Lecture 13

Direct Reconstruction Methods

13.1 Lecture Objectives

- Understand the basics of image reconstruction from Fourier data
- Understand the basics of image reconstruction from projection data
- Relate the acquired data and reconstruction to several common artifacts

13.2 Forward Problem

13.2.1 Fourier measurements

In several imaging modalities (most notably MRI), there is a Fourier relationship between measurements and image, eg in the 2D case:

$$d(k_1, k_2) = \int\int x(r_1, r_2) e^{-i2\pi(k_1 r_1 + k_2 r_2)} dr_1\, dr_2 \tag{13.1}$$

where r_1 and r_2 represent the spatial coordinates in image space, $x(r_1, r_2)$ is the corresponding image, k_1 and k_2 represent the Fourier-space coordinates (termed k-space in MRI lingo), and $d(k_1, k_2)$ represents the k-space measurements as a 2D array.

Note that, upon discretization of our image and Fourier spaces, we can represent this relationship using matrix-vector notation, as:

$$\mathbf{d} = \mathbf{F}\mathbf{x} \tag{13.2}$$

where $\mathbf{F}$ is a matrix that performs a 2D DFT (or 3D DFT if performing 3D imaging) on the image that is represented (in vector form) by vector $\mathbf{x}$.

Also, note that we can use our matrix-vector representation above to express the case where we do not obtain all the samples in the 2D DFT but only a subset of these samples, or the case where we obtain samples that are not located on a Cartesian grid (eg: if we are using radial or other non-Cartesian k-space trajectories in our MRI pulse sequence). In some of these cases, we may not be able to directly use FFTs to obtain fast computational implementations of our transform (there are tricks that enable us to still use FFTs, but we will not cover these in depth in this class), however, we should still be able to express it as a linear matrix-vector relationship.

13.2.2 Projections

For tomographic imaging modalities (eg: CT), we can model our data as projections along a certain angle θ. Specifically, for parallel beam CT, we can model our data (after appropriate transformation) as follows:

$$d(s,\theta) = \int\int x(r_1,r_2)\delta(r_1\cos\theta + r_2\sin\theta - s)dr_1\, dr_2 \tag{13.3}$$

as depicted graphically in Figure 13.1.

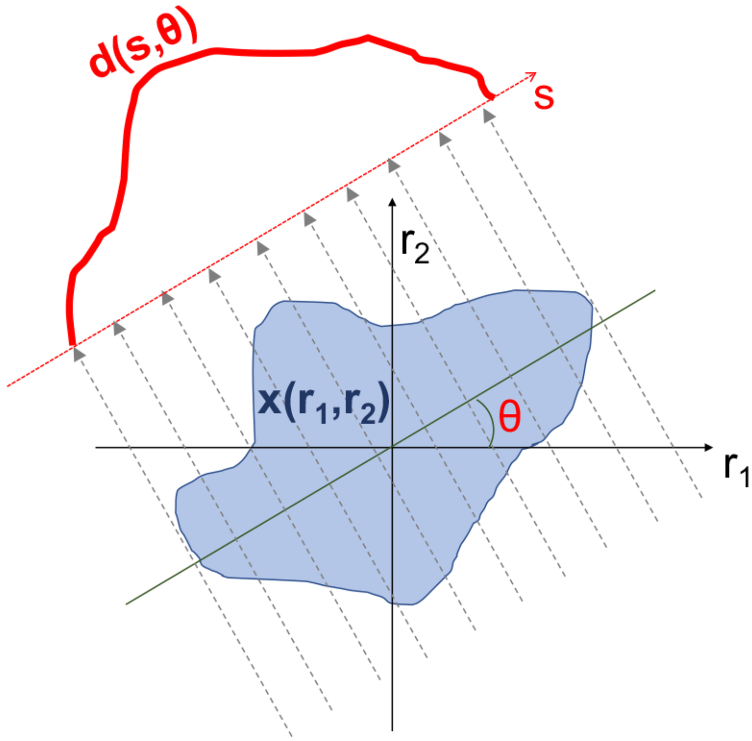

Figure 13.1: Graphical representation of a projection as used in tomographic imaging techniques (eg: CT). If we obtain sufficient projections of our object (image), we are able to reconstruct an accurate depiction of the object.

