Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

Image Denoising

1. Abstract and Motivation :

(a) Gray-level image:

One of the most important consideration in digital processing of images is noise. In all processing systems we must consider how much of the detected signal can be regarded as true and how much is associated with random background events resulting from either the detection or transmission process. These are listed in a category called noise. This noise can result from a vast variety of sources, including the discrete nature of radiation, variation in detector sensitivity, photo-graphic grain effects, data transmission errors etc. In each case the properties of the noise are different, as are the image processing operations that can be applied to reduce their effects.

Few kinds of noises are shown in Fig 17:

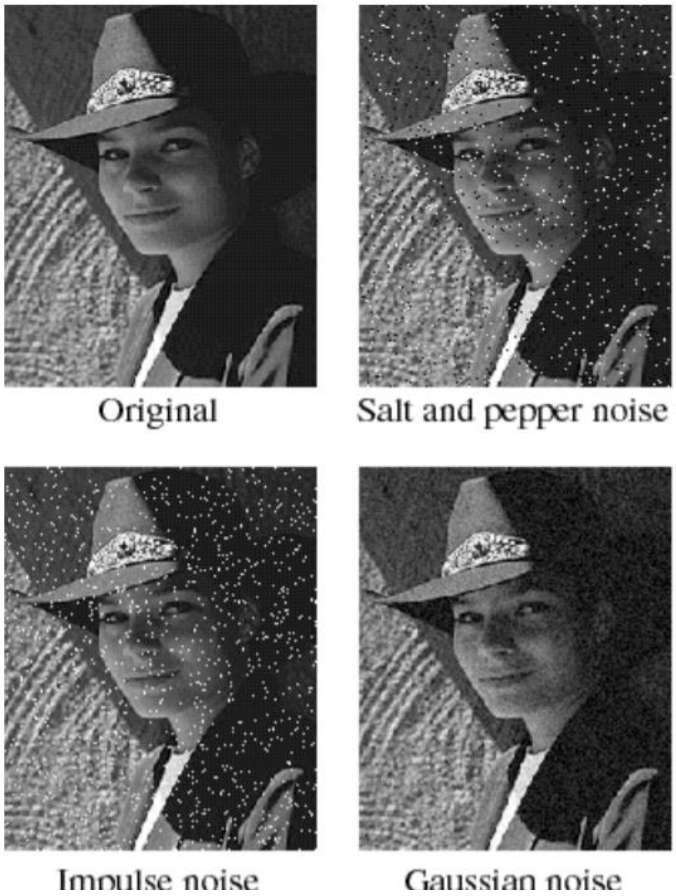

Figure 1: Noise types

Hence, denoising becomes very vital part of processing the initial image. The noise generated in the image is assumed to follow gaussian distribution with 0 mean. Say,

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

where are n noisy images and are noises that are assumed to have gaussian distribution with mean 0 and I is the real image.

We take the average of n noisy Images to get the real image and set n to be very large as shown:

Hence averaging over a large set of noise images will eventual denoise and give you the original image. There are different methods to achieve this:

(i) Linear filter:
Linear filter is a simple averaging filter that averages the neighboring pixels. A filter of size nxn is slid over all the pixels in the input image and convoluted. The convolution gives the average of each pixel in the nxn neighborhood. The linear filters for 3x3, 5x5, 7x7 are shown below:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49
1/49	1/49	1/49	1/49	1/49	1/49	1/49

(ii) Gaussian filter:
Gaussian filter considers the spatial aspect. Like the bell curve distribution of Gaussian, the weight coefficients decrease as it moves from center towards the edges. The typical distribution and equation is shown in Fig 18:

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

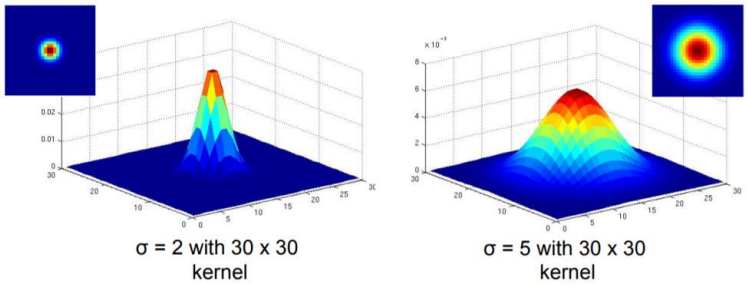

Figure 2: Gaussian distribution and formula

The gaussian filter for 5x5 is as shown below:

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

(iii) Bilateral filter:

