Augmented Lagrangian Method Based X-ray CT Image Reconstruction Using Wavelet Tight Frame Transformation

Name: Mohaiminul Al Nahian
RIN : 662026703

Abstract

In this project, a constraint optimization approach based on Augmented Lagrangian (ALM) has been used to reconstruct the X-ray CT image from a limited number of projections. The constructed ALM technique has been used to address the image reconstruction job, which has been formulated as a regularized (based on wavelet tight frame regularization) constraint optimization problem. The implementation findings indicate that ALM may efficiently converge to the optimum point and produce a reconstructed image with minimal mean-squared error while strongly preserving the constraint with the right selection of hyper-parameters. Furthermore, proper selection of post-processing on the reconstructed image can prove to be very effective for quality enhancement of the reconstruction

1 Problems

The use of X-ray Computed Tomography (CT) to create 2-D images of various internal organs of a human has been widely explored. It has shown usefulness in a variety of medical diagnoses tasks (e.g., cancer diagnostics, bone fracture detection etc) [1]

The X-ray photons are passed through the area of interest (the organ of the human body to be imaged) through various routes to create an X-ray CT image. However, because of varying opacity of different tissues, the amount of photons absorbed by that particular tissue is different [2]. A detector picks up the photons that travel through the tissues. According to Beer's Law, the measured photon intensity along various pathways may be used to estimate the value of the attenuation coefficient μ at various locations of the object. The value of these coefficients can be thought of as the pixel intensity of a 2-D image of that object.
The CT image reconstruction problem can be formulated as

$$p_i = log(\frac{b_i}{I_i}) \tag{1}$$

for the X-ray along the path i, where b_i and I_i are incident and detected photon intensity respectively. Then the discrete case of Beer's law can be formulated as

$$p_i = \sum_{j=1}^{n} a_{ij}\mu_j \tag{2}$$

where, $j = 1, 2, , ..., n$ are the analogous pixels through which those particular X-ray photons passed. μ_j is the attenuation co-efficient of the $j - th$ pixel and a_{ij} is the normalized intersection area between the pixel and the ray path.
Suppose, there are $i = 1, 2, ..., m$ no of X-ray doses (i.e., X-ray path) has been applied to construct a CT

image. Then from (1),

$$A\mu = p \tag{3}$$

where, $A = [a_{ij}], \mu = [\mu_j]$ and $p = [p_i]$. Then the task of X-ray CT image reconstruction problem is to estimate the vectorized image $\mu \in R^{n\times 1}$ from the given system matrix $A \in R^{m\times n}$ and projection vector $p \in R^{m\times 1}$.

However, to minimize radiation side effects, the number of X-ray doses (i.e., the number of X-ray paths, m) should be as low as feasible from a medical standpoint. Therefore, the reconstruction problem becomes an under-determined problem in a real-world environment when $m << n$. In order to narrow down the many potential solutions for (3) to one that is adequate, regularization techniques have typically been used. In other words, the regularization problem that the X-ray CT image reconstruction approach is based on is as follows:

$$\min_{\mu} \; R(\mu) \; s.t. \; A\mu = p \tag{4}$$

In this project, wavelet tight frame transformation [3] has been used as regularizer (i.e. $R(\mu)$)

2 Methodology

2.1 Model

The CT image μ can be represented using an over-complete basis W

$$\mu = \sum_{i=1}^{M} a_i w_i = W^T a$$

Here, $\{w_i \in R^n\}_{i=1}^{M}$ form the basis, where number of basis $M > n$. $a \in R^M, W \in R^{M\times n}$ and a is sparse. W is chosen to be a special wavelet tight frame transform where $W^T W = I$. We can formulate the reconstruction problem by the following minimization problem,

$$\min_{a} |a|_0 \; s.t. \; AW^T a = p$$

Relaxing it to a convex optimization problem gives

$$\min_{a} \; |a|_1 \; s.t. \; AW^T a = p \tag{5}$$

2.2 Algorithm

We can solve the constrained optimization of (5) by representing it by the following unconstrained optimization problem,

