EE569 Digital Image Processing

Homework#1

Wenbo Ye
wenboye@usc.edu

Issued date: 01/26/2020 due date:01/28/2020

1. Problem 1: Image Demosaicing and Histogram Manipulation
   a) Bilinear Demosaicing
      i. Abstract and Motivation

         To transfer the gray image into the RGB image, we should predict the other two colors channel on each pixel. We can use the neighbor pixel of centered one to get the other two value of gray level by calculating the average. Then the image can be shown as colorful picture.

      ii. Approach and Procedures

         Before processing the ". raw" format pictures, we are supposed to consider the boundary pixels. If we need the neighbor pixel to calculate the centered one which located on the edge, we must use the way of reflection to make sure that edge pixels have the same numbers of neighbor as other pixels.

         Make the most outside edge as a symmetry axis and overturn the 2 rows of pixels. Every 4 edges do the same actions. The following figure shows way to reflection.

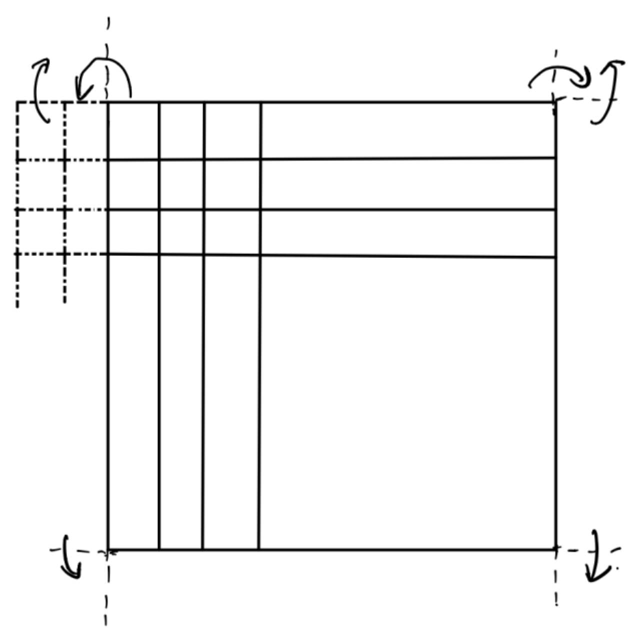
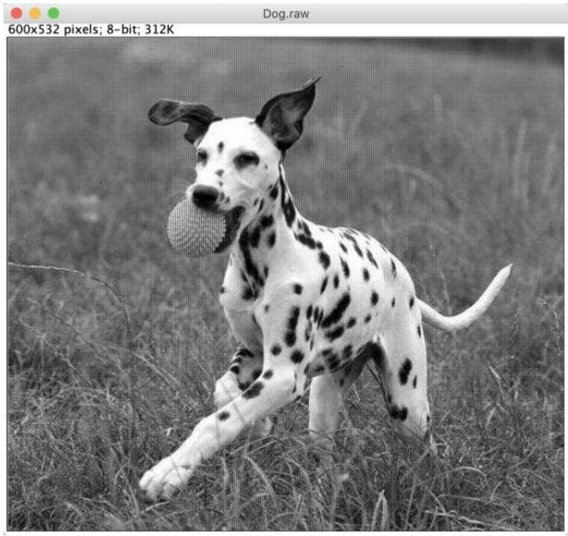
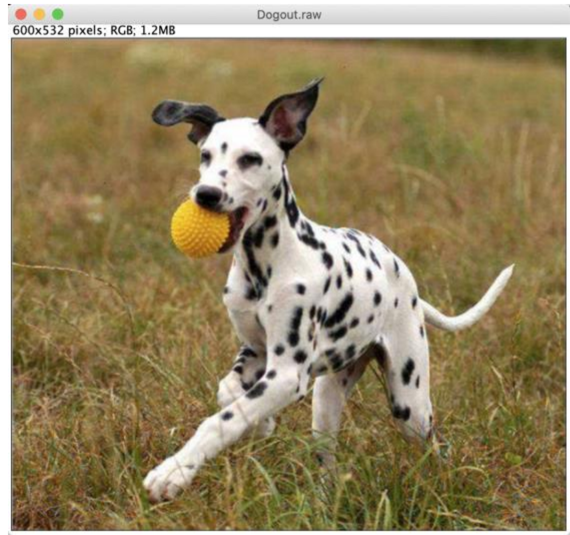


