*Question 1: Color Theory*

* Explain why this happened. Why do some tiles look bright, almost glowing, while others appear muted and devoid of their original color?

Every light source has a unique spectral distribution, which also indicates that each light source has a different spectral energy distribution. When different light source projects on the same object, in this case a Rubik’s cube, corresponding reflection from the same color block will have a different spectrum frequency (e.g. when red light shines on the white side of the Rubik’s cube the reflection on the white block looks much like a red block in comparison with the normal light source). Therefore, different light source (illumination) leads to different color reflection because of the change of visible wavelengths and frequencies.

* Assuming ideal conditions, you have the following lighting conditions to solve the puzzle – under pure yellow light or under red light. Which of these two light choices make it harder to solve? Give reasons for your choice of answer.

Rubik’s cube contains 6 colors: white, green, red, blue, orange, and yellow

When red light projects: white → red, green → black, red → red, blue → black, orange → red, yellow → red

When yellow light projects: white → yellow, green → green, red → red, blue → black, orange → orange, yellow → yellow

Therefore, yellow light sources produces a wider color separation for Rubik’s cube, which makes red light sources a worse choice for solving Rubik’s cube

*Question 2: Color Theory*

Given that:

Simplify the right hand side:

Which equals with the left hand side, thus proved

* Comment (giving reasons) whether this algorithm will work effectively?

Yes, it will work effectively. Human eyes cannot detect the difference between two color pixel if their coloring distance is within some threshold on color map.

* 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?

A cartoon image will perform better since the color variance is smaller in comparison with a varying color tone.

* Can you suggest improvements rather than just choosing the nearest color?

Choose furthest color instead of the nearest color

*Question 3: Entropy Coding*

* Write down the entropy function and plot it as a function of *x* for k=2*.*

![Chart, line chart

Description automatically generated]()

* From your plot, for what value of *x* with *k=*2 does H become a minimum?

When x = 0 or when x infinitely approaching 1

* 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 a minimum

We can calculate the derivative of entropy function given k:

When x < 0 and approach 0, the derivative is turning 0 from negative. When x > 0 and approach 0, the derivative is turning 0 from positive. Thus, x = 0 is a local minimum point for a general k value

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

When x = (1/2)^(1/2) and x = -(1/2)^(1/2)

* 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

When x < (1/2)^(1/k) and approach (1/2)^(1/k), the derivative is decreasing from positive. When x = (1/2)^(1/k), the derivative reaches 0. When x >(1/2)^(1/k) and x < 0, the derivative is increasing from negative. Thus, x = (1/2)^(1/k) is a local maximum point for the general equation. The same conclusion can be drawn for x = -(1/2)^(1/k) only when k is even number.



*Question 4: Huffman Coding/Entropy*

* Find and show a Huffman code for the body of Alice’s postcard (i.e. exclude “Dear Bob” and “Alice”). Treat each word as a symbol, and don’t include punctuation. What is the average code length?

We can count the frequency of each word shown from the letter and following table is the Huffman code result:

|  |  |  |  |
| --- | --- | --- | --- |
| Words | Binary code | Freq | Length |
| You | 0000 | 1 | 4 |
| would | 0001 | 1 | 4 |
| postcard | 0010 | 1 | 4 |
| hear | 0011 | 1 | 4 |
| Paris | 010 | 2 | 3 |
| the | 0110 | 1 | 4 |
| Louvre | 0111 | 1 | 4 |
| hope | 1000 | 1 | 4 |
| got | 1001 | 1 | 4 |
| from | 101 | 3 | 3 |
| you | 1100 | 1 | 4 |
| Hello | 11010 | 1 | 5 |
| this | 11011 | 1 | 5 |
| to | 11100 | 1 | 5 |
| love | 11101 | 1 | 5 |
| I | 1111 | 2 | 4 |

The average code length:

* Find a Huffman code for the telegram message. What is the average code length? How does it compare to the original letter?

We can count the frequency of each word shown from the letter and following table is the Huffman code result:

|  |  |  |  |
| --- | --- | --- | --- |
| Words | Binary code | Freq | Length |
| stop | 00 | 3 | 2 |
| love | 0100 | 1 | 4 |
| would | 0101 | 1 | 4 |
| hear | 0110 | 1 | 4 |
| hope | 0111 | 1 | 4 |
| you | 100 | 2 | 3 |
| in | 1010 | 1 | 4 |
| postcard | 1011 | 1 | 4 |
| paris | 1100 | 1 | 4 |
| louvre | 1101 | 1 | 4 |
| from | 111 | 2 | 3 |

The average code length:

* Which version of the message, postcard, or telegram, contains more information? Show quantitatively and explain qualitatively where the difference (if any) comes from.

The postcard contains more information because postcard has more words than the telegram. Words from postcard take more space to form a Huffman tree data structure. We can calculate the entropy to demonstrate: