MATH473 HW6

Jiasen Zhang

Primal-Dual Algorithm

The problem is:

$$\min_x \frac{1}{2}||Ax-b||_2^2 + \alpha||Wx||_1 \qquad A = PF$$

The corresponding primal-dual problem is:

$$\min_x \max_p \frac{1}{2}||Ax-b||_2^2 + \alpha < Wx, p >$$

First solve the x-sub problem:

$$\min_x \frac{\tau}{2}||Ax-b||_2^2 + \alpha\tau < Wx, p > + \frac{1}{2}||x - x^k||_2^2$$
$$\alpha\tau < Wx, p >= \alpha\tau < x, W^*p >$$

$$\tau A^*(Ax-b) + \alpha\tau W^*p + x - x^k = 0$$
$$(\tau A^*A + I)x = \tau A^*b - \alpha\tau W^*p + x^k$$
$$F(\tau F^*P^*PF + I)F^*Fx = \tau FF^*P^*b + F(-\alpha\tau W^*p + x^k)$$
$$(\tau P^*P + I)Fx = \tau P^*b + F(-\alpha\tau W^*p + x^k)$$

Then solve the p-sub problem:

$$\max_p \alpha\sigma < Wx, p > -\frac{1}{2}||p - p^k||_2^2$$
$$\alpha\sigma Wx - (p - p^k) = 0$$
$$p' = p^k + \alpha\sigma Wx$$
$$p_{ij} = \frac{p'_{ij}}{\max(1, |p'_{ij}|)}$$

1 Radial sampling mask

I set $\tau = \frac{1e-3}{\alpha}$ and $\sigma = 10000 * \tau$ with which the result performs best. From Table 1 we can see $\alpha = 1e-5$ corresponds to the largest SNR and smallest relative error.

α	SNR	Relative Error
1e-3	39.3987	1.0717e-2
1e-4	41.3587	8.5519e-3
1e-5	41.3817	8.5293e-3
1e-6	41.3814	8.5296e-3
1e-7	41.3708	8.5400e-3

Table 1: α vs. SNR and relative error

The result with $\alpha = 1e-5$ is shown in Figure 1.

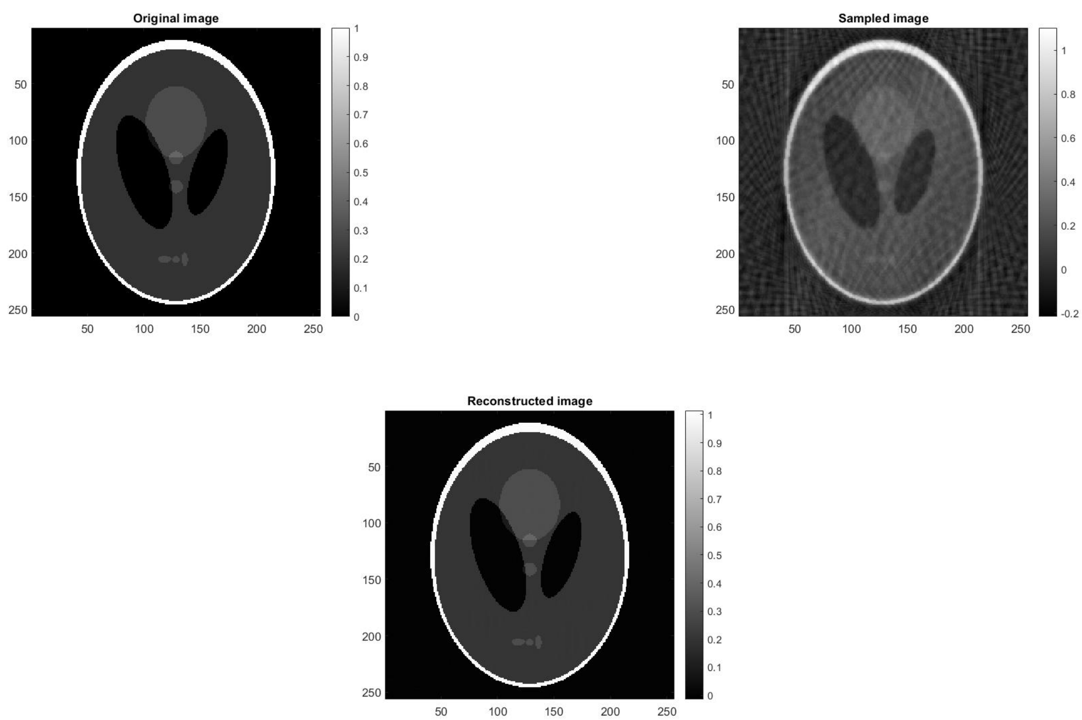

Figure 1: Reconstruction of radial sampling mask.

