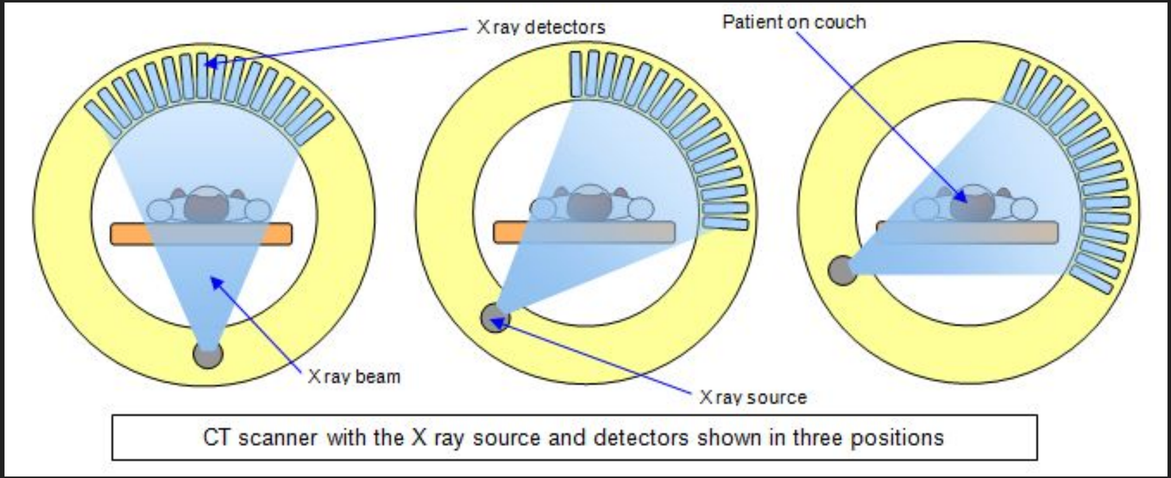
Model-based Image Reconstruction in Computed Tomography: From Iterative To Deep Learning Approaches

Image information in Computed Tomography

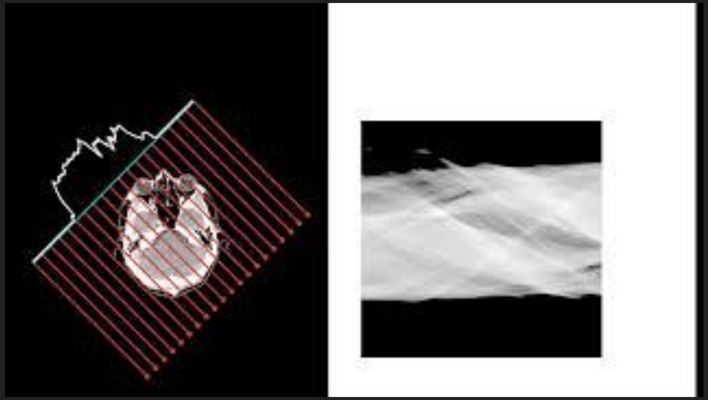
- How to get an image from inside your body?
- A beam of X-rays passes through the body, and the amount that gets absorbed is measured on the other side of the body



Images from: https://www.schoolphysics.co.uk/age16-19/Medical%20physics/text/CT_scanning/index.html, https://en.wikipedia.org/wiki/CT_scan

Image information in Computed Tomography

- When an X-ray beam passes through a medium, the loss of intensity over a small interval is proportional to both beam intensity and attenuation coefficient.
- A CT scan can be modeled as samples of the Radon transform of the spatially-variant attenuation coefficient function (or image)
- The Radon transform represents the projection data obtained by integrating the image intensity along various angles, which is invertible. Meaning that the attenuation image can be reconstructed from its projections
- A sinogram is a visual representation of the Radon transform of an image



Video: https://www.youtube.com/watch?v=q7Rt_OY_7tU

Image information in Computed Tomography

- Trade-off between radiation exposure and image quality

  - The full size sinogram typically has at least the same number of projections as the number of pixels
  - One way to reduce the radiation exposure would be to take fewer projection views, and this is called a “sparse-view” acquisition
  - For example taking only every fourth view, reducing the radiation by a factor of 4