Figure 1. mirror reflection

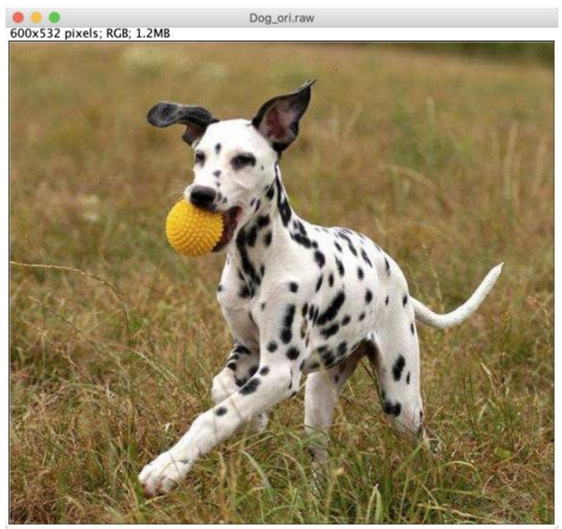
iii. Experimental Results



Dog.raw



Dog_out_Bilinear.raw



Dog_ori.raw

Figure2. result of bilinear demosaicing

iv. Discussion

After comparing bilinear-processed image with original image, we can see some obvious artificial on the processed one. The reason is that we assume all the weight of pixel are uniform. It's true that we can predict the gray level of RBG in some locations which pixels are highly assemble and similar, but we cannot easily get the result of original shape image (like the dog's spots in this picture). This blurs the shape of the image. To better process the image, we can add some deviation factor to predict the offset of each pixel, which is the way MHC Demosaicing do. This may help display more clear boundary of the shape.

b) Malvar-He-Cutler (MHC) Demosaicing

i. Abstract and Motivation

Giving that bilinear demosaicing cannot recover the gray level into RGB image very clearly, MHC is the improved method of demosaicing based on bilinear demosaicing.

ii. Approach and Procedures

First step of MHC demosaicing is to calculate the RGB channel on each pixel by bilinear demosaicing method. Then we should evaluate the gray level deviation by comparing the current gray level of pixel with expected value based on neighbor pixel in same channel. Finally, we fixed the deviation by adding it back to other 2 color channel.

The format for a single pixel shows as follows (a green component at a red pixel location).

iii. Experimental Results

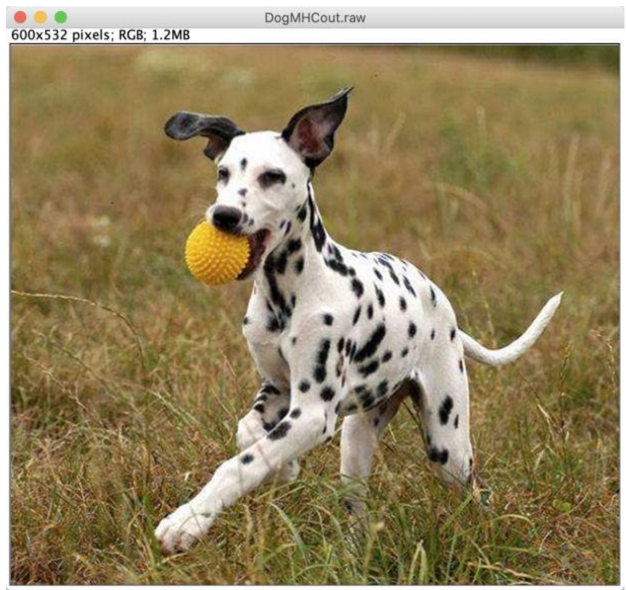


Figure3. result of MHC Demosaicing

iv. Discussion

In MHC method, we can see some tiny improvement on the edge of shape and some area with multiple colors, which is explained by the fixed value $\Delta_{(R,G,B)}$.

So, the performance of demosaicing is better than bilinear.

c) Histogram Manipulation

i. Abstract and Motivation

Given an image whose gray value are in close interval. It might be too light, dark or monotonous for people to see the picture. To solve the problem and improve the hierarchy of the image, there are two ways of histogram manipulation to process the image by re-divide the gray level on each pixel.

