DSP Project: Image Processing

Instractor: Federica Battisti

Author: Ali Bavarchee

Digital Image processing

Image processing mostly refers to the modifying digital images by computer. In fact, the images are manipulated for different purposes, e.g commercial photos need to some retouch to enhance the aesthetic aspects while medical diagnostic pictures are required to enhance the contrast and and removing the noise on them, etc.

Motivation and map road : IP based on the image database TID2013

In the present project tried to restore three type of noises from pre-cooked distorted images from TID2013, purposively tree types of: addictive white Gaussian noise(AWGN), addictive in color components and high frequency noise, and then compare them with their references by calculating two metrics, the signal to noise ratio and mean squared error.

Some words about TID2013

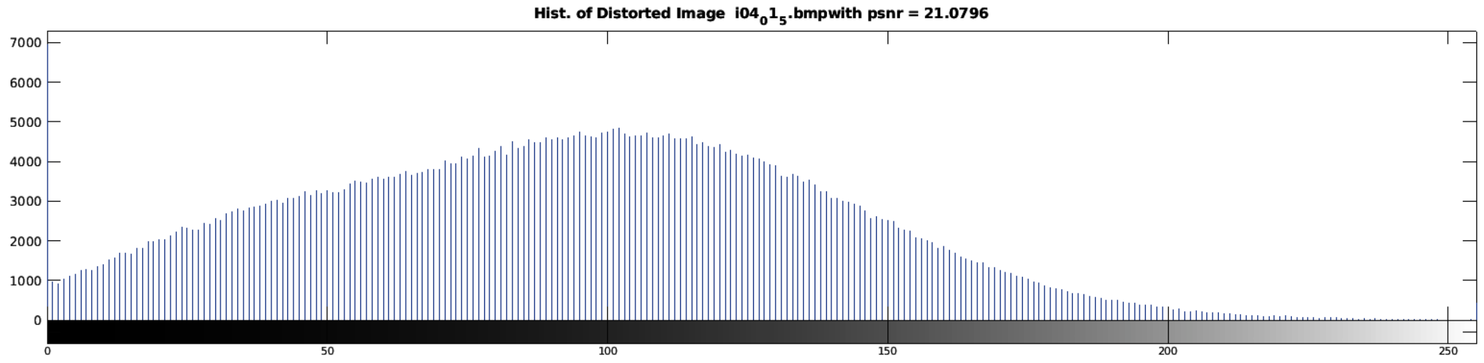
Image database TID2013 contains 25 reference pictures and each of them smeared with 24 different distortions. Also each noises which implemented to images has different intensity marked as level 1 (the least noise) to level 5 (saturated). The objects of pictures are vary from still life and nature to portrait and artificial image. The aim the TID2013 was evaluating the full reference visual quality assessment metrics. For this purpose, by an experiment was conducted by extracting the mean opinion score of images of TID2013. Thanks to peculiarities and enlargement of the set of images of TID2013, it would be obtained the the performance of photo camera by investigation of MOS adequate to HVS and thereunder, it would be helpful to design the photo camera and its exploited software. The distortions have been added to ref. pictures by using software.

