ACO:Paper review Report

Aditya Banerjee

April 23, 2024

1 Introduction

The paper "A FISTA for linear inverse problem" [BT09] promises a faster algorithm that can solve a linear inverse problem and sits on top of the existing algorithm ISTA as a faster one.

A basic linear inverse problem can be stated in the form

$$b = Ax + w$$

Here A can be any linear operator in our case it is the blurring filter that is applied in the Image domain, w is additive white Gaussian noise with a known variance and x is the pristine Image.

The aim of the project is to understand the basics of how to apply the theory we have studied in Advanced Convex optimization to solve a ground level image restoration task.

Let us firstly define Image restoration:

1.1 Background

The inverse problem in question are what are termed as ill conditioned problems meaning they can't be solved as a $Ax = b$ problem this is because the least square terms result in huge norms and a failed restoration task. Thus the natural solution is to regularize the optimization problem using ℓ^2 or ℓ^1 norm. In this paper authors deal with the ℓ^1 norm. The optimization problem becomes

$$\min_x \ \|Ax - b\|_2^2 + \lambda\|x\|_1 \tag{1}$$

ℓ^1 have have seen a lot of attention in the signal processing community due to property that it induces sparsity. The notion of sparsity are actually very nice properties that are seen in a lot of natural settings certainly in a transformed domain. As an example we look at images. Images lie in $\mathbf{R}^{nxn}$ space, it can statistically seen that 90% of an image area is smooth i.e. only 10% of the image area contain edges. This it must be the case that this signal comprises most of the information in only 10% of its image area and thus we have a notion of sparsity in the gradients of the images. Figure(17b) shows the distribution of an the coefficients after performing high pass filtering on it. ℓ^1 regularization in probabilistic viewpoint corresponds to the performing the MAP estimate with a laplacian prior.

Thus we understand that the optimization is to be performed on a transformed domain to take advantage of this property. The go-to transform domain for this purpose is the wavelet transform. The wavelets have been developed over the years and is seen to be a tool adopted in many fields. Unlike Fourier transform, wavelet transform have the property that it can capture changes in the signal in all scales very effectively. In Fourier the coefficients vary with frequency but in wavelets it varies with localization and with scale. Which gives very unique and versatile features about the signal. These are used in almost every field in which any signal is being processed, Image Processing, Data Compression, Biomedical Engineering, Computer Vision, Financial Analysis, Communications, Machine Learning, Geophysics, Environmental Sciences etc.

Thus we perform the optimization problem in the wavelet domain, thus we write the optimization formulation as such

$$\min_x \ \|Ax - b\|_2^2 + \lambda\|Wx\|_1$$

where W is the transformation matrix, the author consider the optimisation problem to be the same as 12 with $A = R.W$, where R is a blurring kernel and W takes a set wavelet coordinates to the image

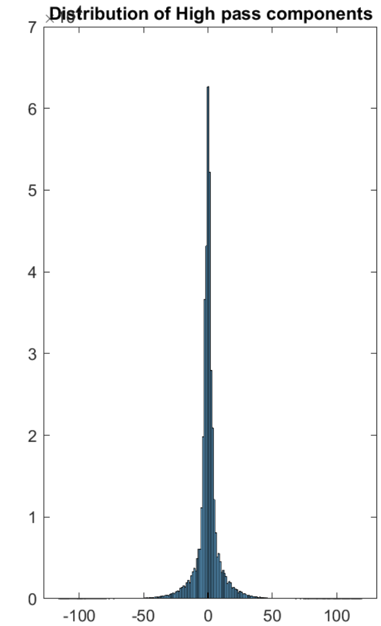


Figure 1: distribution of High pass filtering of natural images

domain. So moving forward we stick to the convention of that the authors follow. Thus we get x in the wavelet domain.

In [MDM04] *"An Iterative Thresholding algorithm for linear inverse problems with a sparsity constraint"* the authors derive the solution for the optimization problem with a lot of mathematical rigour and shows how the linear inverse problems naturally lead to an iterative algorithm. In this review paper I show the derivation of that they give in brief and then show how it is the exact same solution we get in the proximal gradient descent setting.

So the author look a the problem in a more abstract setting,

$$\Phi_{w,p}(f) = \frac{1}{2}\|Kf - g\|^2 + \sum_{\gamma\in\Gamma} w_\gamma |\langle f, \varphi_\gamma\rangle|^p$$

$$\Phi_{w,p}(f) = \frac{1}{2}\|\sum_{\gamma\in\Gamma} f_\gamma . K\varphi_\gamma - g\|^2 + \sum_{\gamma\in\Gamma} w_\gamma |f_\gamma|^p \tag{2}$$

where K is the blurring kernel g is the blurry image, φ_γ are the wavelet basis vectors, w_γ are just weight if kept constants, become the regularization constant and $f_\gamma := \langle f, \varphi_\gamma\rangle$ are refered to as the wavelet coefficients. To solve to this problem we take partial derivatives of $\Phi_{w,p}(f)$ w.r.t. the wavelet coefficients $f_\gamma \in \mathbf{R}$ and simply substitute it to zero (The question about differentiability is discussed later).

$$\Rightarrow \frac{\partial \Phi_{w,p}(f)}{\partial f_{\gamma_i}} = \langle \sum_{\gamma\in\Gamma} f_\gamma . K^* K\varphi_\gamma - g, \varphi_{\gamma_i}\rangle + w_{\gamma_i} sign(f_{\gamma_i}).|f_{\gamma_i}|^{p-1} = 0$$

The difficulty in solving this variational equation arrives in decoupling k and φ_γ, it is easier to solve the problem if k is diagonal w.r.t. the wavelet basis, which they show in their paper [MDM04] for different values of p, and taking K to be diagonal w.r.t. the basis or the identity operator. To solve this problem and remove the K^*Kf term all together a surrogate problem was created which could be solved without any such issues

$$\Phi^{sur}_{w,p}(f; a) = \Phi_{w,p}(f) + C\|f - a\|^2 - \|Kf - Ka\|^2 \tag{3}$$

C is chosen to taken such that $C\mathcal{I} - K^*K$ is a strictly positive operator, to preserve convexity of the optimization problem. Derivative of 3 w.r.t. the wavelet coefficients f_γ, we must question the differentiability of the nonlinear expression. Authors discuss it in brief, for $p > 1$ $\Phi_{w,p}(f)$ is differentiable w.r.t f_γ everywhere, for p=1, 2 is differentiable in f_γ except at $f_\gamma = 0$. and the minimization leads to solving the variational problem The minimization of $\Phi^{sur}_{w,p}(f)$ with $C = 1$ is,

$$f_\gamma = S_{w_\gamma,1}(a_\gamma + [K * (g - Ka)]_\gamma) \tag{4}$$

$$S_{w_\gamma,1}(X) = \begin{cases} x - w/2 & \text{if } x \geq w/2 \\ 0 & \text{if } |x| < w/2 \\ x + w/2 & \text{if } x \leq -w/2 \end{cases} \tag{5}$$

Author gives the derivation for $p = 1$, and gives a proposition which proves the optimalilty of 4 for $p = 1$. Following is the proposition

Proposition 2.1 *Suppose the operator $K : \mathcal{H} \to \mathcal{H}'$ with $\|K^*K\| < 1$ and suppose $g \in \mathcal{H}'$. let $(\varphi_\gamma)_{\gamma\in\Gamma}$ be orthogonal basis for $\mathcal{H}$, and w_γ be a sequence of strictly positive numbers. Pick arbitrary $p \geq 1$ and $a \in \mathcal{H}$. Define the functional on $\mathcal{H}$ by 3*

Then $\Phi^{sur}_{w,p}(f; a)$ has a unique minimizer in $\mathcal{H}$.

