Berfin Kavşut
21602459
EEE475 – HW#1
March 12, 2020

THE REPORT FOR HW#1

Introduction

The performances of four different partial k-space reconstruction techniques , which are trivial reconstruction, PCCS reconstruction, homodyne reconstruction and POCS reconstruction, will be studied in this homework. Bot visual results and IQA results (PSNR and SSIM) are examined.

For MATLAB codes, the parts for creating figures and displaying information are not represented, because the MATLAB code part was getting very long. These parts can be seen in m-file.

Comments, display results, plots, and MATLAB code are presented for each question. The comments are included for the questions requiring comments.

Throughout the solutions, it will be seen that x-y directions and kx-ky direction are in opposite ways. The reason is that k-space data is given with Nro x Npe size (Nro is the number of readout samples, and Npe is the number of phase encode lines). Instead of flipping the images for each result, the order of axes are changed.

Question 1 – Full k-space Data

1. Plots

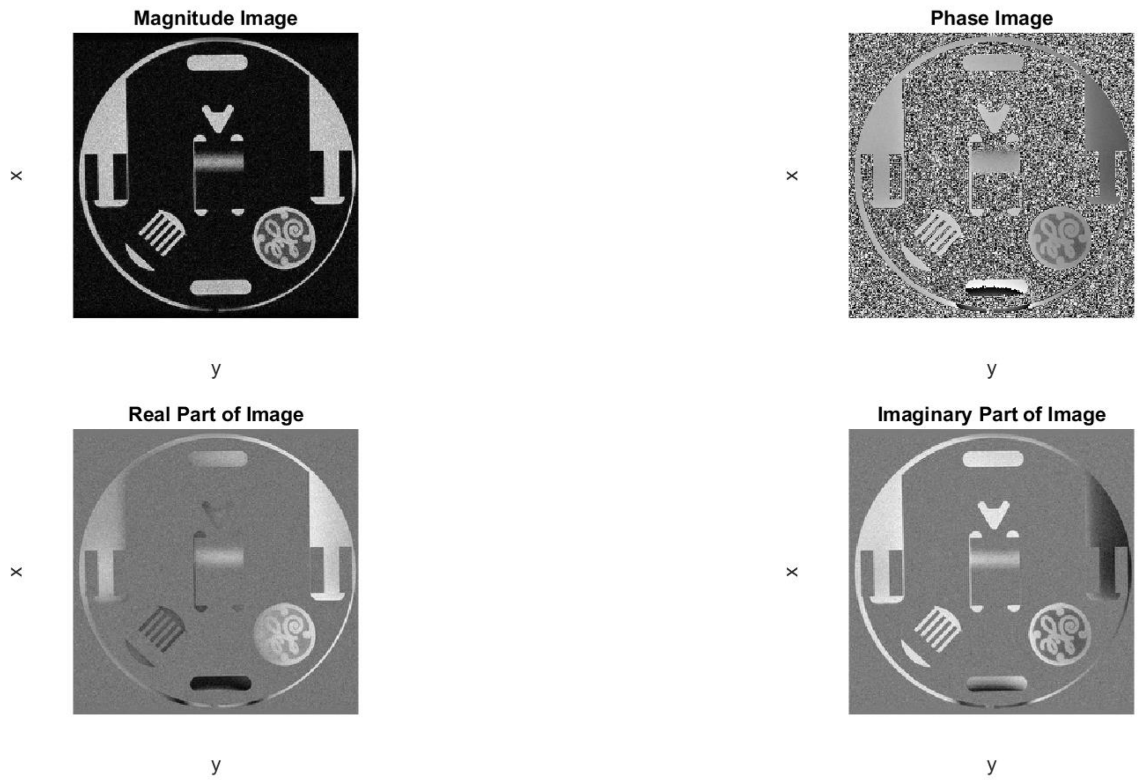

Figure 1. Magnitude Image, Phase Image, Real Part and Imaginary Part of the Image

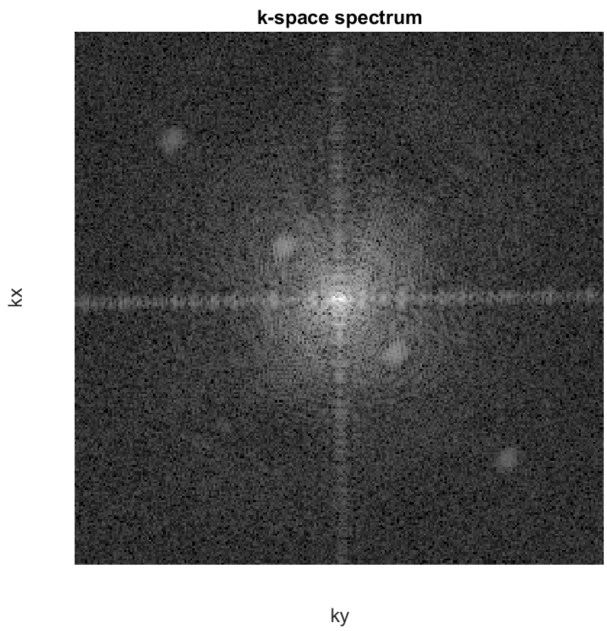

Figure 2. k-space Spectrum of the Image

2. MATLAB Code

```
%full k-space data
load('ge_phantom.mat');

[Nro Npe] = size(kdata);
fprintf('Number of readout samples: %d\n',Nro);
fprintf('Number of phase encode lines: %d\n',Npe);

%display the k-space spectrum as an image
figure;
set(gcf, 'WindowState', 'maximized');
imshow(log(abs(kdata)+1),[]);
title('k-space spectrum');
xlabel('ky'); ylabel('kx');
saveas(gcf,'1.1_k-space_data.png');

%compute reference image
%the reference image will be used for comparison
%in different reconstruction techniques
mf = ifft2c(kdata);
ref = abs(mf);
```

Question 2 – Phase-compensated Full k-space Image with ±1/8th of k-space data

1. Display Results

1.2

Phase-compensated Full k-space Image (±1/8th of k-space data)

