Digital Image Processing: 525 U0920 / ESOE 5096

Computer Assignment 3
Image Restoration

Due: three weeks from today, 9:00pm

*Hand in online via **CEIBA***

Total score: 300

This assignment will introduce you some fundamental techniques in image restoration. In particular, we will show you how to add different types of noise in images and how to design filters to reduce noise using MATLAB functions and your own codes.

What to turn in:

Matlab source codes that perform each individual experiment and the associated results and comments if any. Please consolidate into one single file. For your own codes of filter functions, you can save all of them into another single file.

1. Noise Models

(a) Gaussian noise

Gaussian noise is an idealized form of white noise, which is caused by random fluctuations in the signal. To add noise, we can use the MATLAB function **imnoise**, which takes a number of different parameters as

g = imnoise(f, type, parameters)

where *f* is the input image and *g* is the noisy image corresponding to the *type* and *parameters*. Note that function **imnoise** converts the input image to class double in the range [0, 1] before adding noise to it. For example, to add Gaussian noise, you can use **g = imnoise(f, 'gaussian')**. The default is zero mean noise with 0.01 variance.

(b) Salt-and-pepper noise

It is also called impulse noise, shot noise, or binary noise. Salt-and-pepper noise can be caused by sharp, sudden disturbances in the image, whose appearance is randomly scattered white or black (or both) pixels over the image. To add salt-and-pepper noise: **g = imnoise(f, 'salt & pepper')**. The default is 0.05 noise density.

(c) Speckle noise

While Gaussian noise is modeled by random values added to an image, speckle noise can be modeled by random values multiplied by pixel values; hence it is also called multiplicative noise. Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. As above, **imnoise** can produce speckle noise: **g = imnoise(f, 'speckle')**. The default value of variance is 0.04.

2. Gaussian noise reduction (100%)

Read the *peppers_gray.tif* file and add Gaussian noise to it with same zero mean but two different variance values: (i) 0.05 and (ii) 0.2. Apply the following filters to restore the images:

(a) Arithmetic mean filter: Recall the MATLAB function **fspecial** with type *average* to create a 2-D filter kernel. Then use function **filter2** to filter the images with the kernel.

(b) Gaussian lowpass filter: Similar steps as (a) but with type *gaussian* to create the filter kernel.

(c) Median filter: You can use the MATLAB function **medfilt2** to filter the images.

(d) Wiener filter: Again, you can use the function **wiener2** to filter the images.

(e) Alpha-trimmed mean filter: Please develop a function named **alphatrim** to perform this filter. Please refer to the equation in the lecture notes and textbook.

Note that you need to find out the best result produced by each individual filter by tuning the parameters such as kernel size and others. I suggest you start from a minimum size of 3×3 and gradually increase the size 5×5, 7×7, … to find the optimal result based on the peak signal-to-noise ratio (PSNR) value (the higher the better):

$$\text{PSNR}=10\cdot\log_{10}\left[\frac{L_i L_j I_{\max}{}^2}{\sum_{i=1}^{L_i}\sum_{j=1}^{L_j}\left|I'(i,j)-I(i,j)\right|^2}\right]$$

where I represents the original intact image with size $L_i \times L_j$, I' is the restored image, and $I_{\max}$ is the maximum possible intensity value, for 8-bit case $I_{max} = 255$.

Show me the noisy image along with the best restored images of the five methods in one diagram. Also report the corresponding parameter and PSNR values for each method and discuss the performance of the methods.

3. Salt-and-pepper noise reduction (100%)

First, read the *woman_blonde.tif* file and add salt-and-pepper noise to it with different noise density values: (i) 0.1 and (ii) 0.4. Then, apply the following filters to restore the images:

(a) Median filter.
(b) Alpha-trimmed mean filter.
(c) Midpoint filter: Please develop a function named **midpoint** to perform this filter. Please refer to the equation in the lecture notes and textbook.
(d) Outlier filter: To overcome the sorting of each pixel in the kernel of the median filter, the outlier filter speeds up the computation by treating noisy pixels as outliers. Please develop a function named **outlier** based on the algorithm shown in the following steps:
   1. Choose a threshold value D.
   2. For a given pixel, compare its intensity value p with the mean m of the values of its surrounding neighbors, say eight neighbors for a 3×3 kernel.
   3. If $|p-m|>D$, classify the pixel as noise, otherwise not.
   4. If the pixel is noisy, replace its value with m; otherwise leave its value unchanged.
(e) Adaptive median filter: Please develop a function named **adpmedian** to perform this filter. Please refer to the algorithm in the lecture notes and textbook.

Show me the noisy image along with the best restored images of the five methods based on the PSNR values in one diagram. Also report the corresponding parameter and PSNR values for each method and discuss the performance of the methods.

4. Speckle noise reduction (100%)

Read the *walkbridge.tif* file and add speckle noise to it with two different variance values: (i) 0.1 and (ii) 0.3. Apply the following filters to restore the images:

(a) Arithmetic mean filter.
(b) Gaussian lowpass filter.
(c) Wiener filter.
(d) Outlier filter.
(e) Alpha-trimmed mean filter.
(f) Midpoint filter.
(g) Contraharmonic mean filter: Please develop a function named **chmean** to perform this filter. Please refer to the equation in the lecture notes and textbook.

Show me the noisy image along with the best restored images of the seven methods based on the PSNR values in one diagram. Also report the corresponding parameter and PSNR values for each method and discuss the performance of the methods.