The miinimizer is given by $f = \mathcal{S}_{w,1}(a + K^*(g - Ka))$, where the operator $\mathcal{S}_{w,p}$ are defined by

$$\mathcal{S}_{w,p}(h) = \sum_\gamma S_{w_\gamma,p}(h_\gamma)\varphi_\gamma \quad ,$$

with the functions $\mathcal{S}_{w,p}$ from $\mathbf{R}$ to itself given by 5. For all $h \in \mathcal{H}$ one has

$$\Phi^{sur}_{w,p}(f + h; a) \geq \Phi^{sur}_{w,p}(f; a) + \|h\|^2.$$

The authors give an iterative approach which gives the ISTA algorithm. Which is given as such: *Pick $f^0 \in \mathcal{H}$, and define the functions f^n recursively by the algorithm*

$$f^n = \mathcal{S}_{w,1}(f^{n-1} + [k * (g - kf^{n-1})]_\gamma)$$

Coming back to the FISTA paper we have a more convinient and more familiar vector space. The update refereed in the FISTA paper is given by

$$x_{k+1} = \mathcal{T}_{\lambda t}(x_k - 2tA^T(Ax_k - b)) \tag{6}$$

Beck initially refers to the paper [MDM04] and point to (6) as our iterative steps but afterward connects this to the Proximal Gradient formulation, but it must be proven that both are indeed the same. We shall do that next

$$\textsc{Prox}_g(x) := argmin_z \frac{1}{2}\|z - x\|^2 + g(z) \tag{7}$$

Claim: *This is similar to what is found using the proximal operator formulation of the optimization.*

$$x_{k+1} = \textsc{Prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x))$$

more precisely $\textsc{Prox}_{\beta^{-1}g}(\circ) = \mathcal{T}_{\lambda\beta^{-1}}(\circ)$

before proving the claim, we should take time to state all the notations and assumtions of the problem formally.

For the sake of clarity and ease of reference I will define following convention,

$$F = f + g$$

where

$$f(x) = \frac{1}{2}\|Ax + b\|_2^2$$

is smooth convex function and

$$g(x) = \lambda\|x\|_1$$

is convex a non-smooth function. We say that f is continuously differentiable with lipschiz continuous gradient $L(f)$ i.e.:

$$\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|$$

This mean in our case where A is a known filter and $f(x)$ is a Quadratic in x which is double differentiable. Hence we can say that that $L(f)$ is the maximum eigen value of the Hessian.

$$L(f) = \|A^T A\|_2$$

Now come back to proving that our claim.

Proof of claim. Which was that $\textsc{Prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x)) = \textsc{Prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x))$.
In our problem statement we deal with finite dimensional space, thus WLOG following can be stated.

$$\begin{aligned}
\textsc{prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x)) &= argmin_{z\in\mathbf{R}^n}\frac{1}{2}\|z - (x_k - \beta^{-1}\nabla f(x))\|_2^2 + \beta^{-1}\lambda\|z\|_1 \\
&= argmin_{z\in\mathbf{R}^n}\sum_{i=1}^{n}\frac{1}{2}(z_i - (x_k - \beta^{-1}\nabla f(x))_i)^2 + \beta^{-1}\lambda|z_i|
\end{aligned}$$

these functions are non-negative functions in $\mathbf{R}^n$ thus we can decouple optimization problem to component wise minimization.

$$\begin{aligned}
\textsc{Prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x)) &= \prod_{i=1}^{n} argmin_{z_i\in\mathbf{R}}\frac{1}{2}(z_i - (x_k - \beta^{-1}\nabla f(x))_i)^2 + \beta^{-1}\lambda|z_i| \\
&= \prod_{i=1}^{n}\textsc{Prox}_{\beta^{-1}g_i}((x_k - \beta^{-1}\nabla f(x))_i)
\end{aligned}$$

Now the problem can be solved by simply turning it into a 1 dimensional optimization problem, to make the calculations easier we find $\textsc{Prox}_{\alpha g}(x)$ where $g(x) = |x|$ where $g : \mathbf{R} \to \mathbf{R}$

$$\begin{aligned}
\textsc{Prox}_{\alpha g}(x) &= argmin_{z\in\mathbf{R}}\frac{1}{2}(z - x)^2 + \alpha|z| \\
&= argmin_{z\in\mathbf{R}}\ p(z)
\end{aligned}$$

$$p_x(z) := \begin{cases} \frac{1}{2}(z - x)^2 + \alpha z & \text{if } z \geq 0 \\ \frac{1}{2}(z - x)^2 z - \alpha z & \text{if } z < 0 \end{cases}$$

if $z < 0$ the we get $p_x^{'}(z) = z - x + \alpha$ which gives us $x < \alpha$ and $\textsc{Prox}_{\alpha g}(x) = x - \alpha$. With same arguments if $z > 0$ we get $\textsc{Prox}_{\alpha g}(x) = x + \alpha$ for $x > \alpha$. Thus the most natural value for the proximal operator for $|x| < \alpha$ is 0 but we know the function is non differentiable at $z = 0$ thus an argument for optimality of $\textsc{Prox}_{\alpha g}(x) = 0$ must be given. We look at $p_\alpha(0) - p_\alpha(0 + h)$ for $h > 0$

$$p_\alpha(0 + h) - p_\alpha(0) = 0.5(0 + h - \alpha)^2 + \alpha|0 + h| - \big(0.5(0 - \alpha)^2 + \alpha|0|\big)$$

using $h > 0$,

$$\begin{aligned}
&\Rightarrow p_\alpha(0 + h) - p_\alpha(0) = h^2 \\
&\Rightarrow p_\alpha(0 + h) \geq p_\alpha(0)
\end{aligned}$$

similarly it can be shown $p_\alpha(0 - h) \geq p_\alpha(0)$, and for also that $p_{-\alpha}(z)$ is minimal at $z = 0$. Therefore this gives us the proximal operator.

$$\Rightarrow \textsc{Prox}_{\alpha g}(x) = \begin{cases} x - \alpha & \text{if } x > \alpha \\ x + \alpha & \text{if } x < -\alpha \\ 0 & \text{if } |x| \leq \alpha \end{cases}$$

$$\textsc{Prox}_{\alpha g}(x) = \mathcal{T}_\alpha(x),\ \forall x \in \mathbf{R}$$

where $\mathcal{T}_\alpha$ is the shrinkage operator. Thus we get a component wise shrinkage of the vector. Recall that $\alpha = \lambda\beta^{-1}$ and $x = x_k - \beta^{-1}\nabla f(x)$,

$$\begin{aligned}
&\Rightarrow \textsc{Prox}_{\beta^{-1}g}(x_k - \beta^{-1}\nabla f(x)) = \mathcal{T}_{\lambda\beta^{-1}}(x_k - \beta^{-1}\nabla f(x)) \\
&\Rightarrow \textsc{Prox}_{\beta^{-1}g}(\circ) = \mathcal{T}_{\lambda\beta^{-1}}(\circ)
\end{aligned}$$

$\square$

$$x_k := \text{argmin}_{x\in\mathbf{R}^n}\ \frac{1}{2}\|x-(x_{k-1}-t_k\nabla f(x_{k-1}))\|_2^2+\lambda\|x\|_1$$

The authors give yet another key formulation of the Optimization problem by manipulating the proximal gradient decent definition. It is an important analysis which is key to understanding the said algorithm,

$$x_k = \text{argmin}_{x\in\mathbf{R}^n}\frac{1}{2}\|(x_{k-1}-t_k\nabla f(x_{k-1}))-x\|^2+\lambda\|x\|_1 \tag{8}$$

$$= \text{argmin}_{x\in\mathbf{R}^n}\{f(x_{k-1})+\langle x-x_{k-1},\nabla f(x_{k-t})\rangle+\frac{1}{2t_k}\|x-x_{k-1}\|^2+\lambda\|x\|_1\} \tag{9}$$

This shows that the proximal operator is actually solving the quadratic approximation of the function $F(x)$.

$$\rho_L(x) := argmin_y\ \{Q_L(y,x-t\nabla f(x)):\ y\in\mathbf{R}^n\}$$

$$Q_L(x,y) := f(y)+\langle x-y,\nabla f(y)\rangle+\frac{L}{2}\|x-y\|^2+g(x)$$

It is easy to see that we are solving the quadratic approximation of $F(x)$ at the new gradient step. It is a consequence of the L-smoothness of the function $f(x)$. Thus to summaries and track our steps, we first proved $x_{k+1}=\mathcal{T}_{\lambda t}(x_k-2tA^T(Ax_k-b))$ through the surrogate formulation of the optimization problem. Then it was shown that indeed as the solution we are actually solving the PROX operator,i.e. $\Rightarrow \text{PROX}_{\beta^{-1}g}(\circ)=\mathcal{T}_{\lambda\beta^{-1}}(\circ)$,and finally that

$$\rho_L(x_{k-1}) := argmin_x\{Q_L(x,x_{k-1}-t_k\nabla f(x_{k-1})):x\in\mathbf{R}^n\}$$

$$x_k = p_{\bar{L}}(x_{k-1})$$

is our update.

