CHARU SINGH

USCID:2773417243

charusin@usc.edu

Submission date: 1/28/2020

HOMEWORK 1

Question 1a

I. MOTIVATION

When a picture is taken from the camera you get your colored image as an output image. Since the camera has only sensor so how you will get colored image as an output.

The key factor behind this is we must extract all three colors (RED, GREEN and BLUE) data in order to form colored image.

II. ABSTRACTION

Basically, our output image is the combination of all three colors. Therefore, our goal is to stack these 3 colors on top of each other with different intensities. Sensors of the camera are arranged in the form of Bayer Array, also known as **Color Filter Array (CFA).**

Bayer pattern sensor layout is shown in figure1. As you can see that it is disproportionally biased towards green color, this is because human visual system is more sensitive to green color. Half of the total pixels are green; one quarter is red, and one quarter is blue.

Camera sensor only captures raw color data for the one third of the image. We use **Demosaicing** in order to make the best estimate to fill in remaining two thirds of color data.

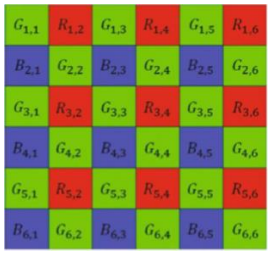


Figure 1 Bayer Pattern Sensor Layout

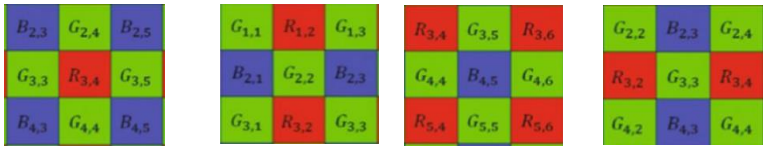
III. APPROACH:

In order to get all three colors at one-pixel location we must interpolate the information from surrounding pixels and this process is known as Color Filter Array Demosaicing. Basically, we will perform a **convolution** on this image. We will have 3*3 Kernel which we will iterate over the whole image. For example, in order to calculate green pixel at red the we will have a convolution mask of . $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$/4. Do element wise multiplication of this convolution matrix and original matrix and the divide by 4. Add all the elements in the window. Then in the resulting image red pixel is replaced by averaged value of green and blue pixels in the surrounding neighbors and red pixel will remain as it is.

Demosaicing can be done using two methods:

1.***Bilinear Demosaicing***:

The key idea is to perform linear interpolation in x direction and then in y direction. In order to calculate the intensity of other two pixels at a given location we reconstruct them using intensity values of neighboring pixels.



IV. PROCEDURE:

a. First, we will find that at a given location what color pixel is present.
b. There are four different patterns in the image as shown above
   To calculate Blue and Red values at Green locations:
   Case1: When green is at odd position ($G_{odd,odd}$):
   $B_{3,3}=0.5(B_{2,3}+B_{4,3})$
   $R_{3,3}=0.5(R_{3,2}+R_{3,4})$
   Case2: When green is at even position ($G_{even,even}$):
   $R_{2,2}=0.5(R_{1,2}+R_{3,2})$
   $B_{2,2}=0.5(B_{2,1}+B_{2,3})$
c. To calculate Red and Green values at Blue locations:
   $R_{4,5}=0.25(R_{3,4}+ R_{3,6} + R_{5,4} + R_{5,6})$
   $G_{4,5}=0.25(G_{3,5}+ G_{4,4} + G_{4,6} + G_{5,5})$
d. To calculate Blue and Green values at Red locations:
   $B_{4,5}=0.25(B_{2,3}+ B_{2,5} + B_{4,3} + B_{4,5})$
   $G_{4,5}=0.25(G_{3,3}+ G_{4,4} + G_{2,4} + G_{3,5})$
e. Now each pixel is replaced by the overlapping of three pixels which we calculated above. For example, if found that at a given location the pixel is green so we will find red and blue pixel values from above given formula. Likewise, we iterate over the input image and do the same.

f. As a result, we will get a colored image.

V. OBSERVATION AND RESULTS:

1. Visually we can observe that in bilinear there are some blurry areas like the ball outline but in MHC its sharper and there are more fine details. MHC method is slightly better because it is using information from farther locations thus interpolation will be better.
2. Visual quality of these method is not good because we are interpolating every channel separately, it causes misalignment of color in high frequency specially near the edges and as result it causes distortion and zipper artifacts near the contours of the object, hence degrading the quality of an image.
3. Computational time is more because of their iterative nature.
4. This demosaicing method would have provide optimal solution if R,G and B were statistically independent however in natural images such independence does not hold.

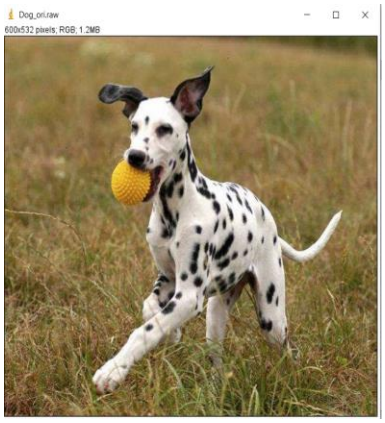


Figure 2 Original Image

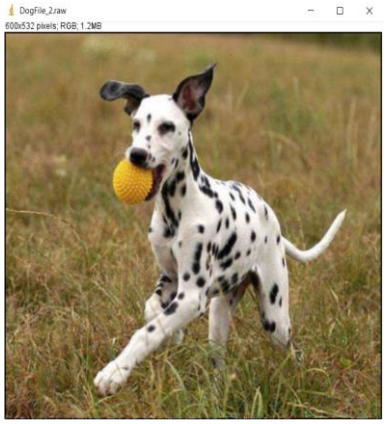


Figure 3 Output Image (Bilinear)

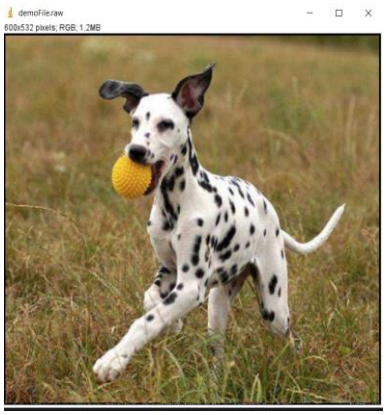


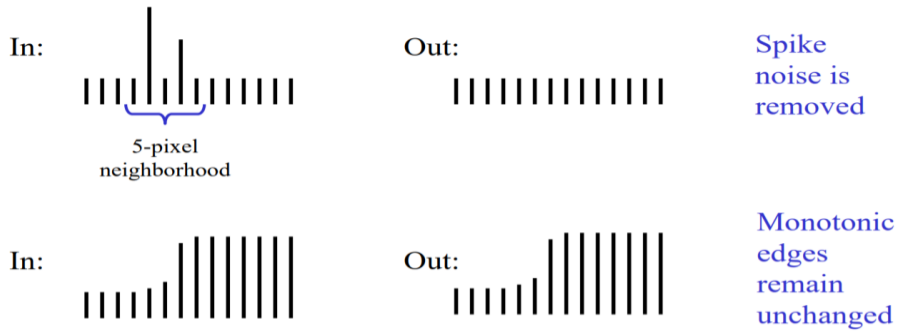
Figure 4 Output Image(MHC)

