HOMEWORK ASSIGNMENT 1

Name : Akshata Bhat
USC ID : 9560895350
Email : akshatab@usc.edu
Submitted Date: 2/4/2018

INDEX

Problem 1: Basic Image Manipulation

A) COLOR SPACE CONVERSIONS

ABSTRACT AND MOTIVATION

The motivation behind this section is to get familiar with the properties of a color and a grayscale image and the operations to be performed to perform operations on each.

Every image can be represented using its width, height and number of bytes used to represent each pixel value. These values are used to perform operations on the image. Any image in the RGB color space has 3 components - Red, Green and Blue. Each of the components contribute various levels of intensities to the image based on the number of bytes used to represent each image and the amount of red, green and blue components present in the captured image. When the R, G and B components are combined using various methods, we obtain the Grayscale image. The R,G,B color space or the 'additive color model' is mainly using in representing values in monitors, TV screens and tablets..

Modern descriptor based systems used for image recognition, use grayscale images to analyse the image content and identify the image, hence it is important to understand different methods to get the best conversion. We analyze the Average, Lightness and Luminosity methods in this section:

1. The Average method takes the mean of the R, G and B values for each pixel to give the image.
2. The Luminosity method is designed to take human brightness perception into account by using a weighted sum of the R,G and B color channels for each pixel. The Green color channel gets the highest weight.
3. The lightness method takes the average of the most prominent and the least prominent color values.

CMY(K) or the 'subtractive color model' is another color space used mainly for printing applications. It is composed of Cyan, Magenta, Yellow and Black(K). Black color is represented as K which stands for 'key', because in 4 color printing, all the color plates are carefully keyed or aligned with the key of the black key plate

1. COLOR TO GRAYSCALE CONVERSION

Approach:

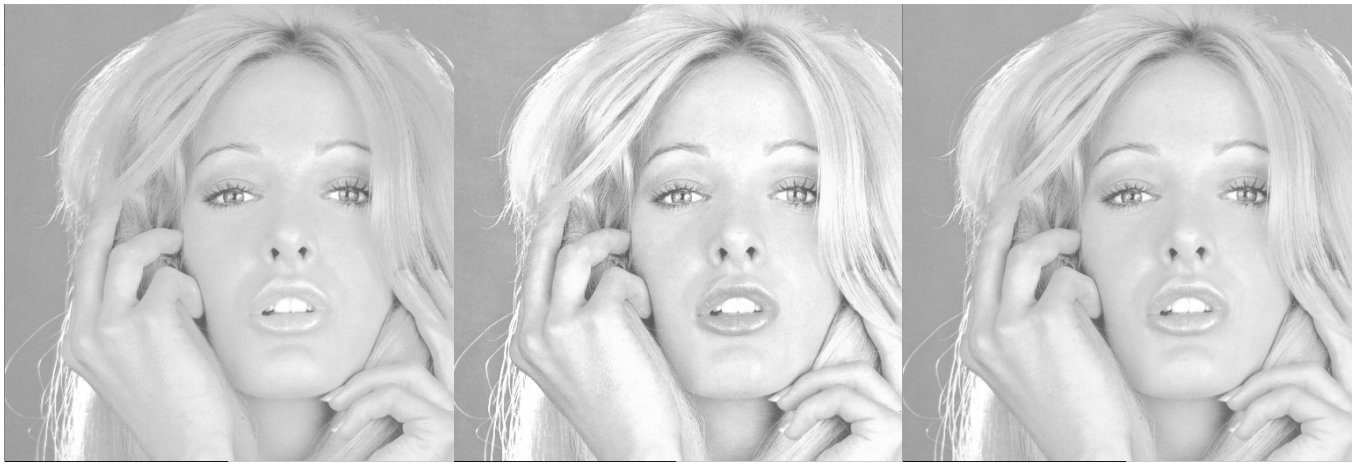
The grayscale images for the Lightness, Average and Luminosity method is obtained by applying the respective formulas to the Red, Green and Blue color channels.

Procedure:

1) Read the color image into a 1 dimensional array(D).
2) Separate the Red, Green and Blue channels into 2D arrays.

3) Calculate the corresponding Grayscale images using each of following methods :
   a) Average method :
      Grayscale = (R + G + B) / 3
   b) Luminosity method
      Grayscale = 0.21 R + 0.72 G + 0.07 B
   c) Lightness method
      Grayscale = (max(R, G, B) + min(R, G, B))/2
4) Write the resultant grayscale values for each method into a file.

Experimental Results:



Lightness method **Luminosity method** **Average method**

Discussion:
When the 3 methods of Color to Grayscale conversion - Lightness, Luminosity and Average methods are considered the Luminosity method gives the best overall output primary because of the weighted sum. This weighted sum takes into consideration that the green component contributes the most when compared to other colors and affects the visual perception of the image.

