## Engineering Mathematics Project Report

**Instructor: Prof. Karbalaei**


## Sharif University of Technology


# Magnetic Resonance Imaging (MRI) Reconstruction from K-Space


By Reza Nayeb Habib
Student ID# 401102694

# Contents

# 1 Introduction

# 2 MRI and K-Space

## 2.1 Question 1

**Explain the concept of 2D Fourier transform. What are the bases of the 2D Fourier transform?**

In fourier transform we had:

$$F(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-j2\pi\omega x}\, dx$$

now using the same idea in a 2D space, 2D fourier transform is defined as:

$$F(u, v) = \int_{-\infty}^{+\infty} f(x, y)e^{-j2\pi(ux+vy)}\, dxdy$$

the basis of 2D fourier is u (or in K-Space $K_x$) which is the frequency in x axis, and v (or in K-Space $K_y$) which is the frequency in y axis.

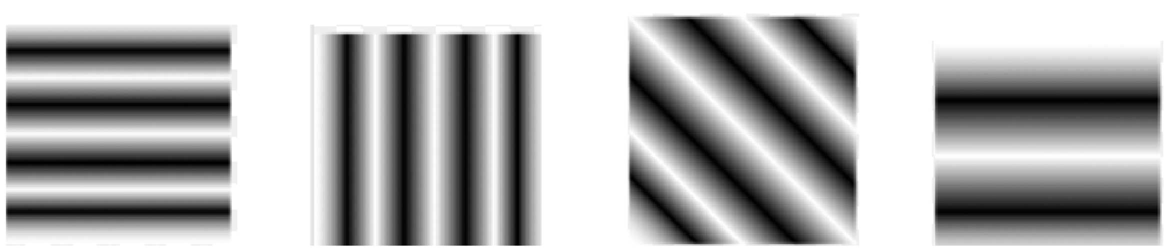
## 2.2 Question 2

**What is the connection between each point in k-space and raw image? What does each point in k-space represent?**

the value of each point $(K_x, K_y)$ in K-Space represents the intensity of that frequency in the raw image.

## 2.3 Question 3

**Consider the images below:**



**(a)** Image 1 **(b)** Image 2 **(c)** Image 3 **(d)** Image 4

**We transform each image into k-space, each point in the plot below represents one of the images in the k-space. For each picture, select the correct point in the k-space.**

we showed the corresponding image to the corresponding points in figure 2.3

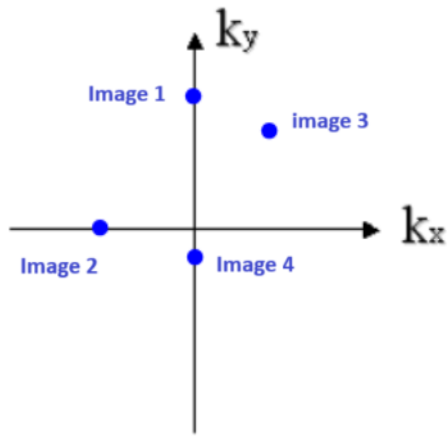
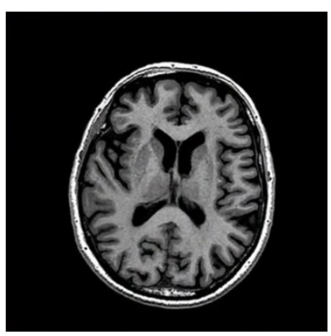



Figure 1: Images on their corresponding points on K-Space

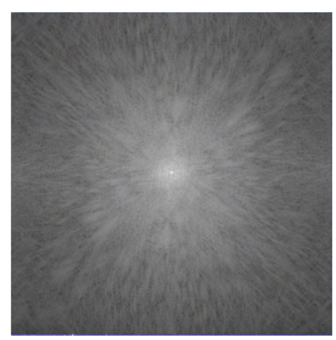
we can see that Image 1 looks like the function $\sin(y)$ and both Images 2 and 4 look like $\sin(x)$ and the third image is like $\sin(x+y)$ function.