2 Random sampling mask

2.1 c=20

Now use a random sampling mask choosing $c\%$ of data outside a small square. Here $c = 20$. From Table 2 we can see $\alpha = 1e-6$ corresponds to the largest SNR and smallest relative error.

α	SNR	Relative Error
1e-3	27.8095	4.0694e-2
1e-4	45.7048	5.1851e-3
1e-5	46.0436	4.9868e-3
1e-6	46.1541	4.9237e-3
1e-7	45.7131	5.1802e-3

Table 2: α vs. SNR and relative error

The result with $\alpha = 1e-6$ is shown in Figure 2.

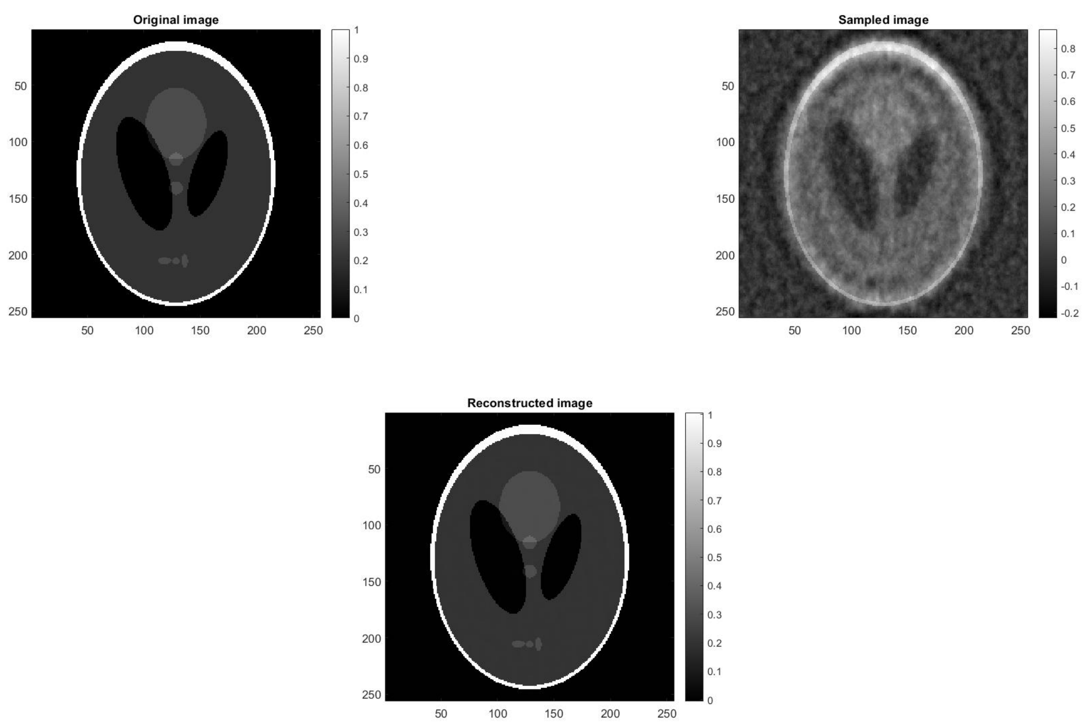

Figure 2: Reconstruction of random sampling mask with $c = 20$.

2.2 c=10

When $c = 10$, from Table 3 we can see $\alpha = 1e-6$ corresponds to the largest SNR and smallest relative error.

α	SNR	Relative Error
1e-3	11.6472	2.6160e-1
1e-4	11.8358	2.5598e-1
1e-5	13.0707	2.2206e-1
1e-6	13.4722	2.1203e-1
1e-7	13.2431	2.1769e-1

Table 3: α vs. SNR and relative error

The result with $\alpha = 1e-6$ is shown in Figure 3.

