CSE 166: Image Processing, Fall 2016 – Assignment 5

Instructor: Ben Ochoa

Due: Wednesday, November 30, 2016, 11:59 PM

Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment must be completed in groups of two.
- This assignment contains both math and programming problems.
- Programming aspects of this assignment must be completed using MATLAB.
- You must prepare a report containing your solutions and results.
- Your report will be a pdf file named `CSE_166_hw5_lastname1_studentid1_lastname2_studentid2.pdf`, where `lastname`{`1,2`} and `studentid`{`1,2`} are the last names and student ID numbers, respectively, of the group members.
- All of your MATLAB source code must be included in an appendix of your report.
- One group member must submit your report on Gradescope and specify the other group member at the time of submission.
- Additionally, you must create a zip file named `CSE_166_hw5_lastname1_studentid1_lastname2_studentid2.zip`, where `lastname`{`1,2`} and `studentid`{`1,2`} are the last names and student ID numbers, respectively, of the group members. This zip file will contain the pdf file and a directory named `code` that contains all of your MATLAB source code.
- Submit your completed assignment by email to `vrg001@eng.ucsd.edu` **and** `dpradhan@eng.ucsd.edu`. The subject of the email message must be `CSE 166 Assignment 5`. Attach the zip file to the message.
- It is highly recommended that you begin working on this assignment early to ensure that you have sufficient time to correctly implement the algorithms and prepare a report.

Problems

1. **Textbook problems (12 points)**
   (a) Problem 8.5(a) (1 point)
   (b) Problem 8.9(a) (1 point)
   (c) Problem 9.5 (3 points)
   (d) Problem 9.6 (3 points)
   (e) Problem 9.7 (1 point)

(f) Problem 9.8 (1 point)

(g) Problem 9.17 (2 points)

2. **Programming: Image compression-related (25 points)**

   (a) **The discrete cosine transform and lossy block processing (10 points)**

   Develop a MATLAB script called `hw5_lossy_dct_block.m` that reads the input image `cameraman.tif` (included with MATLAB), independently processes non-overlapping blocks (i.e., subimages) of size 8×8, and writes the resulting image to file. For each block, compute the discrete cosine transform (DCT), retain the n greatest DCT coefficients (i.e., set the $64 - n$ remaining DCT coefficients to zero), and compute the inverse DCT. Perform this for $n = 32, 16, 8$ and write the resulting images to `cameraman_dct_32.png`, `cameraman_dct_16.png`, and `cameraman_dct_8.png` respectively.

   Include in your report the input image and output images. Additionally, include figures of the scaled error images (with colorbars) and the root-mean-square error associated with each output image. Include a title with each figure. Comment on the resulting images, including any differences between them.

   You may use the `blockproc` function. Use the function `imread` to read the input image in MATLAB. Use `imwrite` to write the output image in MATLAB.

   (b) **The discrete wavelet transform and lossy processing (10 points)**

   Develop a MATLAB script called `hw5_lossy_dwt.m` that reads the input image `cameraman.tif` (included with MATLAB), computes the discrete wavelet transform (DWT) using the Daubechies db4 filters, sets to zero p percent of all detail coefficients, and computes the inverse DWT. perform this for $p = 75\%, 85\%, 95\%$ and write the resulting images to `cameraman_75.png`, `cameraman_85.png`, and `cameraman_95.png` respectively.

   Include in your report the input image and output images. Additionally, include figures of the scaled error images (with colorbars) and the root-mean-square error associated with each output image. Include a title with each figure. Comment on the resulting images, including any differences between them.

   You may use the `dwt2` and `idwt2` functions. Use the function `imread` to read the input image in MATLAB. Use `imwrite` to write the output image in MATLAB.

   (c) **Morphological image processing (5 points)**

   Develop a MATLAB script called `hw5_morphological.m` that reads the input image `rice.png` (included with MATLAB), computes a binary image (called a logical image in MATLAB) where pixel values greater than 120 are set to 1 (and pixel values less than or equal to 120 are set to 0), and performs morphological image processing on the resulting binary image in an attempt to remove the "noise", and writes the results to `rice_morphological.png`. Your results should be at least as good as those shown in Figure 1.

   Include in your report the input image and output image. Describe the approach and comment on the results.

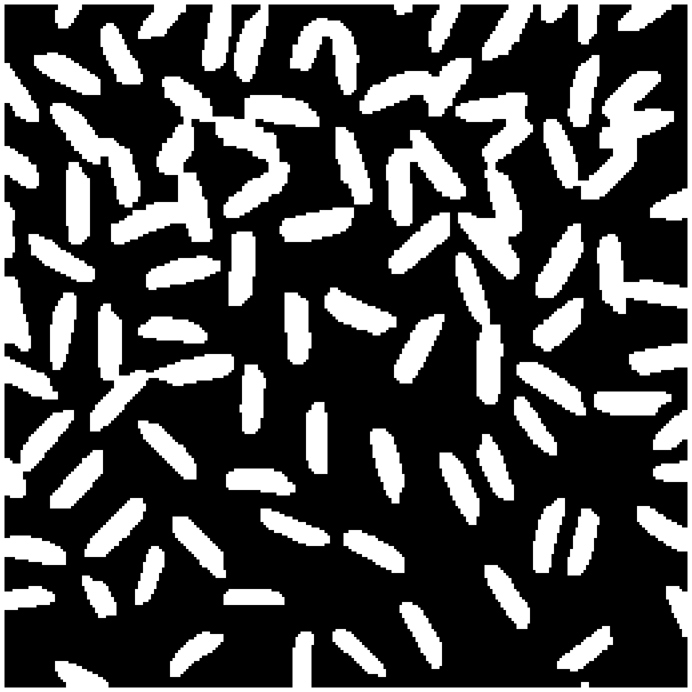

Figure 1: Results from simply morphological image processing.