$$\min_{a} \max_{v} L(a, v) \tag{6}$$

where, $L(a, v) = |a|_1 + \langle v, p - AW^T a\rangle + \frac{\lambda}{2}||AW^T a - p||^2$ is the Augmented Lagrangian Function. a is the primal variable and v is the dual variable. $L(a, v)$ is convex with respect to both a and v. Additionally, the function is smooth for v. L, however, is not smooth as a function of a. So, it is actually the sum of one smooth function and one non-smooth function. Here, the non-smooth function is $|a|_1$, and we can quickly determine its proximal operator. As a result, we may solve the minimization problem using proximal gradient descent and the maximizing problem using gradient ascent at the same time.The two optimization problems become,

$$a^* = arg\ \min_a\ L_a$$

$$v^* = arg\ \max_v\ L_v$$

So, basically we need to implement the following two equations simultaneously during the optimization process,

$$a^{k+1} = arg\ \min_a |a|_1 + \langle v^k, p - AW^T \rangle + \frac{\mu}{2}||AW^T a - p||^2 \tag{7}$$

$$v^{k+1} = v^k + \rho \nabla_v L(v^k) \tag{8}$$

where, $\nabla_v L(v^k) = p - AW^T a^{k+1}$

As already mentioned, L is non-smooth for a, the minimization problem of (7) can be performed by applying proximal gradient descent algorithm by the following equation

$$a^{k,t+1} = Prox_{\tau|.|}(a^{k,t} - \tau \nabla_a L(a^{k,t}))$$

Or,

$$a^{k,t+1} = Shrink[a^{k,t} - \tau(WA^T(\lambda(AW^T a^{k,t} - p) - v^k))] \tag{9}$$

so, the optimization problem can be summarized in the following way

- Initialize a, v, λ
- solve equation (7) to update a
- Solve equation (8) to update v
- Continue the above two steps until converges

The inverse wavelet transformation $istw2(a^* : 1 : 1)$ is then used to reconstruct the CT image.

The algorithms to solve equation (7) and (8) are shown below

Algorithm 1 ALM based CT Image Reconstruction

Step 0. Initialize $a^0, v^0, \lambda, \rho, tol_1, tol_2, maxiter_1, maxiter_2$
while $k \leq maxiter_1$ **do**
  Step 1. Calculating a^{k+1}
    step 0: $a^{k+1,1} = a^k$
    step 1:
    while $t \leq maxiter_2$ Or, $\frac{||a^{k+1,t+1} - a^{k+1,t}||}{max(1,||a^{k+1,t} - a^{k-1,t-1}||)} \geq tol_2$ **do**
      $a^{k+1,t+1} = Shrink[a^{k,t} - \tau(WA^T(\lambda(AW^T a^{k,t} - p) - v^k))]$
    end while
    step 2: $a^{k+1} = a^{k+1,t+1}$
  Step 2. $v^{k+1} = v^k + \rho \nabla_v L(v^k)$
  Step 3.
  if $\frac{||a^{k+1} - a^k||}{max(1,||a^k - a^{k-1}||)} \leq tol_1$ and $\frac{||v^{k+1} - v^k||}{max(1,||v^k - v^{k-1}||)} \leq tol_1$ **then**
    GO TO **Step 4**
  end if
end while
Step 4. $a^* \leftarrow a^{k+1}$

2.3 Hyper-parameter Setting

$a^0 = 0$ and $v^0 = 0$ has been chosen. The other hyper-parameters are chosen to be $\lambda = 0.009, \rho = 0.0005, \tau = 0.001, tol_1 = tol_2 = 0.001$

2.4 Post-processing of the Reconstructed Image

The reconstructed image, albeit resembling the original one to a large extent, has some noise, especially Salt and Pepper type of noise. In image processing domain, median filtering is used often times to remove the mentioned type of noise from an image [4]. so, as a further post-processing step, we have passed the reconstructed image through a median filtering process, where the median filter size has been chosen to be 3×3. The resulting image and corresponding performance metrics are also reported in Section 3.