A bilateral filter is another non-linear filter that evaluates the intensity of each pixel with a weighted average of intensity values from nearby pixels. The weights can be based on a Gaussian distribution. The formula for the bilateral is shown below:

$$Y(i,j) = \frac{\sum_{k,l} I(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_c^2} - \frac{\|I(i,j) - I(k,l)\|^2}{2\sigma_s^2}\right)$$

Where Y(i,j) is the ouput image at (i,j)
I – input image
spread parameters
k,l – neighborhood locations

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

(iv) Non Local Means:

It is another non linear filter which is different from other non linear filters in a way that it just doesn't only consider the neighboring pixels but considers all the pixels in the image weighted by gaussian weights to smooth a given pixel. The algorithm first takes a subwindow of larger size than the neighborhood window and compares neighborhood of every pixel in the sub window with the neighborhood of the pixel under consideration with gaussian weights until the subwindow covers all the pixels in the image. The algorithm is very much time consuming as the complexity is of O(n^4). This method also adds white noise to the image which doesn't degrade the image quality. The formula for NLM is as shown below:

$$Y(i,j) = \frac{\sum_{k=1}^{N'} \sum_{l=1}^{M'} I(k,l) f(i,j,k,l)}{\sum_{k=1}^{N'} \sum_{l=1}^{M'} f(i,j,k,l)}$$

$$f(i,j,k,l) = \exp\left(- \frac{\left\| I(N_{i,j}) - I(N_{k,l}) \right\|_{2,a}^2}{h^2} \right)$$

$$\left\| I(N_{i,j}) - I(N_{k,l}) \right\|_{2,a}^2 = \sum_{n_1, n_2 \in N} G_a(n_1, n_2)(I(i - n_1, j - n_2) - I(k - n_1, l - n_2))^2$$

and

$$G_a(n_1, n_2) = \frac{1}{\sqrt{2\pi}a} \exp\left(-\frac{n_1^2 + n_2^2}{2a^2}\right)$$

where f(i,j,k,l) defines the weights using neighborhood N(k,l) and N(i,j)

(b) Color Image:

Every algorithm explained above works for the color images but except for the fact that there are three channels to be taken care of. The color images can be denoised in one of following ways:

- Consider r, g and b channels separately and use filtering methods on each channel
- Transform the color image into another space, say YUV where Y being luminance is more sensitive to human eye compared to others. Use filtering methods in the Y channel and convert the image back into r,g,b color space.

Different ways work for different kinds of noises. It is up to experimental results that shows what best way could be used for given image.

Median filter:

Median filter is a very important filter to denoise an impulse noise. Impulse noise also called salt and pepper noise consists of black and white dots in the image. To eliminate this noise, median filter makes use of a median of neighborhood of the pixel as shown in fig 19

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

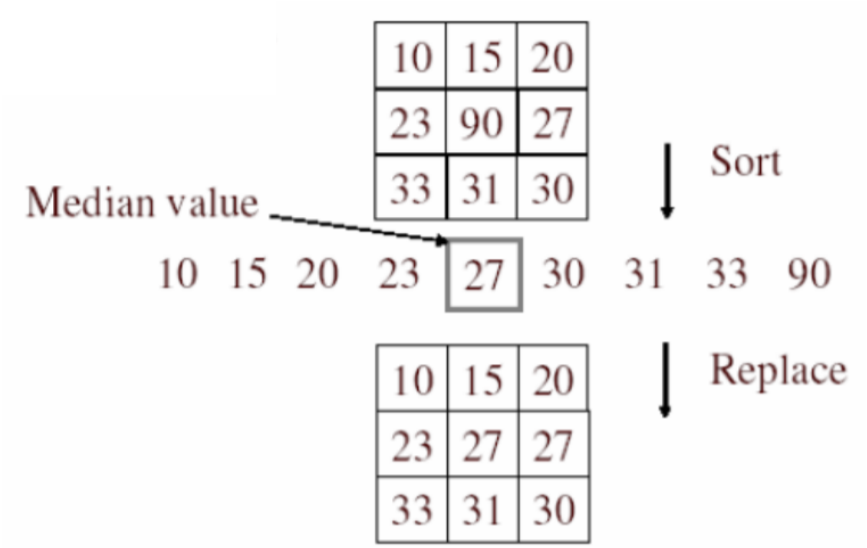

Figure 3: Median filter algorithm

(c) Shot noise

Oftentimes, the electronic devices that capture images tend to have large amount of noise in dark pixels. This kind of noise is called shot noise. Shot noise is assumed to follow Poisson distribution given by:

To handle shot noise, the Anscombe root transformation is done on input image which is given by:

This transforms the input image to be very dark and then any of the above mentioned filter methods are now used to denoise. The denoised image is then inverse transformed to get the denoised original image.

