



Universidad Veracruzana

Maestría en Inteligencia Artificial

Visión por Computadora

**Tarea 9. Segmentación de colores con espacios
HSV y LAB MATLAB.**

Ángel García Báez

Profesor: Dr. Héctor Acosta Mesa

May 6, 2025

Contents

1	Objetivo de la práctica	2
2	Metodología	3
3	Resultados	7
3.1	Segmentación en HSV con un umbral de 0.15	7
3.2	Segmentación en HSV con un umbral de 0.25	8
3.3	Segmentación en LAB con un umbral de 15	9
3.4	Segmentación en LAB con un umbral de 25	10
3.5	Comparativa de la mascara resultante de los 4 experimentos .	11
4	Conclusiones	12
5	Referencias	13
6	Anexos	14
6.1	Implementación de la segmentación para imagenes en HSV y LAB en MATLAB	14

1 Objetivo de la práctica

Para la presente practica, se desea poder implementar una función propia que tome una imagen y permita realizar la segmentación de los pixeles de la misma mediante una transformación del espacio de colores tradicional RGB a otros espacios como HSV y LAB para realizar dicha segmentación basandose unicamente en los canales de colores.

Para ello, la función debe permitir seleccionar cualquier pixel de la imagen para tomarlo como referencia para realizar la segmentación en busca de similitudes mediante un umbral fijado de la distancia de los pixeles de la imagen al pixel de referencia seleccionado.

Para llevar a cabo la practica se aplicara el proceso de segmentación sobre la siguiente imagen de la cantante Denisse Guerrero de Belanova, extraída de su cuenta de instagram oficial:



Figure 1: Foto de Denisse Guerrero.

2 Metodología

De acuerdo con Gonzalez and Woods (2018), un espacio de color es un modelo matemático que describe la representación de los colores en un formato que puede ser interpretado y procesado por sistemas computacionales o visualizado por humanos. Cada color dentro de un espacio está definido por un conjunto de coordenadas, normalmente en tres dimensiones, como en los espacios RGB, HSV o CIE L*a*b*. La elección del espacio de color afecta directamente la facilidad con que ciertas tareas de procesamiento de imagen, como la segmentación o el reconocimiento de objetos, pueden realizarse.

El espacio HSV (Hue, Saturation, Value) es un espacio de color perceptual diseñado para ser más intuitivo para los humanos que el modelo RGB. Está compuesto por tres componentes:

- Hue (H): tonalidad del color, representada como un ángulo en el círculo cromático, en grados (0° – 360°).
- Saturation (S): pureza del color, que varía de 0 (gris) a 1 (color completamente puro).
- Value (V): brillo del color, que varía de 0 (negro) a 1 (máxima luminosidad).

Este modelo es útil en aplicaciones como la segmentación por color, ya que permite separar el matiz de la intensidad de luz, facilitando tareas de procesamiento basadas en la percepción humana.

El espacio de color HSV (Hue, Saturation, Value) permite representar colores de forma más intuitiva que el espacio RGB, ya que separa las propiedades cromáticas del color (tono y saturación) de su brillo. Para convertir un color del espacio RGB al HSV, primero se normalizan los valores $R, G, B \in [0, 1]$, y luego se calculan:

$$C_{\max} = \max(R, G, B), \quad C_{\min} = \min(R, G, B), \quad \Delta = C_{\max} - C_{\min}$$

El matiz H , medido en grados sobre el círculo cromático, se define como:

$$H = \begin{cases} 0^\circ, & \text{si } \Delta = 0 \\ 60^\circ \cdot \left(\frac{G-B}{\Delta} \bmod 6 \right), & \text{si } C_{\max} = R \\ 60^\circ \cdot \left(\frac{B-R}{\Delta} + 2 \right), & \text{si } C_{\max} = G \\ 60^\circ \cdot \left(\frac{R-G}{\Delta} + 4 \right), & \text{si } C_{\max} = B \end{cases}$$

La saturación S , que representa la intensidad o pureza del color, se calcula como:

$$S = \begin{cases} 0, & \text{si } C_{\max} = 0 \\ \frac{\Delta}{C_{\max}}, & \text{en otro caso} \end{cases}$$

El valor V , que representa el brillo del color, es simplemente:

$$V = C_{\max}$$

Este modelo resulta particularmente útil en procesamiento de imágenes y visión por computadora, especialmente en tareas como segmentación basada en color.