Similarly to the Fourier imaging example above, upon discretization we can represent this projection relationship using matrix-vector notation, as:

$$\mathbf{d} = \mathbf{P}\mathbf{x} \tag{13.4}$$

where $\mathbf{P}$ is a matrix that performs a set of projections on our discretized image, along a set of different angles.

13.3 Direct Reconstruction

Sometimes we can directly express a solution to the image reconstruction problem using a closed-form operation on the acquired data.

13.3.1 Fourier measurements

For Fourier imaging, if we acquire samples on a Cartesian grid with sufficiently dense sampling, we can simply perform image reconstruction by applying an inverse DFT (in 2D or 3D as appropriate) to the acquired data:

$$\hat{\mathbf{x}} = \mathbf{F}^{-1}\mathbf{d} \tag{13.5}$$

where $\mathbf{F}^{-1}$ represents an inverse DFT matrix (in 2D or 3D, as appropriate).
Point Spread Function of DFT reconstruction: If we acquire an image with field of view (FOV) of size $D \times D$, measured using a Cartesian grid in Fourier space, with $N \times N$ samples, and reconstruct the image using a DFT reconstruction, the point spread function (PSF) will be:

$$\mathrm{PSF}(r_1, r_2) \approx \frac{\sin(\pi N r_1/D)}{\sin(\pi r_1/D)} \frac{\sin(\pi N r_2/D)}{\sin(\pi r_2/D)} \tag{13.6}$$

A 1D depiction of this PSF is shown in figure 13.2. Example images in high and low resolution are shown in figure 13.3, and an example image with clear Gibbs ringing is shown in figure 13.4.

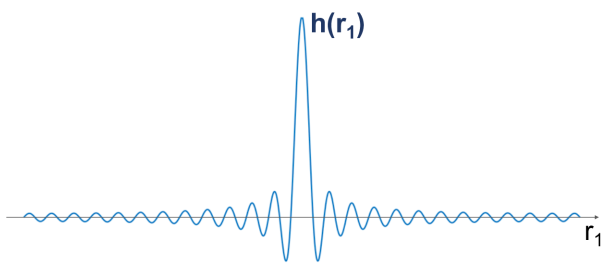

Figure 13.2: Point spread function of DFT reconstruction from Fourier samples. The width of the main lobe is commonly understood as the spatial resolution, and is approximately the size of the field of view divided by the number of equally spaced Fourier samples. Note the substantial secondary lobes, which give rise to so-called Gibbs ringing.

13.3.2 Projection measurements

If we acquire sufficient projections, we can also recover our image using so-called filtered backprojection (FBP). FBP can be understood based on the central slice theorem.