- We can improve the performance by using median filter because in this method interpolation takes place on isolated color channels then it computes the color difference signals then apply median filter on the color difference signal. Lastly reconstruct 3-D image.
- Take correlation among the 3 color planes. It will combine analysis of image detail and judgement and better interpolation will be achieved.
- Edge directed interpolation
- Gradient based algorithm
- Directional based method

Median filter:

Image Demosaicing for Bayer-patterned CFA Images Using Improved Linear Interpolation

For example replace each pixel value by the media performed over N pixels. In this case it is 5.



Question 1b

2. Malvar-He Cutler (MHC):

This method is the modification of bilinear interpolation. In this method, Laplacian or 2nd order channel corrections are added.

- For example, the green component at red pixel is calculated as: here Δ_R is the discrete 5- point Laplacian of Red channel.

$$\hat{G}(i,j) = \hat{G}^{bl}(i,j) + \alpha\Delta_R(i,j).$$

$$\Delta_R(i,j) := R(i,j) - \frac{1}{4}\big(R(i-2,j) + R(i+2,j) + R(i,j-2) + R(i,j+2)\big).$$

- To estimate red component at green pixel:

$$\hat{R}(i,j) = \hat{R}^{bl}(i,j) + \beta\Delta_G(i,j),$$

- To estimate red component at blue pixel:

$$\hat{R}(i,j) = \hat{R}^{bl}(i,j) + \gamma\Delta_B(i,j).$$

NOTE: α=0.5, β=5/8, γ=3/4 are calculated optimally. They determine the weight of Laplacian correction terms. To set these parameters optimally, the values producing the minimum mean squared error over the Kodak image suite were computed. These values were then rounded to dyadic rationals.

The advantage of this rounding is that the filters may be efficiently implemented with integer arithmetic and bit shifting. The filters approximate the optimal Wiener filters within 5% in terms of mean squared error for a 5 × 5 support.

Convolution is performed with set of linear filters. There will be eight different filters for different colors components for interpolating at different locations. For example, at a green pixel location in a red row, the red component is interpolated by [1].

$$\begin{aligned}\hat{G}_{3,4}^{MHC} &= \hat{G}_{3,4}^{bl} + \alpha\Delta_R(3,4)\\ &= \frac{1}{4}(G_{3,3} + G_{2,4} + G_{3,5} + G_{4,4}) + \frac{1}{2}\left(R_{3,4} - \frac{1}{4}(R_{3,2} + R_{1,4} + R_{3,6} + R_{5,4})\right)\\ &= \frac{1}{8}\left(2 * (G_{3,3} + G_{2,4} + G_{3,5} + G_{4,4}) + 4 * R_{3,4} - (R_{3,2} + R_{1,4} + R_{3,6} + R_{5,4})\right)\end{aligned}$$

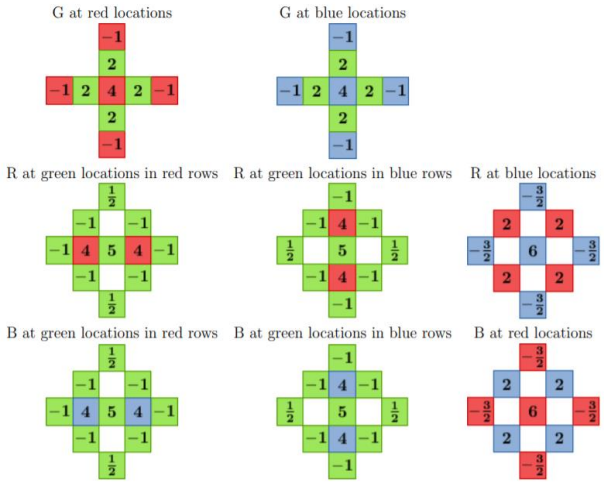


*Figure 5 :5*5 Linear filter where coefficients are scaled by 1/8*

Comparison between Bilinear interpolation and MHC:

1. MHC is visually sharper than Bilinear Demosaicing.
2. Reduced computational complexity.
3. No artifacts due to nonlinear processing.
4. More the neighboring pixels used in interpolation, better will be the reconstructed image. Therefore, it will be having higher PSNR than Bilinear interpolation.

Question 1c

HISTOGRAM:

I. MOTIVATION AND ABSTRACT:

Histograms plays a crucial role in image processing as it provides representation of pixel distribution in the graph as function of the tonal variation.

Histograms of images can be analyzed for peaks or valleys and thus the threshold value can be exploited to determine edge detection, image segmentation etc.

Histogram Equalization is an image processing technique which improves the contrast of images. This is done by effectively spreading the most frequent intensity values.

Question1c(a):

APPROACH:

We have to plot the histogram for all three-color channels in the colored image.

PROCEDURE:

1. We have created three different arrays for R,G and B with intensities ranging from 0-255 and calculated frequency for each intensity value.
2. I have coded the algorithm in c++ and export csv file to matlab in order view the histogram.

RESULT:

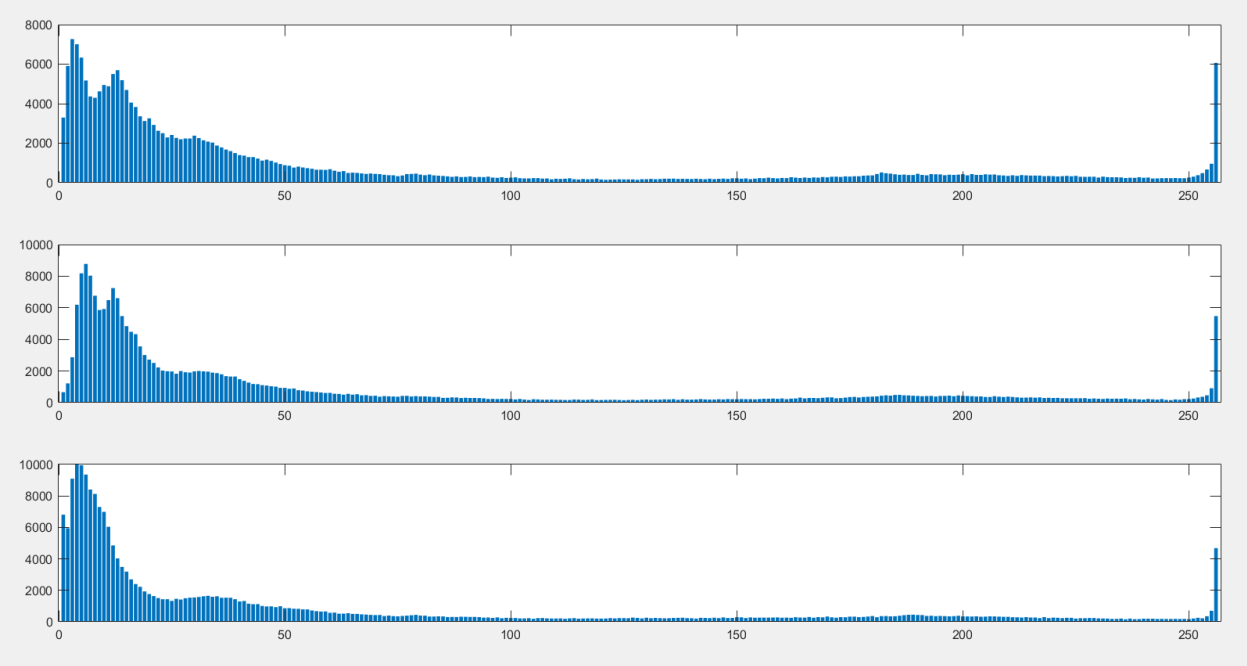


Figure6: Histogram of R G and B

OBSERVATION:

The figure shows histogram for channel red green and blue respectively.

II. APPROACH:

Here we have used two different approaches in order to achieve histogram equalization.

METHOD A:

Based on transfer function.

III. PROCEDURE:

1. First check which color pixel is located at the location. Iterate over the whole image and create 3 different 1-D arrays for red, green and blue with intensity values ranging from 0-255.
2. Get the count for each intensity value for all three colors.
3. Calculate the probability for each of the frequency array that is each value must be divided by total number of pixels in an image.
4. Calculate cumulative distributive function.
5. Multiply each value by 255 since we want to change the intensity range and then floor round the values.
6. Replace the new values in each array with original pixels in the image for three different channels that is red, green and blue.

METHOD B:

Cumulative probability-based histogram equalization method, also known as Bucket method.

PROCEDURE:

1. First, we will create three different 1D array for three different colors.
2. Sorting is done in increasing order of intensities values.
3. Now we iterate over the image and calculate frequency for each intensity value for all R, G and B channels.
4. Then divide the total number of pixels that is 224000 by 256. We get 875. This is the bucket size.
5. Now in order to uniformly distribute the intensity values over the image. Check for each each intensity value for example if 0 intensity have 800 pixels of red color therefore all these pixles will go in bucket 0 and remaining 75 pixels will be borrowed by 1 .
6. Now store the location of those pixels which you have transferred here its 1's.
7. Now all those transferred pixels who has intensity value one will become zero.
8. We will keep filling the buckets until all the buckets have equal number of pixels.

RESULTS:

Figure 7 – Input Image

Figure 8 – Output after Histogram Equalization

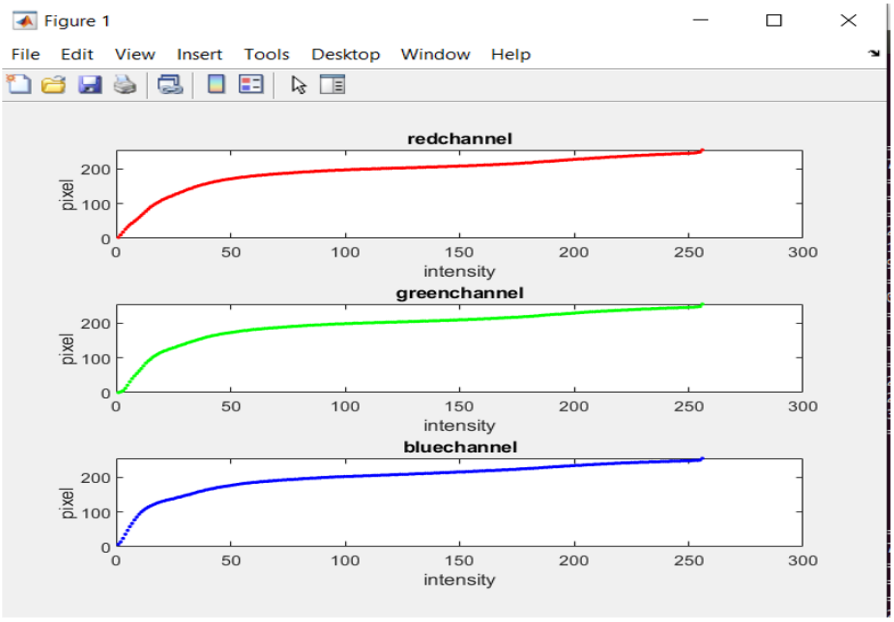


Figure 9 – Transfer function plot for R, G and B

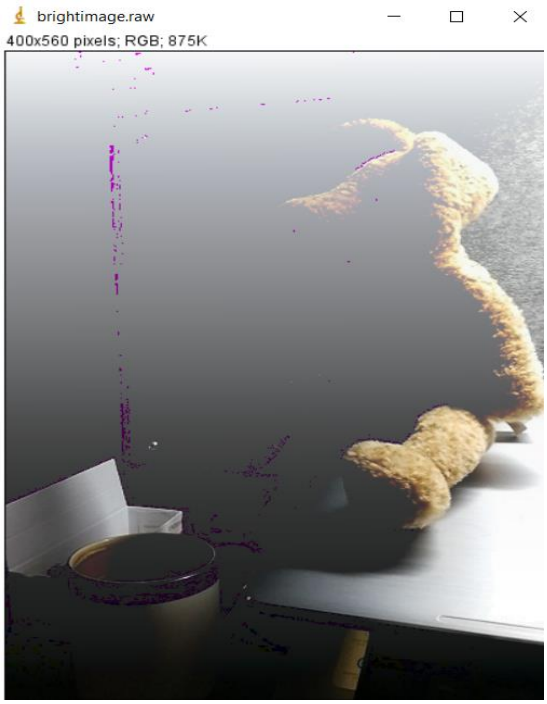


Figure 10: Histogram Equalisation: Bucket method

OBSERVATION:

The intensity levels are distributing over the entire image. The spots which were darker in the original image are now light in figure 9. Therefore, this results in illumination of darker parts and dimming of brighter ones**.** Ideally in bucket method we should get more uniformly distributed intensity levels and more uniform image but I am getting more brighter output it seem that the intensity at darker areas are increased too much.