- Primero se obtuvo el gráfico de los 1520 conjuntos de puntos correspondientes a sus imágenes y se graficaron.
- Segundo, se hizo el registro de cada uno de los 1520 conjuntos de puntos respecto a la imagen 1 y se graficaron.
- Tercero, se obtuvo el ERM de cada conjunto de puntos transformado respecto al conjunto de la imagen 1 para evaluar que tan bueno fue el ajuste. Se hizo el histograma de la distribución de los errores cuadráticos medios y se ordenaron de menor a mayor.
- Finalmente, se obtuvieron únicamente los 100 conjuntos de puntos transformados que presentaron el error cuadrático más pequeño y se graficaron.

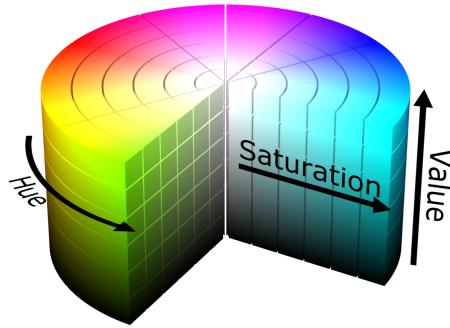


Figure 2: Representación del espacio HSV.

Por otro lado, el espacio de color CIE Lab es un modelo perceptualmente uniforme propuesto por la Comisión Internacional de Iluminación (CIE) en 1976. Está diseñado para representar los colores de forma que la distancia euclidea entre dos colores se aproxime a la percepción humana de la diferencia entre ellos. El modelo se basa en un espacio tridimensional donde:

- L^* representa la luminosidad (de 0 a 100).
- a^* representa la posición entre verde (valores negativos) y rojo (valores positivos).
- b^* representa la posición entre azul (valores negativos) y amarillo (valores positivos).

La representación de dicho espacio se puede observar como una esfera en 3 dimensiones:

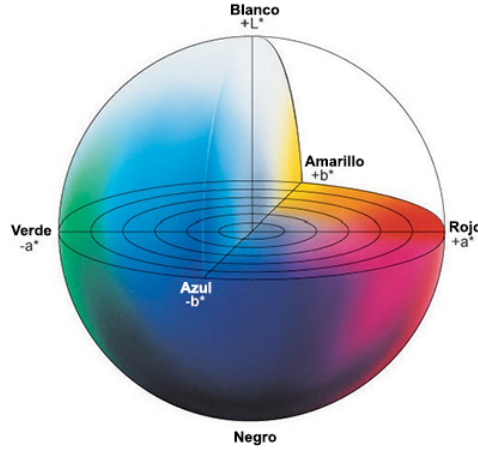


Figure 3: Representación del espacio LAB.

Para convertir de RGB a CIE Lab, primero se convierte RGB a XYZ (espacio de referencia absoluto), y luego XYZ a Lab. Los pasos son:

1. De RGB a XYZ: Primero se normaliza $R, G, B \in [0, 1]$ (si estaban en el rango $[0, 255]$), y se linealizan con:

$$C_{\text{lin}} = \begin{cases} \frac{C}{12.92}, & \text{si } C \leq 0.04045 \\ \left(\frac{C+0.055}{1.055} \right)^{2.4}, & \text{si } C > 0.04045 \end{cases}$$

Luego, se calculan los valores XYZ usando la matriz de transformación para el espacio sRGB con iluminante D65:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \cdot \begin{bmatrix} R_{\text{lin}} \\ G_{\text{lin}} \\ B_{\text{lin}} \end{bmatrix}$$

2. De XYZ a CIE Lab Se normaliza respecto al blanco de referencia (X_n, Y_n, Z_n) , y se define la función intermedia:

$$f(t) = \begin{cases} t^{1/3}, & \text{si } t > \delta \\ \frac{t}{3\delta^2} + \frac{4}{29}, & \text{si } t \leq \delta \end{cases}, \quad \text{donde } \delta = \left(\frac{6}{29}\right)^3$$

Luego, los valores Lab se calculan como:

$$\begin{aligned} L^* &= 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500 \cdot \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \\ b^* &= 200 \cdot \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \end{aligned}$$

Este espacio de color es muy usado en visión computacional y procesamiento de imágenes debido a su uniformidad perceptual.