Distortions

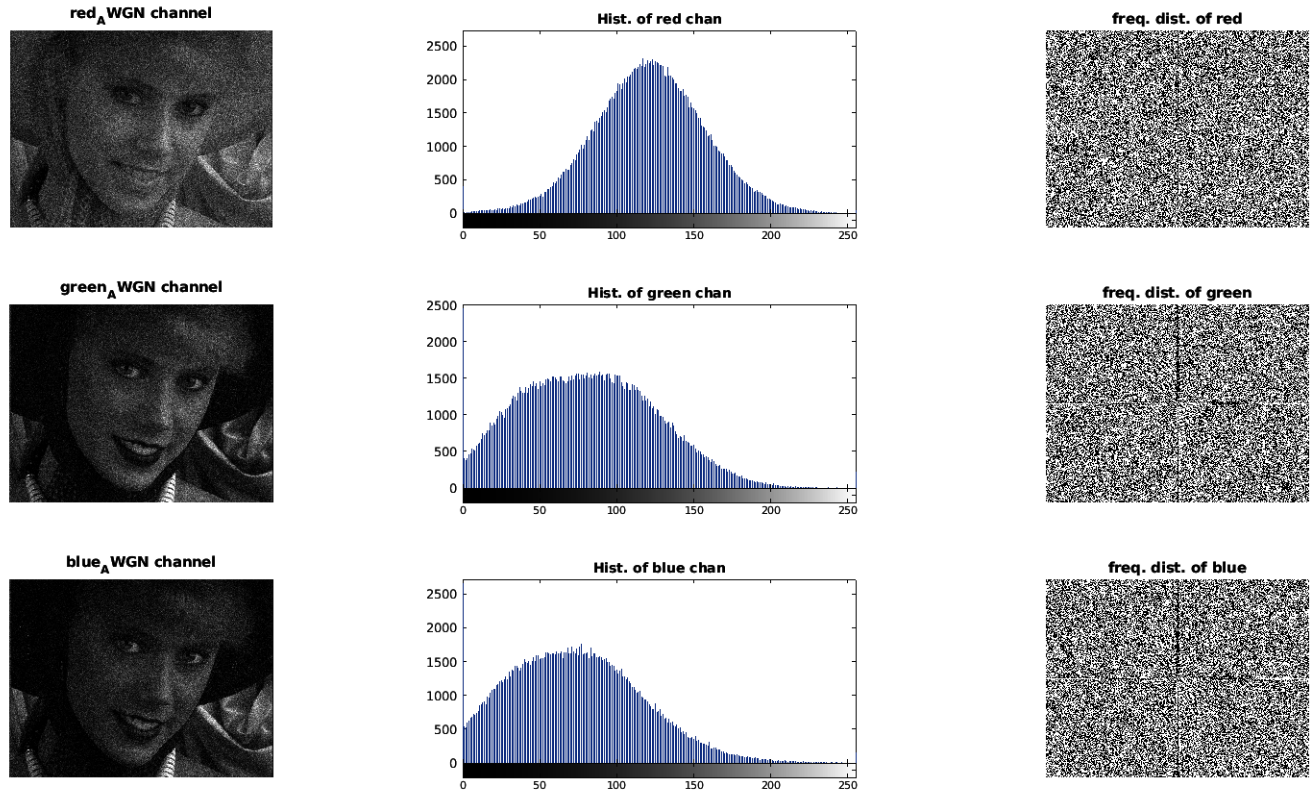
AWGN

AWGN is a statistical noise which is came across the uniform power across the frequency band and it distributes to the whole of image govern by normal distribution. In stochastic processes aspects, this type of random "white noise" to be added to original data(here image) by Gaussian process and its noise power spectrum is constant.

- imshow and histogram of image i04 with addictive gaussian noise (level 5)



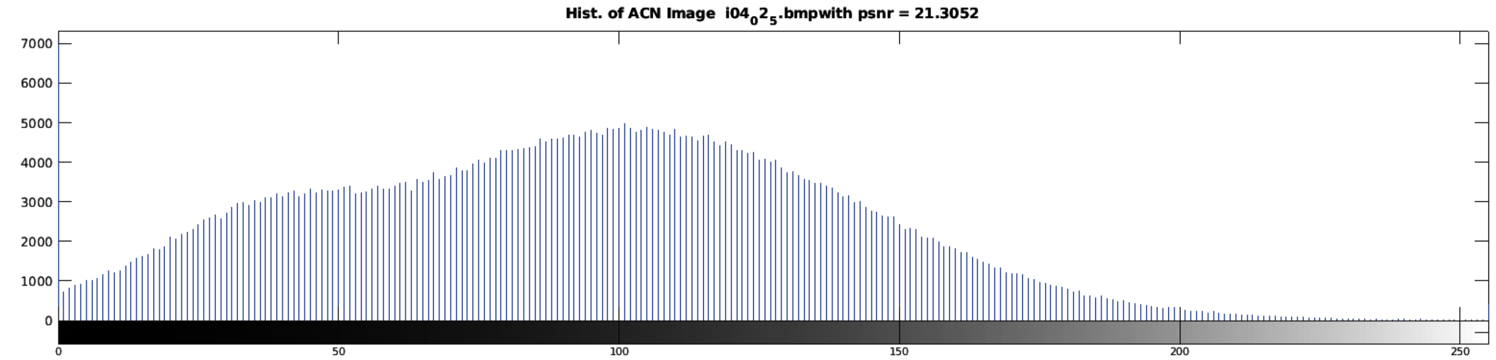
- imshow and histogram of each RGB channels of image i04 with addictive gaussian noise (level 5)



