

# Model-based Image Reconstruction in Computed Tomography: From Iterative To Deep Learning Approaches

Henrique Medeiros Dos Reis

May 1, 2024

## Abstract

This essay surveys recent approaches to model-based image reconstruction in computed tomography (CT), ranging from iterative optimization to deep learning methods. It first introduces the mathematical foundations of CT imaging and formulates the reconstruction problem as a linear inverse problem. Traditional iterative optimization algorithms like proximal gradient descent and FISTA are explored, incorporating sparsity-promoting regularizers such as L1 norm and wavelet transformations. The essay then goes into more recent techniques, including total variation regularization and the Chambolle-Pock primal-dual algorithm for saddle point optimization. Moving away from handcrafted regularizers, the essay investigates deep learning-based approaches that use neural networks to learn data-driven regularizers, which then can be used in optimization algorithms via the plug-and-play framework. Experiments are conducted on a real CT scan dataset of a walnut, providing a comparative analysis of the different methods' performance in terms of reconstruction quality when compared to the ground truth. The essay highlights the potential of combining optimization algorithms with learned priors from deep neural networks for further advancements in high-quality CT image reconstruction.

# 1 Image Formation in Computed Tomography

A computed tomography (CT) scan involves generating an image from thousands of X-ray beams at various angles. Initially, understanding a single beam's interaction with some medium, such as muscle, brain tissue, or an object, is crucial. The matter will always absorb some energy from the X-ray beam, and the remaining intensity is measured by a detector, typically film in conventional X-rays. However, conventional X-rays lack depth perception due to the medium's inhomogeneity. CT scans address this by measuring intensity changes from X-rays passing through the medium at different angles, enabling the determination of absorption variations within the medium. This process involves subdividing a grid into squares, assigning grey-scale values based on absorption properties, and analyzing intensity changes across X-rays to construct an image consistent with the measurements, aiming to reveal structural details within the medium.

In simplifying the analysis of X-ray behavior, several assumptions are made: X-ray beams are considered monochromatic, with photons of the same energy level propagating at a constant frequency. The intensity of the beam at a point is proportional to the number of photons passing through it. X-ray beams are assumed to have zero width and do not refract or diffract. Every substance has an attenuation coefficient, representing the proportion of photons absorbed per millimeter, with bone having a high coefficient, air a low one, and water in between. The Hounsfield unit compares the attenuation coefficient of a substance with that of water. When an X-ray beam passes through a medium, the loss of intensity over a small interval is described by Beer's law, where the rate of change of intensity per millimeter is proportional to both beam intensity and attenuation coefficient[1]. The attenuation coefficient function  $\mu(x, y)$ , where  $(x, y)$  are spatial Cartesian coordinates, which expresses the spatial dependence of attenuation coefficients in the object to be imaged. Since different materials (e.g., bone, water, air) have substantially different Hounsfield units, the

function  $\mu$  can provide detailed anatomical information. From measured X-ray intensity, we can determine the integral of  $\mu$  along the path of the X-ray. To describe the trajectories of X-rays used in a CT scan, it is convenient to use the following parameterization of lines in a plane.

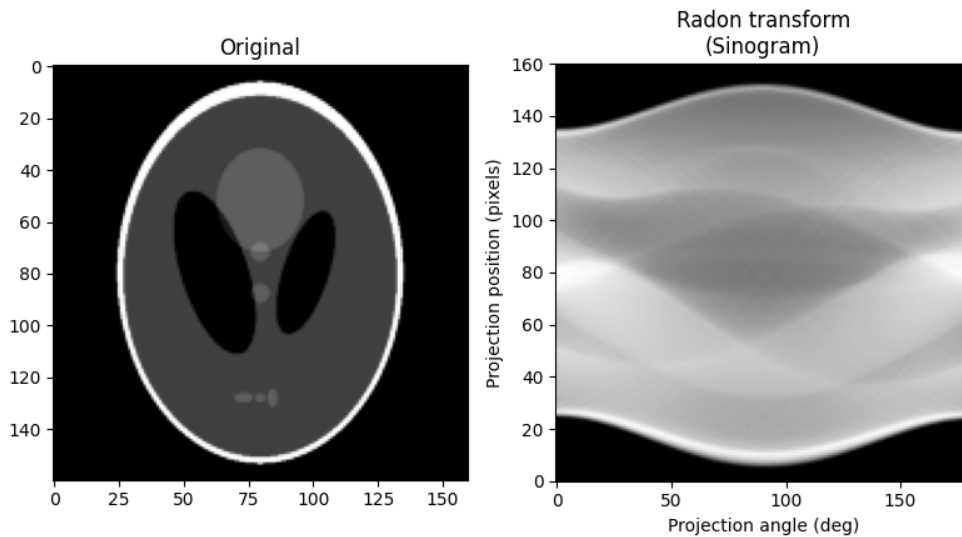


Figure 1: Illustration of the Radon transform

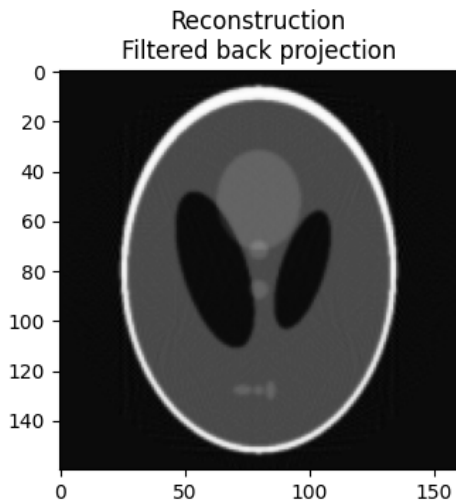


Figure 2: Illustration of Filtered back projection

Formally, a line  $\ell_{t,\theta}$  is defined as the line passing through the point  $(t \cos(\theta), t \sin(\theta))$

and perpendicular to the unit vector  $\mathbf{n} = (\cos(\theta), \sin(\theta))$ . This parametrization using the pair  $(t, \theta)$  uniquely identifies each line in the plane. Moreover, any point  $(x(s), y(s))$  on the line  $\ell_{t,\theta}$  can be expressed as  $(x(s), y(s)) = (t \cos(\theta) - s \sin(\theta), t \sin(\theta) + s \cos(\theta))$ , where  $s$  is a scalar parameter. This parameterization enables the calculation of line integrals along  $\ell_{t,\theta}$ , which is precisely what X-ray emission/detection machines measure. Where all the lines integrals computed for a fixed projection angle is called a “view”, and all the views combined is called a sinogram.

The image on the left of Figure 1 shows the original object, the Shepp-Logan phantom. The image on the right of Figure 1 depicts the Radon transform (sinogram) of the original image. The Radon transform

$$R\{\mu\}(t, \theta) = \int_{\ell_{t,\theta}} \mu(x, y) ds(x, y)$$

represents the projection data obtained by integrating the image intensity along various angles. The sinogram displays these projections as a function of the projection angle  $\theta$  and the offset from the origin  $t$ .

The Radon transform is known to be invertible, meaning that the original image can be reconstructed from its projections[1]. This inversion process is known as the filtered back-projection algorithm, which is illustrated in the Figure 2 [2]. The filtered back-projection (FBP) involves applying a filter to the projections and then backprojecting them to reconstruct the original image.

However, despite the diagnostic power of CT scans, challenges persist in the imaging process. One significant challenge is the trade-off between image quality and radiation exposure. While CT scans provide detailed anatomical information, repeated exposure to ionizing radiation can pose risks to patients. Striking a balance between obtaining high-quality images and minimizing radiation dosage remains an ongoing concern in medical imaging. Addition-

ally, reducing the radiation dose causes the reconstructed CT image to have more artifacts and noise, complicating the task of extracting accurate information.

“Sparse-view” sampling is a technique used in CT scans to reduce the radiation dose received by patients. It involves acquiring fewer projections on the sinogram of the scanned object compared to a conventional CT scan. By collecting only a subset of the number of projections, the radiation exposure is reduced. However, the image quality suffers from it, we can see the Sheep-Logan phantom with a “sparse-view” acquisition in Figure3.