Una vez definidos los conceptos necesarios, la metodología aplicada para la practica en lenguaje MATLAB consiste en lo siguiente:

- Cargar la imagen en RGB.
- Seleccionar el espacio de color a donde va a ser mapeada la imagen original (HSV).
- Seleccionar un pixel de referencia de la imagen.
- Definir un umbral que permita marcar cuando un pixel de la imagen transformada se "parezca" o este cerca al pixel de referencia con respecto a la distancia euclidiana.
- Repetir el proceso de revisar con cada uno de los pixeles de la matriz hasta obtener la mascara de la imagen completa de los pixeles que se parecen al pixel de referencia
- Repetir en el espacio LAB

Nota: Se uso el mismo pixel de referencia para todas las segmentaciones.

3 Resultados

3.1 Segmentación en HSV con un umbral de 0.15



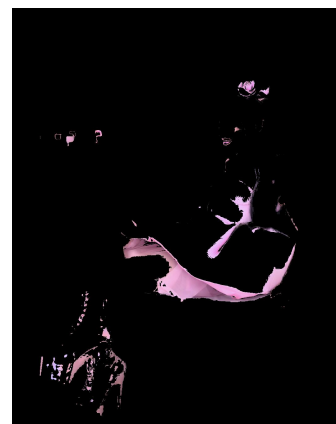
Imagen original.



Pixel seleccionado para la segmentación.



Mascara seleccionada.



Mascara aplicada sobre la imagen original.

Figure 4: Resultados de segmentar en HSV con umbral de 0.15.

Los resultados para la segmentación con HSV y un umbral de 0.15, muestra solo aquellos pixeles que están muy parecidos a ese rosa intenso, pero aquellos que son un rosa más pálido por la iluminación como el que se muestra por la zona del abdomen

3.2 Segmentación en HSV con un umbral de 0.25



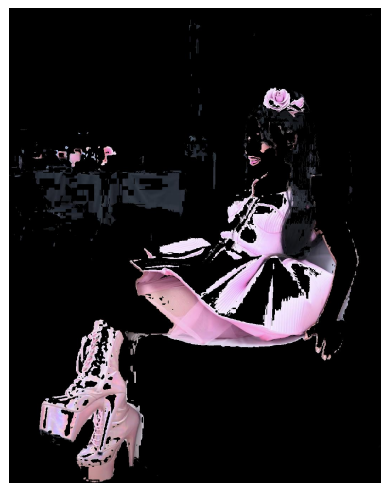
Imagen original.



Pixel seleccionado para la segmentación.



Mascara seleccionada.



Mascara aplicada sobre la imagen original.

Figure 5: Resultados de segmentar en HSV con umbral de 0.25.

Los resultados de aumentar el umbral a 0.25 muestran con el mismo pixel de referencia, que detecta más tonalidades de rosa aunque sean más palidos por la luz, como lo es alrededor del vestido, los labios y las mejillas.

3.3 Segmentación en LAB con un umbral de 15



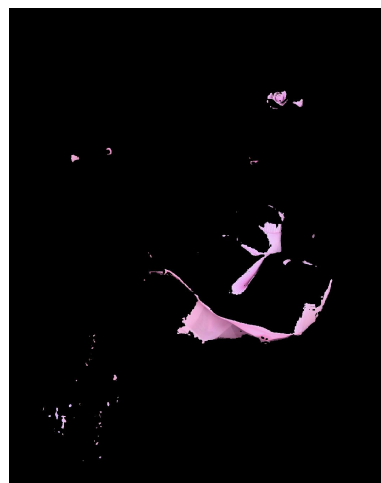
Imagen original.



Pixel seleccionado para la segmentación.



Mascara seleccionada.



Mascara aplicada sobre la imagen original.

Figure 6: Resultados de segmentar en LAB con umbral de 15.

Al cambiar a el espacio LAB, con un umbral de 15 unidades, los resultados son muy similares a los obtenidos en HSV con umbral de 0.15, solo que aquí detecta zonas más concretas, donde el rosa es más intenso.