### 2.4 Question 4

**Consider the MRI image and its k-space transformed image below:**

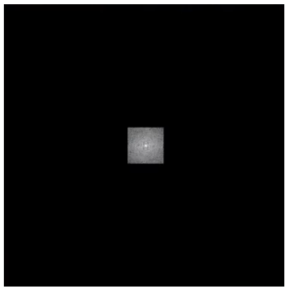


**(a)** MRI image

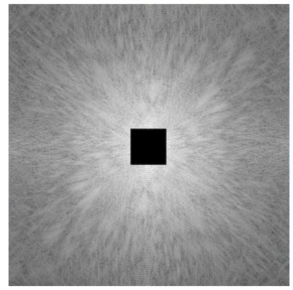


**(b)** k-space representation

**What happens if we delete high-frequency bases in k-space? what about low-frequency bases? which one would preserve the original image more?**



**(a)** k-space representation when we delete high-frequency elements



**(b)** k-space representation when we delete low-frequency elements

it can be seen that the low-frequency bases in K-Space have more intensity in their color this means they store more data than high-frequency bases thus we can conclude that removing high-frequency bases preserves the image better than removing low-frequency bases.

# 3 MRI Reconstruction

## 3.1 Question 1

**For the purpose of this project, search and explain two of the above-mentioned methods for MRI Reconstruction from k-space data.**

**FBP:** in this method we backpropagate the pixel from it's nearest neighbors on the sensor and from the direction the wave source emmits its waves to our detector. FBP (Filtered Back Projection) reconstruction method helps create 2D or 3D images from MRI data. This process involves two key steps. First, the acquired data goes through a filter. This filter emphasizes certain details while reducing noise. Then, the filtered data is back-projected to form the image. The back projection spreads the filtered data back into the image space, revealing the structure of the imaged object. This method is often used in MRI, especially in cases where the data is collected in a radial or spiral trajectory. Though widely used, FBP has limitations, particularly in complex MRI acquisition scenarios. In those cases, iterative reconstruction methods such as SENSE and GRAPPA are often preferred for better image quality.

**Parallel imaging**: Parallel imaging is a technique in MRI that uses multiple receiver coils to accelerate image acquisition. By leveraging information from multiple coils, parallel imaging reduces the amount of data that needs to be acquired, thereby shortening scan times or improving image resolution. Two popular parallel imaging techniques are SENSE (Sensitivity Encoding) and GRAPPA (Generalized Autocalibrating Partial Parallel Acquisitions).

## 3.2 Question 2

**Load the rawkneedata.mat that is attached to this file.**

```
% lodaing data from rawkneedata.mat
clc; clear;
mat = load('rawkneedata.mat');
[ans, dat] = deal(mat.ans, mat.dat);
```

### 3.3 Question 3

**Reconstruct the MRI image using a 2D-inverse Fourier transform. Explain your method.**

```
% picture reconstruction using ifft2
raw_ifft = ifft2(dat);
knee_image = fftshift(raw_ifft);
knee_image = abs(knee_image);
% for removing the phase likely created by
% noise(because the picture should only be real)
imshow(abs(knee_image), []);
```

we first do a 2D inverse fourier transform on the raw data with MATLAB ifft2() function, then we shift the image using fftshift to get the correct inverse fourier and finally we show it with imshow.

### 3.4 Question 4

**Plot the reconstructed MRI.**

Figure 2: Reconstructed MRI picture using ifft2

### 3.5 Question 5

**Can you observe the noises available in your reconstructed MRI? Discuss briefly the noises in medical images, and in particular MRI images..**

the noise in the picture can be seen especially the white spots in the darker areas.
Generally it is common for MRI images to contain noise due to the method they are created(emmiting waves wich can cause us receiving unwanted signals from the emitter device). also detectors can have

inaccuracies and the environment can also cause some noise, thus noise reduction is important for obtaining useful medical images that can be used for treating medical conditions.