Question 2: IMAGE DENOISING

MOTIVATION:

As we all know numerous of digital images are taken every day however images captured by modern cameras are embedded with noise by the influence of transmission channel , influence of the environment and many other factors which decreases the visual image quality and loss of information , As a result tasks such as image video processing, tracking, image analysis are affected. Therefore, in order to capture accurate and visually pleasing images we have perform various denoising algorithms on an image**.** Mathematical representation:

$$X=S+N$$

Where X is the noisy signal, S is the original signal and N is the noise in the image.

ABSTRACT:

In order to reduce the noise from an image without losing any of its important features such as corners, edges, sharpness. Because noise and all these components are of high frequency it is challenging to separate them during denoising process. Nowadays extracting meaningful information from the noisy and distorted image is the matter of concern. Various denoising algorithms have been proposed which have their own pros and cons.

Goal of image denoising:

1. Areas which are flat should be smoothened.
2. Blurring should not occur at edges, should be preserved.
3. Textures should not deteriorate.
4. New artifacts should not be introduced.

Denoising Methods

QUESTION2.a.1

There is a uniform gaussian noise present in the image

QUESTION2.a.2

1. **LINEAR FILTERS:**

a. Averaging Filter:

This method involves the convolution on noisy image.

PROCEDURE:

- Firstly, we will perform convolution on an image where all elements have equal weights. Here is the convolution matrix we have used:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Then we will iterate over the image with the given convolution mask by doing element wise multiplication of the pixels between noisy and the Kernel matrix for a particular central pixel(i , j) where the kernel is masked. Then we add all the elements and get a new value for the pixel at location (i, j).
- We normalized the kernel by 9 before performing convolution.
- Finally, we get a new image with new pixels values
- Peak Signal to Noise ratio is calculated as follows:

$$PSNR(dB)=10 \log_{10}(max^2/MSE)$$

$$\text{MSE} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(Y(i,j)-X(i,j))^2$$

Where N*M is the size of an image

Max: Maximum possible intensity=255

X: original noise free image

Y: filtered image

GAUSSIAN FILTER:

Main application of gaussian filter is to blur an image uniformly being an image content or edges. It computes the similarity between the central pixel that is where the filter is applied and neighboring pixels that is used for blurring. It reduces contrast and blurs edges.

Kernel used in gaussian blur values of the pixels correspond to the gaussian 2D curve. It's a linear operation but it does not preserve the edges in original image.

It removes the high frequency components from input image

There will be bright values at the center and as you go away from the center it gets darker.

Most intensity will be there at the peak of the gaussian curve. It's a linear combination attributing weight giving more weight to the pixels at the center and less value in linear combination to pixel that is far away from the center. Farther pixels will get lower weight.

How to choose sigma:

Choose a mask (gaussian Kernel) with size 3 times standard deviation.

In our case we chose a kernel size of 5 that is sigma value will be around 0.83.

PROCEDURE:

- Firstly, we will perform convolution on an image where all elements have gaussian weights . Convolution matrix we have used is computed by formula below(w(i,j,k,l)):

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

$$w(i,j,k,l) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(k-i)^2+(l-j)^2}{2\sigma^2}\right)$$

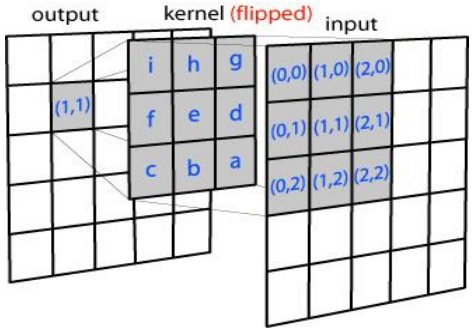
Here, (k,l) is the neighboring pixel location within window size centered around (i.j)

I:Noisy Image

Y:Output Image

σ: Degree of smoothening

- Every pixel is multiplied by Gaussian filter by placing central pixel of filter on the image pixel and multiplying values in the original noisy image and then we add all the values and the result is our new pixel value.
- Then we will follow the same procedure as in averaging except the fact that the gaussian mask will change as we iterate over an image.
- PSNR is calculated using the above formula.



- In the figure above , value at (0,0) in input is multiplied by value at I in gaussian kernel and so on and then add all the values to get new pixel value for (1,1)

RESULTS:

FOR AVERAGE FILTER:

Getting PSNR as 19.0499.

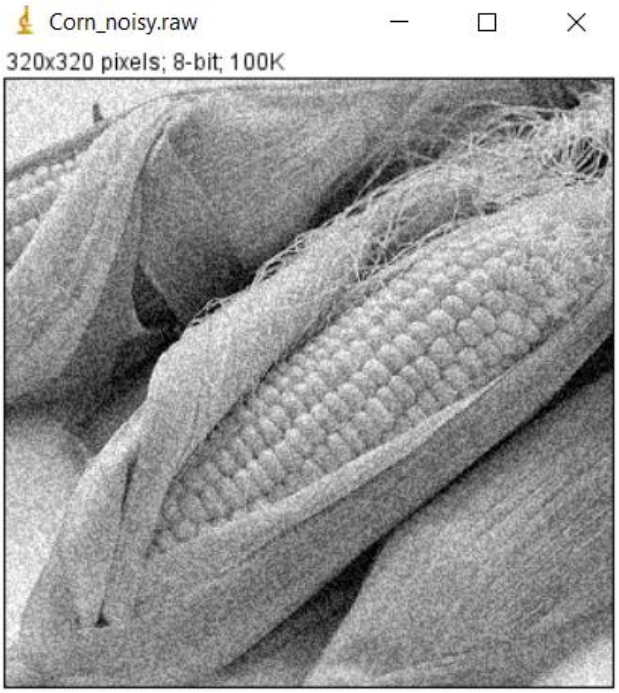


Figure11: Input Noisy Image

Figure12:Output Image_Averaging

GAUSSIAN FILTER:

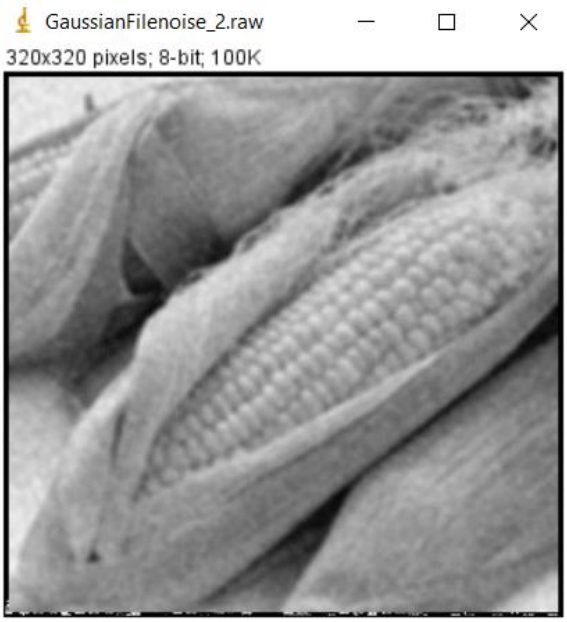


Figure13: Output Image_GAUSSIAN_sigma=0.83_PSNR=18.43 *Output Image_GAUSSIAN_sigma=2 _PSNR=17.54*