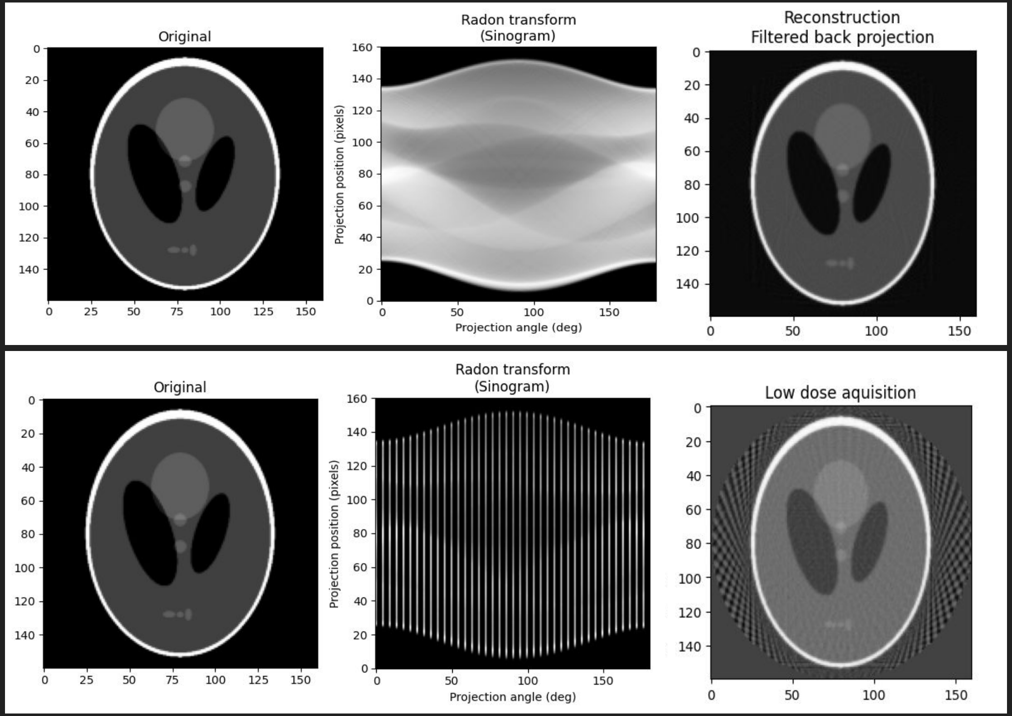
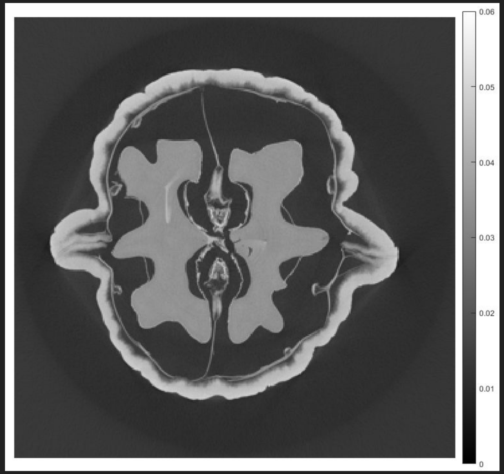


Image information in Computed Tomography

- Simulations based on CT scan of a walnut dataset [2]
- Ground Truth image, acquired using a sinogram of 1200 projections views, the sinogram is then a matrix of size 2296 × 1200
- We will attempt recovering the image using a “sparse-view” sinogram of 120 projections views, and 1/7 detector pixels, which gives a matrix of size 328 × 120



Linear Inverse Problems in Medical Images

- Reconstruction of CT scans can be posed as a linear inverse problem

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

- Least squares formulation

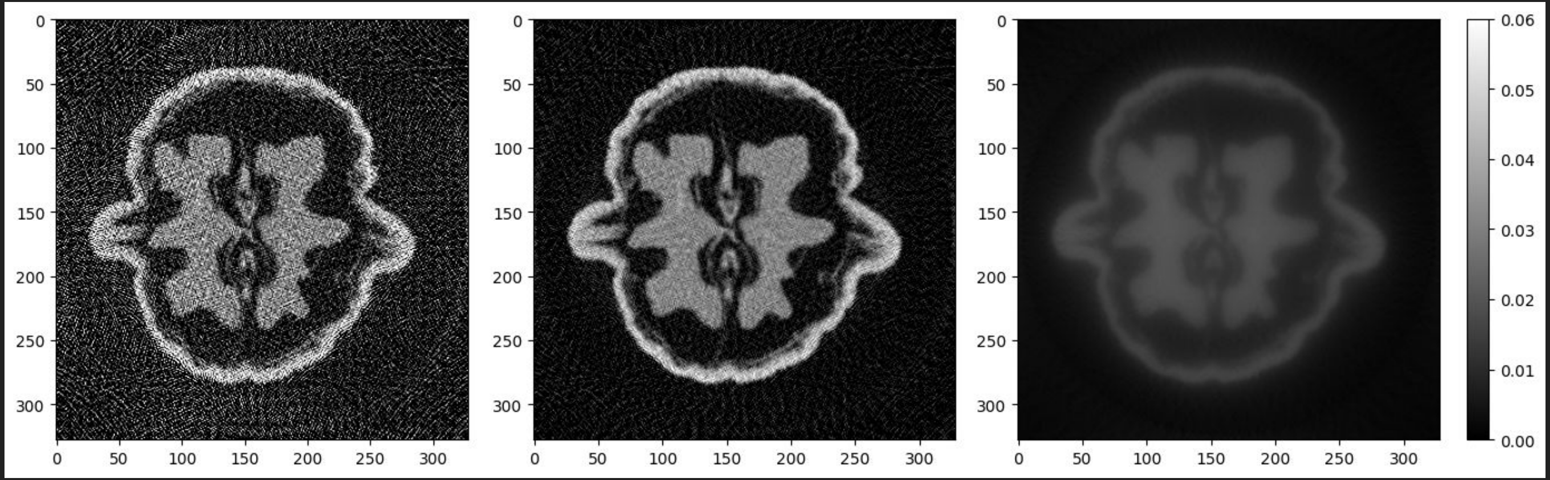
$$\min_{\mathbf{x}} \frac{1}{2}||A\mathbf{x} - \mathbf{m}||_2^2$$

- The matrix A has fewer rows than columns in a low dose acquisition.Thus, A has a null space, and least squares will not have an unique solution
- Which of the multiple solutions should we choose?

