Universidad Popular del Cesar

Tratamiento de Señales

Version 2024-I

Segmentación con Imágenes a Color

[Capítulo 2]

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

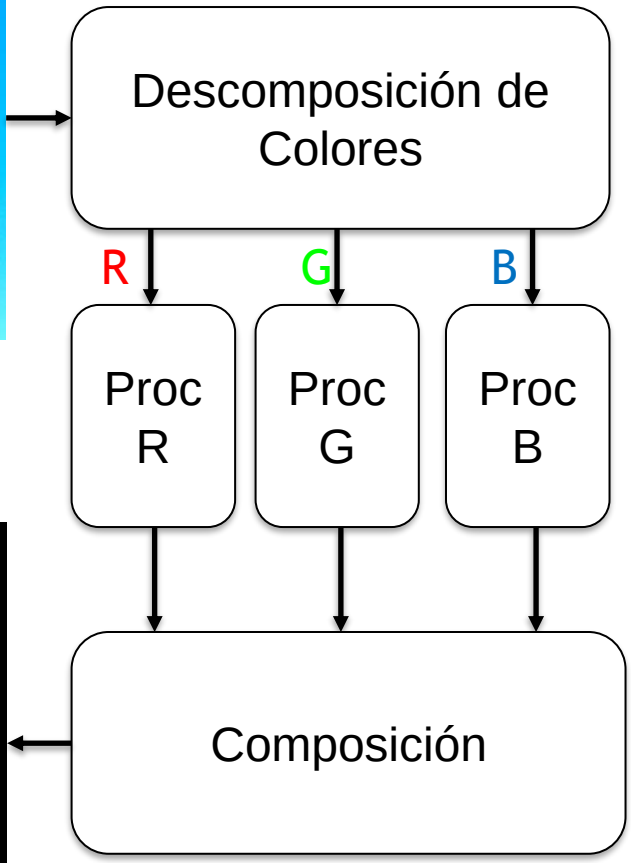
Tres métodos fundamentales:

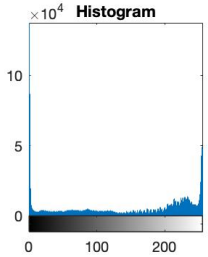
1. Por canal de color
2. Adaptivo por umbrales
3. Clustering

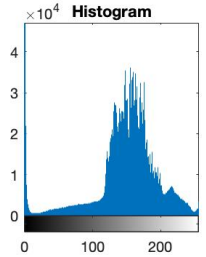
Tres métodos fundamentales:

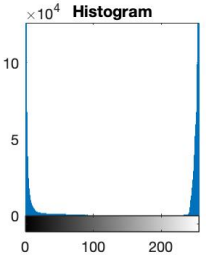
1. Por canal de color
2. Adaptivo por umbrales
3. Clustering