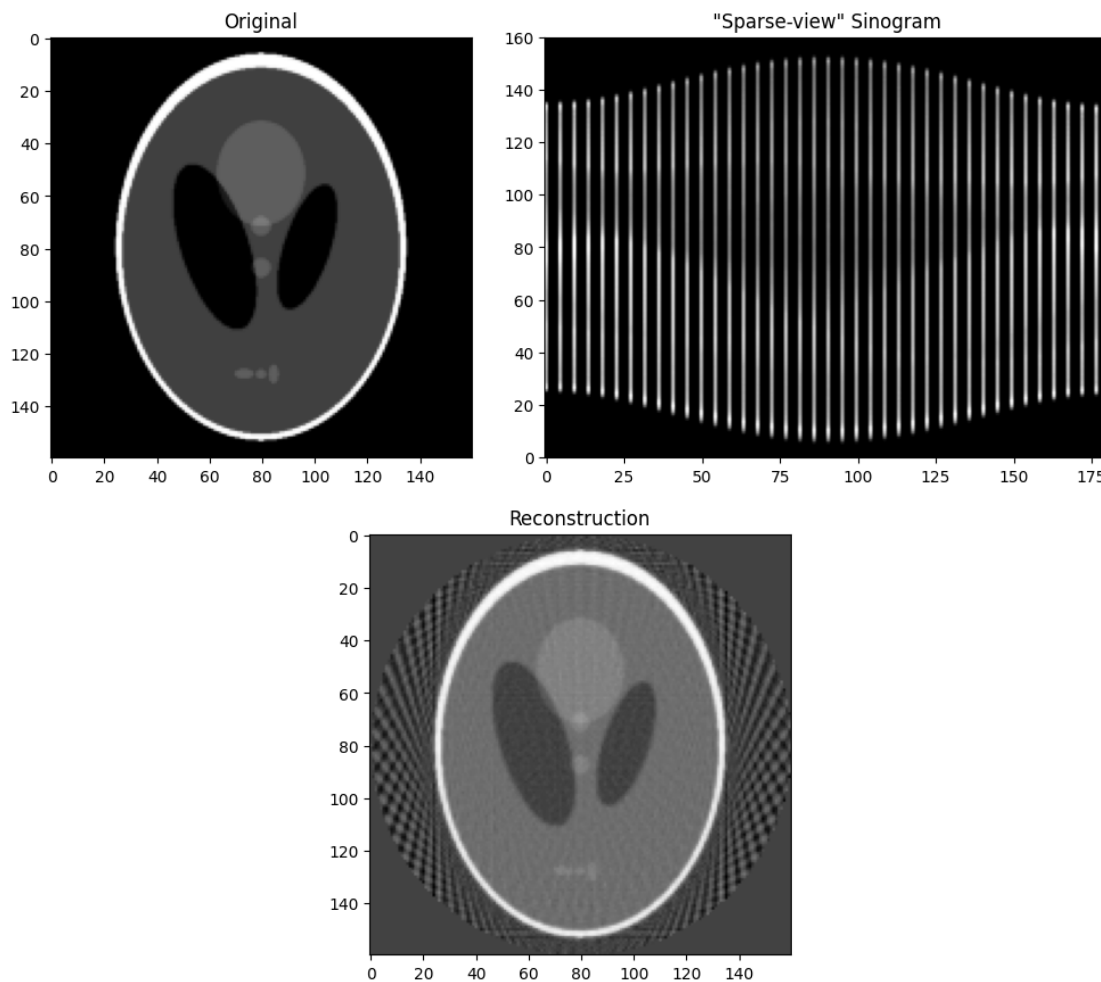


Figure 3: Demonstration of Sheep-Logan phantom, its “sparse-view” sinogram and the FBP reconstruction

In the context of this paper, all simulations and results will be based on a dataset that

includes a CT scan of a walnut [3]. This specific choice allows for a controlled and well-defined imaging scenario, enabling a focused exploration of image reconstruction techniques. The research aims to showcase the efficacy of different mathematical models and machine learning approaches in optimizing a function that will output the reconstructed image, comparing its results to the ground truth.

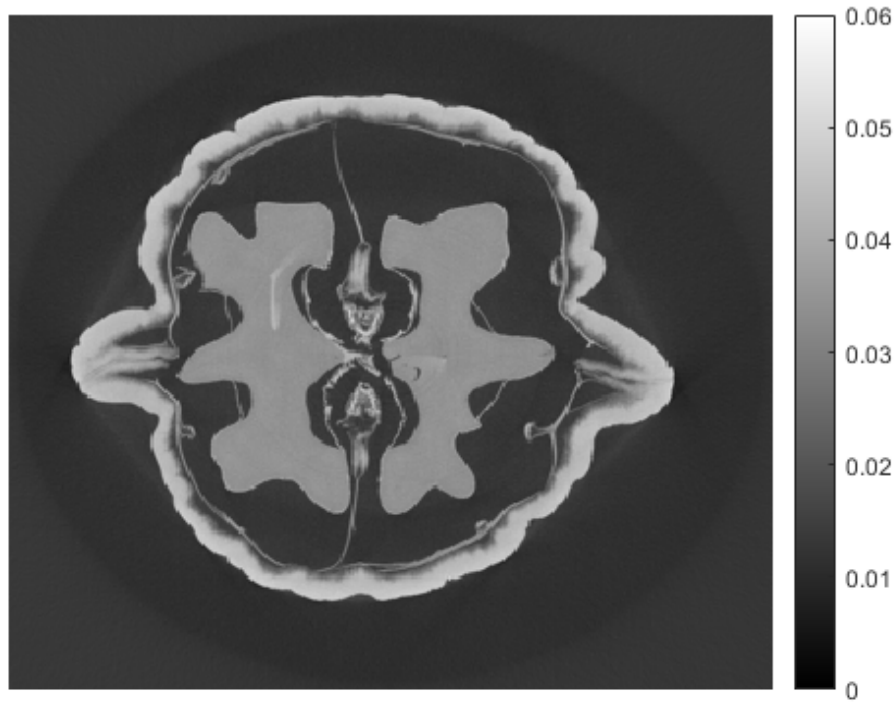


Figure 4: Ground truth image, it shows the an interior 2D slice of the walnut that we are attempting to recover.

## 2 Linear Inverse Problems in Medical Images

Reconstruction of CT scans can be posed as a linear inverse problem, i.e., the solution of a linear system of equations[1]:

$$\mathbf{m} = A\mathbf{x} + \mathbf{n}$$

Here,  $\mathbf{y} \in \mathbb{R}^m$  represents the measurements,  $A \in \mathbb{R}^{m \times n}$  is the forward model,  $\mathbf{x} \in \mathbb{R}^n$  is the vectorized image, and  $\mathbf{n} \in \mathbb{R}^m$  is the noise vector. The primary objective in these problems is to reconstruct the image  $\mathbf{x}$  given the linear noisy measurements  $\mathbf{y}$  and some prior knowledge of the matrix  $A$ .

Addressing the linear inverse problem in X-ray CT scan involves framing it as a least squares problem.

$$\min_{\mathbf{x}} \frac{1}{2} \|A\mathbf{x} - \mathbf{m}\|_2^2$$

However, for certain types of low dose acquisitions, the forward model matrix  $A$ , has a non-trivial null space, which leads to infinitely many solutions to the least squares problem. Therefore, the question arises: Out of the many solutions consistent with measurements, which one should we choose as the reconstruction?

Prior information about the type of images being recovered is crucial to resolving this issue. Regularization is a popular approach for this purpose, where a penalty function is added to the least squares optimization problem. The penalty function, or regularizer, is designed to enforce certain characteristics or constraints on the solution. By including a regularizer, it is possible to find an unique solution, provided the resulting optimization problem can be efficiently solved.

In recent years, machine learning techniques have been employed to enhance the regularization process. Instead of manually designing a regularizer, machine learning methods allow one to learn a regularizer from the data itself, giving a more automatized process. These learned regularizers can then be integrated into the optimization framework, allowing for a more adaptive and data-driven approach to image reconstruction.

### 3 Sparsity Promoting Regularization Techniques

To mitigate the challenges arising from the nontrivial null space in image reconstruction, one common strategy is to introduce regularization. A commonly used approach is to minimize the following objective function:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{m}\|_2^2 + \lambda \|\mathbf{x}\|_2^2$$

The quadratic regularization term  $\lambda \|\mathbf{x}\|_2^2$  gives a unique solution to the least squares problem, but it may not always yield a useful result. For example, in Figure 5 we show a reconstruction obtained by solving the above regularized least squares problem. Note that the reconstruction has significant noise and artifacts, which indicates that this type of regularization is not necessarily helpful in achieving an accurate reconstruction. This creates the need to study different types of regularization that will give a useful unique result.

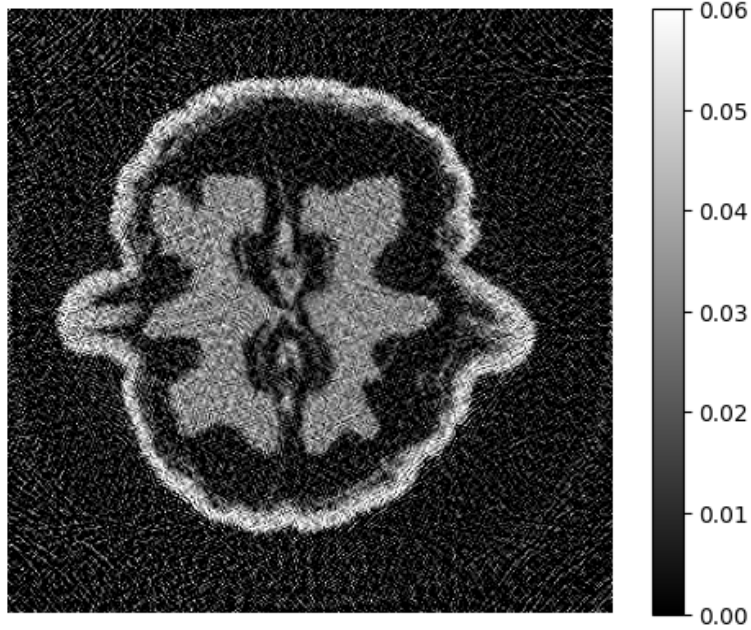


Figure 5: Using regularized least squares with  $\lambda \|\mathbf{x}\|_2^2$  as the regularizer,  $\lambda = 0.1$



In achieving accurate image reconstruction, the selection of the correct regularizer is a crucial step. The regularization term plays an important role in balancing the trade-off between fidelity to the observed data and the desired properties of the solution. However, not all regularization methods are equally effective for every application. Depending on the specific characteristics of the problem and the nature of the data, different types of regularization may be more appropriate. For instance, in certain cases, promoting sparsity in a basis or under a linear transformation regularization might better preserve important features in the image as we discuss in detail below.

### 3.1 Beyond Quadratic Regularizers

In order to attempt improving the reconstructed images, we give a comparative analysis focusing on the role of sparsity-promoting regularizers in image reconstruction. The investigation starts with the introduction of regularizers into our least squares minimization framework.

First, we introduce the  $L^1$ -norm applied to wavelet transform coefficients as the regularizer[4]. This regularizer encourages sparsity in the wavelet basis, aligning with the modeling assumption that natural images are compressible in the wavelet basis. In order to tackle this optimization problem, we investigate two algorithms, such as the Proximal Gradient Descent algorithm and the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [5] [6].

Expanding the exploration, we investigate another regularizer called “total variation”, which penalizes the  $L^1$ -norm of the partial derivatives of the image. This modeling assumption emphasizes the importance of preserving edges and discontinuities in reconstructed images. The resulting optimization problem can be converted to a saddle point problem. Hence, we employ the Chambolle-Pock Primal Dual algorithm[7], which is specifically designed for optimizing saddle point problems.

