Assignment #6

Due date: June 12 (Mon.)

Submission

E-mail a zip file including the source codes, a report and test images to TA. **You must submit function m-files per problems like Appendix at the end of this document. If you don't follow the code structure in Appendix, the score will be deducted.** The filename should be named as student idn_name.zip (e.g. 20183000_kdhong.zip).
TA's e-mail address is **ltk010203@kaist.ac.kr**.

Due date: **06/12 23:59**. (Refer to the delay policy in the web site)

Test images in the web site:
shepp_proj_256x180.raw
bridge_gray_256x256.raw, cameraman_gray_256x256.raw

Notice

All the programming assignments are based on MATLAB. (Do not use any function in MATLAB. But **you can use basic functions available in C++ standard library** like round, ceil, floor, rand, bitshift, sqrt, sum, exponential, log, trigonometric and abs etc. functions **and vectorization**.) All source codes for submission should include comments.
Describe your work and analyze the corresponding results in the report. A proper length of the report is 10 pages of A4 size with figures. Report exceeding the recommended length will get a penalty. The report should include the followings.

1. Simple theoretical backgrounds & programming strategies
2. Result images
3. Analysis of the results

If a **copy version** is found, the score will be **zero** point without any exception.

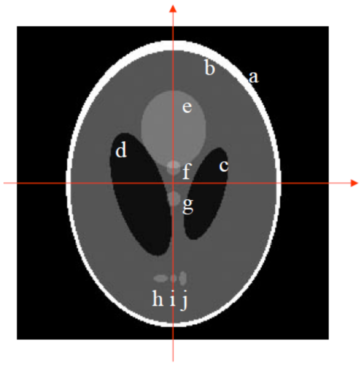
Scoring policy: implementation (60), processing time (10), and report (30)

The criteria of the scoring deduction applied to common problems:

1. Using MATLAB functions
    - Related to 'Language Fundamental' except to basic functions available in C++ standard library: -2 points per problem.
      'Language Fundamental':
      https://kr.mathworks.com/help/matlab/language-fundamentals.html?lang=en
    - the other MATLAB functions: -50% per problem.
2. Inexecutable code: -50% per problem.

1. Parallel beam image reconstruction algorithm

A sinogram is given for the 2D Shepp phantom in shepp_proj_256x180.raw (32bit). The corresponding specifications are as follows.
Range: (-1.0, 1.0), number of samples: 256, number of views: 180
Data type: float (4 byte for each sample)



Ellipse	Center	Major	Minor	θ	Gray
a	0,0	.69	.92	0	2
b	0,-.0184	.6624	.874	0	-.98
c	.22,0	.11	.31	-18°	-.02
d	-.22,0	.16	.41	18°	-.02
e	0,.35	.21	.25	0	.01
f	0,.1	.046	.046	0	.01
g	0,-.1	.046	.046	0	.01
h	-.08,-.605	.046	.023	0	.01
i	0,-.605	.023	.023	0	.01
j	.06,-.605	.023	.046	0	.01

2D Shepp phantom* (level: 1.02, window: 0.04)

(1) Perform the back projection.
(2) Perform the filtered back projection using the Ram-Lak and Shepp-Logan filters.
(3) Compare the results from two different filters in (2), by showing cut views of obtained images. Also, compare the results from (1) and (2).
(4) Repeat (2) with a reduced number of views of 60 (via down-sampling), and discuss the result by comparing it with the result of 180 views.

Ram-Lak filter

$$h_{RL}(0) = B^2 = \frac{1}{4\Delta x^2} \qquad (\text{if } \ k = 0)$$

$$h_{RL}(k) = 0 \qquad (\text{if } \ k \ \text{ even}) \qquad h_{RL}(k) = \frac{-4B^2}{\pi^2 k^2} = \frac{-1}{\pi^2 k^2 \Delta x^2} \qquad (\text{if } \ k \ \text{ odd})$$

Shepp-Logan filter

$$h_{SL}(k) = \frac{-2}{\pi^2 \Delta x^2 (4k^2 - 1)} = \frac{-8B^2}{\pi^2 (4k^2 - 1)}$$

2. Huffman coding

(1) For a given bridge image, perform Huffman coding by produce a Huffman coding table.
(2) Compare the entropy of the original bridge image with the average code length determined by Huffman coding, and calculate the compression efficiency.

3. Transform coding (8 x 8 block DCT + quantization)

For the given cameraman image,

(1) Perform the 8x8 block DCT.

(2) Perform the quantization for the result of (1) using variable quantization parameters. Use quantization parameters of 1 ~ 31.
Ex.) Quantization: (DCT coefficient) / (2*Quantization parameter)

(3) Perform the dequantization and the 8x8 block inverse DCT. Then, compare the result with the original and show the PSNR graph with respect to the quantization parameter.

Results

1.

(1) Back projection image
(2) - Backprojection image with Ram-Lak filter applied
- Backprojcection image with Shepp-Logan filter applied
(3) - Cutview of back projection image
- Cutview of back projection image with Ram-Lak filter applied
- Cutview of back projection image with Shepp-Logan filter applied
(4) Repeat **(2)** after reducing number of views of 90

3.

(3) - the dequantization and the 8x8 block inverse DCT images

Appendix_Methods for reading the input and for displaying the output image in MATLAB **(Important)**

- Please **place a code for reading the input at the beginning of the execution code and place a code for displaying the output image at the end of the execution code,** as shown in the example below.
- Display output images **for each MATLAB figure**, when executing the execution m-file, as shown in the example below.

'Problem_1.m'

```
Clear variables

function Problem_1()
    % Template for EE535 Digial Image Processing
    % Insert the code in the designated area below

    %% Loading directory for image files
    imgdir = uigetdir('Image Directory');
    file = fopen(fullfile(imgdir,'\Gray_Baby_512x256.raw'),'rb');
    gray_image = fread(file,fliplr([256,512]),'*uint8')';
    fclose(file);

    file = fopen(fullfile(imgdir,'\Color_Baby_512x256.raw'),'rb');
    color_image = fread(file,fliplr([256,512*3]),'*uint8')';
    fclose(file);

    %%
    %%---------------------Insert code below ----------------------%%
    output_image = InnerFunction(gray_image); % Sample code - delete
this

    %% Displaying figures (Edit this part as needed)
    figure; imshow(output_image,[]); title('Problem_1.1');
    figure; imshow(output_image2,[]); title('Problem_1.2');
    ...

    %%--------------------------------------------------------------%%

end

function OUTPUT = InnerFunction(INPUT)
%%------------------------------------------------------------------%%
                           %function code%
%%------------------------------------------------------------------%%
end
```