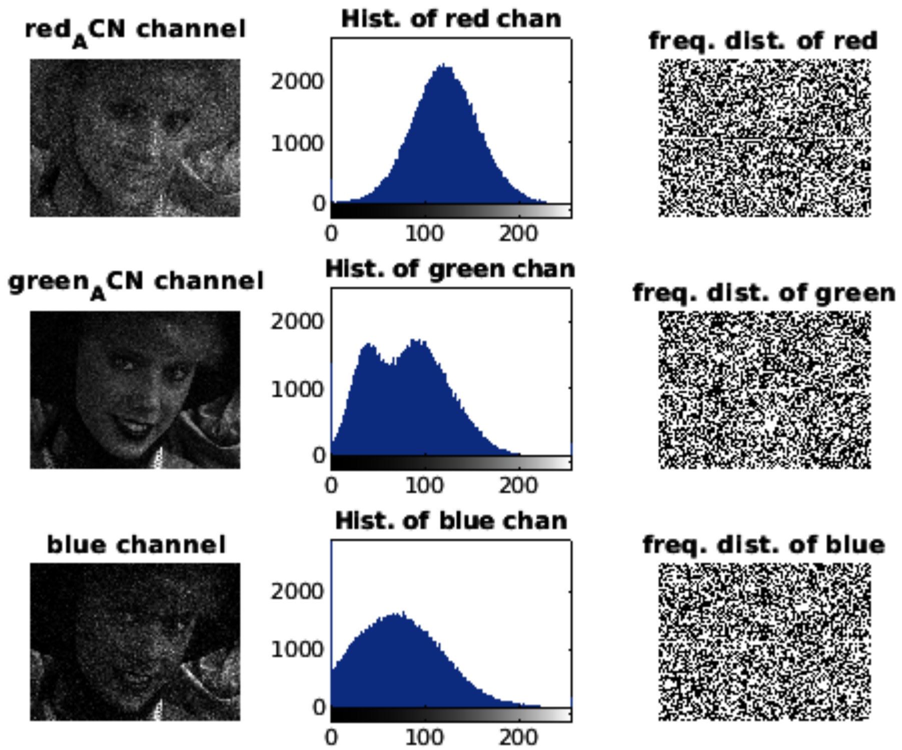
Addictive noise in color components (ACN)

When AWGN is involved with all RGB components, hence it affects more to the image.

- imshow and histogram of image i04 with addictive gaussian noise in color components (ACN)(level 5)



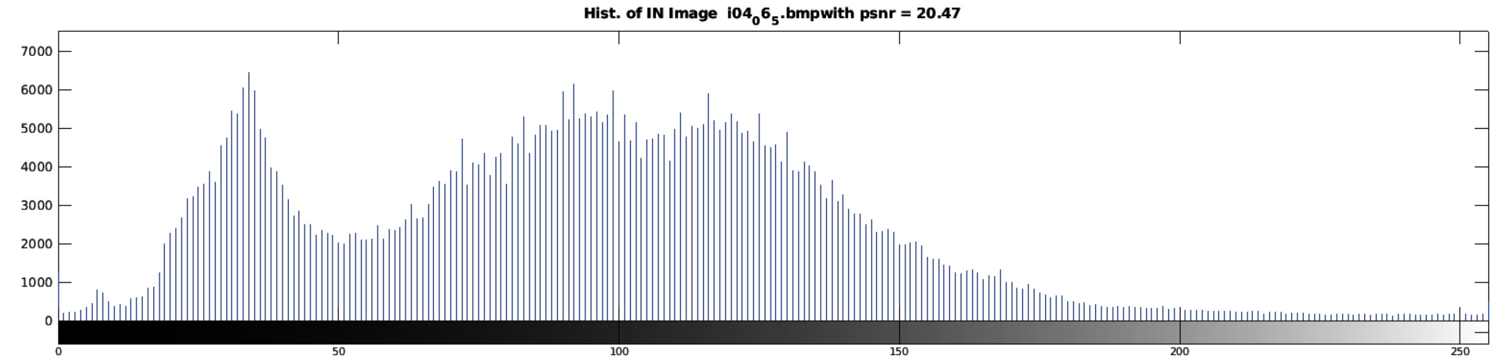
- imshow and histogram of pixels and frequency distribution of RGB channels of image i04 with ACN(level 5)



