Pooja Voladoddi
6571-9160-65
voladodd@usc.edu

EE569 – Homework #1 Report
September 22, 2013

1. IMAGE MANIPULATION

1a.) BLUE SCREEN TECHNIQUE

I. Abstract and Motivation

The Blue Screen Technique is a mechanism used to combine seamlessly- two or more images (or videos) – by using one of them as a background and the other as a foreground. This technique is often used for filming purposes when one of the following scenarios occurs:
→ Location constraints
→ Shooting stunt scenes – both of the above resulting in shooting scenes at a studio in front of a blue or a green screen, creating a layer of desired background and compositing them during Post-Production.
→ Blend the interaction of real world objects or actors with non-existential, computer generated effects or images (like in science fiction movies)

The procedure in which this works is that the actors shoot in front of a big, bright blue screen as the background that can be rendered black by passing the film through any other colour filter (Red, for example) during post-production. (The concept behind this lies in Physics of light - a colour light filter when exposed to other colour lights than itself, absorbs them, allowing only wavelengths same as itself to pass through. This also drives home the point that any object desired to be in the foreground cannot be the same colour as the green screen since it will lead to those objects disappearing during compositing.) The foreground is then blended into a separately shot, desired output background image or video.

In essence, the technique is used to blend background and foreground shot separately resulting in a seamless rendering of real scenarios.

II. Approach and Procedures

In the problem, a 24-bit RAW colour image ‘shapes.raw’ is provided with concentric circles and squares (hence named ‘shapes’) located on a blue background, analogous to the ‘Blue Screen Technique’.

The goal is to separate the background from the shapes so that the blue background is replaced by a white background and the shapes still visible on them.

To do this in MATLAB, first the input image ‘shapes.raw’ is read using the function ‘read_raw.m’ in the m-file ‘main.m’. The image is then separated into three channels – Red, Green and Blue to analyse their individual histograms. This is done in order to identify and distinguish the range of pixels of individual channels R,G,B that lie in the foreground and background respectively.

After finding the relevant **thresholds,** a condition is set that those pixels belonging to the background be turned white and let those belonging to the foreground remain the same. (Thresholds- R<31, G<51, B>213). This essentially removes the Blue Background from the image along with any other variations in the background, while retaining the ‘shapes’ in the foreground.