3.4 Segmentación en LAB con un umbral de 25



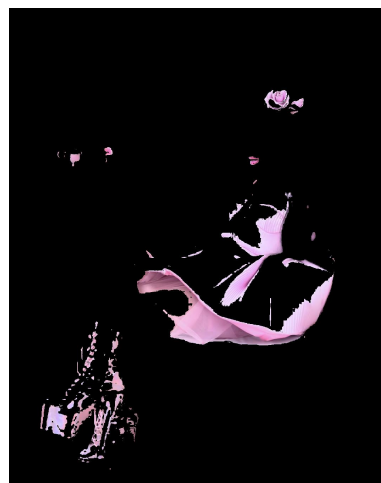
Imagen original.



Pixel seleccionado para la segmentación.



Mascara seleccionada.

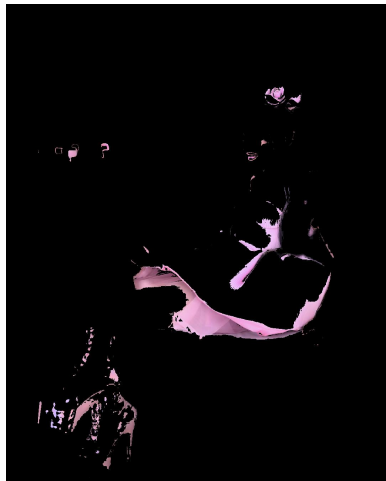


Mascara aplicada sobre la imagen original.

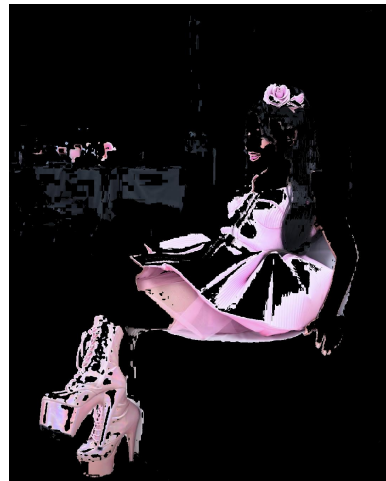
Figure 7: Resultados de segmentar en LAB con umbral de 25.

Al aumentar el umbral a 25 unidades, se nota encontrar como ahora encuentra más puntos en la imagen donde el rosa es más intenso, como en las botas, en los labios, el moño del cabello y elementos de la mesa.

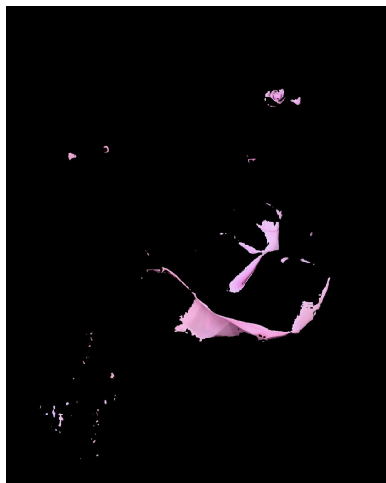
3.5 Comparativa de la mascara resultante de los 4 experimentos



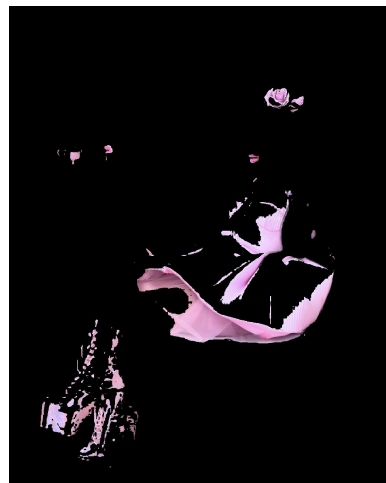
HSV con 0.15.



HSV con 0.25.



LAB con 15.



LAB con 25.

Figure 8: Comparativa de las mascaras resultantes.

Al mostrar la comparativa de las 4 mascaras obtenidas, salta a la vista que HSV con 0.25 de umbral encuentra más pixeles parecidos al rosa que LAB con 25.

4 Conclusiones

Tras la implementación del algoritmo de segmentación y hacer el recorrido entre los espacios de color HSV y LAB, se llega a la conclusión de que son alternativas bastante potentes, al lograr esta separación de los canales de colores del brillo de la imagen, hecho que se ve reflejado en las mascararas obtenidas. Mientras que HSV logra detectar un mayor rango de pixeles dentro de la tonalidad de referencia, LAB encuentra conjuntos de pixeles más concretos respecto a su referencia.