2. **CMY(K) COLOR SPACE**

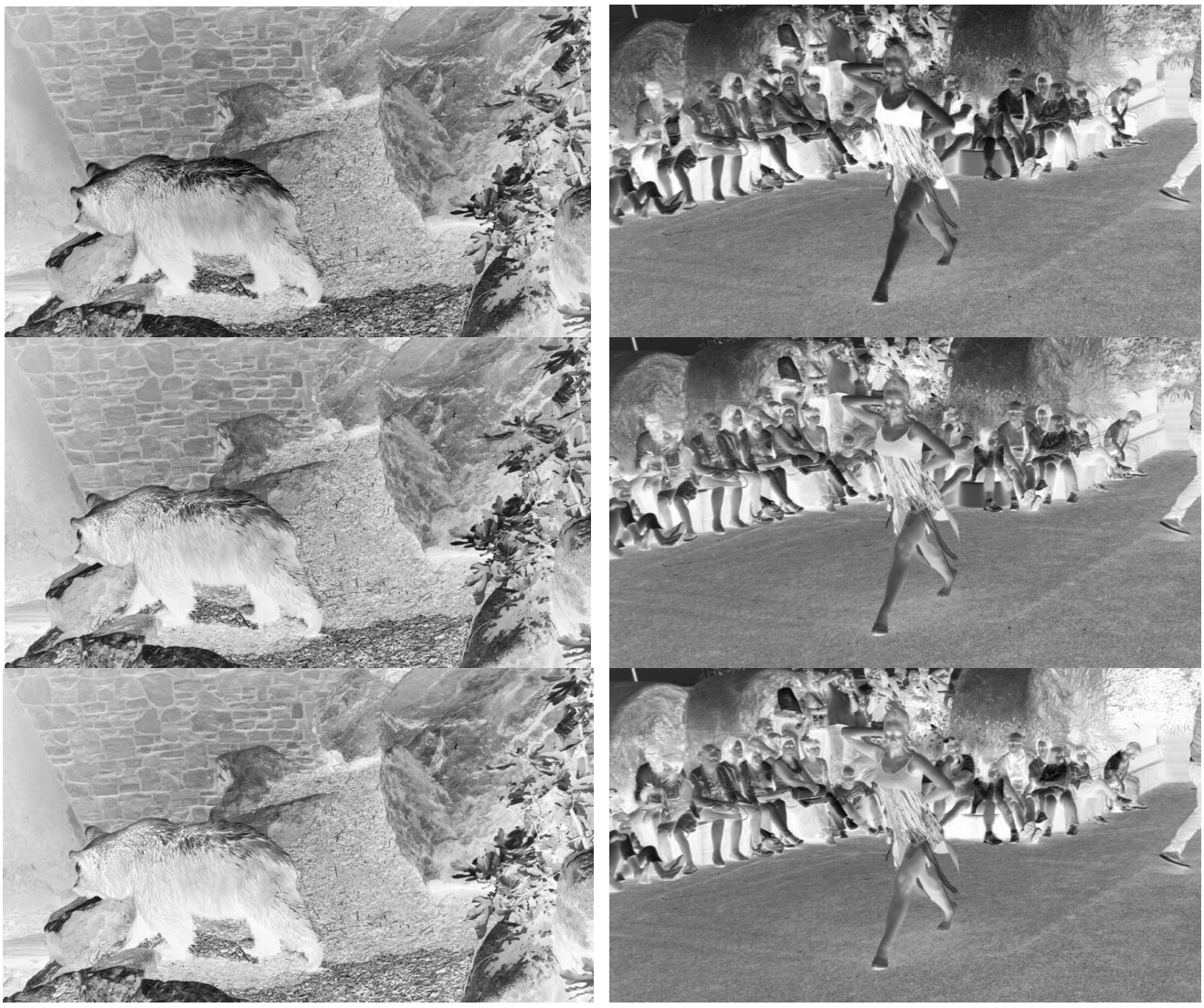
Approach:
The CMY color space representation is obtained by normalizing the R, G and B color channels and subtracting the pixel value in each channel from 1. The resultant value is then converted to initial pixel representation in the range [0, 255].

Procedure:
1. Read the color image into a 1 dimensional array(D).

2. Separate the Red, Green and Blue channels into 2D arrays.
3. Normalize the values in each channel by dividing each value by 255.
4. Subtract the normalized values to get the corresponding subtractive pair :
   Cyan = 1 - Red
   Magenta = 1 - Green
   Yellow = 1 - Blue

5. Multiply each value by 255 to get the pixel values in the range [0, 255].
6. Write the C, M, Y components into a file to obtain the grayscale images and also combine the Cyan, Magenta and Yellow channels to form the CMY representation.

Experimental Results:



C, M and Y components for Bear.raw and Dance.raw (Top, Middle, Bottom respectively)

CMY representation for Bear.raw(left) and Dance.raw(right)

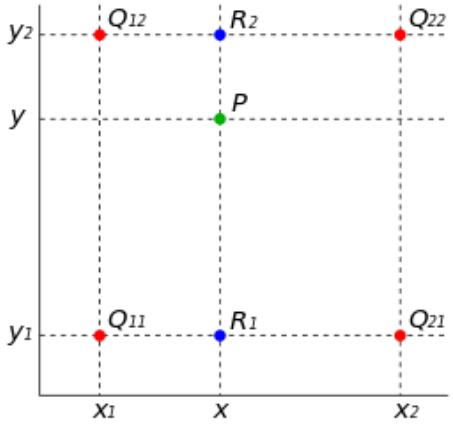
Discussion:
The RGB to CMY conversion helped understand the method of obtaining the Cyan, Magenta and Yellow components and also the concepts of normalizing the pixel intensities and taking the inverse of an intensity.