1. Por canal de color



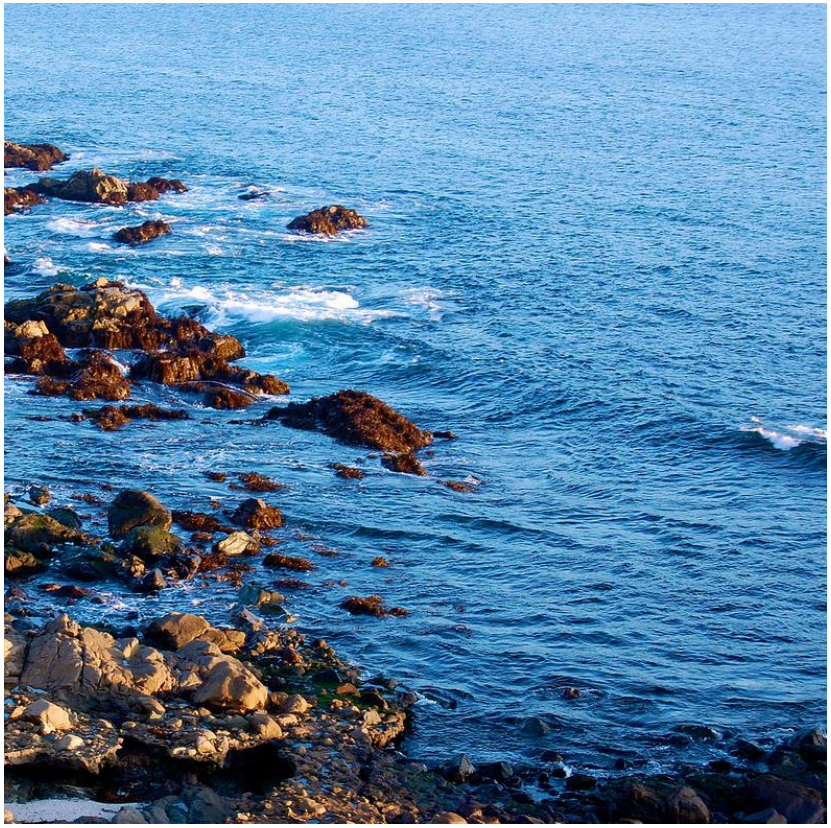


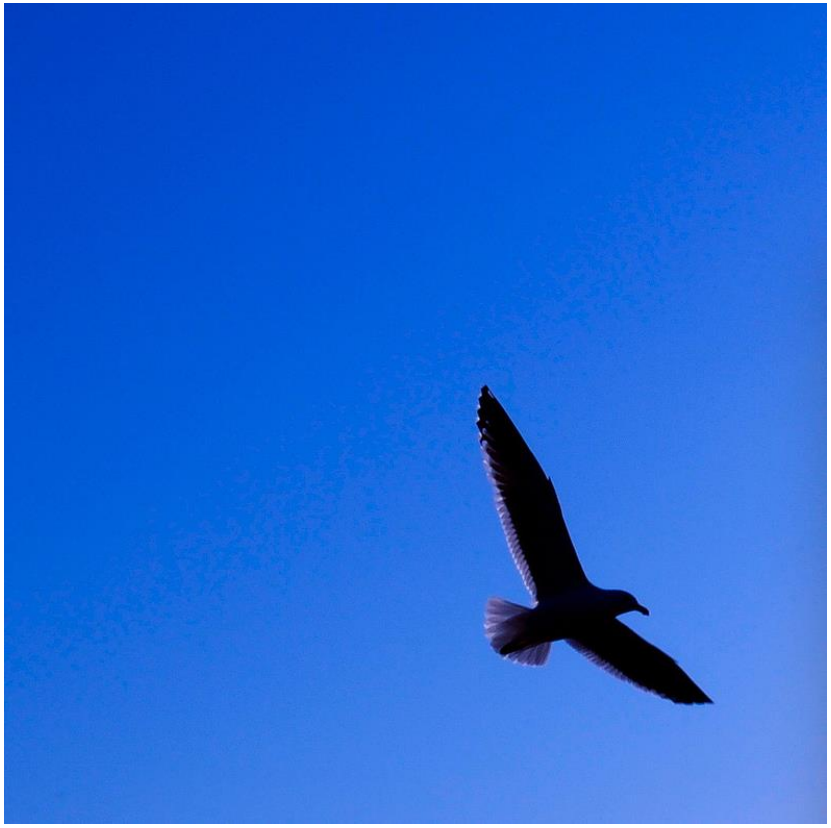


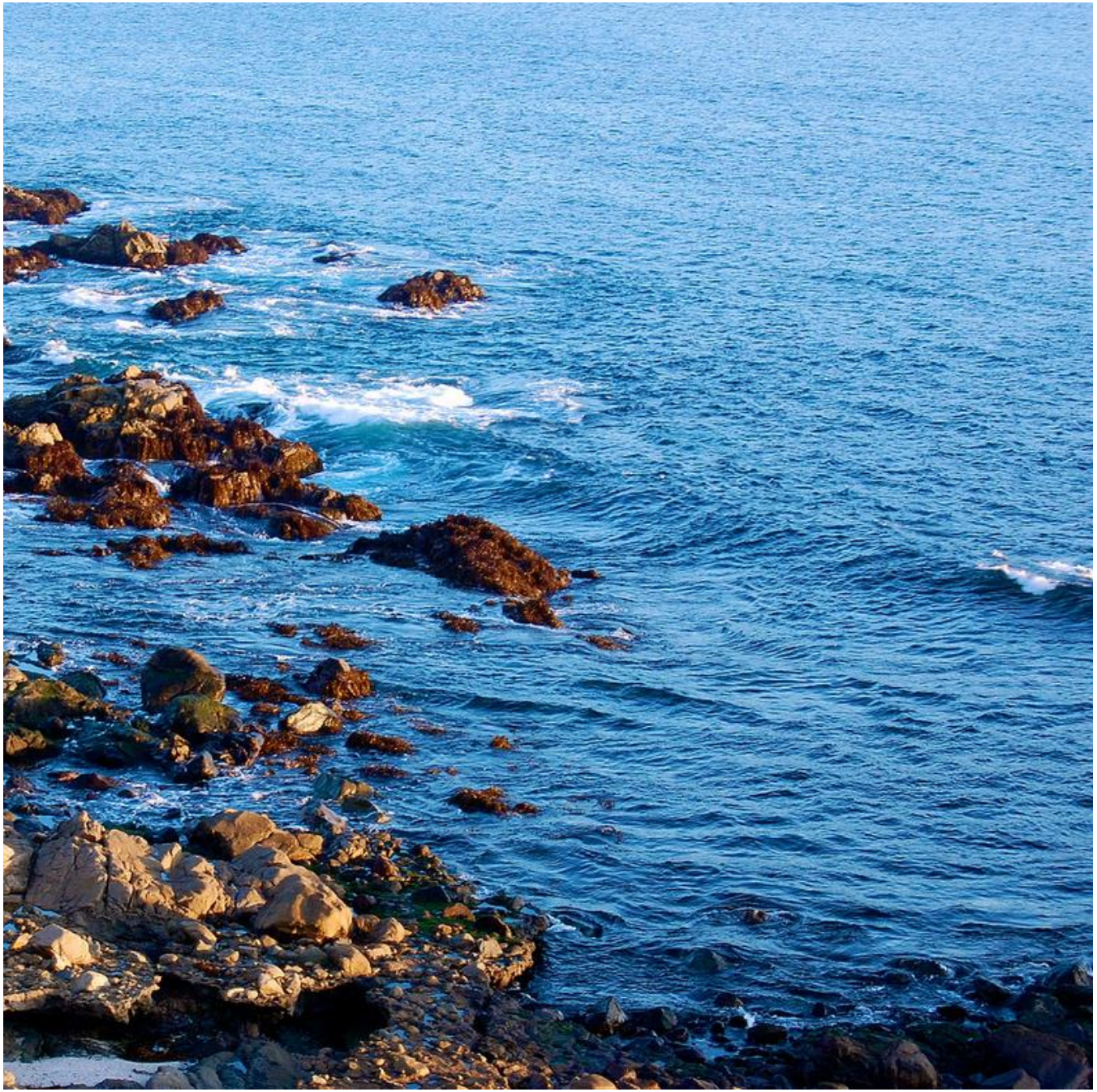


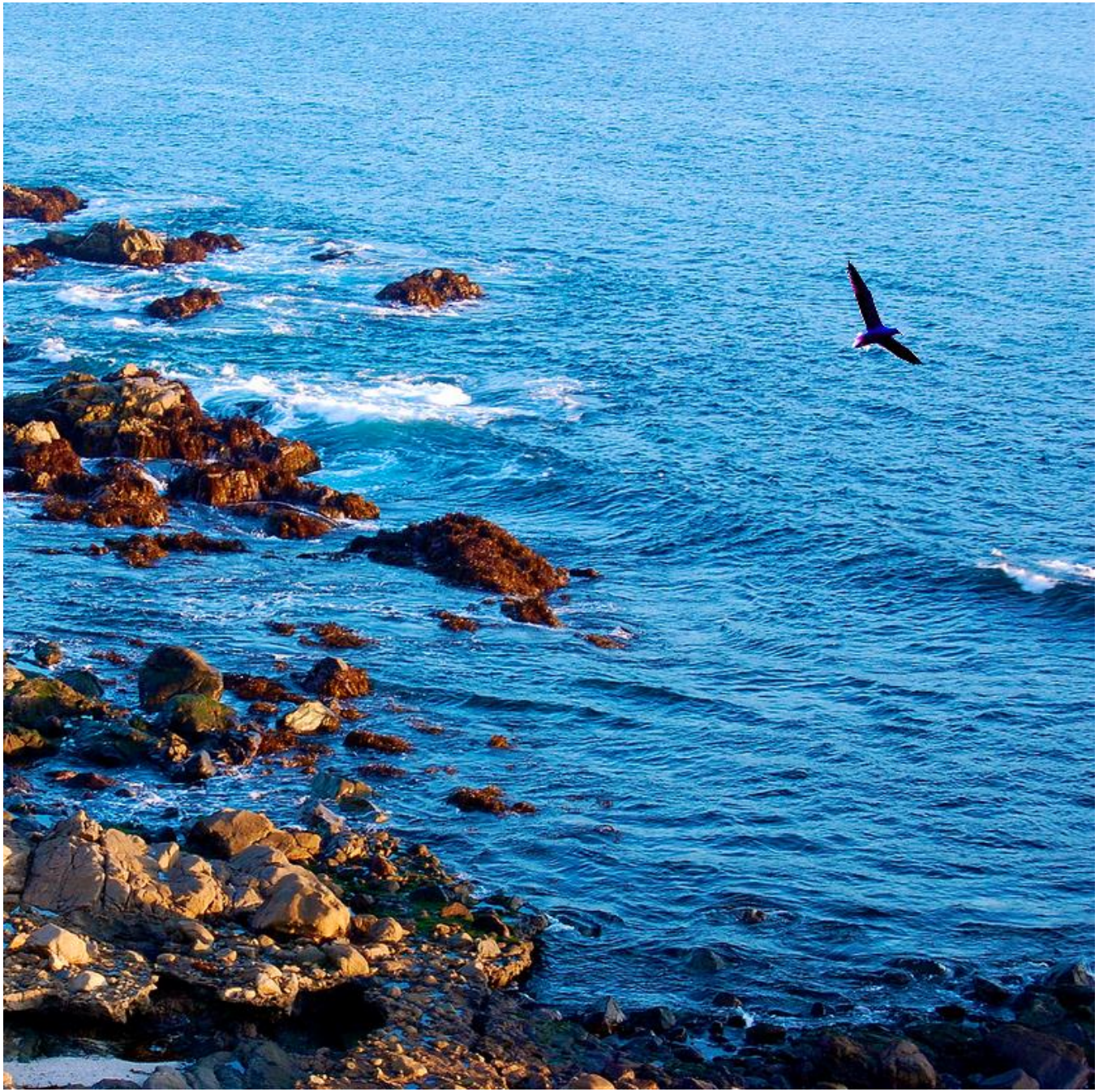

Flor Roja

Ejemplos Similares









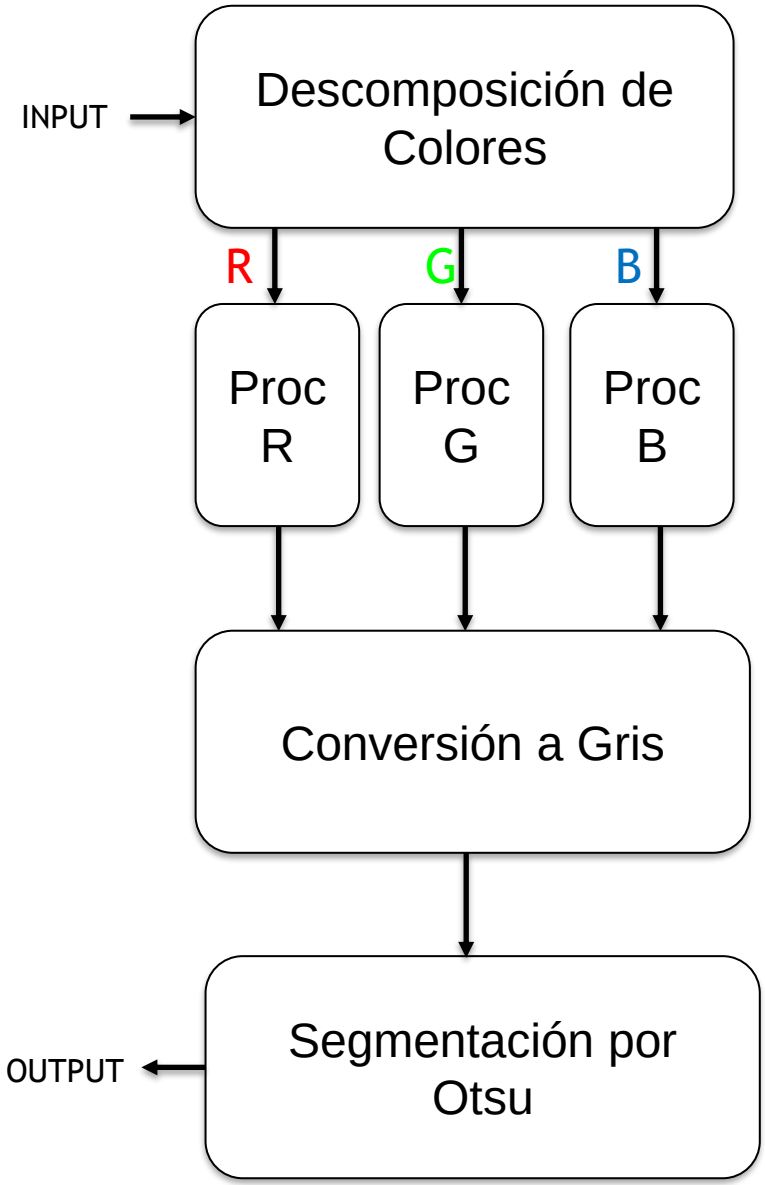
Otros Ejemplos


SCHLAGER
SCHLAGER

Tres métodos fundamentales:

1. Por canal de color
2. Adaptivo por umbrales
3. Clustering

2. Adaptivo por umbral (idea inicial)

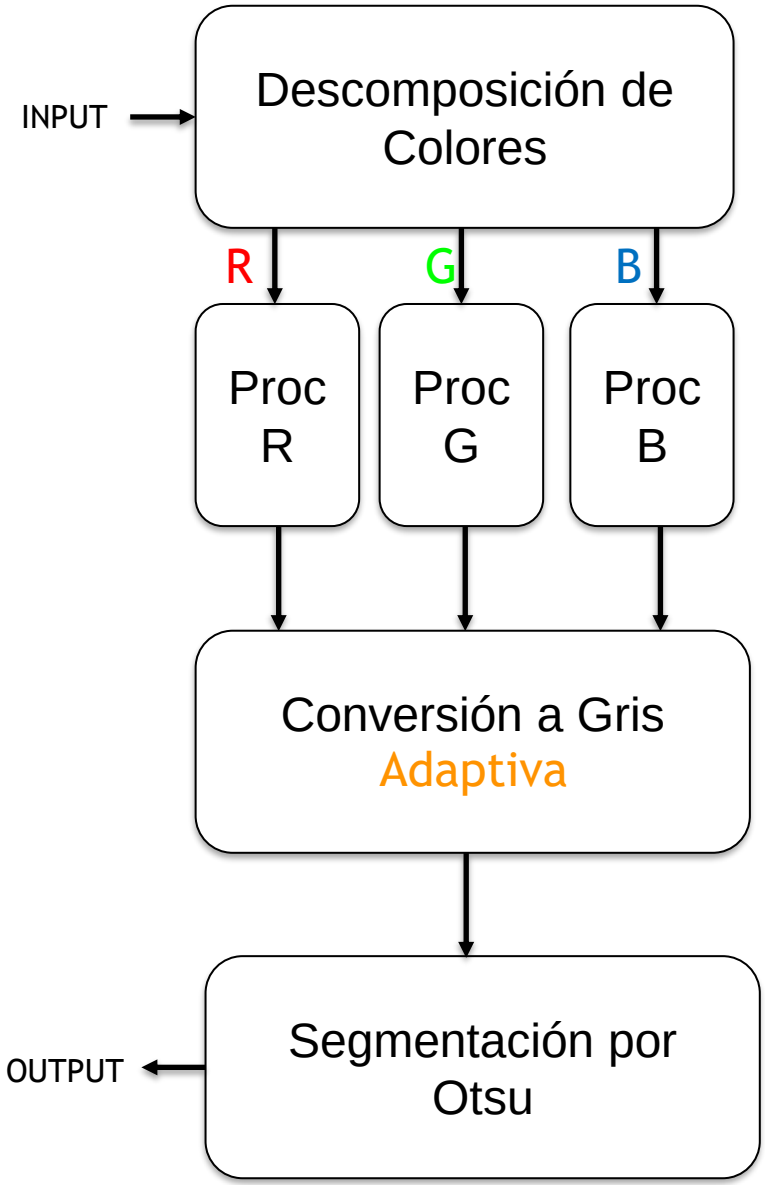



INPUT


RGB > Gris


Otsu

2. Adaptivo por umbral




INPUT


RGB > Gris
(normal)

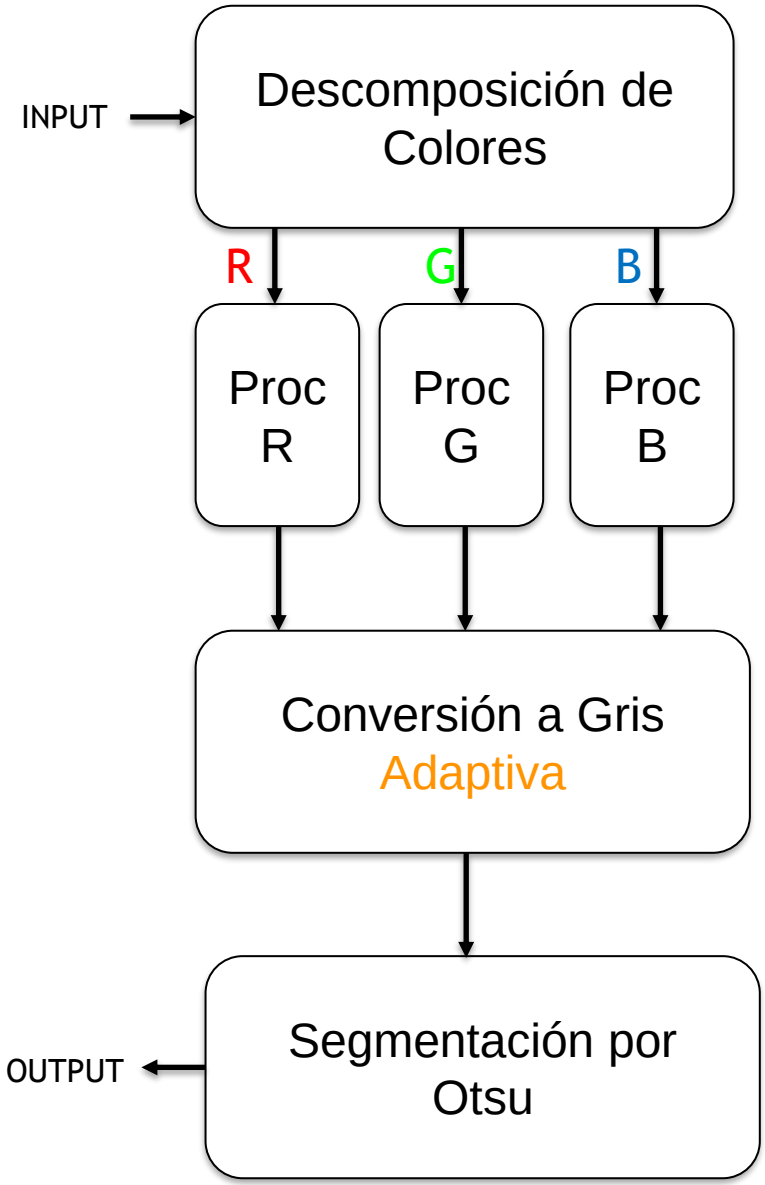

RGB > Gris
(adaptive)
HCI


Otsu

Morfología


OUTPUT

2. Adaptivo por umbral



2. Adaptivo por umbral

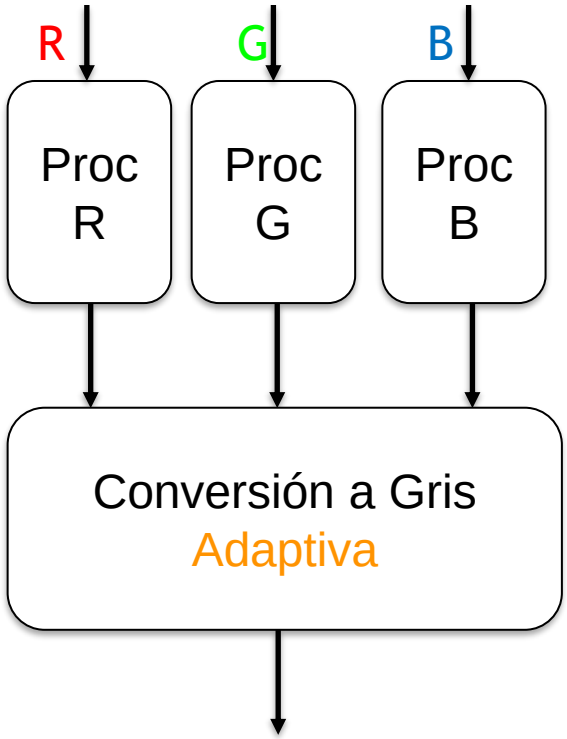
$$I(x,y) = k_r R(x,y) + k_g G(x,y) + k_b B(x,y)$$

En conversión normal:

Los valores k_r, k_g, k_b son aprox. 1/3

Es necesario que siempre sea así?