5 Referencias

References

Gonzalez, R. C. and Woods, R. E. (2018). *Digital Image Processing*. Pearson.

6 Anexos

6.1 Implementación de la segmentación para imágenes en HSV y LAB en MATLAB

```
1  %% Distancia Euclideana %%
2  function d = distanciaE(x, y)
3  d = sqrt(sum((x - y).^2));
4  end
5
6  %% Función que transforma entre espacios
7  function [pixel, mascara, imgres] = SegEspacios(img, espacio, umbral)
8  %% Mostrar imagen y seleccionar pixel %%
9  fig = figure;
10 imshow(img);
11 title('Haz zoom (botón de lupa) y presiona ENTER cuando estés listo');
12 zoom on;
13 pause;
14 zoom off;
15 title('Ahora selecciona un pixel con un clic');
16 [col, fila] = ginput(1);
17 pixel = img(round(fila), round(col), :);
18 close(fig);
19 disp(['Coordenadas del pixel: x=', num2str(round(col)), ', y=', num2str(round(fila))]);
20 disp(['Color RGB: ', num2str(squeeze(pixel))]);
21
22 %% Conversión al espacio de color hsv o lab %%
23 switch lower(espacio)
24 % CASO PARA HSV
25 case 'hsv'
26 img_convertido = rgb2hsv(img);
27 pixel_ref = rgb2hsv(pixel);
28 ref_vec = [pixel_ref(:,:1), pixel_ref(:,:2)];
29 A = img_convertido(:,:1);
30 B = img_convertido(:,:2);
31 % CASO PARA LAB
32 case 'lab'
33 img_convertido = rgb2lab(img);
34 pixel_ref = rgb2lab(pixel);
35 ref_vec = [pixel_ref(:,:2), pixel_ref(:,:3)];
36 A = img_convertido(:,:2);
```

```

37 B = img_convertido(:,:,3);
38 otherwise
39 error('Espacio de color no válido. Usa "hsv" o "lab".');
40 end
41
42 %% Crear máscara basada en distancia euclidea %%
43 [n, p, ~] = size(img); % Obtener las dimensiones
44 mascara = zeros(n, p); % Crear la mascara vacia
45 % Identificar pixel a pixel si cumple o no el umbral
46 for i = 1:n
47     for j = 1:p
48         color_vec = [A(i,j), B(i,j)];
49         if distanciaE(color_vec, ref_vec) <= umbral
50             mascara(i,j) = 1;
51         end
52     end
53 end
54
55 %% Aplicar la máscara a la imagen %%
56 imgres = img;
57 imgres(repmat(~mascara, [1 1 3])) = 0;
58
59 %% Mostrar resultados %%
60 figure;
61 subplot(2,2,1); imshow(img); title('Imagen original');
62 subplot(2,2,2); imshow(pixel); title('Pixel seleccionado');
63 subplot(2,2,3); imshow(mascara); title('Máscara generada');
64 subplot(2,2,4); imshow(imgres); title('Imagen segmentada');
65 end
66
67
68 %% Cargar la imagen %%
69 ruta = "REPORTE\IMG\IM4.jpeg";
70 img = imread(ruta);
71
72 %% Pruebas para HSV color rosa umbral 0.15 %%
73 [pixel, mascara, imgres] = SegEspacios(img, "hsv", 0.15);
74 imwrite(pixel, "R11.jpg")
75 imwrite(mascara, "R12.jpg")
76 imwrite(imgres, "R13.jpg")

```



```

77  %% Pruebas para HSV color rosa umbral 0.25 %%
78  [pixel, mascara, imgres] = SegEspacios(img, "hsv", 0.25);
79  imwrite(pixel, "R21.jpg")
80  imwrite(mascara, "R22.jpg")
81  imwrite(imgres, "R23.jpg")
82  %% Pruebas para LAB color rosa umbral 15 %%
83  [pixel, mascara, imgres] = SegEspacios(img, "lab", 15);
84  imwrite(pixel, "R31.jpg")
85  imwrite(mascara, "R32.jpg")
86  imwrite(imgres, "R33.jpg")
87  %% Pruebas para LAB color rosa umbral 25 %%
88  [pixel, mascara, imgres] = SegEspacios(img, "lab", 25);
89  imwrite(pixel, "R41.jpg")
90  imwrite(mascara, "R42.jpg")
91  imwrite(imgres, "R43.jpg")

```