B) BILINEAR INTERPOLATION

ABSTRACT AND MOTIVATION

The main aim of interpolation is to use known data to obtain values at unknown points. Interpolation algorithms are of two kinds - Adaptive and Non Adaptive. Adaptive algorithms change based on what they are interpolating (Ex. smooth edges versus sharp edges), while non-adaptive interpolation techniques treat all pixels as the same.

Bilinear interpolation is a non-adaptive algorithm which uses 4 known pixel intensities to calculate the intensity at an unknown point. It uses linear interpolation twice. Linear interpolation has three main steps: finding the two neighboring data points, computing the weights for the two points and calculating the unknown point. [2]



If we consider a 2D matrix of values at integer grid locations (e.g., a grayscale image) as shown in the image above. To interpolate values on a 2D grid we need to use bilinear interpolation. For each point within the grid there will be 4 neighbours whose intensities are combined to calculate the intensity of the interpolated point. In the above figure, the Q values represent intensities of neighbouring pixels.[3] To combine these intensities, we perform linear interpolation in the vertical and horizontal directions: we first interpolate in the x direction (to get the value at the blue points), then in the y direction (to get the value at the green points). This is repeated for every pixel within the 2D grid. The distances of the neighbouring points contributes to the weights assigned to each intensity value while calculating the intensity of the interpolated point.

Bilinear interpolation is used in the fields of computer vision and image processing as one of the basic subsampling and image transformation technique. In texture mapping,it is referred to as bilinear filtering and is used to reproduce a reasonably realistic image.

APPROACH AND PROCEDURE

Approach:

Bilinear interpolation is combination of linear interpolation in the x and y directions. It is based on the concept that the closest neighbour to the point to be interpolated, makes the highest intensity contribution compared to the a point which is farther away.

Procedure:

1. Read the input image into a 1D array.
2. Separate the R,G and B components into 2D arrays for convenience while calculating the interpolated point.
3. For each pixel value, compute the horizontal and vertical translation factors - which are the values by how much each pixel in the original image will be translated to in the new image.
4. The translation factor can be calculated by obtaining the ratio of the original size against the new size.
5. Calculate the contributions of neighbouring pixel intensities based on the distance from the pixel to be interpolated.
6. Calculate the new interpolated pixel using the formula :
   $F(p',q') = (1- b)[(1- a)F(p,q) + aF(p +1,q)] + b[(1- a)F(p,q +1) + aF(p +1,q +1)]$
   where a, b - Contributions of the neighbouring pixels
   F(p,q), F(p+1,q), F(p,q+1), F(p+1,q+1) - Neighbouring pixel intensities
   F(p',q') - Interpolated new intensity

6. Repeat the steps 3, 4, 5 for the entire image to get the new interpolated values.

Experimental results:

Original image (512 x 512)

Resized Image (650 x 650)

Discussion:

We observe that during resizing some of the original pixel intensities are being replaced and hence there will be some loss of information and some resultant blurring too. The best way to resize is by using an adaptive algorithm where each pixel intensity and its location in the image is taken into consideration while calculating the interpolated pixel.

PROBLEM 2

HISTOGRAM EQUALIZATION

ABSTRACT AND MOTIVATION

Contrast enhancement is an important area of digital image processing which is extensively used for medical image processing and as a preprocessing step in many speech, image and video processing applications. Histogram equalization in a popularly used contrast enhancement technique. Histogram equalization generally increases the global contrast of the

images, especially when the image has been represented at a lower contrast. This contrast adjustment is achieved by spreading out the most frequent intensity values effectively. This method is especially useful when the images are either too dark or too bright. In this section we try to analyse two methods to perform histogram equalization.

<u>Method A</u> - Transfer-function-based histogram equalization method

Approach:

In this method, the given histogram is mapped to a wider and more uniform distribution of intensity values so the intensity values are spreaded over the whole range by mapping the cdf of the histogram. The processing of histogram equalization relies on the use of the cumulative probability function (cdf) which is the sum of the probabilities. The probabilities indicate how often a certain intensity value occurs in an image.