# 4 Metrics

## 4.1 Histogram

**Use the matplotlib.pyplot.hist in Python on imhist in MATLAB commands to display the histograms of the noisy and noiseless images. Explain the characteristics of the filter used to reduce noise in the frequency domain based on the two histograms obtained. Is it a high-pass, middle-pass, or low-pass filter?**

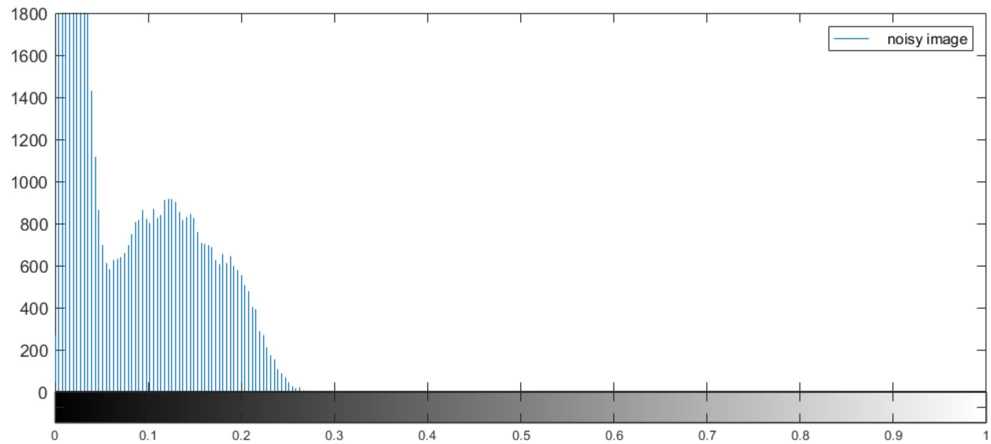



Figure 3: Histogram of the noisy image

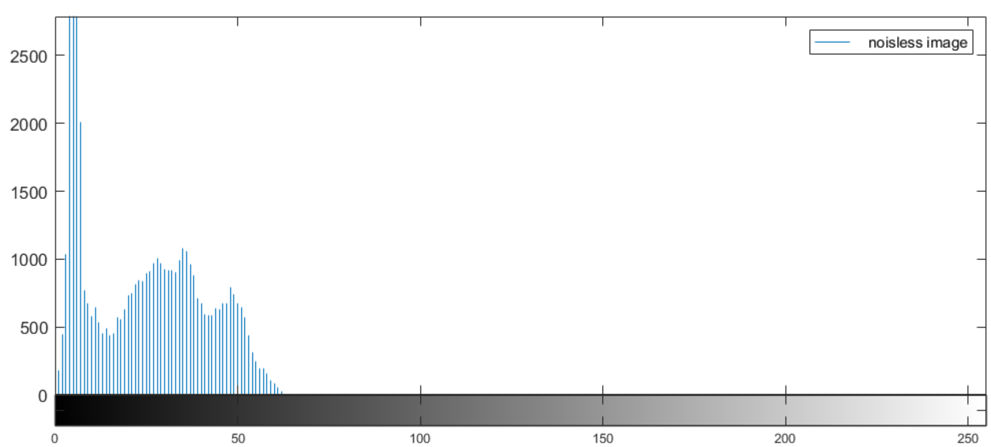



Figure 4: Histogram of the noiseless image

by looking at the histograms we can conclude that the filter used is more like a middle pass on that has reduced both low frequency bases and high frequency ones.