2. Implementation and approach:

(i) Linear filter:

- The input image pepper_uni.raw was read byte by byte into a three-dimensional array Imagedata of single channel.
- The linear filters were initializedto values shown above.
- A function for convolution was written that takes in arguments filter and Image and the position of the current pixel and returns the convolved pixel value at that location.
- The Imagedata was boundary extracted depending on the kernel size used: 2 extra row and cols for 3x3, 4 for 5x5 and 6 for 7x7 filters.
- Iterating through Imagedata, OutputImage was obtained by assigning the convolved value to it.
- OutputImage was written into raw file.

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

- The written raw file was read in the matlab to compare with the noise free image and the PSNR values for each was obtained.
- Outputs were observed and analyzed.

(ii) Gaussian filter:

- The input image pepper_uni.raw was read byte by byte into a three-dimensional array Imagedata of single channel.
- A function getGaussian was written to obtain Gaussian weights for given kernel size.
- The gaussian filters were initialized depending on n=3,5,7 and sigma =2.0.
- A function for convolution was written that takes in arguments filter and Image and the position of the current pixel and returns the convolved pixel value at that location.
- The Imagedata was boundary extracted depending on the kernel size used: 2 extra row and cols for 3x3, 4 for 5x5 and 6 for 7x7 filters.
- Iterating through Imagedata, OutputImage was obtained by assigning the convolved value to it.
- OutputImage was written into raw file.
- The written raw file was read in the matlab to compare with the noise free image and the PSNR values for each was obtained.
- Outputs were observed and analyzed.

(iii) Bilateral filter:

- The input image pepper_uni.raw was read byte by byte into a three-dimensional array Imagedata of single channel.
- A function for bilateral was written which takes in the arguments current pixel under consideration's location, Input image and returns the value of the expression given above of bilateral algorithm.
- The spread parameter was experimented for different values and observed the changes in the image.
- The Imagedata was boundary extracted for 4 extra rows and cols as the filter used in the bilateral uses 5x5 neighborhoods.
- Iterating through Imagedata, OutputImage was obtained by assigning the value returned by bilateral function for each pixel.
- OutputImage was written into raw file. Outputs were observed and analyzed.

(iv) NLM:

- The input image pepper_uni.raw was read byte by byte into a three-dimensional array Imagedata of single channel.
- A function for nlm was written which takes in the arguments current pixel under consideration's location, Input image and returns the value of the expression given above for nlm algorithm.
- Computing for all the pixels is very tedious. According to some research, a subwindow of 21x21 and neighborhood of 7x7 generally gace good experimental results. Hence, I have considered these values for nlm algorithm.

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

- Hence, the image was boundary extracted by 23 rows and 26 cols.
- The spread parameter was experimented for different values and observed the changes in the image.
- Iterating through Imagedata, OutputImage was obtained by assigning the value returned by nlm function for each pixel.
- OutputImage was written into raw file. Outputs were observed and analyzed.

(b) Color Image:

- The input image rose.raw was read byte by byte into a three-dimensional array Imagedata of three channels.
- The image contains uniform and impulse noise. Hence two cascading filters were tried.
- The first combination was Gaussian and median filter. Gaussian to remove uniform noise and median to remove impulse noise.
- Gaussian was implemented for kernel size of 5 for each of the channels just like mentioned in the gray-level gaussian filtering.
- Median filter was implemented for kernel size of 5 for each channels by the algorithm explained above for median filtering.
- The experiment was done with first implementing median filter and then gaussian filter.
- Results were observed and analyzed
- The second combination was NLM and Median filter.
- The NLM was implemented for each channel in the same way as done with gray level image with subwindow size 21x21 and neighborhood size 7x7. The median filter was repeated for kernel size of 5 just like above.
- A function for nlm was written which takes in the arguments current pixel under consideration's location, Input image and returns the value of the expression given above of nlm algorithm.
- The spread parameter was experimented for different values and observed the changes in the image.
- Iterating through Imagedata, OutputImage was obtained by assigning the value returned by bilateral function for each pixel.
- Experiment was also done to reverse the order of cascading median to nlm.
- OutputImage was written into raw file. Outputs were observed and analyzed.

(c) Shot Noise

- The input image pepper_dark_noise.raw was read byte by byte into a three-dimensional array Imagedata of single channel.
- Each pixel of the Imagedata was Ascombee root transformed using the equation described above.
- Two methods were implemented for denoising.
- Gaussian method was implemented with kernel size 5 just like the above description of Gaussian filter method by boundary extracting the transformed image by 4 rows and 4 cols.

