

Universidad del Valle de Guatemala

Visión por computadora

Joaquín Puente 22296

Nelson García Bravatti 22434

Diego Linares 221256

Hoja de trabajo 1

Link del repo:

https://github.com/ItsJapo/VisionXComputadora/tree/main/HDT_1

Task1 - Análisis Teórico

1.

- a. $3840 \times 2160 = 8,294,400 \text{ pixeles} \times 12 = 99,532,800 \text{ bits} / 8 = 12,441,600 \text{ Bytes}$
 $/ 1,024 = 12,150 \text{ KB} / 1024 = 11.8652 \text{ MB}$
- b. $11.8652 \text{ MB} * 8 \text{ Cams} * 60 \text{ Frames} = 5,695.313 \text{ MB/s} / 1,024 = 5.5618 \text{ GB/s}$
- c. $16\text{GB} / 5.5618 \text{ GB/s} = 2.876 \text{ segundos}$
- d. No es viable enviar estos datos a la nube via 5G dado que en promedio 5G tiene un ancho de banda entre 100 y 300 MB/s

2.

- a. $L_{in} = 50 \rightarrow 50/255 = 0.1961$
 - I. Corrección Gamma: $(0.1961)^{0.5} = 0.4428$
 - II. Ajuste Lineal: $1.2 \times 0.4428 + \frac{-10}{255} = 0.4922 \times 255 = 125$
- b. No, porque el resultado final siempre estuvo en el dominio permitido de $[0,1]$
- c. $50^{0.5} = 7.07 \Rightarrow 1.2 \cdot 7.07 - 10 = -1.51 \approx 0$
El error numérico habría sido 125 si no se hubiera hecho la conversión a float 32

3.

- a. $d_{RGB} = \sqrt{(255 - 50)^2} = \sqrt{205^2} = 205$
- b. Conversión
 - I. $(255, 0, 0)$

- a. $H = 0$
 - b. $S = (\text{Max-Min})/\text{Max} = 1$
 - c. $V = \text{Max} = 1$
- II. $(50,0,0)$
- a. $H=0$
 - b. $S = (0.196-0)/0.196 = 1$
 - c. $V = 0.196$
- III. $d_{HSV} = \sqrt{(1 - 0.196)^2} = \sqrt{0.804^2} = 0.804$
4. En el espacio RGB la distancia es demasiada, por lo que en un algoritmo de clustering, ambas lecturas quedan en espacios diferentes, pero en un espacio HSV, el algoritmo de clustering únicamente tomará los valores H y S quienes definen el color mientras que el V quien representa la luminosidad se puede aislar para que no sea tomado en consideración por lo que puede tomar en el mismo cluster los colores independientemente de su luminosidad.

Task 2 - Práctica

Ejericicio 1:

Contraste Alto (Manual)



Ejercicio 2:

Corrección Gamma 0.5



Ejercicio 3:

Segmentación HSV



Task 3 -

Responda brevemente (máximo 3 líneas por respuesta):

1. En la diapositiva 15 se mencionó que "Iterar píxel a píxel en Python es un Pecado Capital". Explique en términos de gestión de memoria y CPU por qué una operación vectorizada en NumPy es órdenes de magnitud más rápida que un for loop.

- Numpy usa bloques de memoria contiguos y tipos de datos fijos, lo que hace que la CPU acceda más rápido que los objetos dispersos de Python, también se aprovecha de instrucciones SIMD del CPU los procesa en un solo ciclo de reloj.

2. Al visualizar imágenes con matplotlib, ¿qué sucede si olvida que OpenCV carga las imágenes en formato BGR? ¿Cómo se ve visualmente el error?

Pues Matplotlib espera RGB, así que lo que pasaría es que, si no se convierte de BGR a RGB, la imagen se muestra con los colores intercambiados, especialmente rojo y azul, produciendo una apariencia visual incorrecta pero no distorsionada.

3. Al visualizar imágenes con matplotlib, ¿qué sucede si olvida que OpenCV carga las imágenes en formato BGR? ¿Cómo se ve visualmente el error?

Misma pregunta, misma respuesta.

Pues Matplotlib espera RGB, así que lo que pasaría es que, si no se convierte de BGR a RGB, la imagen se muestra con los colores intercambiados, especialmente rojo y azul, produciendo una apariencia visual incorrecta pero no distorsionada.