



Course: Image Processing
Prof. Mylène C.Q. de Farias
Semester: 2017.1

LIST 04

Submission Date: 04/05/2017; Cut-off: 14/05/2017

Part 1 – Theory

do livro texto

Question 1.1:

- (a) How many gray levels are there in the RGB color system, taking into account that an image has 8 bits?
- (b) In an RGB image, the R, G, and B components have a horizontal profile of intensities, as presented in the diagram in Figure 1. What is the color of the middle column of this image? What are the color of the borders of the image?

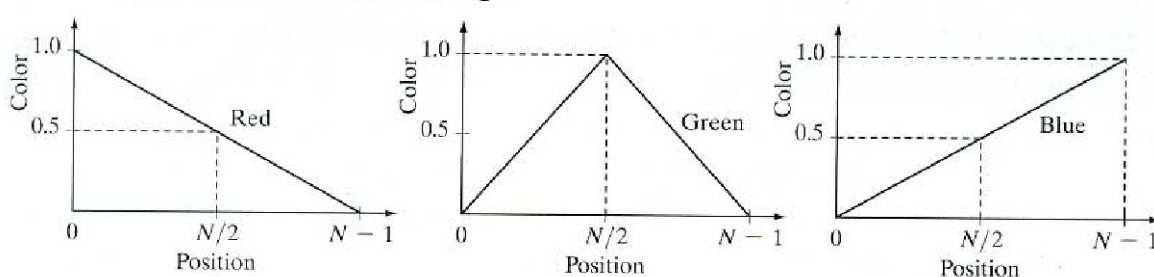


Figure 1: horizontal profile of the R, G and B components.

Question 1.2: Draw a surface in the RGB space for the points that satisfy the following equation:

$$D(\mathbf{z}, a) = [(\mathbf{z} - a)^T C^{-1} (\mathbf{z} - a)]^{1/2} = D_0,$$

where D_0 is a specific constant that has a value different from zero. Consider that $a = 0$ and

$$C = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Question 1.3: Consider the image in Figure 2. All colors have a maximum value of saturation and intensity.

- (a) Draw the RGB components of Figure 2, depicting how they would look in a monochromatic (gray scale or black-and-white) monitor.
- (b) Draw the HSI components of Figure 2, depicting how they would look in a monochromatic (gray scale or black-and-white) monitor.

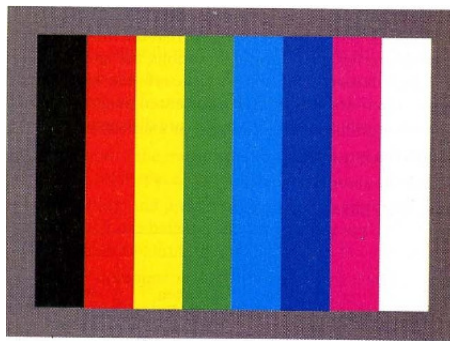


Figure 2: Color Bars

Question 1.4: Consider the color image of size 500×500 shown in Figure 3. The squares in this image correspond to the ‘pure’ colors blue, red, and green.

- (a) Consider that we convert an RGB image to HSI. Next, we filter the H component of the HSI image using an arithmetic mean filter of size 25×25 . Then, convert the HSI image back to RGB. What is the appearance of the resulting image?
- (b) Repeat (a), filtering the S component (instead of the H component). What is the appearance of the resulting image?

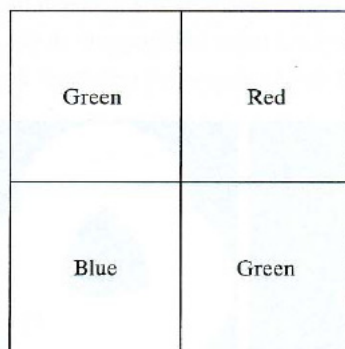


Figure 3: Image in Figure 6.16 of the text book.

Question 1.5:

- (a) Compute the entropy of a source with a probability of symbols given in Table 1 (Table 8.1 from text book).
- (b) Build a Huffman code for the symbols from this source and explain the differences between this code and the binary code (Code 1) in Table 1.
- (c) Build a shift binary code for this source.
- (d) Split the symbols in two groups and build another code using a shifted Huffman.
- (e) Compute the word average lengths for each code and compare these values with the source entropy value calculated in (a).

