EE569: HOMEWORK #1

Issued: 01/07/2019 Due: 01/22/2019

Problem 1: Image Demosaicing and Histogram Manipulation (50%)

1.1 Motivation

Digital camera sensors at a pixel location detects the frequency of one of the primary color (Red/Green/Blue channel) depending on the Color Filter Array (CFA). It is vital to reconstruct the original image by utilizing the neighbouring pixel information/values. The missing colour value at each pixel is approximated by simplest demosaicing method based on bilinear interpolation using the average of two or four adjacent pixels of interest depending of the Bayer array/pattern. Image resizing is implemented using bilinear interpolation. To reconstruct an image with higher quality, Malvar-He-Cutler (MHC) demosiacing method is made use by adding a 2nd order cross channel correction term to the basic bilinear demosaicing result. To enhance the contrast of an image, histogram equalization techniques are essential. Transfer function based and bucket filling method are investigated.

1.2 Approach

To find the missing color value at each pixel, bilinear interpolation technique is adopted. Given the dimensions of an image, the boundary of an image is extended by reflecting the rows(columns) across the first row(column) and last row(column). Analyzing the Bayer array/pattern, each missing color value is approximated by the rule of generalization using bilinear interpolation. Let i, j be the index for rows and columns of an image respectively.

1. Estimation of Red and Blue intensity values at Green location

$G_{i-1, j-1}$	$B_{i-1,j}$	$G_{i-1,j+1}$
$R_{i, j-1}$	$G_{i,j}$	$R_{i, j+1}$
$G_{i+1, j-1}$	$B_{i+1, j}$	$G_{i+1, j+1}$

$R_{i,j} = 0.5 * [R_{i,j-1} + R_{i,j+1}]$
$B_{i,j} = 0.5 * [B_{i-1,j} + B_{i+1,j}]$

2. Estimation of Green and Blue intensity values at Red location

$B_{i-1, j-1}$	$G_{i-1,j}$	$B_{i-1,j+1}$
$G_{i,j-1}$	$R_{i, j}$	$G_{i, j+1}$
$B_{i+1, j-1}$	$G_{i+1,j}$	$B_{i+1,j+1}$

$G_{i,j} = 0.25 * [G_{i-1,j} + G_{i,j-1} + G_{i,j+1} + G_{i+1,j}]$
$B_{i,j} = 0.25 * [B_{i-1,j-1} + B_{i-1,j+1} + B_{i+1,j-1} + B_{i+1,j+1}]$

3. Estimation of Red and Blue intensity values at Green Location

$G_{i-1,j-1}$	$R_{i-1, j}$	$G_{i-1,j+1}$
$B_{i,j-1}$	$G_{i,j}$	$B_{i,j+1}$
$G_{i+1,j-1}$	$R_{i+1, j}$	$G_{i+1,j+1}$

$R_{i,j} = 0.5 * [R_{i-1,j} + R_{i+1,j}]$
$B_{i,j} = 0.5 * [B_{i,j-1} + B_{i,j+1}]$

4. Estimation of Green and Red intensity values at Blue location

$R_{i-1,j-1}$	$G_{i-1,j}$	$R_{i-1,j+1}$
$G_{i,j-1}$	$B_{i,j}$	$G_{i,j+1}$
$R_{i+1,j-1}$	$G_{i+1,j}$	$R_{i+1,j+1}$

$G_{i,j} = 0.25 * [G_{i-1,j} + G_{i,j-1} + G_{i,j+1} + G_{i+1,j}]$

$R_{i,j} = 0.25 * [R_{i-1,j-1} + R_{i-1,j+1} + R_{i+1,j-1} + R_{i+1,j+1}]$

To improve the quality of a reconstructed image, Malvar-He-Cutler (MHC) demosiacing method can be implemented. α, β and Υ are the weights which control the 2nd order correction factor applied to red, green and blue channel.

1. To estimate red and blue component at green location

		$G_{i-2,j}$		
	$G_{i-1,j-1}$	$B_{i-1,j}$	$G_{i-1,j+1}$	
$G_{i,j-2}$	$R_{i,j-1}$	$G_{i,j}$	$R_{i,j+1}$	$G_{i,j+2}$
	$G_{i+1,j-1}$	$B_{i+1,j}$	$G_{i+1,j+1}$	
		$G_{i+2,j}$		

$R_{i,j} = 0.5 * [R_{i,j-1} + R_{i,j+1}] + \beta * [G_{i,j} + 0.1 * G_{i-2,j} + 0.1 * G_{i+2,j} - 0.2 * G_{i-1,j-1} - 0.2 * G_{i-1,j+1} - 0.2 * G_{i+1,j-1} - 0.2 * G_{i+1,j+1} - 0.2 * G_{i,j-2} - 0.2 * G_{i,j+2}]$

$B_{i,j} = 0.5 * [B_{i-1,j} + B_{i+1,j}] + \beta * [G_{i,j} - 0.2 * G_{i-2,j} - 0.2 * G_{i+2,j} - 0.2 * G_{i-1,j-1} - 0.2 * G_{i-1,j+1} - 0.2 * G_{i+1,j-1} - 0.2 * G_{i+1,j+1} + 0.1 * G_{i,j-2} + 0.1 * G_{i,j+2}]$

2. Estimation of Green and Blue intensity values at Red location

		$R_{i-2,j}$		
	$B_{i-1,j-1}$	$G_{i-1,j}$	$B_{i-1,j+1}$	
$R_{i,j-2}$	$G_{i,j-1}$	$R_{i,j}$	$G_{i,j+1}$	$R_{i,j+2}$
	$B_{i+1,j-1}$	$G_{i+1,j}$	$B_{i+1,j+1}$	
		$R_{i,+2\ j}$		

$G_{i,j} = 0.25 * [G_{i-1,j} + G_{i,j-1} + G_{i,j+1} + G_{i+1,j}] + \alpha * [R_{i,j} - 0.25 * R_{i-2,j} - 0.25 * R_{i+2,j} - R_{i,j+2} - R_{i,j-2}]$

$B_{i,j} = 0.25 * [B_{i-1,j-1} + B_{i-1,j+1} + B_{i+1,j-1} + B_{i+1,j+1}] + \alpha * [R_{i,j} - 0.25 * R_{i-2,j} - 0.25 * R_{i+2,j} - 0.25 * R_{i,j-2} - 0.25 * R_{i,j+2}]$

3. Estimation of Red and Blue intensity values at Green Location

		$G_{i-2,j}$		
	$G_{i-1,j-1}$	$R_{i-1,j}$	$G_{i-1,j+1}$	
$G_{i,j-2}$	$B_{i,j-1}$	$G_{i,j}$	$B_{i,j+1}$	$G_{i,j+2}$
	$G_{i+1,j-1}$	$R_{i+1,j}$	$G_{i+1,j+1}$	
		$G_{i+2,j}$		

$R_{i,j} = 0.5 * [R_{i-1,j} + R_{i+1,j}] + \beta * [G_{i,j} - 0.2 * G_{i-2,j} - 0.2 * G_{i+2,j} - 0.2 * G_{i-1,j-1} - 0.2 * G_{i-1,j+1} - 0.2 * G_{i+1,j-1} - 0.2 * G_{i+1,j+1} + 0.1 * G_{i,j-2} + 0.1 * G_{i,j+2}]$

$B_{i,j} = 0.5 * [B_{i,j-1} + B_{i,j+1}] + \beta * [G_{i,j} + 0.1 * G_{i-2,j} + 0.1 * G_{i+2,j} - 0.2 * G_{i-1,j-1} - 0.2 * G_{i-1,j+1} - 0.2 * G_{i+1,j-1} - 0.2 * G_{i+1,j+1} - 0.2 * G_{i,j-2} - 0.2 * G_{i,j+2}]$

4. Estimation of Green and Red intensity values at Blue location

		$B_{i-2,j}$		
	$R_{i-1,j-1}$	$G_{i-1,j}$	$R_{i-1,j+1}$	
$B_{i,j-2}$	$G_{i,j-1}$	$B_{i,j}$	$G_{i,j+1}$	$B_{i,j+2}$
	$R_{i+1,j-1}$	$G_{i+1,j}$	$R_{i+1,j+1}$	
		$B_{i+2,j}$		

$$G_{i,j} = 0.25 * [G_{i-1,j} + G_{i,j-1} + G_{i,j+1} + G_{i+1,j}] + \Upsilon * [B_{i,j} - 0.25*B_{i-2,j} - 0.25* B_{i+2,j} - B_{i,j+2} - B_{i,j-2}]$$

$$R_{i,j} = 0.25 * [R_{i-1,j-1} + R_{i-1,j+1} + R_{i+1,j-1} + R_{i+1,j+1}] + \Upsilon * [B_{i,j} – 0.25* B_{i-2,j} - 0.25* B_{i+2,j} - 0.25* B_{i,j-2} -0.25* B_{i,j+2}]$$

To enhance the contrast of an image, two methods are adopted.

1. Transfer function based histogram equalization method: The distribution of occurrences of pixel against the pixel intensity is to be distributed uniformly, so that the concentrated region of distribution spreads out. One to one mapping between the cumulative distribution of uniform probability density function and the transfer function of the image is performed to obtain enhanced intensities.
2. Cumulative Probability based equalization method: It is the discrete version of transfer function based equalization method. The histogram is arranged into long chain of sequence. Total number of pixels in an image of size N is N^2, the long chain is partitioned into 256 subsets with fixed number of occurrence of pixel intensity.

1.3 Result

1(a)(1) Shown below are the result after applying bilinear demosaicing to the cat image.