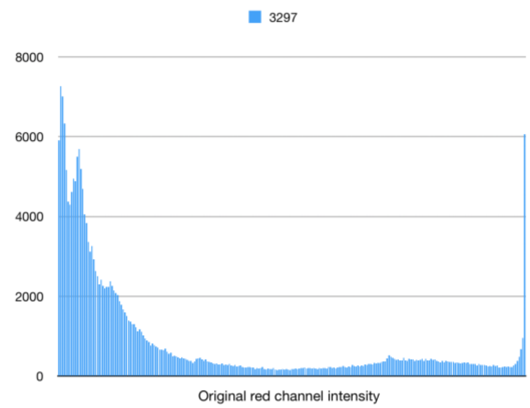
ii. Approach and Procedures

One way to process the histogram manipulation is called as transfer-function-based histogram manipulation. First, calculate the frequency of gray value appeared in images from 0 to 255. Second, calculate the intensity function and cumulative function. Using the cumulative density function to reassign the pixel level and get the final result.

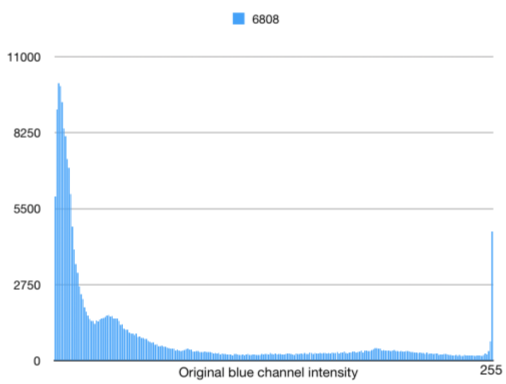
The second way is called bucket filling. First, also get the results of pixel gray value frequency. Second, create 256 buckets from 0 to 255, and filling all the pixel gray value into the bucket evenly by order. Then every pixel get its new gray level and image become more mild.

