

*Pontificia Universidad Javeriana*

*Procesamiento de imágenes y visión*

*Taller 4*

*Leidy Carolina Pulido Feo y Eliana Andrea Romero Leon*

## 1. Segmentación por color

Para la implementación de este punto, se debieron realizar 2 métodos los cuales se llaman k-means y gaussian mixture model. Teniendo dichos métodos se pudo realizar y visualizar la suma de las distancias intra-cluster.

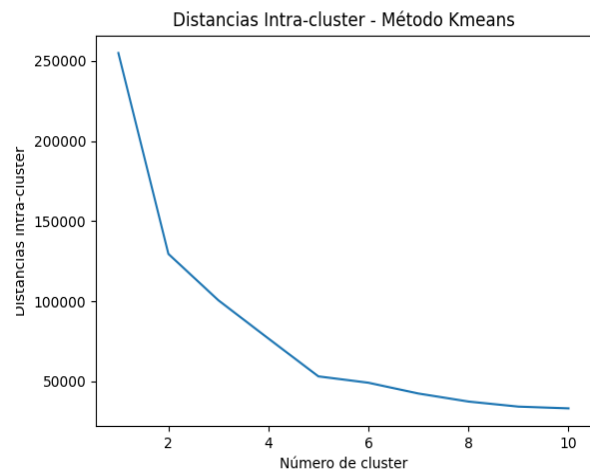


*Figura 1. Imagen original de la bandera.*

Quantized image (10 colors, method=Kmeans)



*Figura 2. Imagen con método kmeans.*



*Figura 3. Gráfico suma de distancias intra-cluster vs n\_color con kmeans.*



Figura 4. Imagen con método gmm.

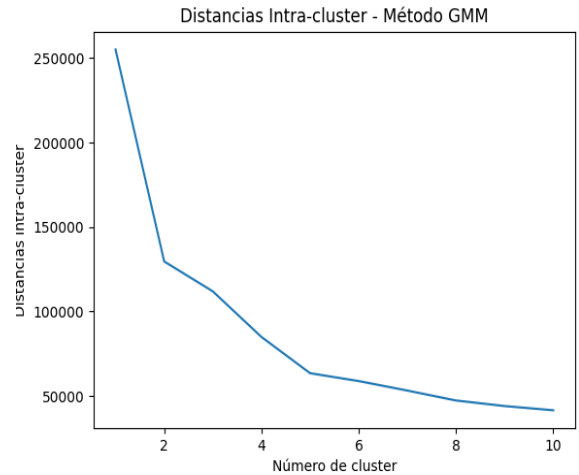


Figura 5. Gráfico suma de distancias intra-cluster vs  $n_{color}$  con gmm.

## 2. Transformaciones geométricas

Para el desarrollo de este punto se visualizó la imagen 'lena' y se escogieron 3 puntos, los cuales fueron los ojos y la nariz. Haciendo click derecho en dichos puntos se pudo obtener la posición de los píxeles. Adicionalmente se realizó el mismo procedimiento con la imagen 'lena\_warped'.

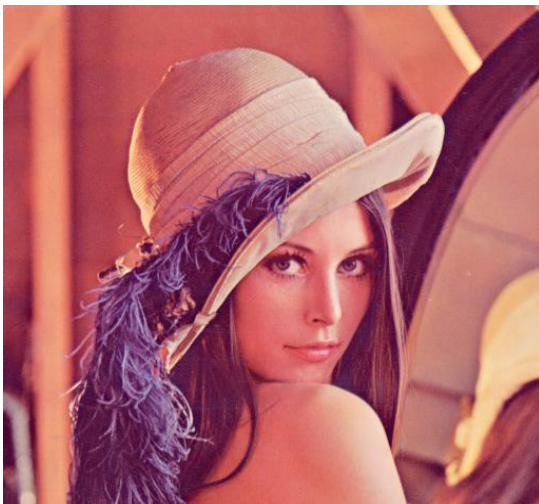


Figura 6. Imagen lena original.

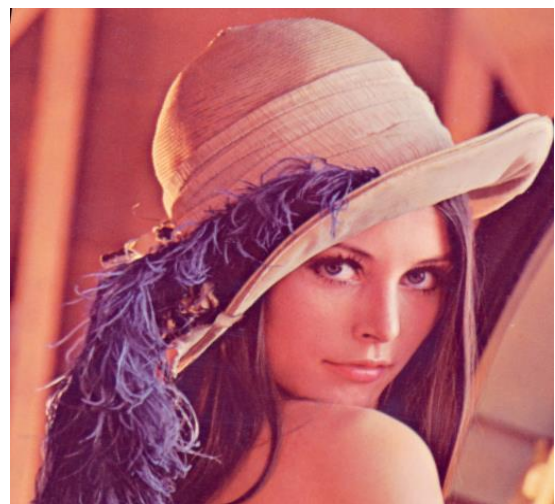


Figura 7. Imagen lena warped.

