



Centro de Enseñanza Técnica Industrial

Diego Sebastián Becerril Vizcaya

22310275

6°G

Practica 6

Visión artificial

Mtro. Mauricio Alejandro Cabrera Arellano

18 de mayo de 2025

El modelo RGB (Red, Green, Blue) es el más común para dispositivos de visualización como monitores, televisores y cámaras. Permite representar una gama extensa de colores mediante mezcla aditiva de los tres colores base.

Por su parte, el modelo HSV (Hue, Saturation, Value) surge como una representación alternativa que facilita la manipulación del color desde un punto de vista perceptual. Se describe como un superconjunto del RGB, donde:

- Hue (Matiz): identifica el color (ángulo en un círculo cromático)
- Saturation (Saturación): pureza del color
- Value (Brillo): intensidad o luminosidad

También se menciona la existencia de otros espacios como YUV, más utilizados en compresión de video y televisión digital, donde:

- Y representa la luminancia (brillo)
- U y V representan la crominancia (información de color)

Estos modelos mejoran el aislamiento del color, ya que permiten separarlo de la iluminación o intensidad.

Aplicaciones prácticas con OpenCV (Python)

Convertir entre modelos de color

RGB a HSV

```
hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

RGB a YUV

```
yuv_img = cv2.cvtColor(img, cv2.COLOR_BGR2YUV)
```

Detección de colores específicos (azul)

Definir rango para azul en HSV

```
lower_blue = np.array([100, 150, 50])
```

```
upper_blue = np.array([140, 255, 255])
```

Crear máscara y aplicarla

```
mask = cv2.inRange(hsv_img, lower_blue, upper_blue)
```

```
res = cv2.bitwise_and(img, img, mask=mask)
```

Esto aísla los píxeles que caen dentro del rango azul.

Eliminación de un color (ej. eliminar rojo)

Rango para rojo en HSV (dos rangos por envolvimiento del círculo)

```
lower_red1 = np.array([0, 120, 70])
```

```
upper_red1 = np.array([10, 255, 255])
```

```
lower_red2 = np.array([170, 120, 70])
```

```
upper_red2 = np.array([180, 255, 255])
```

Crear máscaras y combinarlas

```
mask1 = cv2.inRange(hsv_img, lower_red1, upper_red1)
```

```
mask2 = cv2.inRange(hsv_img, lower_red2, upper_red2)
```

```
full_mask = mask1 + mask2
```

Invertir la máscara y aplicar

```
mask_inv = cv2.bitwise_not(full_mask)
```

```
filtered = cv2.bitwise_and(img, img, mask=mask_inv)
```

Aplicación en video en tiempo real

```
cap = cv2.VideoCapture(0)
```

```
while True:
```

```
    ret, frame = cap.read()
```

```
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

```
    mask = cv2.inRange(hsv, lower_blue, upper_blue)
```

```

res = cv2.bitwise_and(frame, frame, mask=mask)
cv2.imshow('Detectado', res)
if cv2.waitKey(1) & 0xFF == 27:
    break
cap.release()
cv2.destroyAllWindows()

```

Ventajas de usar HSV o YUV sobre RGB

Característica	RGB HSV		YUV
Intuición humana del color	Baja	Alta (matiz saturación)	y Media
Separación brillo-color	No	Sí	Sí
Ideal para segmentación	No	Sí	Sí, especialmente en video
Robusto a luz variable	No	Sí	Sí