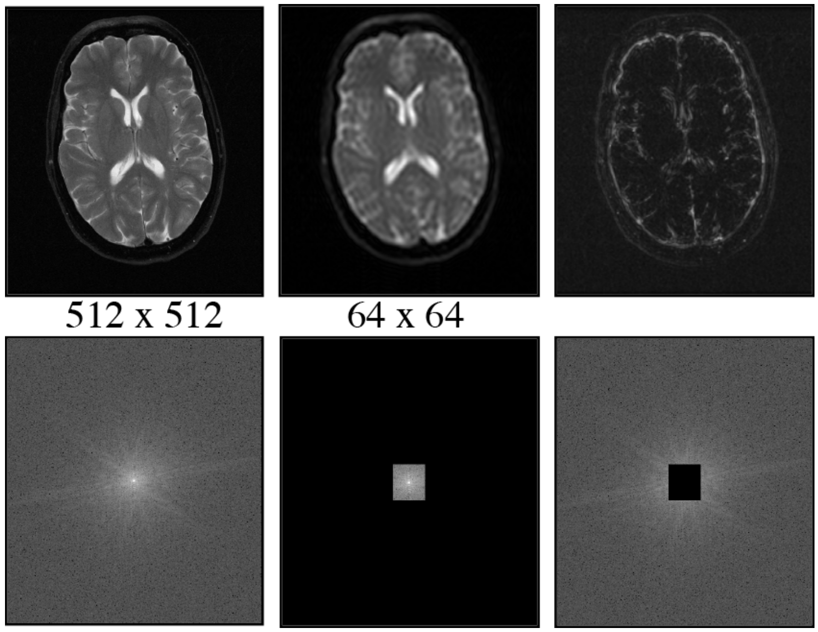

Figure 13.3: Example images (MRI) from Fourier samples, including a high resolution image and the corresponding Fourier samples (left), a lower resolution image and corresponding Fourier samples (center), and the difference between the two images and corresponding Fourier samples (right).

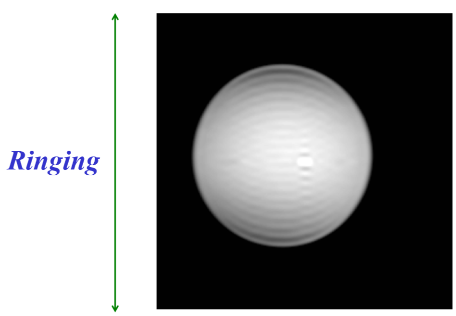

Figure 13.4: Example MR image with substantial Gibbs ringing in the vertical direction due to the low acquired resolution along this direction.

Projections and the Central Slice Theorem. Let us consider the 2D case for simplicity. The Central Section Theorem states that the (1D) FT of $p(s,\theta)$ (with respect to the variable s) is the same as the radial line of the (2D) Fourier Transform of $x(r_1,r_2)$, $X(k_1,k_2)$ along the same angle θ. This important result, with implications in tomographic image reconstruction and magnetic resonance imaging, is also known as the Central Section Theorem or Projection-Slice Theorem.

In FBP, each projection is first filtered using a ramp filter

$$p_{\text{filt}}(s,\theta) = p(s,\theta) * g(s) \tag{13.7}$$

where $g(s)$ is a 'high-pass' filter such that its Fourier transform is a ramp function $G(k) = |k|$, and then a backprojection operation is applied:

$$\hat{x}(r_1, r_2) = \int_0^{\pi} p_{\text{filt}}(s, \theta)|_{s=r_1 \cos\theta + r_2 \sin\theta} \, d\theta \tag{13.8}$$

leading to the reconstructed image $\hat{x}(r_1, r_2)$. Upon discretization of these two operations (filtering and backprojection), a direct reconstruction method for projection data can be obtained. An example of projection measurements, (unfiltered) backprojection, and filtered backprojection is shown in figure 13.5.