Finally, we investigate the concept of a “learned” regularizer, which leverages deep learn-

ing approaches [8] to adaptively promote images resembling the training data. This allows the machine to learn and integrate regularization directly from the data. The resulting learned regularizers are then incorporated into the Plug-and-Play algorithm framework [9]. This methodology attempts to enhance the accuracy and efficiency of CT image reconstruction.

Then in order to evaluate the efficacy of the proposed regularizers and their corresponding optimization algorithms, we conduct a comparison using real CT data, which is the walnut dataset [3]. The walnut dataset serves as a benchmark for assessing the performance of our sparse image reconstruction methods, providing insights into the generalizability, robustness, and speed of our approach. Through analysis of reconstructed images, we evaluate the ability of each regularizer to capture details, and preserve edges.

## 4 L1-Norm as the regularizer

In the 1990s, regularizers promoting sparsity of the image in a known basis or under a known linear transformation became a popular alternative to traditional regularization techniques. One notable approach is sparse linear regression, formulated as:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{m}\|^2 + \lambda \text{nnz}(\mathbf{x})$$

Here,  $\text{nnz}(\mathbf{x})$  represents a function outputting the number of non-zero entries in the vector  $\mathbf{x}$ . The underlying idea is to impose a “penalty” for a high number of nonzero entries.

However, this strategy introduces a new challenge—the objective function is computationally demanding to optimize, constituting an NP-hard problem [10]. As a result, optimizing this function becomes a non-trivial task in terms of computational complexity.

To address the optimization challenges posed by the non-smooth function  $\text{nnz}(\mathbf{x})$ , a “convex relaxation” is necessary. Convexity refers to the property that a function lies below

its chords, ensuring every local minimum is a global minimum.

A convex relaxation, denoted as  $u(\mathbf{x})$ , is introduced, satisfying the conditions of convexity ( $u(\mathbf{x})$  is convex) and  $u(\mathbf{x}) \leq nnz(\mathbf{x})$ . The convex envelope  $\tilde{f}$  is then defined as the “highest” convex relaxation of  $nnz(\mathbf{x})$ , ensuring  $nnz(\mathbf{x}) \geq \tilde{f}(\mathbf{x}) \geq u(\mathbf{x})$ . Restricting to vectors  $\mathbf{x} \in \mathbb{R}^n$  whose entries are bounded as  $|x_i| \leq 1$  for all  $i = 1, \dots, n$ , the convex envelope of  $nnz(x)$  is the  $\ell^1$  norm,  $\|\mathbf{x}\|_1$ . This is because  $\ell^1$  is easily shown to be convex, and we have the following inequality:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i| = \sum_{i:x_i \neq 0} |x_i| \leq \sum_{i:x_i \neq 0} 1 = nnz(\mathbf{x})$$

Replacing  $nnz(x)$  with  $\|\mathbf{x}\|_1$  leads to the modified optimization problem:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{m}\|^2 + \lambda \|\mathbf{x}\|_1$$

Since the objective is the sum of two convex functions, the whole objective is convex. Although this problem is convex, because of the  $\ell^1$  norm it remains non-smooth, which means it is not possible to apply a smooth optimization technique, such as Gradient Descent or Newton’s Method. This problem requires a different class of optimization algorithms, known as proximal methods, as described below.

## 4.1 Proximal Gradient Descent

The Proximal Gradient Descent algorithm[5] aims to minimize the sum of two functions  $f(\mathbf{x})$  and  $g(\mathbf{x})$ . This is achieved through the following iterative procedure:

---

**Algorithm 1** Proximal Gradient Descent

---

$L \leftarrow L(f)$  ▷ A Lipschitz constant of  $\nabla f$   
 $\mathbf{x}_0 = \text{Initial guess}$   
 $\tau \leftarrow \frac{1}{L}$   
**for**  $k \leftarrow 0, 1, 2, \dots$  **do**  
     $\mathbf{z}_{k+1} \leftarrow \mathbf{x}_k - \tau \nabla f(\mathbf{x}_k)$  ▷ Gradient Step with respect to  $f$   
     $\mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} \lambda \|\mathbf{x}\|_1 + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{z}_{k+1}\|^2$  ▷ Denoted as  $\text{prox}_g(\mathbf{z}_{k+1}, \tau)$   
**end for**

---

Note that in each iteration, PCG requires solving another optimization problem, however, for some functions  $g(x)$  this sub-problem is computationally efficient, making the algorithm practical.

For the specific problem at hand, where  $g(\mathbf{x}) = \lambda \|\mathbf{x}\|_1$ , the minimization task reduces to finding  $\arg \min_{\mathbf{x}} \lambda \|\mathbf{x}\|_1 + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{z}_{k+1}\|^2$ . This optimization problem has a closed-form solutions, and is described by a scalar function  $\sigma(s, \mu)$  applied entry wise to the  $\mathbf{z}_{k+1}$  vector, where  $\sigma$  is the “soft-thresholding” operator defined as:

$$\sigma(s, \mu) = \begin{cases} s + \mu, & \text{if } s < -\mu \\ 0, & \text{if } |s| \leq \mu \\ s - \mu, & \text{if } s > \mu \end{cases}$$

This can be efficiently implemented using  $\sigma(s, \mu) = \max(|s| - \mu, 0) \cdot \text{sign}(s)$ , applied entry-wise in a vector operation. In this case  $\mu = \lambda\tau$ .

## 4.2 Fast Iterative Shrinkage Thresholding Algorithm

Proximal gradient descent can be slow when dealing with matrices  $A$  with large condition numbers, a common scenario in CT imaging application. FISTA[6] (Fast Iterative Shrinkage-Thresholding Algorithm) introduces a “momentum” update to accelerate the convergence of the algorithm.

---

**Algorithm 2** FISTA

---

$L \leftarrow L(f)$   $\triangleright$  A Lipschitz constant of  $\nabla f$   
 $\mathbf{y}_1 \leftarrow \mathbf{x}_0 \in \mathbb{R}^n, t_1 \leftarrow 1, \tau \leftarrow \frac{1}{L}$   
**for**  $k \leftarrow 1, 2, 3, \dots$  **do**  
     $\mathbf{z}_k \leftarrow \mathbf{y}_k - \tau A^\top (A\mathbf{x}_k - \mathbf{m})$   
     $\mathbf{x}_k \leftarrow \max(|\mathbf{z}_k| - \lambda\tau, 0) \cdot \text{sign}(\mathbf{z}_k)$   $\triangleright$  applied entry-wise  
     $t_{k+1} \leftarrow \frac{1 + \sqrt{1 + 4t_k^2}}{2}$   
     $\mathbf{y}_{k+1} \leftarrow \mathbf{x}_k + \left( \frac{t_k - 1}{t_{k+1}} \right) (\mathbf{x}_k - \mathbf{x}_{k-1})$   
**end for**

---

Here,  $L$  represents the Lipschitz constant of  $\nabla f$ . The algorithm iteratively updates  $\mathbf{z}_k$  using the proximal gradient step, applies the entry-wise soft thresholding to obtain  $\mathbf{x}_k$ , and incorporates a momentum term in  $\mathbf{y}_{k+1}$  to accelerate convergence. The ratio  $\frac{t_k - 1}{t_{k+1}}$  determines the amount of momentum to be applied in each iteration, which gets closer to 0 as we get closer to convergence, ensuring the iterates do not overshoot the minimizer.

### 4.3 Sparse Image Reconstruction with Wavelet Transformations

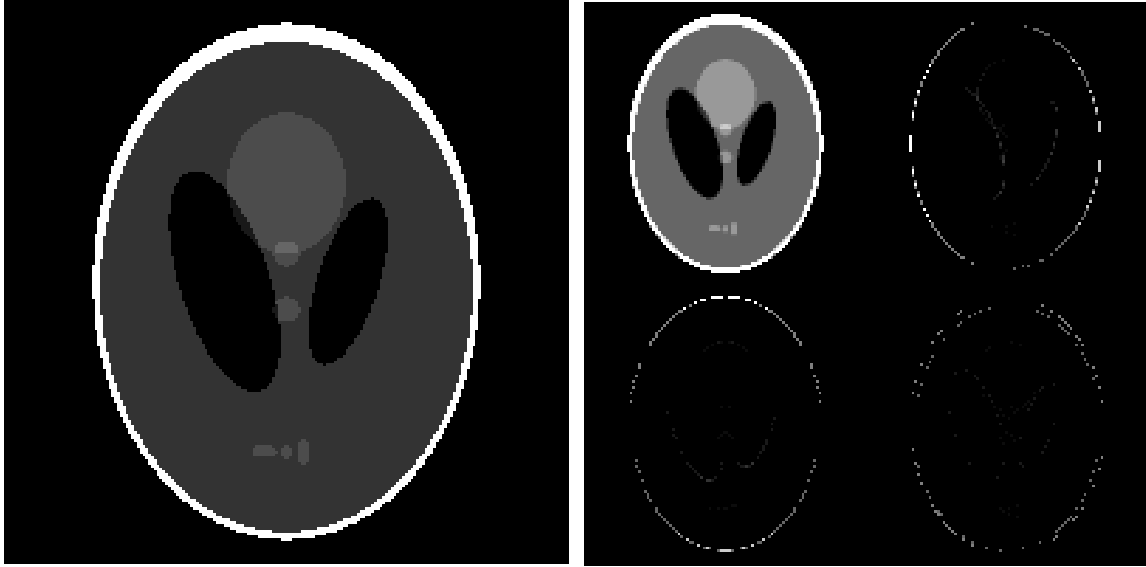
When promoting sparsity in image reconstruction, often images are more sparse in a basis other than the standard pixel basis. One commonly used basis for imaging tasks is a wavelet basis, which decomposes an image into orthogonal features at different scales. If we let  $W$  be the 2D Haar Wavelet transform [4], which converts an image into its Haar Wavelet coefficients, then our modeling assumption is that  $\mathbf{c} = W\mathbf{x}$  is sparse rather than  $\mathbf{x}$  itself.