III. Results

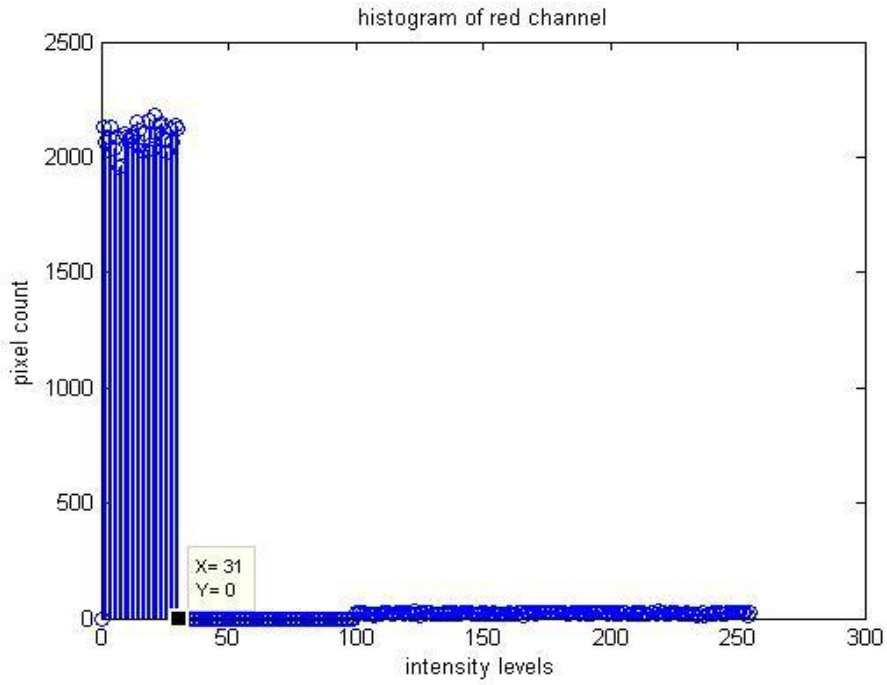

Figure 1- histogram of red channel

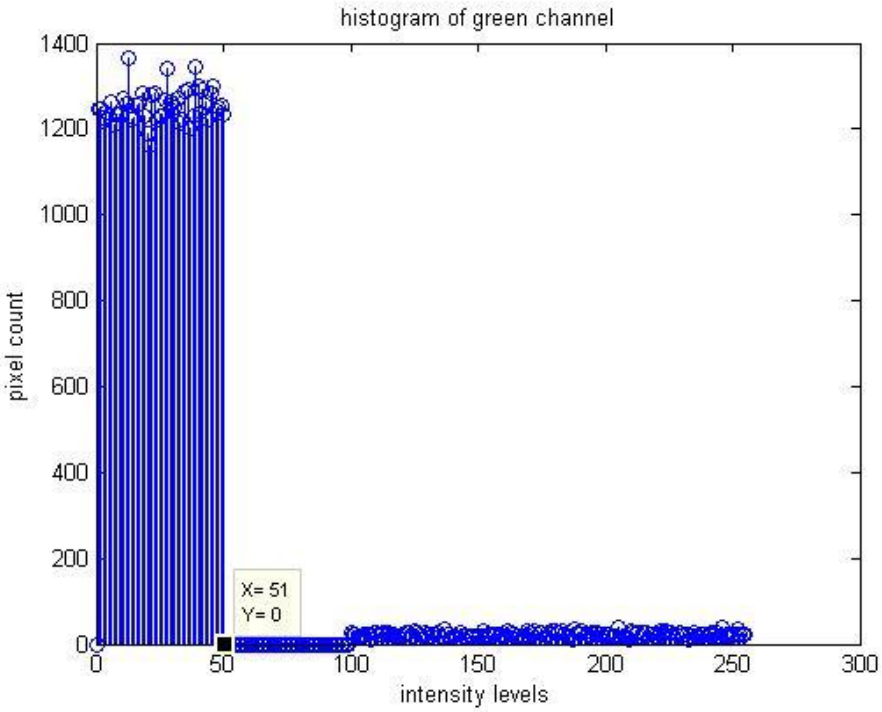

Figure 2- histogram of green channel

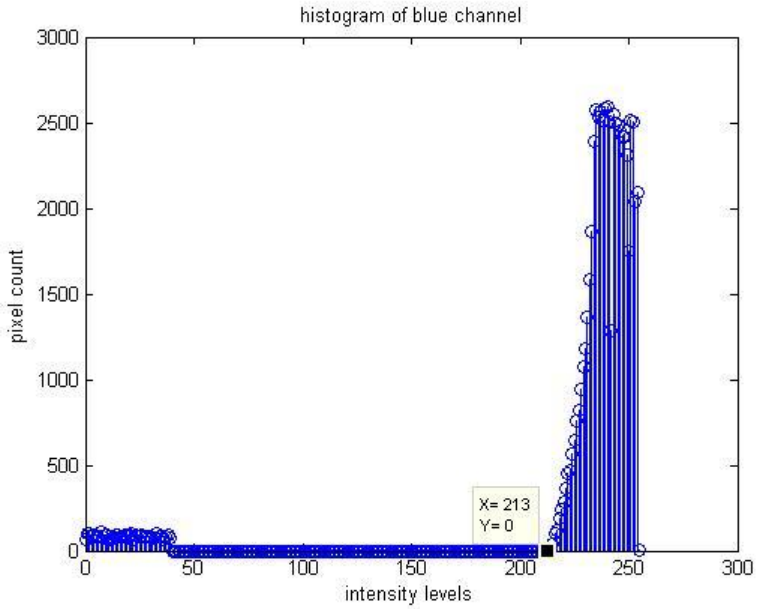

Figure 3- histogram of blue channel

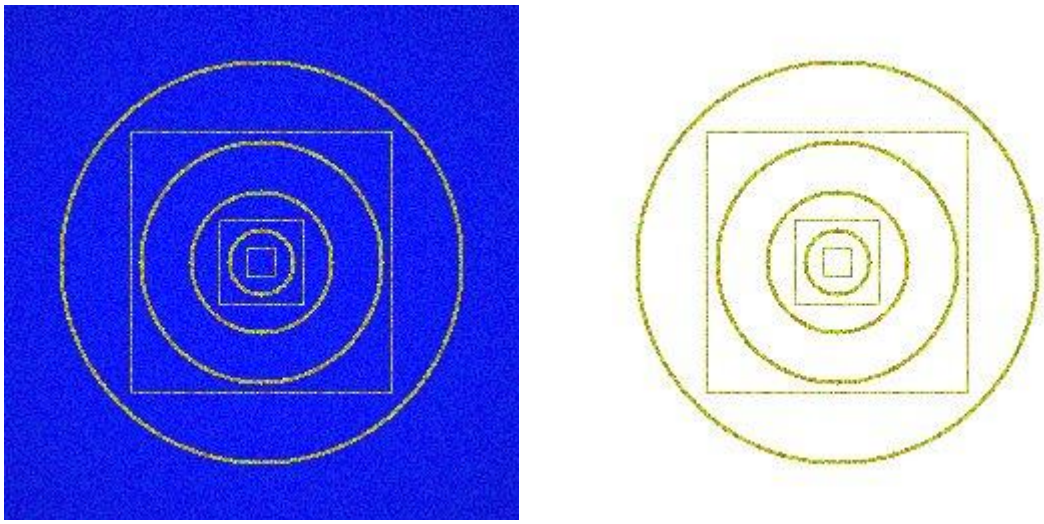

Figure 4 – Input image ‘Shapes.raw’ (left) and Output image ‘Output_shapes.raw’ (right).

IV. Discussion and Conclusion

In Figure 1, 2 and 3 – the histograms of Red, Blue and Green channels are shown. Two conclusions can be drawn from observing the histogram-

1.) Since **blue is the predominant colour in the background** , from analysing the blue channel histogram it is derived that the intensity levels with higher frequency are in the background than compared to lower frequency ones. This leads to observing that the transition point where the Blue in the background can be separated from those in the foreground is the pixel intensity 213, which is set at the threshold. This means that pixels belonging to intensity level **213 or greater** are all in the **background** whereas the ones belonging to intensity levels **below 213** are in the foreground. Therefore, pixels in the background are accessed and set to white (255).

2.) By similar analogy, **Threshold T =31** for **Red** channel (**T<31=background** and hence will be set to white (255), and **T=52** for **Green** channel (**T<51=background** and hence will be set to white).

The other pixels are all left un-manipulated and hence will retain their intensity levels in the background.

In summary,

R: T<31 = 255
G: T<52= 255
B: T>213=255

On combining the final channel outputs after manipulating the pixels using information from histogram analysis, 'Output_shapes.raw' file is created to store the output image. The final result can be observed in Figure 4. The image on the left is the input provided and the one on the right is the output derived.

1b.) IDENTIFYING GEOMETRICAL SHAPES

I. Motivation

The underlying concept of this problem is **Object Recognition**. The area itself is a vast one in the domain of Image Processing. The idea is to be able to identify the desired object or specific object of observation from a captured digital image. The issue to be handled is usually **Object extraction** for which recognising the specific object from a given image is a pre-requisite. Object Recognition is usually preceded by separation of certain features and preprocessing– in this case, the separation of background and foreground.

II. Approach and Procedures

In the given problem, the output of (1a.) 'Output_shapes.raw' is provided as an input argument to the function 'Identify_shape.raw'.

Since 'thresholding' has been done previously, it can be taken for granted that object recognition can be performed merely from extracting one channel's information from the 24-bit RGB image.

The approach followed is identification of the center of the image, traversing horizontally on X-axis to hit non-white or background pixels, identifying their shapes by traversing vertically from that point and finally, counting the objects encountered.

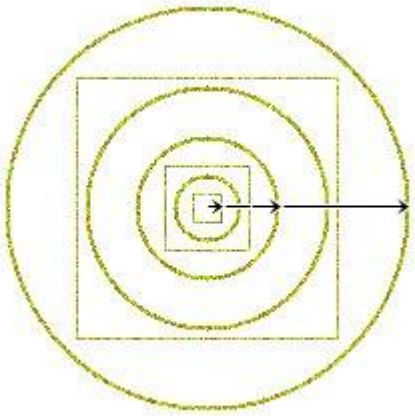

Figure 5- traversing the horizontal X-axis to find non-white pixels

Further explaining this process, figure 5 explains the traversal on horizontal X-axis to encounter the objects of interest.

To identify whether the object is a circle or a square, two conditions are made use of – traverse the vertical path from encountered object up to the distance of its radius (for example, **r1** and **r2)** in **Figure 6 –**

1) If at the end of the traversed path, the pixel is coloured, it is a **square.**
2) Else, it is a **circle.**

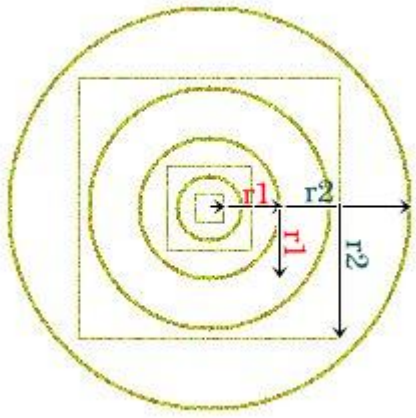

Figure 6 – identifying a circle or a square

III. Results

After running the Algorithm, the following results are shown in Matlab.