Figure 1.1: cat.raw

Figure 1.2: Bilinear demosaicing applied to cat.raw

1(b)(1) Shown below are the result after applying Malvar-He-Cutler (MHC) demosaicing to the cat image

Figure 1.3: cat.raw

Figure 1.4: MHC demosaicing applied to cat.raw

*Original Cat Image for reference:

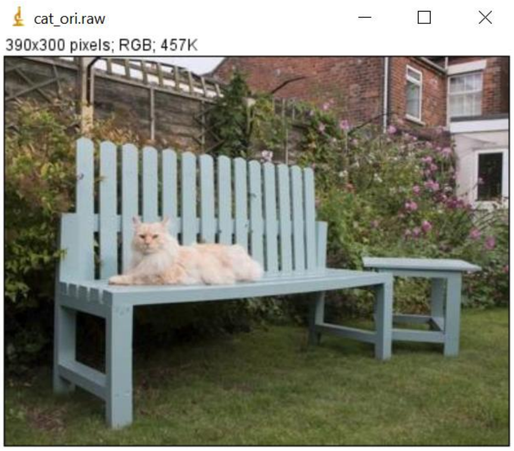


Figure 1.5: cat_ori.raw

1(c)(1) Experimental results are as follows:

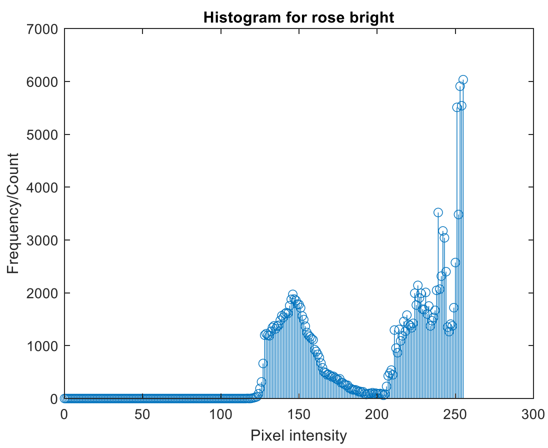


Figure 1.6 Histogram of *rose_bright* image

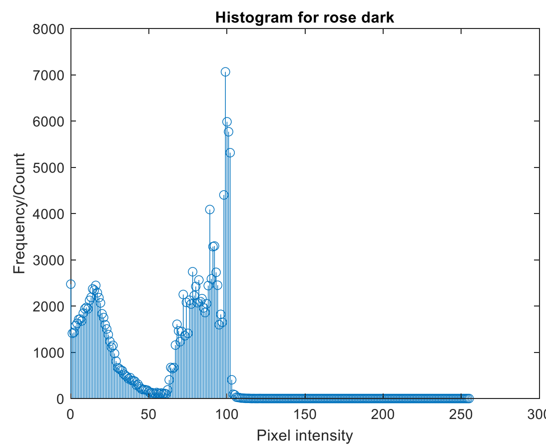


Figure 1.7 Histogram of *rose_dark* image

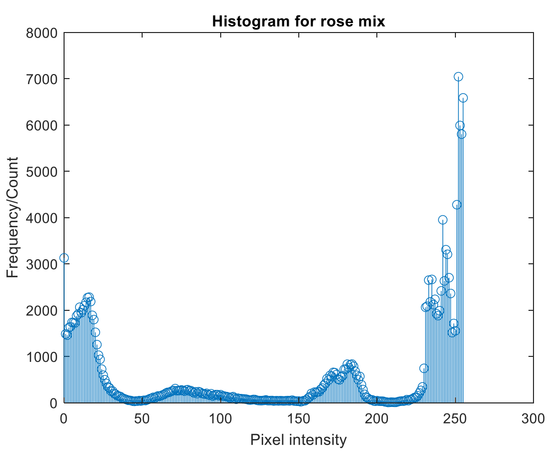


Figure 1.8 Histogram of *rose_mix* image

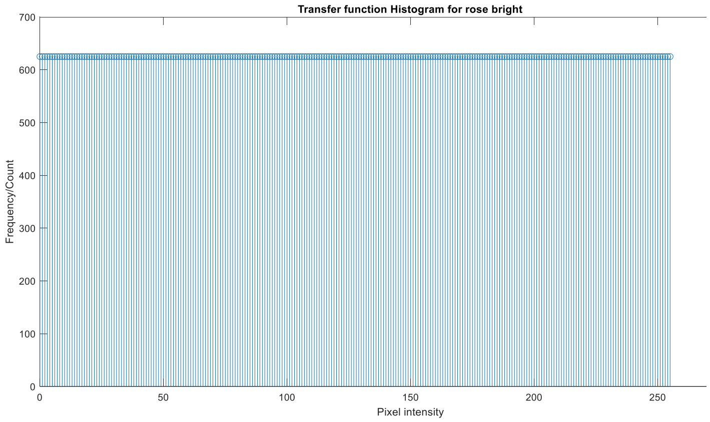


Fig 1.9 Histogram for *rose_bright* image after applying Method A

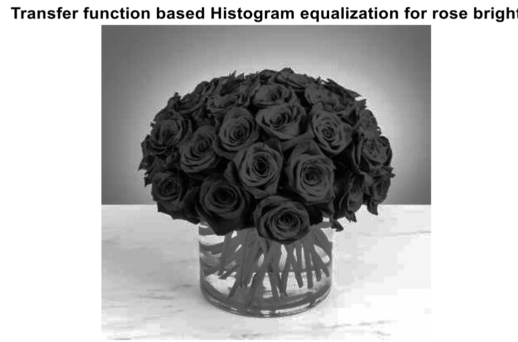


Fig 2. Enhanced image of *rose_bright* image after applying Method A

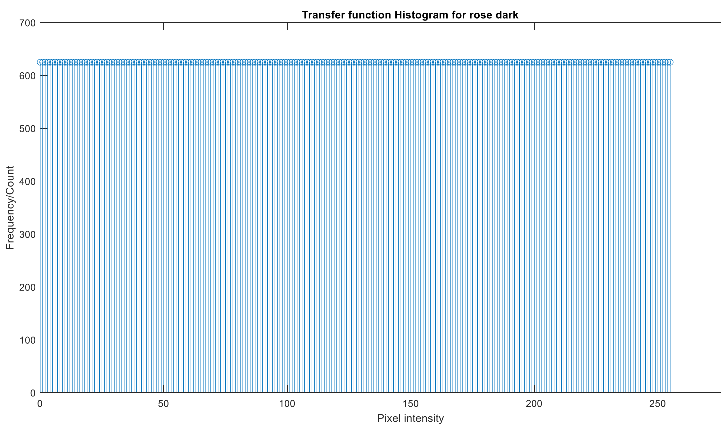


Fig 2.1 Histogram for *rose_dark* image after applying Method A

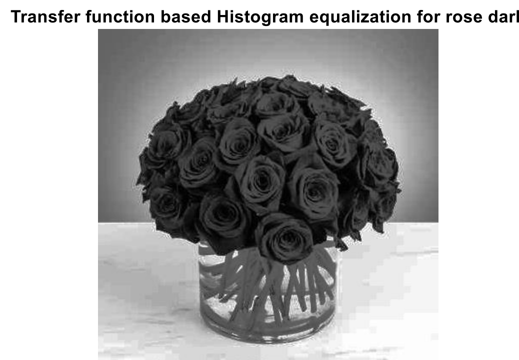


Fig 2.2 Enhanced image of *rose_dark* image after applying Method A

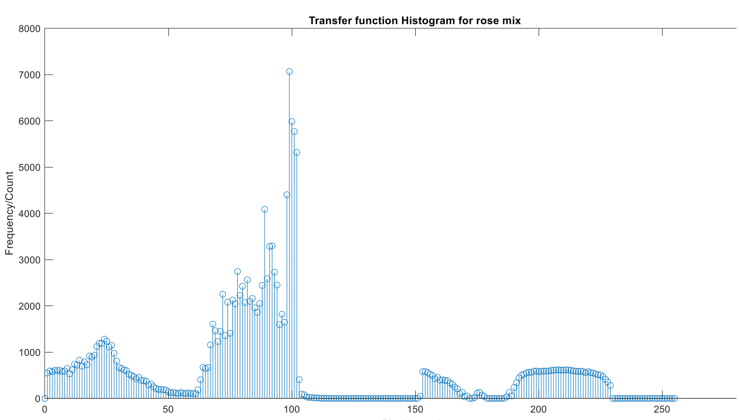


Fig 2.3 Histogram for *rose_mix* image after applying Method A

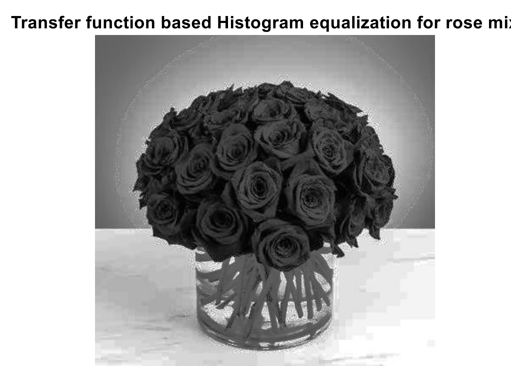


Fig 2.4: Enhanced image of *rose_mix* image after applying Method A

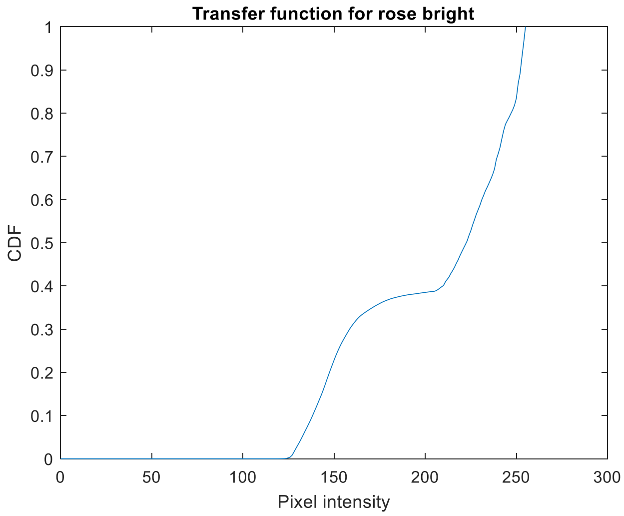


Fig 2.5: Transfer function for *rose_bright* image

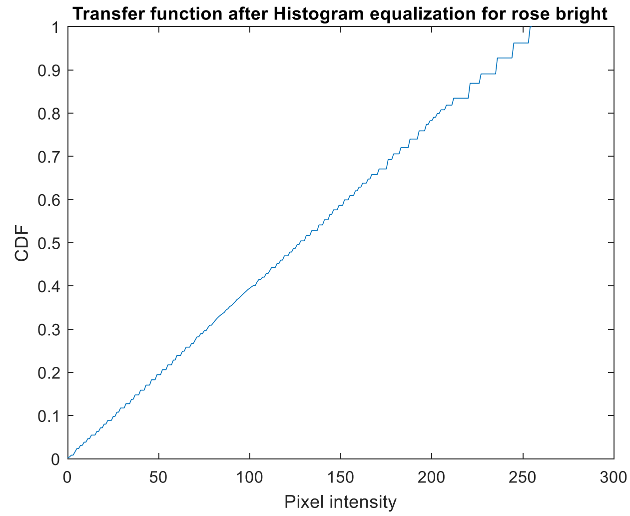


Fig 2.6: Transfer function for *rose_bright* image after applying Method A

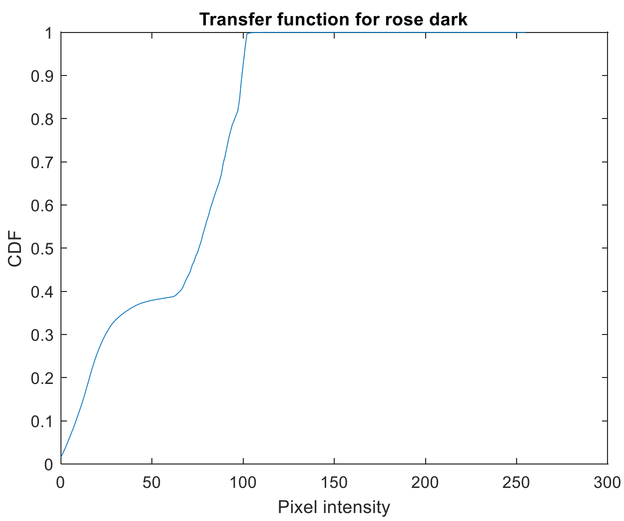


Fig 2.7: Transfer function for *rose_dark* image

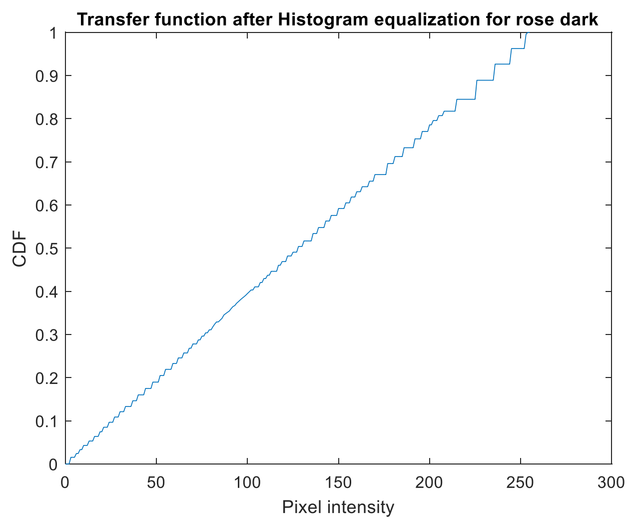


Fig 2.8: Transfer function for *rose_dark* image after applying Method A

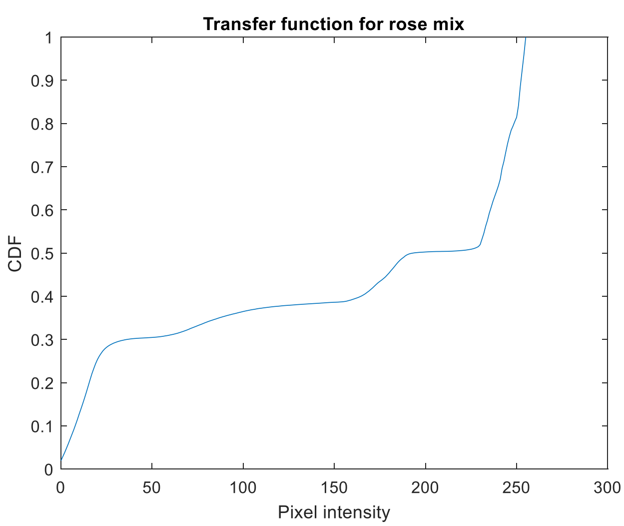


Fig 2.9: Transfer function for *rose_mix* image

