DIP Course
MiniProject 2
Color & Quantization

Part 1: Color spaces

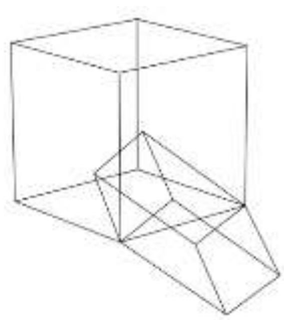
1. Use ‘pasta.tif’ and show three color channels (R, G, and B) separately.
2. Convert color space to HSV and show Hue, Saturation and Intensity channels.
3. Convert color space to YCbCr and show Y, Cb and Cr channels.
4. Convert the original image to grayscale by the use of equation below.

$$GI = 0.299R + 0.587G + 0.114B$$

YIQ color space

This color space is used for TV/video in America and other countries where NTSC is the video standard (Australia uses PAL). In this scheme Y is the “luminance” (this corresponds roughly with intensity), and I and Q carry the color information(chrominance).

5. Use below Equation for producing of YIQ components from RGB image.



The RGB cube and its YIQ transformation.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

6. Now compute the covariance matrix and correlation coefficients for each color space. Explain what you obtain as results.

Note that before computing covariance matrix for each color space reshape each image to 2d matrix with 3 columns for each channel.

Part 2: uniform quantization, Pseudo-Coloring and histogram ,color filtering

1. Use ‘lenna.tif’ and convert it to grayscale.
2. Write a function to implement a uniform quantization for grayscale images and measure the quality of the output image, in terms of MSE.

$$function\ [imgq, mse] = quanz(img, n)$$

where img is the input image, n specify the number of quantization levels, imgq stores the quantized output image, and mse is the MSE value of the quantized image. (The lower the value of MSE, the lower the error).

2. Quantize the grayscale Image by 4 ,6 and 8 gray-levels, then show and compare them.
3. Plot histogram of the original and quantized image.
4. Use the following colormap for pseudo-coloring and map 4 grey-levels quantized image into an assigned color.

Color	Red	Green	Blue
Blue	0	0	1
Magneta	1	0	1
Green	0	1	0
Red	1	0	0

5. Why we need to assign colors to gray scale image?
6. Use ‘HSV.tif’ and try to Filter all colors except yellow.

Part3: Halftoning

1. Read the image ‘pasta.tif’ into your Matlab workspace.
2. Convert the original image to grayscale and show this image.
3. Halftone the image with a constant threshold.
4. Create Bayer threshold matrices of sizes 4 × 4, and 8 × 8.
5. Generate two different halftone images by applying these two matrices.
6. Compare the results.

7. use ‘peppers.tif’ as input image and compute the Y component of YCbCr as its intensity image.

7.1. Apply the Floyd-Steinberg algorithm by yourself to binarize the intensity image of peppers.

7.2. Show the spectral characteristic (using DFT) of the original image and binarized image.

7.3. Show the spectral characteristic (using DFT) of the difference between original intensity image and binarized image (namely the noise added to the original image)

7.4. Show the spectral characteristic (using DFT) of the difference between the input and output images of the quantizer block (namely the quantization noise).

7.5. What range of frequency do you observe and why?

7.6. Compute the correlation between the original intensity image and the noise added to this image. What is your conclusion?

Note that for computing the correlation between two matrices, use the command corr2 of Matlab. Describe this command and its max and min values.

8. Convert ‘lenna.tif’ to binary by using the dither command of Matlab. binarize each color channel of RGB space independently. Then Show binarized color image. (This function uses Floyd-Steinberg's error diffusion dither algorithm)

8.1. Apply the Floyd-Steinberg algorithm by yourself to binarize each color channel of RGB space independently.

9. Compare the Floyd-Steinberg method with ordered dithering (Bayer matrix) method.