In the subsequent section we try to understand how formulation of the ISTA and FISTA algorithm was done through the same lens.

2 Algorithm formulation and key results

2.1 ISTA

We give a basic background on the ISTA algorithm. ISTA was introduced by Thomas Blumensath and Mike E. Davies in their paper titled "Iterative Thresholding for Sparse Approximations" published in the Journal of Fourier Analysis and Applications in 2008. This paper presented ISTA as an iterative optimization algorithm for solving sparse approximation problems, particularly in the context of compressed sensing and sparse signal recovery Given a signal x and a measurement matrix A, the ISTA algorithm aims to solve the optimization problem $F(x)$:

$$\min_x \frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_1$$

where b is the measured signal (corrupted by noise), and λ is a regularization parameter controlling the sparsity of the solution.

The ISTA algorithm iteratively updates the estimate of x by performing the following steps:

ISTA with constant step size: **Input** Take $L := L(f)$ the lipschitz constant of ∇f
Step 0. $x_0\in\mathcal{R}^n$. *Set* $y_1 = x_(0)$, $t_1 = 1$
Step k. (k $\geq$ 1). Compute

$$x_k = p_{\overline{L}}(x_{k-1})$$

ISTA with backtracking:
Step 0. Take $L_0 > 0$, *some* $\eta > 1$, *and* $x_0 \in \mathcal{R}^n$
Step k. (k $\geq$ 1). Find the smallest non-negative integer i_k *such that with* $\overline{L}=\eta^{i_k}.L_{k-1}$

$$F(p_{\overline{L}}(x_{k-1})) \leq Q_{\overline{L}}(p_{\overline{L}}(x_{k-1}),x_{k-1})$$

set $\overline{L}=\eta^{i_k}.L_{k-1}$ and compute

$$x_k = p_{\overline{L}}(x_{k-1})$$

The ISTA algorithm is known for its simplicity and effectiveness in solving problems with sparsity-promoting regularization. It is widely used in various applications, including image processing.

In the algorithm it can be shown that the sequence of $\{x_k\}$ in fact give a non-decreasing objective function value. For this they refer to LEMMA 2.3 and some simple algebra.

Lemma 2.3. *Let $y \in \mathbf{R}^n$ and L be such that*

$$F(\rho_L(y)) \leq Q(\rho_L(y), y).$$

then for any $x \in \mathbf{R}^n$,

$$F(x) - F(\rho_L(y)) \geq \frac{L}{2}\|\rho_L(y) - y\|^2 + L\langle y - x, \rho_L(y) - y\rangle$$

For ISTA they state that the sequence has the following order.

$$F(x_k) \overset{1}{\leq} Q_{L_k}(x_k, x_{k-1}) \overset{2}{\leq} Q_{L_k}(x_{k-1}, x_{k-1}) = F(x_{k-1}),$$

where $L_k \geq L(f)$ for backtracking case, note that this will change the result of the lemma. As it just relaxes an upper bound. They justify the first in equality using the lemma 2.3 and second inequality comes simply from the fact that $x_k = p_{\bar{L}}(x_{k-1})$, subsequently the theorem of convergence of the ISTA algorithm is give in,

Theorem 3.1. *let x_k be the sequence generated by the backtrack ISTA iterates. Then for any $k \geq 1$*

$$F(x_k) - F(x^*) \leq \frac{\alpha L(f)\|x_0 - x^*\|^2}{2k} \quad \forall x^* \in X_*$$

where $\alpha = 1$ for constant stepsize and $\alpha = \eta$ for backtrack stepsize.

For FISTA the inequality $F(x_k){\leq}F(x_{k-1})$ is in fact not true because of the additional momentum term. Next we look at the main result of the paper i.e The FISTA algorithm.

2.2 FISTA

FISTA (Fast Iterative Shrinkage-Thresholding Algorithm):

FISTA, introduced by Amir Beck and Marc Teboulle in 2009, is an accelerated version of the Iterative Shrinkage-Thresholding Algorithm (ISTA). FISTA is designed to solve convex optimization problems with a smooth component$f(x)$ and a non-smooth regularization term$g(x)$.

History in Optimization Theory:

FISTA builds upon the work of Yurii Nesterov, who introduced the concept of acceleration in convex optimization algorithms. Nesterov's Accelerated Gradient Descent (AGD) algorithm, proposed in the late 1980s, achieved significantly fast convergence rates compared to traditional first order gradient descent methods in particular, he showed $O(\frac{1}{k^2})$ convergence for first order gradient descent methods. He had worked on non-smooth composite functions and was also successful in doing so.

Amir Beck independent to the work of Y.E. Nesterov developed an $O(\frac{1}{k^2})$ convergence algorithm. Beck mentions in brief about the paper of Nesterov on $O(\frac{1}{k^2})$ convergence of composite functions which was publish 2 years before the FISTA paper. Nesterov's method as mention by beck in his paper uses combination of past functions $(\psi_k(\circ))$ to approximate the objective function $(F(\circ))$. Which is fundamentally very different from the proximal formulation of FISTA. He mentions due to the key difference in the building block of the algorithms the algorithm found by him is indeed novel. Theory of accelerated algorithms and proposed FISTA as a generalization of Nesterov's earlier results in the case of non-smooth composite problems for the additional term $g(x)$.

Comparison with ISTA:

Both ISTA and FISTA are iterative optimization algorithms used for solving convex optimization problems with sparsity-inducing regularization. However, FISTA typically converges faster than ISTA, especially for problems with Lipschitz continuous gradients.

While ISTA updates the estimate of the solution using a fixed or backtracking step size, FISTA additionally finds a special point based on the iterate found and the iterate of the previous step to find a new point. This is popularly known as the momentum term, it incorporates Nesterov's acceleration technique, leading to faster convergence rates compared to ISTA.

FISTA with constant step size: **Input:** Take $L := L(f)$ the lipschitz constant of ∇f
Step 0. $x_0 \in \mathbf{R}^n$. *Set* $y_1 = x_{(0)}$, $t_1 = 1$
Step k. (k $\geq$ 1). Compute

$$x_k = p_L(y_k)$$

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

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

FISTA with backtracking:
Step 0. Take $\mathrm{L}_0 > 0$, *some* $\eta > 1$, and $x_0 \in \mathbf{R}^n$. *Set* $y_1 = x_{(0)}$, $t_1 = 1$
Step k. (k $\geq$ 1). Find the smallest non-negative integer i_k *such that with* $\overline{L} = \eta^{i_k}.L_{k-1}$

$$F(p_{\overline{L}}(x_{k-1})) \leq Q_{\overline{L}}(p_{\overline{L}}(x_{k-1}), x_{k-1})$$

set $\overline{L} = \eta^{i_k}.L_{k-1}$ and compute

$$x_k = p_{\overline{L}}(y_k)$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \tag{10}$$

$$y_{k+1} = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1}) \tag{11}$$

Previously in ISTA $\rho_L(\circ)$ was applied to the x_k to get x_{k+1} but in FISTA it is applied to a special term y_k which is a special linear combination of the x_k and x_{k-1}. We will see that the relataion (10) and (11) are actually the outcome of the construction of proof of convergence itself. These additional terms were the *hyper-parameters* which they kept free during the derivation.

generally we try to create a telescopic sum when we try to build a bound for optimization algorithm. It is simply because it removes the time variable from the equation. Essentially only two terms remain after the telescopic sum. The initial state and the final state. The state can be can be the objective function as we have seen in the proof of linear convergence ISTA or any other measure of optimality we want to check

$$F_t - F_{t-1} \geq f(t)$$

$$\sum_{t=a}^{b} F(x_t) - F(x_{t-1}) = F(b) - F(a) \geq \sum_{t=a}^{b} f(t)$$

where F is the measure of optimality and $f(\circ)$ is a sequence of time The right hand can be a sequence of time as we have seen in ISTA it is the distance from x^* which was a monotone converging sequence. If it was not so in ISTA then we would have had to work harder for getting a bound.