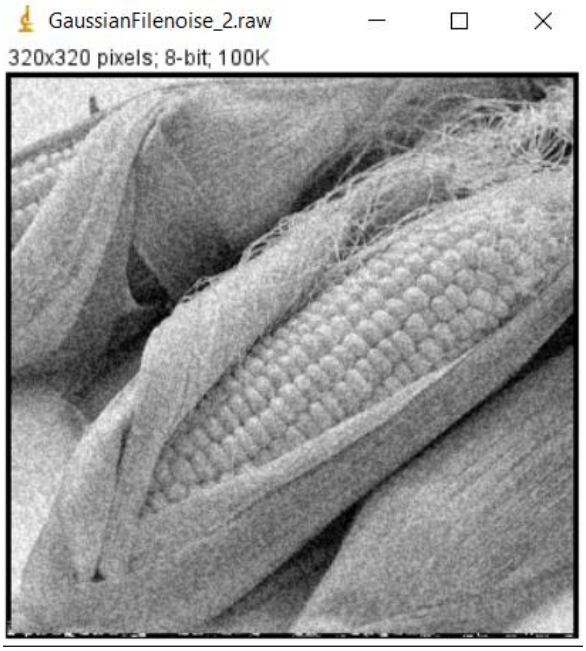


Figure15:Output Image_GAUSSIAN_sigma=0.5_PSNR=20.17

DISCUSSION:

- As you can see that in both Averaging and Gaussian filter the output image has blurred edges and the contrast is reduced. This is because the filters interpolate new pixel values on the edges and thereby blur the image and reduces the sharpness of edges.
- However, Gaussian filtering is better than Averaging filter because in gaussian denoising pixels which are close have a greater impact on the smoothed rather than pixels which are located far away this means farther pixels will assign lower weights. In Average filter complexity depends only on size of an image not on the radius of the filter as equal weight is given to every pixel around the center regardless its distance from center pixel but in gaussian filter sigma determines the amount of smoothing.
- Also, we can observe from PSNR values calculated above that Gaussian denoising gives better result than Averaging denoising.
- Also, the blur effect depends on sigma in gaussian denoising, more the sigma value more will be the blurring effect.

BILATERAL FILTER:

It's a nonlinear filter and it preserves the edges unlike gaussian filter. While averaging within the smooth regions of the input image it prevents averaging across the edges and thereby preserves edges. It's an non iterative method, hence reduces computational complexity.it considers similarities of gray and color levels as well as geometric closeness of neighboring pixels.

If pixels are very close to each other it will multiply the gaussian coefficient by a number close to 1 and works same like in gaussian filtering technique

But if pixels are very different from each other gaussian coefficient is multiplied by a number close to 0 thus disabling the gaussian filter for this pixel.

Therefore, in uniform areas gaussian filtering is applied and there will be no filtering across borders of an object.

Space and range parameters:

Space (sigma s): spatial extent of kernel

Range (sigma r): minimum amplitude of edge

Pixels which are enclosed in space and range are considered.

σ_s , σ_r determine the behavior of the filter. Optimal σ_s value is relatively insensitive to variance of noise as compared to σ_r.

Large σ_s will blur more as it combines pixels from distant locations. Good range of σ_s [1.5-2.1].

σ_r changes significantly as standard deviation of noise (σ_n)changes its because if σ_r is smaller than σ_n , noisy data can remain isolated and when it is large then σ_s play role.

If the image is attenuated/amplified σ_r should be adjusted accordingly.[1]

PROCEDURE:

Follow the same steps as in Gaussian denoising except that the convolution kernel is calculated as shown below:

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

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

Space (sigma c): spatial extent of kernel

Range (sigma s): minimum amplitude of edge

RESULTS:

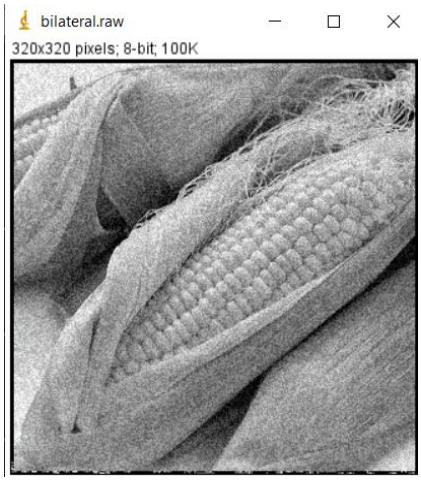


Figure16: Denoised image_Bilateral_PSNR:20.98

OBSERVATION:

Bilateral denoising performs better than Averaging denoising and Gaussian denoising algorithms. Unlike bilateral, they are linear filters which produces smoothed and blur image with poor localization of features and poor noise compression.

We have PSNR value as 20.98 greatest amongst all.

Gaussian filters have an advantage that their shapes can be easily identified. If we have narrow frequency domain filter, then sigma will be more, and hence spatial extent will be wider that attenuates low frequency resulting in blurring.

Fails at edges where we don't have smooth spatial variation. Bilateral overcome this issue by filtering in both space and range

NON- LOCAL MEAN FILTER:

ABSTRACT:

In order to measure similarities between neighborhood pixels it uses gaussian weighted Euclidean distance. But when there is increase in noise then this method fails to give good results, thereby deviation is caused in the similarity coefficient because of the error in measurement of similarity of neighborhood.

PROCEDURE:

1. Noise image model is considered as:

$$V(i)=X(i)+N(i)$$

X(i): Original image

N(i): Gaussian white noise with mean 0 and variance σ^2

V(i): Noisy image

2. For a given noisy image:
   $V=\{V(i)|i \in I\}$
   I: image field for given pixel in domain
3. Now weighted averaging of all pixels in the noisy image will take place and for a particular point we get the estimated value.

$$NL[V] = \sum_{J \epsilon I}^{n} \mathrm{w(i,j)v(j)}$$

   Value of w(,j) depends on similarity of i and j pixels.
   $0<=w(I,j)<=1$ and $\sum_j w(i_2 j)=1$
4. The degree of similarity of i and j pixels are calculated using Gaussian weighted Euclidean distance d(I,j) of two matrices which are centered on pixel i and j :
   $d(I,j)=||v(N_i)-v(N_j)||^2_{2,\alpha}$

above equation represents L^2 norm of gaussian weighted Euclidean distance (between neighboring matrices that is N_i and N_j

α: Standard deviation

5. When we measure distance from neighborhood the center of matrix has larger weight.
   The weight $w(i,j)$ is defined as:

   c(i): normalization factor

$$w(i,j) = \frac{1}{c(i)} f_k(d(i,j))$$

$$f_k = \exp(-\frac{d(i,j)}{h^2})$$

$$c(i) = \sum_{j \in I} \exp(-\frac{d(i,j)}{h^2})$$

h:attenuation factor of exponential function.