iii. Experimental Results

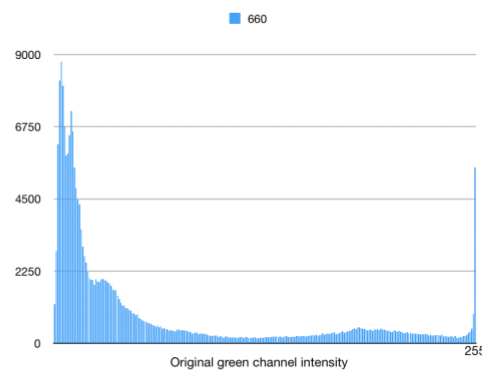
Original image



original red pixel intensity



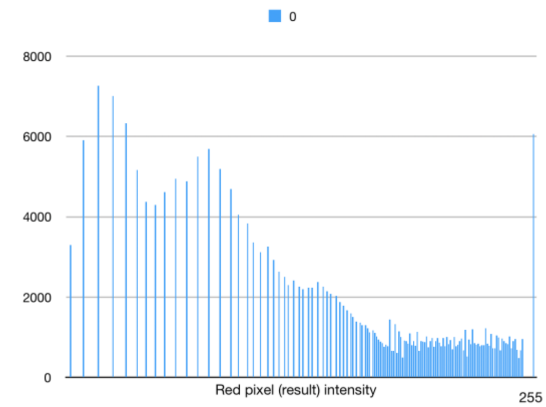
original green pixel intensity



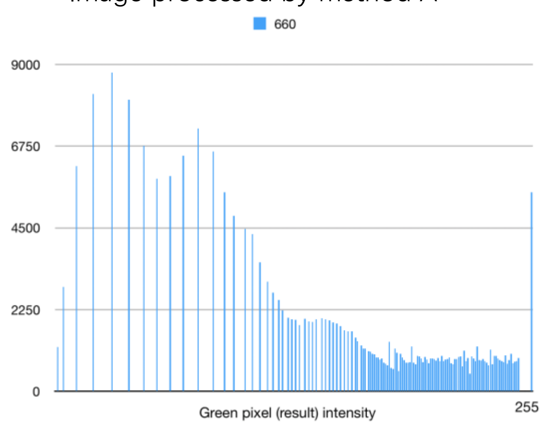
original blue pixel intensity

Figure4. Histogram of original picture in three channel pixel

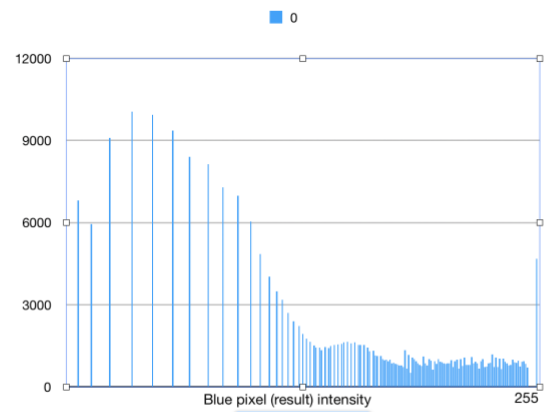
Image processed by method A



methodA red pixel intensity



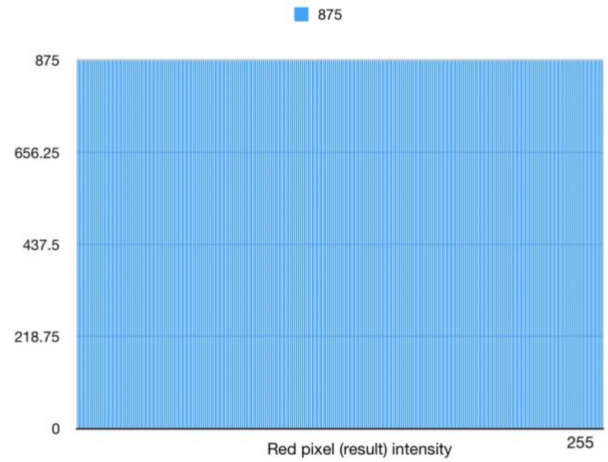
methodA green pixel intensity.



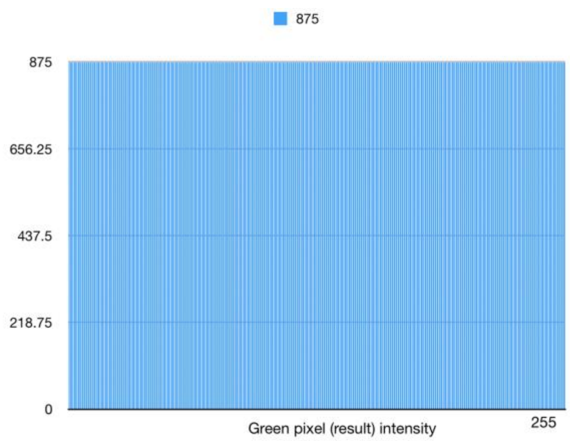
methodA blue pixel intensity

Figure5. Histogram manipulation after transfer-function-based method

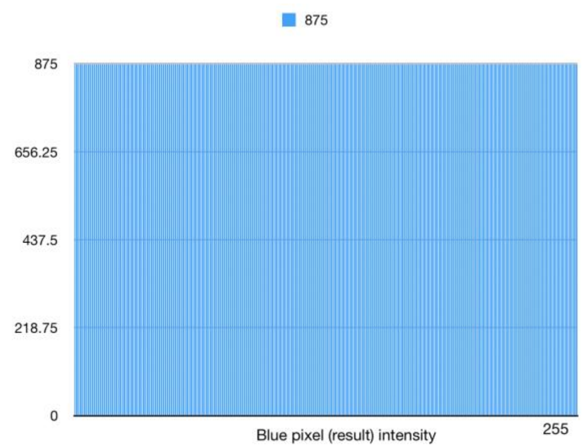
Image processed by method B.



methodB red pixel intensity



methodB green pixel intensity.



methodB blue pixel intensity

Figure5. Histogram manipulation after bucket filling method

iv. Discussion

For method A, the result shows that intensity of grey level are discreate which there may be a huge span between two non-zero value. This may make the image and the shape become too sharp to observe. For method B, every value of gray level has the same number of pixels, which makes the picture become too smooth to recognize the obvious boundary. So, I think it might be better if we can merge two method thought into one action. First, we can apply method A to the image and then apply the thought of method B, filling the span of gray level intensity by nonzero value between the zero interval. Then the picture may become smoother than before.

2. Problem 2: Image Denoising

Background Corn_noise PSNR = 17.7062

a) Basic denoising methods

i. Abstract and Motivation

There are two different ways to help remove the noise on image. First is improve the physical device, second is using the algorithm to process the image. It is absolutely that former way can be much expensive and hard to realize in different situation. The uniform weight function and the Gaussian weight function filters are two common methods in this realm. Both can easily help to remove the noise in different extent.