In the Figure 6a below, we can see the image of the Sheep-Logan phantom, and after we apply  $W$  to the phantom, we get the resulting Figure 6b. The 2D Haar wavelet transform, when applied to the Shepp-Logan phantom image, calculates the differences between the pixel values in the vertical, horizontal, and diagonal directions. The Haar wavelet transform

applies the following four  $2 \times 2$  filters to the images

$$f_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, f_2 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, f_3 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \text{ and } f_4 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

This is followed by a downsampling by a factor of 2 in each dimension. When reorganized, the filtered and downsampled images have the same dimensions as the original image.



(a) Sheep Logan

(b) Wavelet Transform of Sheep Logan

Figure 6: Wavelet transform of the Sheep-Logan phantom

We aim to recover an image that will be sparse in the domain of the Wavelet transform. This leads to the following model and algorithm:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{m}\|^2 + \lambda \|\mathbf{W}\mathbf{x}\|_1$$

To address the optimization challenge in the second step, where  $\|\mathbf{W}\mathbf{x}\|_1$  is not straightforward to optimize, a change of variables is introduced. Assuming  $\mathbf{W}$  is invertible, let  $\mathbf{c} = \mathbf{W}\mathbf{x} \Leftrightarrow \mathbf{x} = \mathbf{W}^{-1}\mathbf{c}$ . The cost function becomes:

$$\min_{\mathbf{c}} \frac{1}{2} \|AW^{-1}\mathbf{c} - \mathbf{m}\|^2 + \lambda \|\mathbf{c}\|_1$$

This is then solved using Proximal Gradient Descent or FISTA, replacing  $A$  with  $\tilde{A} = AW^{-1}$  and  $\mathbf{x}$  with  $\mathbf{c}$ .

We take advantage of the fact that the matrix  $W$  is orthogonal, then calculating its inverse is simple, since  $W^{-1} = W^\top$ , and applying  $W^\top$  has similar complexity to applying  $W$ , requiring an upsampling step followed by four filtering operations.

## 4.4 Results

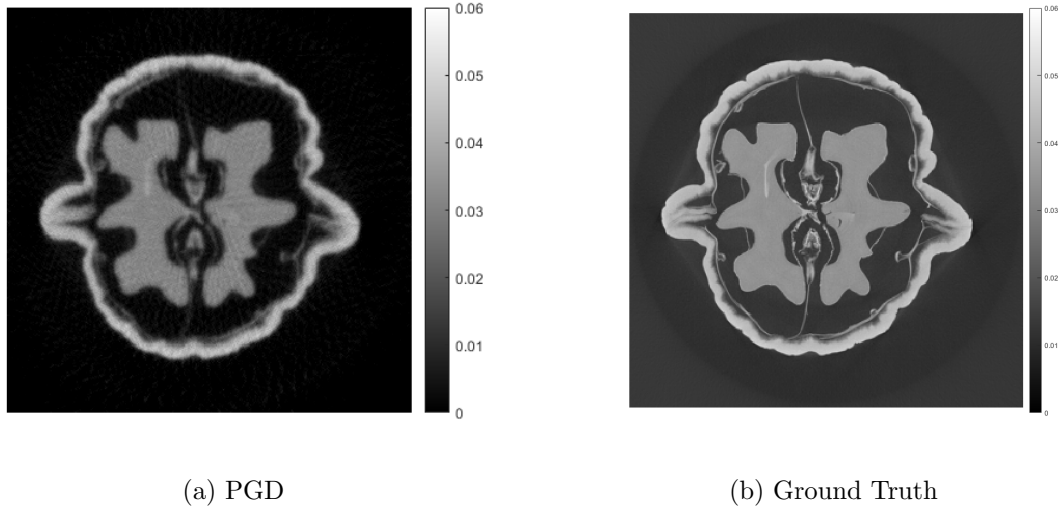


Figure 7: L1 norm regularized least squares optimized with PGD,  $\lambda = 0.01$ , 100 iterations, run-time: 11.5569s vs Ground Truth

We can observe the results of applying the Proximal Gradient Descent (PGD) method for image reconstruction using L1 norm regularization on a CT scan of a walnut.

Figure 7 the comparison between the PGD reconstruction and the ground truth image after 100 iterations. We can see that the PGD reconstruction captures the overall structure and shape of the walnut, but there are noticeable differences in the details and sharpness

compared to the ground truth.

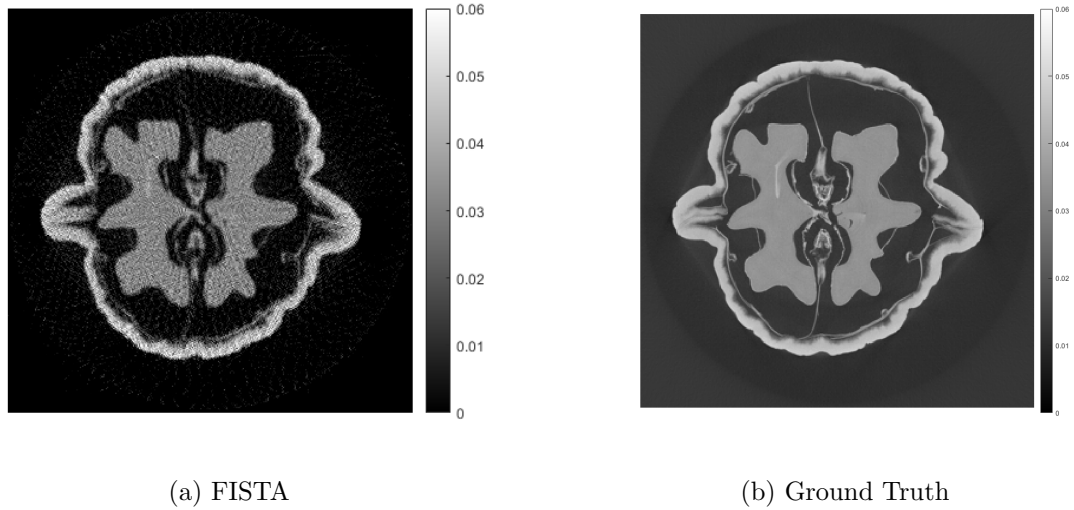


Figure 8: L1 norm regularized least squares optimized with FISTA algorithm,  $\lambda = 0.01$ , 100 iterations, run-time: 10.5603s vs Ground Truth

These images showcase the results of applying the FISTA method for image reconstruction using the L1 norm regularization on the CT scan of a walnut.

Figure 8 compares the FISTA reconstruction with the ground truth image after 100 iterations. Similar to the previous method, the FISTA reconstruction captures the overall structure and shape of the walnut, but there are differences in the details and sharpness when compared to the ground truth. There seems to be some noise added, but the overall reconstruction seems to capture more detail relative to applying 100 iterations of PGD.



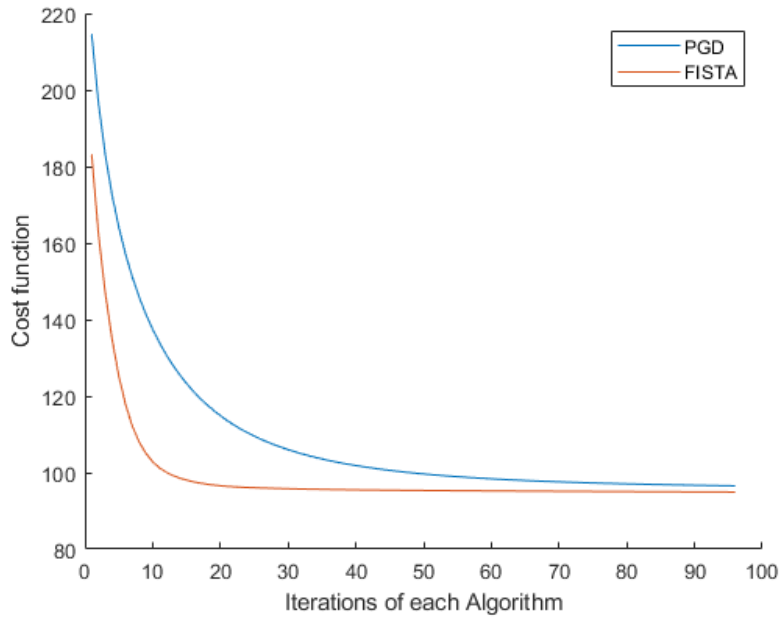
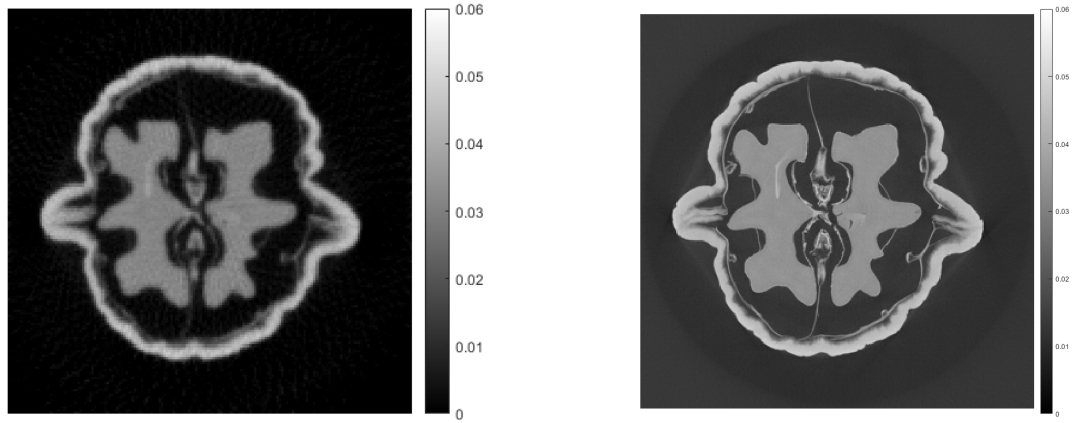


Figure 9: Cost function using the L1 norm as the regularizer evaluated at different iterates  $\mathbf{x}_k$ ,  $\lambda = 0.01$

The FISTA algorithm is an accelerated variant of the PGD, as shown in Figure 9 above. FISTA is known for its fast convergence rate. By incorporating the L1 norm regularization, the method promotes sparsity in the solution, aiding in the recovery of sharp edges and fine details in the reconstructed image.