No

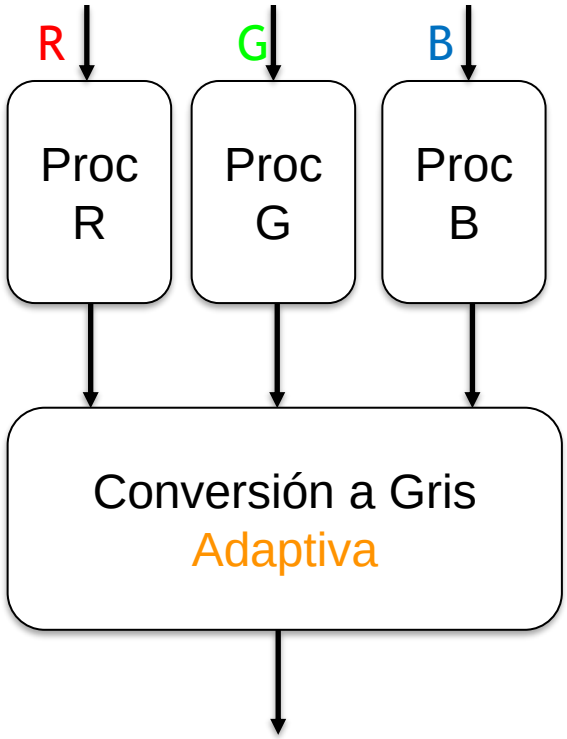


2. Adaptivo por umbral

$$I(x,y) = k_{\mathrm{r}}R(x,y) + k_{\mathrm{g}}G(x,y) + k_{\mathrm{b}}B(x,y)$$

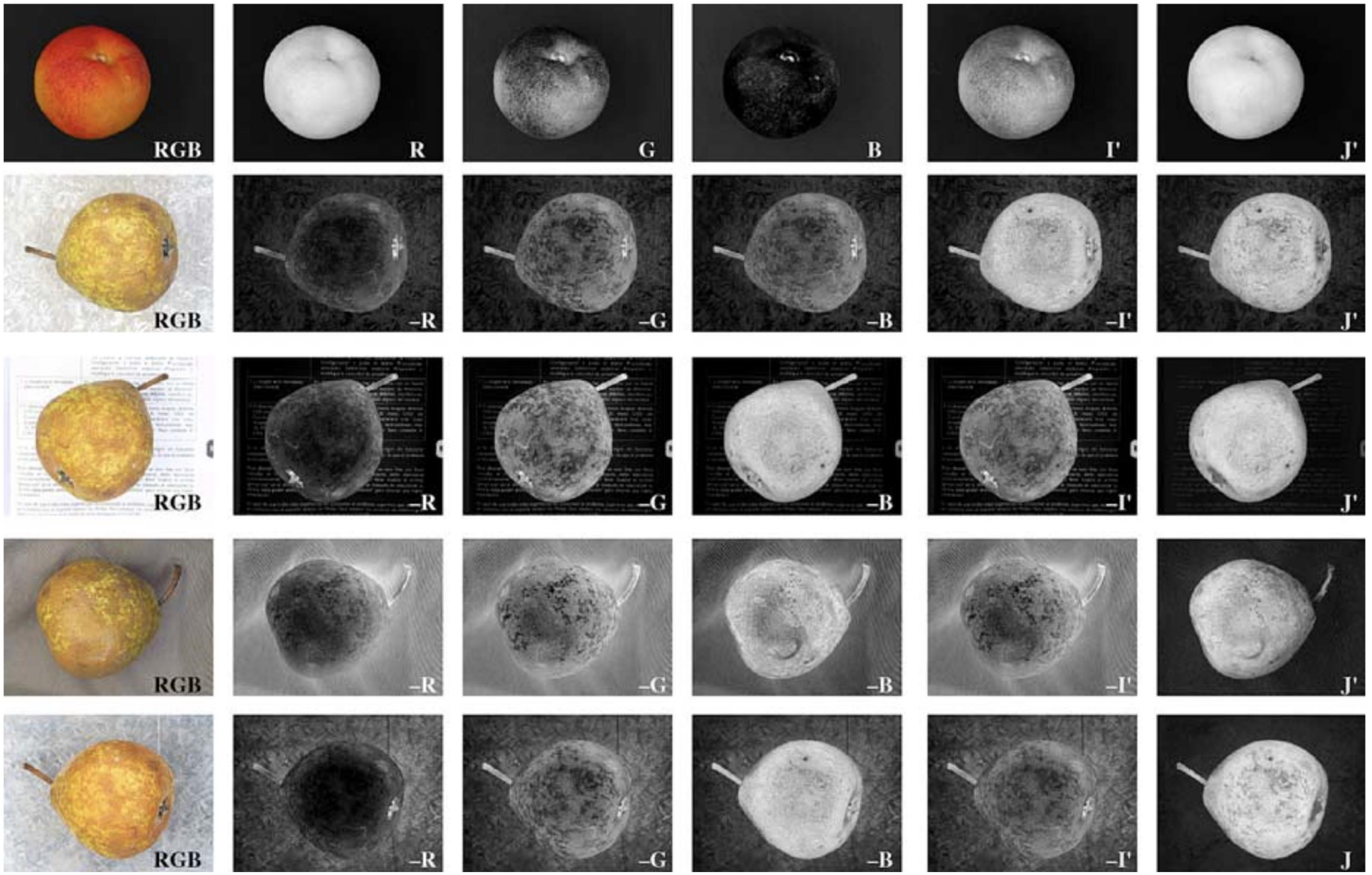
$$J(x,y) = \frac{I(x,y) - I_{\min}}{I_{\max} - I_{\min}}$$

$$\sigma_J^2(k_{\mathrm{r}}, k_{\mathrm{g}}, k_{\mathrm{b}}) \rightarrow \max$$



Optimización: Estimar k_r, k_g, k_b tal que la varianza de J sea máxima

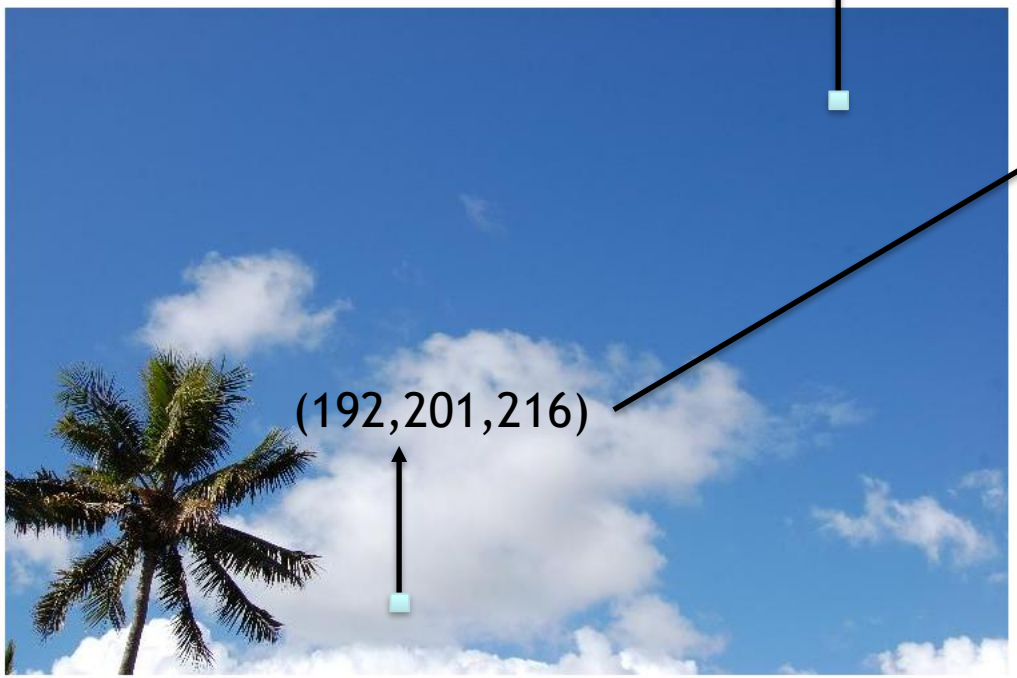
Solución: J es una imagen de alto contraste (High Contrast Image)

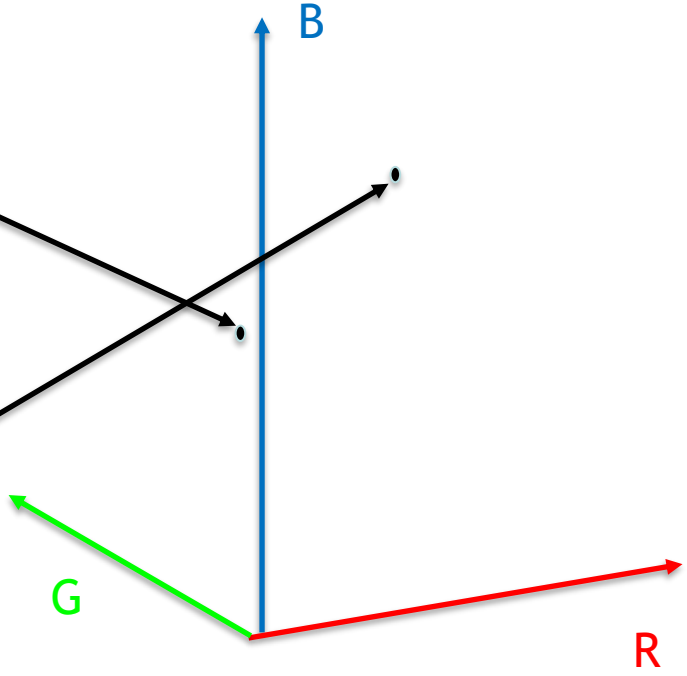

RGB
R
G
B
Gris
HCI
RGB
R
G
B
I'
J'
RGB
–R
–G
–B
–I'
J'
RGB
–R
–G
–B
–I'
J'
RGB
–R
–G
–B
–I'
J'
RGB
–R
–G
–B
–I'
J

Tres métodos fundamentales:

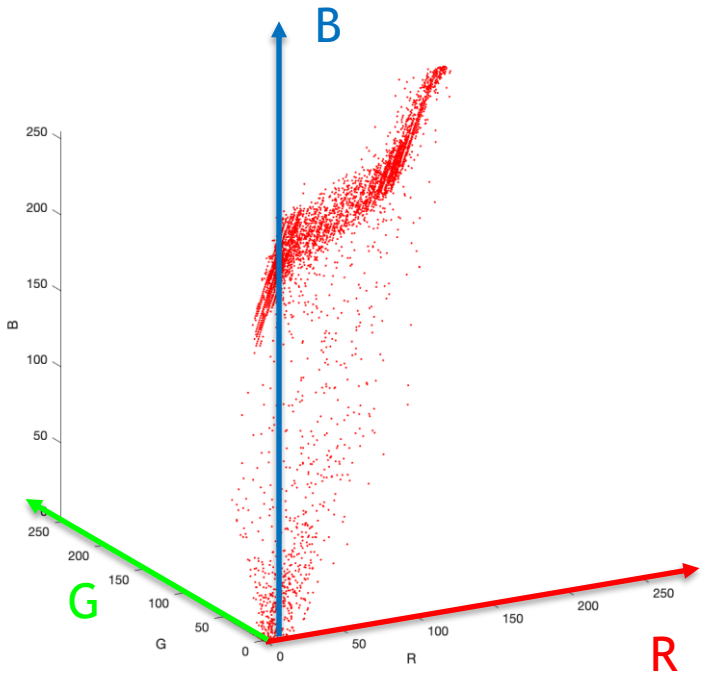
1. Por canal de color
2. Adaptivo por umbrales
3. Clustering