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

RESULTS:

Figure17: Denoised Image

DISCUSSION:

In this method when the noise is large the gaussian weighted template having fix coefficient will subject to interference. To solve this Laplacian operator can be used. This method is also computationally extensive. The process of gaussian convolution only preserve flat zones and fine structure as well as contours are blurred whereas NLM provides a feasible method by reducing the noise in every geometric configuration.

PARAMETERS ANALYSIS:

The standard deviation alpha determines the spreaders of the noise distribution of an image so if we alpha becomes too large then smoothness increases therefore fine details are lost and we decrease the alpha

value too much then the effect of non-local neighborhood will be less and it reduces noise to a smaller extent. Here h is the filtering parameter and it is directly proportional to smoothening factor.

BLOCK MATCHING 3D (BM3D):

I. MOTIVATION AND ABSTRACT:

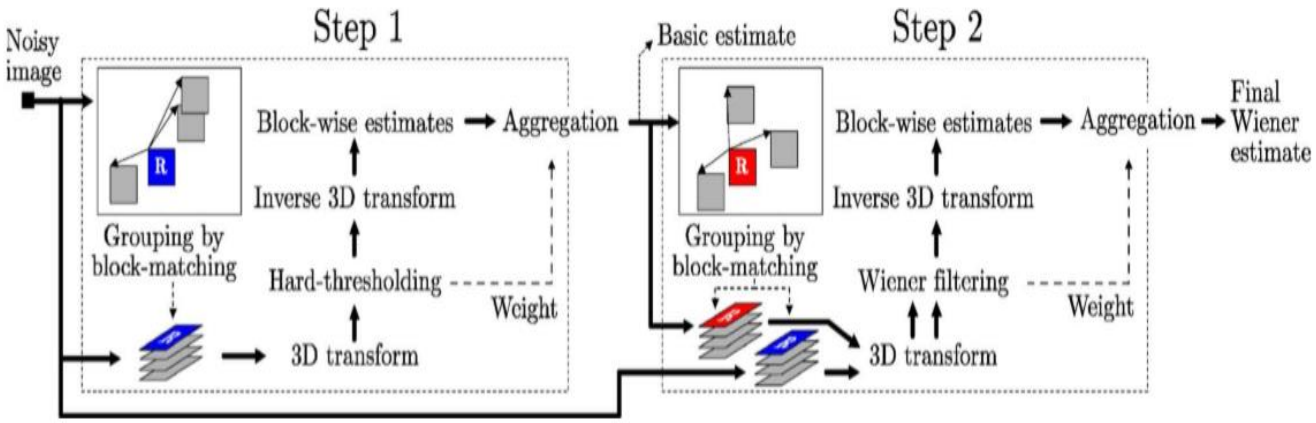
A novel denoising strategy is proposed based on spatial representation enhanced in transform domain. Grouping of similar 2D blocks to 3D data arrays called groups is done to enhance the sparsity. To deal with 3D groups we use technique called *Collaborative Filtering* which reveals the finest details those shared by the group blocks and preserves unique features of individual block Aggregation takes place in order to overcome the redundancy of the blocks overlapping by obtaining different estimates of a particular pixel. There are three main steps involved:

a. 3D transformation of group
b. Shrinkage of transform spectrum
c. Inverse 3D transformation.
d.

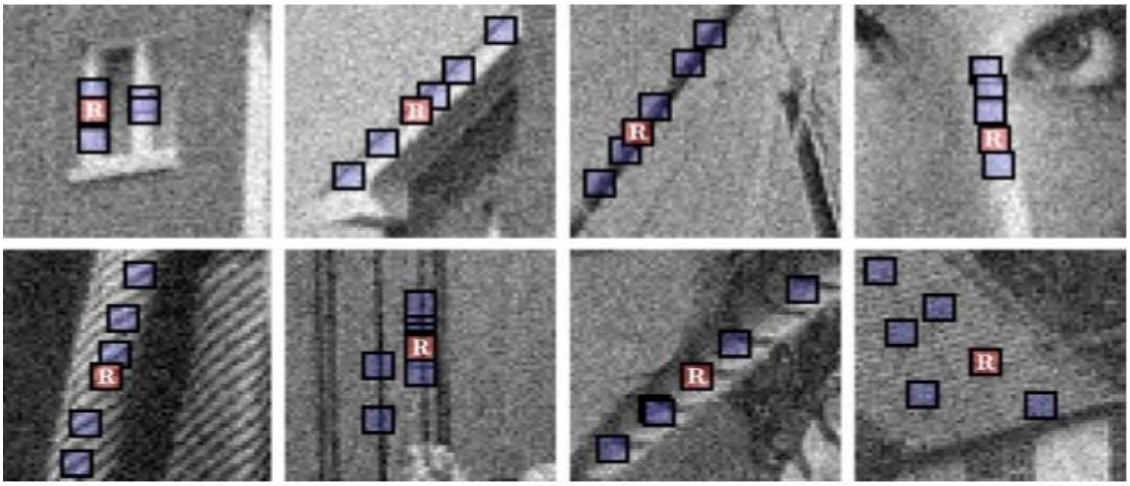
II. APPROACH:

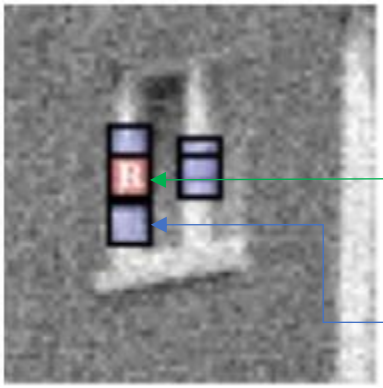
1. It builds on patches.
2. It capitalizes nonlocal means on the idea of exploiting self-similarity that most images possess.
3. It introduces collaborative filtering.
4. Aggregation is performed.

III. PROCEDURE:



Block Diagram:BM3D





Reference Patch

Sum of patches quite similar to reference patch that can be used to better estimate the structure of reference patch.