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0,25	01010111	8	01	2
$r_{128} = 128$	0,47	10000000	8	1	1
$r_{186} = 186$	0,25	11000100	8	000	3
$r_{255} = 255$	0,03	11111111	8	001	3
r_k para $k \neq 87, 128, 186, 155$	0	-	8	-	0

Table 1: Probabilities of the source in Question 1.5.

Question 1.6 Decode the message 0.23355, which was coded using a arithmetic code with symbols probabilities depicted in Table 2. Consider that the message has 5 symbols.

Símbolo	Probabilidade
a	0,3
e	0,2
b	0,2
c	0,1
d	0,1
f	0,1

Table 2: Probabilities of the symbols of the source in Question 1.6.

Question 1.7

- (a) Explain in details the JPEG algorithm. Include a block diagram in your explanation.
- (b) Explain in details the MPEG algorithm. Include a block diagram in your explanation.

Part 2 – Practice

Question 2.1:

- (a) Write a program that converts a color RGB image to a color RGB-safe image.
- (b) Download the image in Figure 4 from the course homepage and convert this image to RGB-safe.

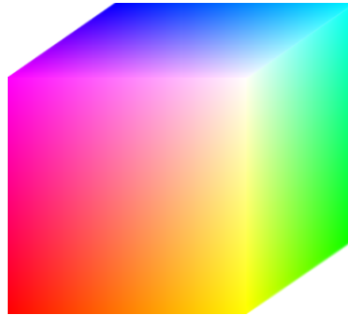


Figure 4: Cube in RGB space.

Question 2.2:

- (a) Implement a color processing algorithm that generates false colors (pseudo-colors), similarly to what was presented in class. In this system, you must specify two intervals of gray scale levels in the input image. The algorithm must generate a color image (RGB) with pixels with a specific color for each interval of gray scale levels of the input image. The remaining pixels of the input image (outside the two intervals) should keep the original gray scale levels.
- (b) Download the image in Figure 5 from the course homepage (Figure 1.10 (4) of the text book). Using the algorithm developed in (a), process the image in a way that the river in the center of the image appears yellow, while the rest of the image maintains the original gray scale levels. It is acceptable to have a small number of isolated yellow pixels in the output image.



Figure 5: Figure 1.10 (4) of the text book.

Question 2.3:

- a) Pick a gray scale image of your choice, with a minimum size of 256×256 and save in an uncompressed format.
- b) Read and visualize the chosen image.
- c) Apply a 2D-DCT to each of the 8×8 blocks of the image. Normalize the result, dividing all resulting matrix elements by 8. Visualize the result.
- d) The quality parameter in compression (JPEG) varies from 1 to 100. This value controls the quantization compression scale:

$$\text{scale_quant} = \begin{cases} \frac{50}{\text{quality}}, & \text{quality} < 50 \\ 2 - \frac{\text{quality} * 2}{100}, & \text{otherwise} \end{cases} .$$

The standard quantization table is given by:

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} .$$

The quantization matrix for the desired quality level can be obtained using the following function in Matlab: `QuantT = max(ones(size(Q)), round(escala_quant * Q));`

- e) Set the quality level to 50. Divide the elements in each one of the 8×8 blocks of DCT coefficients by the corresponding element in the quantization matrix, rounding the result to the next integer. Notice that `QuantT` has 8 lines and 8 columns, similarly to the DCT coefficient blocks. You have just performed a simplified JPEG.
- f) Compute the original image entropy. Compute the entropy of the image obtained in the last step. How do these 2 numbers compare? What can you conclude from this result?
- g) To recover the image, multiply each of the blocks obtained in (e) by the same `QuantT` element. Next, compute the IDCT of the 8×8 blocks and multiply the result by 8. Visualize the recovered image. What type of distortions can be visualized?
- h) Compute the mean squared error between the original (`Io`) and recovered (`Ir`) images: `[l,c]=size(Io); MSE=sum((Io-Ir)^2)/l*c;`
- i) Repeat (c)-(h) for quality levels 100, 90, 80, 70, and 60. Compare the MSE values for each of these values. Plot a graph of MSE versus quality level (100-50).
- j) Do the same test in (i) for 3 different images. Are the MSE results still consistent? Can you compare the MSE results among the different images? Plot a graph with the 4 MSE curves (versus quality levels).
- k) Download the SSIM metric from the link below. Repeat the same tests performed in (i) and (j) using SSIM, instead of MSE? Plot a graph of SSIM versus quality level (100-50). Compare the results obtained with SSIM with the results obtained with MSE.

<https://ece.uwaterloo.ca/~z70wang/research/ssim/>