3 Results

To evaluate the performance of the implemented algorithm, following metrics has been calculated for both the raw reconstructed image and the median filtered image

- **Mean Squarred Error (MSE):** The per-pixel MSE is a good measure of performance for reconstructed images. The value is calculated using the equation,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mu^i - \mu^i_{true})^2$$

  where n is the total number of pixels, μ_i is reconstructed image pixel and μ_{true} is the actual image pixel intensity. Low MSE means better reconstruction

- **Normalized distance:** the normalized distance between the true image and the reconstructed image has been calculated using the equation

$$d = \frac{||\mu - \mu_{true}||_F}{||\mu_{true}||_F}$$

  A low value of d indicates better reconstruction.

- **Normalized Constraint Value:** A small normalized constraint value $\frac{||A\mu - p||}{||p||}$ indicates that the constraint has been preserved during the optimization process

Table 1 summarizes the results of the experiment. The total number of iteration required for convergence for the selected hyper-parameters is 111. Both the reconstructed image and post-processed (median filtered) image have low MSE, normalized distance and normalized constrained value. However, the post-processed image performs slightly better in terms of MSE and normalized distance, whereas, the raw reconstructed image has better normalized constrained value.
Figure 1 shows the original image and the raw reconstructed image side by side. Figure 2 shows the reconstructed image and the slightly visually better post-processed image side by side. Figure 3, 4 and 5 show the MSE, Normalized Constraint Value and the Augmented Lagrangian value w.r.t iterations respectively

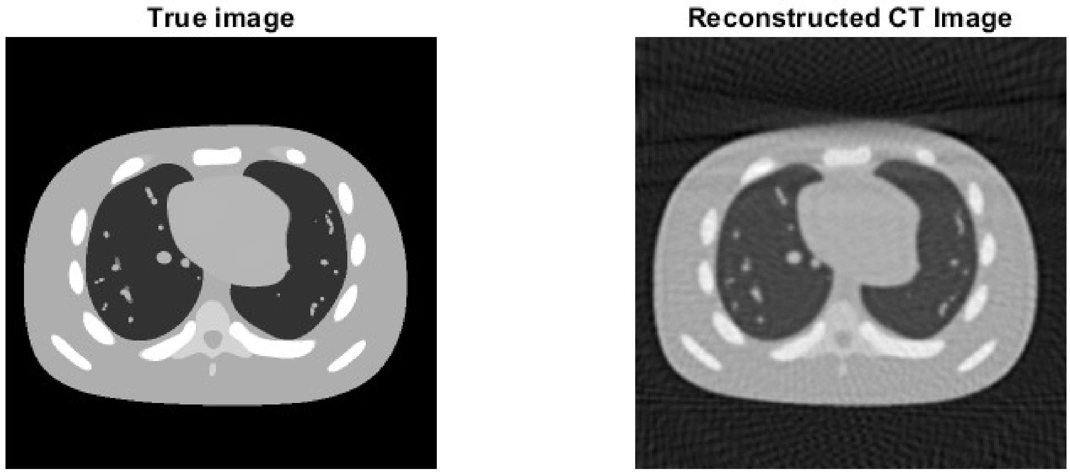

Figure 1: Left:True Image, Right: Reconstructed Image.

	Reconstructed Image	**Median Filtered Image**
Iteration for convergence	111	-
MSE	$2.312e^{-6}$	$2.111e^{-6}$
Distance (normalized)	0.1038	0.0992
Normalized Constrained Value	0.0041	0.0045