BASIC IDEA:

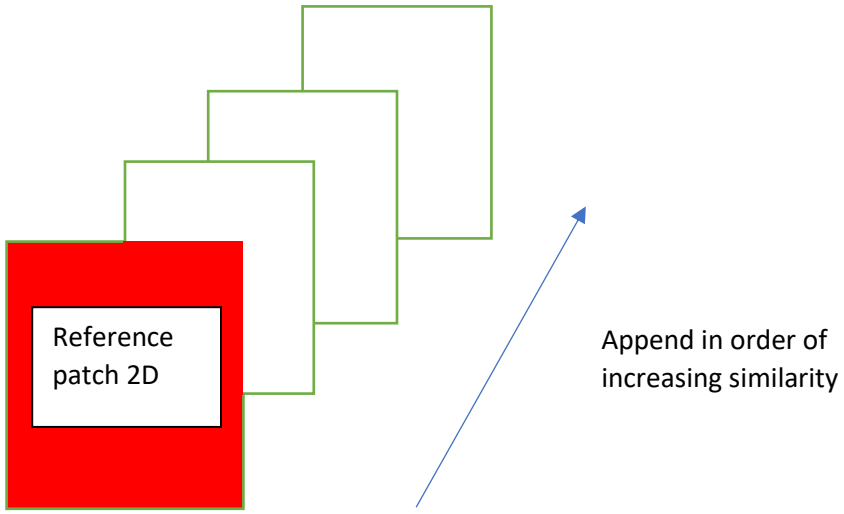
If the original image is 2D then it is stacked into 3D.The patches are usually sorted by the proximity of the reference patch along the new artificial dimension then the 3D denoising algorithm is used which potentially changes the look of every patch in this block and afterwards this information is projected back to image domain.

HOW TO PROJECT THE INFORMATION BACK TO IMAGE DOMAIN:

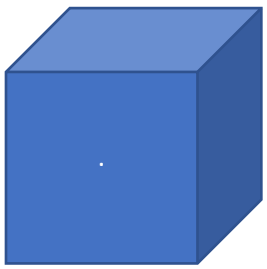
Its very difficult to achieve this task because one location in an image may be a part of different multiple groups so we use Aggregation.

PROCEDURE:

1. In order to denoise the patch, first denoise the image
2. Next form the groups of denoised patch with similar denoised patches as shown in figure



In figure shown above each of patches is 2D and there is extra dimension so that overall we get something in 3D as shown below

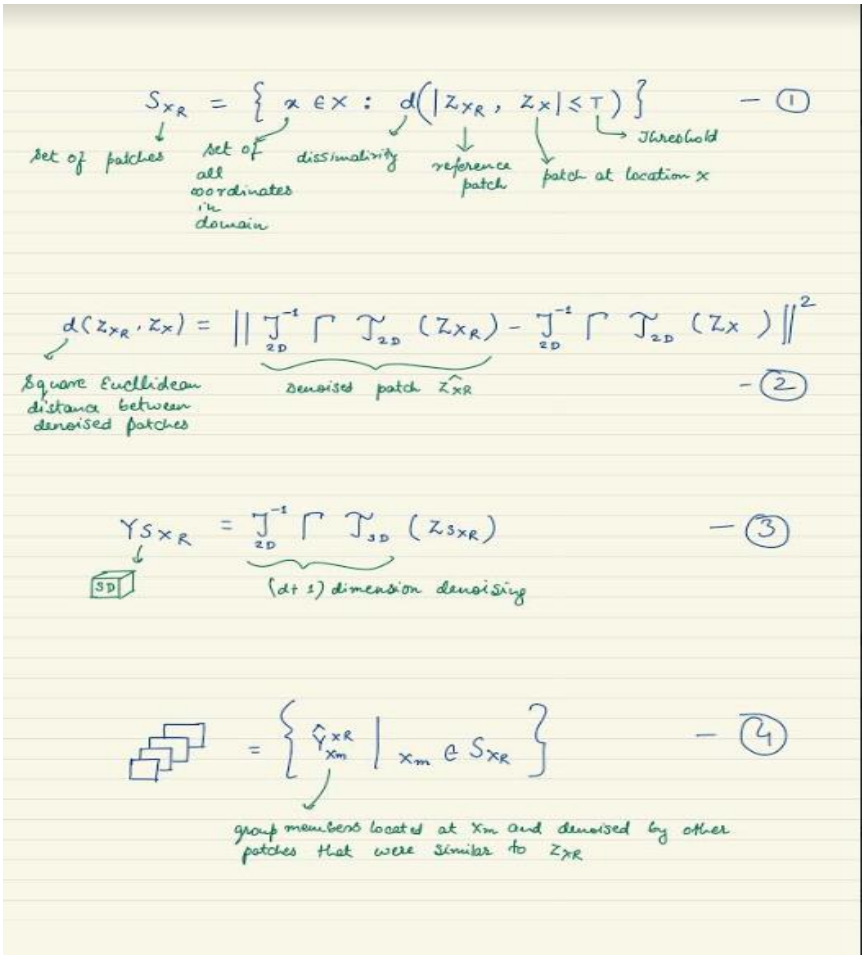


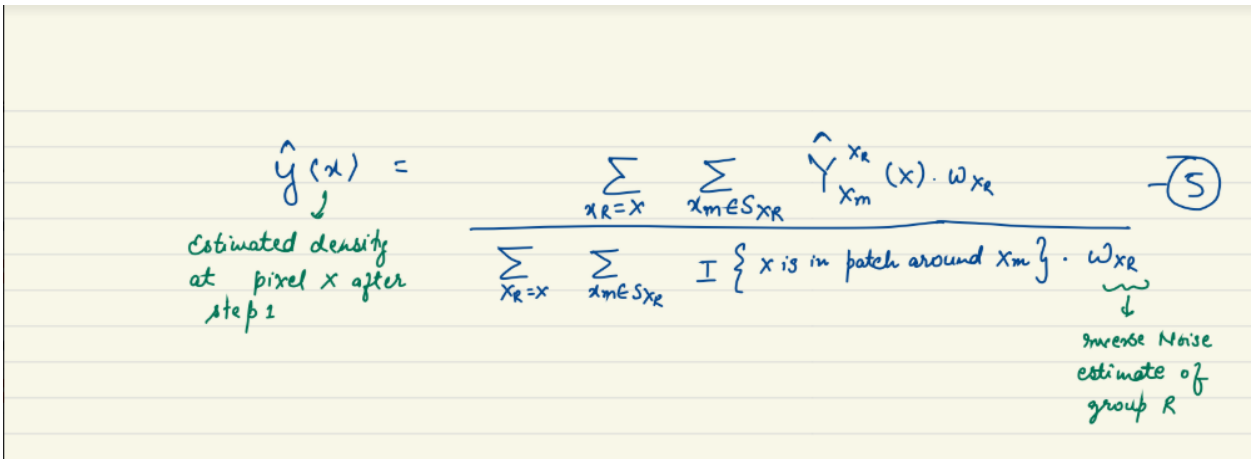
Stacked collections of patches/blocks

It is a d+1 dimensional group (d is the dimension of original image).