3. Clustering

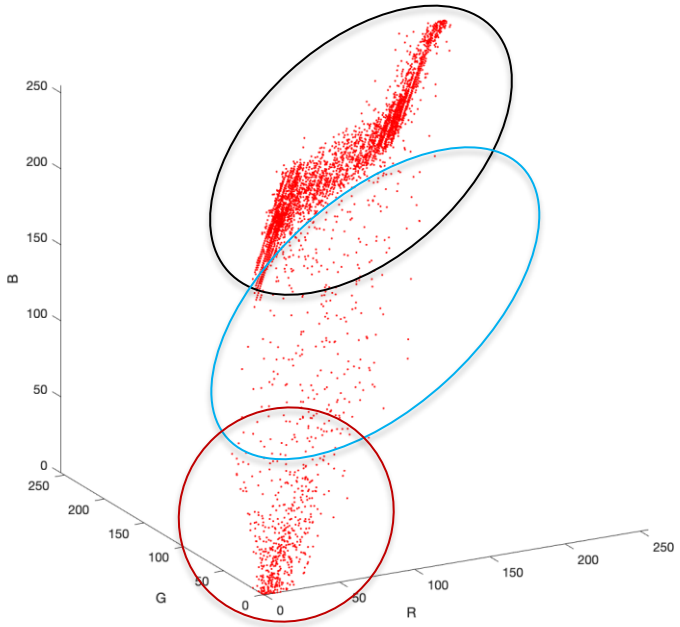




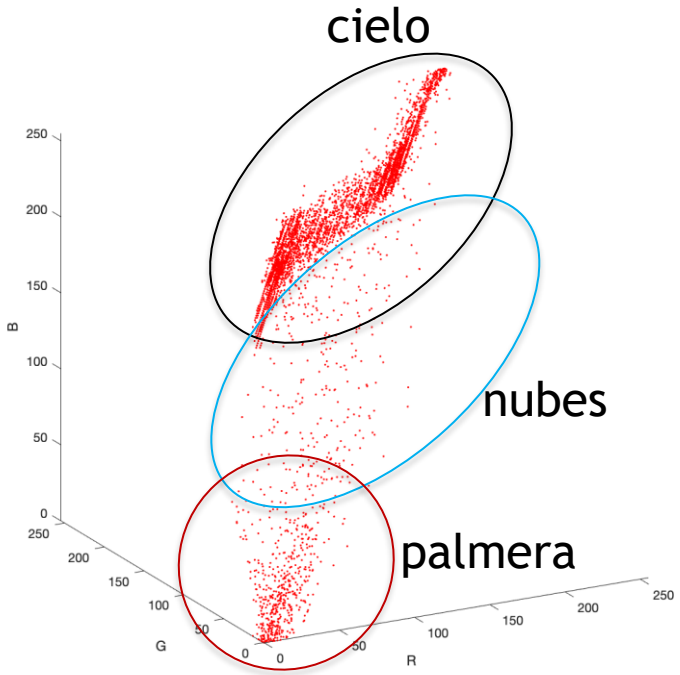
3. Clustering



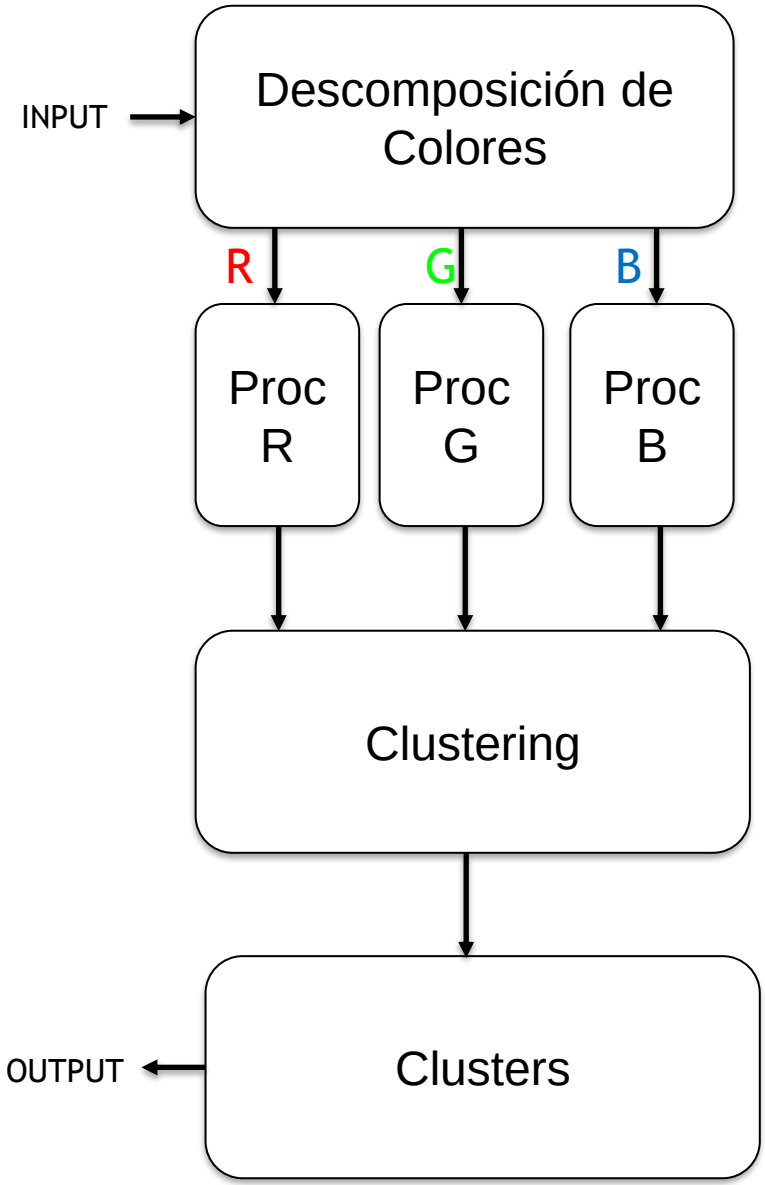
3. Clustering



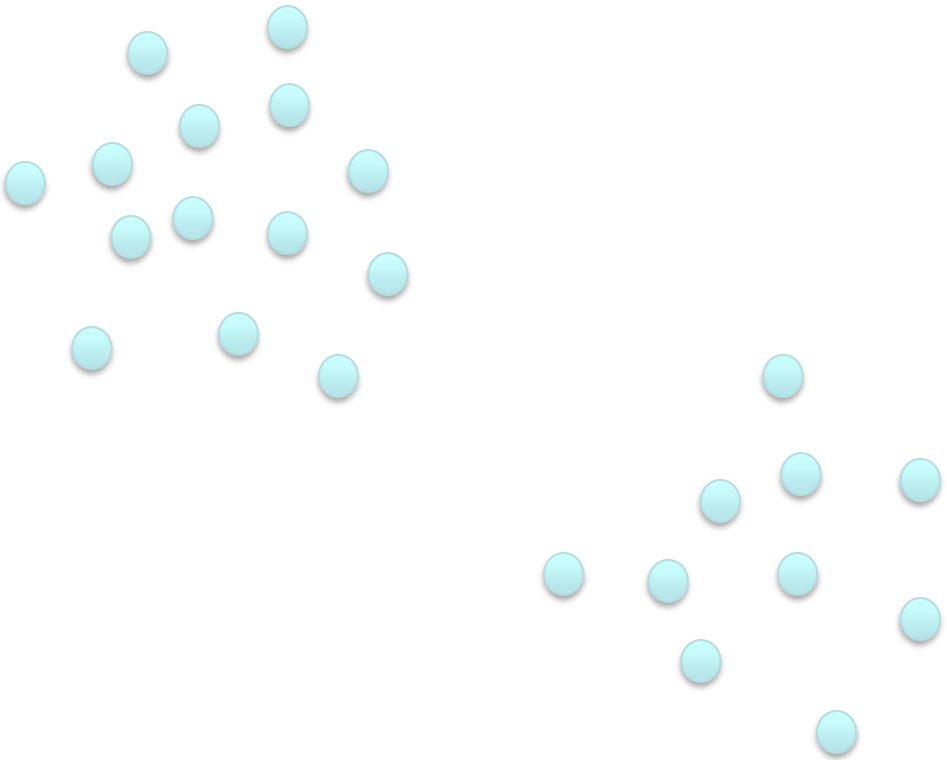
3. Clustering