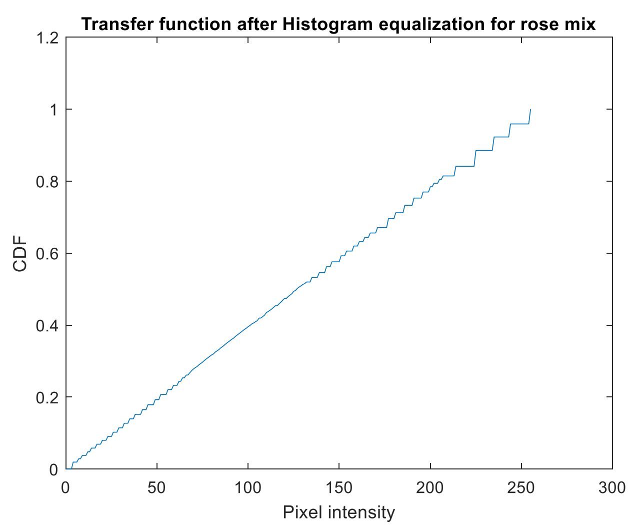


Fig 3: Transfer function for *rose_bright* image after applying Method A

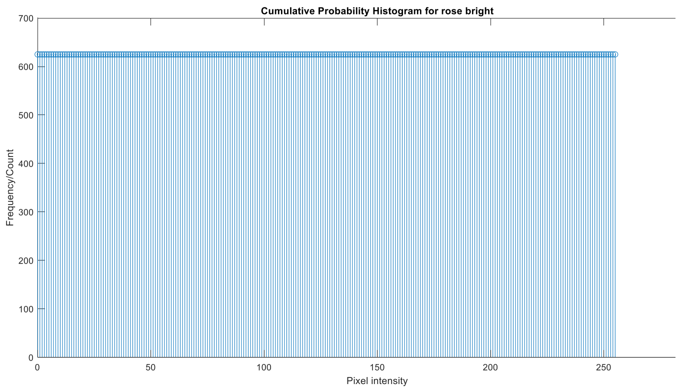


Fig 3.1: Histogram for *rose_bright* image after applying Method B

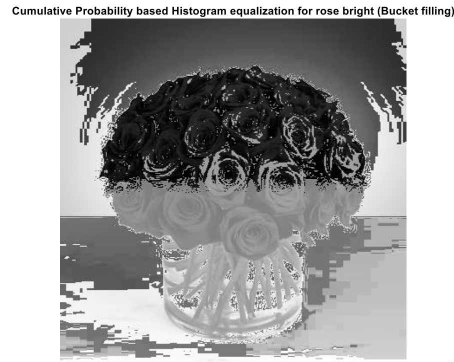


Fig 3.2: Transfer function for *rose_bright* image after applying Method B

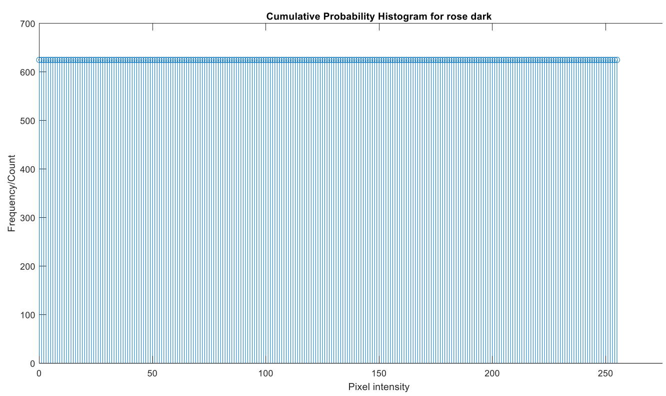


Fig 3.3: Histogram for *rose_dark* image after applying Method B

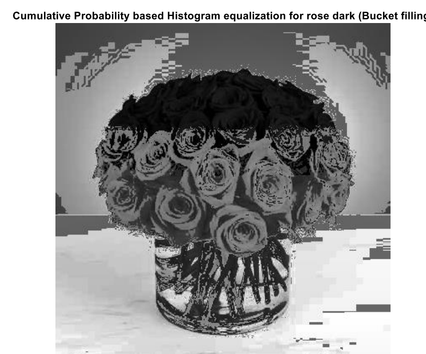


Fig 3.4: Transfer function for *rose_dark* image after applying Method B

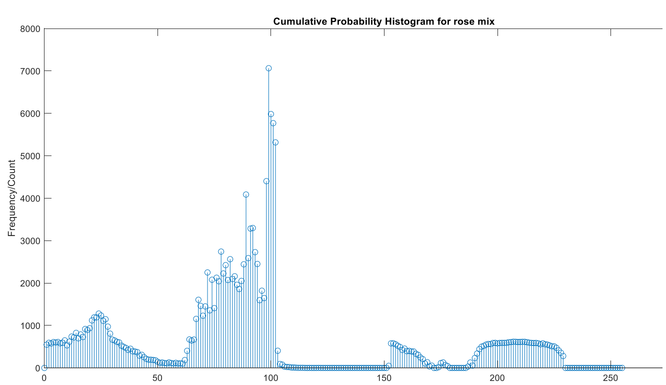


Fig 3.5: Histogram for *rose_mix* image after applying Method B

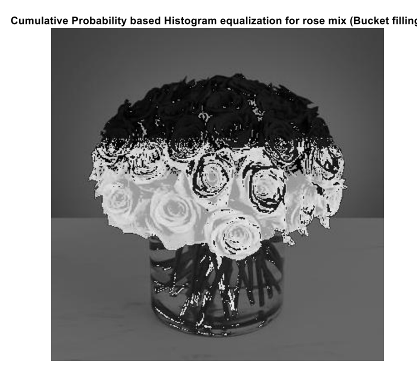


Fig 3.6: Transfer function for *rose_mix* image after applying Method B

1.4 Discussion

1(a)(2) The artifacts observed when bilinear demosaicing is applied are:

1. Zipper Effect: Along the edges of a region in the image, alternating pattern are visible. Figure 3.7a, 3.7b shows the edges of the bench degraded by zipper effects of bilinear interpolation.

Fig 3.7 a: Zipper effect along the edges of bench

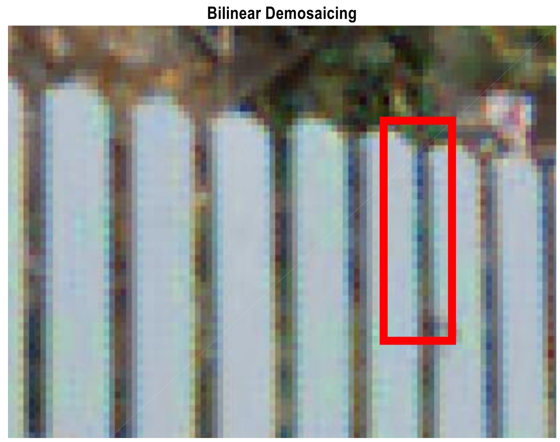


Fig 3.7 b: Highlighted zipper effect version of the edges

Zipper effect are caused along the edges of a region, due to inappropriate averaging along the boundary, where the pixel intensity difference along the neighbourhood are too high. To improve the demosaicing performance, weighted averaging depending on the neighborhood pixel intensity can be recommended, resulting in edge/object boundary dedection.

2. False Color effect – The color which are present in demosaiced image but not in the original image is termed false color effect. Color intenity difference (tint) between the original and the demosaiced image is due to the combination of red,green and blue planes.Figure 3.8a, 3.8b shows tint between background color of the bench in the original and demosaiced image. To remove the false color effect, correlation between red,blue and green planes is to be made use of.