Addressing the Issue: Introducing Regularization

- A common approach L2-norm $\min_{\mathbf{x}} \frac{1}{2}\|A\mathbf{x} - \mathbf{m}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$
- May not always give useful solutions
- Trade-off between data fidelity and the desired properties of the solution
- Different types of regularization should be used for different problems

λ = 0.001 λ = 0.01 λ = 0.5



Seeking a New Approach

- Comparative analysis focusing on different regularizers in image reconstruction, commonly used in computed tomography (CT).

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

- Regularizers employed:
  - L1-Norm
  - L1-Norm with Wavelet Transform
  - Total Variation
  - Machine learned denoisers
- Qualitatively evaluate image quality with the walnut data

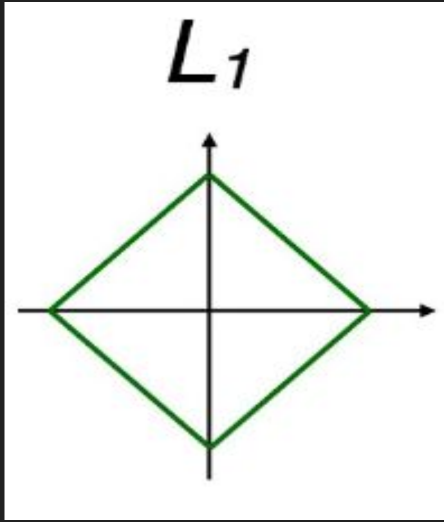
L1-Norm as the regularizer

- Sparse Linear Regression Formulation

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

- Computational challenge - non-smooth and non-convex function
- L1-norm $\lambda\|\mathbf{x}\|_1$ is the convex envelope
- Leads to a convex, but not smooth optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|A\mathbf{x}-\mathbf{m}\|^2 + \lambda\|\mathbf{x}\|_1$$



Proximal Gradient Descent

- The Proximal Gradient Descent algorithm aims to minimize the sum of two functions f(x) and g(x), where f is convex and L-smooth, and g is convex and potentially not smooth [3].

Algorithm 1 Proximal Gradient Descent

$L \leftarrow L(f)$ ▷ A Lipschitz constant of ∇f

$\mathbf{x}_0 =$ Initial guess

$\tau \leftarrow \frac{1}{L}$

for $k \leftarrow 0, 1, 2, \ldots$ **do**

  $\mathbf{z}_{k+1} \leftarrow \mathbf{x}_k - \tau \nabla f(\mathbf{x}_k)$ ▷ Gradient Step with respect to f

  $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} g(\mathbf{x}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{z}_{k+1}\|^2$ ▷ Denoted as $prox_g(\mathbf{z}_{k+1}, \tau)$

end for

- Note: the proximal operator of g(x) involves solving another optimization problem in every iteration

Proximal Gradient Descent

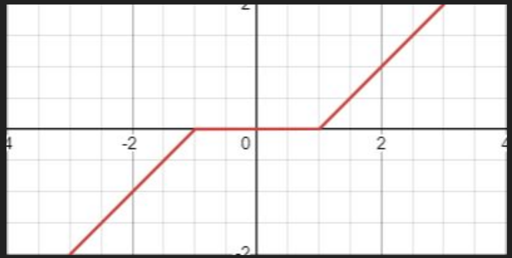
- Specifically for the L1-regularized least squares problem we have

Algorithm 1 Proximal Gradient Descent

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

- But, it has known solution in the case of L1-norm: $\sigma(s, \mu) = max(|s| - \mu, 0) * sign(s)$ applied entrywise
- The updates of $\mathbf{x}_{k+1}$ are:

$$\mathbf{x}_{k+1} = \sigma(\mathbf{z}_{k+1}, \lambda\tau)$$



Fast Iterative Shrinkage Thresholding Algorithm

- Proximal Gradient Descent convergence depends on the condition number of A, which can be large for CT imaging
- A momentum update to accelerate convergence is introduced in [4], called the Fast Iterative Shrinkage Thresholding Algorithm

Algorithm 2 FISTA

$L \leftarrow L(f)$ ▷ A Lipschitz constant of ∇f

$\mathbf{y}_1 \leftarrow \mathbf{x}_0 \in \mathbb{R}^n,\ t_1 \leftarrow 1, \tau \leftarrow \frac{1}{L}$

for $k \leftarrow 1, 2, 3, \ldots$ **do**

  $\mathbf{z}_k \leftarrow \mathbf{y}_k - \tau A^\top (A\mathbf{x}_k - \mathbf{m})$

  $\mathbf{x}_k \leftarrow \max(|\mathbf{z}_k| - \lambda\tau, 0) \cdot \mathrm{sign}(\mathbf{z}_k)$ ▷ 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

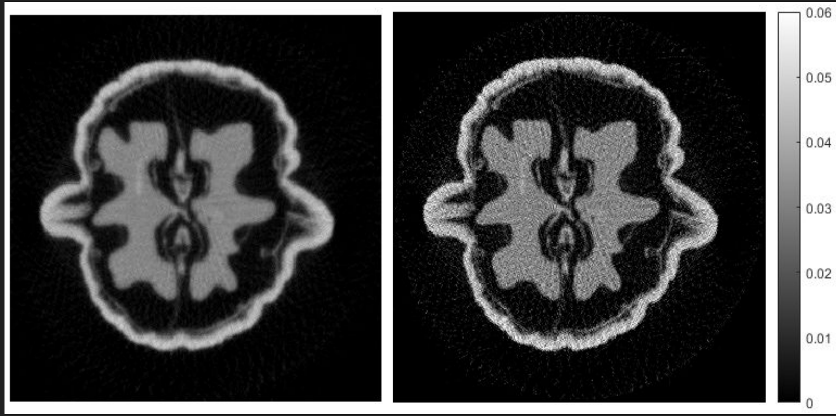
Results using L1-Norm:

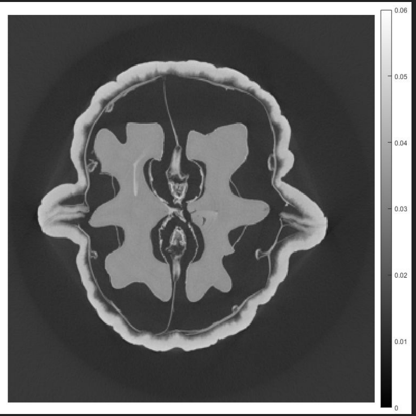
PGD: 100 iterations
Run time:11.5569s
λ = 0.01

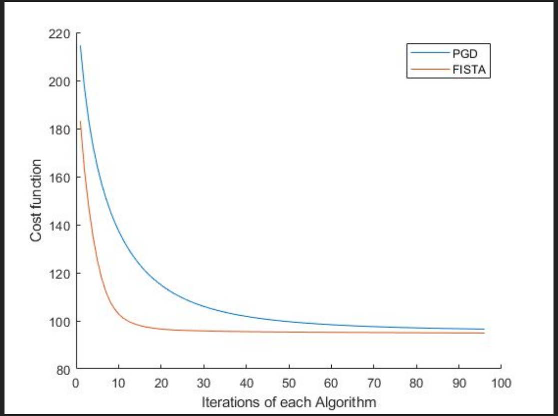
FISTA: 100 iterations
Run time:10.5603s
λ = 0.01

Ground Truth:

Cost vs Iteration:





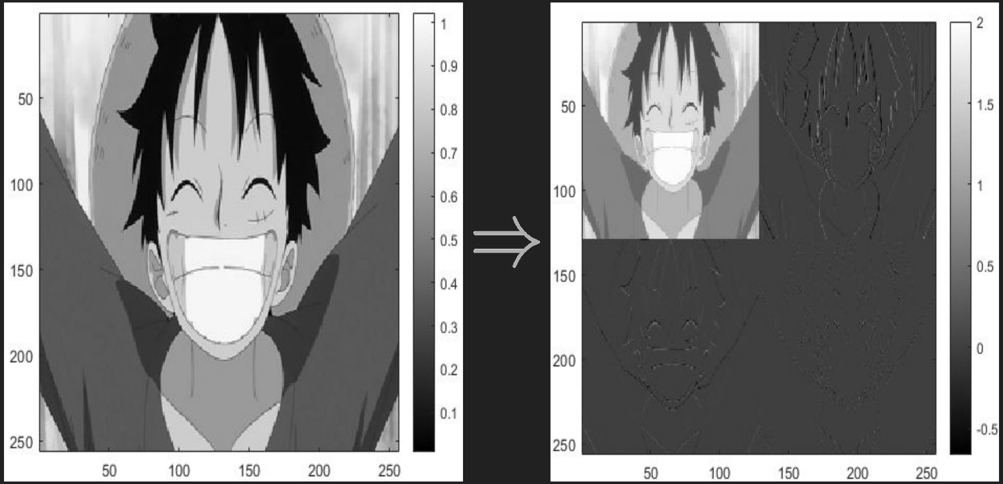


Sparse Image Reconstruction with Wavelets

- Promotion of sparsity in image reconstruction through a different basis in wavelet domain[8]

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

- The 2D Haar wavelet transform is computed by applying four 2D filters. And the resulting outputs are downsampled by a factor of 2
- It is important to note that the 2D Haar wavelet transform is an orthogonal transformation



Sparse Image Reconstruction with Wavelets

- If we let W denote the wavelet transform,the new cost function then becomes

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

- Since W is invertible, we can do a change of variables

$$\mathbf{c} = W\mathbf{x} \Leftrightarrow \mathbf{x} = W^{-1}\mathbf{c}$$

- Which gives the following cost function,

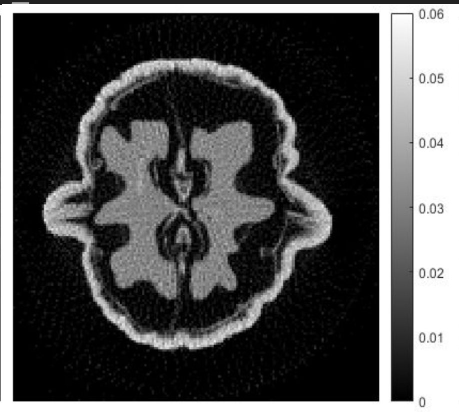
$$\min_{\mathbf{c}} \frac{1}{2}\|AW^{-1}\mathbf{c} - \mathbf{m}\|^2 + \lambda\|\mathbf{c}\|_1$$

that can be solved using the previous two approaches with $B = AW^{-1}$

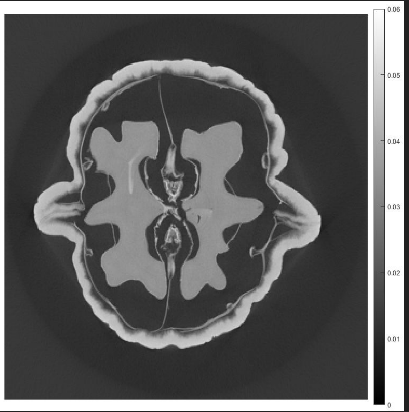
Results using L1-Norm with Wavelets:

PGD: 100 iterations
Run time:12.3549s
λ = 0.015

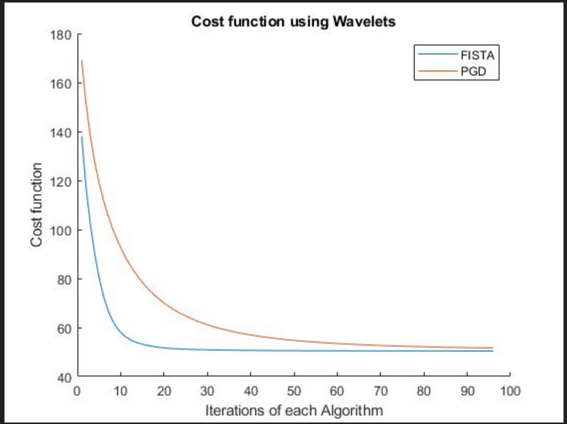
FISTA: 100 iterations
Run time:11.2107s
λ = 0.015



Ground Truth:



Cost vs Iteration:

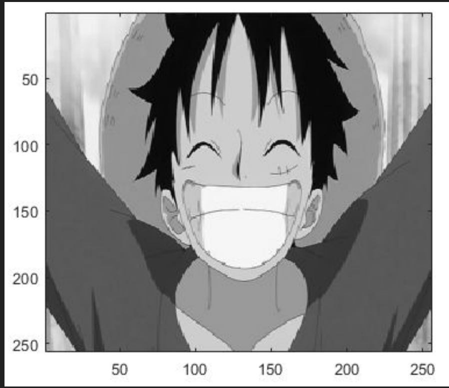


Total Variation Regularization

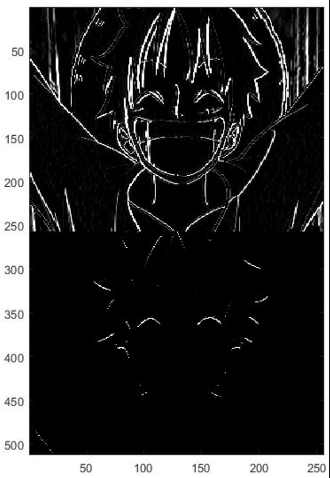
- Wavelets only use non-overlapping vertical, horizontal and diagonal differences. But maybe, all the differences should be used.
- If only vertical and horizontal differences are used, and we use all overlapping differences
- Sparsifying transform for images that approximately piecewise constant
- If we introduce K, the concatenation of finite difference matrices, into our cost function as the regularizer, we have

$$\mathbf{x} \Rightarrow K\mathbf{x}$$

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



$\Rightarrow$



- But this time, K is not invertible, since K is not a square matrix
- Hence, a different algorithm is needed

Chambolle Pock Primal Dual Algorithm

- The Chambolle Pock is an algorithm designed to solve primal-dual problems. Meaning it solves a primal problem, along with its dual formulation
- We can start by formulating our primal minimization problem

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

- Introducing the variable splitting $\mathbf{y} = B\mathbf{x}$, and taking the convex dual in the $\mathbf{y}$ variable yields the equivalent saddle point formulation

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

- We can write the cost function(primal problem) as

$$min_{\mathbf{x}\in\mathcal{R}^n} F(B\mathbf{x}) + G(\mathbf{x}), \text{ where } F([\mathbf{y};\mathbf{z}]) = \frac{1}{2}\|\mathbf{y}-\mathbf{m}\|^2 + \lambda\|\mathbf{z}\|_1, \text{ and } B = \begin{bmatrix} A \\ K \end{bmatrix}, G(x) = 0$$

- 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:

Chambolle Pock Primal Dual Algorithm

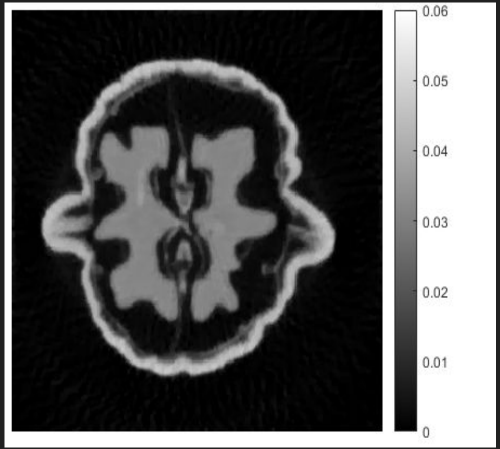
Algorithm 3 CP

$L \leftarrow \|B\|_2^2$, $\tau \leftarrow 1/L, \sigma \leftarrow 1/L, \theta \leftarrow 1$
for $k \leftarrow 1, 2, ...$ **do**
  $\mathbf{p}_{k+1} \leftarrow (\mathbf{p}_k + \sigma(A\mathbf{x} - \mathbf{m}))/(1+\sigma)$ ▷ Update for $\mathbf{p}$
  $\mathbf{y}_{k+1} \leftarrow \text{clip}(\mathbf{y}_k, \lambda)$ ▷ Projection onto ℓ^∞-norm ball
  $\mathbf{x}_k \leftarrow \mathbf{x}_k - \tau A^\top \mathbf{p}_{k+1} - \tau K^\top(\mathbf{y}_{k+1})$ ▷ Update for $\mathbf{x}$
  $\overline{\mathbf{x}}_{k+1} \leftarrow \mathbf{x}_{k+1} + \theta(\mathbf{x}_{k+1} - \overline{\mathbf{x}}_k)$ ▷ Update momentum term
end for