Table 1: Performance Metrics Values for Reconstructed Image

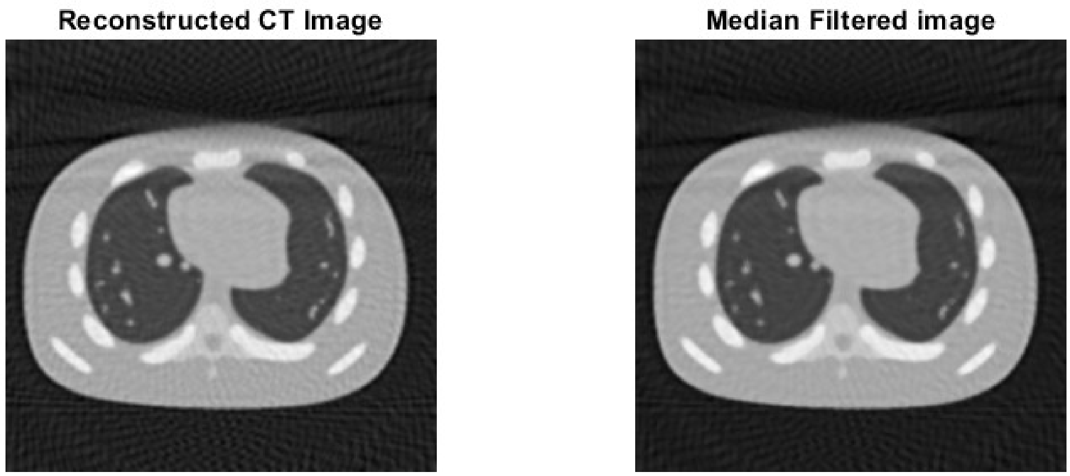

Figure 2: Left:Reconstructed Image, Right: Post-processed Image.