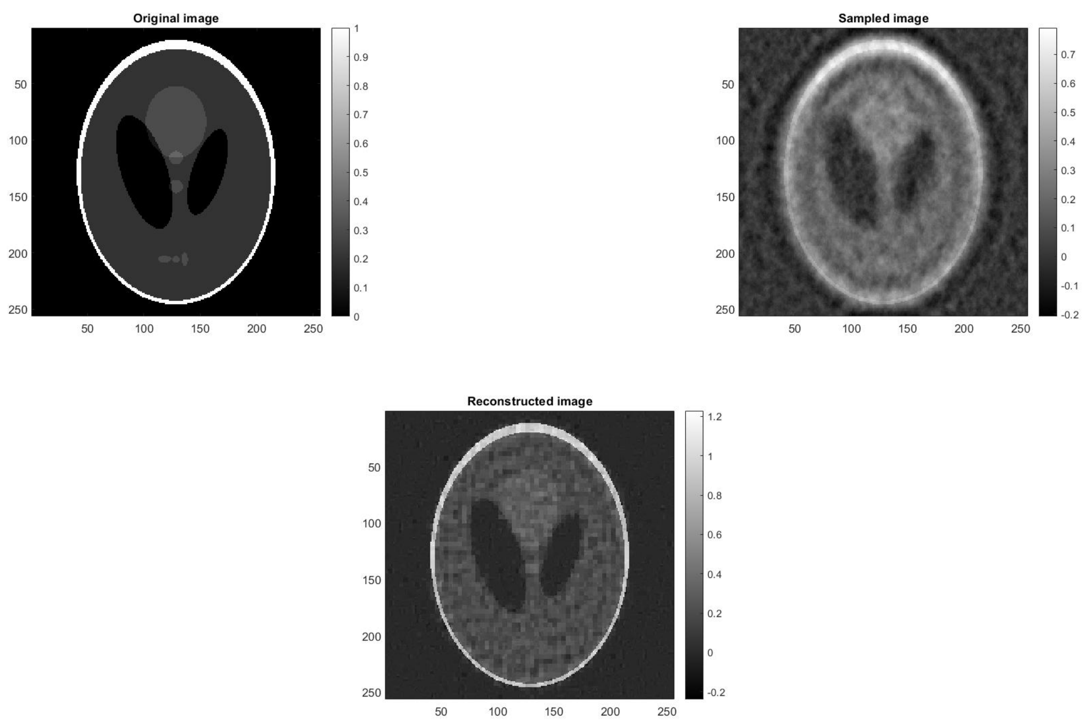

Figure 3: Reconstruction of random sampling mask with $c = 10$.

2.3 c=5

When $c = 5$, from Table 4 we can see $\alpha = 1e - 7$ corresponds to the largest SNR and smallest relative error.

α	SNR	Relative Error
1e-4	6.3579	4.8096e-1
1e-5	6.3940	4.7896e-1
1e-6	6.4531	4.7571e-1
1e-7	6.6115	4.6712e-1
1e-8	6.3627	4.8069e-1

Table 4: α vs. SNR and relative error

The result with $\alpha = 1e - 7$ is shown in Figure 4.