### 4.2 Low Pass Filters

**We want to reduce image noise by applying spatial filters. Read carefully about the Mean, median, and Gaussian filters, and briefly explain each in your report.**

**Mean Filter:** this method convolves a matrix with size nxn with all elements being $\frac{1}{\mathrm{n}}$. this method is technicaly just taking the mean of all of the neighbors(and the pixel itself) of the pixel and replacing the pixel with the mean value thus being names Mean Filter.(See 4.2)

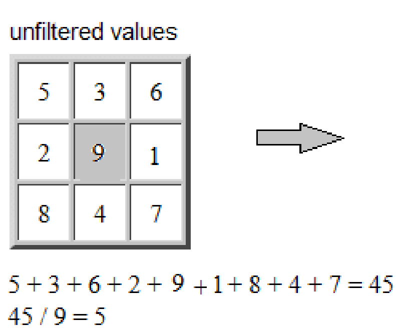



Figure 5: the working of a mean filter

**Median filter:** this method chooses an nxn matrix of the pixels neighborhood and chooses the mean between those pixels(see 4.2)

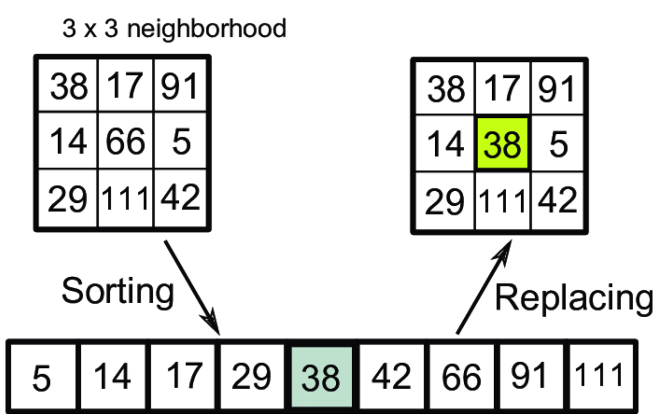



Figure 6: the working of a median filter

**Gaussian Filter:** for applying the Gaussian Filter we first create a kernel matrix in the shape of a gaussian distribution, with formula:

$$G_{2D}(X,Y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{X^2+Y^2}{2\sigma^2}}$$

where X and Y are nxn matrices with shape:

$$X = \begin{bmatrix} -\left[\frac{n}{2}\right] & -\left[\frac{n}{2}\right]+1 & ... & 0 & ... & \left[\frac{n}{2}\right]-1 & \left[\frac{n}{2}\right] \\ ... & ... & ... & 0 & ... & ... & ... \\ -\left[\frac{n}{2}\right] & -\left[\frac{n}{2}\right]+1 & ... & 0 & ... & \left[\frac{n}{2}\right]-1 & \left[\frac{n}{2}\right] \end{bmatrix}$$

$$Y = \begin{bmatrix} -\left[\frac{n}{2}\right] & \ldots & -\left[\frac{n}{2}\right] \\ -\left[\frac{n}{2}\right]+1 & \ldots & -\left[\frac{n}{2}\right]+1 \\ \ldots & \ldots & \ldots \\ 0 & 0 & 0 \\ \ldots & \ldots & \ldots \\ \left[\frac{n}{2}\right]-1 & \ldots & \left[\frac{n}{2}\right]-1 \\ \left[\frac{n}{2}\right] & \ldots & \left[\frac{n}{2}\right] \end{bmatrix}$$

after creating the kernel we convolve it with our image.

### 4.3 Mean Filter

**Write a function that applies a Mean filter to an input image and returns the filtered image. You are not allowed to use any pre-built Mean filter functions.7 (kernel size is up to you.)**

```
% mean filter
function mean = meanflt2(data, n)
    N = ones(n) ./ n;
    mean = conv2(data, N, 'same');
end
```

Figure 7: filtered image using Mean Filter

### 4.4 Median Filter

**Write a function that, upon receiving the kernel size, applies the median filter to the input image and returns the filtered image. (Hint: Use scipy.signal.medfilt in Python or medfilt2 in MATLAB)**

```
% median filter
function median = medianflt2(data, n)
    median = medfilt2(data, [n, n]);
end
```