PSNR: 25.7442

SSIM: 0.6387

2. Plots

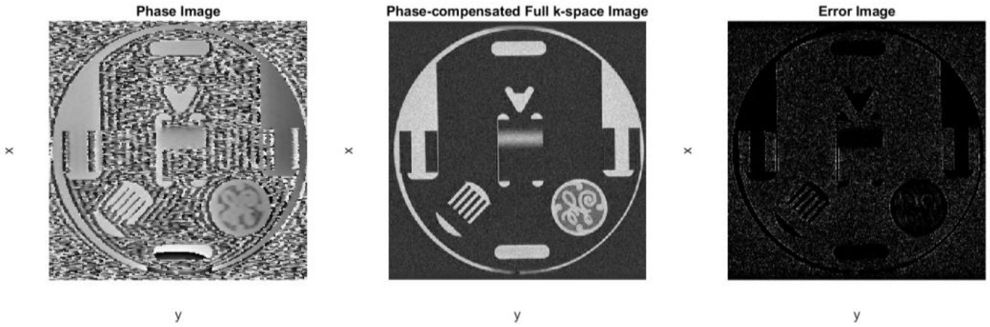

Figure 3. Phase-compensated Full k-space Image Computed with ±1/8th of k-space data

3. MATLAB Code

```
%phase-compensated full k-space image
%with ±1/8th of k-space data
load('ge_phantom.mat');
mf = ifft2c(kdata);
ref = abs(mf);

%use ±1/8th of k-space data, which means
%use the central 64 phase encode lines
ratio_symmetric = 1/8;
[Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);
%phase-compensated full-kspace image
m = real(phase_correct(mf,ms));
%phase image
p = exp(1i*angle(ms));
%error image
```

```
    error_image = ref-m;

    %normalize the reference image
    ref = ref./(max(max(ref)));
    %normalize the result image
    m = m./(max(max(m)));
    %IQA results
    PSNR = psnr(m,ref);
    SSIM = ssim(m,ref);

    disp(strcat('PSNR:',num2str(PSNR)));
    disp(strcat('SSIM:',num2str(SSIM)));
```

<u>symmetric_kspace function</u>

```
function [Ms,ms] = symmetric_kspace(kdata,ratio_symmetric)
  [Nro Npe] = size(kdata);
  %create the window to have the symmetric part of k-space data
  filter_Ms = zeros(1,Npe);
  filter_Ms((Npe/2-Npe*ratio_symmetric):(Npe/2+Npe*ratio_symmetric-1))= 1;
  %multiplication of the k-space data with the window function
  Ms = kdata.*filter_Ms;
  ms = ifft2c(Ms);
end
```

<u>phase_correct function</u>

```
function m = phase_correct (mpk,ms)
  %phase of ms(x,y)
  p = exp(j*angle(ms));
  %phase compensated full k-space image
  p_conjugate = exp(-j*angle(ms));
  m = mpk.*p_conjugate;
end
```

Question 3 - Phase-compensated Full k-space Image with ±1/16th of k-space data

PSNR and SSIM values decreased. The difference between the reference image and the result image increased, so that SSIM decreased. The decrease in PSNR might be explained by loss of information. The resolution got worse, hence PSNR value decreased. The loss of information can be seen from phase image. We compensated the phase of our partial k-space data with less symmetric partial k-space data, which resulted in loss of information.

It is seen from the error image that the errors increased compared to phase-compensated full k-space image with ±1/16th of k-space data, as well.

We can state phase-compensation technique works better with larger part of symmetric k-space data. Otherwise, we lose information in terms of phase and image are affected negatively.

1. Display Results

1.3

Phase-compensated Full k-space Image (±1/16th of k-space data)

PSNR:23.8932

SSIM:0.48175

2. Plots

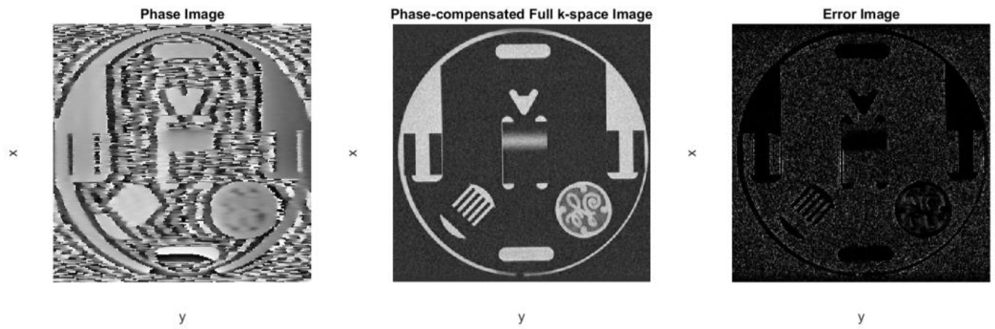

Figure 4. Phase-compensated Full k-space Image Computed with ±1/16th of k-space data

3. MATLAB Code

```
%phase-compensated full k-space image
%with ±1/16th of k-space data
load('ge_phantom.mat');
mf = ifft2c(kdata);
ref = abs(mf);

%use ±1/16th of k-space data, which means
%use the central 32 phase encode lines
ratio_symmetric = 1/16;
[Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);
%phase-compensated full-kspace image
m = real(phase_correct(mf,ms));
%phase image
p = exp(1i*angle(ms));
%error image
error_image = ref-m;

%normalize the reference image
ref = ref./(max(max(ref)));
%normalize the result image
m = m./(max(max(m)));
```

```
%IQA results
PSNR = psnr(m,ref);
SSIM = ssim(m,ref);
```

Question 4 – Partial k-space Image

1. Plots

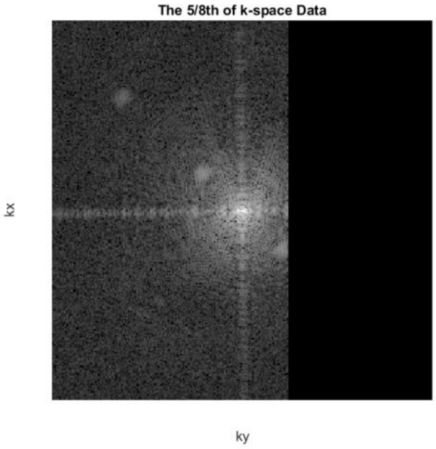

Figure 5. The 5/8th Partial k-space Data

2. MATLAB Code

```
%partial k-space data
%with 5/8th of k-space data
load('ge_phantom.mat');
ratio_partial = 5/8;
[Mpk,mpk] = partial_kspace(kdata,ratio_partial);
```