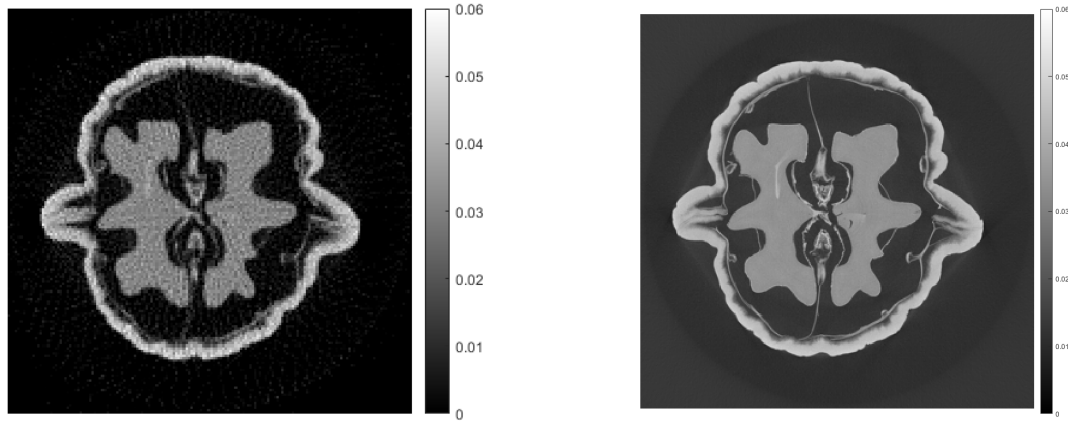


(a) PGD with wavelet

(b) Ground Truth

Figure 10: Wavelet L1 norm regularized least squares, optimized with PGD algorithm, 100 iterations,  $\lambda = 0.015$ , run-time: 13.1835s vs Ground Truth

Figure 10 displays the results after 100 iterations of PGD with wavelet. The reconstructed image seems more detailed than its version without the wavelet transform, although it seems more pixelated. Yet, it is still possible to note some lack of finer details when compared to the ground truth.



(a) FISTA with wavelet

(b) Ground Truth

Figure 11: Wavelet L1 norm regularized least squares, optimized with FISTA algorithm, 100 iterations,  $\lambda = 0.015$ , run-time: 11.9885s vs Ground Truth

Figure 11 displays the results after 100 iterations of FISTA with wavelet. The reconstructed image appears sharper and more detailed, with better preservation of the intricate patterns within the walnut. However, there are still some discrepancies when compared to the ground truth, particularly in the finer details, as the image appears a little more pixelated.

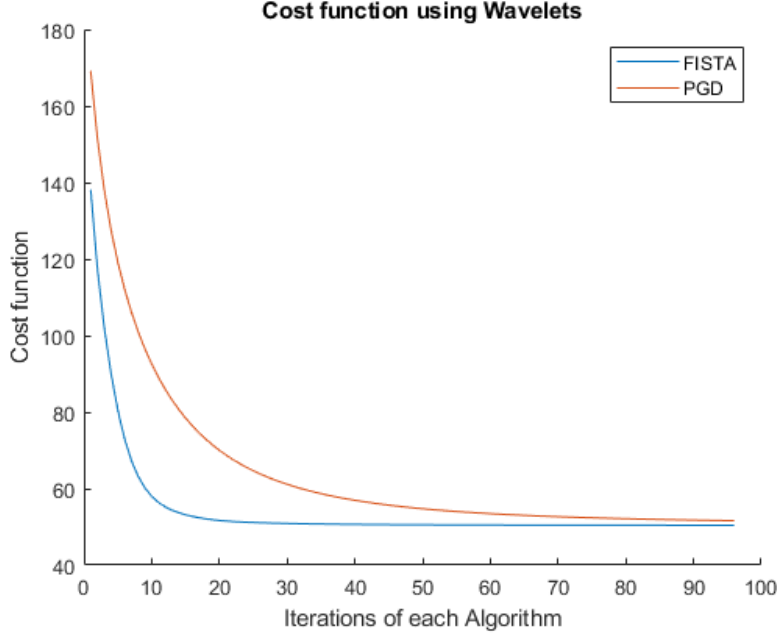


Figure 12: Cost function using the Wavelet L1 norm as the regularizer evaluated at different iterates  $\mathbf{x}_k$ ,  $\lambda = 0.015$

The plot in Figure 12 displays a curve that rapidly decreases initially and then levels off, indicating that the majority of the computational effort is expended in the early iterations of the algorithm. FISTA took roughly 15 iterations to reach the same cost value as PGD reached after 90 iterations.

Comparing these results to previously seen results of FISTA and PGD methods, we can observe that FISTA with both regularizers provides a good result in both reconstruction quality and computational efficiency, since we can see in the cost that it converges much faster than PGD.

## 5 Total Variation Regularization

### 5.1 A saddle point approach for sparse image reconstruction

Enhancing the sparsity-promoting model, a matrix  $K$  is introduced, focusing solely on the vertical and horizontal differences.  $K$  is a vertical concatenation of two matrices  $K_h$  and  $K_v$ , representing two filtering operations that corresponds to horizontal finite differences, and vertical finite differences. Different from the wavelet transform, no downsampling step is applied. The resulting transformed image  $K\mathbf{x}$  has twice the size of image  $\mathbf{x}$ . Similarly to the Wavelet transform, we can visualize the resulting image demonstrated with the Sheep-Logan phantom bellow in Figure 13.

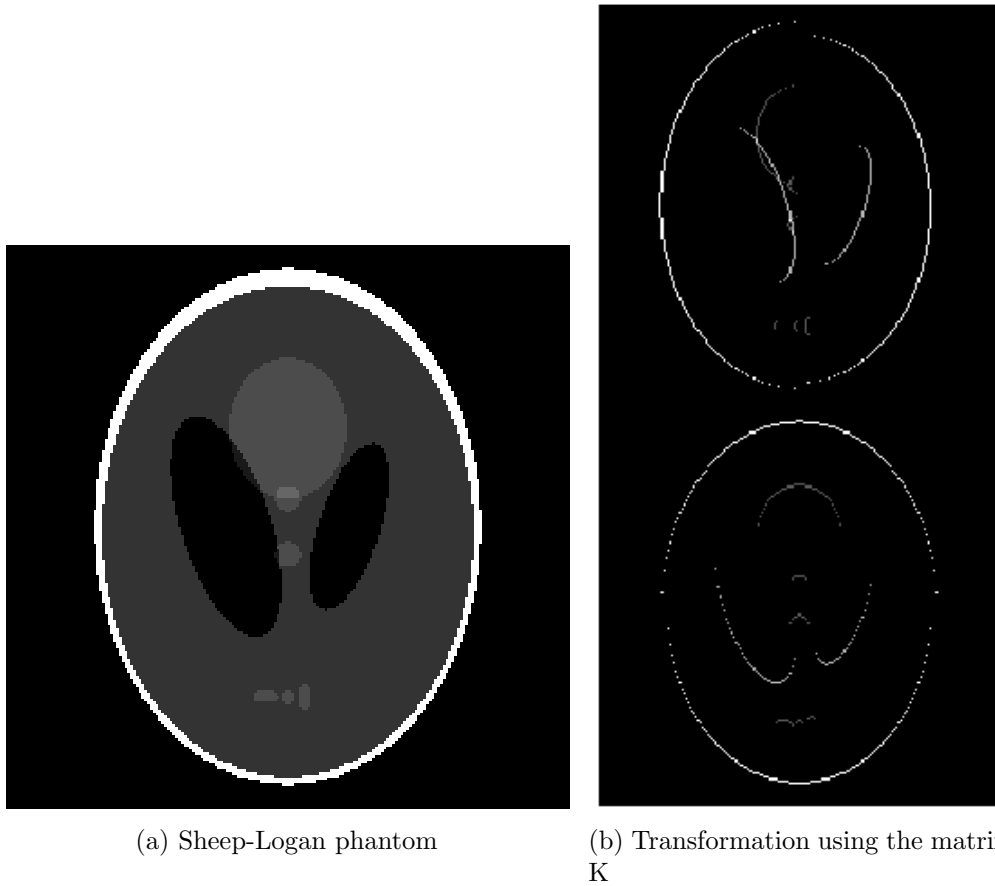


Figure 13: Sheep-Logan phantom and its transformation after multiplying by the  $K$  matrix

Figure 13 demonstrates the application of a modified matrix  $K$  to the Shepp-Logan phantom, which calculates only the vertical and horizontal differences between pixels. The resulting image in Figure 13 (b) highlights the edges and discontinuities present in the original phantom in Figure 13(a), then we can try promoting sparsity in the transform domain. The adjusted cost function becomes:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{K}\mathbf{x}\|_1$$

The objective of matrix  $K$  is to encourage images that are not only sparse but also possess sparse derivatives, emphasizing edges.

However, solving this modified cost function poses a challenge since the matrix  $K$  is not invertible. This renders conventional approaches presented earlier ineffective for this specific problem. To overcome the challenge posed by non-invertibility of the finite differences matrix  $K$ , the problem will need to be reformulated as a concave-convex saddle point problem.