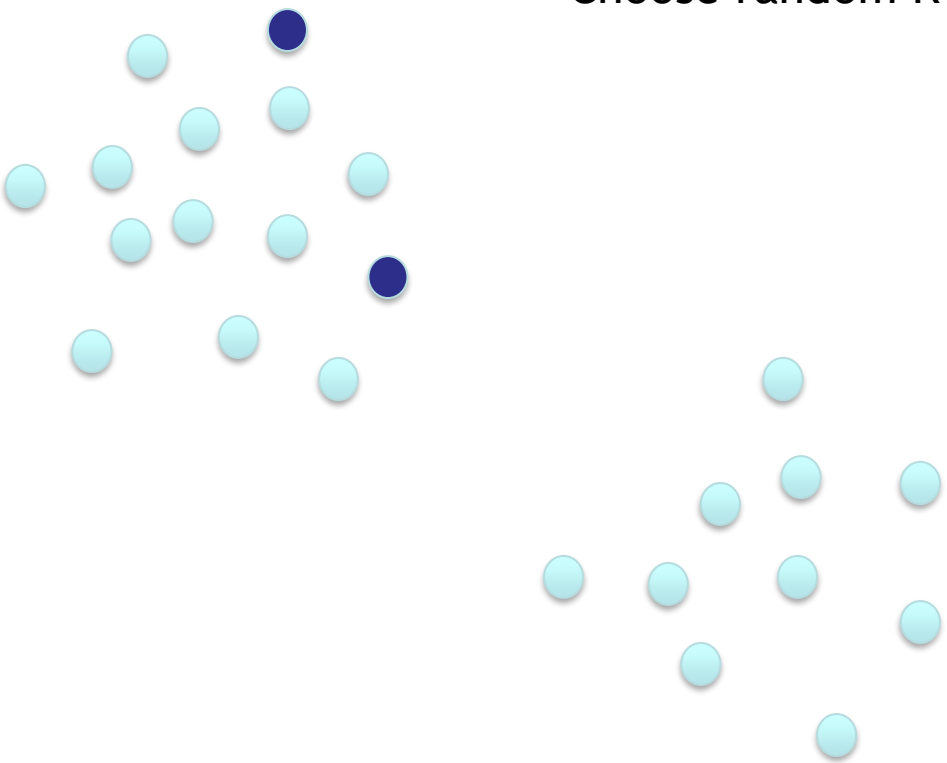
3. Clustering



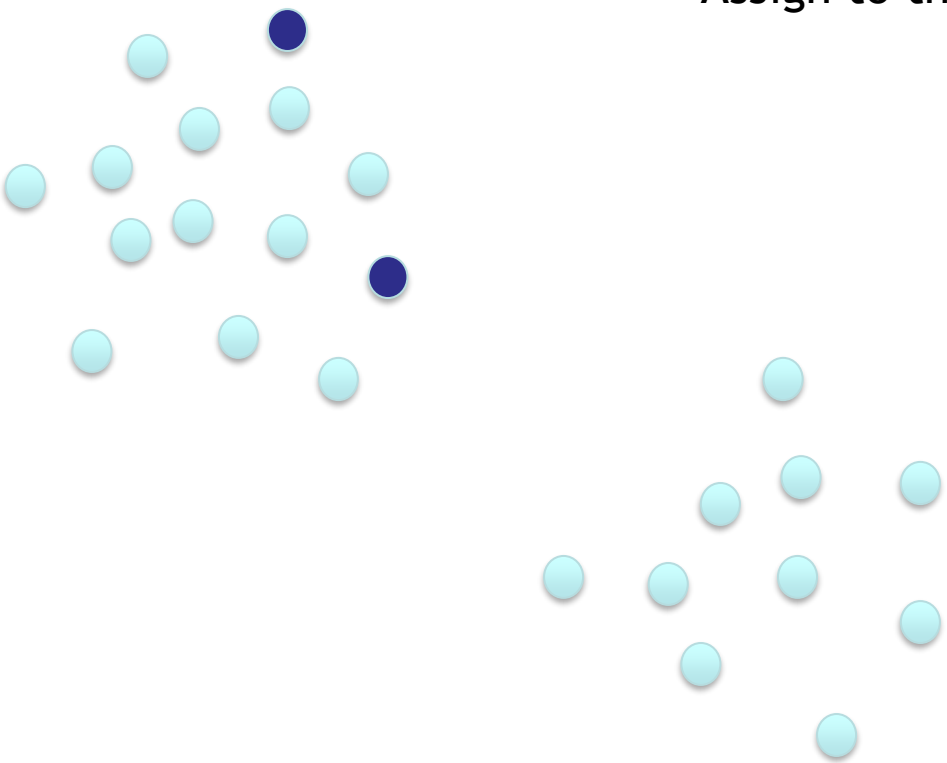
Clustering usando k-means

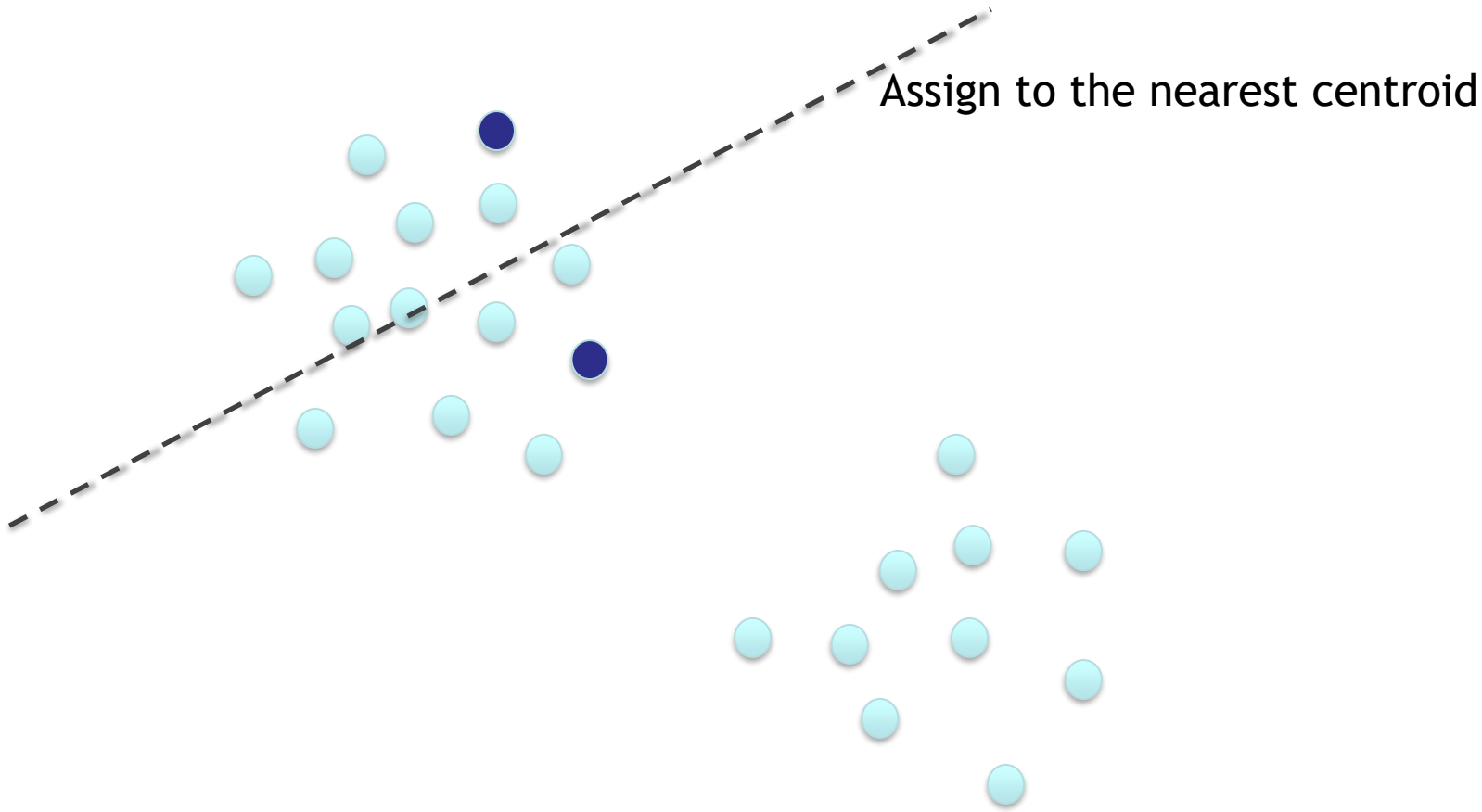


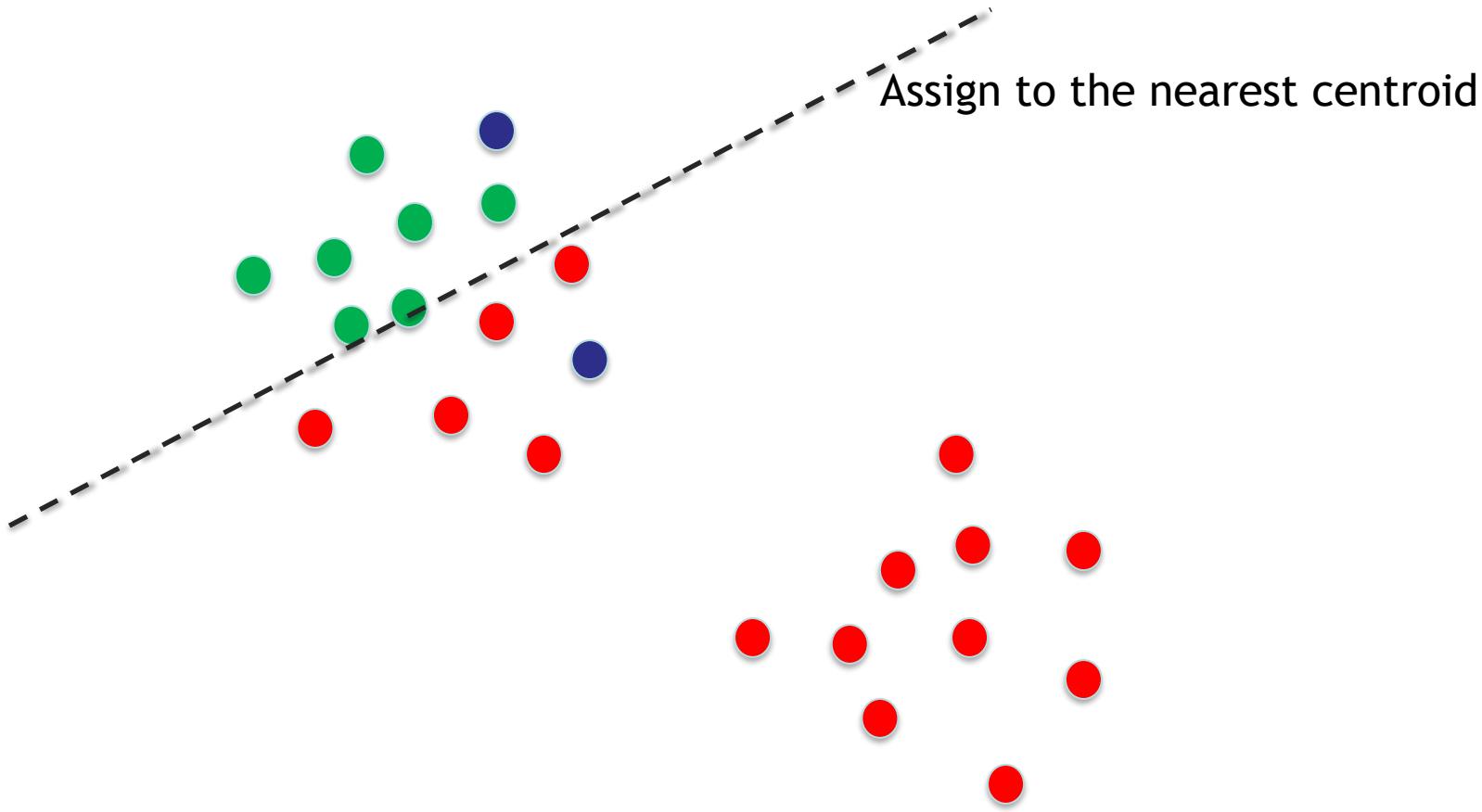
Choose random K=2 points (centroids)