<u>partial_kspace function</u>

```
function [Mpk,mpk] = partial_kspace(kdata,ratio_partial)
  [Nro Npe] = size(kdata);
  %create the filter to have partial k-space data
  %with the given ration
  filter_Mpk = zeros(1,Npe);
  filter_Mpk(1:Npe*ratio_partial-1)= 1;
  %multiplication of the k-space data with the filter
  Mpk = kdata.*filter_Mpk;
  mpk = ifft2c(Mpk);
end
```

Question 5 – Trivial Reconstruction

We have considerable blurring in y-direction in our result image. The reason is that we multiplied our k-space data with a weighting function to have zero fills. This multiplication with weighting function means convolution in image domain. The inverse Fourier transform of the weighting function is impulse response. The impulse response causes blurring in y-direction.

The blurring in y-direction can been from the error image. At the edges in y-direction, the error increases.

SSIM is higher than other reconstruction results, which means it performs better in terms of SSIM. But, it does not mean the visual result is the best of four reconstruction techniques.

1. Display Results

1.5

Trivial Reconstruction

PSNR:25.4582

SSIM:0.64226

2. Plots

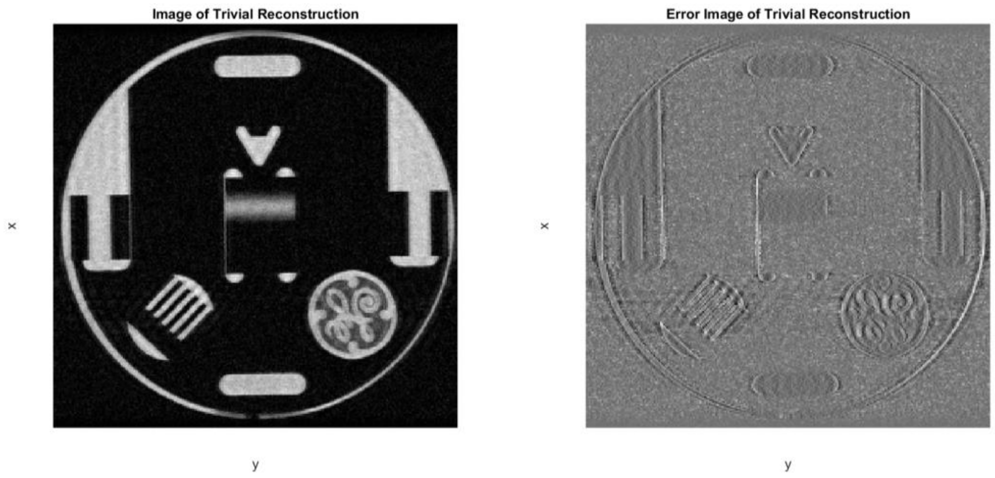

Figure 6. Trivial Reconstruction

3. MATLAB Code

```
%trivial reconstruction
 %with 5/8th of k-space data
```

```
ratio_partial = 5/8;
ratio_symmetric = 1/8;
[M,m,error_image,PSNR,SSIM] = trivial(ratio_partial,ratio_symmetric);
```

trivial function

```
function [M,m,error_image,PSNR,SSIM] = trivial(ratio_partial,ratio_symmetric)
    load('ge_phantom.mat');
    %reference image
    mf = ifft2c(kdata);
    ref = abs(mf);

    %partial k-space data with the desired ratio
    [Mpk,mpk] = partial_kspace(kdata,ratio_partial);
    %the symmetric part of the k-space data
    [Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);

    %phase corrected partial image data
    m = phase_correct(mpk,ms);
    %phase corrected partial k-space data
    M = fft2c(m);
    %get rid of the added parts to the end of k-space data
    %by taking partial k-space data with the desired ratio
    [M,m] = partial_kspace(M,ratio_partial);

    %trivial reconstruction image
    %take real part of image as result image
    m = real(m);
    %error image
    error_image = ref - m;

    %normalize the reference image
    ref = ref./(max(max(ref)));
    %normalize the result image
    m = m./(max(max(m)));
    %IQA results
    PSNR = psnr(m,ref);
    SSIM = ssim(m,ref);
end
```

Question 6 – PCCS Reconstruction

As a remedy for trivial reconstruction, PCCS (phase correction and conjugate synthesis) reconstruction is proposed. We will phase correct the partial k-space data, and then fill the zeros with its conjugate symmetric elements, because we assume the image is real-valued function. However, this is not true for MRI images and application of PCCS means the loss of imaginary part of image, which is shown in Figure 1. The blurring increases where the imaginary part of the image is higher.

Similar to trivial reconstruction, we multiplied the k-space data with a weighting function whose inverse Fourier transform is impulse response. At the edges of error image, the blurring in y-direction can be distinguished.

I was expecting PCCS to perform better, but the visual results are worse because of the loss of imaginary part of the image. The change in k-space spectrum can be seen by comparing with the original k-space spectrum, as well.

PSNR performed better, but SSIM performed worse with PCCS reconstruction. The change of PSNR can be explained by the fact that phase corrected image got rid of its imaginary part in image domain due to application of conjugate symmetry, therefore its contribution into total signal decreased. The change in SSIM is due to the blurring especially at the right-bottom shape.