The next lemma create recursive inequality that successfully gives such a telescopic sum through time,

LEMMA 4.1. *The sequences x_k, y_k generated via FISTA with either a constant or backtracking step-size rule satisfy for every $k \geq 1$*

$$\frac{2}{L_k} t_k^2 v_k - \frac{2}{L_{k+1}} t_{k+1}^2 v_{k+1} \geq \|\mathbf{u}_{k+1}\|^2 - \|\mathbf{u}_k\|^2$$

where $v_k := F(x_k) - F(x^*)$, $u_k := t_k x_k - (t_k - 1)x_{k-1} - x^*$.

Proof of lemma 4.1. we will go through the proof in a more intuitive manner and try to get an insight as to what the author might have thought coming up with such a scheme. We know that in the end result is that he creates a sequence of updates $x_k = \rho_{L_k}(y_k)$ and not $\rho_{L_k}(x_{k-1})$, thus we can imagine the these y_k are some free variable which much have some constraint. We can in fact assume a very nice constraint on y_k, in every iterate we store the previous iterate and make y_{k+1} a combination of x_k and x_{k-1}, more so that y_{k+1} must lie on the line joining x_k and x_{k-1}:

$$y_{k+1} = x_k + \alpha_{k+1}(x_{k-1} - x_k)$$

$$\alpha_{k+1} \in \mathbf{R}$$

we can flow with the proof given by the author and interpret the proof from this lens.

Firstly LEMMA 2.3 is used to get two inequalities, one with $(x = x_k,\ y = y_{k+1})$ and $(x = x_*,\ y = y_{k+1})$ both with $L = L_{k+1}$,

$$2L_{k+1}^{-1}(v_k - v_{k+1}) \geq \|x_{k+1} - y_{k+1}\|^2 + 2\langle x_{k+1} - y_{k+1}, y_{k+1} - x_k\rangle$$
$$-2L_{k+1}^{-1}(v_{k+1}) \geq \|x_{k+1} - y_{k+1}\|^2 + 2\langle x_{k+1} - y_{k+1}, y_{k+1} - x_*\rangle$$

in the previous set the fact is used that $x_{k+1} = \rho_{L_{k+1}}(y_{k+1})$

now we multiply the fist inequality by a $\gamma_{k+1} - 1$ and add it to the second, only thing we have to be careful about is that $\gamma_{k+1} \geq 1$ to respect the inequality,

$$2L_{k+1}^{-1}((\gamma_{k+1}-1)v_k - \gamma_{k+1}v_{k+1}) \geq \gamma_{k+1}\|x_{k+1} - y_{k+1}\|^2 + 2\langle x_{k+1} - y_{k+1}, \gamma_{k+1}y_{k+1} - (\gamma_{k+1} - 1)x_k - x^*\rangle$$

We multiply γ_{k+1} throughout the equation again for homogeneity's sake, hence we get an interesting inequality,

$$2L_{k+1}^{-1}(\gamma_{k+1}(\gamma_{k+1}-1)v_k - \gamma_{k+1}^2 v_{k+1}) \geq \|\gamma_{k+1}x_{k+1} - \gamma_{k+1}y_{k+1}\|^2 + 2\langle\gamma_{k+1}x_{k+1} - \gamma_{k+1}y_{k+1}, \gamma_{k+1}y_{k+1} - (\gamma_{k+1}-1)x_k - x^*\rangle$$

We reduce the right hand inequality using Pythagoras relation,

$$\|b - a\|^2 + 2\langle b - a, a - c\rangle = \|b - c\|^2 - \|a - c\|^2$$
$$a := \gamma_{k+1}y_{k+1}, \quad b := \gamma_{k+1}x_{k+1}, \quad c := (\gamma_{k+1} - 1)x_k - x^*$$
$$2L_{k+1}^{-1}(\gamma_{k+1}(\gamma_{k+1}-1)v_k - \gamma_{k+1}^2 v_{k+1}) \geq \|\gamma_{k+1}x_{k+1} - (\gamma_{k+1}-1)x_k - x^*\|^2 - \|\gamma_{k+1}y_{k+1} - (\gamma_{k+1}-1)x_k - x^*\|^2$$

Now at this we must realise that we have found the exact recursive inequality we are looking for,

to make it more clear we see that to make this work $\gamma_k = \gamma_{k+1}(\gamma_{k+1} - 1)$ which will give us the update mentioned in the algorithm (i.e:

$$\gamma_{k+1} = \frac{1 + \sqrt{1 + 4\gamma_k^2}}{2}$$

). On the right we have a term that is independent of the free variables thus we must fix it to,

$$u_{k+1} := \gamma_{k+1}x_{k+1} - (\gamma_{k+1} - 1)x_k - x^*$$

then to make $u_k = \gamma_{k+1}y_{k+1} - (\gamma_{k+1} - 1)x_k - x^*$ we have y_{k+1} as the free variable. Thus we get,

$$u_k = \gamma_{k+1}y_{k+1} - (\gamma_{k+1} - 1)x_k - x^*$$
$$\Rightarrow \gamma_k x_k - (\gamma_k - 1)x_{k-1} - x^* = \gamma_{k+1}y_{k+1} - (\gamma_{k+1} - 1)x_k - x^*$$
$$\Rightarrow \gamma_{k+1}y_{k+1} = \gamma_{k+1}x_k - (\gamma_k - 1)(x_{k-1} - x_k)$$
$$\Rightarrow \alpha_{k+1} = -\frac{(\gamma_k - 1)}{\gamma_{k+1}}$$

returning to the proof from this substituting all the values we get what is required,

$$\frac{2}{L_k}t_k^2 v_k - \frac{2}{L_{k+1}}t_{k+1}^2 v_{k+1} \geq \|\mathbf{u}_{k+1}\|^2 - \|\mathbf{u}_k\|^2$$

□

before the proof of the main theorem we return to our regular notation of using, $\gamma_{k+1} = t_{k+1}$

LEMMA 4.2. *Let x_k, y_k be a positive sequence of reals satisfying*

$$a_k - a_{k-1} \geq b_{k+1} - b_k, \quad \forall k \geq 1, with a_1 + b_1 \leq c, c > 0$$

then we have $a_k \leq c$ for every $k \geq 1$.

LEMMA 4.2. *The positive sequence $\{t_k\}$ generated in FISTA with $t_1 = 1$ satisfies $t_k \geq (k + 1)/2$ for every $k \geq 1$.*

Next we look at the proof of the final theorem that proves the convergence rate of $O(1/k^2)$.

Theorem 4.4. *Let x_k, y_k be generated by FISTA. Then for any $k \geq 1$*

$$F(x_k) - F(x^*) \leq \frac{2\alpha L(f)\|x_0 - x^*\|^2}{(k + 1)^2} \quad \forall x^* \in X_*$$

where $\alpha = 1$ for constant step size and $\alpha = \eta$ for backtracking stepsize setting.

Proof. the proof is quite intuitive and short after all the work that was done leading to it. We define the following,

$$a_k := \frac{2}{L_k} t_k^2 v_k, \quad b_k := \|u_k\|^2, \quad c := \|x_0 - x^*\|^2,$$

then from *lemma 4.1* we have that

$$a_k - a_{k+1} \geq b_{k+1} - b_k,$$

and hence assuming $a_1 + b_1 \leq 1$ holds true, invoking *lemma 4.2*, we obtain that

$$\frac{2}{L_k} t_k^2 v_k \leq \|x_0 - x^*\|^2,$$

combined with $t_k \geq (k+1)/2$ (by *lemma 4.3*) yeilds

$$v_k \leq \frac{2L_k\|x_0 - x^*\|^2}{(k+1)^2}$$

here it can be mentioned they have not used the scheme of telescopic sum but the essence of a telescopes sequence $a_k + b_k \leq c$ remains in the proof of the analysis. It is also clear that the $(k+1)^2$ term comes from the fact that $t_k \geq k+1, \quad \forall k \geq 1$. The question of whether k^2 can be changed to k^3 or greater is an open question for research.

now we only have to prove the validity of the relation $a_1 + b_1 \leq c$,

$$a_1 := \frac{2}{L_1} t_1^2 v_1 = \frac{2}{L_1} v_1, \quad b_1 := \|u_1\|^2 = \|x_1 - x^*\|^2$$

Applying lemma 2.3 to the points $x := x^*, y := y_1$ with $L = L_1$, we get

$$F(x^*) - F(\rho_{L_1}(y_1)) \geq \frac{L_1}{2}\|\rho_{L_1}(y_1) - y_1\|^2 + L_1\langle y_1 - x^*, \rho_{L_1}(y_1) - y_1\rangle$$

$$F(x^*) - F(x_1) \geq \frac{L_1}{2}\|x_1 - y_1\|^2 + L_1\langle y_1 - x^*, x_1 - y_1\rangle$$

consequently using the same pythagorian relation we used earlier we get what is required,

$$\frac{2}{L_1} v_1 \leq \|y_1 - x^*\|^2 + \|x_1 - x^*\|^2$$

□

3 Demonstration

In this section I aim to formulate the image restoration process firstly and then present some results of the same.