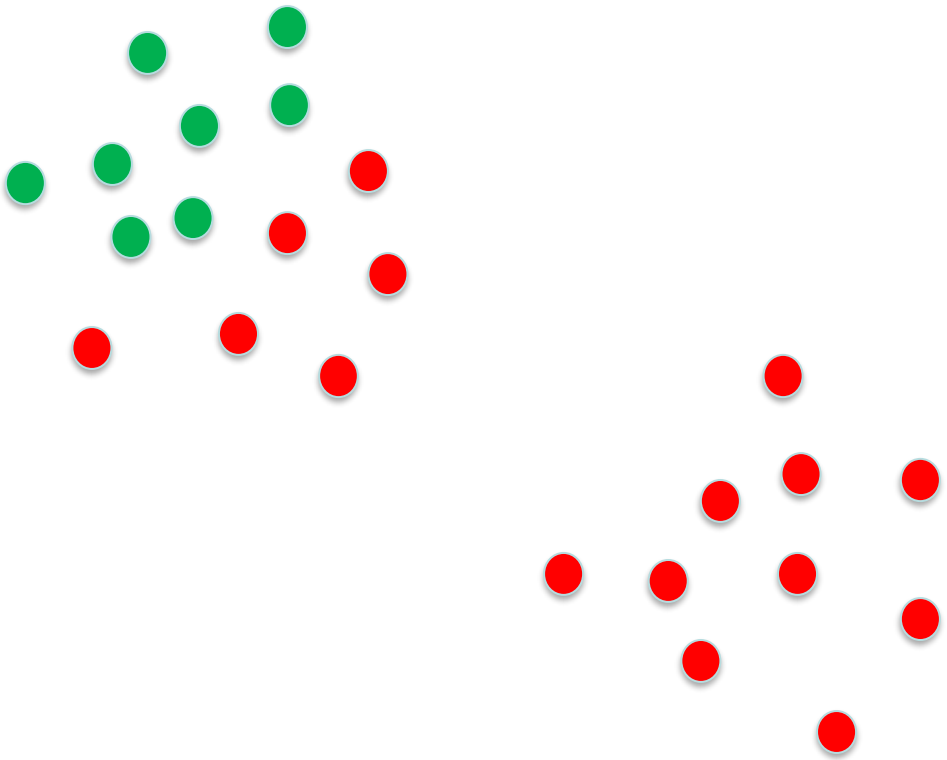


Assign to the nearest centroid

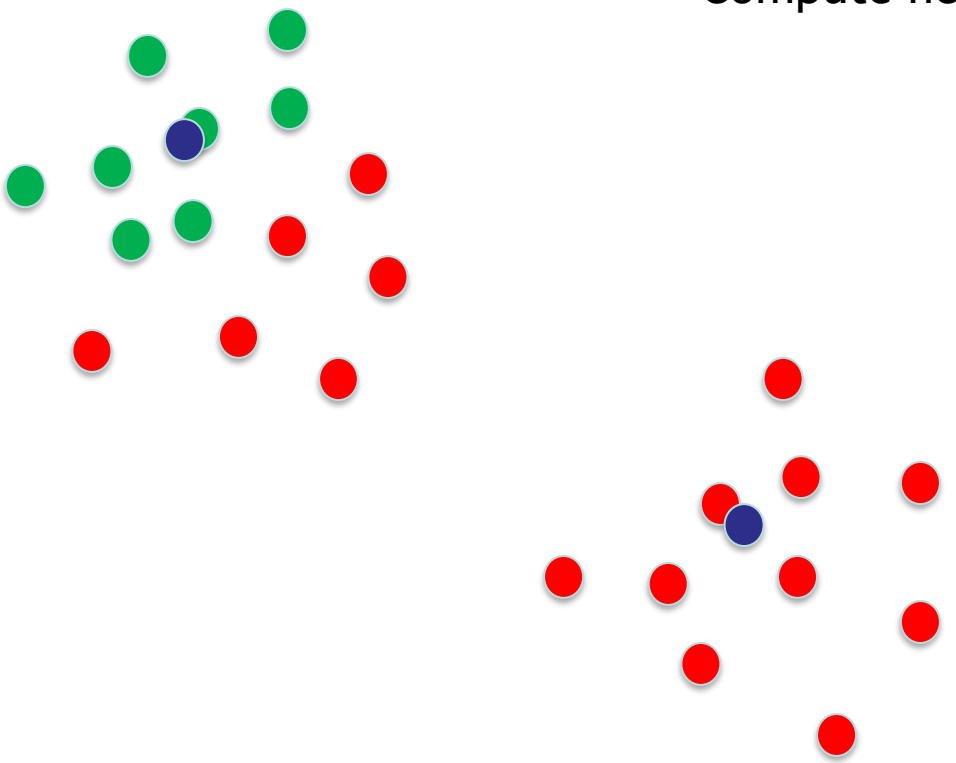


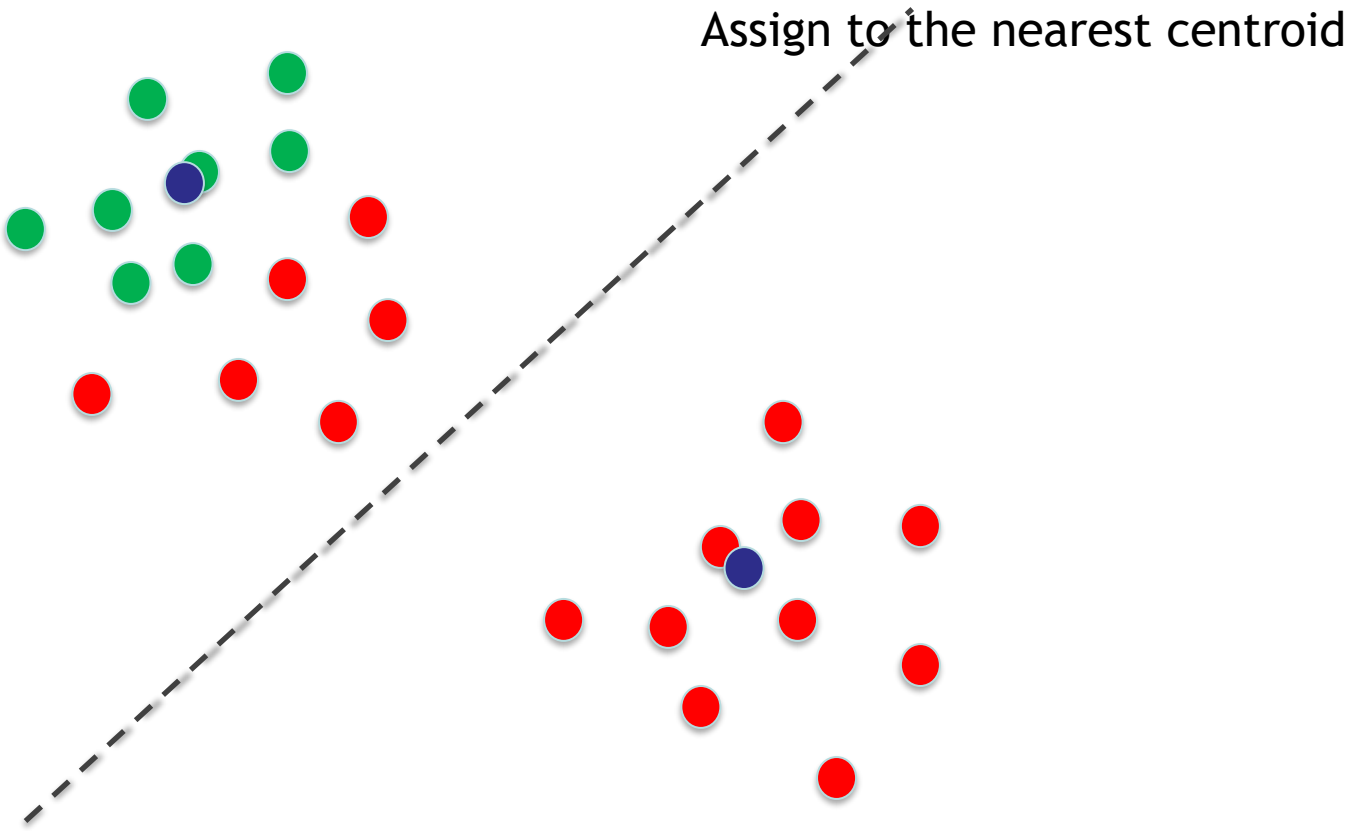

Assign to the nearest centroid

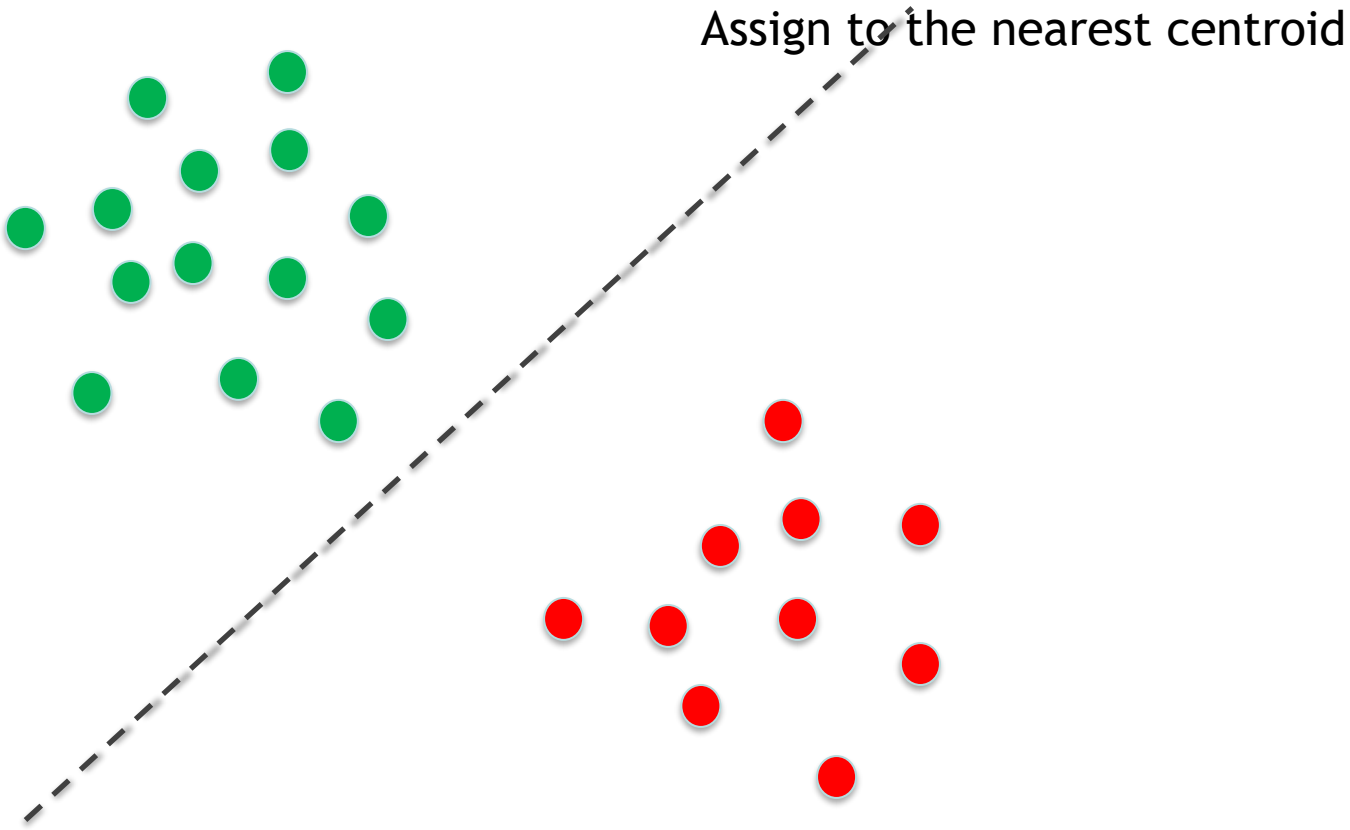

Assign to the nearest centroid



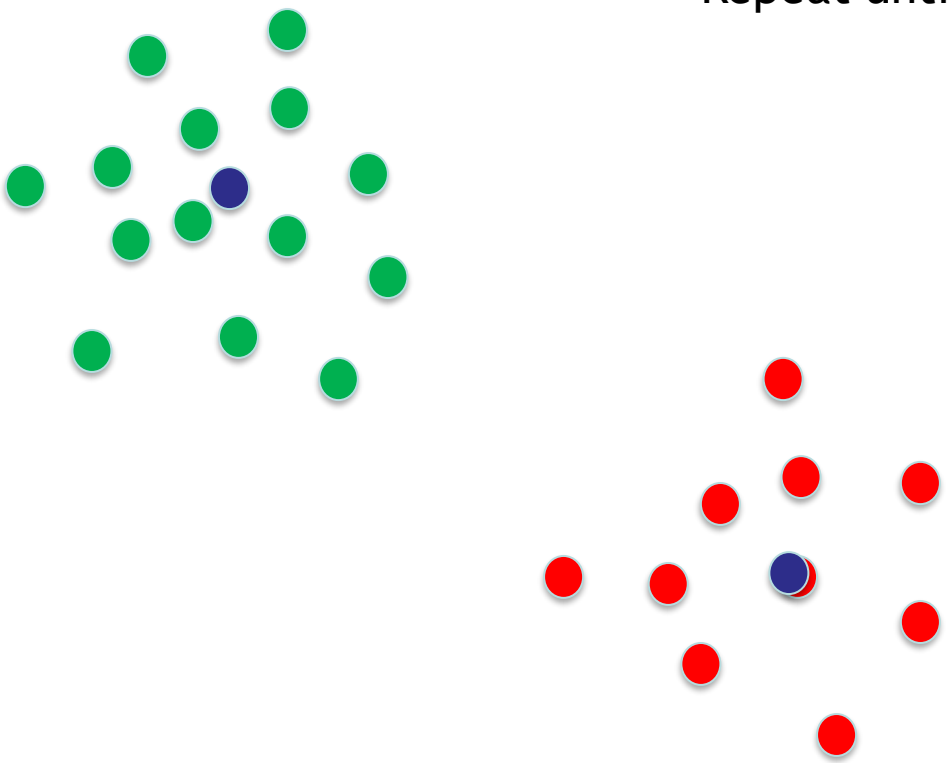
Compute new centroids




Assign to the nearest centroid


