CSE 166: Image Processing, Spring 2019 – Assignment 3
Instructor: Ben Ochoa
Due: Monday, April 29, 2019 11:59 PM

Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment must be completed individually.
- This assignment contains both math and programming problems.
- Programming aspects of this assignment must be completed using MATLAB.
- Unless specified below, you may not use MATLAB functions contained in toolboxes, including the image processing toolbox. Use the MATLAB `which` command to determine which toolbox a function is contained in. If you are unsure about using a specific function, then ask the instructor for clarification.
- You must prepare a report as a pdf file. The report must contain your solutions and results, and all of your MATLAB source code as a listing in the appendix of your report.
- Additionally, you must create a zip file containing all of your MATLAB source code.
- Your source code must contain a file `main.m` which runs all code necessary to produce results for your report. `main.m` should run start to finish without error. Use relative paths to read input data. For questions which require numerical output, `main.m` should print a message indicating what question is being answered followed by the numerical output for that question. Example: `display('Problem 2b')` followed by your answer to problem 2b. The instructors should be able to reproduce your report by running `main.m`
- You must submit both files (.pdf and .zip) on Gradescope. Further, you must mark each problem on Gradescope in the pdf file.
- It is highly recommended that you begin working on this assignment early.

Problems

1. **Textbook problems (11 points)**
   (a) Problem 4.3 (1 point)
   (b) Problem 4.4 (1 point)
   (c) Problem 4.9 (2 points)
   (d) Problem 4.27 (1 point)
   (e) Problem 4.33 (2 points)

(f) Problem 4.40 (3 points)

(g) Problem 4.45 (1 point)

2. **Programming: The Fourier transform and filtering in the frequency domain (35 points)**

   (a) **The Fourier transform pair (5 points)**

   Develop a MATLAB function called `dft2d` that computes the 2D discrete Fourier transform of an array. The function should take as input a 2D array and return as output the 2D array of complex numbers corresponding to the discrete Fourier transform. Include code in your `main` script to time the execution of your `dft2d` function on an array of size 100×100 and print the result to the console. Include this result in your report. Compare the time of your `dft2d` function with the built in MATLAB function `fft2`. Why do you think the results are different? We will post runtimes - as measured on Anthony's computer - anonymously on Piazza. If you do not wish the runtime of your code to be posted, please send the instructors a message on Piazza. Hint: use the `tic` and `toc` commands in MATLAB. Use the "run and time" button in MATLAB to see a breakdown of your code's execution time.

   Develop a MATLAB script called `hw3_dft_ift.m` that reads the input image `cameraman.tif` (included with MATLAB), uses your `dft2d` function to compute the discrete Fourier transform (DFT) $F(u,v)$ (shifted such that the zero-frequency component $F(0,0)$ is centered) of the input image $f(x,y)$, calculates the magnitude $|F(u,v)|$ and phase $\phi(u,v)$ of $F(u,v)$, calculates the DFT $G(u,v) = |F(u,v)|e^{j\phi(u,v)}$, computes the inverse discrete Fourier transform (IDFT) $g(x,y)$ of $G(u,v)$ (after inverting the centering shift), and writes the real part of $g(x,y)$ to the output image `cameraman_dft_ift.png`. Use the function `imread` to read the input image in MATLAB. Use `imwrite` to write the output image in MATLAB. You may use MATLAB built-in functions to perform the intermediate steps.

   Include in your report the input image and output image. Additionally, include figures of $\log|F(u,v)|$ and $\phi(u,v)$, both with colorbars to show the scale. Include a title with each figure. What are the row and column indices of the $F(0,0)$ component before and after the centering shift?

   (b) **The convolution theorem (20 points)**

   The objective of this problem is to show that the output image

   $$g(x,y) = f(x,y) \bigstar h(x,y) = \mathfrak{F}^{-1}\{F(u,v)H(u,v)\}$$

   where $F(u,v)$ and $H(u,v)$ are the DFTs of the input image $f(x,y)$ and kernel $h(x,y)$, respectively.

   Develop a MATLAB function called `filterInFrequencyDomain` that applies a filter to an image in the frequency domain. The function inputs are a grayscale image and a kernel, and the function output is the filtered (double precision) image corresponding to the input image. The inputs and output are in the spatial domain. Zero padding must be used to mitigate wraparound error. The calculated output image must be the same size as the input image.

Develop a MATLAB script called `hw3_convtheorem.m` that reads the input image `moon.tif` (included with MATLAB) and applies the Laplacian kernel

$$h(x, y) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

to the input image $f(x, y)$ in spatial domain and in the frequency domain. Use the function `imread` to read the input image in MATLAB. The script must call the function `filterInFrequencyDomain` to apply the filter in the frequency domain. You may use the function `imfilter` to apply the filter in the spatial domain.

Include in your report the input image. Additionally, include figures of both resulting filtered images (with colorbars to show the scale). Include a title with each figure. Comment on the resulting filtered images. Why would it be beneficial to implement filtering in the frequency domain? (Optional: subtract a filtered image from the input image to yield a sharpened image.)

(c) **Lowpass filtering (10 points)**

i. Develop a MATLAB script called `hw3_lpf.m` that reads the input image `testpat1.png` (included with MATLAB) and applies an ideal lowpass filter in the frequency domain. Your lowpass filter must must have radius $D_0 = 50$. Try at least two additional radii. Include the original image and all three ideal lowpass filtered images in your report and comment on the difference.

ii. Compare your results from the previous question with a Gaussian lowpass filter. Try at least three different values for the filter variance (σ^2). Include the original image and all three Gaussian lowpass filtered images in your report and comment on the difference with ideal lowpass filtering.