`[267, 267], [326, 267], [314, 315]`

`[303, 262], [383, 273], [353, 328]`

Después de obtener los puntos se procedió a realizar la matriz afín teniendo como puntos de referencia los seleccionados anteriormente. Adicionalmente se visualizó la imagen obtenida con dicha transformación.



Figura 8. Imagen con transformación afín.

```
[[ 1.27419355 -0.13709677 1.39516129]
 [ 0.11290323 1.17569124 -74.59043779]]
```

Con el fin de realizar una aproximación de la matriz similitud con la matriz afín obtenida anteriormente, se procede a calcular los parámetros de la matriz RST o matriz H de la siguiente manera:

$$RST = \begin{bmatrix} s_0 \cos \theta & s_1 \sin \theta & s_0 x_0 \cos \theta + s_1 x_1 \sin \theta \\ -s_0 \sin \theta & s_1 \cos \theta & s_1 x_1 \cos \theta - s_0 x_0 \sin \theta \end{bmatrix}$$

Donde los parámetros están dados por las siguientes ecuaciones:

$$\begin{aligned} s_0 &= \sqrt{H_{1,1}^2 + H_{2,1}^2} \\ s_1 &= \sqrt{H_{1,2}^2 + H_{2,2}^2} \\ \theta &= -\arctan(H_{2,1}, H_{1,1}) \\ x_0 &= (H_{1,3} \cos \theta - H_{2,3} \sin \theta) / s_0 \\ x_1 &= (H_{1,3} \sin \theta + H_{2,3} \cos \theta) / s_1 \end{aligned}$$

Obteniendo finalmente los parámetros deseados en PyCharm como se muestra a continuación:

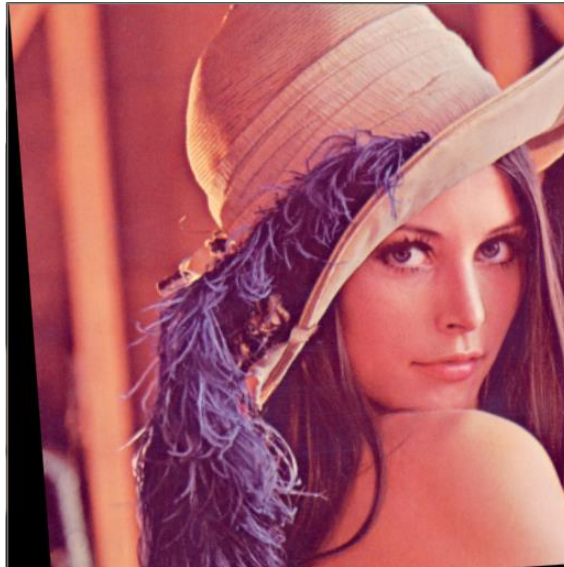
```
#COMPUTO DE PARÁMETROS

#Parámetros de escala
sx = np.sqrt((M_affine[0, 0]**2) + (M_affine[1, 0]**2))
sy = np.sqrt((M_affine[0, 1]**2) + (M_affine[1, 1]**2))

#Parámetro de rotación
theta = -np.arctan(M_affine[1, 0]/M_affine[0, 0])

#Parámetro de traslación
tx = ((M_affine[0, 2] * np.cos(theta)) - (M_affine[1, 2] * np.sin(theta))) / sx
ty = ((M_affine[0, 2] * np.sin(theta)) + (M_affine[1, 2] * np.cos(theta))) / sy
```

Para llevar a cabo la transformación de similitud de la imagen 'lena' se tuvo en cuenta los parámetros de escala, rotación y traslación hallados anteriormente, como se visualiza en la Figura 9.



*Figura 9. Imagen con transformación de similitud.*

```
[[ 1.2741935  0.08826179 -4.0602145 ]  
 [ -0.08826179  1.1790382 -62.875   ]]
```

Finalmente, se calculó el error entre los puntos que se obtienen de la transformación de similitud correspondiente a la imagen 'lena' con los puntos de la imagen 'lena\_warped', esto se puede observar a continuación:

```
Error = [67.95642917 76.1186494 81.37652917]
```