→ square.
Side Length= 7

→ circle.
radius= 15

→ square.
Side Length= 21

→ circle.
radius= 34

→ circle.
radius=59

→ square.
Side Length= 65

→ circle.
radius= 100

→ no.of circles= 4

→ no.of squares= 3

→ total objects= 7

IV. Discussion and Conclusion

In day-to-day activities that use Image Processing Techniques, **object recognition** is not so simple as the one in given problem.

Usually the flow of the procedure is Pre-Processing→Feature Extraction→Object Recognition.

The procedures for Object Recognition range from Edge Detection to Gradient Matching to Clustering etc.

The algorithm or the procedure discussed in **II.** is just one of the simpler approaches and will suffice for the complexity of the given problem.

1c.) EXTRACTING GEOMETRICAL OBJECTS

I. Motivation

The concept behind this problem is **Image Extraction or Object Extraction.** Sometimes it may be desired to perform some image processing operations, or even analysis, on a singled out object in an image.

For this, the target object first needs to be extracted from the given input image. However, in this problem the simple idea is to manipulate the pixels based on the size of the circle/ square the user decides to extract and display the extracted object.

II. Approach and Procedure

A MATLAB function 'extract_shape.m' is written and invoked in 'main.m' for this purpose.

It is assumed that the images are all centered and hence the center coordinates of the whole image (x,y)=(129,129) are noted down after viewing the input RAW file on an image viewing software such as ImageJ.

The approach taken is as follows:

- For extraction of a circle, the equation of a circle is used.
  More specifically,

$$(x^{1-} x)^2 + (y^{1-} y)^2 = r^2$$

  where 'x1','x2' are the coordinates away from the centre ,on the boundary of the circle. The pixel coordinates located at (x1,y1) not satisfying the above condition are turned to white (pixel value 255 is assigned to those) i.e

  if

$$(x^{1-} x)^2 + (y^{1-} y)^2 < r^2$$
$$(x^{1-} x)^2 + (y^{1-} y)^2 > r^2$$

  then (x1,y1) =255.

- For extraction of a square, the four corner coordinates of the square are derived from the help of the side length gotten from the previous problem.

  The pixels lying outside the edges formed by the 4 corner coordinates are assigned a value of 255 (background) and hence the square with the target side length is extracted.

III. Results

Figure 7 – Second largest square (left) and Second largest circle (right)

As seen above in the Figure 7, (left) - the second largest square (distance =21) from center of the image, and (right) – the second largest circle (radius =59) are extracted.

IV. Discussion and Conclusion

For the given problem, this approach works just fine. The radius entered can be varied while executing the 'main' M-file and the algorithm may be verified for different radii/distances from the center.

----------End of Problem 1 ----------

2. IMAGE ENHANCEMENT

2a. HISTOGRAM EQUALIZATION FOR GRAYSCALE IMAGES

I. Motivation

A histogram is a plot of pixels v/s intensity levels of an image. Using this, the distribution of pixels can be observed. histogram equalization is a technique used to enhance the contrast of an image.

Contrast can be defined vaguely as the arrangement of the lighter and the darker pixels in an 8-bit image. An image with a better contrast is one which has its pixels spread out uniformly over the entire range of specified gray level values (0 to 255 for an 8 bit image).

Contrast enhancement is relevant in many areas such as medical imaging, aerial imaging etc. Enhancing the contrast allows for much better analysis of imaging data collected, for example photos captured in space (of the planet / other celestial body surfaces) are often underexposed to light. In such scenarios, Contrast Enhancement becomes important.

II. Approach

To enhance a contrast, several methods exist. Two of them are – 'Cumulative Histogram Equalisation' and 'Full Scale Linear Contrast Stretching'.

In **Full-Scale Linear Stretching**, as the name suggests, the range of existing pixel intensities in an image are stretched over a specifically defined range of pixel intensities as shown below-

$$[F_{min}, F_{max}] \text{ " } [G_{min}, G_{max}]$$

where {fmin,fmax} are the existing intensity levels and {gmin,gmax} are the desired intensity levels to stretch the image.

The term 'linear' suggests that the pixel intensities are only linearly scaled. There is no provision for non-linear scaling in this method and the **transfer function** that is obtained is a linear function as well.

The scaling is done as

$$P_{out} = (P_{in} - c)\left(\frac{b - a}{d - c}\right) + a$$

Where the Pout stands for output intensity value

Pin stands for input intensity value

a,b specify the lower and the upper limits of the intensity values for stretching.

c,d specify the current maximum and minimum intensity values.

In the given HW problem, (a,b)=(0,255)

(c,d)= (0,255)

In **Cumulative Histogram Equalization,** the idea is to distribute the cumulative probability densities of the pixels as uniformly and equally as possible over all the intensity levels. The emphasis is on the number of pixels in each intensity level rather than the upper and lower bounds when compared to full scale linear scaling method, therefore leading the utilisation of the full range of available intensity levels.

The transfer function can be obtained by

$$s_k = T(r_k) = \left(\frac{L - 1}{M.N}\right)\sum_{j=0}^{k} n_j$$

Where s=output pixel intensity value

r= input pixel intensity value

L= intensity levels

M,N = area of image (M=height,N=width)

n= no.of pixels in each intensity level ‘k’ from [0,L-1]

Steps followed are-

- The Probability Density Functions of each intensity level are cumulated to obtain the Cumulative Density Function (CDF) over each intensity level.
- Make a mapping table of the original image CDF obtained to match to the CDF of the uniform distribution.
- Map the original intensity levels to the target equalized ones using the transfer function above.

III. RESULTS

Figure 8. Original input images 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

Figure 9. Contrast stretched images for 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

Figure 10. Histogram equalised images for 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

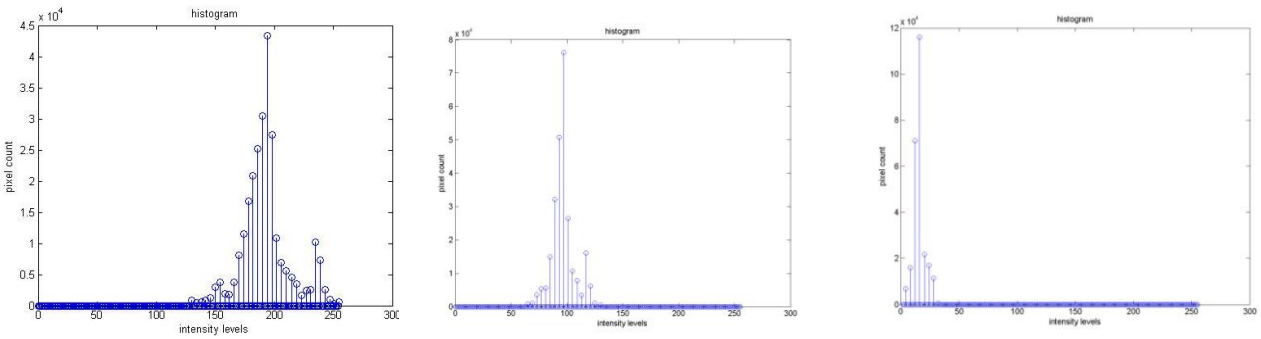

Figure 11. Original Input histograms for 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