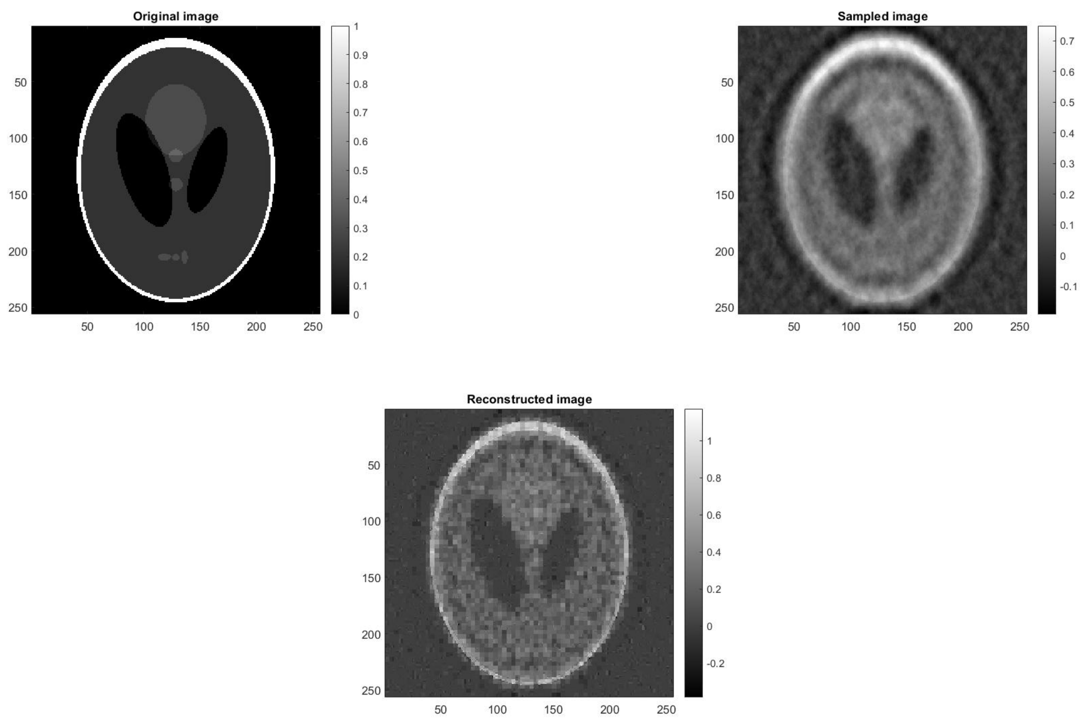

Figure 4: Reconstruction of random sampling mask with $c = 5$.

3 Matlab Code

```
clear all;clc;
tic
X = phantom(256);

% %%%%%%%Mask 1
% n = size(X,1); % X is a square matrix with dimension nxn.
% Ls = 50;
% sigma = 0.05;
% SpMsk= fftshift(MRImask(n,Ls));
% idx = find(SpMsk≠0);
% len = nnz(SpMsk); % number of nonzeros in SpMsk;
% K = fft2(X).*SpMsk; % simulate partial Fourier data K
% % add complex valued noise to the locations with data sampled
% K(idx) = K(idx)+sigma*(randn(len,1)+sqrt(-1)*randn(len,1));
% P = zeros(n,n);
% P(SpMsk≠0)=1;

% %%%%%%%Mask 2
n = size(X,1);
Mask = zeros(n,n);
c=20; % select c% of pixels in Mask
num = round(n*n*c/100);
idxAll = randperm(n^2);
idxChoose = idxAll(1:num);
Mask(idxChoose) = 1;
center = round(n/2);
h = 10;
Mask(center-h+1:center+h,center-h+1:center+h) = 1;
Mask = fftshift(Mask);
K = fft2(X).*Mask;
P = zeros(n,n);
P(Mask≠0)=1;

Psi = @(x) Wavedb1Phi(x,0);
PsiT = @(x) Wavedb1Phi(x,1);

% initialization
x = zeros(n,n);
p = zeros(1,n^2);

% parameters
alpha = 1e-6; % to be tuned
tau = 10;
sigma = 10000*tau;
snr = 0;

temp = sum(sum(X.^2))/(n*n);
% Primal-Dual Algorithm
for k=1:100000
    xold=x;
    snr0 = snr;

    p0 = p + sigma*alpha*Psi(x);
    p=p0./max(1,abs(p0));

    rhs = tau*K + fft2(-alpha*tau*PsiT(p) + x);
    x = ifft2( rhs./(tau*(P)+1) );
    x=real(x);


    mse = sum(sum((x-X).^2))/(n*n);
```

```
    snr = 10*log10(temp/mse);
    if mod(k,100)==0
        imagesc(x);colormap(gray);axis square;drawnow;
        fprintf('%d \t %f \n',k,snr);
    end
    if snr<snr0
        break;
    end
end

% compute MSE and relative error
mse = sum(sum((x-X).^2))/(n*n);
temp = sum(sum(X.^2))/(n*n);
snr = 10*log10(temp/mse);
fprintf('SNR = %f \n',snr);
re = norm(x-X,'fro')/norm(X,'fro');
fprintf('Relative Error = %e \n',re);

% plot
B = real(ifft2(K));
figure(1);imagesc(X);colormap(gray);colorbar;
title('Original image');axis square;
figure(2);imagesc(B);colormap(gray);colorbar;
title('Sampled image');axis square;
figure(3);imagesc(x);colormap(gray);colorbar;
title('Reconstructed image');axis square;
toc
```