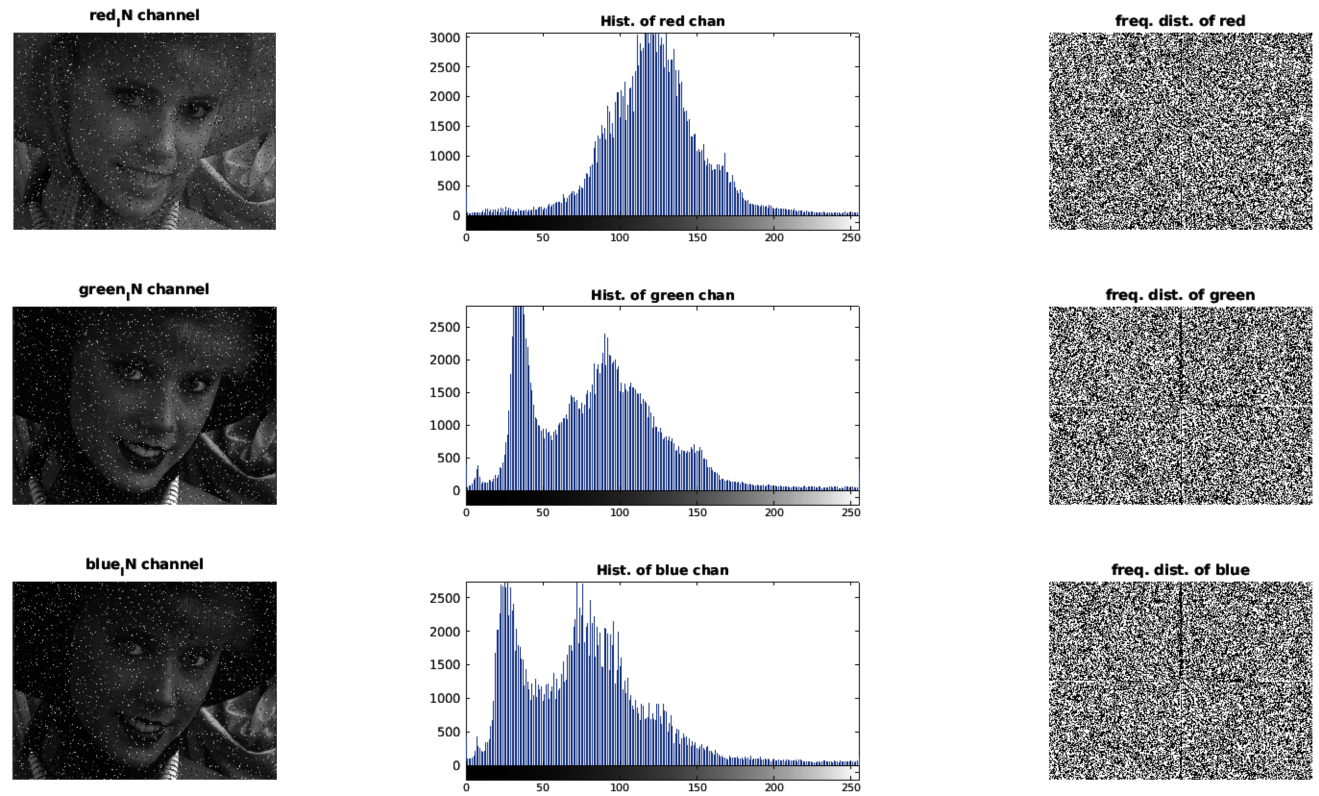
Impulse noise (IN) a.k.a salt and pepper

Salt and Pepper noise or something called data drop noise caused by the random introduction of white (high) and black (low) pixels in to the image. It means that 0's or 255's are replaced by real value of the pixels.

- imshow and histogram of image i04 with Impulse Noise(IN) (level 5)



- imshow and histogram of pixels and frequency distribution of RGB channels of image i04 with ACN(level 5)



Typical IP operations

There is a wide and diverse methods and techniques to remove noise and/or retrieve the quality of an image. For instance Sharpening, Noise Removal, Deblurring, Contrast Enhancement etc.which accomplished mainly by the Image Filtering. The goal of IF is the modification of image properties by implementing a filter. One could classified IFs based on their operations; operations in spatial domain where arithmetic calculations and logical operations are performed on the original pixel values and Neighborhood-Oriented operations is one of the main functions of that. In the Neighborhood-Oriented operations or area operations, the input image is treated on a pixel by pixel basis and the resulting value for a processed pixel is a function of its original value and the values of its neighbours. The spatial-domain filters are the examples of this operations. Operations in a Transform Domain is another main category of IP operations. In this category, the image undergoes a mathematical transformation- FFT for instance, the algorithm works in the transform domain. Frequency-domain filtering techniques are from this category.

Filters

Linear filters

By imposing linear filters, the resulting output pixel is computed as a sum of products of the pixel values and the mask coefficients in pixel's neighborhood in the reference image. The well-known linear filter is mean filter.

Non-linear filters

In these type of filters, the outcome of pixel is selected from an ordered or ranked sequence of pixel values in the pixel's neighborhood in the reference image. Median and ranked filters are non-linear filters.