Procedure:

1. Read the image file into a 1D array
2. Separate the R, G and B channels into 1D arrays for convenience while performing operations
3. Calculate the frequency of occurrence of each intensity value in the image and store it in a 1D array
4. Calculate the probability of each intensity value:

`Probability(i) = Intensity / (Width * Height)` where i is in the range [0, 255]

5. Calculate the cumulative probability for each intensity value
6. Calculate the new intensity value :

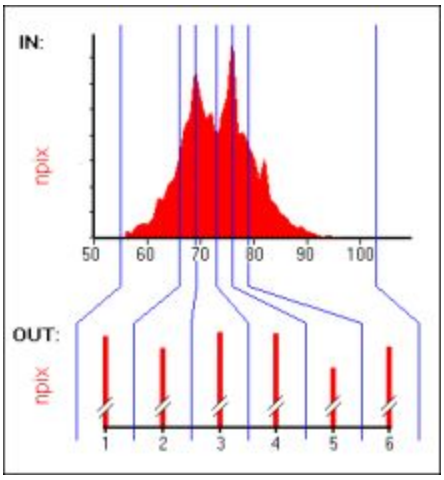
`New Intensity(i) = Cumulative probability*(255)` where 255 is the max intensity

7. Map the new intensity values to the original intensity values.
8. Repeat steps 3, 4, 5, 6 and 7 for each color channel.
9. Combine all the three channels and write into an .raw file.

<u>Method B:</u> Cumulative-probability-based histogram equalization method

Approach:

The second method mentioned in this section is also called the bucket filling method where we ensure each intensity value has the same number of pixels and hence when the number of pixels in each intensity value is summed up, the cumulative value is equal to the dimension of the image.



[6]

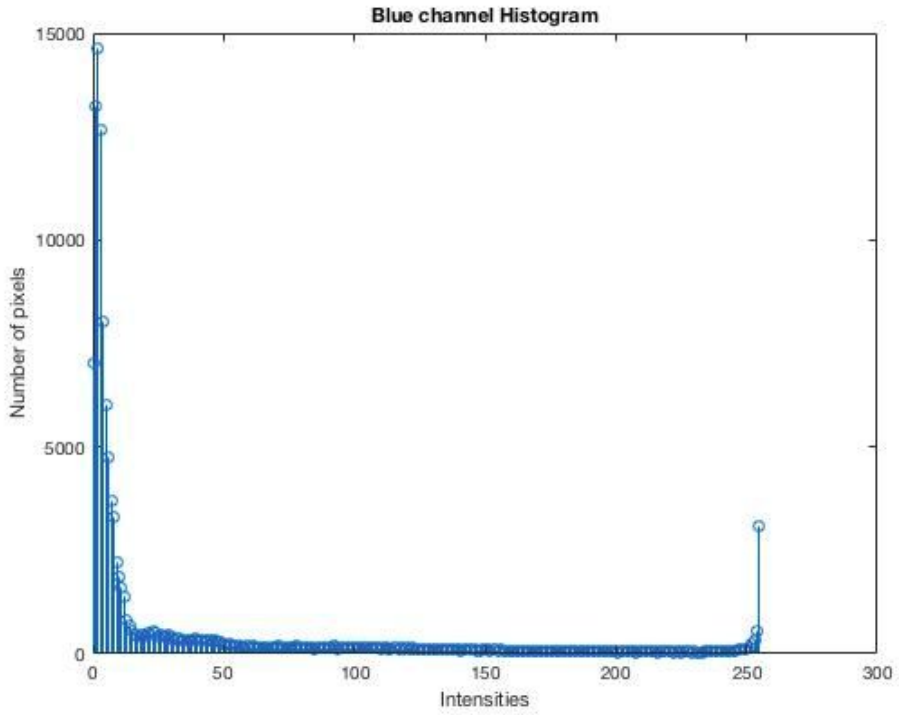
Procedure:

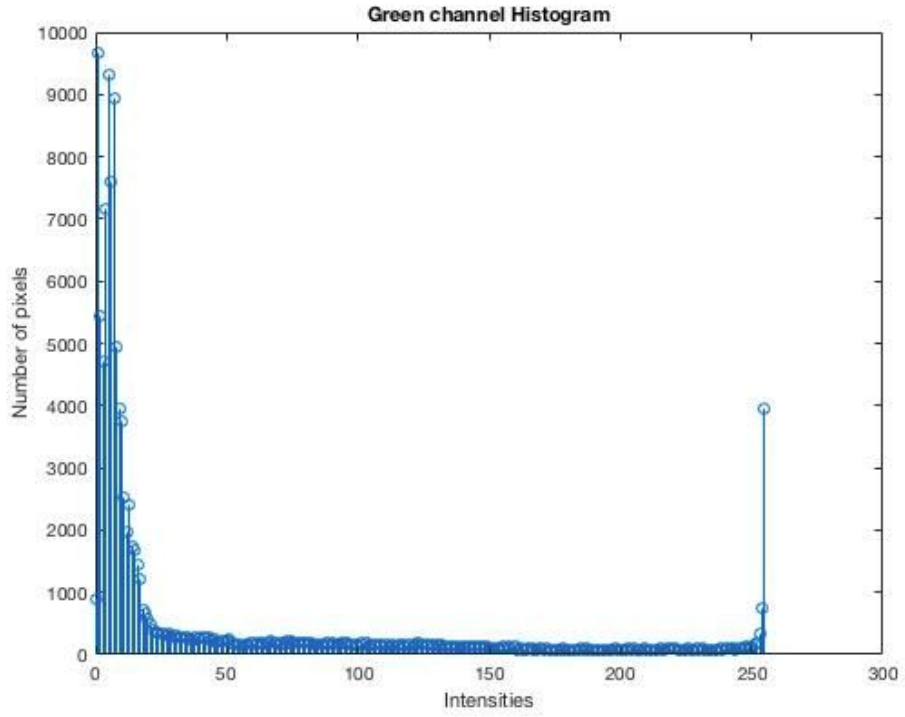
1. Read the input image file into a 1D array
2. Separate the R, G and B channels into 1D arrays for convenience while performing operations
3. Sort the pixels in the increasing order of intensity values with 0 being the smallest intensity value and 255 being the largest intensity value.
4. Store the corresponding index values of the sorted pixels in another array which is needed later in this algorithm.
5. Calculate the size of each bin - that is the number of pixels which are to be assigned the same intensity value using the below formula:

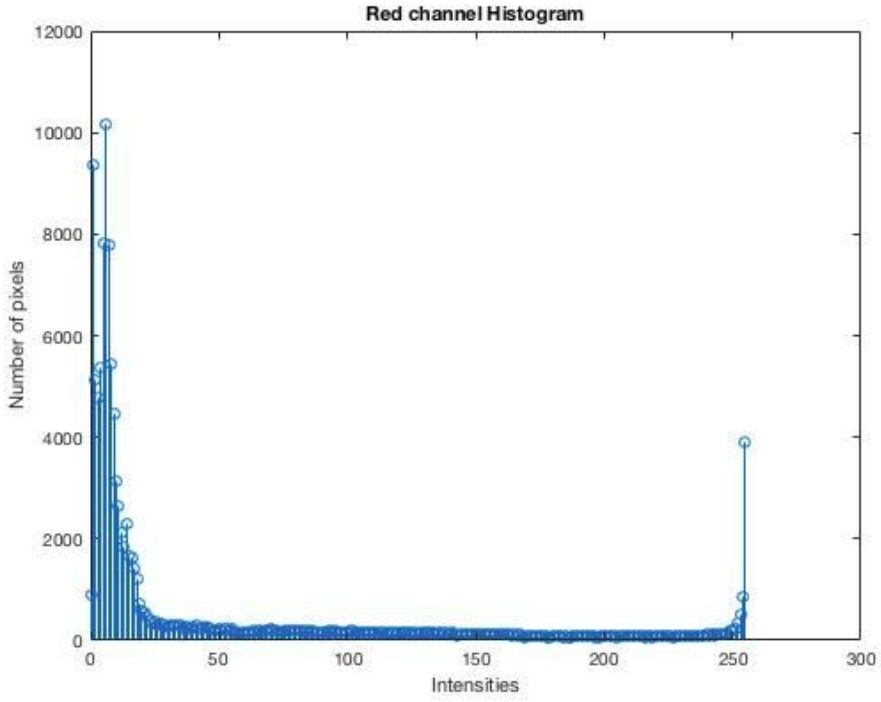
   ```
   Size of each bin = (Width_of_Image * Height_of_Image) / (Number of intensity levels)
   ```
6. For each set of sorted pixels (size of set is equivalent to the size of the bin), assign the one intensity value starting from 0 to 255.
7. Repeat the steps 3, 4, 5 and 6 for R, G and B channels
8. Combine the 3 channels into a 1D array and write it into .raw file.

Experimental Results:

<u>Histograms for each channel of the Original Image</u>




Green channel Histogram
Number of pixels
Intensities
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
50
100
150
200
250
300


Red channel Histogram
Number of pixels
Intensities
0
2000
4000
6000
8000
10000
12000
0
50
100
150
200
250
300

<u>Method A</u> - Transfer-function-based histogram equalization method

Original Image

Histogram Equalized Image

<u>Method B:</u> Cumulative-probability-based histogram equalization method

Original Image

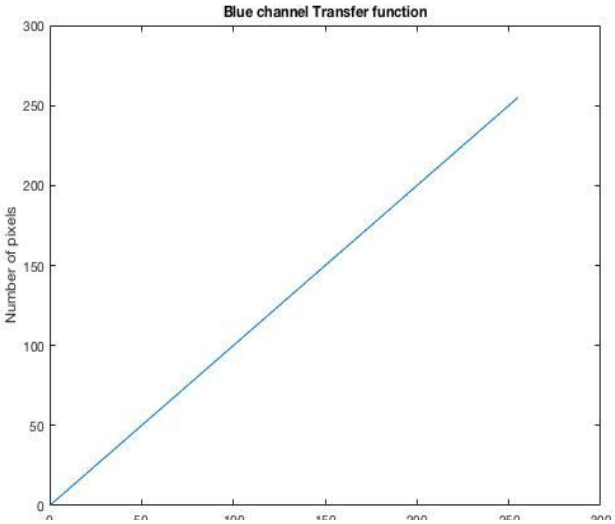
Histogram Equalized Image

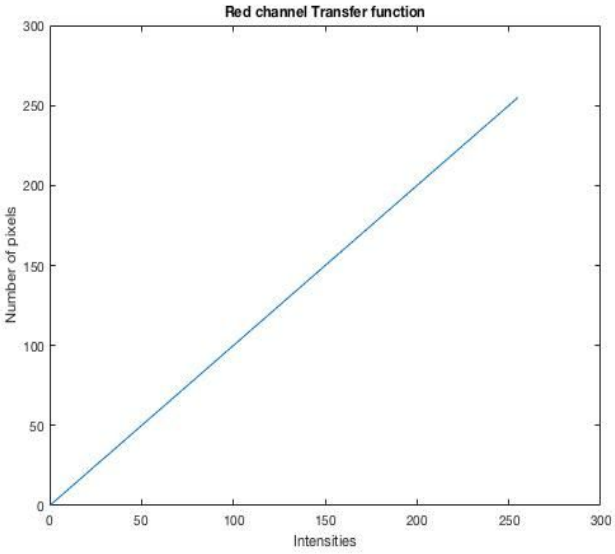
Discussion:

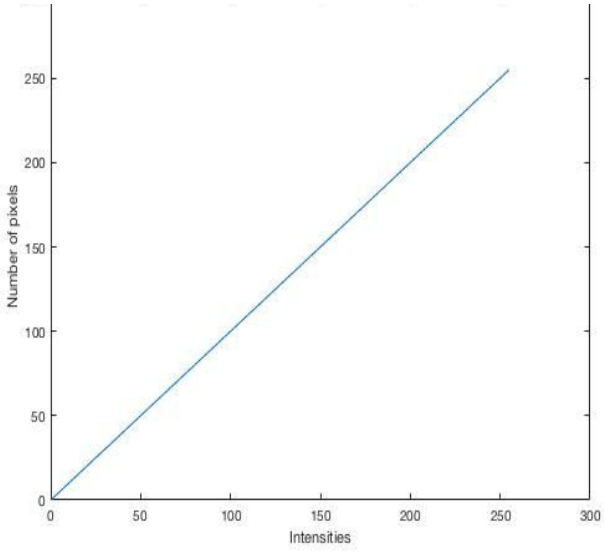
It was observed that the contrast of the Desk.raw image improved upon performing both the methods of Histogram Equalization. Both the methods showed distortion in the histogram equalized image and both did not show stark differences in quality when compared to each other. Histogram equalization is generally expected to produce unrealistic effects in real life images, but are useful for scientific images like thermal or satellite images.

The images look like they have some blocky distortions in the corners. Hence a smoothing filter can be used to correct it after performing the histogram equalization. A different histogram equalization method like the contrast limiting adaptive histogram equalization or CLAHE, multipeak histogram equalization (MPHE) can also be used instead of the above methods.

