Theory Part (40 points)

Question 1: Color Theory (10 points)

1. The chromaticity diagram in (x, y) represents the normalized color matching functions X, Y and Z. Prove tha (2 points)

$$Z = [(1-x-y)/y]\,Y$$

Solu:

$$[(1-x-y)/y]\,Y \Rightarrow$$

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

$$[(1-x-y)/y]\,Y = \left[\frac{1 - \frac{X}{X+Y+Z} - \frac{Y}{X+Y+Z}}{\frac{Y}{X+Y+Z}}\right] Y$$

$$= \left[\frac{\frac{(X+Y+Z)-X-Y}{\cancel{X+Y+Z}}}{\frac{\cancel{Y}}{\cancel{X+Y+Z}}}\right] \cancel{Y}$$

$$= \cancel{X}+\cancel{Y}+Z-\cancel{X}-\cancel{Y}$$

$$\therefore \boxed{[(1-x-y)/y]\,Y = Z}$$

2) Here you are tasked with mapping the gamut of a printer to that of a color CRT monitor. Assume that gamuts are not the same, that is, there are colors in the printer's gamut that do not appear in the monitor's gamut and vice-versa. So in order to print a color seen on the monitor you choose the nearest color in the gamut of the printer (8 points)

- Comment (giving reasons) whether this algorithm will work effectively? (2 points)

<u>Soln:</u> There are actually two types of color gamuts, additive and subtractive. Subtractive style is used in printed media such as photos, magazines and books. It is also generally referred to as CMYK based on the Cyan, Magenta, Yellow and Black pigments used in the printing.

Subtractive color is that used by mixing together dyes that prevent reflection of light that then produce a color.

Here the algorithm used is clipping wherein map all the values that are in the source gamut but outside the destination gamut onto the closest colors on the boundary of destination gamut.

This algorithm works well because: Our eyes are much better in evaluating color relationships then they are evaluating absolute colors. Our eyes adapt to different colors of white which is called chromatic adaptation.

When printing there is often a visible paper white border. Since white areas in an image will almost always have some color tint, the clipping will have a color cast because our eyes adapt to paper-white surround and not the image white.

- You have two images - a cartoon image with constant color tones and a real image with varying color tones. Which image will this algorithm perform better - give reasons (2 points)

<u>Solu:</u> Tones are obtained by adding gray to a pure hue and each gradation gives is a different tone.

As in clipping, in some color channels when an image is rendered to a different color space, colors that fall outside the target color space are said to be clipped. Such colors are referred to as out of the gamut.

For a cartoon image with constant color tone this algorithm works well because our eyes are much better in evaluating color relationships than they are evaluating absolute colors, and our eyes adapt to paper white surround and not the image white while printing. So when clipping is done for any image with constant tone, the effect is not significant to a human eye.

Whereas with an image with varying color tones, the difference is ~~ofen~~ significant to human eye and so clipping is ineffective.

- Can you suggest improvements rather than just choosing the nearest color? (4 points)

<u>Solu:</u> In nearly every translation process, we have to deal with the fact that the color gamut of different devices vary in range which makes an accurate reproduction impossible. Therefore we need some rearrangement near the borders of the gamut.

This so-called gamut mismatch occurs for eg: when we translate from the RGB color space with a wider gamut into the CMYK color space with a narrower gamut range.

The dark highly saturated purplish-blue color of a typical computer monitor's "blue" primary is impossible to print on paper with a typical CMYK printer. Conversely, an inkjet printer's "CYAN" primary, a saturated mid-brightness blue, is outside the gamut of a typical computer monitor.

Rendering Intent:

When the gamut of source color space exceeds that of the destination saturated colors are liable to become clipped or more formally burned.

The color management module can deal with this problem in several ways.

Color management module:

Color matching module (also method or system) is a software algorithm that adjusts the numerical values that get sent to or received from different devices so that the perceived color they produce remains consistent.

Some well known CMM's are ColorSync, Adobe CMM, LittleCMM and ArgyllCMS.

Operating System Level:

Apple's classic Mac OS and macOS operating systems have provided OS-level color management APIs since 1993, through ColorSync.