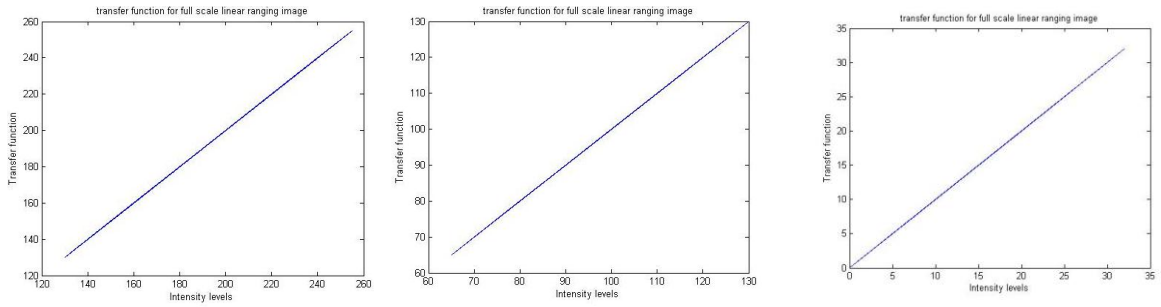

Figure 12. Transfer function of Full Scale Linear Ranging (Contrast Stretching) for 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

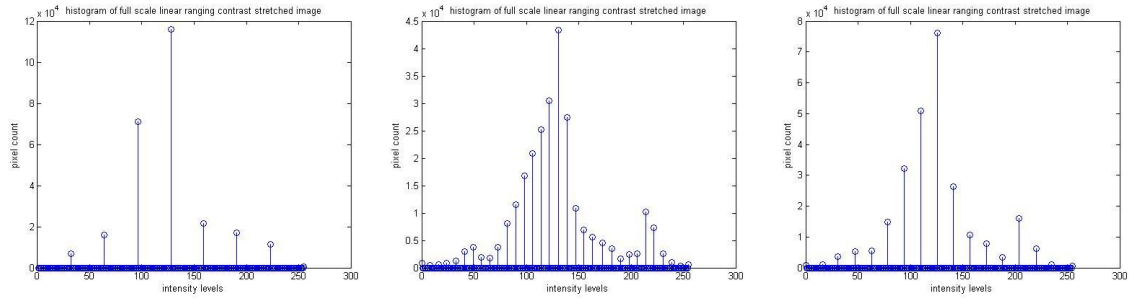

Figure 13. Histogram of contrast stretching for 1)Barbara_Bright 2)Barbara_mid 3)Barbara_dark

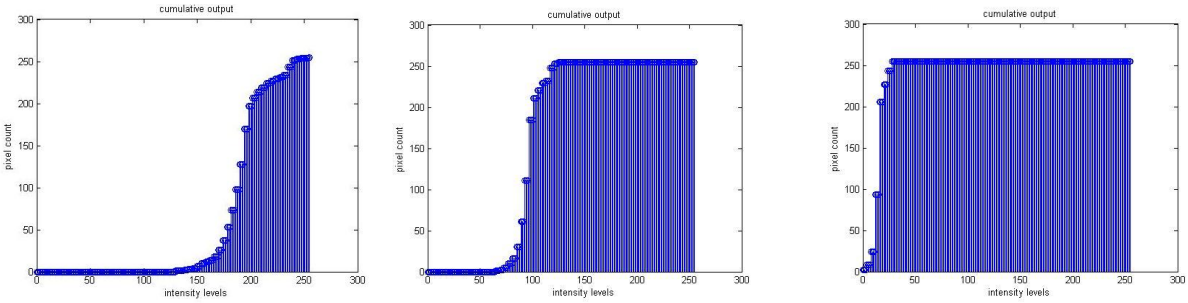

Figure 14. CDF transfer function for 1)Barbara_bright 2)Barbara_mid 3)Barbara_dark

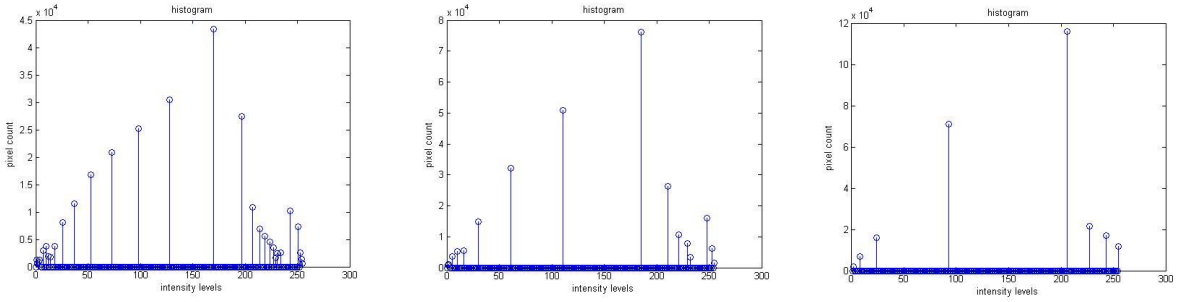

Figure 15. equalized histograms for 1) Barbara_bright 2)Barbara_mid 3)Barbara_dark

IV. Discussion and Conclusion

It can be seen clearly from Figs 8, 9 and 10 that 'Histogram Equalization' yields better contrast enhancement than 'Contrast Stretching' by Full Scale Linear Ranging (FSLR) in the literal sense. However, the images look artificial in the sense that they look almost inverted on the gray scale.

Since the scaling is linear in Full Scale Linear Ranging histogram, the enhancement is less harsh compared to Histogram equalization and this effect can be clearly observed when figs 9 and 10 are compared. To expand more, what 'harsh' means is that the background pixels are also very highly enhanced leading to very inconspicuous textural exposure. The texture of all the background objects can be clearly seen.

It can be noticed that while the output of FSLR has a pleasing effect on the eye, there is quite a little distortion in the form of contouring that can be observed, especially when applied to dark images.

Technically speaking, Histogram Equalisation results in the utilisation of Full Dynamic Range however in this case, the enhanced images look a little too bright – because histogram equalisation results in enhanced background as well.

It can be concluded that FSLR works well when sort of a smooth contrast enhancement is required where as the histogram equalisation works well for cases that absolutely require the full dynamic range exposure.

2b. HISTOGRAM EQUALIZATION FOR COLOUR IMAGES

I. Motivation

As discussed in the previous problem, **Histogram Equalization** works well in case of images that require the utilisation of Full Dynamic Range either in order to be visually pleasing to the eye or for the purpose of exposing the texture/features of the poorly lit background.