1. Display Results

1.6

PCCS (phase corrected and conjugate synthesis)

PSNR:23.0299

SSIM:0.54327

2. Plots

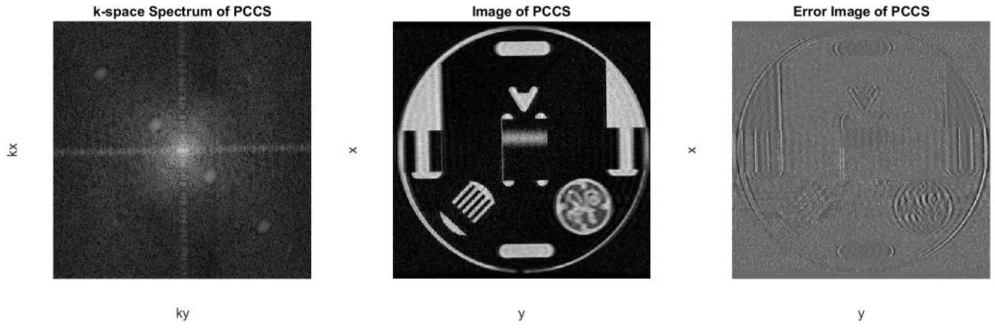

Figure 7. PCCS

3. MATLAB Code

```
%PCCS (phase corrected and conjugate synthesis)
%with 5/8th of k-space data
ratio_partial = 5/8;
ratio_symmetric = 1/8;
[M,m,error_image,PSNR,SSIM] = pccs(ratio_partial,ratio_symmetric);
```

pccs function

```
function [M,m,error_image,PSNR,SSIM] = pccs(ratio_partial,ratio_symmetric)
    load('ge_phantom.mat');
    %reference image
    mf = ifft2c(kdata);
    ref = abs(mf);

    %partial k-space data with the desired ratio
    [Mpk,mpk] = partial_kspace(kdata,ratio_partial);
    %the symmetric part of the k-space data
    [Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);

    %phase corrected partial image data
    phase_corrected_image = phase_correct(mpk,ms);
    %phase corrected partial k-space data
    phase_corrected_kdata= fft2c(phase_corrected_image);
    %enforce conjugate symmetry to the phase corrected partial k-space data
    M = conjugate_symmetry(phase_corrected_kdata,ratio_partial);

    %PCCS result image
    %take real part of the image as result image
    m = real(ifft2c(M));
    %error image
    error_image = ref-m;

    %normalize the reference image
    ref = ref./(max(max(ref)));
    %normalize the result image
    m = m./(max(max(m)));
    %IQA results
    PSNR = psnr(m,ref);
    SSIM = ssim(m,ref);
end
```

Question 7 – Homodyne Reconstruction

Homodyne reconstruction is applied with the 0-1-2 step function. We can decompose the 0-1-2 step function to its symmetric and asymmetric parts. The symmetric part is for the real part of the image. The anti-symmetric part is for the imaginary part of the image. By the multiplication of asymmetric part with the partial k-space data, we can state the imaginary component is suppressed.

We know that homodyne reconstruction will cause ringing artifacts due to discontinuities of pre-weighting function. We will see blurring due to ringing artifacts. The ringing artifacts cannot be distinguished from the result image, but it can be distinguished from the error image. It appears like sweeping the shapes on y-direction.

Still, the visual results are better than PCCS and trivial reconstruction. The image on the right-bottom of the result image is clearer and has less blurring. Performance of PSNR increased compared to PCCS. The performance of SSIM is the lowest in four reconstruction techniques.

1. Display Results

1.7

Homodyne Reconstruction (by using 0-1-2 step function)

PSNR:23.2972

SSIM:0.5311

2. Plots

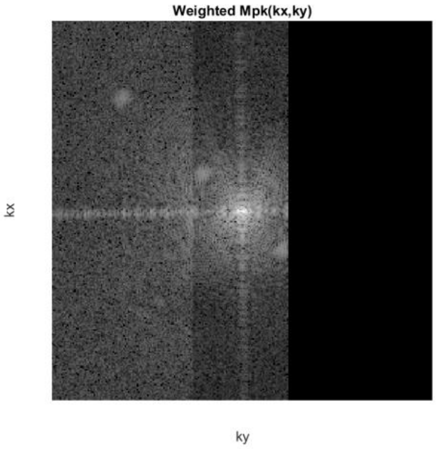

Figure 8. Mpk(kx,ky) after Multiplied by Pre-weighting Function W(ky)

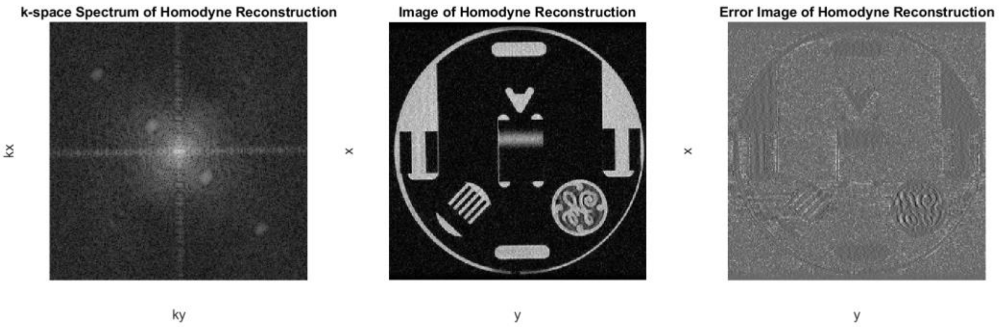

Figure 9. Homodyne Reconstruction

3. MATLAB Code

```
    %homodyne reconstruction by using the 0-1-2 step function
    %with 5/8th of k-space data
    ratio_partial = 5/8;
    ratio_symmetric = 1/8;
    [M,m,error_image,weighted_Mpk,PSNR,SSIM] =
homodyne(ratio_partial,ratio_symmetric);
    load('ge_phantom.mat');
    %reference image
    mf = ifft2c(kdata);
    ref = abs(mf);

    %partial k-space data with the desired ratio
    [Mpk,mpk] = partial_kspace(kdata,ratio_partial);
    %the symmetric part of the k-space data
    [Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);

    %multiplication of the partial k-space data with pre-weighting function
    weighted_Mpk = weightFunc(Mpk,ratio_symmetric);
    weighted_mpk = ifft2c(weighted_Mpk);
    %phase corrected image
    phase_corrected_image = phase_correct(weighted_mpk,ms);

    %homodyne reconstruction image
    %take real part of the image as result image
    m = real(phase_corrected_image);
    %k-space spectrum of the result image
    M = fft2c(m);
    %error image
    error_image = ref - m;

    %normalize the reference image
    ref = ref./(max(max(ref)));
    %normalize the result image
    m = m./(max(max(m)));
    %IQA results
    PSNR = psnr(m,ref);
    SSIM = ssim(m,ref);
end
```

<u>weightFunc function</u>

```
function weighted_Mpk = weightFunc(Mpk,ratio_symmetric)
    [Nro Npe] = size(Mpk);
    %create the weighting function
    %the weighting function is the 0-1-2 step function
    W = zeros(1,Npe);
    W(1:(Npe/2-Npe*ratio_symmetric-1)) = 2;
    W((Npe/2-Npe*ratio_symmetric):(Npe/2+Npe*ratio_symmetric-1)) = 1;
    %multiplication of the partial k-space data with the weighting function
```

```
    weighted_Mpk = Mpk.*W;
end
```

Question 8 – POCS Reconstruction

We had errors in PCCS due to applying phase correction, and then conjugate synthesis. In comparison, we had errors in homodyne reconstruction due to applying conjugate synthesis, and then phase correction. POCS reconstruction proposes the idea of applying both of them iteratively, so that the difference between their results will converge to zero by going on the iterations.