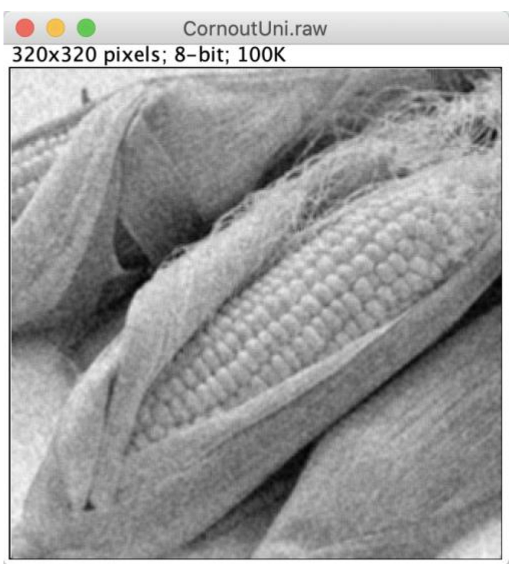
ii. Approach and Procedures

First, we should manipulate the boundary of the picture and make sure while applying the filter, code goes well on pixels on margin.

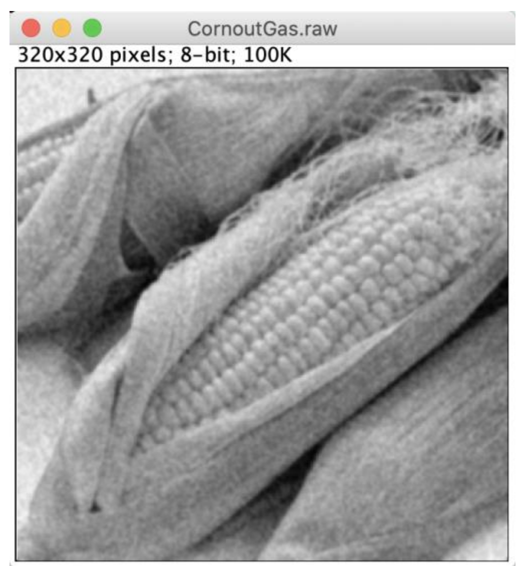
Second, we can apply different filters on each pixel. As for uniform weight function, size can be N and all weight in sliding window are uniform. As for Gaussian weight function, numbers of weight are supposed to be displayed as Gaussian distribution referring on centered location in window.

Finally, calculate the result of image, divide the sum of weighted grayscales by sum of weight and get images.

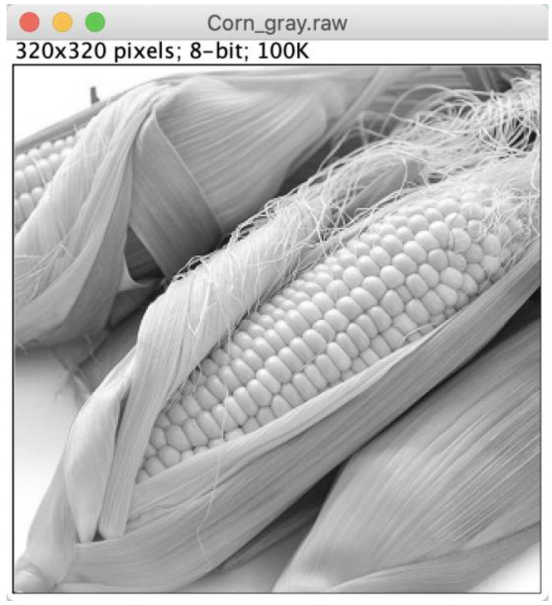
iii. Experimental Results



Uniform



Gaussian



Normal

Figure6. basic denoising methods result

iv. Discussion

It's Gaussian white noise.

The result of 2 filter on denoising method is quite assemble with trivial difference. By using uniform filter and set the size of window as N=3, we get the PSNR=19.3083; By using uniform filter and set the size of window as N=5, we get the PSNR=19.3715. We can see there are some little improvement thanks to the filters' weight are of Gaussian distribution.

b) Bilateral Filtering

i. Abstract and Motivation

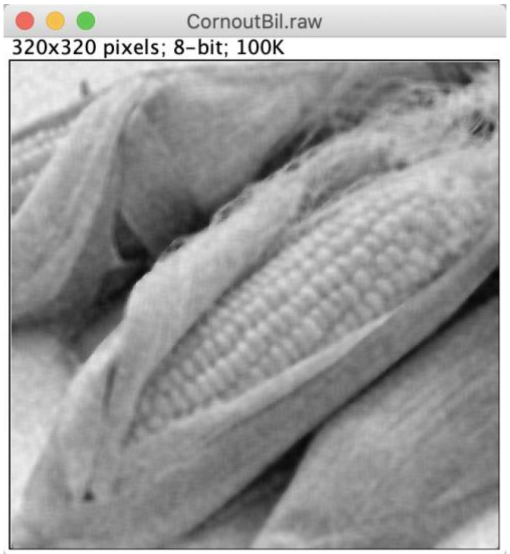
The Bilateral filter is a nonlinear filtering method. It is a compromise processing combining spatial proximity and pixel value similarity of the image and considering spatial information and gray level similarity at the same time to achieve the purpose of edge preserving and denoising.