In the latter case, the applications of this is particularly useful in the field of Medical Imaging or Thermal imaging when contours are very much needed (and not to be avoided for) observational and analytical purposes, example, scans.

However, in spite of equalisation in most cases the histogram is not truly flat – that is the pixels are evenly spread out but not necessarily equally across all the intensity levels. In such cases, a method called **Bucket Filling** method can be followed to force equal number of pixels to occupy each intensity level.

II. APPROACH AND PROCEDURE

The approach for **Equalized Cumulative Histogram** remains the same as in the previous problem, but is repeated for all the three channels of the 24-bit color image 'ocean_contrast.raw'.

The steps are as follows:-

1) Calculate the probability of occurrence of each pixel in an intensity level (normalization)

$$p^x(i) = \frac{n_i}{n},\ 0 \# i < L$$

Where i=the current intensity level, L= maximum intensity level, (n i) = pixels belonging to the i -th intensity level.

2) Based on the PDF (probability distribution function) the CDF (Cumulative distribution function) is obtained.

$$cdf^x(i) = \sum_{j=0}^{i} p^x(j)$$

where CDF= denotes sum of PDF across all intensity levels.

3) Make a mapping table of the original image CDF obtained to match to the CDF of the uniform distribution.

This essentially gives the transformation function

$$y = T(x) = cdf(x)$$

4.) Map the original intensity levels to the target equalized ones using the transfer function above. The inverse function for this is as below:

$$y' = y.(\max\{x\} - \min\{x\}) + \min\{x\}$$

The approach taken for the **Equalised histogram** is called the 'Bucket Filling' method –

Comment: Equalised histogram differs from cumulative histogram because of the digital implementation and the imperative need to store equal number of pixels in each bin in order to produce a flat, uniform histogram.

1) Initialise the bin (or the intensity level value) to be zero.

Bins=0;

2) Construct a matrix with the same dimension as the input image to replace pixels according to Bucket filling condition, i. e

N^2/256 pixels per each intensity level (N=height/width of input image) –

Output_Matrix=zeros(image_height,image_width);

In the given problem, the product of the image dimensions is not a perfect square hence the $N^2/256$ is approximated to the nearest integer (in this case – (500*332)/256 ~ 649)

2) A counter is put in place to scan through each intensity level for each pixel intensity value. If the condition is satisfied (*i.e. if each intensity level has <=649 pixels)* those pixels are put in the respective bin of the output image *Output_Matrix* else the *Bins* is incremented so that the pixels go to the next intensity level.

3) Since the problem is a colour image, steps 1-3 are repeated for all the R,G,B channels and the output matrices of all these channels are finally convolved to give the resultant histogram equalized image.

III. RESULTS

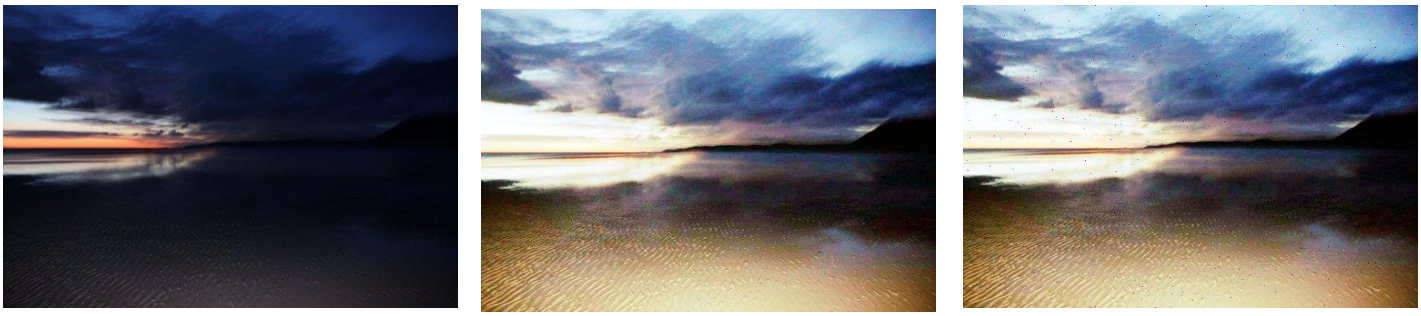

Figure 16. 1) Original 'Ocean_contrast' image 2) Equalised Cumulative Histogram output 3)Equalised histogram output

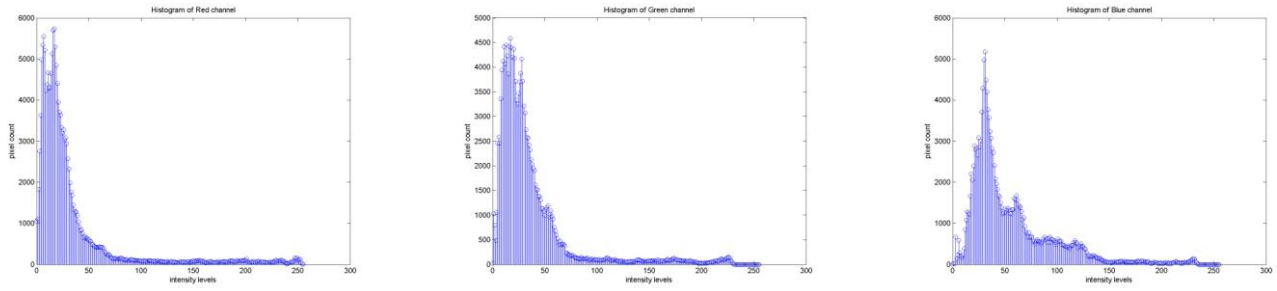

Figure 17. Histograms of R, G,B channels for original image ' Ocean_contrast'

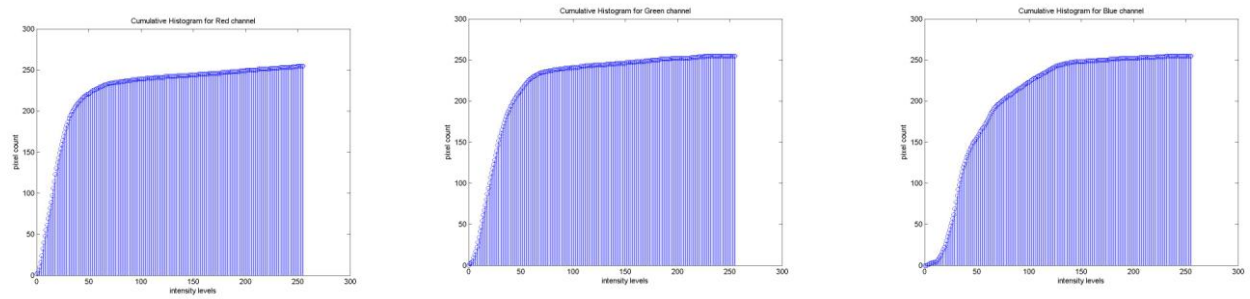

Figure 18. Cumulative Equalised histograms for R,G,B channels for output image '2b_eqCumHist' [2.)in fig.16]