## 5.2 Chambolle-Pock

The Chambolle and Pock (CP) algorithm is primal-dual algorithm [7], meaning that it solves an optimization problem simultaneously with its dual formulation. It will be extra work, since solving two problems, instead of one. On the other hand, it guarantees convergence in both the problems. We can start by formulating the primal minimization problem:

$$\min_{\mathbf{x}} F(\mathbf{B}\mathbf{x}) + G(\mathbf{x})$$

Introducing the variable splitting  $\mathbf{y} = \mathbf{B}\mathbf{x}$ , and taking the convex dual in the  $\mathbf{y}$  variable

yields the equivalent saddle point formulation[7]:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \max_{\mathbf{y} \in \mathbb{R}^k} \{ \langle B\mathbf{x}, \mathbf{y} \rangle_Y + G(\mathbf{x}) - F^*(\mathbf{y}) \}$$

We can then reformulate the problem  $\min_{\mathbf{x}} \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|^2 + \lambda \|K\mathbf{x}\|_1$  to be able to use the CP algorithm, which is tailored for finding the saddle point of two combined functions.

The core idea is to concatenate the CT forward model matrix  $A$  and the finite differences matrix  $K$  matrix into a tall matrix  $B = \begin{bmatrix} A \\ K \end{bmatrix}$ . The cost function is then redefined as:

$$\min_{\mathbf{x} \in \mathcal{X}} F(B\mathbf{x}) + G(\mathbf{x}), \text{ where } F\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}\right) = \frac{1}{2} \|\mathbf{y} - \mathbf{m}\|^2 + \lambda \|\mathbf{z}\|_1, G(\mathbf{x}) = 0$$

$F$  combines both the data fit and total variation terms. Applied to this choice of  $F$ ,  $G$ , and  $B$ , the Chambolle-Pock iterates simplify as follows:

---

**Algorithm 3** CP

---

$L \leftarrow \|B\|_2^2, \tau \leftarrow 1/L, \sigma \leftarrow 1/L, \theta \leftarrow 1$

**for**  $k \leftarrow 1, 2, \dots$  **do**

$\mathbf{p}_{k+1} \leftarrow (\mathbf{p}_k + \sigma(A\mathbf{x} - \mathbf{m})) / (1 + \sigma)$

$\triangleright$  Update for  $\mathbf{p}$

$\mathbf{y}_{k+1} \leftarrow \text{clip}(\mathbf{y}_k, \lambda)$

$\triangleright$  Projection onto  $\ell^\infty$ -norm ball

$\mathbf{x}_k \leftarrow \mathbf{x}_k - \tau A^\top \mathbf{p}_{k+1} - \tau K^\top (\mathbf{y}_{k+1})$

$\triangleright$  Update for  $\mathbf{x}$

$\bar{\mathbf{x}}_{k+1} \leftarrow \mathbf{x}_{k+1} + \theta(\mathbf{x}_{k+1} - \bar{\mathbf{x}}_k)$

$\triangleright$  Update momentum term

**end for**

---

A detailed derivation is given in [7].

Here the clip function is defined as  $\text{clip}(\mathbf{y}_k, \lambda) = \arg \min_{\|\mathbf{y}\|_\infty \leq \lambda} \|\mathbf{y} - \mathbf{z}\|_2^2$ , which clips the values that are larger than a specific threshold value.

### 5.3 Results and comparison

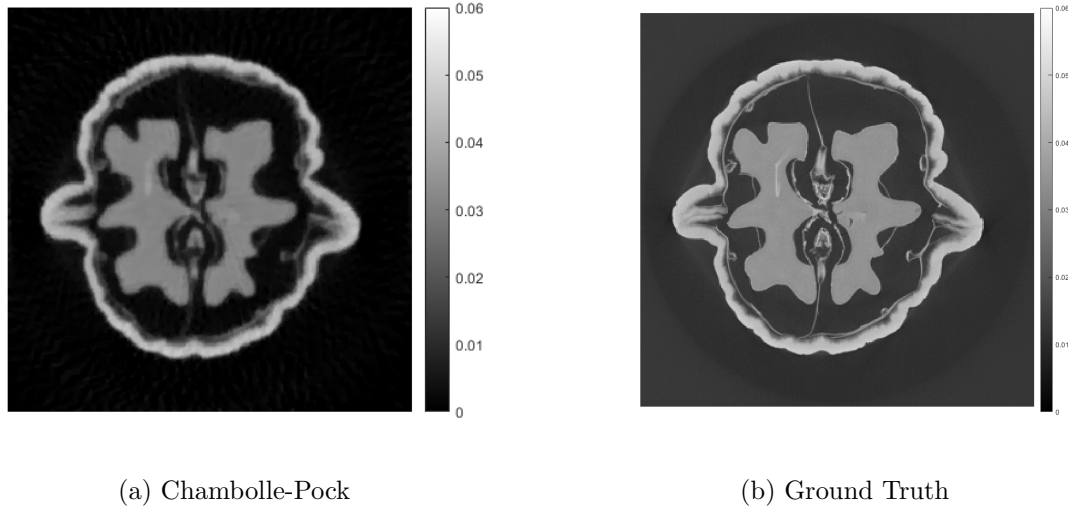


Figure 14: Total Variation regularized least squares optimized using Chambolle-Pock algorithm, 135 iterations,  $\lambda = 0.01$ , run-time: 43.230286s vs Ground Truth

Figure 14 presents the results from the CP algorithm. The reconstructed Figure 14 (a) appears visually similar to the Ground Truth image, and appears to be more piecewise constant than the previous reconstructions using  $L^1$  regularization or wavelet  $L^1$  regularization.

The total variation regularizer is a reasonable choice for the walnut image since it exhibits an approximately piecewise constant. A piecewise constant image can be segmented into distinct regions, where the intensity values within each region are relatively uniform or constant, but there are abrupt changes at the boundaries between these regions. The walnut image consists distinct regions with relatively uniform intensities separated by a well-defined, sharp boundary. The shell region appears to have a consistent grayscale intensity throughout its area, while the background region displays a uniform intensity different from the shell.



## 6 Deep Learning: Letting the Machine Learn the Filters

### 6.1 Introduction to the deep learning approach for image reconstruction

In traditional optimization approaches to image reconstruction, we manually define the regularization term  $R(\mathbf{x})$  in an objective function like:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{m}\|_2^2 + R(\mathbf{x})$$

The regularization term acts as a prior that enforces desired properties on the solution  $\mathbf{x}$ , such as smoothness or sparsity. Manually designing regularization terms often amounts to designing denoising filters by hand, such as wavelet filters, or the finite differencing filters used in total variation regularization.

Recent advances in deep learning allow us to essentially learn these filters automatically. Instead of hand-engineering a regularization term, we can train a neural network as a denoising operator. The network learns an implicit image prior from training data, removing the need to explicitly define a regularization term.

These learned denoisers can act as plug-and-play priors in optimization methods. By replacing a hand-designed regularization term with a learned denoising neural network, we obtain a flexible framework for image recovery. The network is trained once on appropriate data and then plugged into various optimization algorithms as needed.

## 6.2 Using a learned denoiser in a previously seen algorithm

A Convolution Neural Network (CNN) is a cascade of filtering operations, where each layer applies a set of learnable filters to the input, extracting features from the data. These filtering operations are followed by a non-linear activation function, such as ReLU. CNNs sometimes include downsampling operations to reduce the spatial dimensions of the input, and upsampling operations to recover spatial resolution when needed. CNNs are effective for various computer vision and signal processing tasks.

In order to train a CNN we will need a training dataset, and a specific architecture. The training dataset we will be using is a publically available dataset of breast CT phantom images used in (2022) AAPM sparse-view, low-dose reconstruction challenge[11], and the architecture of the CNN is the U-Net [12].

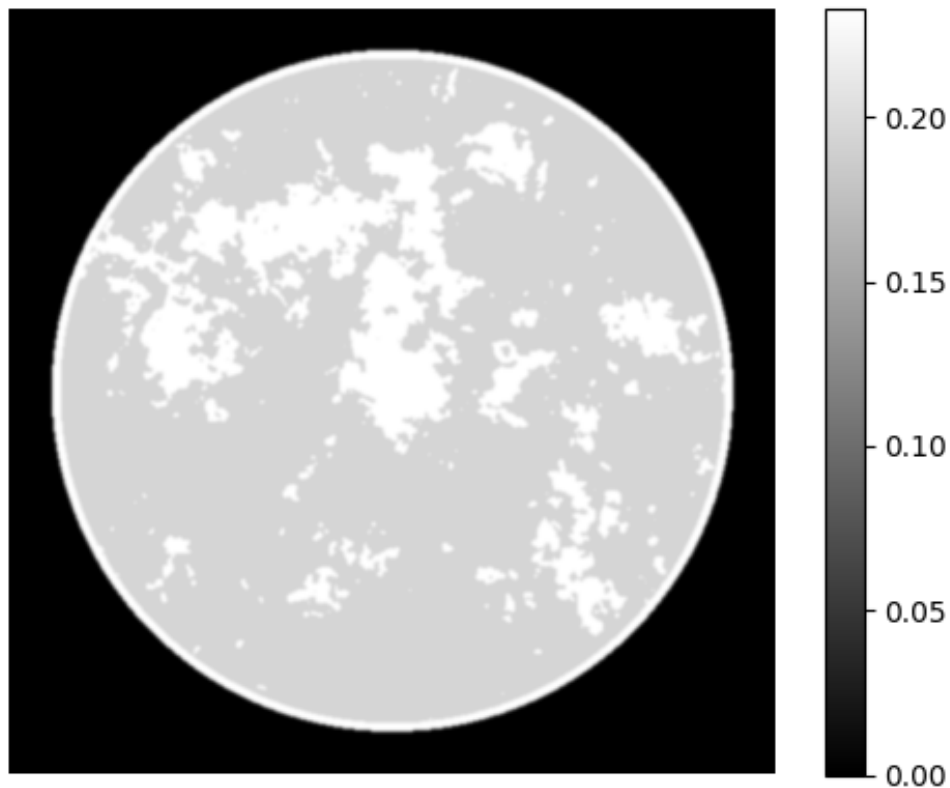


Figure 15: First image in the training dataset.