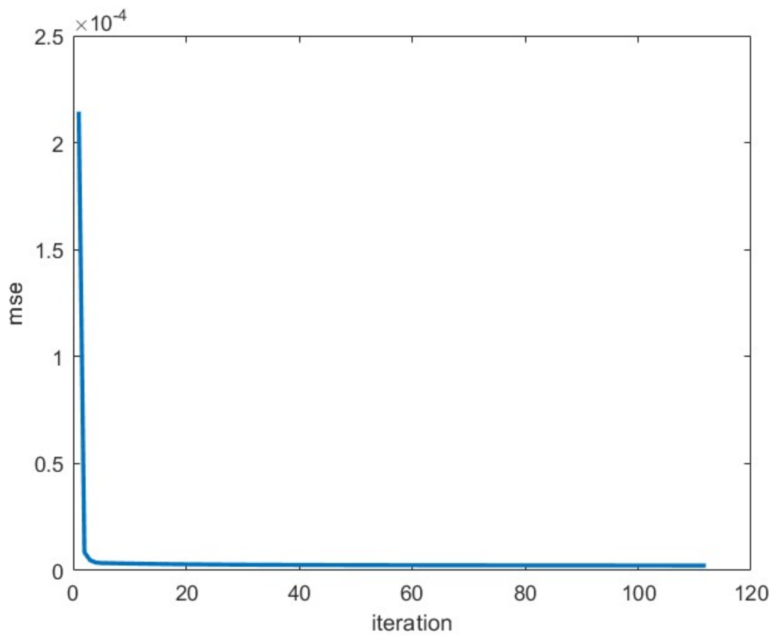

Figure 3: Mean Squarred Error vs Iteration

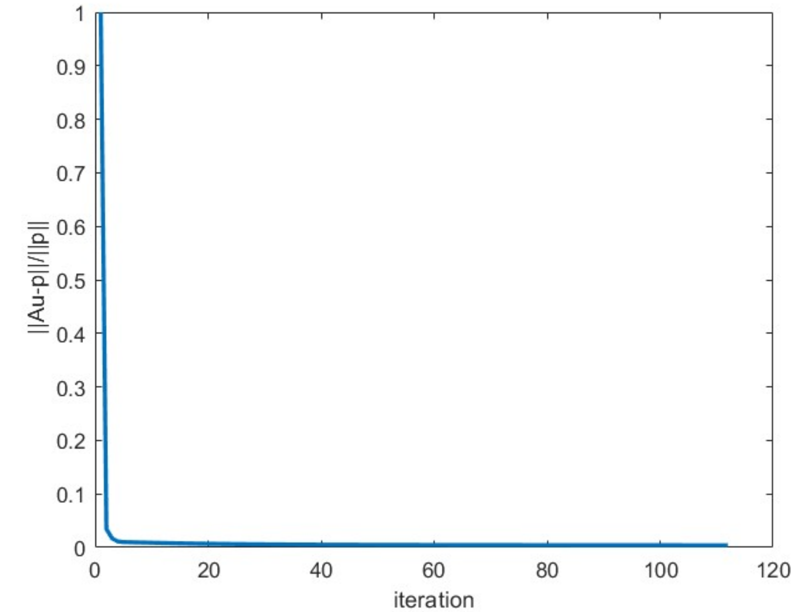

Figure 4: Normalized Constraint vs Iteration

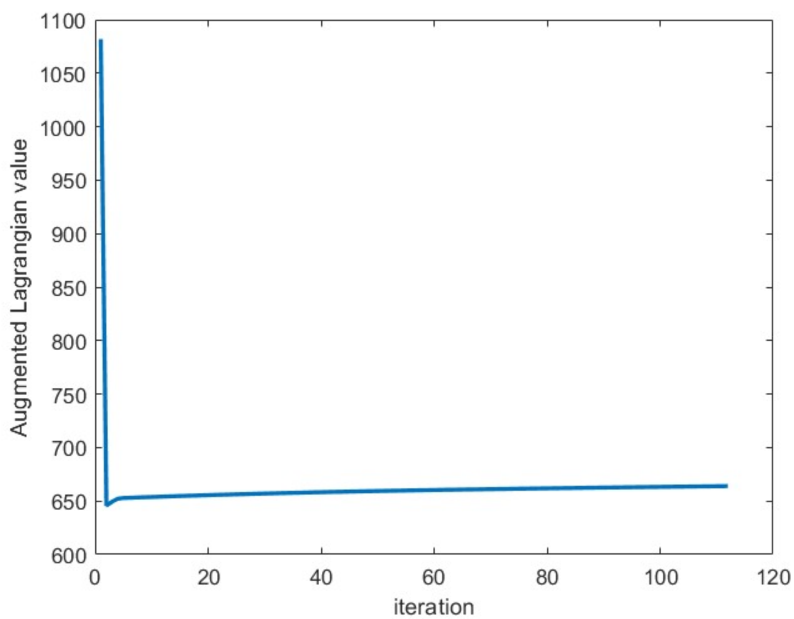

Figure 5: Augmented Lagrangian Value vs Iteration

4 Observation and Conclusions

Reconstructed image of Figure 1 reveals that, despite the algorithm's convergence, there is some background noise present. This background noise has been slightly reduced by applying a median filter and the slightly visually attractive image has been found in Figure 2. The lower MSE and Normalized Frobenius distance of the filtered image as compared to the raw reconstructed image indicate that some noise has been removed from the reconstructed image. However, the raw reconstructed image has better normalized constrained value, which is obvious, because the raw reconstructed image was found by preserving the constraint during the optimization process, whereas post-processed image is just a filtered version, agnostic about the actual optimization process, so not keeping the constraint intact. Another issue is that, the optimization process highly depend on the proper hyper-parameter selection, influencing both convergence time and final performance. Nevertheless, the fairly excellent reconstruction result suggests that the Augmented Lagrangian Method appears to be a powerful tool for reconstructing CT-image accurately.

References

[1] Landis K Griffeth. Use of pet/ct scanning in cancer patients: technical and practical considerations. In *Baylor University Medical Center Proceedings*, volume 18, pages 321–330. Taylor & Francis, 2005.

[2] Rangaraj M Rangayyan. *Biomedical image analysis*. CRC press, 2004.

[3] Yong Zhang, Bin Dong, and Zhaosong Lu. minimization for wavelet frame based image restoration. *Mathematics of Computation*, 82(282):995–1015, 2013.

[4] Uğur Erkan, Levent Gökrem, and Serdar Enginoğlu. Different applied median filter in salt and pepper noise. *Computers & Electrical Engineering*, 70:789–798, 2018.