The error gets lower as iterations continue. It is hard to see the difference between each consecutive iteration, but the difference is distinguishable between the first iteration and the end result. The error at the edges on y-direction decreased considerably. Also, the improvement in each iteration can be seen from the right-bottom shape on the image. It is getting clearer as the iterations continue.

Performance of PSNR decreased compared to homodyne reconstruction, but performance of SSIM increased. In the first iteration, the PSNR and SSIM values increases, but as the iterations continue, the values of PSNR and SSIM decreases. This means it performs better in terms of PSNR, and worse in terms of SSIM as iterations continue. Nevertheless, the visual results get better. This means PSNR and SSIM are sometimes not sufficient for Image Quality Assessment of images.

1. Display Results

1.8

POCS Reconstruction

PSNR #1:25.6393, SSIM #1:0.64815

PSNR #2:25.9769, SSIM #2:0.65767

PSNR #3:25.8725, SSIM #3:0.65037

PSNR #4:25.7334, SSIM #4:0.64448

PSNR #5:25.6297, SSIM #5:0.64064

PSNR:25.6297

SSIM:0.64064

2. Plots

The k-spectrum in each iteration is the k-spectrum data used in the ith iteration to fill the partial k-space data. The k-spectrum of the last image is the k-space data of the result image.

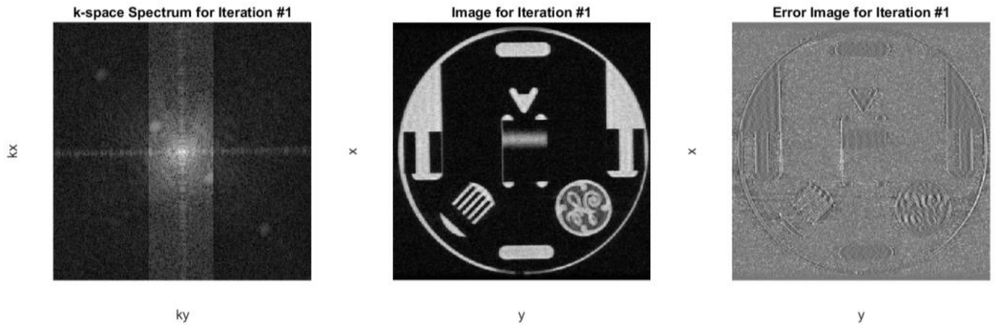

Figure 10. POCS Iteration #1

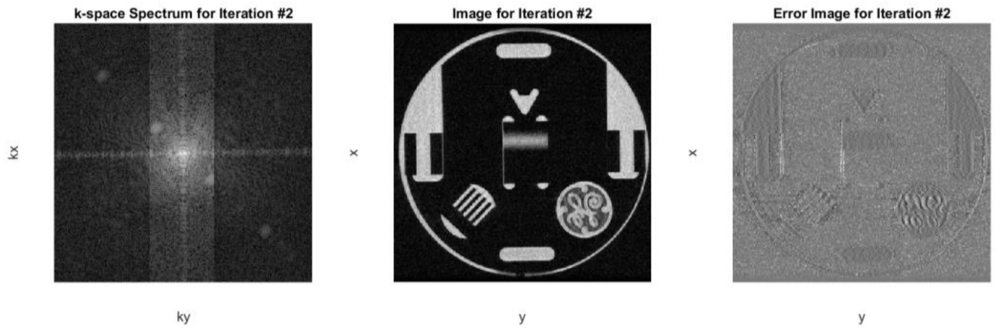

Figure 11. POCS Iteration #2

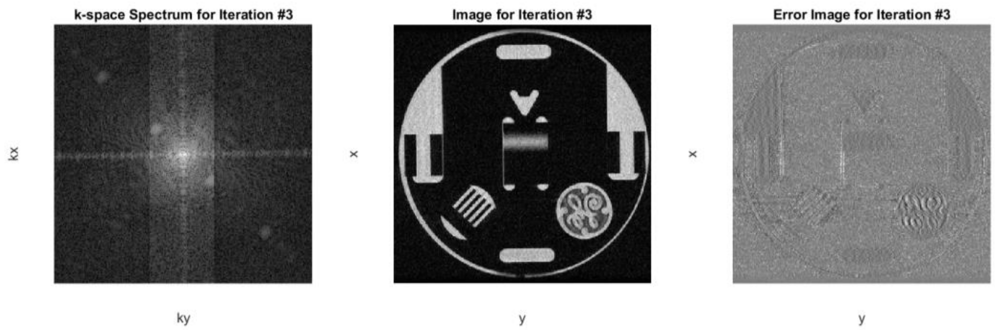

Figure 12. POCS Iteration #3

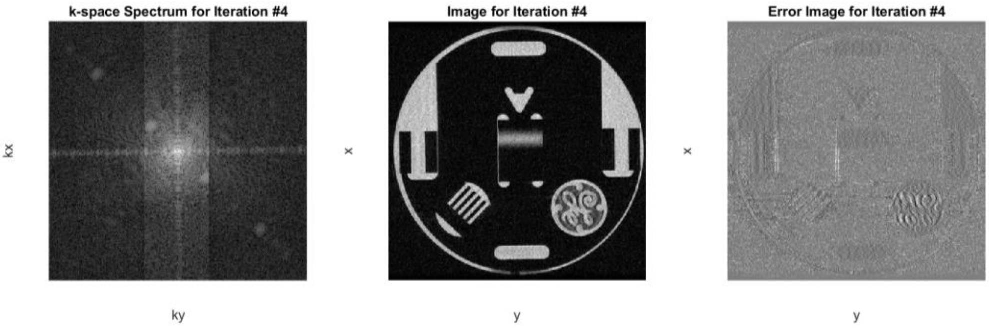

Figure 13. POCS Iteration #4

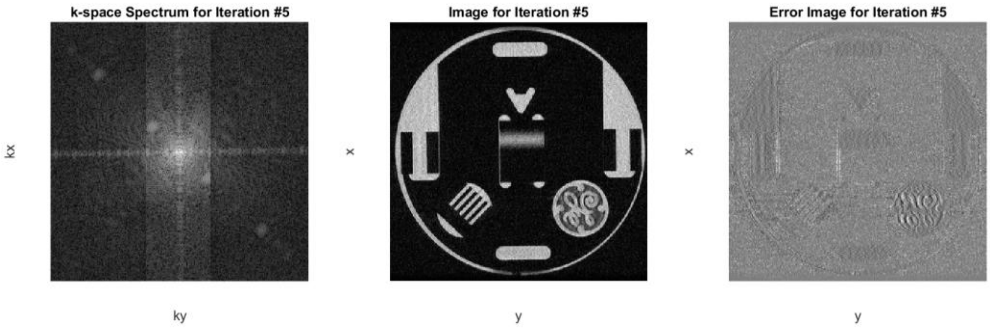

Figure 14. POCS Iteration #5

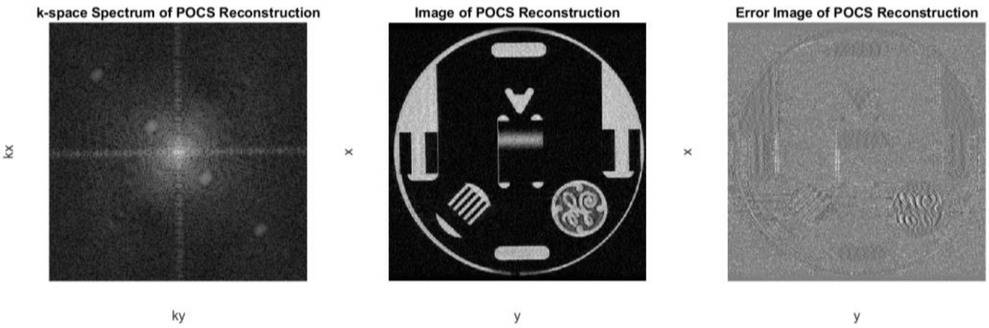

Figure 15. POCS End Results (Iteration#5)

3. MATLAB Code

```
%POCS reconstruction
%with 5/8th of k-space data
disp('1.8');
ratio_partial = 5/8;
ratio_symmetric= 1/8;
[M,m,error_image,PSNR,SSIM] = pocs(ratio_partial,ratio_symmetric,true);
```

pocs function

```
function [M,m,error_image,PSNR,SSIM] = pocs(ratio_partial,ratio_symmetric,display_iter)
    load('ge_phantom.mat');
    %reference image
    mf = ifft2c(kdata);
    ref = abs(mf);

    %partial k-space data with the desired ratio
    [Mpk,mpk] = partial_kspace(kdata,ratio_partial);
    %the symmetric part of the k-space data
    [Ms,ms] = symmetric_kspace(kdata,ratio_symmetric);
    %initial Mi(kx,ky) is Mpk(kx,ky)
    %later, the zeros will be filled with the phase constrained image data
    Mi = Mpk;

    for iter = 1:5
        if(iter >1)
          %data constrained k-space data  Mi(kx,ky)