From the ground truth images, Figure 15, we generate corrupted images, Figure 16, having noise/artifacts similar to what we would expect in a CT imaging context. First, we reshape the whole dataset to match the dimensions of the Walnut image. Then, to each image in the training set, we apply the forward model  $A$ , then add noise, and form an initial reconstruction using 10 iterations of a least squares solver (LSQR) to approximate a least squares solution. Then the network is trained on images that are similar to Figure 16.

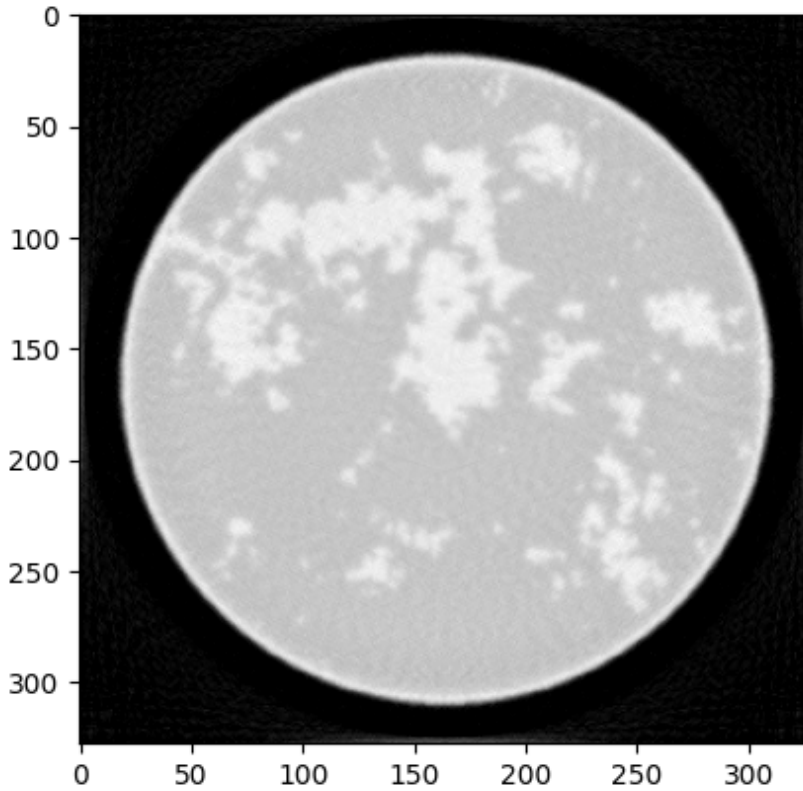


Figure 16: First image in the training dataset after applying forward model  $A$ , adding noise and forming an initial reconstruction using LSQR.

We split the dataset into 80% being the training set, and 20% being the validation dataset. The training process goes over a predefined number of epochs, each epoch representing a complete pass through the entire training dataset. Within each epoch, the training samples are shuffled randomly, ensuring that the model does not become overly reliant on any specific

order of presentation. For each training sample.

With the stage set, the U-Net[12] model takes the noisy/artifacted image and produces its estimation of the clean image. This estimation is then compared to the ground truth image using the Mean Squared Error loss function, described as

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N \|U_{\theta}(\mathbf{y}_i) - \mathbf{x}_i\|^2$$

Where  $U_{\theta}(\cdot)$  is the U-net, and  $\mathbf{y}_i$  is the initial reconstruction of the noisy sinogram data simulated from the ground truth phantom  $\mathbf{x}_i$ . Through the backpropagation algorithm, the gradients of this loss are calculated and propagated backwards through the network, allowing for the adjustment of the model's internal parameters. We trained the U-net using the Adam optimizer with learning rate 0.0001 and trained for 50 epochs using a batch size of 1, the final training MSE was  $3.998360 \times 10^{-8}$ .

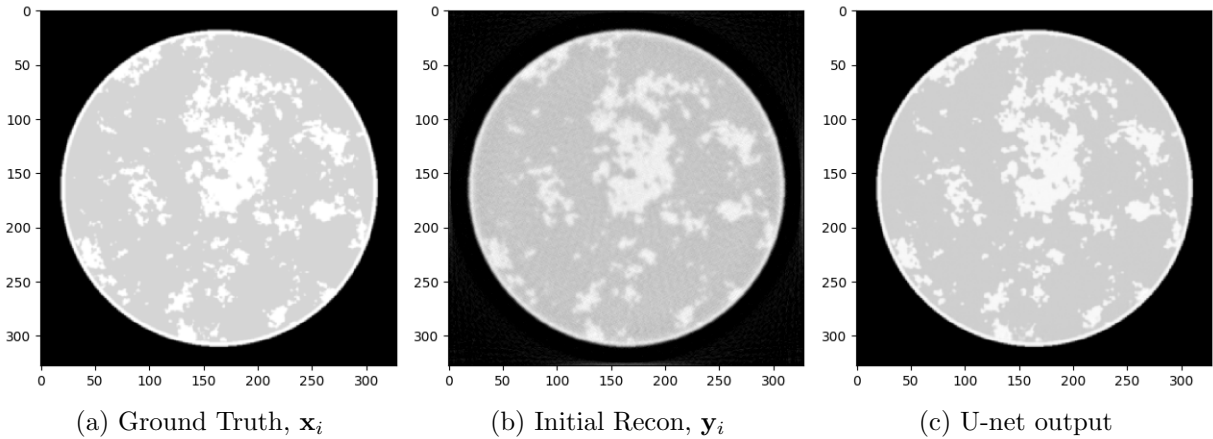


Figure 17: Results of the U-net applied in a single image from the test data

Finally, the resulting CNN is assessed using the validation dataset to see if it is performing as expected, which gave a MSE of  $4.765610 \times 10^{-4}$ . After that, we have acquired a learned denoiser, and can integrate it into an existing algorithm such as Proximal Gradient Descent (PGD) involves straightforward adjustments. Here, we illustrate the incorporation of the

regularizer within the PGD framework:

---

**Algorithm 4** Plug and Play using Proximal Gradient Descent

---

```

 $L \leftarrow L(f)$  ▷ A Lipschitz constant of  $\nabla f$ 
 $\mathbf{x}_0 = \text{Initial guess}$ 
 $\tau \leftarrow \frac{1}{L}$ 
for  $k \leftarrow 0, 1, 2, \dots, K$  do
     $\mathbf{z}_{k+1} \leftarrow \mathbf{x}_k - \tau A^\top (A\mathbf{x}_k - \mathbf{m})$  ▷ Data consistency step
     $\mathbf{x}_{k+1} = (1 - \alpha)\mathbf{x}_k + U_\theta \cdot \text{net}(\mathbf{x}_k)$  ▷ Denoising step
end for

```

---

In this algorithm, the function  $\text{net}()$  represents the output of the learned denoiser, while the parameters  $K$  (number of iterations) and  $\alpha$  (denoising strength) are hyperparameters that require fine-tuning to optimize performance, we will tune  $\alpha$  and set  $K=10$  for computational efficiency.

In many machine learning tasks, the choice of hyper parameters can significantly impact the model's performance, accuracy, and generalization ability. Finding the optimal set of hyper parameters is often a challenging and time-consuming process, as it involves exploring a large space of possible combinations, in our case, the space is not so large, since we only have 1 hyper parameter,  $\alpha$ .

Since the space is relatively small, we can perform a brute-force approach to hyper parameter tuning, where we perform a grid search in a range of values for each hyper parameter, and the model is evaluated at those grid points. For each choice of  $\alpha$ , We looked for the settings that gave the best root mean squared error (RMSE) on a single test image, from the test dataset,  $\alpha$  and RMSE is then stored, allowing for comparison across different hyper parameter combinations, allowing us to pick  $\alpha$  that gave the smallest RMSE.

It is important to note that the brute-force approach can be computationally expensive, especially when dealing with a large number of hyper parameters or when the range of values for each hyper parameter is extensive. In such cases, more advanced techniques may be employed to efficiently explore the hyper parameter space.