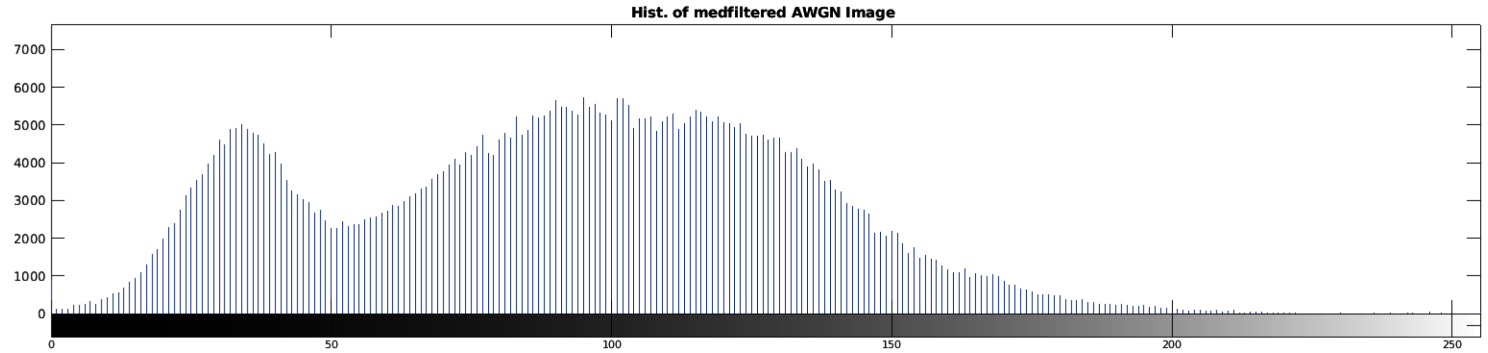
Median filter

As mentioned, the most popular nonlinear filter in IP is the median filter. It works by sorting pixel values with a neighborhood, finding the median value and replacing the original pixel value with the median of its neighborhood. Generally the median filter is very efficient to removing the impulse noise and also high frequency noise. Here median filter was be imposing to each channel of the 5th grade distorted image with a 5×5 convulsion mask. Because the the images are in color, it is required (only for 2D filters) to separate the RGB channels and then impose the filters to each color and finally concatenate the filtered channels. As one could notice in the results, med filter has a acceptable performance on the impulse noise(IN) where the MSE reduced from before IP 583.5563 to 39.0515 in comparison to the reference image.

% Median filter - AWGN distortion Rmed_AWGN01= medfilt2(red_AWGN01, [5,5]); FOR RED CHANNEL Gmed_AWGN01 = medfilt2(green_AWGN01, [5 5]); FOR GREEN CHANNEL Bmed_AWGN01 = medfilt2(blue_AWGN01, [5 5]); FOR BLUE CHANNEL medfil_img_AWGN01 = cat(3, Rmed_AWGN01, Gmed_AWGN01, Bmed_AWGN01); Psigtonoi2_AWGN01 = psnr(medfil_img_AWGN01, I); figure() title('AWGN distor.') subplot(2,1,1) imshow(medfil_img_AWGN01) title(['medfiltered AWGN Image with psnr =

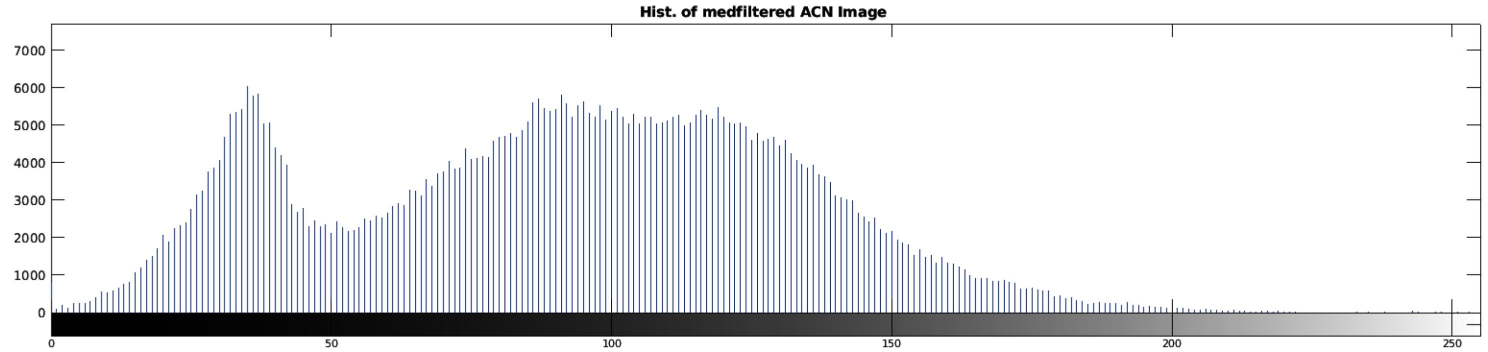
',num2str(Psigtonoi2_AWGN01)]) subplot(2,1,2) imhist(medfil_img_AWGN01) title('Hist. of medfiltered AWGN Image ')