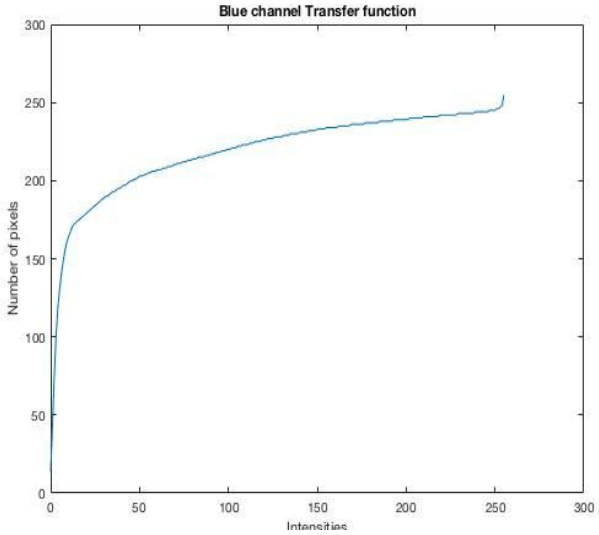
Method B: Transfer Function

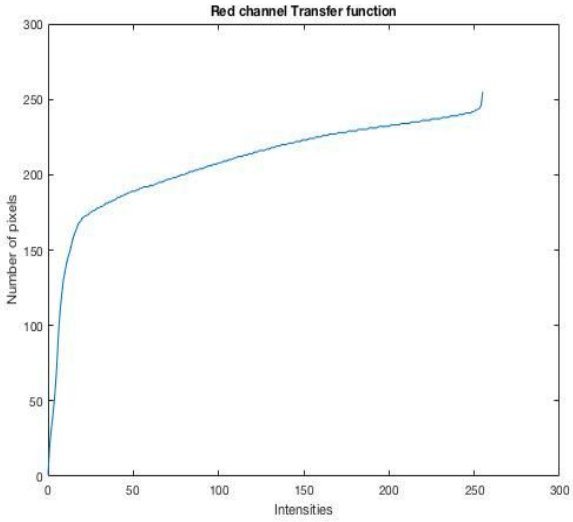






Method A : Transfer Function





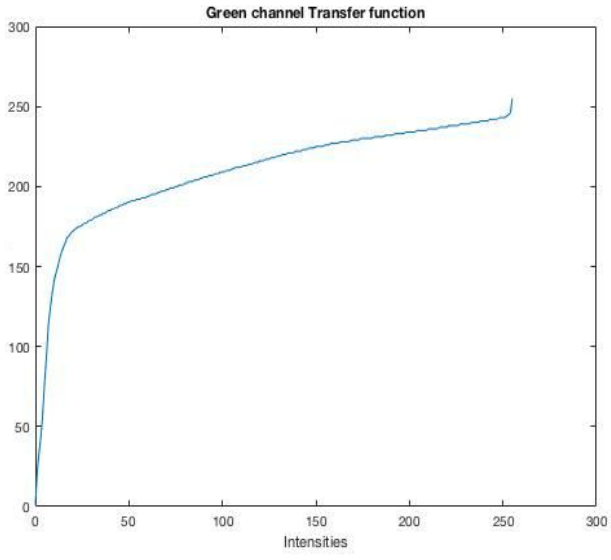


IMAGE FILTERING - CREATING OIL PAINTING EFFECT

ABSTRACT AND MOTIVATION

Image filtering is a technique of either improving the quality or modifying an image to make it look a certain way. One may want to filter an image an to either enhance the edges or make them smooth to get an even image. Generally, filtering is an operation involving pixels of the neighbourhood. The value of every pixel in an image is calculated by applying a certain formula or algorithm to set of the pixels surrounding the image within a relative distance. The distance of the neighbourhood pixels from the target pixel is determined by the size of the mask. Linear filtering in an image is the method of filtering where the output pixel is a linear combination of the pixels in its neighbourhood. This is generally accomplished by convolution or correlation operators.

In this section we explore the oil painting effect, the input image is transformed into an image with the visual appearance of a oil painting.

APPROACH:

The oil painting effect consists of two steps - Quantization and Image filtering. In the quantization step, the colors in the input image are quantized. This is done by mapping the 256 colors in each of the channels(R, G and B) into 64 colors. This in effect results in each channel having only 4 colors. In the second step of image filtering, for each of the quantized channels, an NxN neighbourhood is considered and the most frequent color in that neighbourhood is chosen a the representative color in the final oil painting image.

PROCEDURE:

1. Read the input .raw file into a 1D array.
2. Separate the R, G, B channels by writing to a 1D array
3. Sort the pixels in the increasing order of intensity values with 0 being the smallest intensity value and 255 being the largest intensity value.
4. Store the corresponding index values of the sorted pixels in another array which is needed later in this algorithm.
5. Divide the sorted color intensities in the entire image into 4 bins of 64 intensities each.
6. Calculate the average of pixels in each pixel bin and replace, all the pixels in that bin by that mean value.
7. Repeat the steps 3 - 6 for all the channels
8. Combine the 3 channels to create the Quantized.raw image as output.
9. To perform the filtering operation, an N x N mask has to be created which will pick the most frequent pixel in that N xN neighbourhood and replace the target pixel.
10. This mask has to traverse through the entire image and has to take into account the edges of the image.
11. Repeat steps 9, 10 for all the 3 color channels of the image
12. Combine the 3 channels to write the Oil_Painting.raw image as output.

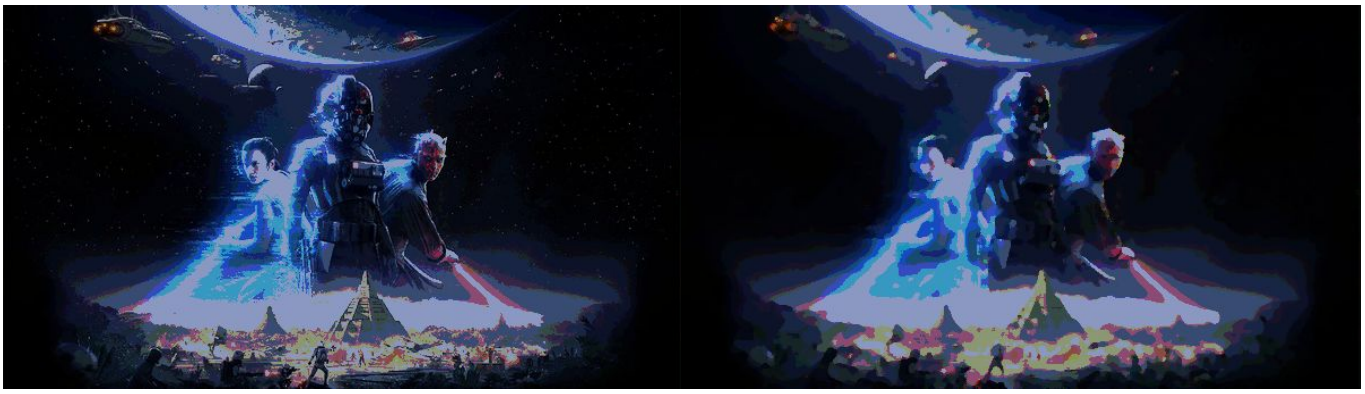
Experimental Results:

Quantized Star Wars image
(256 colors to 64 colors conversion)

Star Wars image with the oil painting effect
(N = 5)

Trojans Quantized Image
(256 colors to 64 colors conversion)

Trojans Oil Painting Image
(N = 5)



Quantized Star Wars image
(256 colors to 64 colors conversion)

Star Wars image with the oil painting effect
(N = 5)

Trojans Quantized Image
(512 colors to 64 colors conversion)

Trojans Oil Painting Image
(N = 5)

Discussion:
It was observed that as the size of the NxN mask increased the Oil Painting output image becomes more and more blocky and blurred. The optimal mask size was found to be N = 5. When the N=5 sized was used with the colors in the input file set to 512, the image quality in the oil painting image proved. For example, the sword in the Trojans_Oil_Painting.raw which was not seen in the image with input colors as 256, but was seen in the output image when the input colors value was increased to 512.