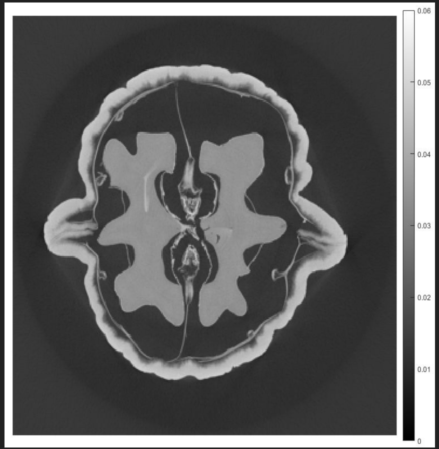
- For simplicity, we dont show the generalized algorithm, and a detailed derivation is given in [5]
- Here the clip function is defined as $\text{clip}(\mathbf{y}_k, \lambda) = argmin_{\|\mathbf{y}\|_\infty \leq \lambda} \|\mathbf{y} - \mathbf{z}\|_2^2$, which clips the values that are larger than a specific threshold value.

Results TV

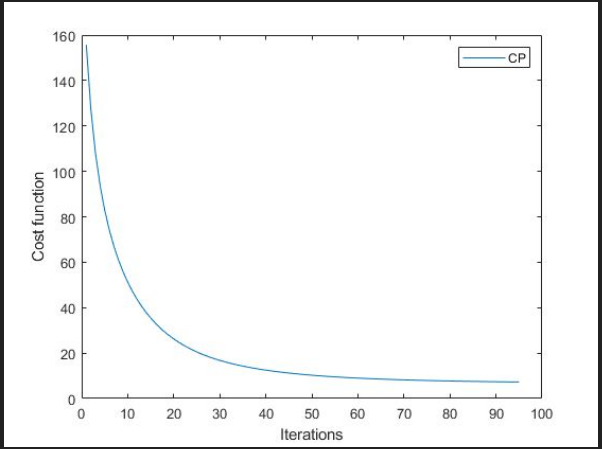
CP: 135 iterations
Run time:43.23029s
$\lambda = 0.01$



Ground Truth:



Cost vs Iteration:



Deep Learning: Letting the Machine Learn the Regularizer

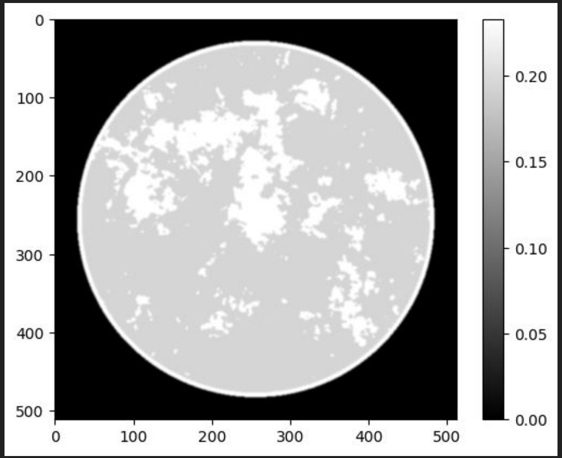
- Before:
  - Manually define the regularization term $\lambda R(x)$
  - Indulces a denoising step in an iterative algorithm (e.g. PGD)
    - In PGD, we had the following iterates

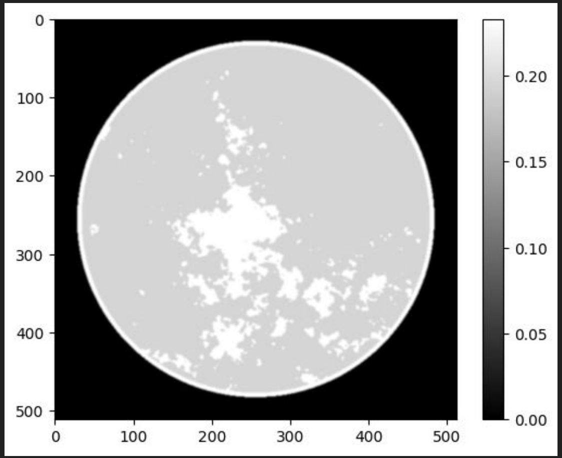
$$\mathbf{z}_{k+1} \leftarrow \mathbf{x}_k - \tau \nabla f(\mathbf{x}_k)$$
$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} g(\mathbf{x}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{z}_{k+1}\|^2$$

    - And we found the proximal step $\sigma(s) = max(|s| - \mu, 0) * sign(s)$
    - Which can be interpreted as a denoising step shrinking or removing small coefficients associated with noise
- Now:
  - Train a Neural Network as a denoiser
  - Prior (training dataset) that enforces certain characteristics
  - Use it as a Plug-and-Play in optimization frameworks [6][7]

Training Data

- Collection of 1000 breast CT phantoms
- Used in the AAPM low-dose breast CT reconstruction challenge [10]
- Training data examples

