

1 Methods

As we know from the physics of computed tomography, the measured sinogram image is convolved with several sources of noise, notably electronic noise and structural noise[?]. Hence, using an iterative method that handles noise is necessary. In formulating the CT reconstruction as an optimization problem, one of the important considerations is to find a regularization method for the ill-posed problem that preserves the edges in the true image, since identifying boundaries of objects (such as the boundary between normal and abnormal tissues) has high diagnostic value.

We decided to follow the fast gradient-based algorithm proposed by Beck and Teboulle[?], and the following section will discuss the specific implementation of their method for CT image reconstruction.

Specifically, we chose isotropic TV regularization¹. The original publication of FISTA with TV[?] describes two types of TV regularization, anisotropic and isotropic TV, and the specific python package of choice, pyunloc[?], implements isotropic TV functionals. We decided to use the implemented TV function in the package and conducted some research on the differences of those two types of TV regularization functions. As being described in [?], anisotropic TV is known to favor horizontal and vertical structures for 2D images, and isotropic TV is shown to be a superior choice for regularization for a 2D image that preserves the off-axis edges.

The optimization problem can be formulated as:

$$\min_x \{ \|A(x) - d\|^2 + 2\lambda \text{TV}(x) \} \quad (1)$$

where $x \in \mathbb{R}^{m \times n}$ is the true image with the size $\sqrt{m \times n}$. $d \in \mathbb{R}^{p \times q}$ is the sinogram image where $p : |\text{rays}| \times q : |\text{views}|$. $A \in \mathbb{R}^{m \times n}$ is the forward operator that transforms an image to a sinogram image. λ is the regularization parameter. $\text{TV}(x) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ is an isotropic total variation functional, where

$$x \in \mathbb{R}^{m \times n}, \text{TV}_I(x) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2} \quad (2)$$

$$+ \sum_{i=1}^{m-1} |x_{i,n} - x_{i+1,n}| + \sum_{j=1}^{n-1} |x_{m,j} - x_{m,j+1}| \quad (3)$$

¹In our presentation, we claimed that we were using the anisotropic TV because we falsely assumed that we were optimizing an 1D vector rather than a 2D image. For the 1D case, isotropic and anisotropic TV are essentially the same. The error was resulted from the fact that the majority of the group members are not in the field of imaging science as are unfamiliar with how matlab implicitly handles an image. And the error of assuming a 1D vector resulted in thinking we can use anisotropic TV as well as creating vertical artifacts in reconstructed images, we have corrected the errors in the implementation and rewritten this section. This has been a good learning opportunity for us both in terms of learning new knowledge as well as learning the process of gaining such knowledge from outside of the field. We also attempted to rewrite the method section to increase clarity, we would appreciate any feedback on this version of the description of the method.

To solve the problem in eq.1, it requires a method that solves non-differentiable optimization problems. We decided to use fast iterative shrinkage/thresholding algorithm(FISTA) as the algorithm to solve the optimization problem formulated for this project?. FISTA has two main advantages: first, it has a global convergence rate of $\mathcal{O}(\frac{1}{k^2})$ (The proof of convergence rate is detailed in ?); second, FISTA is suitable for dealing with a non-smooth regularization function.

The general principle of the algorithm will be discussed in this paragraph and the pseudocode for the specific implementation will follow.

We discuss now how FISTA solved the optimization problem in eq.1. Let $f(x)$ denote the data fidelity term, it is clear that $f(x)$ is smooth and differentiable. The simple iterative method for solving the minimization of $f(x)$ is the gradient descent method:

$$x_k = x_{k-1} - t_k \nabla f(x_{k-1}) \quad (4)$$

As stated in ?, the optimization problem in eq.1 can be considered as a proximal regulation problem of $f(x_k)$ at x_k . Skipping the derivation, we have:

$$x_k = \underset{x}{\operatorname{argmin}} \left\{ \frac{1}{2t_k} \|x - (x_{k-1} - t_k \nabla f(x_{k-1}))\|^2 + \lambda \|x\|_{tv} \right\} \quad (5)$$

Specifically, for FISTA, each update is:

$$x_k = p_L(y_k) \quad (6)$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \quad (7)$$

$$y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}) \quad (8)$$

where $p_L = \underset{x \in \mathbb{E}}{\operatorname{argmin}} \left\{ \frac{L}{2} \|x - (y - \frac{1}{L} \nabla f(y))\|^2 + \|x\|_{tv} \right\}$

Compare to ISTA, FISTA uses the information from past 2 steps instead of 1 step to decide the new start x_k , This smarter choice of x_k allows the proximal gradient approaches the minimum faster.

After establishing FISTA is a fast algorithm for solving the optimization problem in eq., we discuss the dual approach to solve for isotropic total variation. The optimal solution for the inner problem is:

$$x_k = P_C(d - \lambda \mathcal{L}(p_k, q_k)) \quad (9)$$

where linear operator $\mathcal{L} : \mathbb{R}^{((m-1) \times n)} \mathbb{R}^{(m \times (n-1))} \rightarrow \mathbb{R}^{m \times n}$ is defined by

$$\mathcal{L}(\mathbf{p}, \mathbf{q})_{i,j} = p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1} \quad (10)$$

$$i = 1, \dots, m, j = 1, \dots, n \quad (11)$$

and P_C is an orthogonal projection operator onto a feasible set to ensure smoothness. To apply the dual approach, we need to use the mapping from $(p, q) \rightarrow (r, s)$ to ensure the smoothness.

We found an existing package in python called `pyunlocbox` ? to implement the algorithm. This package has all the required components for constructing the functions for data fidelity and regularization terms, the pseudocode is in algorithm1.

Algorithm 1 TV-FISTA

Input $d, A, \lambda, \text{maxiter}, \text{tol}$

$(r_0, s_0) = (p_0, q_0) = (0_{(m-1) \times n}, 0_{m \times (n-1)}), t_1 = 1$

if $k \leq \text{maxiter}$ **then**

Forward

Compute $x_f = \frac{2}{\lambda} A^T (A(x_{k-1}) - d)$

Backward

$x_k = x_f - \frac{1}{\lambda} \nabla \cdot (r_k, s_k)$

$\text{obj}_k = \frac{1}{2} \|x_f - x_k\|^2 + \lambda \|\nabla x_k\|^2$

$\text{tol}_k = \frac{|\text{obj}_{k-1} - \text{obj}_k|}{\text{obj}_k}$

Update projector

$r_k = r_{k-1} - \frac{1}{8\lambda} \frac{\nabla x_k}{\partial p_k}$

$s_k = r_{k-1} - \frac{1}{8\lambda} \frac{\nabla x_k}{\partial q_k}$

FISTA

$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$

$p_k = \frac{r_k}{\max\{1, \sqrt{r_k^2 + s_k^2}\}}$

$q_k = \frac{s_k}{\max\{1, \sqrt{r_k^2 + s_k^2}\}}$

if $\text{tol}_k \leq \text{tol}$ **then**

Stop

$k = k + 1$

Now we discuss other implementation choices in this algorithm. First, we decided to use relative tolerance as our stopping criterion since the stability of the proximal gradient can also be used in a similar way as gradient in a first-order optimization problem. Second, we used L-curve criterion to decide the regularization parameter, because it gives us a principled way to decide the proper amount of regularization without relying on the reference image.