- imshow and histogram of image i04 with AWGN(level 5) after medfiltering



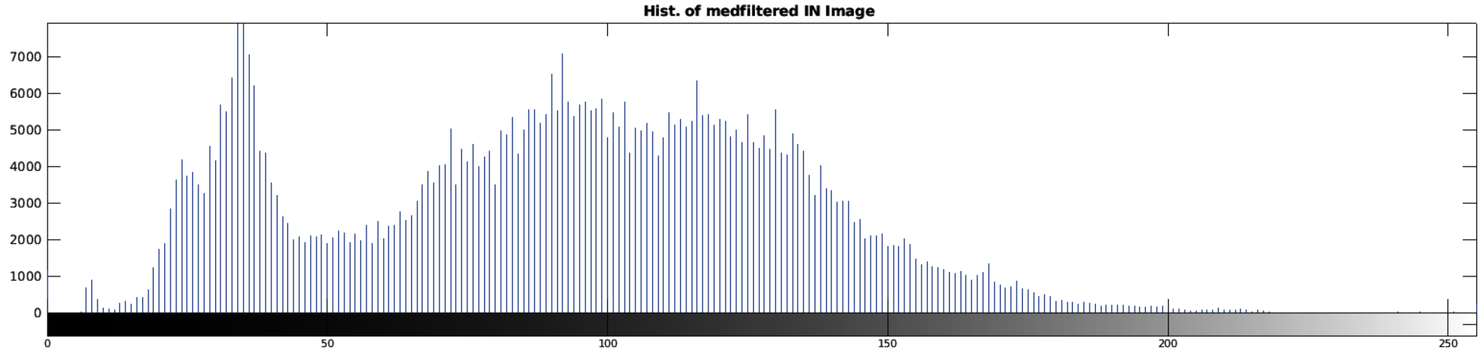
% Median filter - ACN level IV Rmed_ACN= medfilt2(red_ACN, [5,5]); % red channel Gmed_ACN = medfilt2(green_ACN, [5 5]); % green channel Bmed_ACN = medfilt2(blue_ACN, [5 5]); % blue channel medfil_img_ACN = cat(3, Rmed_ACN, Gmed_ACN, Bmed_ACN); Psigtonoi2_ACN = psnr(medfil_img_ACN, I); figure() subplot(2,1,1) imshow(medfil_img_ACN) title(['medfiltered ACN Image with psnr = ',num2str(Psigtonoi2_ACN)]) subplot(2,1,2) imhist(medfil_img_ACN) title('Hist. of medfiltered ACN Image ')

- imshow and histogram of image i04 with ACN(level 5) after medfiltering



% Median filter - IN level IV Rmed_IN6= medfilt2(red_IN6, [5,5]); Gmed_IN6 = medfilt2(green_IN6, [5 5]); Bmed_IN6 = medfilt2(blue_IN6, [5 5]); medfil_img_IN6 = cat(3, Rmed_IN6, Gmed_IN6, Bmed_IN6); Psigtonoi2_IN6 = psnr(medfil_img_IN6, I); figure() subplot(2,1,1) imshow(medfil_img_IN6) title(['medfiltered IN Image with psnr = ',num2str(Psigtonoi2_IN6)]) subplot(2,1,2) imhist(medfil_img_IN6) title('Hist. of medfiltered IN Image ')

- imshow and histogram of image i04 with salt and pepper noise(level 5) after medfiltering