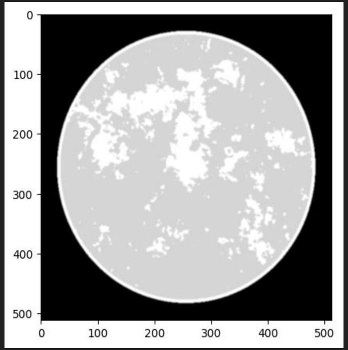




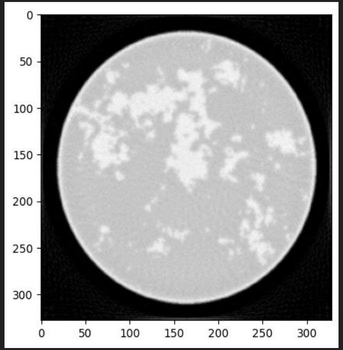
Preprocessing

- Since the dataset was different than walnut data, we need to make some changes to the training data to match the task it will perform
    - Reshape it to match in dimensions, phantoms are 512x512, we shaped to be 328x328
    - Simulate sparse view measurements
    - Add noise
    - Generate initial reconstruction through 10 iterations of least squares solver

Ground truth, x:

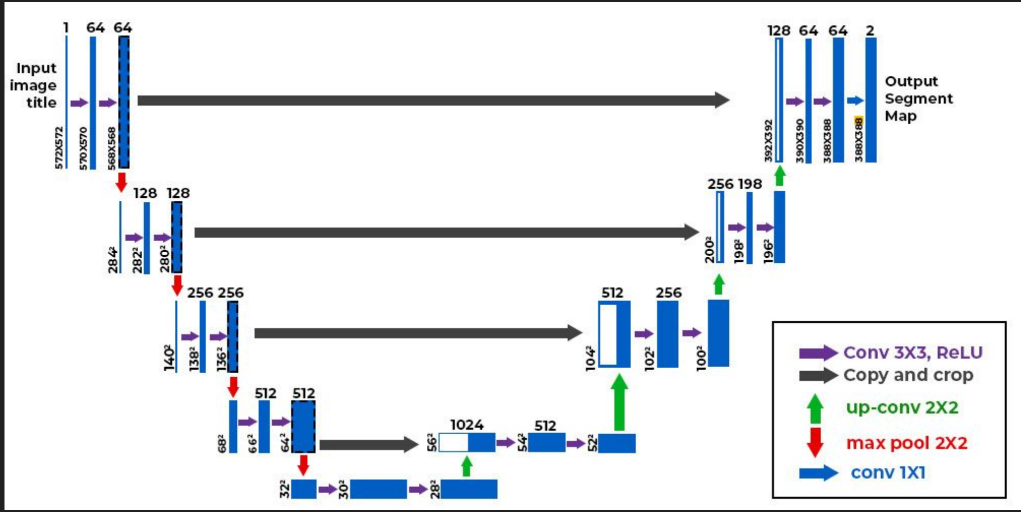


Initial Reconstruction (noisy/blurry), y:



U-Net Architecture

- U-Net is a Convolutional Neural Network architecture designed specifically for image-to-image tasks.[9]
- It has an encoder-decoder structure, where the encoder is a path that performs downsampling, and the decoder is a path that performs upsampling.