Figure 8: filtered image using Median Filter

### 4.5 Gaussian Filter

**Write a function that applies a Gaussian filter to an input image and returns the filtered image. You are not allowed to use any pre-built Mean filter functions.6 (kernel size is up to you.)**

```
% gaussian filter
function gfilt = gaussianflt2(data, n, sigma)
    ind = -floor(n/2) : floor(n/2);
    [X Y] = meshgrid(ind, ind)

    %// Create Gaussian Mask
    h = exp(-(X.^2 + Y.^2) / (2*sigma*sigma));
    %// Normalize so that total area (sum of all weights) is 1
    h = h / sum(h(:));
    gfilt = conv2(data, h, "same");
end
```

Figure 9: filtered image using Gaussian Filter

### 4.6 Non-Local Means

**The non-local means (NLM) noise reduction algorithm is well known as an excellent technique for removing noise from a magnetic resonance (MR) image to improve diagnostic accuracy. [5] Research this algorithm and describe its method. Then reduce the noise of the image using the pre-built functions imnlmfilt in MATLAB or skimage.restoration.denoise-nl-means in Python.**

Non-Local Means algorithm gets a weighted sum over all pixels in the image, weighted by how similar two pixels are together, unlike unlike local Mean(Mean Filter discussed in 4.3) which gets the mean of the pixels close to the target pixel. the formula for this weighted sum is as follows.
consider the pixels $p$ and $q$, the set of all pixels in the image shown by $\Omega$ function $u(p)$ which gives the filtered value of point $p$ then we have($v(q)$ is the unfiltered value of $q$):

$$u(p) = \frac{1}{C(p)} \int_{\Omega} v(q) f(p,q) dq$$

where $f(p,q)$ is the weight function denoting the similarity of $p$ and $q$ and $C(p)$ is the normalizing factor. the $C(p)$ normalizing factor is also given by:

$$C(p) = \int_{\Omega} f(p,q) dq$$

unlike the normalizing favtor the weighting functions isn't unique and various formulas can be used for it but one of the commonly used functions is the Gaussian weighting function:

$$f(p,q) = \exp - \frac{|B(q) - B(p)|^2}{h^2}$$

where $h$ is the filtering parameter(the standard deviation) and $B(p)$ is the local mean value of the image point values surrounding $p$.

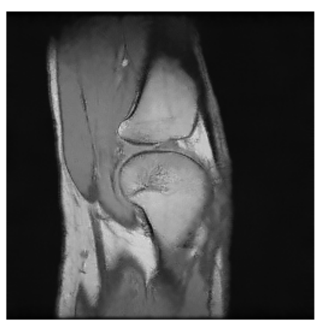


Figure 10: filtered image using Non-Local Mean Filter

### 4.7 Evaluation

**Evaluate the output of the four methods discussed in this section using SNR and PSNR criteria and present the results, compared with kneeMRI.jpeg in a table. Also, calculate these two criteria for the initial noisy image and declare which method performed better in noise reduction.**

```
% SNR Evaluation
function snr = mSNR(I_noisy , K_noisless)
    n = I_noisy.^2;
    num = sum(n, "all");
    d = (I_noisy - K_noisless).^2;
    denum = sum(d, "all");
    snr = 10 * log10(num/denum);
end

% PSNR Evaluation
function psnr = mPSNR(I_noisy , K_noisless)
    [m, n] = size(I_noisy);
    MSE = sum(  (I_noisy - K_noisless).^2  , "all"  ) / (m*n);
    psnr = 10 * log10(    max((I_noisy).^2, [], "all") / MSE);
end
```

by coding the given algorithms in matlab we derived the following table:

| info | noisy | mean | median | gauss | imnlm |
|---|---|---|---|---|---|
| "SNR" | 3.4987 | 3.4987 | 23.986 | 26.53 | 21.953 |
| "PSNR" | 25.842 | 11.412 | 31.985 | 34.48 | 31.243 |

wich shows that the Gaussian and Local-Mean are the best filters and the mean filter is the worst.
by looking at the pictures by eye we can see that although the SNR and PSNR of the Gaussian and Loacal-Mean are close the Local-Mean gives better resoloution so that may be the reason it is prefered for use in MRI denoising.