3.1 Formulation of image restoration problem

In the previous section we saw the PROX operator, it is not obvious from the defination of the PROX operator that if it well define but it it is indeed one.

In this project we look at applications of ISTA and FISTA in image restoration but they are used for solving otter problems for eg. 1D signal processing problems, ℓ_1 regularized regression problems etc.

We have the optimization problem of the form:

$$\min_x f(x) + g(x)$$

Where:

- $f(x)$ a smooth convex function is the data fidelity term, which measures how well the solution x fits the observed data.

- $g(x)$ a convex function is a regularization term, which promotes certain properties or structures in the solution x.

There are two main steps that are performed in the algorithm

The data fidelity term $f(x)$ typically involves comparing the solution x to the observed data y, often in terms of some norm or distance measure. The goal is to find a solution x that not only minimizes the regularization term $g(x)$ but also fits the observed data y well according to the data fidelity term $f(x)$.

The proximal operator, in general looks for a solution with respect to regularization we have chosen, one very popular in signal processing field is total variation. In FISTA it is the ℓ_1 regularization.

Additionally FISTA has the acceleration step also popularly known as the momentum term, the Adam optimizer used in training of deep network implement this step, these models are obviously are not solving a convex problem but they seem to work well.

$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{12}$$

Beck in his paper performs demonstrations for debluring problem but, this can very well be extended to other restoration tasks as it will be seen.

The algorithm is flexible in the sense that any degradation that can be modeled as a linear operator can be solved. The most easy to see is image denoising in which case $A = I.W^T$. we look at the main models experimented with here.

- Image deblurring:- In this the model is,

$$A = R.W^T \tag{13}$$

  R being the blurring operation and W^T the inverse wavelet transform.

- Image inpainting:- In this the model is,

$$A = M.W^T \tag{14}$$

  M here is the masking operation which simulates random missing pixels

- Image super resolution:- In this the model is,

$$A = R.D.U.W^T \tag{15}$$

  R is the blurring operation, D and U are down sampling and up sampling respectively. This model tries to simulate a low resolution image.

4 Blur

Blur is performed with a Gaussian filter of size 9 and standard dev. 4. The image has been scaled to be in 0 to 1. Haar wavelet is used with 4 levels.

The results are taken on varying noise levels,

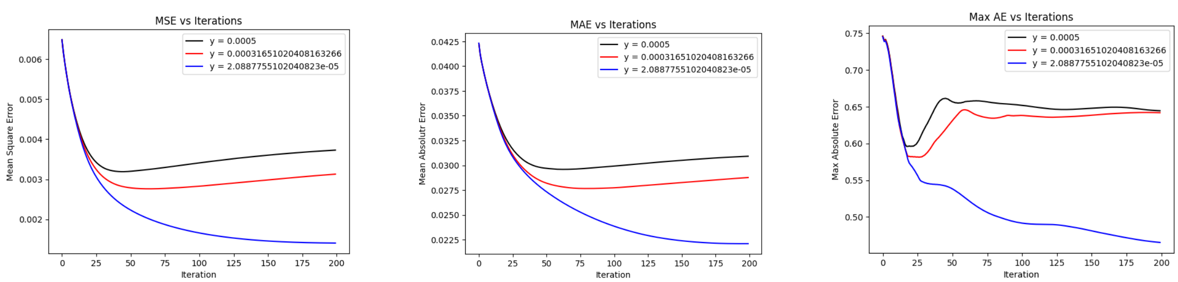
4.1 noise 1e-3

for noise standard dev. 1e-3 results are shown in .

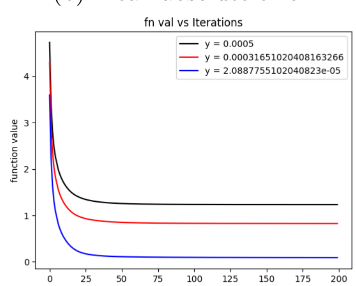
4.1.1 FISTA

(a) $\lambda = 5e - 4$ (b) $\lambda = 3e - 4$ (c) $\lambda = 2e - 5$

Figure 2: FISTA results for varying regularization and noise 1e-3



(a) Mean square error (b) Mean absolute error (c) Max. absolute error



(d) Function value

Figure 3: FISTA statistics plots for 1e-3

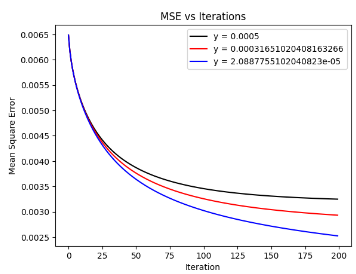
4.1.2 ISTA

(a) $\lambda = 5e-4$

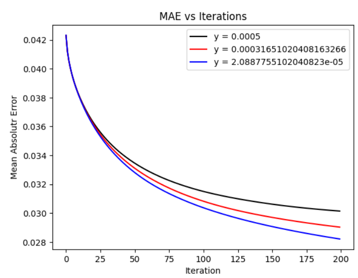
(b) $\lambda = 3e-4$

(c) $\lambda = 2e-5$

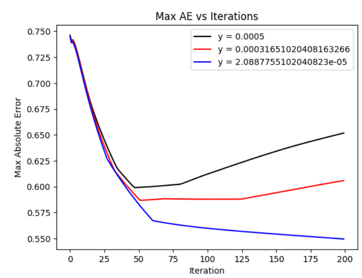
Figure 4: ISTA results for varying regularization



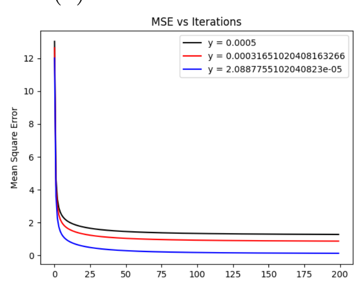
(a) Mean square error



(b) Mean absolute error



(c) Max. absolute error



(d) Function value

Figure 5: ISTA statistics plots

4.2 noise 1e-2

for noise standard dev. 1e-2 results are shown in figure 1

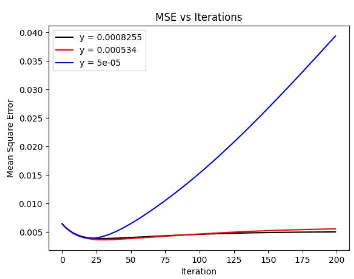
4.2.1 FISTA

(a) $\lambda = 8.255e - 4$

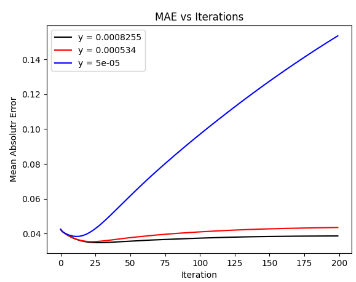
(b) $\lambda = 5.34e - 4$

(c) $\lambda = 5e - 5$

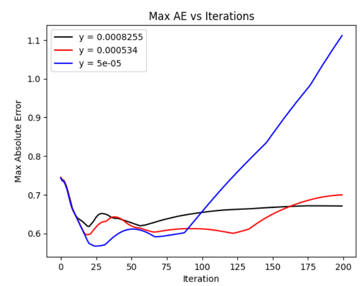
Figure 6: FISTA results for varying regularization