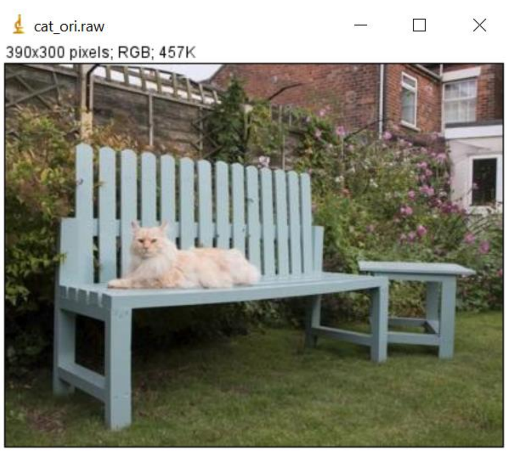


Fig 3.8 a: Original cat.raw

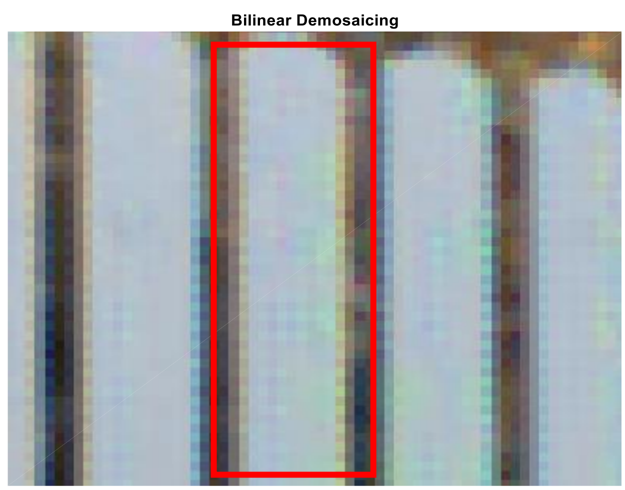


Fig 3.8 b: Highlighted false color efffect

1(b)(2) Compare the MHC and the bilinear demosaicing results and explain the performance differences between these two algorithms in your own words.

Below are the experimental results of MHC and bilinear interpolation results:

Fig 3.9: cat.raw

Fig 4: cat_ori.raw

Fig 4.1: Bilinear demosaicing applied to cat.raw

Fig 4.2: Malvar-He-Cutler demosaicing applied to cat.raw

Bilinear interpolation and Malvar-He-Cutler (MHC) demosaicing are simplest models for reconstruction of original image from the raw file. The performance difference between the two algorithms are as follows:

- The computational complexity of Malvar-He-Cutler (MHC) demosaicing is higher than bilinear interpolation because of the 2^{nd} order cross channel correction factor applied to bilinear interpolation results of red, green and blue channels.
- The quality of the image is higher in MHC than in bilinear interpolation, since the shortcomings of bilinear interpolation as mentioned in 1(a)(2) of section 1.4 is overcome by adopting Malvar-He-Cutler (MHC) demosaicing which uses weighted averaging.
- The distribution of the luminosity is linear in bilinear interpolation, while MHC captures a better curve than bilinear. Thus, the distribution of the luminosity is better captured in case of MHC in contrast to bilinear interpolation according to figure 4.3-4.5.

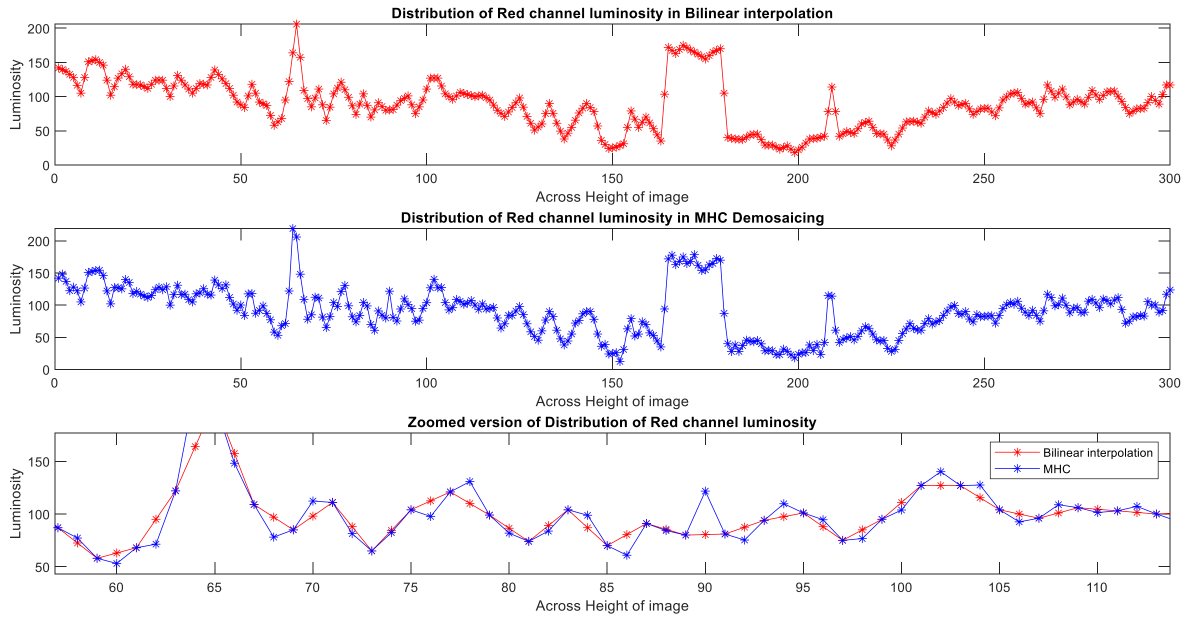


Fig 4.3: Distribution of Red channel luminosity of bilinear interpolation and MHC

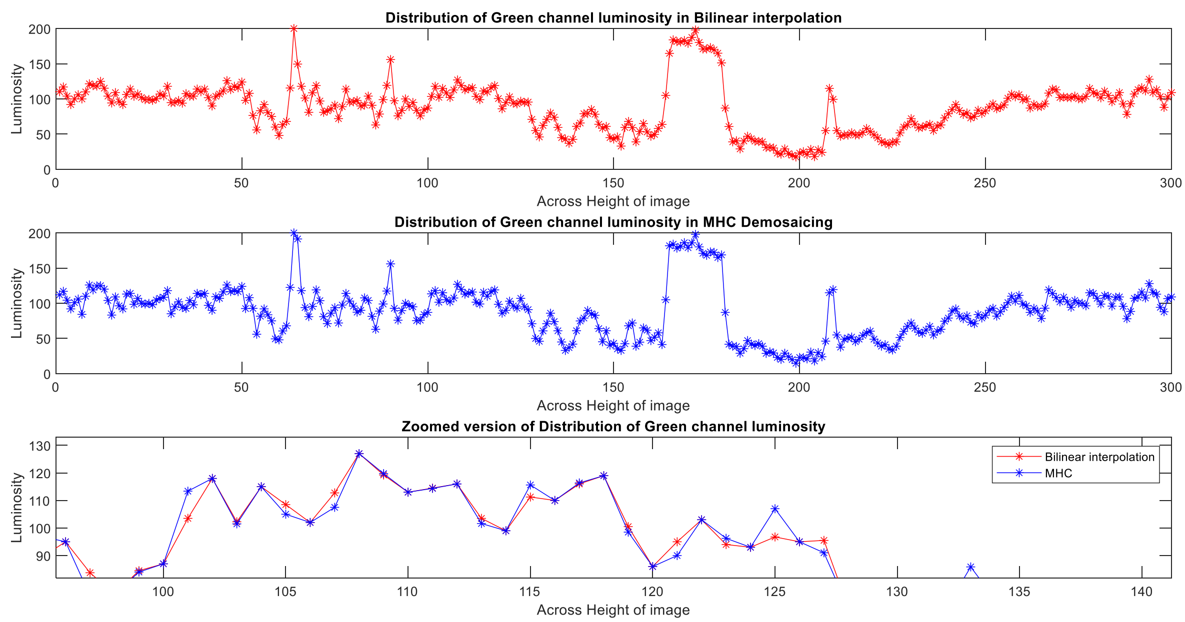


Fig 4.4: Distribution of Green channel luminosity of bilinear interpolation and MHC

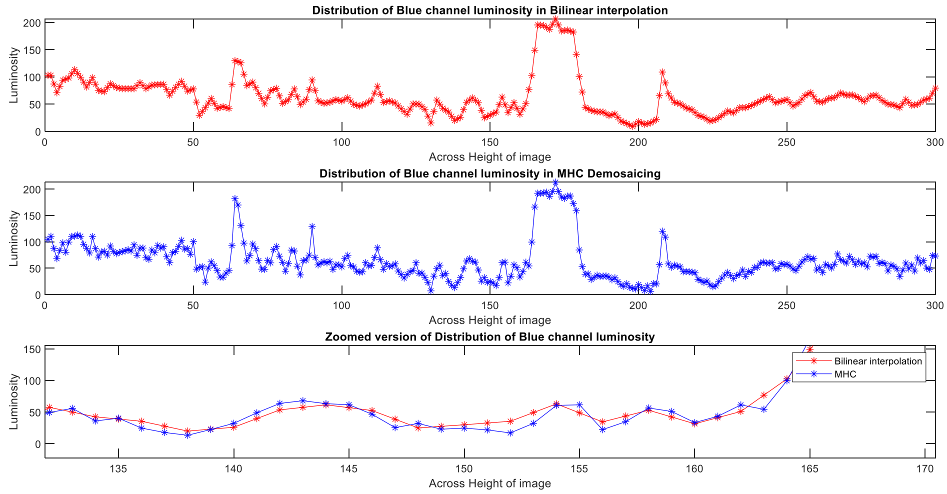


Fig 4.5: Distribution of Blue channel luminosity of bilinear interpolation and MHC

1(c)(4)

Histogram equalization redistributes the pixel occurrences evenly. It is used to enhance the contrast of an image.

- Transfer Function based histogram equalization method makes use of cumulative distribution function of uniform probability density function to enhance the pixel intensity values based on the mapping rule.
- Cumulative Probability based histogram equalization method partitions the linear chain of occurrences of pixel intensity values into $N^2/256$ pixels per bucket/bin.
- Transfer function based histogram spreads out the concentrated region of distribution, thus causing non-uniform jumps.
- Cumulative Probability based histogram equalization method (Bucketization) does not consider the order/position of the pixel, whose intensity values are changed. Hence, there is no uniqueness in the enhanced image. For a single image, many contrasted images can be obtained by this method as shown in Fig 4.6.
- To improve Transfer function based histogram, instead of distributing near by or local neighborhood pixel intensities, it is better to redistribute the highest occurrence of any pixel intensity to many other occurrences causing many to one mapping.
- To improve Cumulative Probability based histogram equalization and to retain uniqueness in the enhanced/contrast image, probability or weighted functions needs to be introduced.

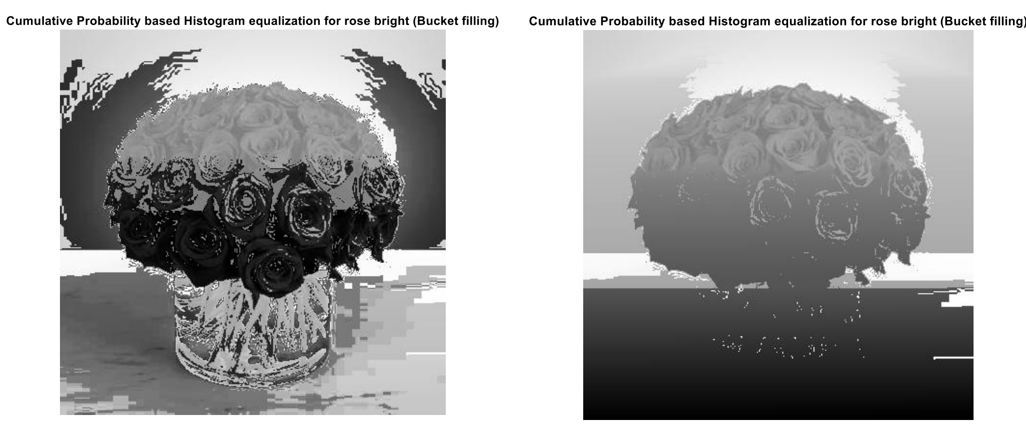


Fig 4.6: Two Cumulative Probability based histogram (Bucketization) applied on the same *rose_bright* image

1(c)(5) Similar results cannot be obtained from method A and B , since *rose_mix* has high occurrences of dark and bright pixels. Modification is by redistributing the pixels which have high pixel occurrences.

Problem 2: Image Denoising

2.1 Motivation

Images are corrupted at different stages or levels of processing. Noise may be due to dead/saturated sensors or due to any other environmental conditions. To remove the noise embedded in an image, several filters like uniform, gaussian, bilateral, non-local mean, BM3D are used. If Noise is assumed to be a normal distribution with 0 mean ensamples, samples can be stacked to remove the noise according to Law of Large numbers.
If there is are not many samples, then the information can be gathered from neighbouring pixel information if the surface is smooth else with far away pixels which are highly correlated pixel of interests (non-local mean).
Shot noise is the poisson distributed noise which depends on the original pixel information caused by photon fluctuation.

2.2 Approach

To remove noise of high frequency bands, low pass filters are used. Uniform, Gaussian, bilateral, non-local mean, BM3D are uniform noise removal filters.
Uniform: low pass filter have unit weights across its neighbouring pixel of 3x3 filter size, where Y(*i,j)* is the weighted average at each pixel of an image.

1	1	1
1	1	1
1	1	1

3 X 3 Kernel

$$Y(i,j) = \frac{\sum_{k,l} I(k,l)\, w(i,j,k,l)}{\sum_{k,l\ w1} w(i,j,k,l)} \qquad w(i,j,k,l) = \frac{1}{w1 \times w2} \tag{1}$$

Gaussian: low pass filter with weights defined for each neighbouring pixel of *N*x*N* filter size, where Y(*i,j)* is the weighted average at each pixel of an image.

$$Y(i,j) = \frac{\sum_{k,l} I(k,l)\, w(i,j,k,l)}{\sum_{k,l\ w1} w(i,j,k,l)} \qquad w(i,j,k,l) = \frac{1}{\sqrt{2\pi}}\, e^{\left(\frac{-(k-i)^2 + (l-j)^2}{2\sigma^2}\right)} \tag{2}$$

Bilateral filter: weights are defined according to *w(i,j,k,l)* for each neighbouring pixel of *N*x*N* filter size, where Y(*i,j)* is the weighted average at each pixel of an image.

$$Y(i,j) = \frac{\sum_{k,l} I(k,l)\, w(i,j,k,l)}{\sum_{k,l\ w1} w(i,j,k,l)} \qquad w(i,j,k,l) = \frac{1}{\sqrt{2\pi}}\, e^{\left(\frac{-(i-k)^2 + (j-l)^2}{2\sigma^2} - \frac{||I(i,j)-I(k,l)||^2}{2\sigma s^2}\right)} \tag{3}$$

Non-Local mean filter: weights are defined according to *f(i,j,k,l)* for each neighbouring pixel of *N'*x*M'* filter size, where Y(*i,j)* is the weighted average at each pixel of an image. It incorporates the measure of correlation between the far away pixels of interest(similarity).

$$Y(i,j) = \frac{\sum_{k=1}^{N'} \sum_{l=1}^{M'} I(k,l)\, f(i,j,k,l)}{\sum_{k=1}^{N'} \sum_{l=1}^{M'} f(i,j,k,l)} \qquad w(i,j,k,l) = e^{\left(\frac{||I(N(i,j)-I(N(k,l))||^2}{h^2}\right)} \tag{4}$$

$$||I(N(i,j)-I(N(k,l))||^2 = \sum_{n1,n2} Ga(n1,n2)(I(i-n1,j-n2)-I(k-n1,l-n2))^2 \tag{5}$$

$$G(n1,n2) = \frac{1}{a\sqrt{2\pi}}\, e^{\left(-\frac{n1^2 + n2^2}{2}\right)} \tag{6}$$

To remove the poisson distributed shot noise, Anscombe root transformation is applied on each pixel z, to stabilize the noise variance and convert it into additive Gaussian noise of unit variance in order to achieve signal independent noise.

$$D = f(z) = 2\sqrt[2]{z + 3/8} \quad (7)$$

Any denoising filter like uniform, gaussian, bilateral, non-local mean, BM3D can be applied. Anscombe inverse transform is applied to remove the shot noise and to obtain the denoised image.

$$\text{Biased } z = f^{-1}(D) = \left({}^{D}/_{2}\right)^{2} - {}^{3}/_{8} \quad (8)$$

$$\text{Unbiased } z = f^{-1}(D) = \left({}^{D}/_{2}\right)^{2} - {}^{1}/_{8} \quad (9)$$

Biased inverse transform can be directly obtained from equation (7)

2.3 Results

2(a)(1) Uniform is the noise embedded in the image pepper_uni.raw. Below fig 4.7 showcases the CDF of uniform noise [-64 to 64].