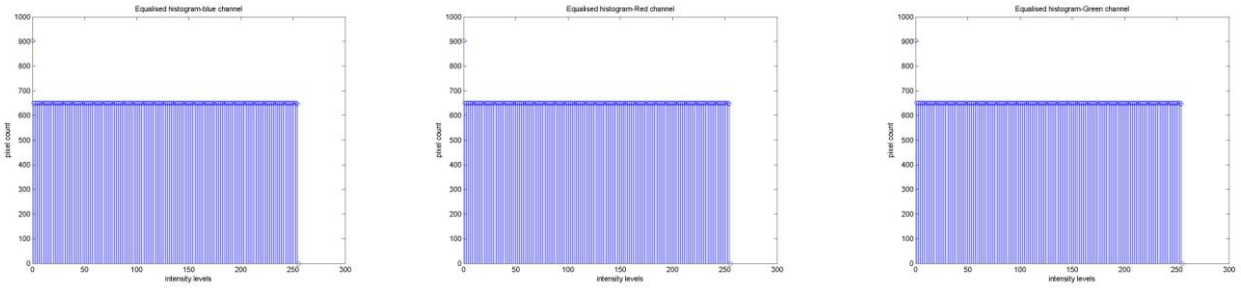

Figure 19. Uniform, flat, equalised histograms for R,G,B channels for output_image 'equalised_hist' [3.) in figure 16]

IV. Discussion and Conclusion

As can be observed from figure 18, histogram equalization has enhanced the contrast of the landscape picture 'ocean_contrast.raw' significantly and gives a visually pleasing image as compared to its original counterpart.

Fig.19 shows a nice flat histogram for all the channels of the image after digital implementation of histogram equalization – bucket filling method.

However, it is also noticed that the orange parts of the original image are almost white in both the output images – the one with the cumulative equalised histogram and the digital implementation of it.

This is due the fact that Histogram Equalization results in enhancing the contrast across all the intensity levels. Therefore, while the darker backgrounds are enhanced, the lighter parts of the image are also enhanced – contrast wise.

Ideally, both the methods would give the same output and not much difference would be visible to the naked eye - the extent of the contrast stretch would be same across all the intensity levels. In this case however, the image dimensions do not form a perfect square when multiplied and as a result, there are some spots observed on the digital implementation output of histogram equalized image (due to slightly uneven pixel distribution).

In conclusion, it is important to take note that if the gray scale values were centralised in an image, then the result is that Bucket Filling method would cause a slight amount of distortion due to redistribution of pixels/scattering of pixels. Therefore, this method would not be appropriate for all images. In the given homework problem, there is not much of a difference in the colour distribution across the image hence the equalization tends to be uniform and aesthetically pleasing.

2c. HISTOGRAM TRANSFORM

I. Motivation

Sometimes, it is not necessary for images to have all intensities to be enhanced. Of course, it depends on the context of usage but to speak in general terms, an image where centralization of gray levels is required and less focus is to be given to details in the background, smoothing is done.

Gaussian filter or a Gaussian distribution function is one such example for implementation of a smoothed image. It is a low pass filter which filters out higher frequencies and hence gives a smoothed image.

According to lectures in class, Gaussian filter used before implementing 2-D Laplacian transform for edge detection (LoG) improves the performance of the latter due to desensitization to noise levels . In such scenarios, transforming a given image to a Gaussian distributed one is of immense importance.

II. Approach and Procedure

A Gaussian Probability Density Function is given by the equation

$$p(x/n,v) = \frac{1}{\sqrt{2P\,v}}\exp\left(\frac{-(x-n)^2}{2v^2}\right)$$

Where n =mean of the Gaussian distribution

And v =standard deviation

The mean defines the average value of the distributed pixels and Standard Deviation defines the range of the pixels from the mean that are taken into account while distribution.

The steps are –

1) Plot the histogram of the input image and obtain its CDF (CDF1)
2) Plot a Gaussian distribution function according to the above equation
3) Renormalise the PDF so that $\int_{x=0}^{255} p(x/n,v) = 1$
   This is because the Gausian distribution should be truncated between the intensity levels of the gray image [0,255] and hence the probabilities would not add up to 1.

3.) Plot the CDF2 from 2) $cdf^x(i) = \int_{j=0}^{i} p^x(j)$

where j=[0,L-1] (L=255 in the given problem)

4.) CDF2 gives the transfer function from input to output target histogram intensity values, which are used to map the CDF1 values to nearest corresponding CDF2 values. Therefore, inverse mapping this gives the redistributed enhanced image by histogram matching. The example below explains this in detail

Intensity level	CDF1 values
0	0.19
1	0.23
2	0.44
3	0.65
4	0.76
5	0.90
6	0.92
7	1

CDF2 values	**Intensity level**
0	0
0	1
0.20	2
0.35	3
0.65	4
0.70	5
0.75	6
1	7

Therefore, CDF1[0.19, 0.23] would be replaced by CDF2[0.20]
CDF1[0.44] would be replaced by CDF2[0.35]
CDF1[0.90, 0.92,1] would be replaced by CDF2[1]

This gives the original image input histogram a Gaussian fit and the final image is considerably enhanced and smoothed.

III. Results

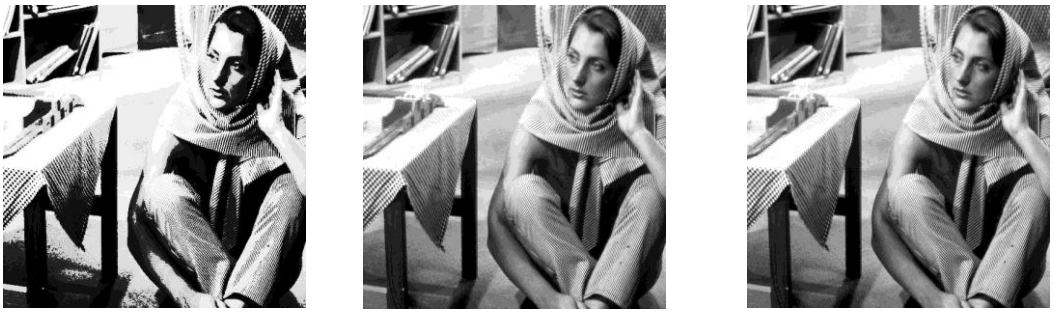

Figure 20. [L-R] 1.) original 'barbara_contrast.raw' 2.) output (mean=70,SD=20) 3.) output (mean=120,SD=50)

