Prof. A. Zakhor Spring 2019

EE225b Homework 11 – Tomography

due on 03/13/2019 12:00pm; submitted through Gradescope

Overview:

In this assignment, you explore applications of the projection slice theorem to tomography.

Assignment specifics:

On the class website, there is a file `Pyramid.bmp`, 256 pixels wide and 384 pixels tall in 8-bit (256) gray. Using the `radon` function provided by Matlab, generate 180 equally spaced projections of the image. Take a moment to familiarize yourself with the capabilities of `radon` using the help files, and what it is computing for you. The radon function treats (192, 128) as the center of the image. For each projection, generate 465 sample points of $p_\theta(t)$ at equally spaced radius, roughly equivalent to one sample point per image unit.

Using the `iradon` function provided by Matlab, which implements the convolution (filtered) back-projection method, restore the original image. Save the reconstructed image as a bitmap file (`ConvBack.bmp`). Take a moment to familiarize yourself with the capabilities of `iradon` using the Matlab help files, and what it is computing for you. `iradon` includes an interpolation parameter (default linear), and a filter parameter (default Ram-Lak). How do these two parameters affect the convolution back-projection algorithm (refer to p. 429-432 in the textbook and lecture slides)? What happens if you reduce the number of θ samples by a factor of 2? By a factor of 5? Save these reconstructed images as well in bitmap files (`ConvBack2.bmp, ConvBack5.bmp`).

Implement the polar sampling method using nearest neighbor interpolation in the Fourier domain described in class, and compare it to the output of `iradon`. Save the reconstructed image as a bitmap (`Polar.bmp`).

Note:

For each problem, you need to:

(a) Email your source code (zip it before you email) to ee225bsp19@gmail.com if the question asks for any implementations.
- Make sure it is executable because I need to run your code to give you a score. Either MATLAB or Python is okay.
- Email title: FirstName_LastName_HW#. For example, Luya_Zhang_HW1

(b) Submit a single PDF file (not word or other formats) on Gradescope which contains:
- your answer for each problem;
- your source code (please also paste your source code here; screenshots are okay);
- your output image.

Make sure to prepare your solution to each problem on a separate page. On Gradescope, please select and match each page to the corresponding problems.

Here are some helpful Matlab commands:

`X = fft2(x)`	Computes the 2D-DFT of the matrix `x`
`x = ifft2(X)`	Computes the inverse 2D-DFT of the matrix `X`
`I = uint8(x)`	`I` is a matrix of integers ranging from 0..255
`imshow(I)`	Displays `I` as a grayscale image in the current figure
`I = imread('small.bmp', 'bmp')`	Reads the image file small.bmp and stores it in matrix `I`
`imwrite(I, 'result.bmp', 'bmp')`	Writes the matrix I to the image file result.bmp
`stuff = load('Phase.dat', '-mat')`	Loads the contents of `Phase.dat` stored in matrix format into the structure `stuff`
`stuff.ImagePhase`	Access the `ImagePhase` matrix stored in `stuff`