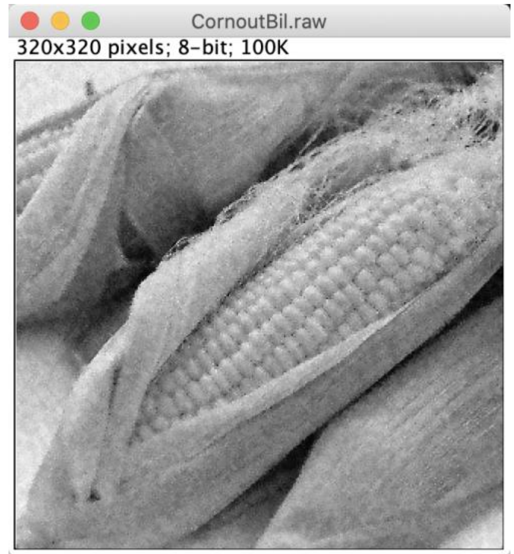
ii. Approach and Procedures

For gaussian filtering, the gray value of the center point is determined only after the convolution of the kernel of the weight coefficient of the spatial distance with the image. That is, the closer the point is to the center, the greater the weight coefficient.

The weight of grayscale information is added to the bilateral filtering, that is, in the neighborhood, the point whose grayscale value is closer to the center point's grayscale value has a greater weight, and the point whose grayscale value is more different has a smaller weight. The weight is determined by the range gaussian function.

iii. Experimental Results





σ_s = 100 and σ_c = 100, PSNR = 19.2043. σ_s = 10 and σ_c = 10, PSNR = 19.1296

Figure7. result of Bilateral denoising

iv. Discussion

While σ_s denotes the Range standard deviation, which can help calculate the distance between current spot and centered spot. And σ_c Spatial standard deviation, which is the factor of absolute value of the difference between the current point grayscale and the center point grayscale. They all satisfy the Gaussian distribution.

In the experiment, The larger the Sigma, the vaguer the edge, the limit case is Sigma infinity, the range coefficient is approximately equal (ignoring the constant, nearly exp (0) = 1), and the gaussian template (spatial template) can be considered equivalent to gaussian filtering after multiplying.

The smaller the Sigma, the clearer the edge, the limit case is Sigma infinite close to 0, the range coefficient is approximately equal (close to exp (-∞) = 0), and the gaussian template (space template) after multiplying, can be approximately equal to the coefficients, equivalent to the source image.

To get the better PSNR, I selected σ_s = 100 and σ_c = 100 and get PSNR = 19.2043, σ_s = 10 and σ_c = 10 and get PSNR = 19.1296. which is much better than the original PSNR = 17.7062.

c) Non-Local Means (NLM) Filtering

i. Abstract and Motivation

Non-Local Means is a new denoising technique proposed in recent years. This method makes full use of the redundant information in the image and can keep the detail features of the image to the maximum extent while removing the noise. The basic idea is that the estimated value of the current pixel is obtained by the weighted average of the pixels in the image that have a similar neighborhood structure to it.

ii. Approach and Procedures

In theory, the algorithm needs to judge the similarity between pixels in the whole image range, that is, every pixel is processed to calculate the similarity between it and all pixels in the image. Actually, we should set the size of window to make sure the code can be run on computer in limited time. The neighborhood window slides in the search window and determines the pixel weight according to the similarity between the neighbors.