Since 1997 color mgmt in Windows is available through an ICC color management system

File Level:

Certain image filetypes (TIFF and Photoshop) include the notion of color channels for specifying the color mode of the file. The most commonly used channels are RGB (mainly for display) and CMYK (for commerical printing)

Application Level:

As of 2005, most web browsers ignored color profiles. Notable exceptions were Safari, starting with version 2.0 and Firefox starting with version 3. Notable browser support for color mgmt are:

- Firefox: from version 3.5 enabled by default for ICC v2 tagged images.
- Internet Explorer: version 9 is the first Microsoft browser to partly support ICC profiles
- Google Chrome: uses the system provided ICC v2 and v2 support on macOS, and from version 2 supports ICC v2 profiles
- Safari: has support starting with version 2.0
- Opera: has support since 12.0 for ICC v4.
- Pale Moon supported ICC v2 from its first release, and V4 since Pale Moon 20.2 (2013)

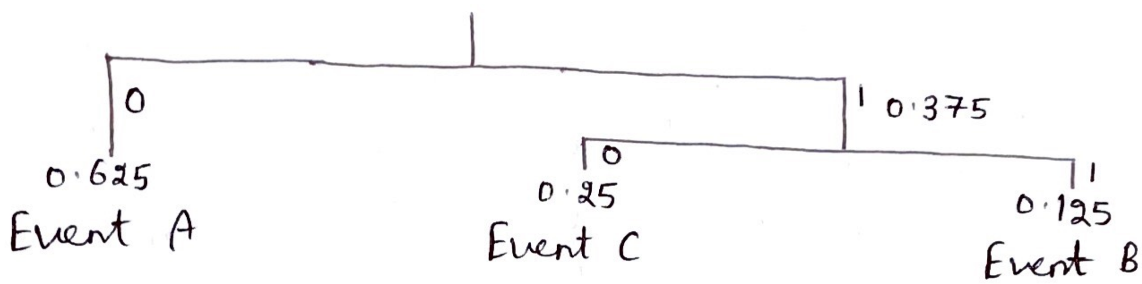
Question 2: Generic Compression: (10 points)

Consider a source that emits an alphabet with three symbols A, B, C having probability distribute as follows − $P(A) = 0.625$, $P(B) = 0.125$, $P(C) = 0.25$

- Construct a Huffman code and calculate its average code length. (2 points)

Solu: $P(A) = 0.625$, $P(B) = 0.125$, $P(C) = 0.25$

Event name	Probability
A	0.625
B	0.125
C	0.25



Event	Probability	Code	Length
A	0.625	0	1
B	0.125	10	2
C	0.25	11	2

Average length $= \sum_{i=1}^{N} P_i \times L_i$

$= 0.625 \times 1 + 0.125 \times 2 + 0.25 \times 2$

$= 1.375$

• For this three-symbol vocabulary, how many Huffman codes are possible. What are they? (2 points)

Solu: Total number of different Huffman code tree is given 2^{n-1}, where n is no. of symbols.

$$2^{3-1} = 2^2 = 4$$

Codes:

0	0	1	1
10	11	00	01
11	10	01	00

• Is the code you have defined optimal - give reasons! If not, what can you do to improve upon this. Show a solution by an example computing the average code length (6 points)

Solu: The necessary conditions for an optimal variable length binary code are:

1) Given any two letters a_j and a_k if $P(a_j) \geq P(a_k)$ then $l_j \leq l_k$.

2) The two least probable letters have codeword with the same maximum length

3) In the tree corresponding to the optimum code, there must be two branches stemming from each intermediate node.

4) Suppose we change an intermediate node into a leaf node by combining all the leaves descending from it into a composite word of a reduced alphabet. Then, if the original tree was optimal for the original alphabet, the reduced tree is optimal for reduced alphabet.

Huffman coding is known to satisfy all the above conditions of Optimal codes, so the code obtained is optimal.

Question 3: Entropy Coding (10 points)

Consider a communication channel system that gives out only two symbols X and Y. Assume that the parameterization followed by the probabilities are $P(X) = x^k$ and $P(Y) = (1-x^k)$

- Write down the entropy function and plot it as a function of x for k=2 (1 point)

Solu:

$$P(X) = x^k \quad , \quad P(Y) = 1 - x^k$$

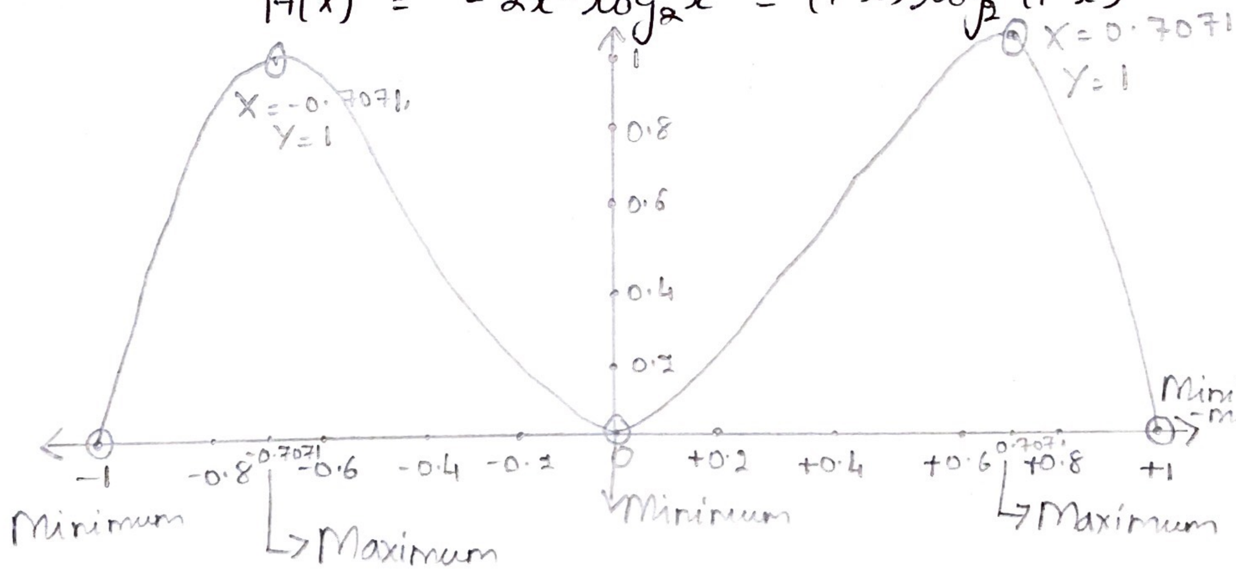
$$k = 2$$

$$P(X) = x^2 \quad , \quad P(Y) = 1 - x^2$$

$$H(X) = -\sum P_i \log P_i$$

$$= -x^2 \log_2 (x^2) - (1-x^2) \log_2 (1-x^2)$$

$$H(X) = -2x^2 \log_2 x - (1-x^2) \log_2 (1-x^2)$$



Entropy Plot

• From your plot, for what value of x with $k=2$ does H become a minimum (1 point)

Solu: Minimum entropy can occur in the case when only 1 symbol is present in the entire system or

$$P(X)=1 \quad \text{or} \quad P(Y)=1$$

$$x^2=1 \quad \text{or} \quad 1-x^2=1$$

$$x=\pm 1 \qquad x=0$$

$\therefore$ Entropy will be minimum at $x=\pm 1, 0$

• Your plot visually gives you the minimum value of x for $k=2$, find out a generalized formula for x in terms of k for which H is minimum (1 point)

Solu: Minimum entropy can occur in the case when only 1 symbol is present in the entire system or

$$P(X)=1 \quad \text{or} \quad P(Y)=1$$

$$x^k=1 \quad \text{or} \quad 1-x^k=1$$

$$\boxed{x=\pm 1} \qquad x^k=0$$

$$\boxed{x=0}$$

• From your plot, for what value of x with $k=2$ does H become a maximum?

Solu: Entropy will be maximum when both the probabilities are equal

$$P(X)=P(Y)$$

$$(1-x^2)=x^2$$

$$x^2=\frac{1}{2}$$

$$x = \pm \frac{1}{\sqrt{2}}$$

$$\boxed{x = \pm 0.7071}$$

- Your plot visually gives you the maximum value of x for $k=2$, find out a generalized formula for x in terms of k for which H is a maximum (4 points)

Solu: Entropy will be maximum when both the probabilities are equal

$$P(X) = P(Y)$$

$$1 - x^k = x^k$$

$$x^k = \frac{1}{2}$$

$$x = \sqrt[k]{\frac{1}{2}}$$

$$\boxed{x = \pm \frac{1}{\sqrt[k]{2}}}$$

Question 4: DCT Coding (10 points)

In this question you will try to understand the working of DCT in the context of JPEG. Below is an 8X8 luminance block of pixel values and its corresponding DCT coefficients

188	180	155	149	179	116	86	96
168	179	168	174	180	111	86	95
150	166	175	189	165	101	88	97
163	165	179	184	135	90	91	96
170	180	178	144	102	87	91	98
175	174	141	104	85	83	88	96
153	134	105	82	83	87	92	96
117	104	86	80	86	90	92	103

- Using the 2D DCT formula, compute the 64 DCT values. Assume that you quantize your DCT coefficients uniformly with $Q=100$. What does your table look like after quantization? (3pts)

Soln: The FDCT and IDCT are defined as follows:

FDCT:

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} S_{yx} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

IDCT:

$$S_{yx} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{vu} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

where C_u and C_v are defined by

$$C_u = \begin{cases} 1/\sqrt{2}, & \text{for } u=0 \\ 1 & \text{for } u \neq 0 \end{cases}$$

The function dct2 in matlab conforms to this definition.

Quantization is done using the formula round $\left(\frac{x_{i,j}}{Qval}\right)$, where Qval = 100.