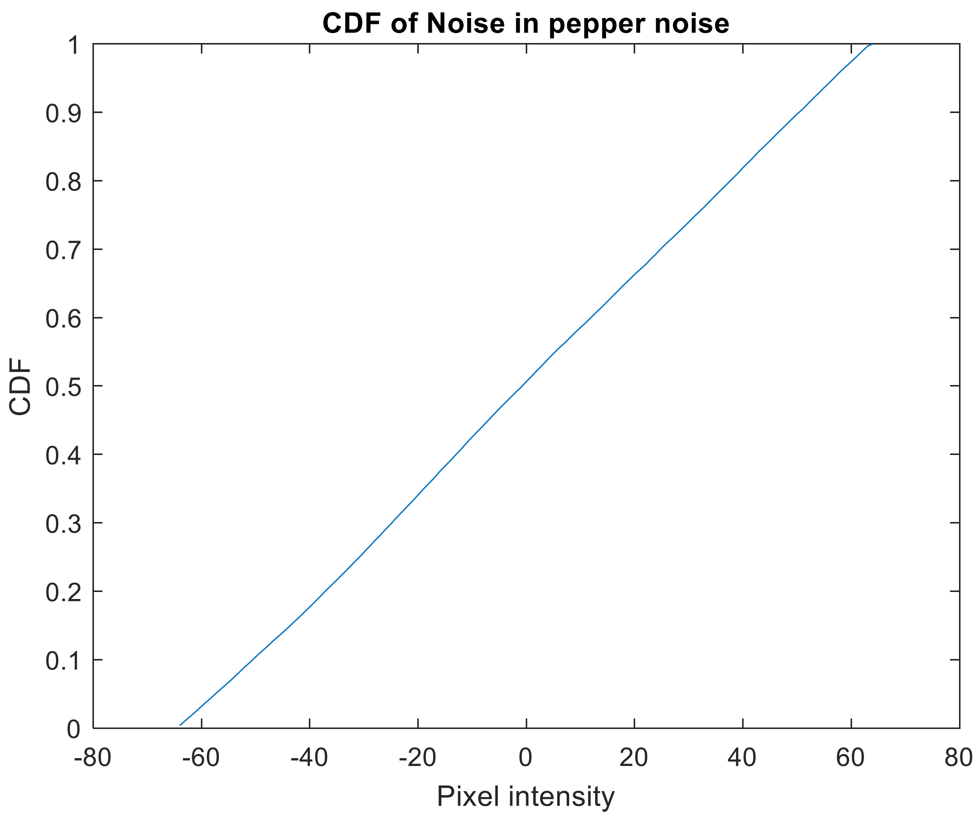


Fig 4.7: Cumulative Distribution function of Uniform noise between intensity -64 to 64.

2(a)(2)

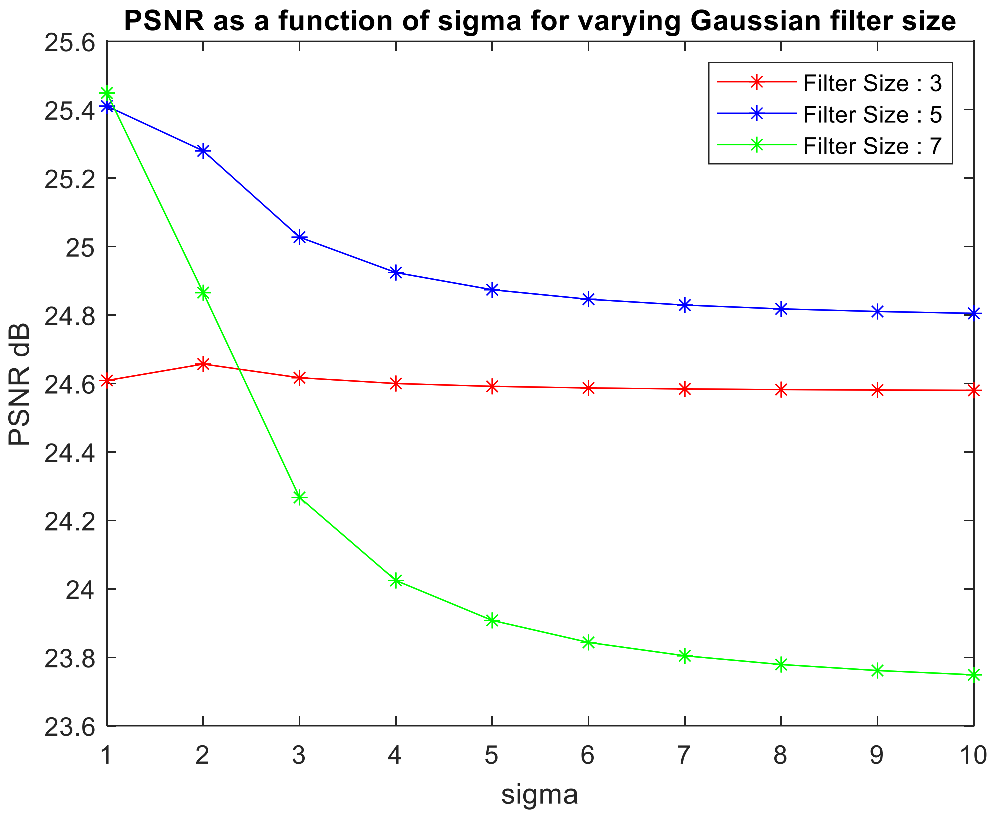


Fig 4.8 PSNR as a function of sigma for varying Gaussian filter size

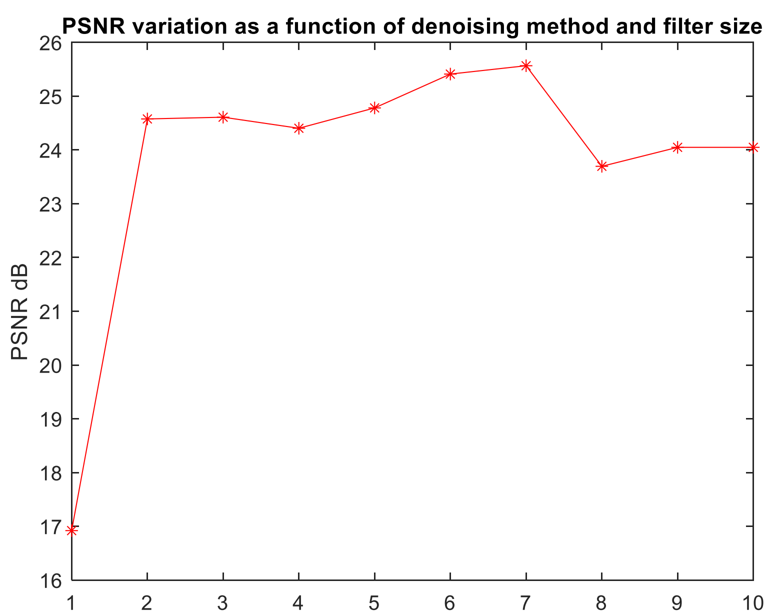


Fig 4.9: PSNR Variation as a function of denoising method and filter size.

