Practical 2 Part 2 Spatial Filtering

February 9, 2021

Jaya Parmar, MATLAB filename part2_spatialfiltering.m

1 Introduction

Our aim is to reduce the specific noise present in given images by applying appropriate filters.

2 Method

2.1 Study the images and their histograms

1. **Read in images**
   We start the study by reading in all 3 images,

$$poison = imread(poison_noise.jpg') \tag{1}$$

$$salt = imread('salt_and_pepper_noise.png') \tag{2}$$

$$quant = imread('paris_quant_noise.jpg') \tag{3}$$

   Image variables poison and quant are color images hence we will seperate their RGB channels below

$$pred = poison(:,:,1) \tag{4}$$

$$pgreen = poison(:,:,2) \tag{5}$$

$$pblue = poison(:,:,3) \tag{6}$$

$$qred = quant(:,:,1) \tag{7}$$

$$qgreen = quant(:,:,2) \tag{8}$$

$$qblue = quant(:,:,3) \tag{9}$$

2. **Image histograms**
   Then we study the histograms of these images, Figure1 shows each of original image histograms

$$subplot(2,2,1), imhist(poison), title('poison') \tag{10}$$

$$subplot(2,2,2), imhist(salt), title('salt') \tag{11}$$

$$subplot(2,2,3), imhist(quant), title('quant') \tag{12}$$

   Histogram of poison image shown in Figure 2 is generated from code below,

$$subplot(2,2,1), imhist(pred), title('poisonred') \tag{13}$$

$$subplot(2,2,2), imhist(pgreen), title('poisongreen') \tag{14}$$

$$subplot(2,2,3), imhist(pblue), title('poisonblue') \tag{15}$$

   Histogram of quant image shown in Figure 3 is generated from code below,

$$subplot(2,2,1), imhist(qred), title('quantred') \tag{16}$$

$$subplot(2,2,2), imhist(qgreen), title('quantgreen') \tag{17}$$

$$subplot(2,2,3), imhist(qblue), title('quantblue') \tag{18}$$

2.2 Filter Selection

1. **Poison noise**
   Poison noise or shot noise is a linear noise which can be filtered by a linear filter. We use average filter to remove the noise as below

$$hp = fspecial('average', 5) \tag{19}$$

$$P = imfilter(poison, hp) \tag{20}$$

   We tried with filter sizes 3X3, 5X5 and 7X7 and found the best results with filter size 5X5. Also, the code works on a RGB image and we need not separate out the channels.
   The results are shown in **Figure4** of the results section.

2. **Salt and Pepper noise**
   Salt and Pepper or Impulse noise is a non-linear noise which needs a non-linear filtering.
   Median filters work best for this kind of noise. We used the below code to filter out the noise from the original image.

$$Med1 = medfilt2(salt) \quad (21)$$

The results are shown in **Figure5** in the results section.

3. **Quantization noise**
   Quantization noise occurs when an image is converted from analog to digital form.
   As the histogram shows sharp gaps, we will use a gaussian filter to smoothen out the image.
   We use imgaussfilt3 instead of imgaussfilt as it can be used directly to filter a RGB image.

$$Q = imgaussfilt3(quant); \quad (22)$$

   Results in **Figure6** of the results section.

