haar approach, the parameters for Symlet-5 denoising were determined through visual evaluation of the denoising effects, focusing on reaching a balance between noise reduction and detail preservation.

Both methods were evaluated on images reconstructed from undersampled and limited-angle sinograms. The efficacy of each denoising technique was assessed through visual measures in Fig .26 and Fig .27.

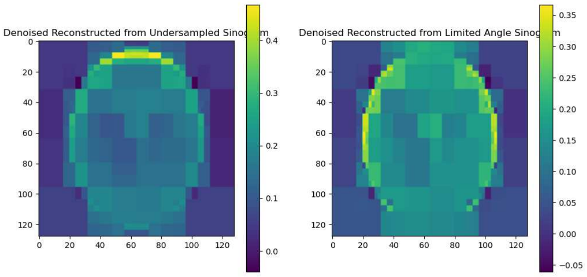

Denoised Reconstructed from Undersampled Sinogram
0
20
40
60
80
100
120
0 20 40 60 80 100 120
0.4
0.3
0.2
0.1
0.0
Denoised Reconstructed from Limited Angle Sinogram
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05

Figure 26: Denoised reconstructed images using Haar wavelets. The left image is from the undersampled sinogram, and the right is from the limited-angle sinogram.

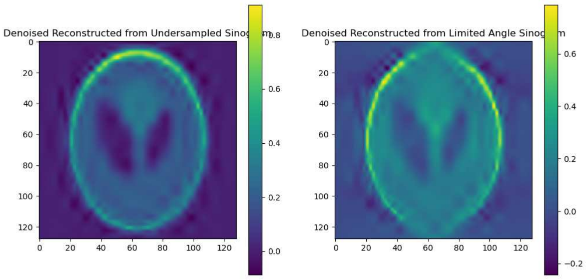

Denoised Reconstructed from Undersampled Sinogram
0
20
40
60
80
100
120
0 20 40 60 80 100 120
0.8
0.6
0.4
0.2
0.0
Denoised Reconstructed from Limited Angle Sinogram
0.6
0.4
0.2
0.0
−0.2

Figure 27: Denoised reconstructed images using Symlet-5 wavelets. The left image is from the undersampled sinogram, and the right is from the limited-angle sinogram.

These figures highlight the differences in denoising performance and image quality between the two wavelet types, offering insights into their suitability for specific imaging conditions.

2.5 Incorporating Smoothed TV Inpainting

To explore more advanced regularization techniques[1], we employed smoothed Total Variation (TV) inpainting to address missing data in sinograms. This approach leverages an anisotropic regularization scheme to enhance data accuracy while maintaining image details. The TV regularization was implemented as follows:

1. **Gradient Computation:** Calculate the gradients along both the horizontal (Dx) and vertical (Dy) directions using first-order differences.

2. **Weight Adjustment:** Adapt the regularization strength dynamically based on the local gradient magnitudes to preserve edges and reduce smoothing across edges.

3. **System Assembly:** Formulate the problem as a linear system, where the regularization term (**R**) adds to the data fidelity term scaled by the mask. The linear system is represented as $A^T A + R$, where A is a diagonal matrix representing the known data locations.

The mathematical formulation of the TV term integrated into the system is:

$$R = D_x^T M_x D_x + D_y^T M_y D_y$$

where M_x and M_y are diagonal matrices containing the weights derived from the exponential decay of the squared gradients, controlled by the parameter α. This term ensures that the inpainting respects the image structure while filling in the gaps efficiently.

Due to the large matrices involved, the system is solved using sparse linear algebra techniques, particularly the conjugate gradient method. Results are reconstructed from the inpainted sinograms using filtered backprojection, demonstrating significant improvements in image quality and reduction of artifacts typically associated with incomplete data.

Figures 28 showcase the reconstructed images from undersampled and limited angle sinograms, respectively, after applying the smoothed TV inpainting. These images illustrate the effectiveness of this method in producing clearer and more accurate reconstructions compared to traditional methods.

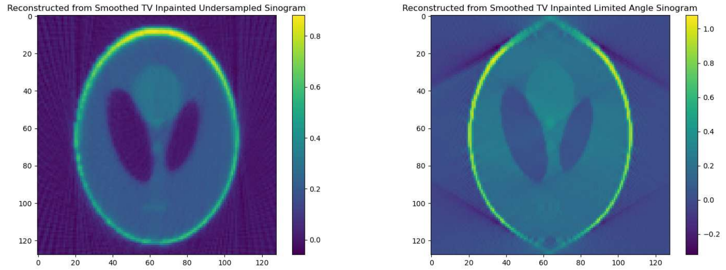


Figure 28: Comparison of reconstructions using smoothed total variation inpainting for undersampled (left) and limited angle (right) sinograms.

These results confirm the benefits of integrating anisotropic regularizers in the reconstruction process, providing a robust framework for handling various types of data incompleteness in computed tomography.

References

[1] Tovey, R., Benning, M., Brune, C., Lagerwerf, M.J., Collins, S.M., Leary, R.K., Midgley, P.A., Schönlieb, C.B.: Directional sinogram inpainting for limited angle tomography. Inverse problems **35**(2), 024004 (2019)

Index of comments

1.1	good
3.1	good
19.1	good