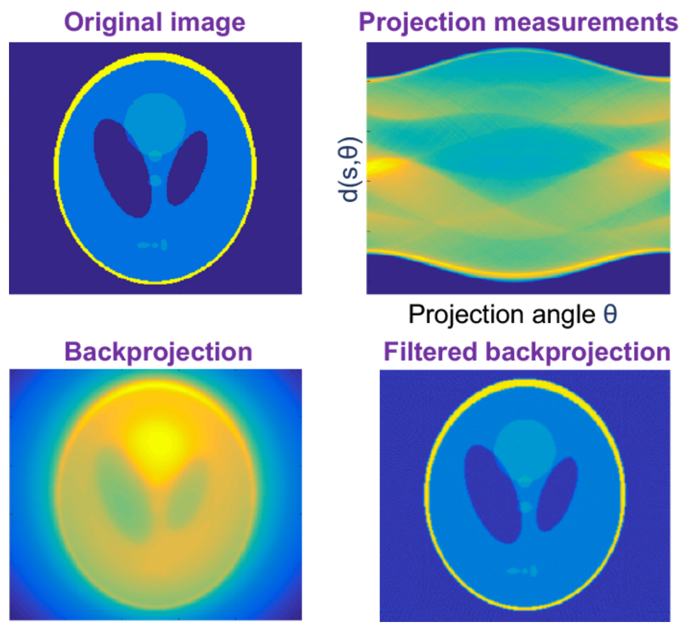

Figure 13.5: Example of projection measurements, (unfiltered) backprojection, and filtered backprojection from the Shepp-Logan phantom.

Figure 13.5 was generated in Matlab using the following simple code:

```
x = phantom(256); % Original image (Shepp-Logan phantom)
theta = 0:179; % Projection angles
p = radon(x,theta); % Calculate projections
xhat1 = iradon(p,theta,'linear','none'); % Backprojection reconstruction
xhat2 = iradon(p,theta); % Filtered backprojection reconstruction
figure;imagesc(x);
figure;imagesc(p)
figure;imagesc(xhat1);
figure;imagesc(xhat2);
```

Question: Why do we need the filtering? Why not simply do a backprojection of each of our projections and add them all up?

As can be seen in figure 13.6, having too few projections leads to artifacts in the filtered backprojection reconstructions (ie: not enough data). This figure has been generated using the following Matlab code.

```
x = phantom(256); % Original image
for SKIP = [0,2,4,6,8]
    theta = 0:(SKIP+1):179; % Projection angles
    p = radon(x,theta); % Calculate projections
    xhat = iradon(p,theta); % Filtered backprojection reconstruction
    figure;imagesc(xhat);
    title(['#projection angles = ' num2str(length(theta))]);
end
```

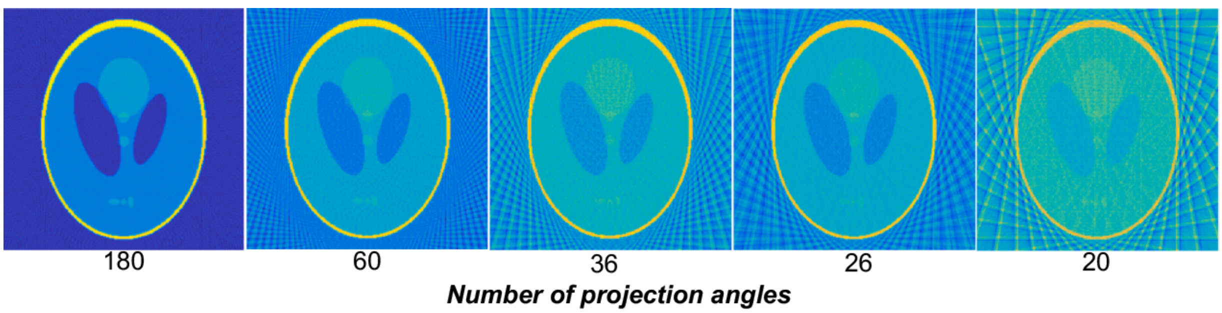

Figure 13.6: Example of filtered backprojection reconstructions using various numbers of projections of the same object.

Here are a few strengths and weaknesses of FBP reconstruction for projection data[1]:

- Strengths:
  - Fast: Based on FFT and a single back-projection. Few parameters to adjust.
  - Conceptually easy to understand and implement.
  - Reconstruction behavior well understood.
  - Typically works well for complete and good data
- Weaknesses:
  - Large number of projections required.
  - Full angular range required.
  - Only modest amount of noise in data can be tolerated.
  - Fixed scan geometries others require own inversion formulas.
  - Cannot make use of prior knowledge such as non-negativity.

[1]http://www2.compute.dtu.dk/ pcha/HDtomo/SC/Week1Days1and2.pdf