

## Problema 5

### Espacio de color HSV

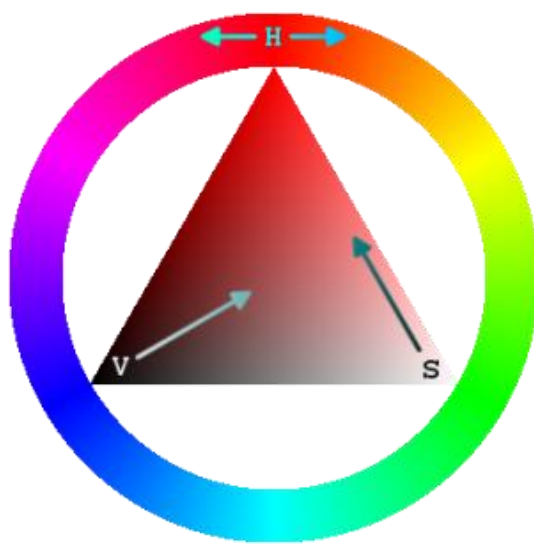
El espacio de color HSV (Hue, Saturation, Value) es un modelo de color que se utiliza para representar colores de manera más intuitiva que el tradicional modelo RGB (Red, Green, Blue). En el espacio de color HSV, los colores se describen mediante tres componentes:

1. **Matiz (Hue):** Representa el tipo de color, como rojo, verde, azul, etc. Se representa como un ángulo en un círculo de colores, donde  $0^\circ$  y  $360^\circ$  corresponden a rojo puro,  $120^\circ$  a verde puro y  $240^\circ$  a azul puro.
2. **Saturación (Saturation):** Indica la pureza o intensidad del color. Una saturación alta significa colores más vivos y puros, mientras que una saturación baja se acerca a tonos de gris.
3. **Valor (Value):** Representa la intensidad o brillo del color. Un valor alto indica colores más brillantes, mientras que un valor bajo representa colores más oscuros.

El espacio de color HSV es útil en aplicaciones de procesamiento de imágenes y visión por computadora, ya que permite separar la información sobre el tipo de color (matiz) de la información sobre la intensidad (saturación y valor). Esto lo hace especialmente útil en aplicaciones como segmentación de objetos por color, ajuste de brillo y contraste, y detección de colores específicos en una imagen.

El mapeo de colores al espacio de color HSV se realiza a través de una transformación matemática que toma las componentes de color (R, G, B) del espacio de color RGB y las convierte a sus equivalentes en el espacio HSV (H, S, V). El proceso de conversión varía dependiendo del software o la librería que se utilice, pero en general, se emplean fórmulas matemáticas específicas para calcular los valores de H, S y V a partir de R, G y B.

En Python, es común utilizar librerías como OpenCV para manipular imágenes y realizar conversiones entre diferentes espacios de color, incluyendo RGB a HSV y viceversa.



**PASO 2:  
TRANSFORMAR  
DE BGR A HSV**

Cuando una imagen es leída en OpenCV, por defecto la lee en BGR, por ello en este paso necesita ser transformada al espacio de color HSV. Para ello se emplea la siguiente línea:

`imagenHSV = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)`

Imagen BGR → Transformación BGR a HSV → Imagen resultante

**¿Por qué HSV?**

HSV, es un espacio de color cuyos componentes son: Matiz, Saturación y Brillo (Hue, Saturation and Value). Se usa este espacio de color debido a que es más sencillo determinar el o los colores a detectar, para ello se emplea especial atención al componente H, que corresponde al matiz. En OpenCV sus componentes pueden tomar los siguientes valores:

- H: 0 a 179
- S: 0 a 255
- V: 0 a 255



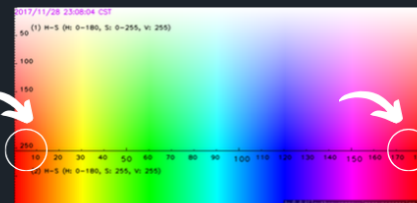
## DETECCIÓN DE COLORES EN OPENCV - PYTHON (EN 4 PASOS)

Otra forma de encontrar los rangos del color que se desea detectar es, tomar una porción de la imagen de entrada en donde se encuentre dicho color. Luego podrás determinar los rangos.

**PASO 3:  
DETERMINAR LOS  
RANGOS EN  
DONDE SE  
ENCUENTRA EL  
COLOR A  
DETECTAR**

Se necesita construir dos arrays en donde esté presente el límite inicial y final en HSV del color a detectar. Se busca en H donde podría estar ubicado dicho color, mientras que para S y V se pueden usar rangos de entre 100 a 255 y de 20 a 255 respectivamente.

Si se deseara detectar el color rojo, en H se encuentra en dos lugares, al principio y al final. Entonces se debe establecer entre que valores se encuentra, por ejemplo de 0 a 8, y de 175 a 179.



Para determinar los rangos, en OpenCV se usa:

`imagenBinaria = cv2.inRange(imagenHSV, limiteIni, limiteFin)`

Imagen HSV → Límite inicial → Límite final → Imagen resultante

Si se necesitara sumar de dos imágenes (como en el caso del color rojo en el cual se obtendrán 2 imágenes binarias):

`imagenResult = cv2.add(imagen1, imagen2)`

Imagen 1 → Imagen 2 → Imagen resultante