HW4: Image Restoration and Color Image Processing

DIP Teaching Stuff, *Sun Yat-sen University*

Congratulations! You have survived from the most difficult parts of this course. HW4 is a quite easy journey. Most programming tasks will reuse codes that you have finished before. Take it easy and submit a report (in **PDF** format) and all relevant codes as the homework solution. However, please pay attention again: Plagiarism = Fail, and there may be at least 30% penalty for late homework.

1 Exercises

Please answer the following questions in the report.

1.1 Color Spaces (20 Points)

Consider the following 64×64 RGB image, which is divided into 4 non-overlapping blocks. Each block contains exactly 1 color, with the corresponding RGB value shown. Now we convert this RGB image into the HSI color space.

Figure 1: *An image which contains* 4 *non-overlapping blocks..*

1. Describe the appearances of the hue channel, the saturation channel, and the intensity channel, respectively. (6 Points)
2. Suppose we blur the saturation channel by a 16×16 arithmetic mean filter, and then convert the HSI image back to the RGB color space. Describe the appearance of the result (ignoring image border effects due to filtering). (4 Points)
3. Repeat Question 2 for the hue channel. (10 Points)

2 Programming Tasks

Write programs to finish the following three tasks, and answer questions in your report. Don't forget to submit all relevant codes.

2.1 Pre-requirement

Language Any language is allowed.

Others There remain some issues that you should pay attention to:

1. You can use third-party packages for operating images. But you should manually implement your programming tasks. For example, though you can use "imread" of Matlab to load an image, you cannot invoke "medfilt2" of Matlab for median filtering.
2. Good UX (User Experience) is encouraged, but will only bring you negligible bonuses. Please don't spend too much time on it, since this is not an HCI course.
3. Keep your codes clean and well-documented. Bad coding styles will result in 20% penalty at most.

2.2 Image Filtering (20 Points)

Input Please download the archive "hw4.zip", unzip it and take the image named "task_1.png". It is the input of this task. You can convert the image format (to BMP, JPEG, ...) via Photoshop if necessary.

Target The white bars in Fig. 2 are 8 pixels wide and 224 pixels high. The separation between bars is 16 pixels. For other details please refer to your input image.

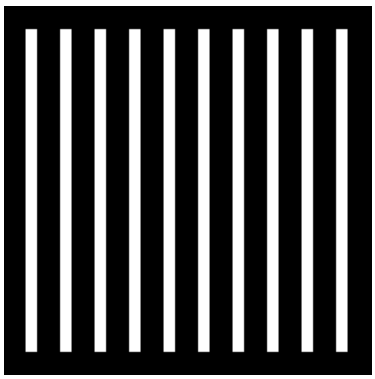

Figure 2: *Input image for "Task 2.2 Image Filtering".*

Finish the following applications (reuse the function "filter2d" in HW2 if you like):

1. Filter your input image with 3×3 and 9×9 arithmetic mean filters respectively. Paste your two results on the report. Also briefly describe what each result looks like, e.g. the width/height/color of bars in the report. (4 Points)

2. Repeat the first application with 3×3 and 9×9 harmonic mean filters. Paste your results in the report and briefly describe what each result looks like. (8 Points)

3. Repeat the first application with 3×3 and 9×9 contraharmonic mean filters with $Q = 1.5$. Paste your results in the report and briefly describe what each result looks like. (8 Points)

2.3 Image Denoising (34 Points)

Input Please download the archive "hw4.zip", unzip it and take the image named "task_2.png". It is the input of this task. You can convert the image format (to BMP, JPEG, ...) via Photoshop if necessary.

Target You are required to finish two applications:

1. Write a noise generator to add Gaussian noise or salt-and-pepper (impulse) noise to an image. Your generator should be able to specify the noise mean and standard variance for Gaussian noise, and the probabilities of each of the two noise components for salt-and-pepper noise. (4 Points)

2. Add Gaussian noise to your input image with mean 0 and standard variance 40, and paste the noisy image in your report. Then try to denoise it via arithmetic mean filtering, geometric mean filtering, and median filtering. Paste your filtering results in the report. Compare these results, and discuss which one looks better / worse, and why, within 1 page. (9 Points)

3. Add salt noise to your input image by setting its probabilities to 0.2, and paste the noisy image in your report. Then try to denoise it via min filtering, harmonic mean filtering and contraharmonic mean filtering. Paster your filtering results in the report. In addition, for contraharmonic mean filtering, you are required to paste two results: one for $Q > 0$ and the other for $Q < 0$. Discuss why setting a wrong value for Q would lead to terrible results within 1 page. (8 Points)

4. Add salt-and-pepper noise to your input image by setting both of the probabilities to 0.2, and paste the noisy image in your report. Then try to denoise it via arithmetic mean filtering, geometric mean filtering, harmonic mean filtering and median filtering. Paste your filtering results in the report. Compare these results, and discuss which one looks better / worse, and why, within 1 page. (8 Points)

5. Discuss how you implement all the above filtering operations, i.e., arithmetic, geometric, harmonic, contraharmonic, min, and median filtering, in less than 1 page. (5 Points)

Note: Under different configurations, the behaviors of filters may change significantly. You should select the best parameters (e.g., filter size) when perform denoising. Please write down your selections in the report, as well.

2.4 Histogram Equalization on Color Images (26 Points)

Input Please download the archive "hw4.zip", unzip it and pick up the image in the directory "task_3" according to the last two digits of your student ID. You can convert the image format (to BMP, JPEG, ...) via Photoshop if necessary.

Target Read the input image in RGB mode. Then:

1. Use the function "equalize_hist" that you have written in HW2 to process the R, G, B channels separately. Rebuild an RGB image from these three processed channels and paste it in the report. (6 Points)

2. Calculate the histogram on each channel separately, and then compute an average histogram from these three histograms. Use the average histogram as the basis to obtain a single histogram equalization transformation. Apply this transformation to the R, G and B channels individually, and again rebuild an RGB image from the three processed channels. Paste the RGB image in the report. (6 Points)

3. Convert the input image to the HSI color space, and then perform histogram equalization on the intensity channel. Convert the result back to the RGB color space and paste it in the report. (6 Points)

4. Compare and explain the differences in the above three results within 1 page. (8 Points)