## 6.3 Results

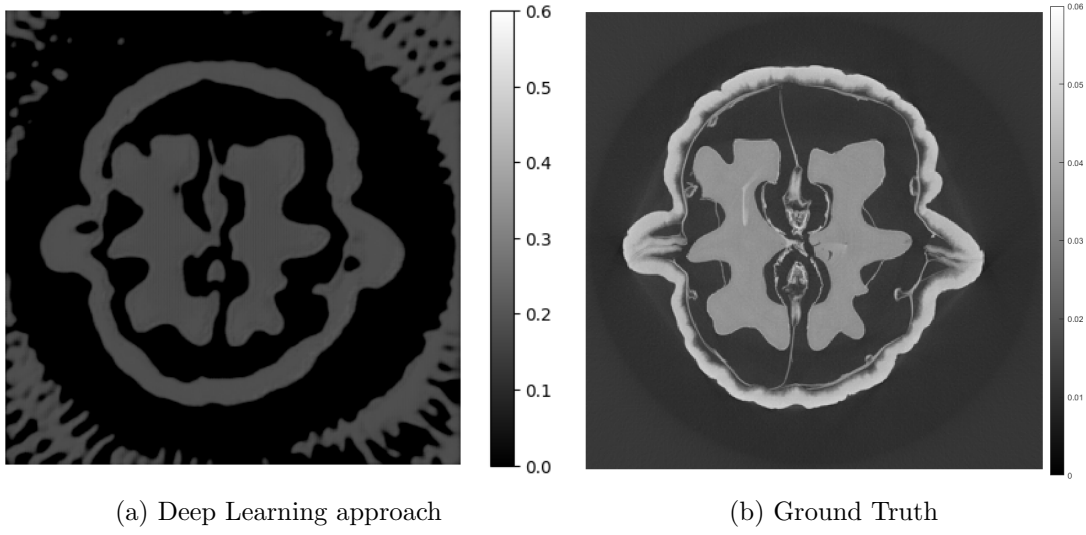


Figure 18: Deep Learning approach,  $\alpha=0.5$ ,  $K=10$  vs Ground Truth

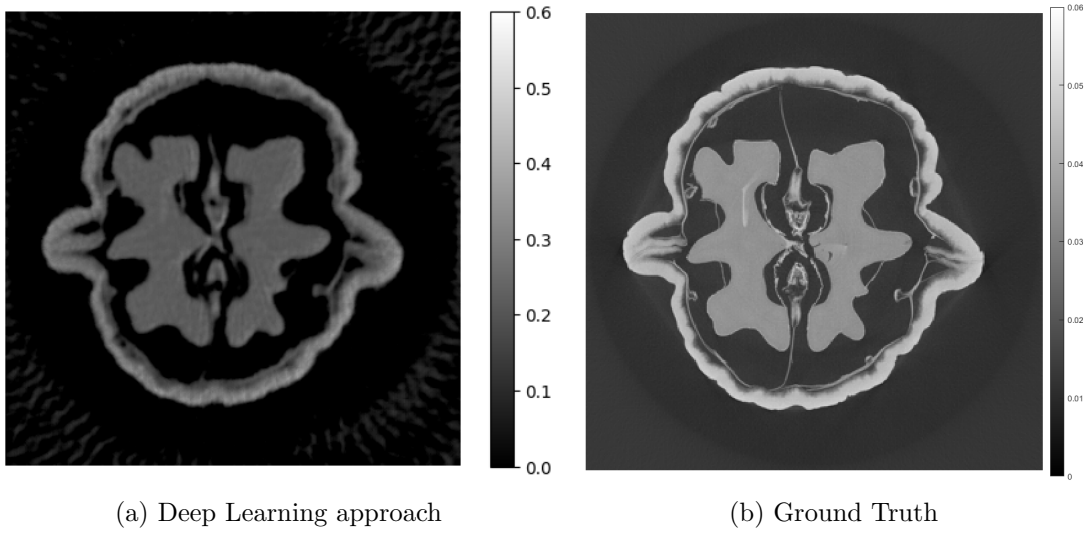
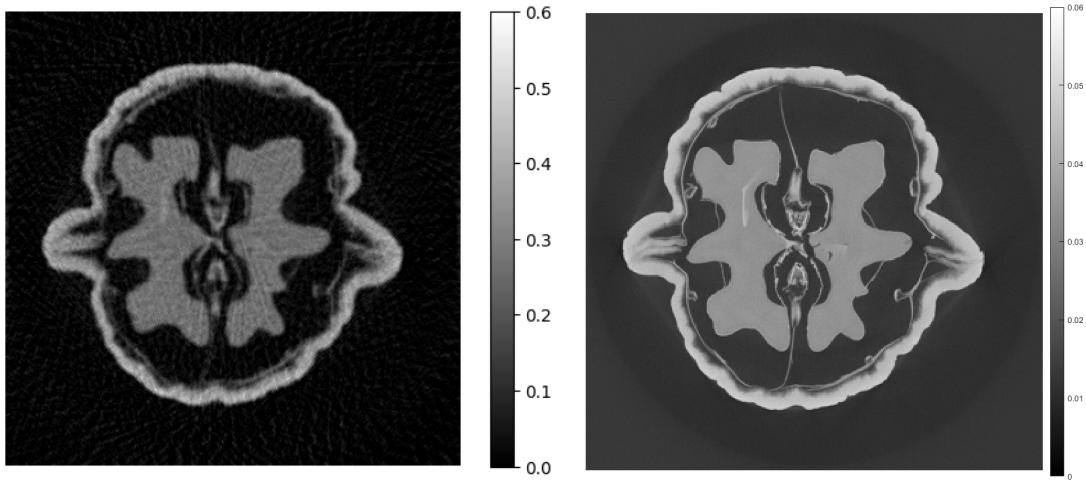


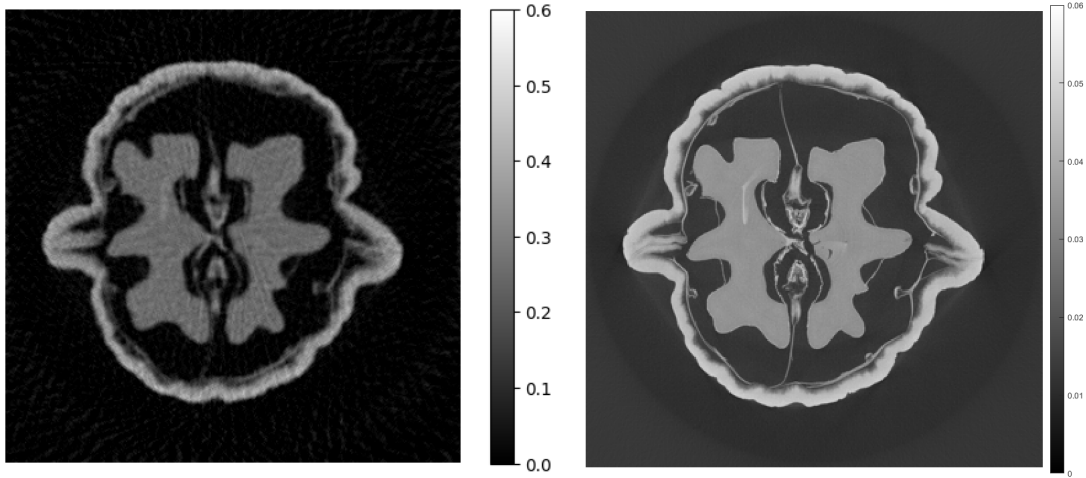
Figure 19: Deep Learning approach,  $\alpha=0.1$ ,  $K=10$  vs Ground Truth



(a) Deep Learning approach

(b) Ground Truth

Figure 20: Deep Learning approach,  $\alpha=0.05$ ,  $K=10$  vs Ground Truth



(a) Deep Learning approach

(b) Ground Truth

Figure 21: Deep Learning approach,  $\alpha=0.02$ ,  $K=10$  (optimal hyper-parameter tuning) vs Ground Truth

While the previously seen optimization methods did an excellent job preserving fine details and textures within the walnut structure, the deep learning reconstruction shows superior overall quality and fidelity to the ground truth image, when aligned with the correct hyper parameters.

On the other hand, when using hyper parameters that were not properly selected, such as  $\alpha = 0.5$ ,  $K = 10$ , our results are not the best, when comparing Figure 19 with the Ground Truth, since it denoised the reconstructed image more than needed, losing fine details in the image. The reconstructed image takes the learned regularizer with a higher impact, causing the reconstructed image to resemble more the training data images, losing most of the details and smoothing almost all the sharp edges we can see in the Ground Truth.

Plug-and-play priors provide a flexible way to integrate learned denoisers into optimization methods. However, there are some limitations, proper training data and network design are critical. Also, the denoiser is treated as a black box, and there is limited interpretability of the network’s learned prior.

## 7 Conclusion

In this essay we have explored various model-based approaches for image reconstruction in computed tomography, ranging from classical iterative optimization techniques to deep learning methodologies. By formulating the reconstruction problem as a linear inverse problem, we went through different types of strategies to address the challenges associated with CT scan image recovery.

Traditional methods like proximal gradient descent and FISTA, coupled with sparsity-promoting regularizers such as the L1 norm and wavelet transformations, demonstrated their effectiveness in recovering structural details while promoting sparsity in the reconstructed images. The total variation regularization approach, implemented through the Chambolle-Pock primal-dual algorithm, performed better in preserving sharp edges and discontinuities, a crucial aspect in many imaging applications, but suffers from longer reconstruction times.

Lastly, we investigated the integration of optimization algorithms and deep learning techniques by training neural networks as data-driven denoisers, and incorporating them into



iterative frameworks like PGD.

While the optimization methods relied on handcrafted regularizers like wavelets and total variation, plug-and-play techniques seek to get the best of both worlds - the speed of learned models with the transparency of optimization. The deep learning approach is a promising technique and with proper tuning, demonstrated promising results when attempting to recover real CT data even when trained on synthetic phantoms.

## 8 Bibliography

- [1] T. G. Feeman, “The mathematics of medical imaging,” *Springer*,, 2010.
- [2] M. J. Willemink and P. B. Noël, “The evolution of image reconstruction for ct—from filtered back projection to artificial intelligence,” *European radiology*, vol. 29, pp. 2185–2195, 2019.
- [3] K. Hämäläinen, L. Harhanen, A. Kallonen, A. Kujanpää, E. Niemi, and S. Siltanen, “Tomographic x-ray data of a walnut,” *arXiv preprint arXiv:1502.04064*, 2015.
- [4] J. Gallier and J. Quaintance, “Linear algebra for computer vision, robotics, and machine learning,” *University of Pennsylvania*, 2019.
- [5] N. Parikh, S. Boyd, *et al.*, “Proximal algorithms,” *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [6] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [7] E. Y. Sidky, J. H. Jørgensen, and X. Pan, “Convex optimization problem prototyping for image reconstruction in computed tomography with the chambolle–pock algorithm,” *Physics in Medicine & Biology*, vol. 57, no. 10, p. 3065, 2012.

- [8] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, “Deep learning techniques for inverse problems in imaging,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.
- [9] K. Wei, A. Aviles-Rivero, J. Liang, Y. Fu, H. Huang, and C.-B. Schönlieb, “Tfnpnp: Tuning-free plug-and-play proximal algorithms with applications to inverse imaging problems,” *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 699–746, 2022.
- [10] T. T. Nguyen, C. Soussen, J. Idier, and E.-H. Djermoune, “Np-hardness of  $\ell_0$  minimization problems: Revision and extension to the non-negative setting,” in *2019 13th International conference on Sampling Theory and Applications (SampTA)*, IEEE, 2019, pp. 1–4.
- [11] E. Y. Sidky and X. Pan, “Report on the aapm deep-learning sparse-view ct grand challenge,” *Medical physics*, vol. 49, no. 8, pp. 4935–4943, 2022.
- [12] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.