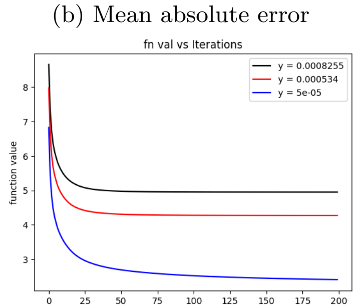
(a) Mean square error



(b) Mean absolute error



(c) Max. absolute error



(d) Function Value

Figure 7: FISTA statistics plots for noise 1e-2

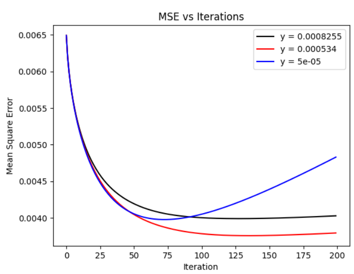
4.2.2 ISTA

(a) $\lambda = 8.255e - 4$

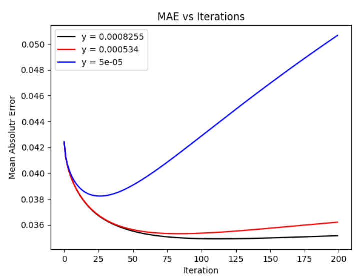
(b) $\lambda = 5.34e - 4$

(c) $\lambda = 5e - 5$

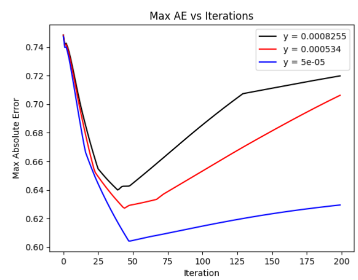
Figure 8: ISTA results for varying regularization



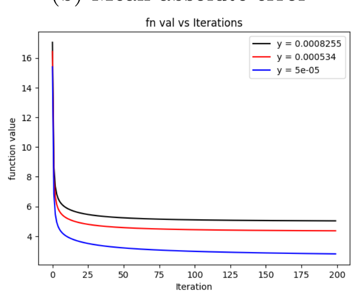
(a) Mean square error



(b) Mean absolute error



(c) Max. absolute error



(d) Function Value

Figure 9: ISTA statistics plots for noise 1e-2

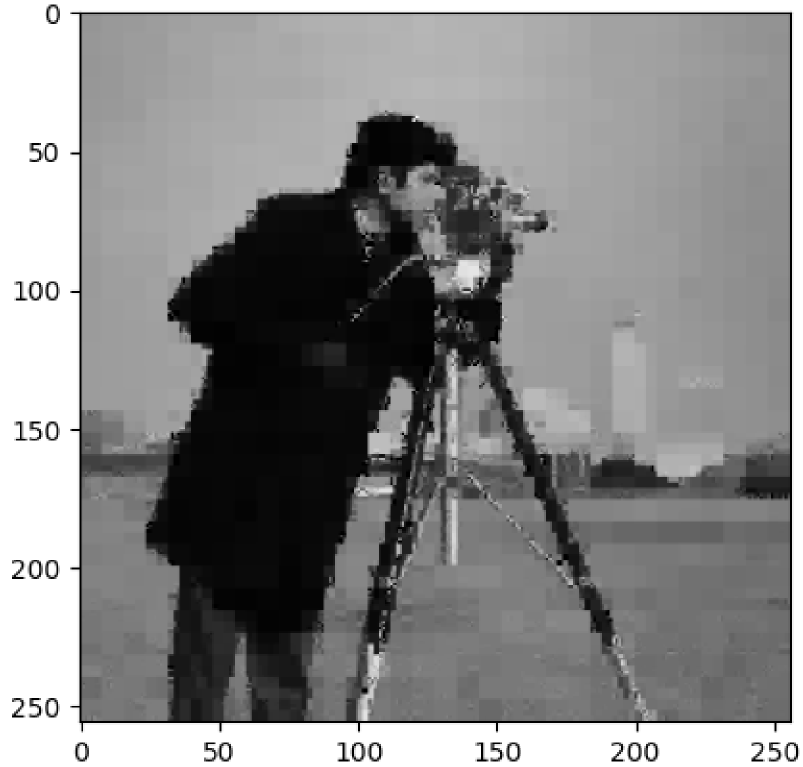


Figure 10: initial observation

5 Inpainting

The image initially was of size 128x128 after looking at some of the early results on this size, it can be seen that there were visible haar residuals(shown in (10)) in the image, the simple idea was to enlarge the image. this enables the haar basis to explain more of the information.

The results are taken by varying % loss in the image,

5.1 Lower loss images

for 50 percent masked image the mean square error of the final result is $6.2e-4$. which is the best I have been able to take it results are as given. Results only for FISTA is given

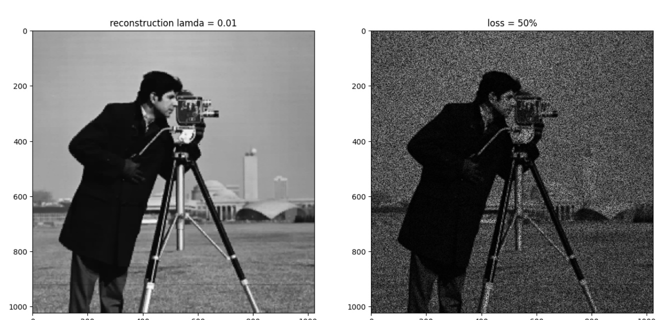


Figure 11: 50% masked image

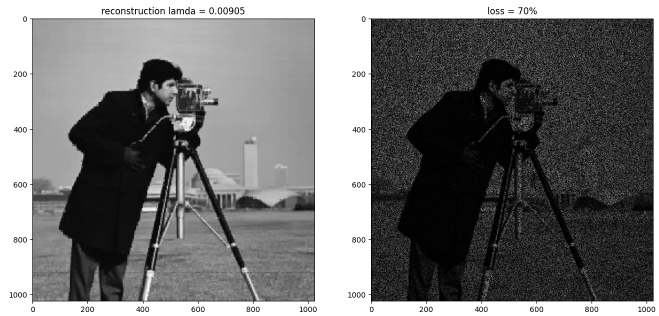


Figure 12: 70% masked image

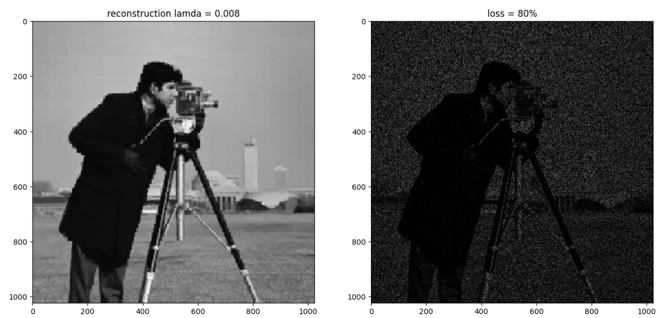
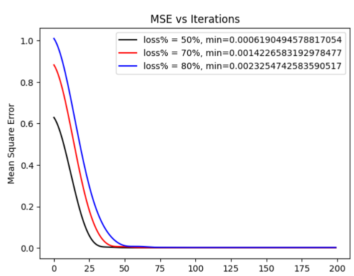
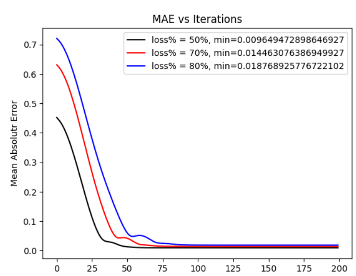


