Biomedical Imaging

Exercise XCT #2 – Sinogram and CT Image Reconstruction

The objective of the exercise is to emulate the process of CT data generation (sinogram) and implement various CT image reconstruction approaches including simple backprojection (BP), filtered backprojection (FBP) and Fourier transform (FT) based reconstruction. Different filter settings in FBP and FT reconstruction and their impact on spatial resolution will be investigated.

Task 2.1

- Reconstruct the object whose sinogram taken at angle of 0 to 180 deg. is shown below (*using pencil and paper*, *Figure 1*).

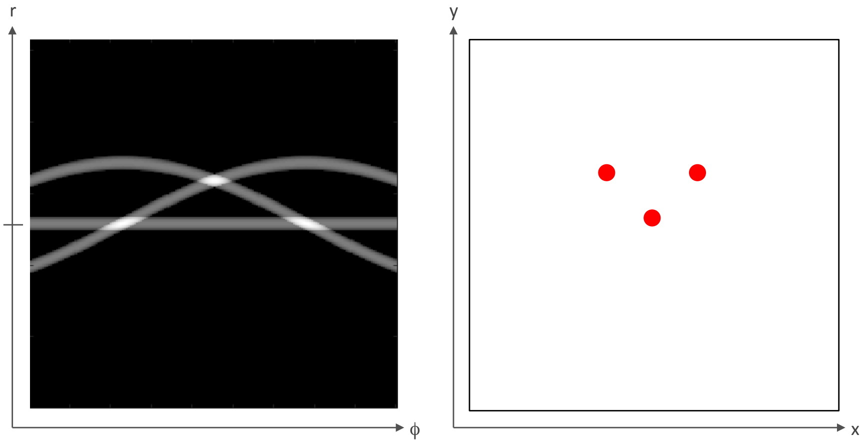

Figure 1: Sinogram (left); sketch the corresponding object (right).

- Start Matlab and enter "XCT_EXERCISE2"; the simple thorax phantom as implemented in Task 1.3 is displayed.
- Open XCT_EXERCISE2.m in the editor and read the code lines and comments carefully.
- Run code to generate the sinogram of the thorax phantom and interpret the result.

Task 2.2

- Implement image reconstruction using simple backprojection

```
rs = round(x*cos(phi/180*pi)+y*sin(phi/180*pi));
rs = rs+ceil((matrix+1)/2);
ix = find((rs>=1)&(rs<=(matrix+1)));
```

- Explain why the resulting image is blurred?

  Because of partial overlap between different projections after summation, the point spread function is proportional to abs(1/r), which causes the image to be blurred.

Task 2.3

- Copy code from Task 2.2 to implement filtered backprojection.

```
rs = round(x*cos(phi/180*pi)+y*sin(phi/180*pi));
rs = rs+ceil((matrix+1)/2);
ix = find((rs>=1)&(rs<=(matrix+1)));
filteredprojection = conv(phantom.sino(:,idx),filter,'same');
```

- Implement an ideal high-pass filter in function *CalcFilter* to correct for image blurring[1].

```
filter = U2R(filter);
```

- Why are "streak" artifacts seen? Correct "streaking" by combing the ideal high-pass with a low-pass filter.

  Streak artifacts are seen when too few projections are acquired.

```
weight = 0.5+0.5*cos(filter/matrix*2*pi);
filter = U2R(filter.*weight);
```

- Plot the Fourier transform pair of the ideal and the modified filter and discuss trade-offs (*see Figure 2*; here *r* refers to position in space while *u* denotes spatial frequency).

  Ideal filter: sharp edges are well-represented, but SNR is reduced.

  Modified filter: edges are less sharp compared to the ideal filter, but SNR is better.

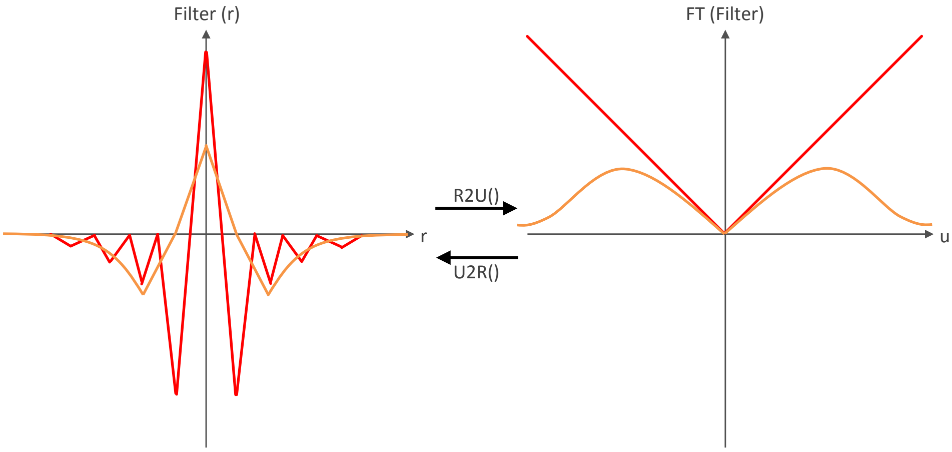

Figure 2: Filter in spatial domain (left) and corresponding depiction in spatial frequency domain (right).

Task 2.4

- State the Fourier-Slice theorem. Derive the relation between a projection and its Fourier transform.

  The 1D Fourier transform of a projection of a two-dimensional function is equal to the slice through the origin, parallel to the projection line, of the 2D Fourier transform of this function.

- Implement image reconstruction from projections using the two-dimensional (2D) Fourier transform.

```
for phi = projection_angles
        kx = round(r*sin(phi/180*pi))+ceil((matrix+1)/2);
        ky = round(r*cos(phi/180*pi))+ceil((matrix+1)/2);
        for r_ =1:length(r)
                transformedprojection = R2U(phantom.sino(:,idx));
                phantom.fft(kx(r_),ky(r_)) = transformedprojection(r_);
        end
        idx = idx+1;
end
```

- Compare reconstruction result with result obtained Task 2.2.
- Why are there image artifacts? (hint: inspect the Fourier space of your data)

  There are "holes" in k-space.

- Implement data gridding[2] in the Fourier space to correct for image artifacts.

[1] You may use functions *R2U()* and *U2R()* to transform between image and spatial frequency space (forward/inverse Fourier Transform (FT)).
[2] See: griddata() in Matlab

```
[kx,ky] = meshgrid(-fix(matrix/2):+fix(matrix/2),-fix(matrix/2):+fix(matrix/2));
for phi = projection_angles
    kx_(:,idx) = r*cos(phi/180*pi);
    ky_(:,idx) = r*sin(phi/180*pi);
    transformedprojection(:,idx) = R2U(phantom.sino(:,idx));
    idx = idx+1;
end
phantom.fft = griddata(kx_,ky_,transformedprojection,kx,ky);
phantom.fft(find(isnan(phantom.fft))) = 0;
```

Questions?

Andreas Dounas (adounas@biomed.ee.ethz.ch)
Jonathan Weine (weine@biomed.ee.ethz.ch)
Sebastian Kozerke (kozerke@biomed.ee.ethz.ch)