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

- The denoised Image was later inverse transformed to get back the original denoised image.
- The output was written into a raw file. Results were observed and analyzed.
- The other method of denoising was done using BM3D.
- The BM3d zip file was installed and the readraw.m was run to read Ascombee root transformed image of InputImagedata.
- BM3D.m was run. The pixel values were multiplied by 255.
- The writeraw.m was run to write the denoised image into a raw file.
- The image was then read into C++ source file to inverse transform the data to get original denoised Image.
- Output was written into a raw file. Results were observed and analyzed.

3. Discussion:

(1) Linear filter:

The input and output image for pepper_uni for linear filter of size 3, 5 and 7 are shown below:

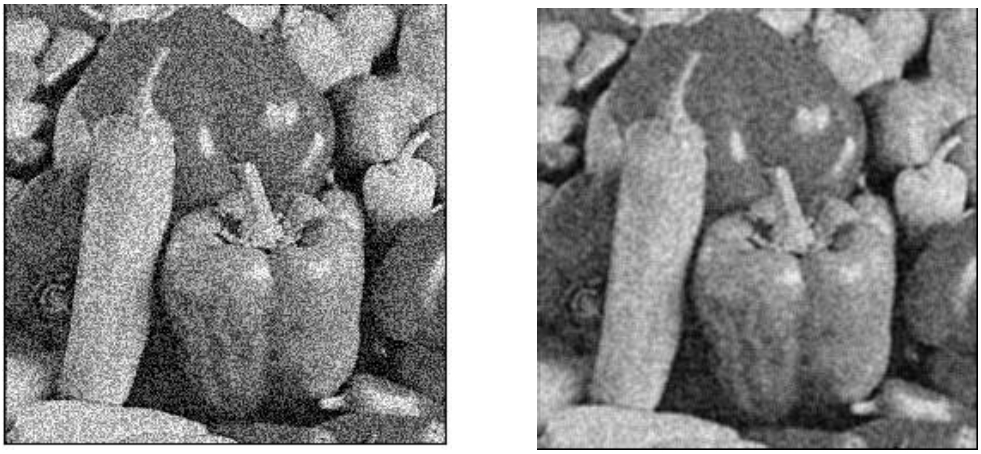

Figure 4: (a): Input image for pepper_uni (b)Output image for pepper_uni with linear 3x3 filter

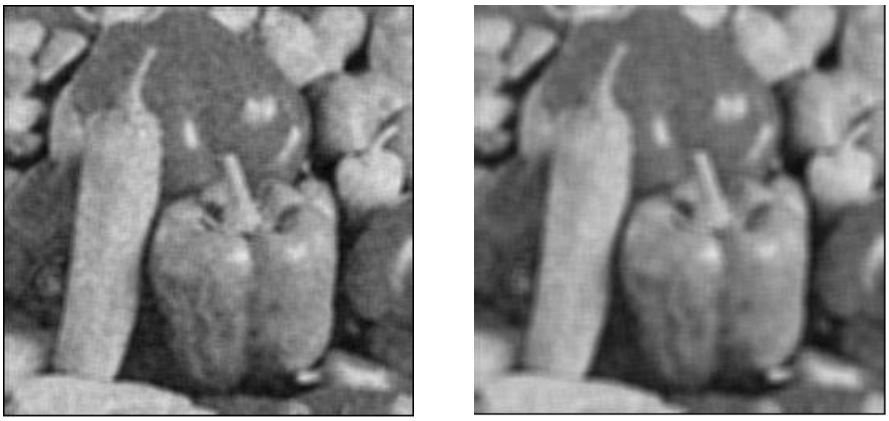

Figure 5 (a): Output image for pepper_uni with linear 5x5 filter (b)Output image for pepper_uni with linear 7x7 filter

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

The PSNR values are shown in the table:

Filter Size	PSNR
3	24.5796
5	24.7837
7	23.9677

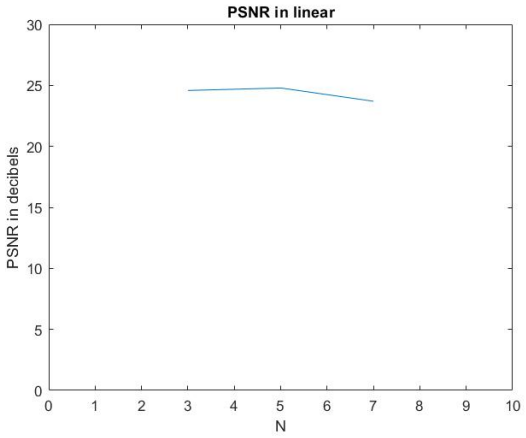

Figure 6:PSNR value for linear filter against kernel sizes

The outputs for gaussian filter for kernel sizes 3,5 and 7 are shown below:

Figure 7(a): Gaussian kernal 3x3 (b): Gaussian kernel 5x5, (c): Gaussian kernel 7x7

The PSNR values are shown in the table:

Filter Size	PSNR
3	24.6591
5	25.2831
7	24.8676

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

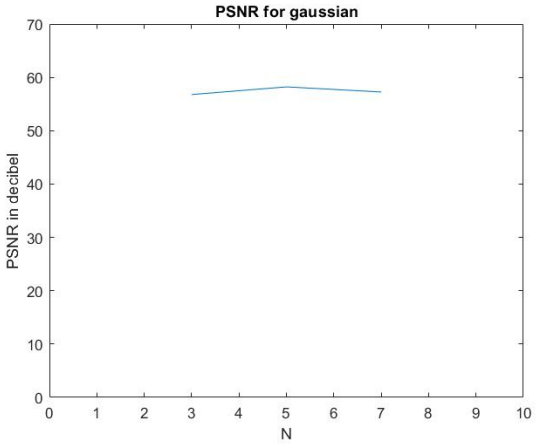

Figure 8:PSNR for gaussian

(iii) Bilateral:

The outputs for Bilateral filter is shown below:

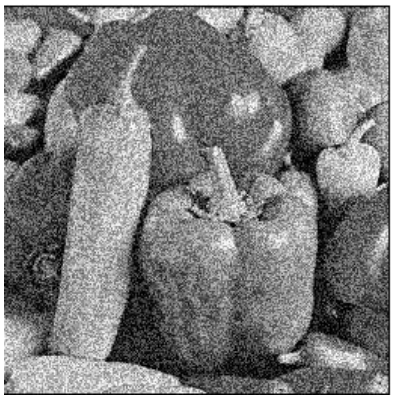

Figure 9: Output image for pepper_uni with bilateral

No significant improvement even with large variations in spread parameters.

(iv) Non Local Means:

The output for NLM is shown below:

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

Figure 10: Output Image for pepper_uni with nlm

Oversmoothing of the image.

(b) Color Image:

Figure 11:(a): Original color rose image, (b): Outputimage with gauiian + median cascading, (c): Output image for nlm + median cascading

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019


Figure 12: (a): Noisy color rose image, (b): Output image for median + nlm cascading

- The filtering was done on each channels of the color image.
- The gaussian/nlm filter was used to remove uniform noise and median filter was used to remove impulse noise.
- There is a little difference as can be seen from the images above with reversing the order of cascading. The image is dull when median is first implemented and then nlm. This is because both the filters are not linear so that linearity holds.
- Other approaches is to transform given image into whole new color space where dominant channel can be used to filter out the noise and transformed back to r,g,b, space.

(c) Shot Noise:


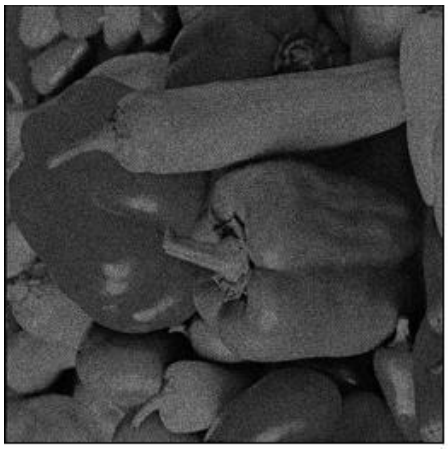

Figure 13(a): Original pepper_dark image, *(b): Noisy pepper_dark image*

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

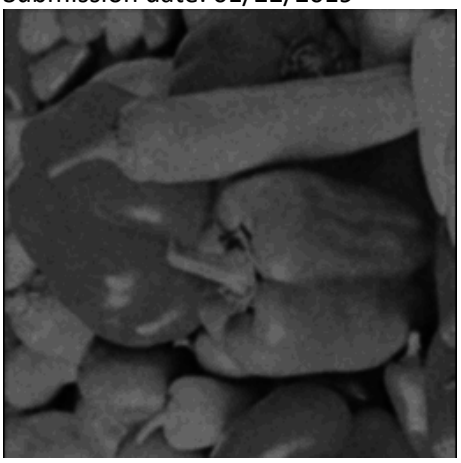

Figure 14(a): Denoise with gaussian, *(b): Denoise with BM3D*

Akash Mohan Das
USC Id: 7247503828
USC email: amohanda@usc.edu
Submission date: 01/22/2019

References:

1. Wikipedia

2. Lecture/ Discussion notes

3. Paper publications on denoising / demosaicing