iii. Experimental Results

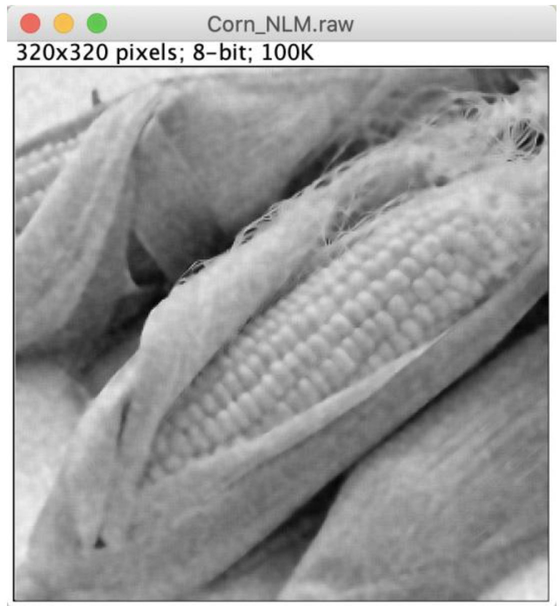


Figure8. result of NLM denosing

iv. Discussion

According to the performance and PSNR value, I can say NLM filter can be more reliable than uniform weight filter or Gaussian weight filter. Uniform or Gaussian weight filter is an easy application based on the distance between centered pixel and nearby pixel. The consideration is limited in only one factors locally. Non-Local Mean filter uses the full range of the picture in theory and also think about the neighborhood similarity relationship between target pixel and centered pixel. When I used the search window size = 5, neighbor window size = 5, I get a smoother picture as a result and PSNR = 19.504 which is higher than uniform and gaussian weight filter application.

Compared with Bilateral filter, which is an improved function based on uniform and gaussian weight filter, in this picture, I cannot obviously see the difference and PSNR are similar. Indeed, NLM is wider used. But NLM also have a disadvantage which cost too much time on algorithm to process the image.

d) Block matching and 3-D (BM3D) transform filter

i. Abstract and Motivation

BM3D is the upgraded version of NLM (non-local mean), because it mainly uses the idea of non-local block matching. Firstly, similar blocks are found. Different from traditional NLM, which uses L2 distance, BM3D uses hard threshold linear transformation, which reduces the complexity of L2 distance. After finding the similar block, NLM does a mean processing, while BM3D converts the similar block domain, proposes Collaborative filtering to reduce the noise contained in the similar block itself (NLM does the mean and introduces the noise of the similar block), and weights the similar block at the aggregation to obtain the target block after noise reduction.

ii. Approach and Procedures

There are two steps: Basic estimate and Final estimate. The two steps are similar except for Collaborative filtering. Among them:

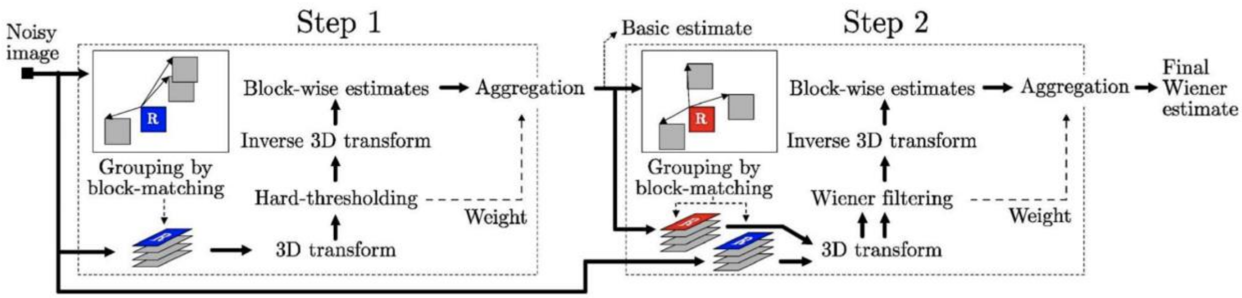


Figure9. step of BM3D

Step1: Basic estimate:

1.1 Block - wise estimate

1.1.1 Grouping finds similar blocks for the target block, and the block set has no order limit.

1.1.2 Collaborative hard thresholding uses a domain transformation method to obtain a "transformed set of similar blocks". It USES a Collaborative (determined by the similar blocks) hard threshold strategy to "weaken" the noise of similar blocks and inversely transforms back to the original block field.

1.2 Aggregation weighted average similar block (after 1.1.2 treatment) was added to obtain the target block after basic estimate.

