EEE 475/575 Medical Image Reconstruction & Processing
Homework 1
12 March 2020, Thursday at 23:59

GUIDELINES FOR HOMEWORK SUBMISSION

Instructions:

1. NO submission via E-MAIL (all emails will be discarded).
2. NO submission of RAR or ZIP files (rar or zip files will be discarded).
3. You should upload your solutions to Moodle as two separate files. One PDF file (should be named name_surname_id_hw#.pdf), and one MATLAB m file (should be named name_surname_id_hw#.m).
4. Submission system will remain open for 1 day after the deadline. No points will be lost if you submit your assignment within 12 hours of the deadline. There will be a 50% penalty if you submit after 12 hours but within 24 hours past the deadline. No submissions beyond 24 hours past the deadline.

MATLAB File Guidelines

1. It should be a single m file containing the codes for all questions (if you upload many m files, we will not evaluate any of them).
2. There is a template on Moodle to help you organize your solution in one m file, you should use it in all HWs.
3. Read the guidelines in the template file and follow each and every step.
4. If you do not upload your m file, your homework will not be evaluated (i.e., 0 pts). There will not be any exceptions to this rule.
5. If your m file gives runtime errors, your homework will not be evaluated (i.e., 0 pts). There will not be any exceptions to this rule.

PDF Report Guidelines

1. You get your points mainly from the PDF report file. The report should be typeset (no handwriting allowed). The PDF file should contain all results and plots. Unclear presentation of results will be penalized heavily. No partial credits to unjustified answers.
2. Maximum file size that can be uploaded to Turnitin is 40 MB. Pay attention to this before submitting your file.
3. You should properly name your pdf file and upload it to Turnitin upload area. We will discard all improperly named submissions (i.e., 0 pts).
4. After each question or each part you should properly display your MATLAB results, plots and MATLAB codes for that part (zero points for missing outputs).

PREPARATION 1) Centered FFT: In medical image reconstruction, the preferred way to display the k-space data and the image is such that the origin is at the center of the image or data array. The usual convention for the FFT in MATLAB, however, is that the origin is at the beginning of the array, or the upper left corner of a 2D array. To do a centered FFT, you want to do fftshift/ifftshift before/after the FFT. You will be doing this a lot in this course, so define the following functions:

```
function d = fft2c(im)
% d = fft2c(im)
%
% fft2c performs a centered fft2
im = fftshift(fft2(ifftshift(im)));
end

function im = ifft2c(d)
% im = fft2c(d)
%
% ifft2c performs a centered ifft2
im = fftshift(ifft2(ifftshift(d)));
end
```

When you type 'help fft2c' in MATLAB, you will now see the commented text that gives the function usage. Pay attention to including "help" sections when you create your own functions.

You will also want the corresponding one dimensional versions fftc and ifftc. Note that fftshift and ifftshift give exactly the same result when the array size is even-valued, but are different otherwise.

PREPARATION 2) Displaying the k-space spectrum: To display the k-space spectrum as an image, we typically do the following:

```
>> imshow(log(abs(d)+1),[])
```

In k-space, the value at DC (i.e., at the origin of k-space) is much larger than the values elsewhere. The "log" operation brings these values closer together, so that we can display the entire k-space more easily. The addition of 1 is to avoid the *log(0)* problem.

PREPARATION 3) For quantitative image quality assessment (IQA), we will use MATLAB's built-in functions *psnr* and *ssim*. Get acquainted with these functions. When using these functions, the images to be compared need to be normalized. For this homework, normalize each image by its maximum pixel intensity before using *psnr* and *ssim.*

PREPARATION 4) From the Moodle page of the class, download the following files:

- name_surname_idNumber_hw_.m: Template for the MATLAB code you will submit.
- ge_phantom.mat: Line-by-line acquired full k-space data for GE phantom.

PARTIAL K-SPACE RECONSTRUCTION

In this homework, you will implement the partial k-space algorithms that we have covered in class. If coded efficiently (e.g., vector arithmetic instead of *for* loops), the MATLAB code for each algorithm should not take longer than 20 lines or so.

1) Full k-space Data: Load ge_phantom.mat. This gives the k-space data acquired in a line-by-line fashion, with size $N_{ro} \times N_{pe}$, where N_{ro} is the number of readout samples and N_{pe} is the number of phase encode lines (both are 256 for this data set). Display the k-space spectrum as an image. Compute the corresponding MRI image, $m_f(x, y)$, and display the following: magnitude image, phase image, real part of the image, and the imaginary part of the image. For the rest of the homework, use $\left|m_f(x, y)\right|$ as the *reference image* for quantitative IQA.

2) Phase-compensated Full k-space Image: Use the central ±1/8th of k-space data (i.e., by using the central 64 phase encode lines) to estimate a low-resolution phase image, $p(x, y) = e^{j\angle m_s(x,y)}$. Display the phase of this image. Next, display the phase-compensated full k-space image, i.e., $Re\left\{m_f(x, y)p^*(x, y)\right\}$. Display the magnitude of the error image between this image and the reference image. Compute PSNR and SSIM for this image with respect to the reference image.

3) Repeat the steps in Question 2, this time using the central ±1/16th of k-space data. Comment on both the visual and quantitative results.

4) Partial k-space Data: Generate 5/8th partial k-space data by retaining the first 5/8th of the phase encode lines, and setting the remaining k-space data to zero. Display the k-space spectrum for this partial k-space data. You will use this 5/8th partial k-space data for Questions 5-8 below.

5) Trivial Reconstruction: Display the magnitude image for the trivial reconstruction (i.e., zero fill solution). Display the magnitude of the error image between this reconstruction and the reference image. Compute PSNR and SSIM for this reconstruction. Comment on the results.

6) PCCS: Implement the phase corrected conjugate synthesis (PCCS) method. Display the resulting k-space spectrum, the resulting image, and the error image. Compute PSNR and SSIM. Comment on the results.

7) Homodyne Reconstruction: Implement the Homodyne reconstruction using the weighting function of your choice (i.e., the 0-1-2 step function, or the ramp version, or a smoother version). Display the resulting k-space spectrum, the resulting image, and the error image. Compute PSNR and SSIM. Comment on the result.

8) POCS: Implement the POCS algorithm. Display the following images for the first 5 iterations: k-space spectrum, resulting image, error image. Compute PSNR and SSIM for the first 5 iterations. Comment on the results.

9) Performance at 9/16th k-space data: Generate 9/16th k-space data and display the k-space spectrum. Compare trivial reconstruction, PCCS, Homodyne, and POCS for this data. Display the k-space spectrums, resulting images, and error images. Compute PSNR and SSIM values. For POCS, display only the result of the 5th iteration. Comment on the results.

10) Comparison at 17/32th k-space data: Repeat Question 9 for 17/32th data. Comment on the results.

11) Based on your answers, comment on the performances of the compared reconstructions as a function of the sampled k-space ratio for this phantom data. Also, comment on whether PSNR and SSIM accurately reflect your visual observations.