3. Now denoise the d+1-dimensional group.
   RATIONALE OF THE METHOD:
   Now the idea is to exploit the interpatch correlation resulting from smoothness of natural images and at the same time exploit the intrapatch correlation resulting from the self-similarity of the patch.
4. After doing this process to all the groups and the groups comprises of overlapping patches so that the same patch can be a member of multiple groups and our goal is to bring it back to the original image. This we do by aggregation. This involves a concept of averaging which determines different estimate of each pixels present in different blocks.
5. Repeat the above process for step 2 in figure using Weiner denoising strategy.

MATHEMATICAL UNDERSTANDING:





Equation 1 : To form a group we need to find a set of coordinates/patches/blocks similar to current reference patch denoted as Zxr in above equations.

Equation 2: Transformed to a 2D operator typically a wavelet transforms into another domain. In this domain hard thresholding is applied and then inverse transform to get back denoise image.

There is a transformation of image into different representation and make assumption that this different representation is only needed to explain noise and thereby deleting the small coefficient using threshold operator and then transform it back.

So now we have S_{xr} a set of patches that does not look very different from the reference patch according to equation 2.

GROUP DENOISING:

In equation 3 LHS represents a (d+1) dimensional array holding denoised group and RHS is noisy (d+1) dimension group. It's a collection of patches set by index S. S : enumerates patches which are not different from pixels located at X_R.

In equation 4 LHS represents the denoised group which is a collection of all patches in that group.

We needed this extra notation to discuss aggregation step.

***AGGREGATION**:*

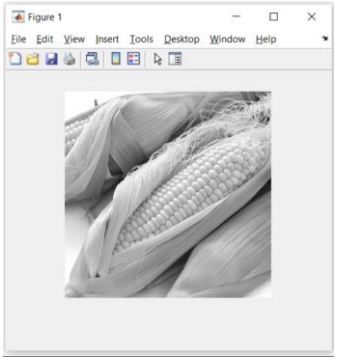
Refer Equation 5 : Each pixel in the noisy image has probably been a member of several different groups and by the collaborative filtering the patch around this pixel was modified in different ways in each of the groups. Now reconcile all these estimates.

Weighing is done by inverse amount of noise in that group and hence noisy groups gets smaller weights. And in the end normalize it. W gives larger weight to non-noisy/smooth groups.

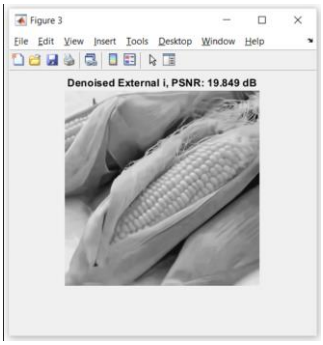
HOW TO PUT IT BACK IN ORIGINAL IMAGE:

If patches are dissimilar in group, then group will look noisy-very large coefficients in Wavelet transform and gets low weight. To combine patches into groups and then to denoise both along axis that span image and along extra new artificial axis that result from stacking all patches into a group.

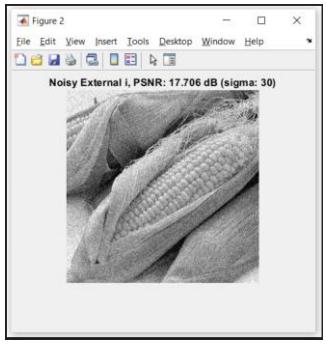
RESULTS:



Original_Image 1



Denoised_Image 2



Noisy_Image

DISCUSSION:

Performance of BM3D is better among all denoising methods because firstly we are not applying denoising algorithm on 3 channels that is red green and blue separately, but we are introducing a grouping constraint which is done only once.

Visually and according to PSNR BM3D performs better than denoising algorithms shown above.

As we increase sigma image becomes smoother and thus deblurring occurs therefore optimally sigma should be selected in order to get the noise free image without blurring.

Also, 3D group processing is done along spatial dimensions as well as third dimension and would simply average patches which are similar. This is the main difference from Nonlocal mean filter.

The second step retrieve al the essential details lost in step 1 thereby improving image contrast. This is done during patch matching where we keep relatively low maximum number of similar patches which provides flexibility in choosing the value for hard threshold.

Also, 2d patches are being denoised during the process of collaborative filtering gives a considerable gain for the aggregation as it gives large number of good estimates to each pixel.

Question2.e

1. In figure shown there are mixed noises present in it that is Gaussian noise and Salt and pepper noise.
2. It should be performed on three different channels
3. In order to remove Gaussian noise, we use gaussian filter and to remove salt and pepper noise we use mean filter
   Salt and pepper noise also referred as Impulse noise. Error in data transmission caused these types of noises

The corrupted pixels are set to minimum or maximum value causing salt and pepper like pattern. In camera sensors there is a malfunctioning of pixel elements.

Model of PDF of impulse noise:

$$pI(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

If b>a then gray level appears as salt (light dot) and a will appear as pepper (dark dot).

Median filtering can remove the salt and pepper noise without reducing the sharpness of the image as it is less sensitive than the other linear filters to extreme changes in pixel values.

PROCEDURE:

1. The median filter iterates over the image replacing each pixel with median values of the neighboring pixels.
2. The window which represent the pattern of neighbors which slides pixel by pixel over the entire image.
3. First sort the pixel values in ascending order and replace the desired pixel with the median pixel value.
4. Now we have removed salt and pepper noise

2. It should be performed filtering on individually channel separately because analyzing the noise profile separately on each channel and then denoising it is computationally less intensive as compared to denoise the RGB image as whole

3. Whenever there is linear as well as nonlinear noise profiles then the denoising is performed initially by linear filter followed by nonlinear filter in order to get the desired output result. We will cascade filter which Gaussian filter in order to get rid of uniform noise. We have already discussed above how to remove the noise using gaussian filtering.

REFERNCES:

[1] B.K Shreyamsha Kumar “IMAGE DENOISING BASED ON GAUSSIAN/BILATERAL FILTER AND ITS METHOD NOISE THRESHOLDING” DOI: 10.1007/s11760-012-0372-7

[2] Dabov, Kostadin, et al. "Image denoising with block-matching and 3D filtering." Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning. Vol. 6064. International Society for Optics and Photonics, 2006.

[3] Dabov, Kostadin, et al. "Image denoising with block-matching and 3D filtering." Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning. Vol. 6064. International Society for Optics and Photonics, 2006.

[4]. http://www.123seminarsonly.com/Seminar-Reports/029/42184313-10-1-1-100-81.pdf

[5] https://www.ipol.im/pub/art/2011/g_mhcd/article.pdf