Step2: Final estimate:

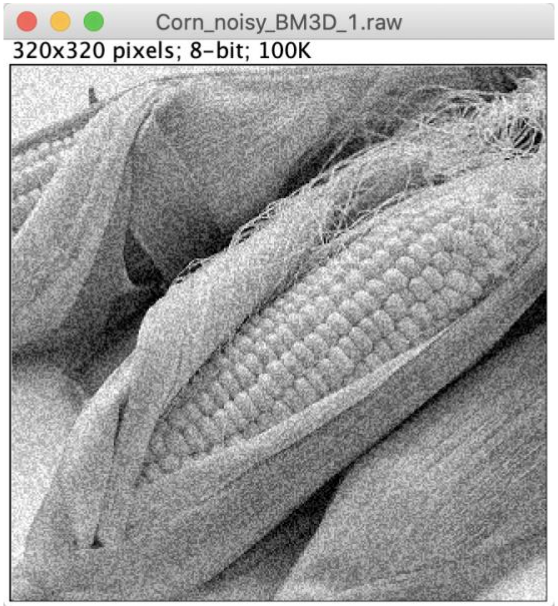
2.1 Block - wise estimate

2.1.1 Grouping uses the block set of 1.1.1 steps and the image processed by Step1 to recalculate the block set

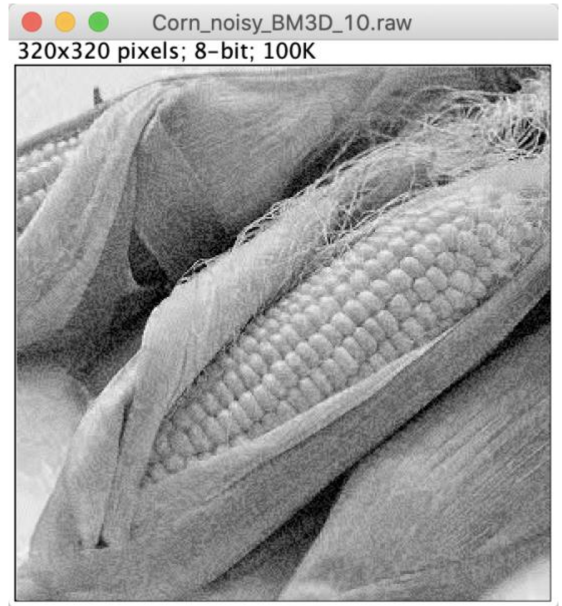
2.1.2 After filtering domain transform, Wiener filtering is used

2.2 Aggregation

iii. Experimental Results

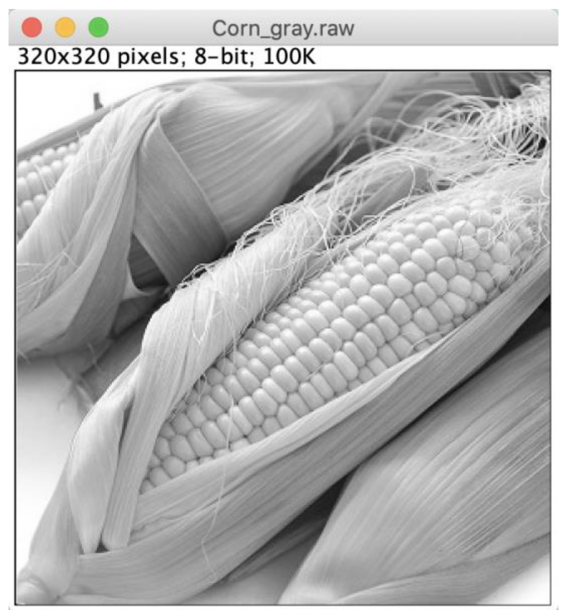


Sigma = 1, PSNR = 17.7175.



Sigma = 10, PSNR = 19.0154

Sigma = 20, PSNR = 19.4880



normal

Figure10. result of BM3D

iv. Discussion

The result shown in picture can simply draw a conclusion that in specific range, the larger sigma, the larger PSNR value. It shows that picture can be smoother with the increasement of the Sigma but it also loses more details.

e) Mixed noises in color image

i. Discussion

This picture is mixed by Gaussian noise and impulse noise (Salt and Pepper Noise).

If we just apply one filter to denoise the image, performance will not be good enough.

For gaussian noise, the distribution of the noise is random based on gaussian distribution and range is (0, 255) grayscale. For impulse noise, the grayscale is almost 0 or 255. To solve the gaussian noise, we can use the mean filtering. In contrary, median filtering is more suitable to remove the impulse noise, by decreasing the influence caused by the limit value. We cannot simply cascade the different filter because the performance is on dependence. We should take window size, factor in the filters, RGB channels into consideration, cascade can be in any different orders.