Appendix

Codes used in this project

```
clc;
clear all;
close all;

load CTReconPhantom

figure;
imshow(p_img,[]);
title('p')
figure;
imshow(TrueImage,[]);
title('True image');

max_iteration=10000;
max_iteration_a=100;
tol_1=1e-3;
tol_2=tol_1;
tol_a_1=1e-3;
a=zeros(256,256,9);
v=zeros(30720,1);

tau=0.001;
```

```
lam=0.009;
rho=0.0005;
a_pre=a;
v_pre=v;
mse_plot=[];
cons_plot=[];
Lag_plot=[];
Recon=iswt2(a,1,1);
constr=p-(A*Recon(:));
Lag_plot(1,1)=norm(a(:),1)+(v'*constr)+((lam/2)*(norm(constr))^2);
mse_plot(1,1)=immse(Recon,TrueImage);
cons_plot(1,1)=norm((A*Recon(:))-p)/norm(p);
for j=1:max_iteration
    a_pre_one=a_pre;
    a_pre=a;
    v_pre_one=v_pre;
    v_pre=v;
    [a,i]=inn_loop(a,tau,A,p,v,lam,max_iteration_a,tol_a_1);
    w_a=iswt2(a,1,1);
    v=v+(rho*(p-(A*w_a(:))));
    Recon=iswt2(a,1,1);
    mse_plot(1,j+1)=immse(Recon,TrueImage);
    cons_plot(1,j+1)=norm((A*Recon(:))-p)/norm(p);
    constr=p-(A*Recon(:));
    Lag_plot(1,j+1)=norm(a(:),1)+(v'*constr)+((lam/2)*(norm(constr))^2);
    if (norm(a(:)-a_pre(:))/max(1,norm(a_pre(:)-a_pre_one(:)))<tol_1)
    &&(norm(v-v_pre)/max(1,norm(v_pre-v_pre_one))<tol_2)
        break
    end
    fprintf('Iteration:%i\n',j);
end

fprintf('Iteration required for convergence:%i\n',j);

figure;
imshow(Recon,[]);
title("Reconstructed CT Image")
x=1:j+1;
figure;
plot(x,mse_plot,'LineWidth',2)
xlabel('iteration')
ylabel('mse')
figure;
plot(x,cons_plot,'LineWidth',2)
xlabel('iteration')
ylabel(' ||Au-p||/||p||')
figure;
```

```
plot(x,Lag_plot,'LineWidth',2)
xlabel('iteration')
ylabel('Augmented Lagrangian value')

Recon=iswt2(a,1,1);

Immed=medfilt2(Recon,[3 3]);
figure;
imshow(Immed,[])
title("Median Filtered image")

cv_p=norm((A*Recon(:))-p)/norm(p);
d_raw=norm(Recon-TrueImage,'fro')/norm(TrueImage,'fro');
mse=immse(Recon,TrueImage);
fprintf('recon image ||Au-p||/||p||:%f\n',cv_p);
fprintf('Normalized Frobenius distance between true image and reconstructed image:%f\n',d_raw);
fprintf('Mean squared error of the reconstructed image:%f\n',mse);

mse2=immse(Immed,TrueImage);
fprintf('Mean squared error of the medfilt image:%f\n',mse2);
d_med_normalized=norm(Immed-TrueImage,'fro')/norm(TrueImage,'fro');
fprintf('Normalized Frobenius distance between true image and Median filteres image:%f\n',d_med_normalized);
cv_p_med=norm((A*Immed(:))-p)/norm(p);
fprintf('median filtered image ||Au-p||/||p||:%f\n',cv_p_med);



function [a,i]=inn_loop(a_s, tau, A, p, v, mew,max_iter_a,tol_a_loop)
a_t_prev=a_s;
for i=1:max_iter_a
    a_s_prev_prev=a_t_prev;
    a_t_prev=a_s;
    W_a=iswt2(a_s,1,1);
    c=A'*(mew*((A*W_a(:))-p)-v);
    a_s=a_s-(tau*(swt2(reshape(c,[256,256]),1,1)));
    a_s=cat(3,a_s(:,:,1),wthresh(a_s(:,:,2:end),'s',tau));

    if norm(a_s(:)-a_t_prev(:))/max(1,norm(a_t_prev(:)-a_s_prev_prev(:)))<tol_a_loop
        break;
    end
end
a=a_s;
end
```