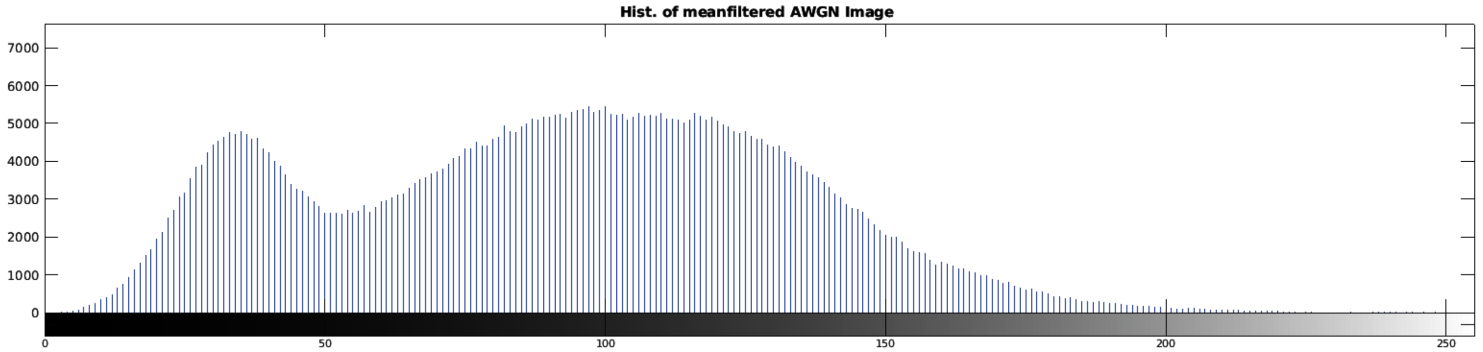


Mean Filter

Mean filter a.k.a neighborhood averaging is the most widely know spatial smoothing filter. It uses convolution with a mask whose coefficients have a value of 1 divides the total number of elements in the mask which is the scaling factor. Here the imfilter built-in function of MATLAB was used to create 3D mean filter. It means it is not necessary to detachment of channels of the image. Also two type of mean filters have been exploited, uniform and non-uniform mean filter. Generally the results of mean filter on the noised images are is acceptible.

% Mean Filter: h_mean = fspecial('average'); meanfil_img1_AWGN01 = imfilter(N_AWGN01, h_mean); % AWGN distor. Psigtonoimean_AWGN01 = psnr(meanfil_img1_AWGN01, I); % AWGN distor. figure() subplot(2,1,1) imshow(meanfil_img1_AWGN01) title(['meanfiltered AWGN Image with psnr = ',num2str(Psigtonoimean_AWGN01)]) subplot(2,1,2) imhist(meanfil_img1_AWGN01) title('Hist. of meanfiltered AWGN Image ')

- Mean Filtered AWGN



meanfil_img1_ACN = imfilter(N_ACN, h_mean); % ACN Psigtonoimean_ACN = psnr(meanfil_img1_ACN, I); % ACN figure() subplot(2,1,1) imshow(meanfil_img1_ACN) title(['meanfiltered ACN Image with psnr = ',num2str(Psigtonoimean_ACN)]) subplot(2,1,2) imhist(meanfil_img1_ACN) title('Hist. of meanfiltered Image ')

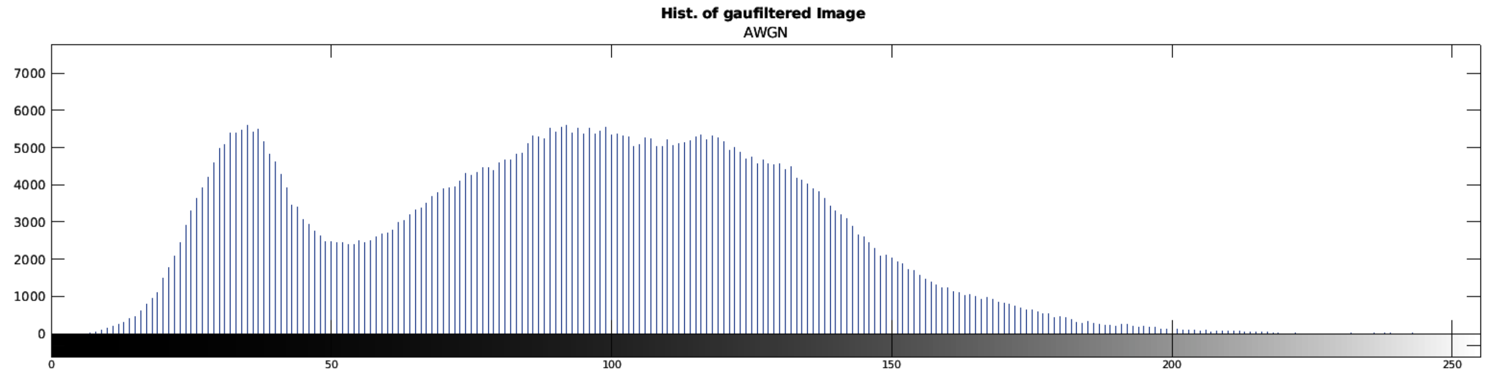
- Mean Filtered ACN

Gaussian filter

Gaussian filter was implemented by imfilter function like Mean Filter. The Gaussian filter filters the image by using a discrete kernel drived from a rapidly symmetric form of the gaussian function. Discrete approximations to the continuous the gaussian function are specified using two free parameters: 1. The arbitrary size of the kernel (mask) which was agin 5 × 5 in our case, and 2. The value of σ, the standard deviation of the gaussian function which is 2 in this code. As anticipated, the results of Gaussian filter on AWGN and ACN are a bit better than mean filter.