IMAGE FILTERING - CREATING FILM EFFECT

ABSTRACT

The aim of this section is to analyse the given image with a certain effect and replicate the same effect on another input image. This has to be done by analyzing the histograms and pixel intensities of the input image.

APPROACH:

When the input Film.raw image and the Original.raw images were analyzed, it was observed that the Film.raw image was flipped vertically and the negative of the pixel intensities were taken. Upon further analysis it was observed that the in the histograms of the color channels were to be compressed for the film effect appearance.

PROCEDURE:

1. Read the Film.raw, Original.raw and the Girl.raw images into 1D arrays and separate the R, G and B channels of the image into 1D arrays
2. Flip the Original.raw and Girl.raw images vertically by traversing to the end of each row and writing into new files, after subtracting it from 255(max intensity value).
3. Obtain the cdf of both the Film.raw and the flipped and negative Original.raw
4. Compare both the cdfs and obtain the new set of intensity values.
5. The new intensity values are obtained by comparing each cdf value of the Original.raw against the Film.raw to find a closest match.
6. The intensity value of the corresponding closest match cdf value, will be the new intensity value.
7. Repeat the steps 3 - 6 for all the color channels.
8. Replace the intensity values in the flipped and negated Girl.raw image by the new set of intensity values obtained by the above method.

Experimental Results:

DISCUSSION:
The Film effect was applied on the Girl.raw image and the output was observed to be similar to the Film.raw image. The Girl.raw image was flipped and negated. The difference in the histograms of Film.raw and the Original.raw was used to get the final Girl_Film_effect.raw image.

PROBLEM 3
NOISE REMOVAL

A. MIX NOISE IN COLOR IMAGE

ABSTRACT
Noise in an image can cause a problem while analyzing and processing the image. Hence noise removal is a very important pre-processing step to get clear and useful image. Noise removal can be done by using multiple filters. Median, mean, Gaussian are some of the filters which can be used.

APPROACH
In this section we try to analyse input Lena image to identify the different kinds of noise, remove them using different combinations of filters on each of the channels and obtain the final denoised image.

PROCEDURE

1. Read the noisy input image and separate the R, G and B color channels
2. Apply the Median filter on each of the color channels to obtain a filtered image.
3. The median filter is actually a NxN mask which sorts the pixels in the NxN neighbourhood according to the value of intensities and gives the center value in the list
4. Next apply the Mean filter to only the Blue channel.
5. The Mean filter is an averaging filter, which takes the average of all the pixels in the NxN neighbourhood and the representative pixel with replace the target pixel in the Final Image after all the denoising.
6. Combine the 3 color channels - R,G and B and write them into a single file to give the denoised Lena image.

EXPERIMENTAL RESULTS:

Original Image

Denoised Lena Image

PSNR for Clean vs mixed-noise image : 16.8762 dB
PSNR for R component : 26.2306 dB
PSNR for G component : 26.6024 dB
PSNR for B component : 26.0669 dB
PSNR for ALL components combined : 26.2943 dB

DISCUSSION

It was observed that the results obtained on applying the median and mean filters one after the other to each of the color channels had less effect on the noisy image compared to the current case where the median filter was applied to R,G and B channels, but the mean was filter was applied to only the Blue channel. The optimum value of PSNR was obtained with the median filter applied to all channels and the mean filter to the Blue channel with N = 5(filter size). Hence

it can be said that the R, G and B channels - all contain salt and pepper noise and the B filter contains the Gaussian noise, hence the above combination of filters produced a better result

B. **PCA**

ABSTRACT

PCA is a dimensionality reduction method which also reduces the noise when applied to a noisy image. PCA arranges all the signal components of the image along the line of maximum variance. The other noisy components are hereby eliminated as we consider only the components along the maximum variance line. It has been hypothesised that the ratio of signal variance to noise variance should be about 2.5 - 3.

EXPERIMENTAL RESULTS

20.19dB **31.30 dB** **39.8 dB**

DISCUSSION

PCA is computationally more complex and efficient, thereby performs better than the filters used in Part A of this section. In Patch based local PCA. the results obtained are better since several patches exhibiting similar variance are considered at a time. This makes sure the signal component in the image is preserved.

References:

1. Pratt W (2007) Digital image processing. Wiley-Interscience. W. Pratt2007Digital image processingWiley-Interscience
2. https://en.wikipedia.org/wiki/Bilinear_interpolation
3. http://www.cs.cornell.edu/courses/cs1114/2009sp/assignments/A5P1.pdf
4. http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029740
5. http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html#histogram
6. http://spatial-analyst.net/ILWIS/htm/ilwisapp/stretch_algorithm.htm