

Problem #1: [20 pts]

A given colored image has 300-pixel width and 200-pixel height. The image is coded in RGB where each component is coded using 8-bits.

1. Compute the number of pixels in the image [2 pts]. Compute the pixel size in bits [2 pts] and the image size in bytes [4 pts].
2. If the image is quantized by using 3 bits for the red component, 3 bits for the green component and 4 bits for the blue component. Compute the new pixel size and the new image size [4 pts]. Is this transformation loosely or lossless, justify! [3pts].
3. The original image is sub-sampled by taking the average of each 3x2 pixels. Compute the new width and height of the image [3 pts]. Deduce the new image size in bytes [2 pts].

Problem #2: [10 pts]

An indexed image of 1200x800 pixels have 120 different RGBA colors.

1. [4 pts] How many entries will be in the index table? Compute the size of the table in bytes.
2. What will be the size of the image in bytes if :
 - a. [4 pts] we store the index table and the indexed image
 - b. [2 pts] we store each pixel in RGBA format

Problem #3: [20 pts]

An application produce images where the most of pixels are black, white or fully transparent, and few pixels are colored with a level of transparency. The image is stored as following:

- 2 bits are used to indicate if the pixel is white, black, transparent or colored (00 for black, 11 for white, 01 for full transparent and 10 for colored pixel)
- If the pixel is colored then 32 bits (8*4) that represent the RGBA value of the pixel follow the previous 2 bits. In case of white, black or fully transparent we do not store the pixel value.

1. [5 pts] How will be stored the following RGBA image (use 0, 1,2,3 instead of 00 01 10 11)
(Note: A = 0 → full transparent, A=255 full opaque)

(100,20,0,20)	(0,0,0,50)	(250,50,30,0)	(255,255,255,100)
(0,0,0,20)	(0,0,0,50)	(255,255,255,10)	(255,200,255,0)
(100,20,0,20)	(0,0,0,50)	(0,0,0,60)	(255,255,255,50)

2. [10 pts] Write a pseudo-code algorithm to transform an RGBA image to the above described format
3. [5 pts] Given an image of 1000x2000 pixels where 40%, 20%, 30%, 10% are black, white, full transparent and colored respectively. Compute the image size in Kbytes using full RGBA format and using the above-described format.

Problem #4: [10 pts]

A video source produces 300x200 images, interlaced at a field rate of 60Hz, and full color (24 bpp). You first convert the video to 4:1:1 color format representation. The component Y of the pixel is quantized using 8 bits and Cr , Cb using 6 bits each.

1. [7 pts] Compute the bite rate of the video source.
2. [3 pts] Compute the size in bytes of a 10 minutes video.

Problem #5: [8 pts]

The enclosed 4x4 matrix represent a 256 level gray scale image. Convert it to binary image by applying the following methods:

1. [2 pts] The naive conversion
2. [3 pts] The average dithering
3. [3 pts] The median cut dithering

100	181	240	80
130	30	90	110
60	40	5	95
65	80	70	80

Problem #6: [14 pts]

A message contains two symbols (A and B) with the distribution probabilities 20% for A and 80% for B.

1. [7 pts] Encode the message ABAA using arithmetic coding (show all calculation steps)
2. [7 pts] Decode the number 0.4 to obtain a 3 symbols length message (show all calculation steps)

Problem #7: [18 pts]

A message contains 6 symbols with the following distribution probabilities

1. [8 pts] Apply the Huffman coding to obtain symbols' codes (show the encoding tree)
2. [2 pts] Compute the average symbol length
3. [4 pts] Compute the entropy and the compression efficiency
4. [2 pts] Encode the word : ABED
5. [2 pts] Decode the message: 111001111110

A	0.4
B	0.09
C	0.04
D	0.07
E	0.25
F	0.15

Good Work