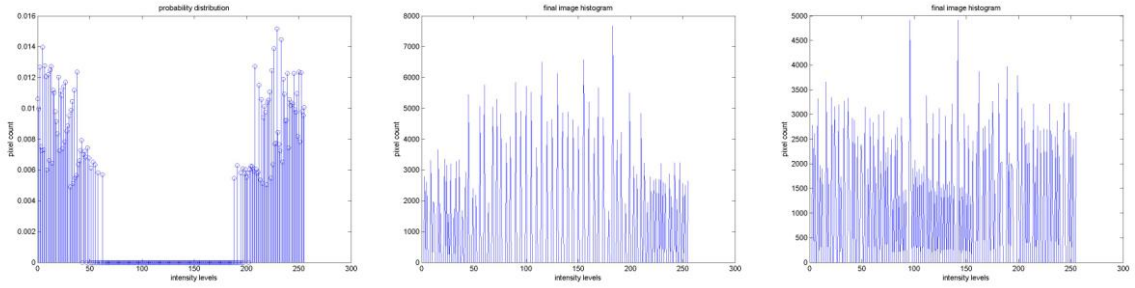

Figure 21. [L-R] 1)input histogram 2)output(70,20) histogram 3)output (120,50) histogram

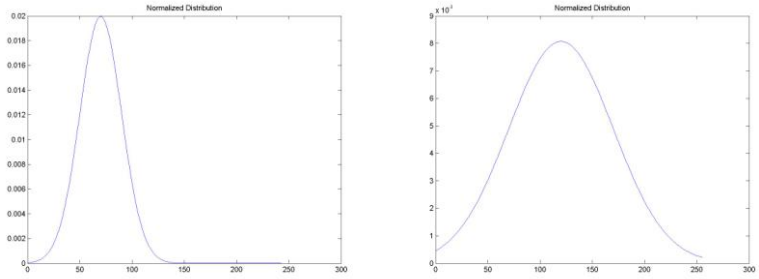

Fig22. [L-R] 1)normalised Gaussian PDF (mean=70,SD=20) 2)normalised gaussian PDF (mean=70,SD=20)

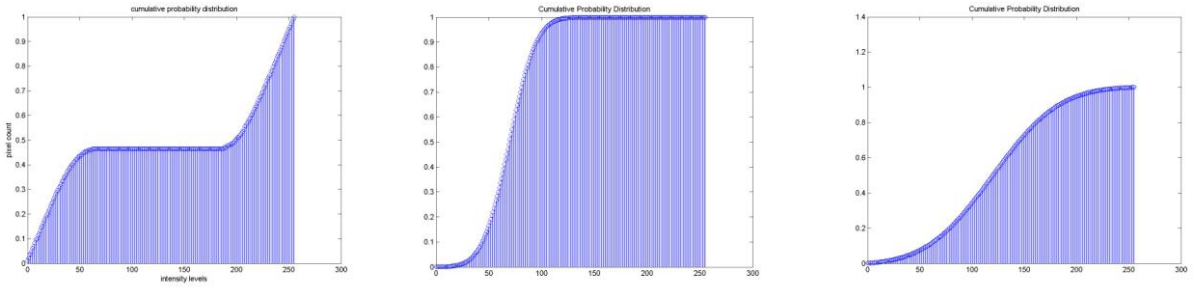

Fig23. [L-R] CDFs of 1)input image 2) output image(mean=70,sd=20) 3)output image(mean=120,50)

IV. Questions and Answers

1) Comparison of two output images after histogram equalisation –
   While not much difference can be noticed with the naked eye in the output images 1 (mean=70,SD=20) and 2 (mean=120,50) , the equalised histograms offer a much deeper perspective.

   The differences in the images itself are minute but on careful observance, it is obvious that the output 2 is slightly better enhanced than output1 – the background details are a tad bit clearer in output2 than in output1.

2) Changing of parameters Mean and SD.
   On setting mean=70, SD = 20, the pixels are centered around the intensity value 70 but the SD=20 signifies the range of intensity values to be considered is only [50,90] This results in a flatter histogram compared to output2

   On setting mean=120, SD=50 the pixels are centered around the average intensity value 120 (which is a higher value compared to mean=70) and the SD=50 is taking [70,170] intensity values into consideration. Therefore, a higher range of intensity values are subject to enhancement, resulting in a slightly brighter image with more details.

V. Conclusion

Gaussian Histogram equalisation is required for images which need to have a smooth distribution and less noise (especially that in the background).

----------End of problem 2----------

3. NOISE REMOVAL

3a. Mix noise removal in colour image

I. Motivation

Noise refers to the presence of undesirable signals in a captured image. This could occur because of various reasons. However, the most common ones are due to inefficient sensors of the capturing device (intrinsic /internal errors) or the mixing of noise due to presence of noise in the signals being captured (extrinsic/environmental noise due to external sources).

In the literal sense, since noise stands for unwanted signals, they should be removed before utilising them for the relevant purpose.

The challenge of this area of image denoising is the removal of the noise as well as preserving the features of the images such as edges. It is always a trade-off between those two in order to estimate the original image.

Denoising is an integral part in case of Image Processing for areas such as image restoration, classification and segmentation.

II. Approach and Procedure

The given problem has a 'Peppers.raw' image that has mixed noise – this usually is a combination of uniform/Gaussian noise and impulse noise (salt and pepper)

Step 1) The histograms of individual channels are plotted to check which channel contains what noise.

Step 2)Various filters are applied individually to each of the channels – Median filter (non-linear), Mean/Average filter and Weighted Average Filter (linear)

Step 3) Two combinations of cascaded filters are applied.

Step 4) Observations are made.

III. Results

Figure 24. Given original 'Peppers' image with mixed noise.

Figure 25. 3x3 Median Filtered Peppers image (left) and 5x5 median filtered (right)

Figure 26. 3x3 Average filtered Peppers image (left) and 5x5 Average filtered image (right)

Figure 27. 3x3 Weighted average filtered Peppers image (weights [1 2 1; 2 4 2; 1 2 1])

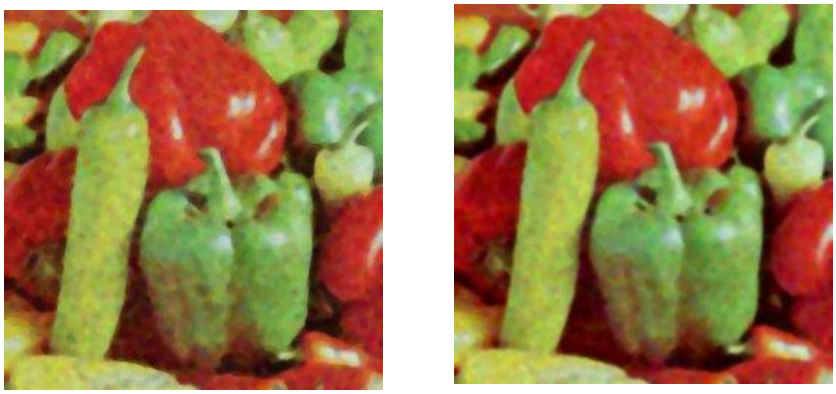

Figure 28. (Left) Cascaded filtered output of 3x3 Weighted Average filter, 5x5 median filter, 3x3 median filter

