Assignment V – Part I

Biomedical Imaging & Analysis (ECE J1-791) - Fall 2014

Sun Yat-sen University – Carnegie Mellon University (SYSU-CMU)
SYSU-CMU Joint Institute of Engineering
Instructor: Prahlad G Menon, PhD

Instructions

Please show your solutions to each problem in full, writing explanations neatly along with mathematical justifications, as needed. For computer programs, please remember to turn in your code through the course's blackboard session, as well as any plots / figures that are requested. This assignment is due on **Tuesday, 21 Oct 2014** via Blackboard, including an a Report with explanations associated with each question in the assignment, as well as any associated code and result files. If you have collaborated with another student on solving this homework assignment please state so (e.g. "I helped John with question 1" etc.).

LEARNING GOALS:

Part 1:

- *Basic Filtering without K-Space + Point Spread Functions:*
  - *Inverse Linear Operations* and the effects of Image Noise.
  - *Convolution and de-convolution using Matrixes and K-Space*
- *Using ImageJ / Fiji and MIJ (Matlab + ImageJ) for image processing and rendering.*

1. **(40 points, total)** The goal of this problem is to provide an intuition about how a point spread function (PSF) can degrade an image, and how the degradation can be removed using a linear system of equations setup. Using Matlab, load the data file "HW5P1Q1data.mat". Your workspace should now contain an image "img" which was digitally degraded using a point spread function (PSF) stored in variable "h" (also given in the MAT file), by convolution. That is:

$$img[m,n] = \Sigma_{p=-inf}^{inf} \Sigma_{q=-inf}^{inf} I[p,q]\, h[m-p, n-q] \qquad (1)$$

   a) (20 points) The equation above can be written in vector form as $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{b}$ is the vector form of image "img", above, $\mathbf{x}$ is the image I reshaped in a vector form, and the matrix A is the convolution operator matrix prepared from the PSF h(m, n). Using this linear system representation, solve for the original image I(p; q). Display the result image. **Remember to include your code.**
   HINT 1: Remember to 'pad' your images with 'zeroes' so that the convolution operator is effectively applied to the boundaries of your

images (refer class notes from Math Lecture on convolution). Eg: If Nx and Ny are the size of the image I in the x and y directions, img_padded = zeros(Ny+6,Nx+6); and img_padded(4:4+Ny-1,4:4+Nx-1) = I; If the size of the kernel, h, is (7x7).
HINT 2: Use "sparse" matrices (see: 'sparse' function, in Matlab) to reduce computation time.

b) **(10 points)** Repeat the problem above (solve for the original image I(p, q) from img and h) using the fft2 (and ifft2) Matlab functions, instead of the linear system described above. In your report mention the equations explaining the steps in your code using the Fourier transform of Equation (1). Display your resulting de-convoluted image again and submit your code.

c) **(10 points)** Add random noise noise to I[Ny,Nx] using the following method:

```
% noise = 0.1*randn(Ny,Nx);
% I_noise = I + noise;
```

Display your resulting de-convoluted image again. What changed..? Why..? How does the result change when you include noise of "10x" the amplitude (viz. noise = 1.0*randn(Ny,Nx)) ..?

2. **(60 points)** Smoothing / Blurring + de-convolution:

a) **(30 points)** Use the "median" filter in Matlab to interactively 'remove' the speckle noise you added in Q1 Part (c), and then repeat the deconvolution process using the Fourier Method. Report your findings / observations and submit your code. Explain in a few simple equations how the Median filter works.

i. Download ImageJ (the Fiji version): http://fiji.sc/Fiji . Attempt to apply the same Median Filter (in Process \ Filters \ Median) interactively using ImageJ and visualize your results there. Embed a snapshot in your report along with a brief explanation of the steps you followed including any parameter settings.
Note: you can also use plain old "ImageJ" from the NIH website:

ii. Repeat Q2 part (a) using MIJ – a plugin for Fiji – (http://fiji.sc/wiki/index.php/Miji) which will transfer the median filter processed result from ImageJ into your Matlab deconvolution code. Embed a screenshot of how you do this, in your report. Also, write a single unified Matlab script that uses MIJ to call the function in ImageJ / Fiji to perform this median filtering process, and then perform deconvolution. Please submit your code.
Note: Using plain ImageJ (not Fiji) , MIJ Will need to be installed using the following steps: http://bigwww.epfl.ch/sage/soft/mij/

b) **(10 points)** Use “deconvlucy” – the Matlab in-built function for the Richardson-Lucy deconvolution algorithm and repeat Q1 Part (c), using 10 iterations and the same PSF as Q1. How did your result improve..? Explain how the Richardson-Lucy algorithm works after doing a search reporting the functionality of the method in a few straightforward equations.

c) **(20 points)** If $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ has the same meaning as in Q1, the following is another method of deconvolution which is robust to noise, by minimizing the following equation for ‘x’.

$$\text{Min } ||\mathbf{A}\boldsymbol{x} - \boldsymbol{b}||^2 + \lambda\ ||\mathbf{L}\boldsymbol{x}||^2$$

Reduce the equation by opening up the expression above mathematically to show that the deconvoluted image, $\boldsymbol{x}$, can be given in the form: $\mathbf{X} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{(??)})^{-1}\mathbf{A}^T\mathbf{b}$. What is **(??)** in this expression...?

If **L** is a convolution matrix built used on the following 2D array for the Laplacian operator (i.e. the sum of the second derivatives in each dimension), write a program to compare the results of deconvoluted image (after reshaping it back to a 2D image!) using this method with Q2 Part (b).

0	1	0
1	-4	1
0	1	0

:

Remember to experiment with λ using different scalar values until you get the best result!

HINT: What happens when λ = 0..?