          Mi = replace_kspace(Mpk,M,ratio_partial);
        end

        %phase of ms(x,y)
        p = exp(1i*angle(ms));
        %phase constrained image data
        mi = ifft2c(Mi);
        m = abs(mi).*p;
        %go back to k-space to use in the next iteration
        M = fft2c(m);

        if (display_iter)
            figure;
            set(gcf, 'WindowState', 'maximized');
            %display the resulting k-space spectrum
            subplot(1,3,1);
            imshow(log(abs(M)+1),[]);
            title(strcat('k-space Spectrum for Iteration # ',num2str(iter)));
            xlabel('ky'); ylabel('kx');

            %display the magnitude image
            subplot(1,3,2);
            imshow(abs(m),[]);
            title(strcat('Image for Iteration # ',num2str(iter)));
            xlabel('y'); ylabel('x');

            %error image
            error_image = ref - real(mi.*exp(-j*angle(ms)));
            %display the error image
            subplot(1,3,3);
            imshow(error_image,[]);
```

```
        title(strcat('Error Image for Iteration # ',num2str(iter)));
        xlabel('y'); ylabel('x');
        file = strcat('1.8_iteration',num2str(iter),'.png');
        saveas(gcf,file);

        %normalize the reference image
        ref_IQA = ref./(max(max(ref)));
        %normalize the result image
        m_iter = real(mi.*exp(-1i*angle(ms)));
        m_iter = m_iter./(max(max(m_iter)));
        %IQA results
        PSNR = psnr(m_iter,ref_IQA);
        SSIM = ssim(m_iter,ref_IQA);
        disp(strcat('PSNR #',num2str(iter),':',num2str(PSNR),...
            ', SSIM #',num2str(iter),':',num2str(SSIM)));
    end

    %POCS reconstruction image
    m = real(mi.*exp(-j*angle(ms)));
    %k-space spectrum of the result image
    M = fft2c(m);
    %error image
    error_image = ref - m;

    %normalize the reference image
    ref = ref./(max(max(ref)));
    %normalize the result image
    m = m./(max(max(m)));
    %IQA results
    PSNR = psnr(m,ref);
    SSIM = ssim(m,ref);

end
```

Question 9 - Comparison at 9/16th k-space data

The visual results get worse when the used part of partial k-space data decreases. Trivial reconstruction and PCCS are affected worse than homodyne reconstruction and POCS. The best visual result is still POCS's result image. This can also be seen from error images.

PSNR performed better, but it does not give valuable information about the visual results. Only looking at PSNR results might be misleading. PCCS has the lowest PSNR, but it has the worst visual result image in our case.