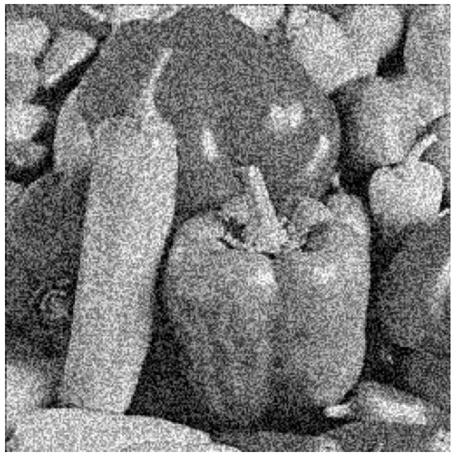


Fig 5. Original Pepper noise

Fig 5.1: Uniform filter Size -3

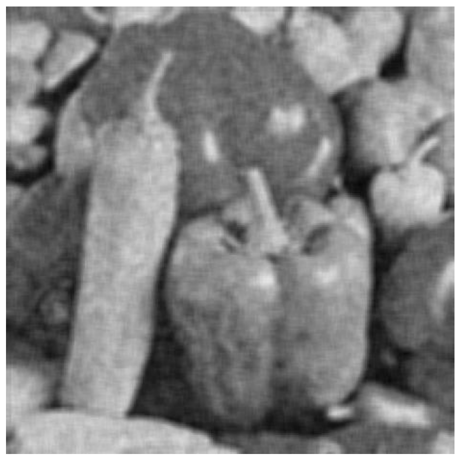


Fig 5.2: Uniform filter Size -5

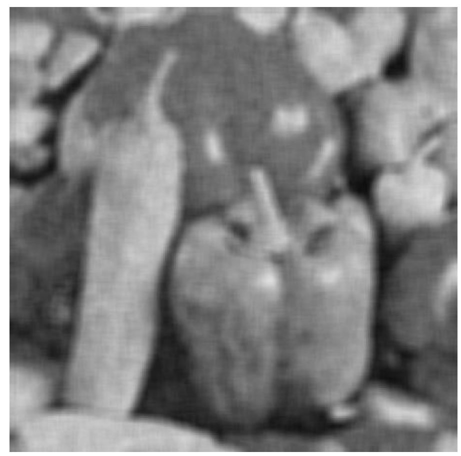


Fig 5.3: Uniform filter Size -7

Fig 5.4: Gaussian filter Size =3, σ=1

Fig 5.5: Gaussian filter Size =5, σ=1

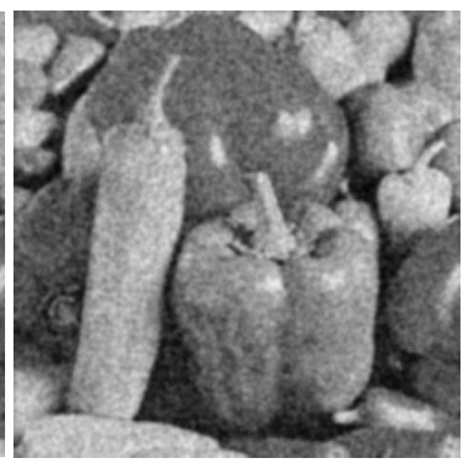


Fig 5.6:Gaussian filter Size =7, σ=1

2(a)(3)

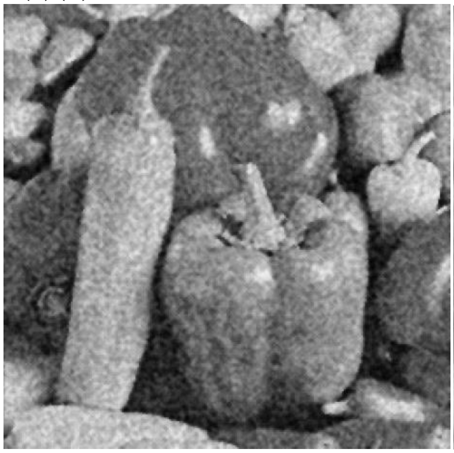


Fig 5.7: Bilateral filter Size =3, σs=120, σc=40

Fig 5.8: Bilateral filter Size =5, σs=120, σc=40

Fig 5.9:Bilateral filter Size =7, σs=120, σc=40

2(a)(4)

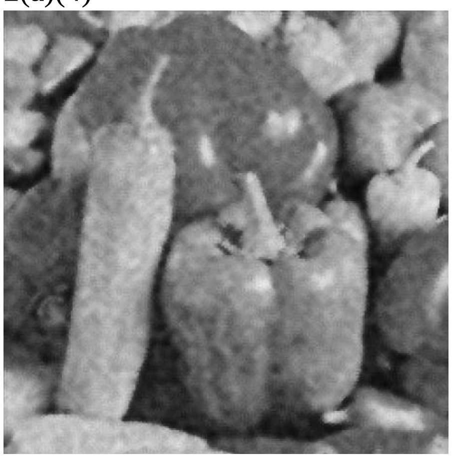


Fig 6: Non Local Mean Filter

2(b)(1) Yes, filtering must be performed on individual channels separately.
(2) Weighted median filter can be used to remove mixed noises

Fig 6.1 Roise noise

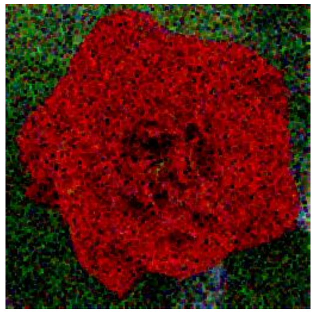


Fig 6.2 Median Filter

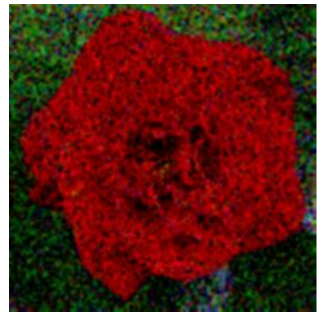


Fig 6.3 Uniform applied to median filter output

2(c)

Fig 6.3: Pepper short noise

Fig 6.4:Pepper original image

Fig 6.5: Biased Gaussian Inverse Transform

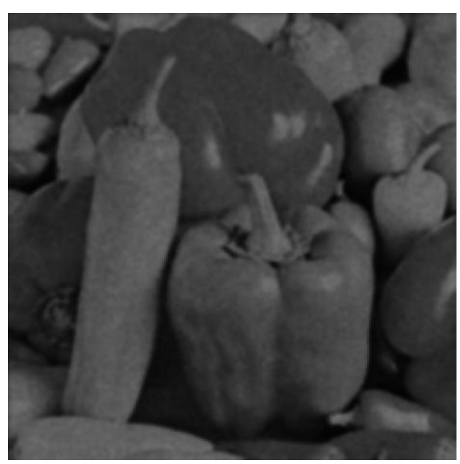


Fig 6.6: Unbiased Gaussian Inverse Transform

Fig 6.7: Biased BM3D Inverse Transform

Fig 6.8: Unbiased BM3D Inverse Transform

Discussuion:

2(a)

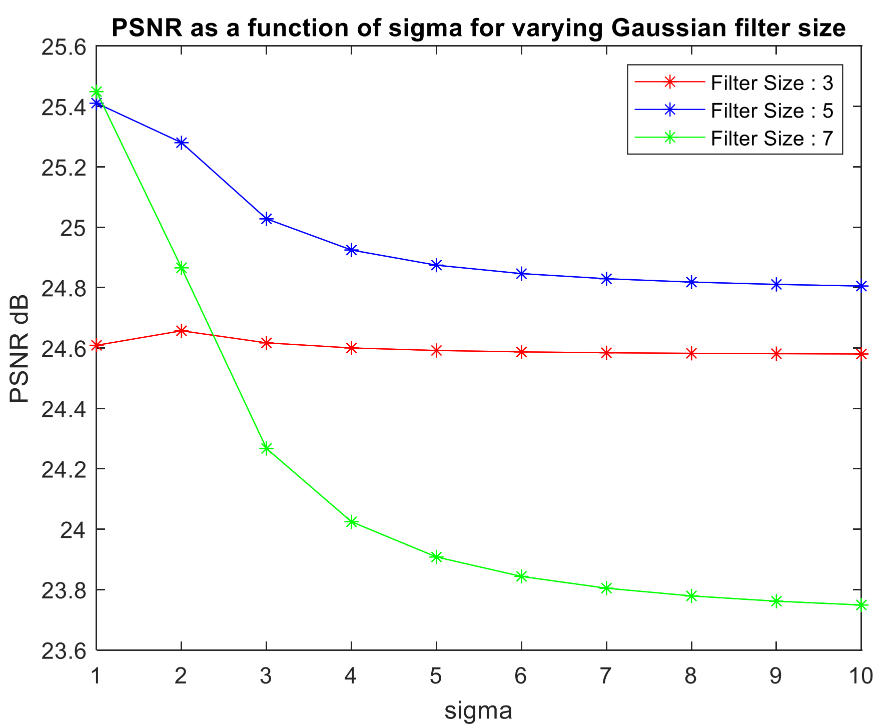


Fig 6.9 PSNR as a function of sigma for varying Gaussian filter size

The parameter of Gaussian filter, sigma and filter size are varied. The above figure depicts that the PSNR is highest when sigma is equal to 1 and as the filter size increases the performance of gaussian filter increases steadily up to a certain point and then increases with less increment.