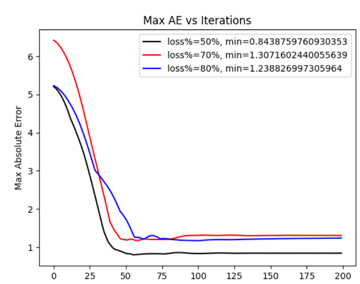
Figure 13: 80% masked image



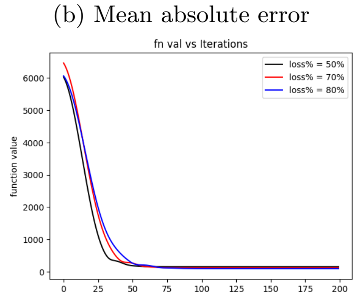
(a) Mean square error



(b) Mean absolute error



(c) Max. absolute error



(d) Function value

Figure 14: statistics plots

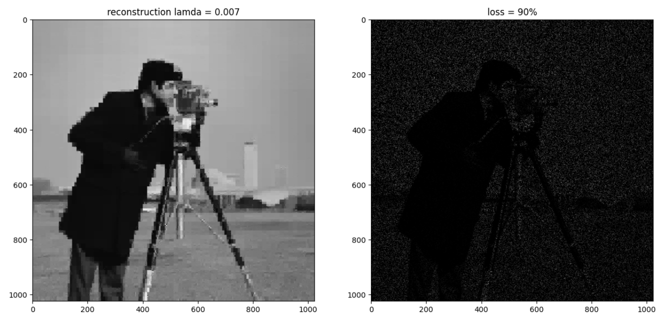


Figure 15: 90% masked image

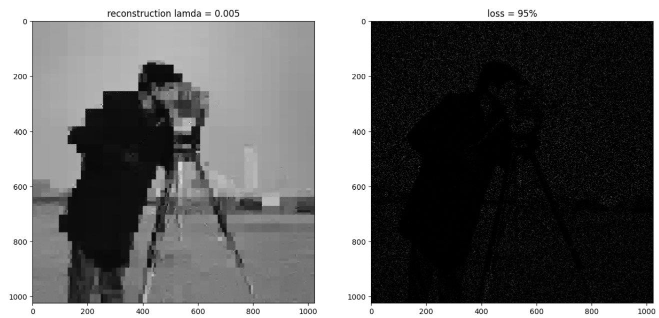


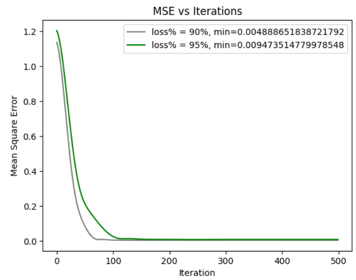
Figure 16: 95% masked image

5.2 Higher loss images

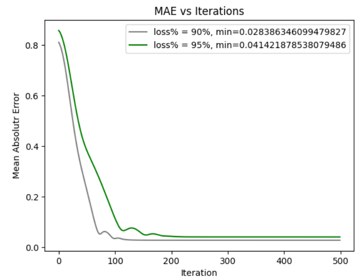
Results for mask 90% and 95% are shown in figure (15), (16) and statistic for the same in figure(17)

6 Super resolution

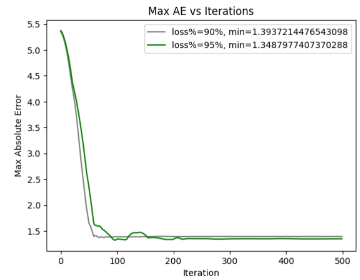
- 2x2 pixelation figure(18), (19), (20)
- 4x4 pixelation figure(21), (22), (23)
- 8x8 pixelation figure(24), (25), (26)