SSIM performance for all of the reconstruction techniques is affected negatively. This can also be estimated from the increase of error in error images.

1. Display Results

1.9

Comparison of 4 Reconstruction Techniques with 9/16th of k-space Data

PSNR Results

PSNR of Trivial Reconstruction:22.6901

PSNR of PCCS Reconstruction:20.1

PSNR of Homodyne Reconstruction:21.1605

PSNR of POCS Reconstruction:22.8752

SSIM Results

SSIM of Trivial Reconstruction:0.51023

SSIM of PCCS Reconstruction:0.40163

SSIM of Homodyne Reconstruction:0.4013

SSIM of POCS Reconstruction:0.52884

2. Plots

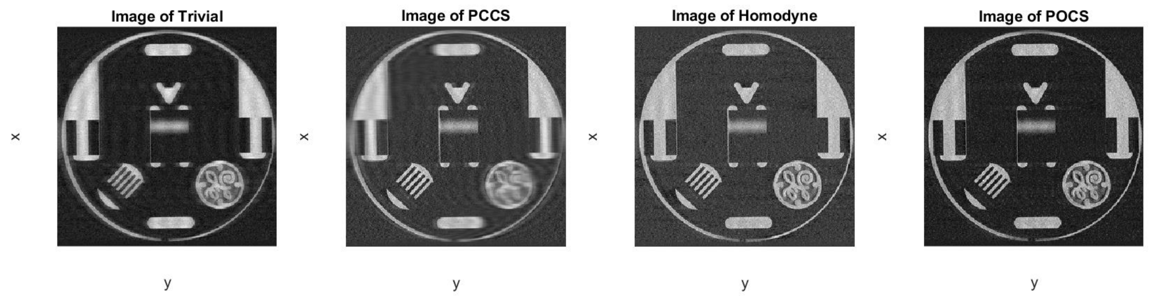

Figure 16. The Comparison of Images of the Applied Reconstruction Techniques

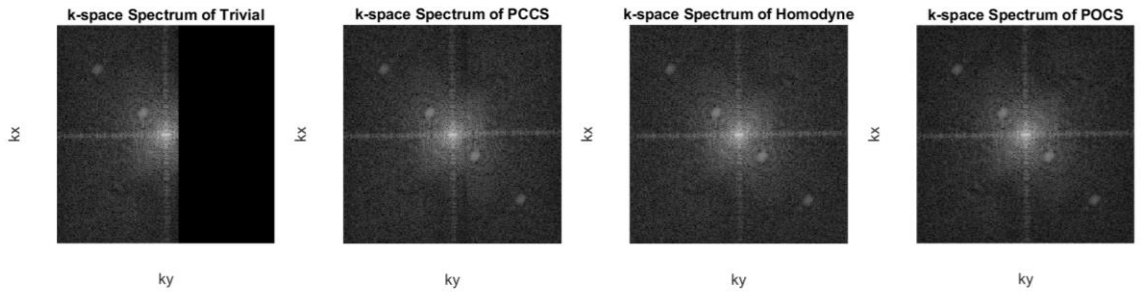

Figure 17. The Comparison of k-space Spectrums of the Applied Reconstruction Techniques

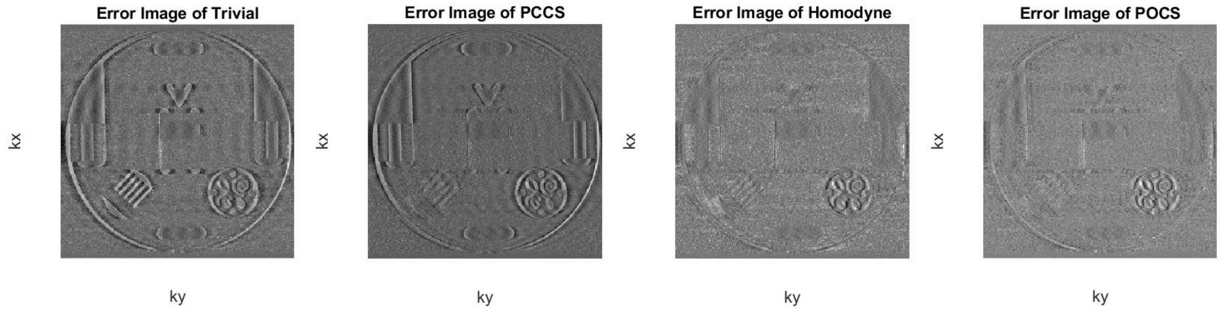

Figure 18. The Comparison of Error Images of the Applied Reconstruction Techniques

3. MATLAB Code

```
ratio_partial = 9/16;
ratio_symmetric= 1/16;
[M_trivial,m_trivial,error_trivial,PSNR_tri,SSIM_tri] =
trivial(ratio_partial,ratio_symmetric);
[M_pccs,m_pccs,error_pccs,PSNR_pccs,SSIM_pccs] =
pccs(ratio_partial,ratio_symmetric);
[M_homodyne,m_homodyne,error_homodyne,weighted_Mpk,PSNR_homo,SSIM_homo]
= homodyne(ratio_partial,ratio_symmetric);
[M_pocs,m_pocs,error_pocs,PSNR_pocs,SSIM_pocs]=
pocs(ratio_partial,ratio_symmetric,false);
```

Question 10 - Comparison at 17/32th k-space data

Again, PSNR performed better, but it does not give valuable information about the visual results. The results for PSNR between the reconstruction techniques are similar to comparison at 9/16th k-space data.

SSIM performance for all of four reconstruction techniques is worse than comparison at 9/16th k-space data. This can also be estimated from the increase of error in error images.

Trivial reconstruction and PCCS is the worst now. Their images cannot be acceptable. The resolution of homodyne might be still acceptable, but we lost information in terms of contrast. We can continue using it by improving its contrast. POCS performed the best, it might be still acceptable for our usage. However, there are ghosting in the images of homodyne reconstruction and POCS. If our purpose for images are affected badly with the ghosting effects, the images of homodyne and POCS cannot be acceptable like trivial reconstruction and PCCS.