3 Results

Figure 1

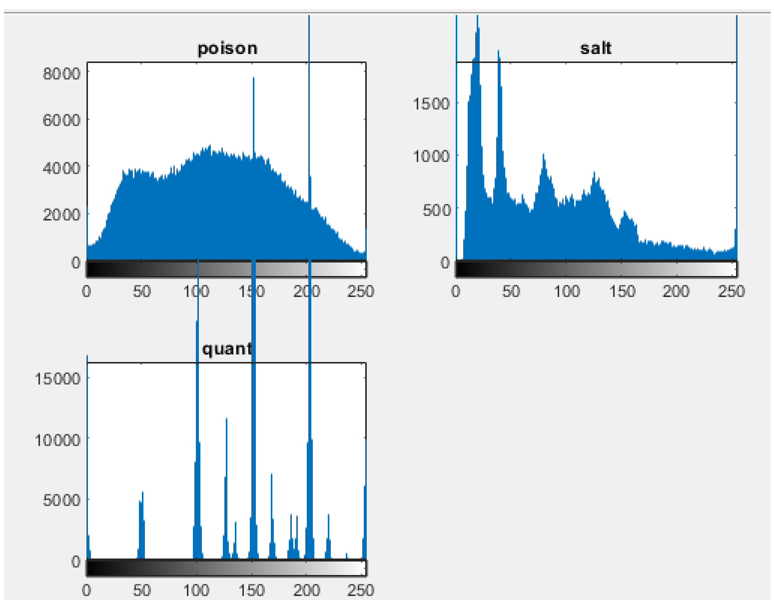


Figure 2

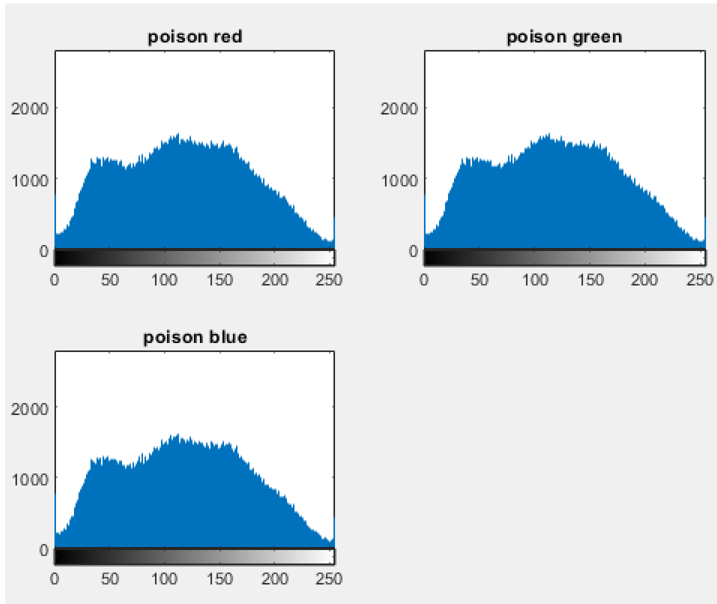


Figure 3

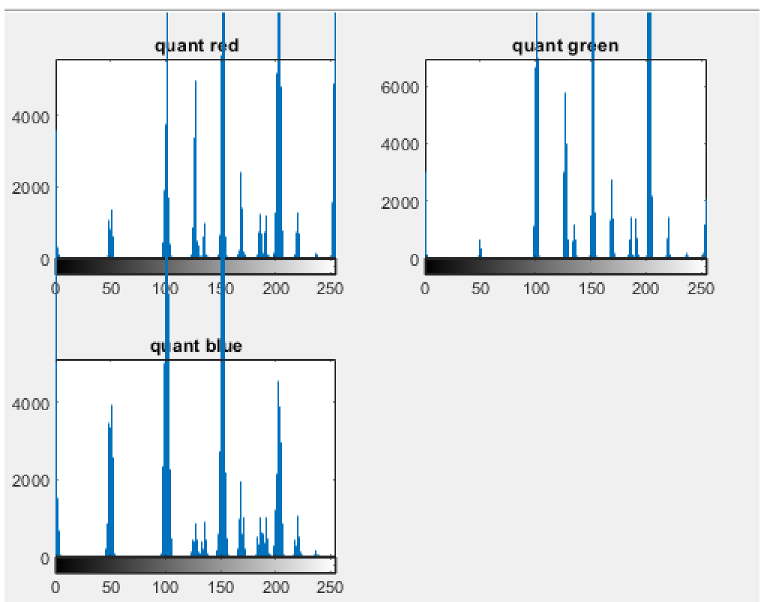


Figure 4

Figure 5

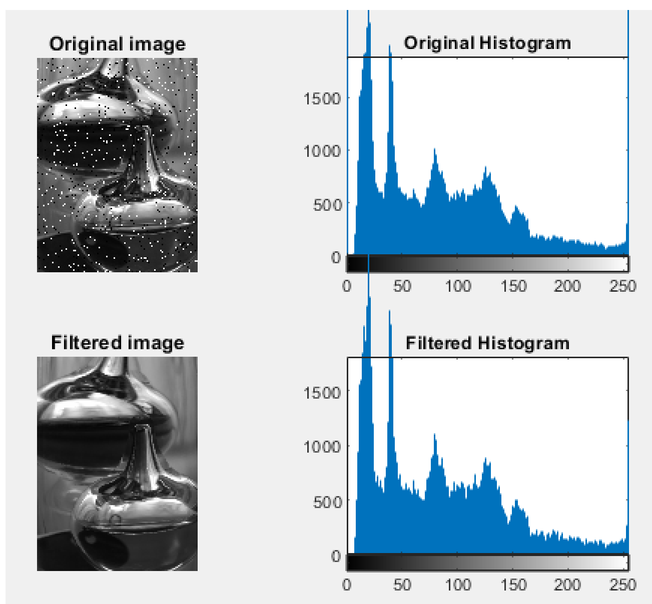
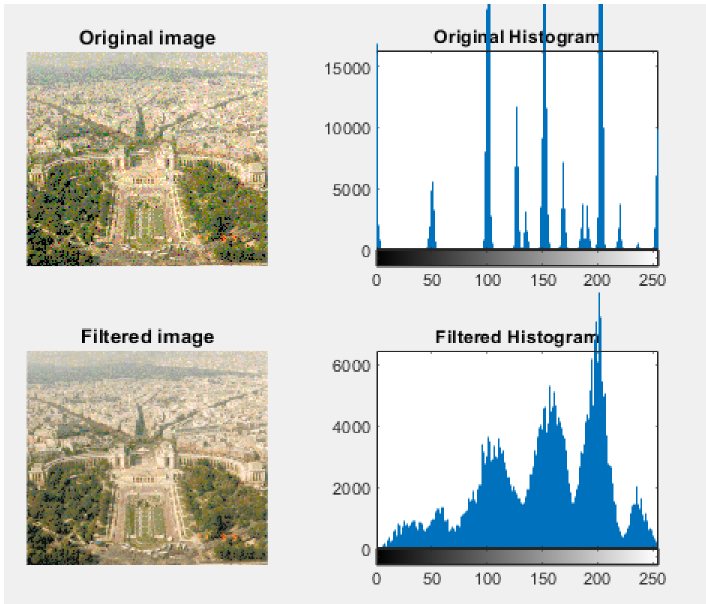


Figure 6



4 Discussion

- As the original histogram of poison noise shows, it is spread out a the x-axis. Filtered histogram shows the y-axis has multiple sharp peaks spread across a 'narrower' range of the histogram. The resulting image lookes less noiser than the original image.

- The filtered histogram of a salt and peppar noise image resembles the histogram of the original image. One difference is the smaller peaks have become 'sharper' in the filtered histogram.

- Quantization noise original histogram has sharp peaks spread discontinously across the range. Guassian noise filter makes the filtered histogram more continuous across the range. The final image has less noise, for example, see the blue river on the top of the image.

5 Appendix

```
set(0, 'defaulttextinterpreter','Latex');
poison = imread('poison_noise.jpg'); %color
salt = imread('salt_and_pepper_noise.png');
```

```
quant = imread('paris_quant_noise.jpg');  %color

pred = poison(:,:,1);
pgreen = poison(:,:,2);
pblue = poison(:,:,3);

qred = quant(:,:,1);
qgreen = quant(:,:,2);
qblue = quant(:,:,3);