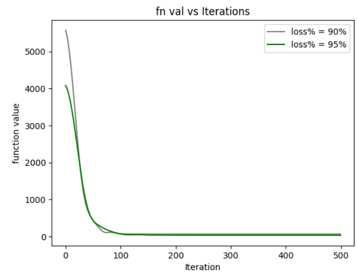
(a) Mean square error



(b) Mean absolute error



(c) Max. absolute error



(d) Function value

Figure 17: statistics plots for 15 and 16

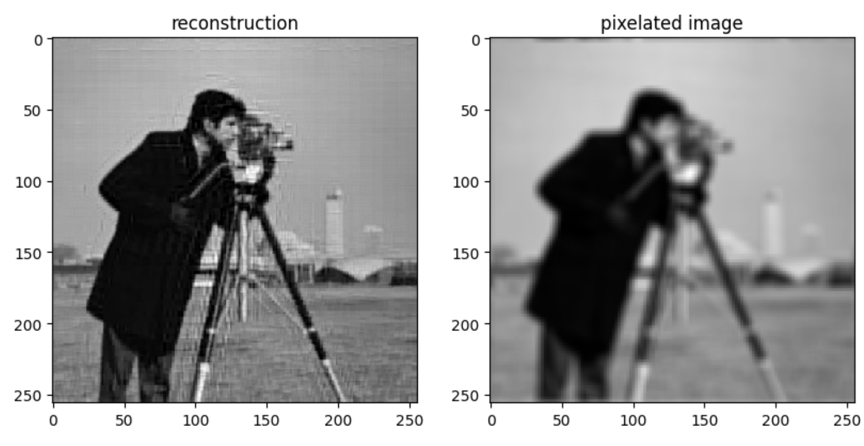


Figure 18: 200 iterations

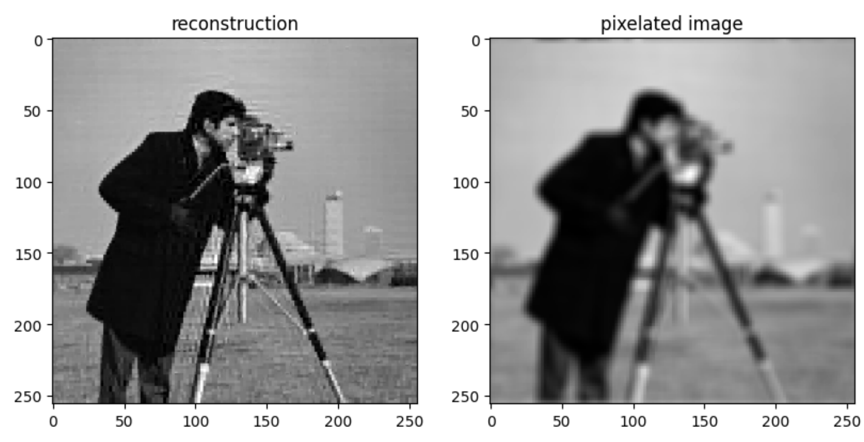


Figure 19: 500 iterations

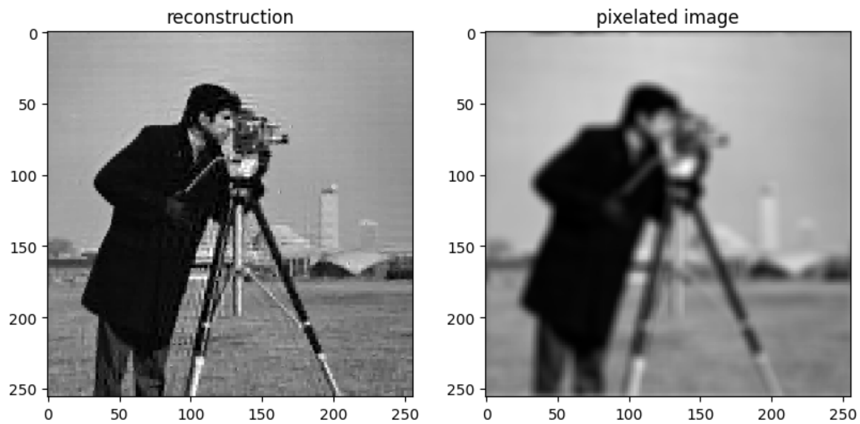


Figure 20: 1000 iterations

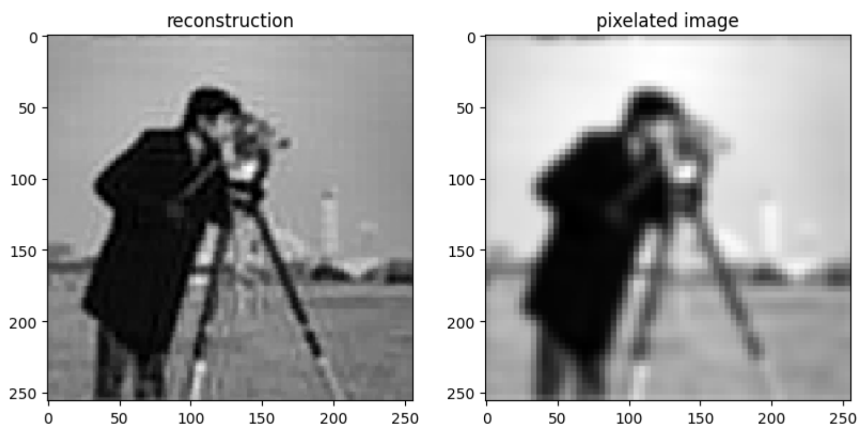


Figure 21: 200 iterations

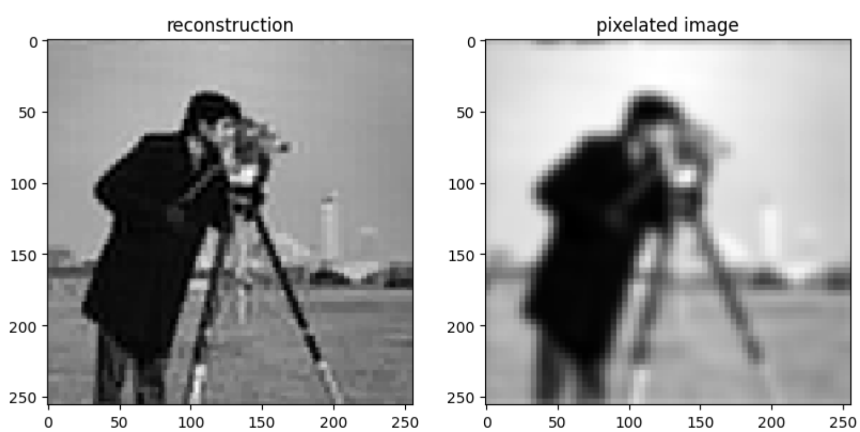


Figure 22: 500 iterations

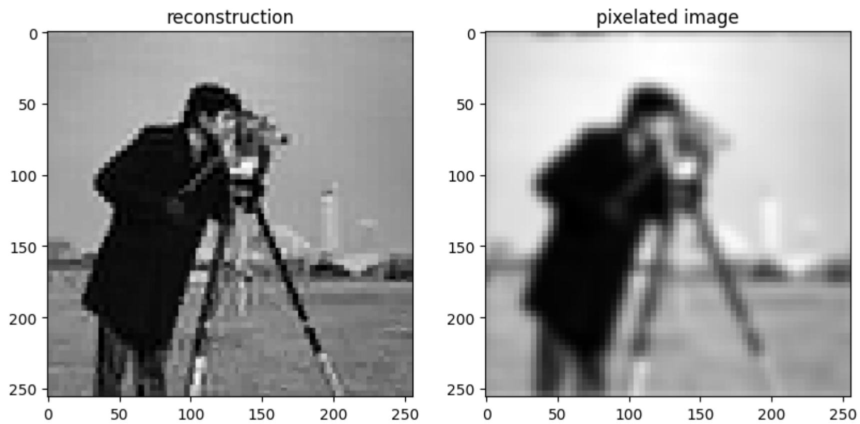


Figure 23: 1000 iterations

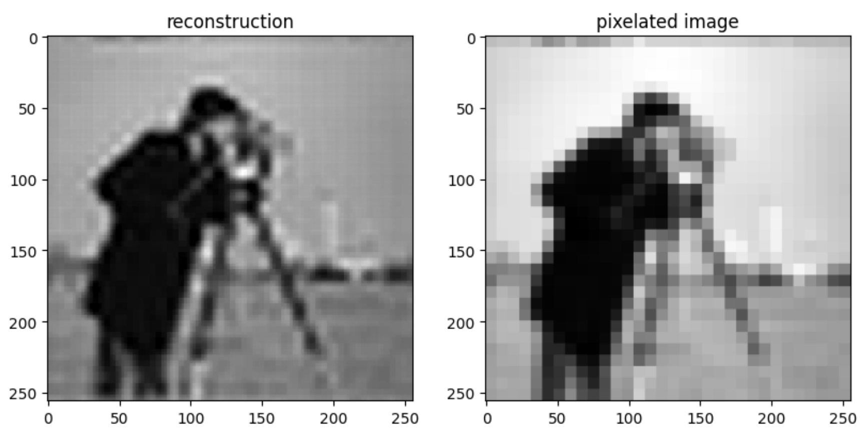


Figure 24: 200 iterations

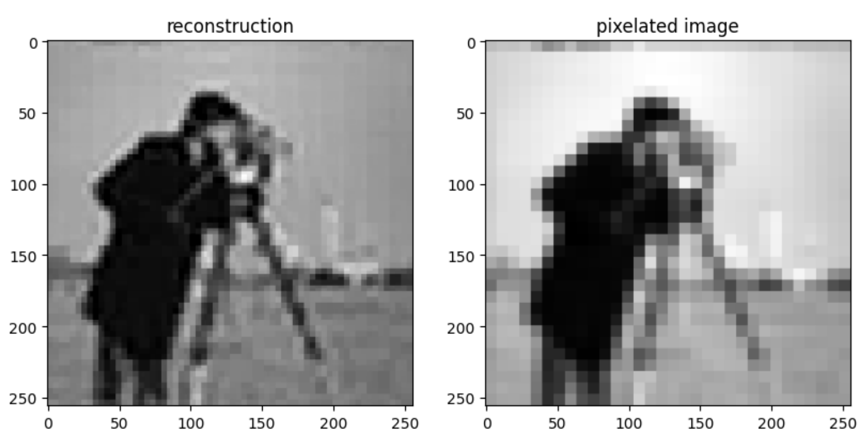


Figure 25: 500 iterations

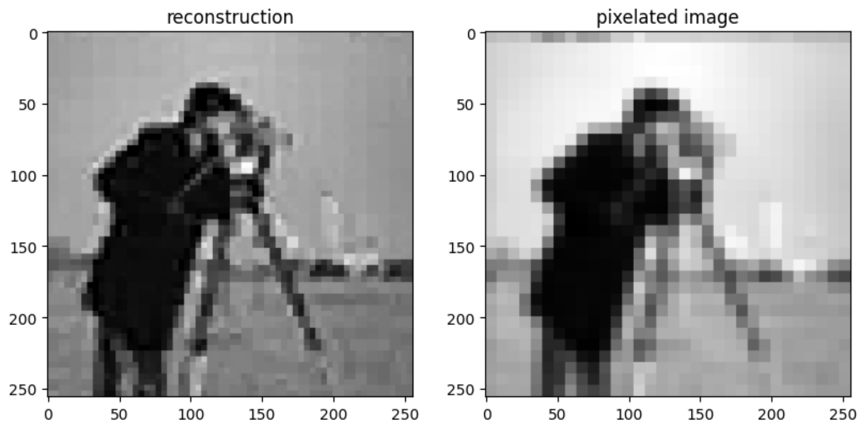


Figure 26: 1000 iterations

7 Code

Listing 1: Code for ISTA

```
def ISTA(params, op):

  x_prev = params.x0
  Lk_1 = params.L0

  for i in range(params.max_iter):
    # get Lipchitz.
    Lk = get_lipchitz(params, Lk_1, x_prev, op)
    # prox of gradent step.
    x_k = p(params, Lk, x_prev, op)

    # update for next iter.
    x_prev = x_k
    Lk_1 = Lk

  reconsISTA  = x_k

  return reconsISTA
```

Listing 2: Code for FISTA

```
def FISTA(params, op):
  x_prev = params.x0
  y_k = x_prev
  t_k = 1
  Lk_1 = params.L0

  for i in range(params.max_iter):

    # get Lipchitz.
    Lk = get_lipchitz(params, Lk_1, x_prev, op)
    # prox of yk.
    x_k = p(params, Lk, y_k, op)
    t_kp1 = (1+np.sqrt(1 + 4 * t_k**2))/2

    # get y_k+1
    alpha = -(t_k - 1) / t_kp1
    y_k = add_wav_coff(1 - alpha, x_k , alpha, x_prev, op)

    # update for next iter.
    x_prev = x_k
    t_k = t_kp1
    Lk_1=Lk

  reconsFISTA  = x_k

  return reconsFISTA
```

References

[BT09] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *Society for Industrial and Applied Mathematics*, 2(1):183–202, 2009.

[MDM04] I. Daubechies M. Defrise and C. D. Mol. An Iterative Thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm.Pure Appl. Math.*, 57(3):346–367, 2004.