1. Display Results

1.10

Comparison of 4 Reconstruction Techniques with 17/32th of k-space Data

PSNR Results

PSNR of Trivial Reconstruction:20.1303

PSNR of PCCS Reconstruction:17.8777

PSNR of Homodyne Reconstruction:19.2059

PSNR of POCS Reconstruction:20.4091

SSIM Results

SSIM of Trivial Reconstruction:0.42774

SSIM of PCCS Reconstruction:0.31254

SSIM of Homodyne Reconstruction:0.33209

SSIM of POCS Reconstruction:0.44854

2.Plots

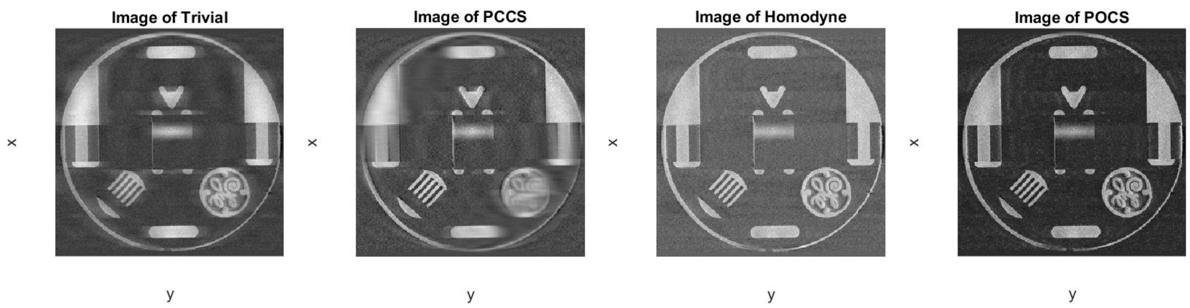

Figure 19. The Comparison of Images of the Applied Reconstruction Techniques

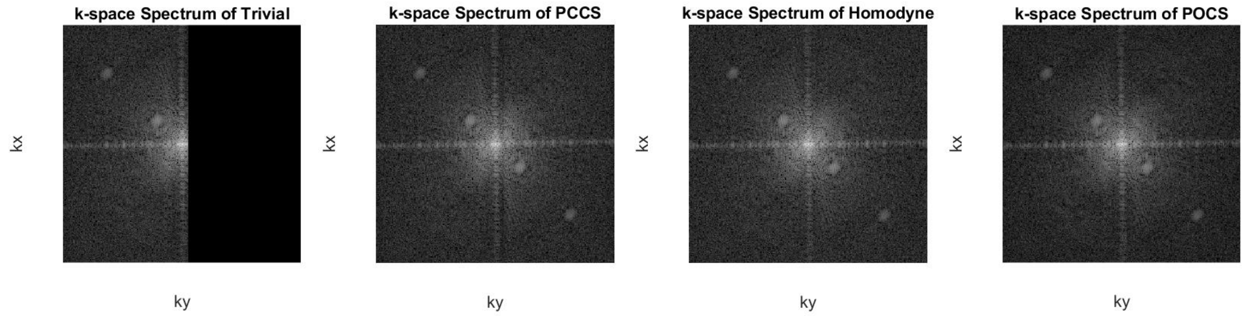

Figure 20. The Comparison of k-space Spectrums of the Applied Reconstruction Techniques

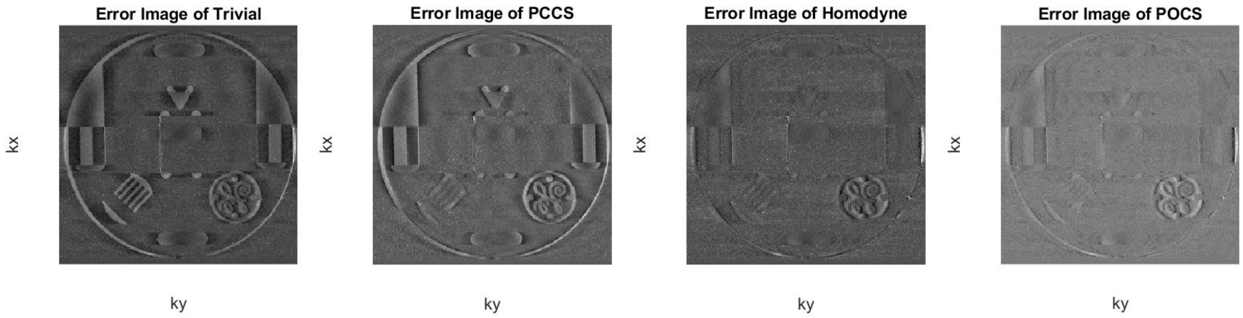

Figure 21. The Comparison of Error Images of the Applied Reconstruction Techniques

3. MATLAB Code

```
    ratio_partial = 17/32;
    ratio_symmetric= 1/32;
    [M_trivial,m_trivial,error_trivial,PSNR_tri,SSIM_tri] =
trivial(ratio_partial,ratio_symmetric);
    [M_pccs,m_pccs,error_pccs,PSNR_pccs,SSIM_pccs] =
pccs(ratio_partial,ratio_symmetric);

[M_homodyne,m_homodyne,error_homodyne,weighted_Mpk,PSNR_homo,SSIM_homo] =
homodyne(ratio_partial,ratio_symmetric);
    [M_pocs,m_pocs,error_pocs,PSNR_pocs,SSIM_pocs]=
pocs(ratio_partial,ratio_symmetric,false);
```

Question 11

We saw phase-compensation reconstruction technique works better as partial k-space data increases. We saw POCS has always better performance in terms of visual results and it could resist more when using less partial k-space data compared to other techniques even though there were some ghosting with 17/32th partial k-space data.

PSNR gives good information when we use it as an IQA method for one specific reconstruction technique. It gives valuable information about the loss of information by using less partial k-space data. However, it did not give any information about the comparison of reconstruction techniques with the same partial k-space data. It might be misleading to compare different reconstruction techniques with PSNR values.

SSIM gives better information because it is more direct comparison method of reference image and result image. It is more suitable for human visual abilities to evaluate images. We could detect the improvement from homodyne reconstruction to POCS by SSIM. But the values of SSIM were higher for trivial reconstruction and PCCS reconstruction, even though they had worse visual results. Similar to PSNR, SSIM gives valuable information for comparison of the same reconstruction technique with different partial k-space data, but it might be misleading to compare different reconstruction techniques with SSIM.