After FDCT:

1.0e+03 *

1.0162	0.2160	-0.0068	-0.0272	0.0293	-0.0208	-0.0112	0.0080
0.1361	0.0526	-0.0935	-0.0073	0.0340	-0.0188	-0.0113	0.0106
-0.0459	-0.0492	0.0139	0.0538	0.0111	-0.0247	-0.0001	0.0084
0.0088	0.0381	0.0479	0.0156	-0.0179	-0.0109	0.0042	0.0037
-0.0013	-0.0059	-0.0012	-0.0047	0.0008	0.0066	0.0048	0.0002
-0.0045	-0.0012	0.0033	0.0081	0.0070	0.0061	-0.0002	0.0012
-0.0029	-0.0021	0.0009	-0.0015	0.0000	-0.0034	-0.0009	-0.0012
-0.0008	-0.0034	-0.0006	-0.0018	-0.0042	-0.0013	0.0023	0.0016

After Quantization

10	2	-1	-1	0	-1	-1	0
1	0	-1	-1	0	-1	-1	0
-1	-1	0	0	0	-1	-1	0
0	0	0	0	-1	-1	0	0
-1	-1	-1	-1	0	0	0	0
-1	-1	0	0	0	0	-1	0
-1	-1	0	-1	0	-1	-1	-1
-1	-1	-1	-1	-1	-1	0	0

- In the JPEG pipeline, the quantized DCT values are then further scanned in a zigzag order. Show the resulting zigzag scan when Q = 100 (1 point)

Solu:

DC Coefficient = 10

AC Coefficient = 2 1 -1 0 -1 -1 -1 -1 0 -1

0 0 -1 0 -1 0 0 0 -1 -1 -1 -1 -1

0 0 -1 -1 0 -1 -1 -1 -1 0 -1 -1 -1

0 0 0 -1 -1 0 0 0 0 0 -1 -1 -1

0 0 0 0 0 -1 -1 -1 -1 -1 0 -1 0 0

- For this zigzag AC sequence, write down the intermediary notation (2 points)

Solu: For DC coefficient, let us assume previous block was 7, the DPCM diff is 3, which needs 2 bits to encode ∴ <2> <3>

For AC coefficients.

<0,1> <2>, <0,1> <1>, <0,1> <-1>, <1,1> <-1>, <0,1> <-1>,
<0,1> <-1>, <0,1> <-1>, <1,1> <-1>, <2,1> <-1>, <1,1> <-1>,
<3,1> <-1>, <0,1> <-1>, <0,1> <-1>, <0,1> <-1>, <0,1> <-1>, <2,1> <-1>,
<0,1> <-1>, <1,1> <-1>, <0,1> <-1>, <0,1> <-1>, <0,1> <-1>,
<1,1> <-1>, <0,1> <-1>, <0,1> <-1>, <3,1> <-1>, <0,1> <-1>,
<5,1> <-1>, <0,1> <-1>, <0,1> <-1>, <5,1> <-1>, <0,1> <-1>,
<0,1> <-1>, <0,1> <-1>, <0,1> <-1>, <1,1> <-1>

• Assuming these are luminance values, write down the resulting JPEG bit stream. for the bit stream you may consult standard JPEG VLC and VLI code tables. You will need to refer to the code tables from the ITU-T JPEG standard which also uploaded with your assignment (2 points)

Intermediary symbol	Binary representation of first symbol (prefixed Huffman codes)	Binary representation of second symbol (non-prefixed variable integer code)
<2> <3>	011	11
<0,1> <2>	00	10
<0,1> <1>	00	1
<0,1> <-1>	00	0
<1,1> <-1>	11	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<1,1> <-1>	11	0
<2,1> <-1>	11100	0
<1,1> <-1>	11	0
<3,1> <-1>	111010	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<2,1> <-1>	11100	0
<0,1> <-1>	00	0
<1,1> <-1>	11	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<1,1> <-1>	11	0

<0,1> <-1>	0 0	0
<0,1> <-1>	00	0
<3,1> <-1>	11100	0
<0,1> <-1>	00	0
<5,1> <-1>	1111010	0
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<5,1> <-1>	1111010	0
<0,1> <-1>	00	0
	00	0
<0,1> <-1>		
<0,1> <-1>	00	0
<0,1> <-1>	00	0
<1,1> <-1>	11	0
EOB	1010	

<u>Binary</u> <u>Stream</u>:

011110010 001 000 110 000 000 000 110 111000
110 1110100 000 000 000 000 11100 0 000110.0 00.000
000 110 000 000 11100 0 000 11110100 000 000
11110100 000 000 000 000 110 1010

- What compression ratio do you get for this luminance block? (2 points)

<u>Soln</u>: The compression ratio can go upto 50:1