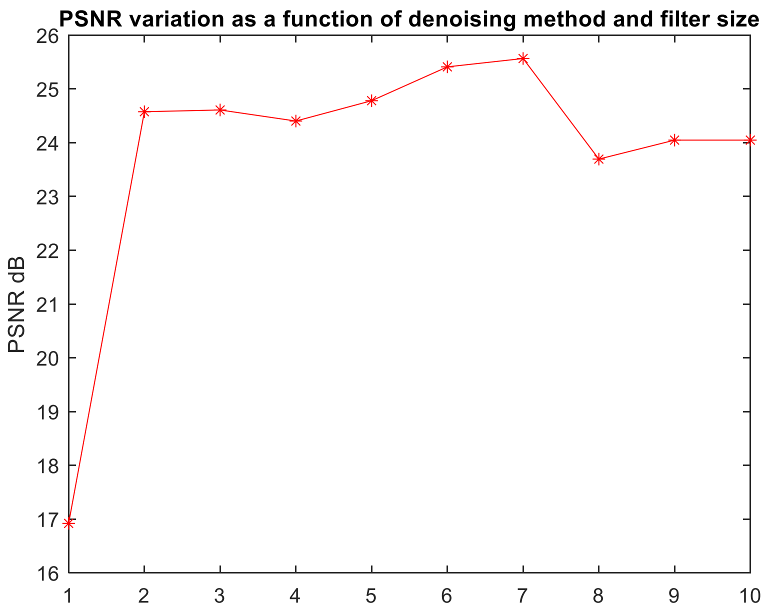
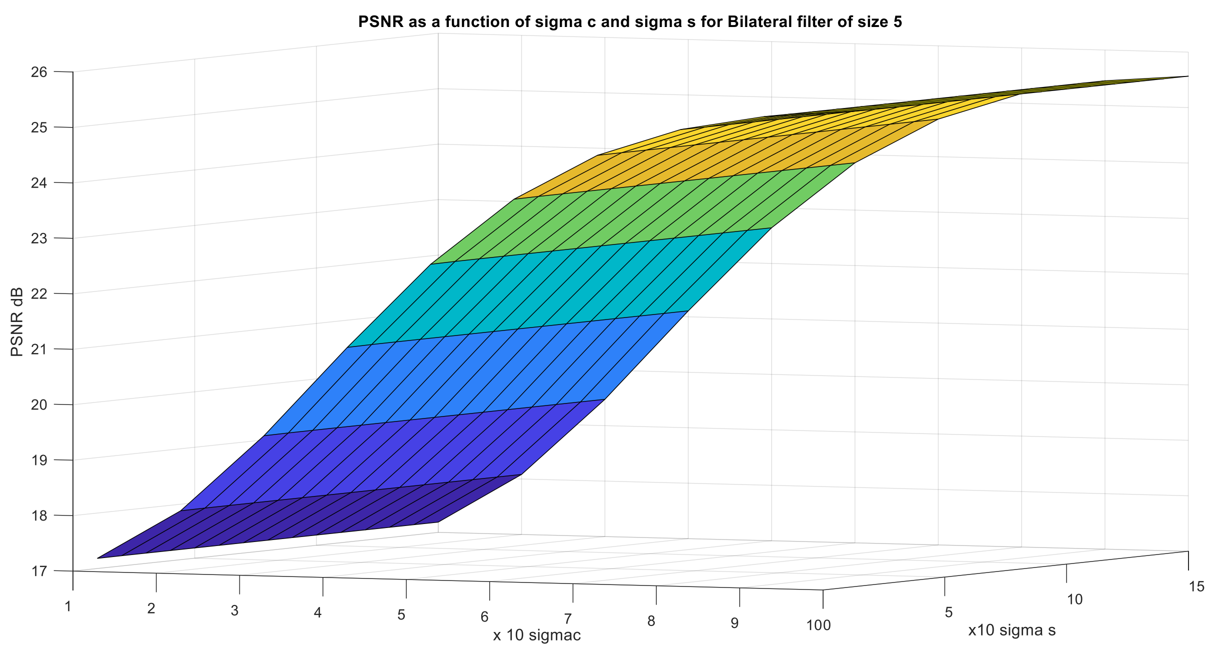
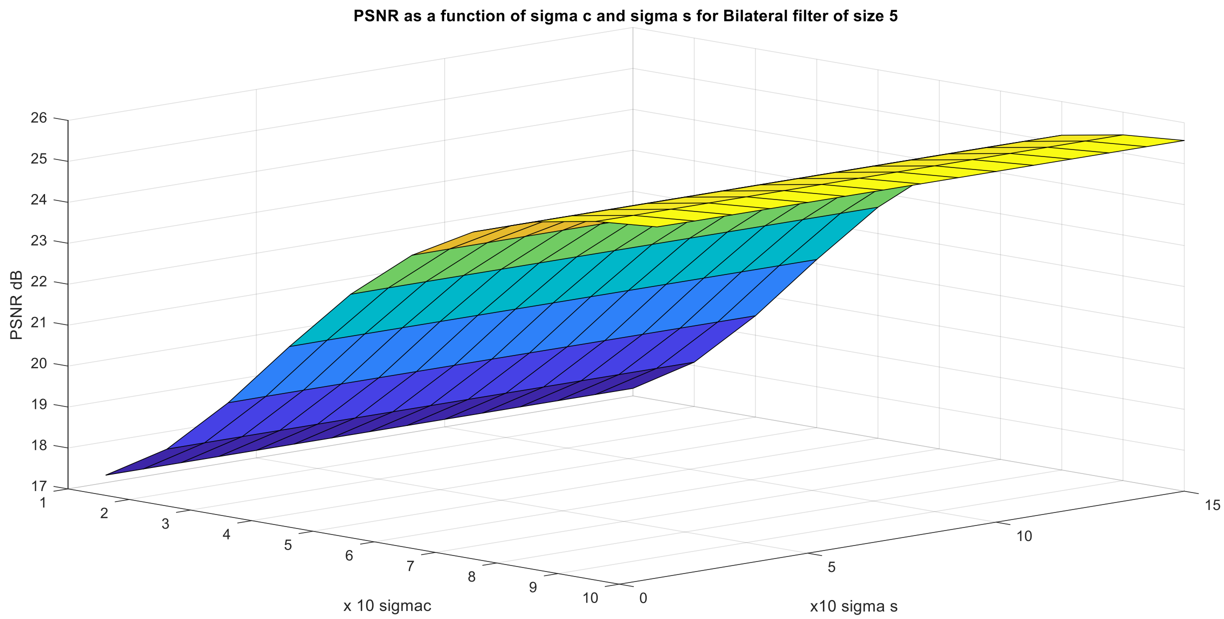


Fig 7: PSNR Variation as a function of denoising method and filter size

The position along x axis refers to x=0 – Pepper_noise, x=1: Uniform filter of size 3, x=2: gaussian filter of size 3, x=3: bilateral filter of size 3, x=4: Uniform filter of size 5, x=5: gaussian filter of size 5, x=6: bilateral filter of size 5, x=7: Uniform filter of size 7, x=8: gaussian filter of size 7, x=3: bilateral filter of size 7.
Incase of uniform, gaussian and bilateral filters, the performance or PSNR is highest in case of filter size =5. The performance of the filters steadily increases with filter size and then the performance drops down after a certain point. Here, filter PSNR is less when filter size =7.





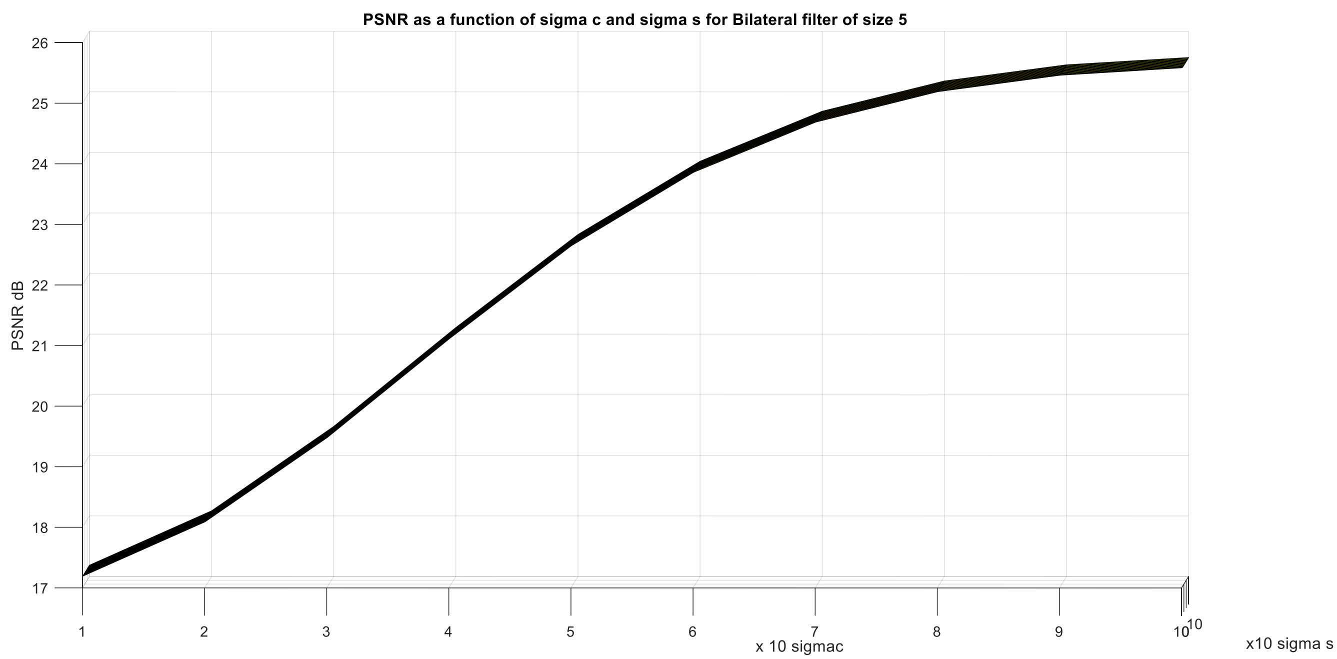


Fig 8. PSNR variation as a function of σs, σc for Bilateral filter of size 5. From the above plots, it is evident that as sigma c (σc) increases, the performance of the bilateral filter as well increases. The performance of the bilateral filter is less dependent on sigma s (σs).

2(c)

- The choice of the inverse transformation is vital to minimize the error when non-linear forward transformations are applied.
- The unbiased inverse transform is better than biased inverse transform because it is computationally efficient and can be interpreted as maximum likelihood.
- The PSNR is relatively better in case of unbiased gaussian inverse transform method in comparison to biased gaussian inverse transform method. (Table 1)

- BM3D denoising method yields better quality and PSNR compared to gaussian denoising method, reference (Table1)

Table 1: PSNR in dB for inverse transform methods applied to various denoising method along with parameters

	Unbiased Gaussian	Biased Gaussian	Unbiased BM3D	Biased BM3D
Gaussian filter size = 5 Sigma=1	34.6912006046961	34.6586349200638	39.1575852808582	39.2544887706779
Gaussian filter = 3 Sigma=1	35.1328236912677	35.1064207145589	39.1575852808582	39.2544887706779