CSE 166: Image Processing, Spring 2020 – Assignment 4
Instructor: Ben Ochoa
Due: Wednesday, May 20, 2020, 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 6.1 (5 points)
   (b) Problem 6.3 (3 points)
   (c) Problem 6.17 (1 point)
   (d) Problem 6.30 (2 points)

2. **Programming: The wavelet transform and wavelet-based image processing (30 points)**

   Use image analysis-related functions contained in the MATLAB Wavelet Toolbox to complete these problems.

   (a) **The wavelet transform (10 points)**

      i. Create a 256×256 image of data type `double` where every pixel value is 1. Then, use the function `wavedec2` to compute a 5-scale Haar wavelet decomposition of the image and use the function `appcoef2` to reconstruct the approximation coefficients for levels 1, 2, 3, 4, and 5. Determine the equation that scales pixel values in the original image (i.e., the image at level $n = 0$) to pixel values in the image at level n. Ensure this equation holds regardless of the pixel values in the original image. Include this equation in your report.

      ii. Develop a MATLAB script called `hw4_dwt.m` that performs the following. Read the input image `cameraman.tif` (included with MATLAB), convert it to data type `double`, and scale it by 1/255. Then, use the function `wavedec2` to compute a 3-scale Haar discrete wavelet decomposition of the image, and use the outputs of the functions `appcoef2` and `detcoef2` to create a figure similar to the one shown on slide 29 of lecture 13 (hint: scale each detail coefficients "image" by 1 over the quantity two times the maximum absolute value of the coefficients, then add 1/2 to the result). Ensure the approximation coefficients are scaled correctly using the scale determined in the previous subproblem. Additionally, reconstruct the approximation coefficients for levels 2 and 1, scale the approximation coefficients correctly, scale them again by 255 and convert the resulting "images" to data type `uint8`, then write the images to `cameraman_A2.png` and `cameraman_A1.png`, respectively. Use the function `imread` to read the input image in MATLAB. Use `imwrite` to write the output image in MATLAB.

      Include in your report the input image, the figure similar to the one shown on slide 29 of lecture 13, and output approximation coefficients images.

   (b) **Wavelet-based "edge" detection (10 points)**

   Develop a MATLAB script called `hw4_edges.m` that reads the input image `cameraman.tif` (included with MATLAB) and computes a 1-, 2-, 3-, and 4-scale Symlets 4 wavelet decomposition of the input image. For each resulting decomposition, set the approximation coefficients to zero and perform wavelet reconstruction (back up to level 0), again using Symlets 4 wavelet filters. For this problem (and the next one), it is easier to use the functions `dwt2` and `idwt2` than `wavedec2` and `waverec2`, but ensure you set the DWT extension mode to periodized first (otherwise, you will encounter issues with different sized images).

   Include in your report the input image. Additionally, include figures of the reconstructed "edge" images (with colorbars to show the scale). Briefly discuss the resulting images, including any differences between them.

   (c) **Wavelet-based noise removal (10 points)**

Develop a MATLAB script called `hw4_nr.m` that reads the input image `rice.png` (included with MATLAB), computes a 2-scale Symlets 4 wavelet decomposition of the input image, sets level 1 and level 2 detail coefficients with absolute value less than 20 to 0, performs wavelet reconstruction (back up to level 0), again using Symlets 4 wavelet filters, and writes the results to `rice_nr.png`.

Include in your report the input image and output image. Briefly discuss the results.