(Right) Cascaded filtered output of 3x3 median filter, 5x5 median filter, 3x3 weighted average filter

IV. Answers to questions (Discussion)

1)

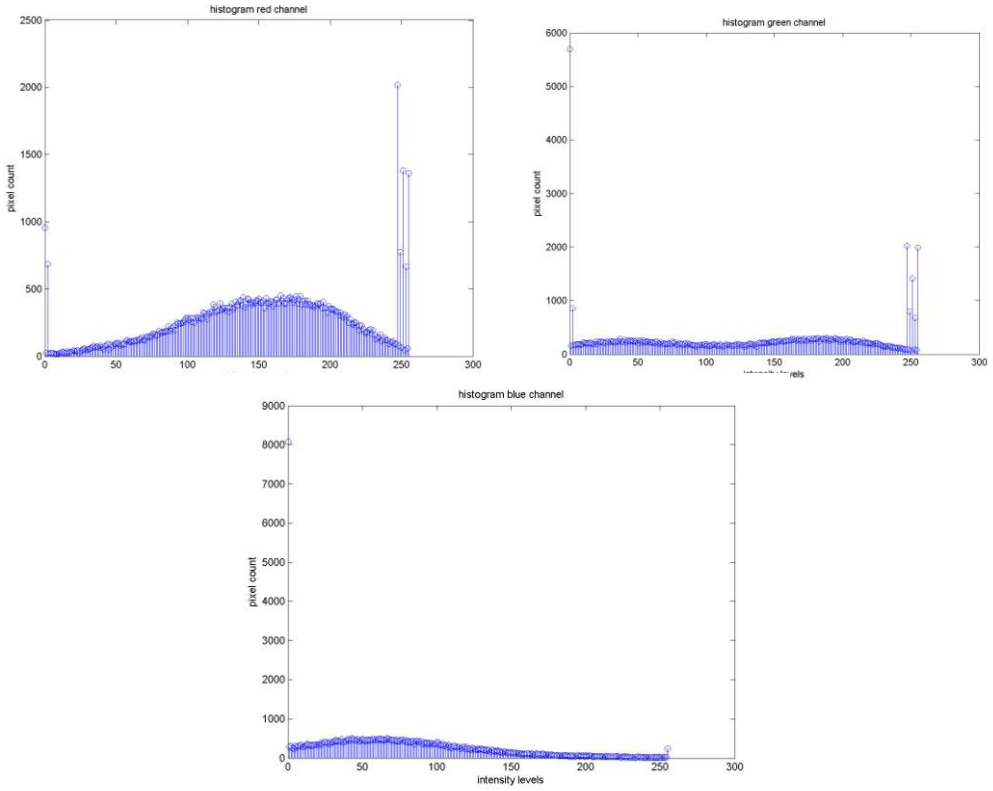

Figure29. [L-R clockwise] Top-Left Histogram of Red channel **Top Right** Histogram of Green channel **Bottom** Histogram of Blue channel

a) Observation of the R,G,B channels of the picture show that across all the channels, impulse noise (Salt noise in the darker regions of all the channels and Pepper noise in the brighter regions) is the most prominent. However, the degree of the salt noise is lesser compared to that of the pepper noise. Blue channel has almost negligible salt noise. Apart from impulse noise, there seems to be also the presence of Gaussian noise looking at the distribution of pixels along the Gaussian curve. One cannot be too sure about it without looking at the histograms of all the channels of the original, uncorrupted image.

b) Yes. Filtering should be performed separately for both the noise types as different filters give different results. The presence of noise is neither equal nor same in all the channels and this affects the choice of filters.

c) To remove the Gaussian noise and the salt & pepper noise, three types of filters are used in the approach used.

Two linear filters – Average and Weighted Average, and one non-linear filter – Median filter are experimented with.

From the results posted above, we can see that median filter of 5x5 has denoised the image best compared to other filters.

d) From observing the filtered output images **25 and 27**, it is evident that the best results was given by 5x5 median filter (removal of salt and pepper noise) and 3x3 Weighted Average filter (not in terms of noise removal but in terms of edge preservation). In order to combine the best of both filters, it was thought of to try two different cascaded filter arrangements, the outputs of which are shown in **Figure 28.**

A-- 3x3 weighted Average, 5x5 Median and 3x3 median filtering
B-- 3x3 median , 5x5 median and 3x3 weighted average.

In both cases, A and B it is noticed that, along with combining the best of median and weighted average filters' capabilities, the result also combined the disadvantages of the used filters.

In A, the result is better than weighted average alone but the image has lost edge details considerably and also its contrast due to the use of weighted Average first (which replaces the intensity values of the neighbourhood pixels by their average) Clearly, having an impulse noise in the neighbourhood affects the brightness and colour preservation adversely.

In B, the colour situation is a little better than in A and seems to be due to the fact that weighted average is used after the median filters. Median filters are able to preserve the colour better than average filters since the median intensity value is the one that is already present in the image and hence leads to better outlier removal.

To conclude, the cascading of filters result in better noise removal but there is loss of details, colour and contrast.

e) Window sizes – Mean filters or Averaging filters are known for reducing additive white Gaussian noise – their linear characteristic fits the situation for denoising. However, as the **figure 26** shows – while averaging filter removes Gaussian noise to an extent, it often leads to loss of details and this loss **inreases** with an increase in window size.

In case of median filters however, the result seems to improving in terms of detail preservation as well as the noise removal as we move from a 3x3 to 5x5 window size. However, the edge details are lost to an extent.

2) The shortcomings as previously discussed, are the facts that
- One filter cannot deal with mixed noise effectively.
- In using a combination of filters (cascaded filters) the disadvantages of the constituent filters used individually are also seen in the result.
- It is a trade off between detail preservation and noise removal.

Suggestions to improve the performance –
The filter combinations could probably use multiple iterations to observe the resultant noise levels.

One of the references suggests the use of conservative smoothing technique that is more effective when the trade off between edge preservation and impulse noise removal is to be considered.

Another possibility could be the use of adaptive filters which, as the name suggests, lead to optimisation of the filtering action based on error correction.

V. Conclusion

For the given problem, in summary,

Median filter (5x5) seems to have reduced the impulse noise most effectively while preserving the colour details while the Weighted average filter seems to have removed the Gaussian noise to a certain extent while preserving the sharpness to the maximum when compared to the other filters used.

----------End of problem 3----------

REFERENCES

1) Digital Image Processing, Second Edition, Rafael C. Gonzalez and Richard E.Woods.
2) http://entertainment.howstuffworks.com/blue-screen2.htm
   - "How Blue Screen Works"
3) http://homeworks.inf.ed.ac.uk/rbf/HIPR2/stretch.htm
   - "Point operations", Contrast Stretching.
4) http://eeweb.poly.edu/~yao/ee3414/image_filtering.pdf
   - "Spatial filtering"