figure(1),
subplot(2,2,1),imhist(poison),title('poison');
subplot(2,2,2),imhist(salt),title('salt');
subplot(2,2,3),imhist(quant),title('quant');

figure(2),
subplot(2,2,1),imhist(pred),title('poison red');
subplot(2,2,2),imhist(pgreen),title('poison green');
subplot(2,2,3),imhist(pblue),title('poison blue');

figure(3),
subplot(2,2,1),imhist(qred),title('quant red');
subplot(2,2,2),imhist(qgreen),title('quant green');
subplot(2,2,3),imhist(qblue),title('quant blue');

%% Poison results
hp = fspecial('average',5);
P = imfilter(poison,hp);

figure(4),
subplot(2,2,1),imshow(poison),title('Original image');
subplot(2,2,2),imhist(poison),title('Original
   Histogram');
subplot(2,2,3),imshow(P),title('Filtered image');
subplot(2,2,4),imhist(P),title('Filtered Histogram');

%% Salt & peppar results
Med1 = medfilt2(salt);
figure(5),
subplot(2,2,1),imshow(salt),title('Original image');
subplot(2,2,2),imhist(salt),title('Original Histogram'
   );
subplot(2,2,3),imshow(Med1),title('Filtered image');
subplot(2,2,4),imhist(Med1),title('Filtered Histogram'
   );
```

```matlab
%% Quantization noise
Q = imgaussfilt3(quant);
figure(6),
subplot(2,2,1),imshow(quant),title('Original image');
subplot(2,2,2),imhist(quant),title('Original Histogram
    ');
subplot(2,2,3),imshow(Q),title('Filtered image');
subplot(2,2,4),imhist(Q),title('Filtered Histogram');
```