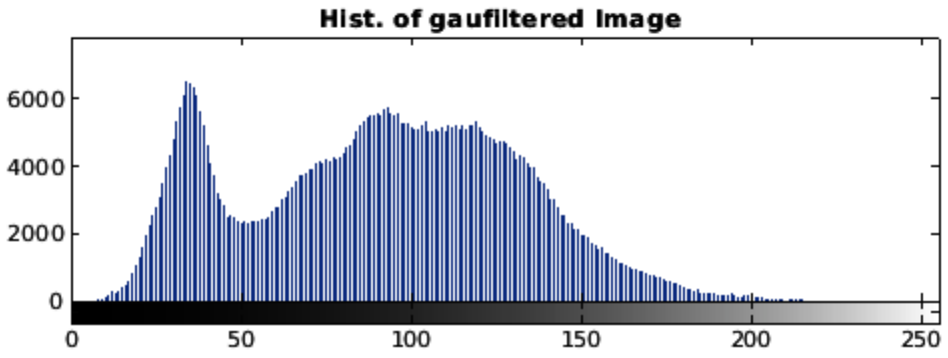
% Gaussian filters: h_gau = fspecial('gaussian', 5,2); % sigma = 2 % AWGN gaufil_img_AWGN01 = imfilter(N_AWGN01, h_gau) ; Psigtonoigau_AWGN01 = psnr(gaufil_img_AWGN01, I); % AWGN figure() subplot(2,1,1) imshow(gaufil_img_AWGN01) title(['gau-filtered Image with psnr = ',num2str(Psigtonoigau_AWGN01), 'AWGN']) subplot(2,1,2) imhist(gaufil_img_AWGN01) title('Hist. of gaufiltered Image ', 'AWGN')

- Perfomance of Gaussian Filter with sigma = 2 over AWGN



%ACN gaufil_img_ACN = imfilter(N_ACN, h_gau) ; Psigtonoigau_ACN = psnr(gaufil_img_ACN, I); %ACN figure() subplot(2,1,1) imshow(gaufil_img_ACN) title(['gau-filtered ACN Image with psnr = ',num2str(Psigtonoigau_ACN)]) subplot(2,1,2) imhist(gaufil_img_ACN) title('Hist. of gaufiltered Image ')

- Perfomance of Gaussian Filter with sigma = 2 over ACN



Analysis of mean squared error

There are different type of metrics to evaluate the performance of filters and outcomes and IP in generall. Mean squared error(MSE) along with peak signal to noise ratio could be investigated, especially in this work that the original images are present.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2$$

where e_is

$$e_i = Y_i - \hat{Y}_i$$

are the errors. The "immse" function in MATLAB replys the MSE between two arrays. Hence it is usually applied to exratct the MSE in IP.

- Befor and After the imposing Guassian filter to AWGN

before with mse = 507.1353AWGN

after with mse = 76.6704AWGN

- Befor and After the imposing Mean filter to ACN

ACN - before with mse = 481.4566

after with mse = 85.9606

- Befor and After the imposing Median filter to IN

IN - before with mse = 583.5563

after with mse = 39.0515

Consequences

By looking at the results specially at MSE, one could say that the median filter is adjusted to Impulse Noise, which is expected as the inheriant of this filter. Guassian filter and Mean filter suite to addictive noise distortions but as it deeds smoothing, therefore extra treatment like sharnening and border segregation are requared.

References:

- Ponomarenko, Nikolay, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Lina Jin, Jaakko Astola, Benoit Vozel et al. "Color image database TID2013: Peculiarities and preliminary results." In european workshop on visual information processing (EUVIP), pp. 106-111. IEEE, 2013.
- Solomon, Chris, and Toby Breckon. Fundamentals of Digital Image Processing: A practical approach with examples in Matlab. John Wiley & Sons, 2011.
- Marques, Oge. Practical image and video processing using MATLAB. John Wiley & Sons, 2011.
- Charbit, Maurice, and Gerard Blanchet. Digital Signal and Image Processing using MATLAB, Vol. 1. ISTE-WILEY, 2014.