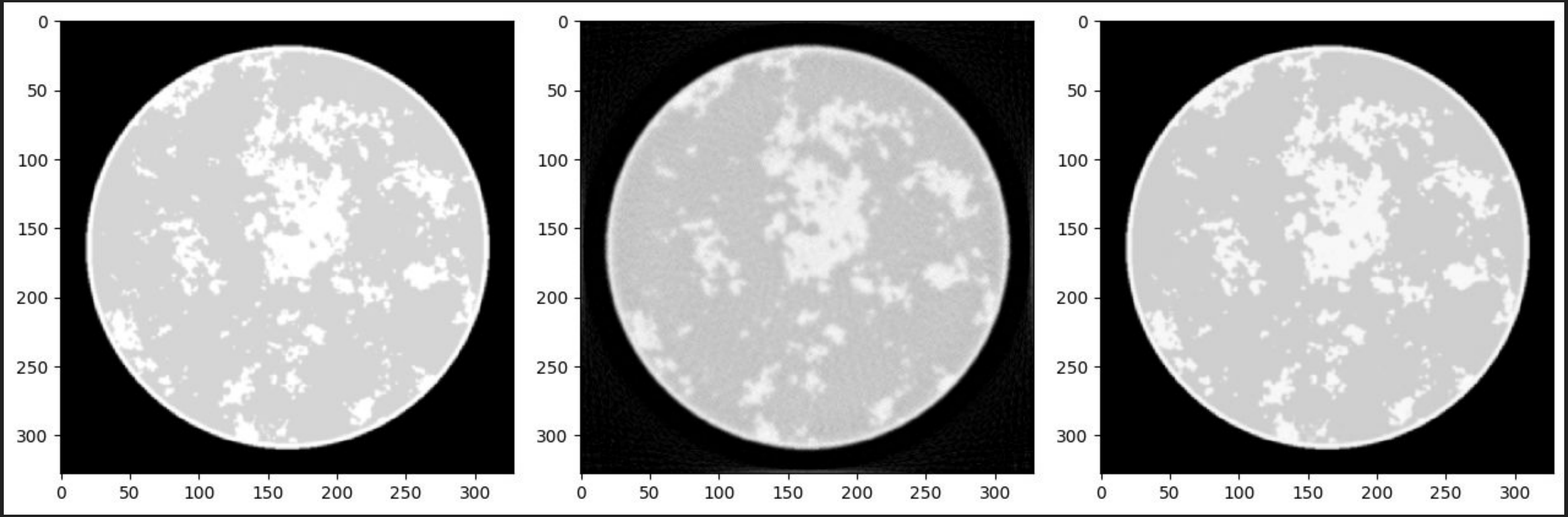
Training the CNN

- Split data 80% for training and 20% for testing
- Train
  - Adam optimizer
  - Learning rate: 0.0001
  - Training epochs: 50
  - Batch size: 1
  - Training cost function: MSE $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|U_\theta(y_i) - x_i\|^2$, U_θ is the U-net, where θ are its trainable parameters, y_i is the initial recon (with noise and blur), and x_i is the ground truth phantom
- Then, we can test our trained denoisers in the testing data set
- The MSE for the test set was 4.765610e-04

Performance of U-Net

- We can then look at the results of the U-net in a single image from the test data

Ground truth phantom: Initial Recon: U-net Output:



Using the learned denoiser in a previously seen algorithm

- With the learned denoiser we then adapt the PGD algorithm

Algorithm 4 Plug and Play using Proximal Gradient Descent

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

- U_θ is the result from applying the learned denoiser to $\mathbf{x}_k$
- We have two Hyperparameters, α (denoising strength) and K (iterations)
  - We will tune α , and set K to 10, for computational efficiency
- Hyper parameter Selection
  - Select α that gives the lowest root mean squared error on a single test image, from the test dataset
  - Expensive and takes a lot of time, but can be done, since only 1 hyper parameter
  - This process yielded α = 0.02

Results: Plug-and-Play

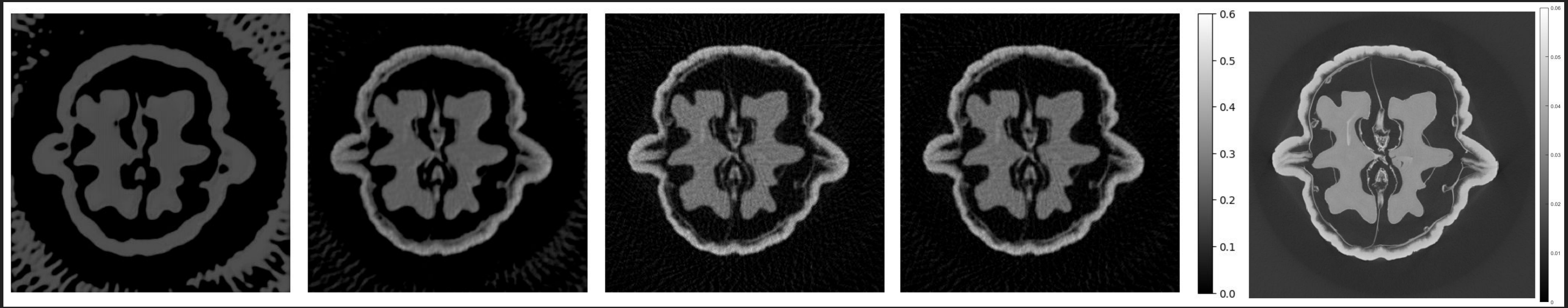
α: 0.5
iterations: 10

α: 0.1
iterations: 10

α: 0.05
iterations: 10

α: 0.02
iterations: 10

Ground Truth:



Results: Plug-and-Play

- Benefits
  - Easier than manually designing filters and then designing a custom solver
  - Flexibility, adapt dynamically to different data sets
  - Requires only 10 iterations (after pre-processing step)
  - Under some assumptions, convergence is guaranteed
- Limitations
  - Proper training data and network design are critical
  - The denoiser is treated as a black box, and not interpretable
  - Quality of the results are highly dependent on the correct set of hyper parameters

Conclusion

- Explored model-based approaches for CT image reconstruction, ranging from iterative optimization to deep learning methods
- Traditional proximal gradient descent and FISTA with L1 norm/wavelet regularizers effectively recovered structural details
- Total variation regularization with Chambolle-Pock primal-dual algorithm better preserved sharp edges and discontinuities, but increased computation time
- Integrated deep learning denoisers as data-driven regularizers into optimization algorithms like plug-and-play PGD
- Deep learning approach is a promising technique, with proper tuning, demonstrated good results when attempting to recover images in CT scans
- Despite training on synthetic data, saw good generalization to the walnut dataset

Thank you!
Questions?

References

[1] T. G. Feeman, “The mathematics of medical imaging,” Springer,, 2010.

[2] 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.

[3] N. Parikh, S. Boyd, et al., “Proximal algorithms,” Foundations and trends® in Optimization, vol. 1, no. 3, pp. 127–239, 2014.

[4] 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.

References

[5] 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.

[6] 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.

[7] K. Wei, A. Aviles-Rivero, J. Liang, Y. Fu, H. Huang, and C.-B. Sch ̈onlieb, “Tfpnp: 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.

References

[8] J. Gallier and J. Quaintance, “Linear algebra for computer vision, robotics, and machine learning,” University of Pennsylvania, 2019.

[9] 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. 234–241.30

[10] 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.