Assign to the nearest centroid

Repeat until convergence



Algorithm K-Means:

1. Input Data $X = \{x_1, x_2, \ldots x_N\}$ and number of clusters K

2. Centroids $\{c_1, c_2, \ldots c_K\}$ = random K points of X

3. For each data point x_i

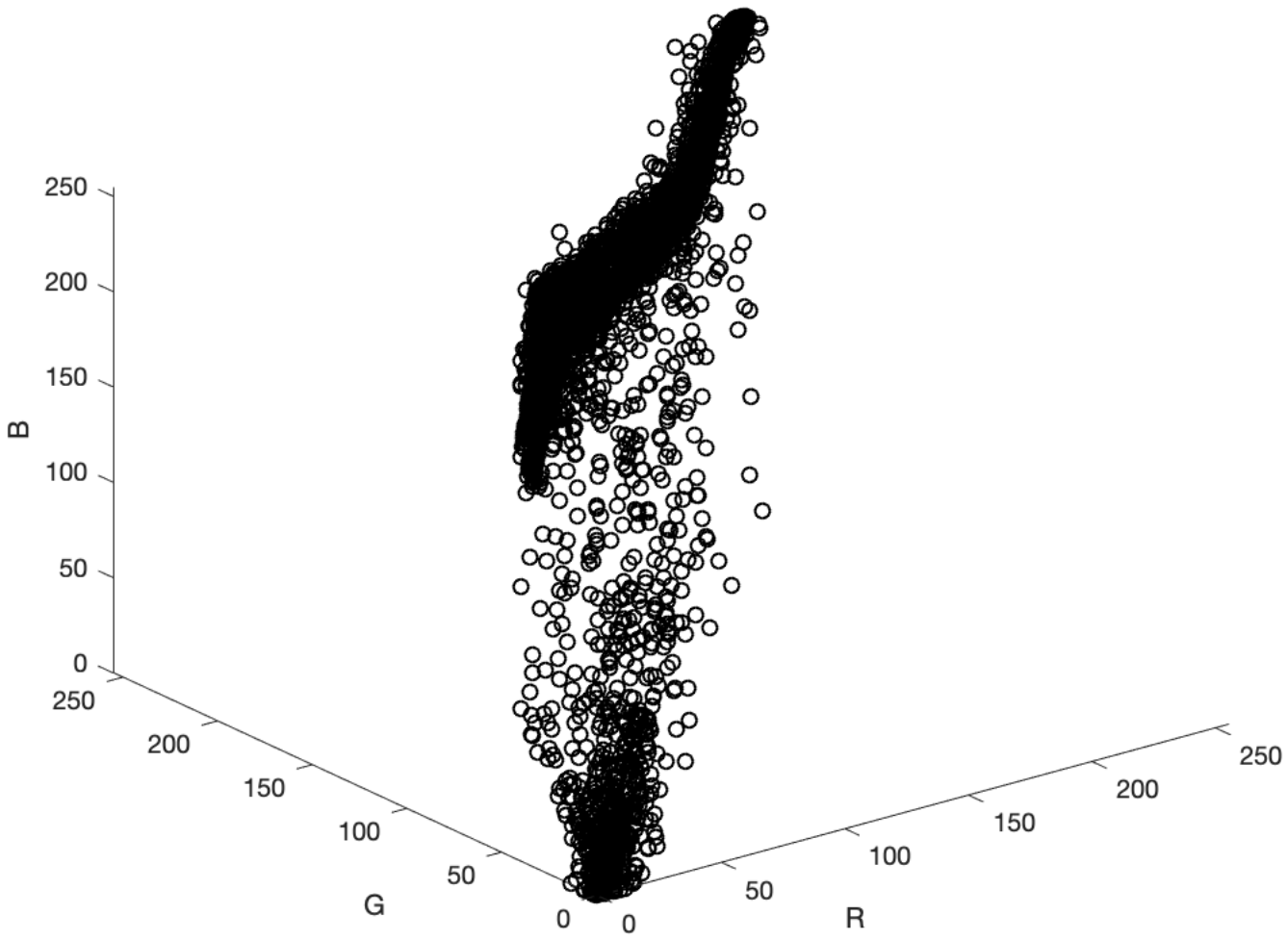
4. Compute distance $d_{ij} = d(x_i, c_j)$ $i=1,\ldots,N,\ j=1,\ldots K$

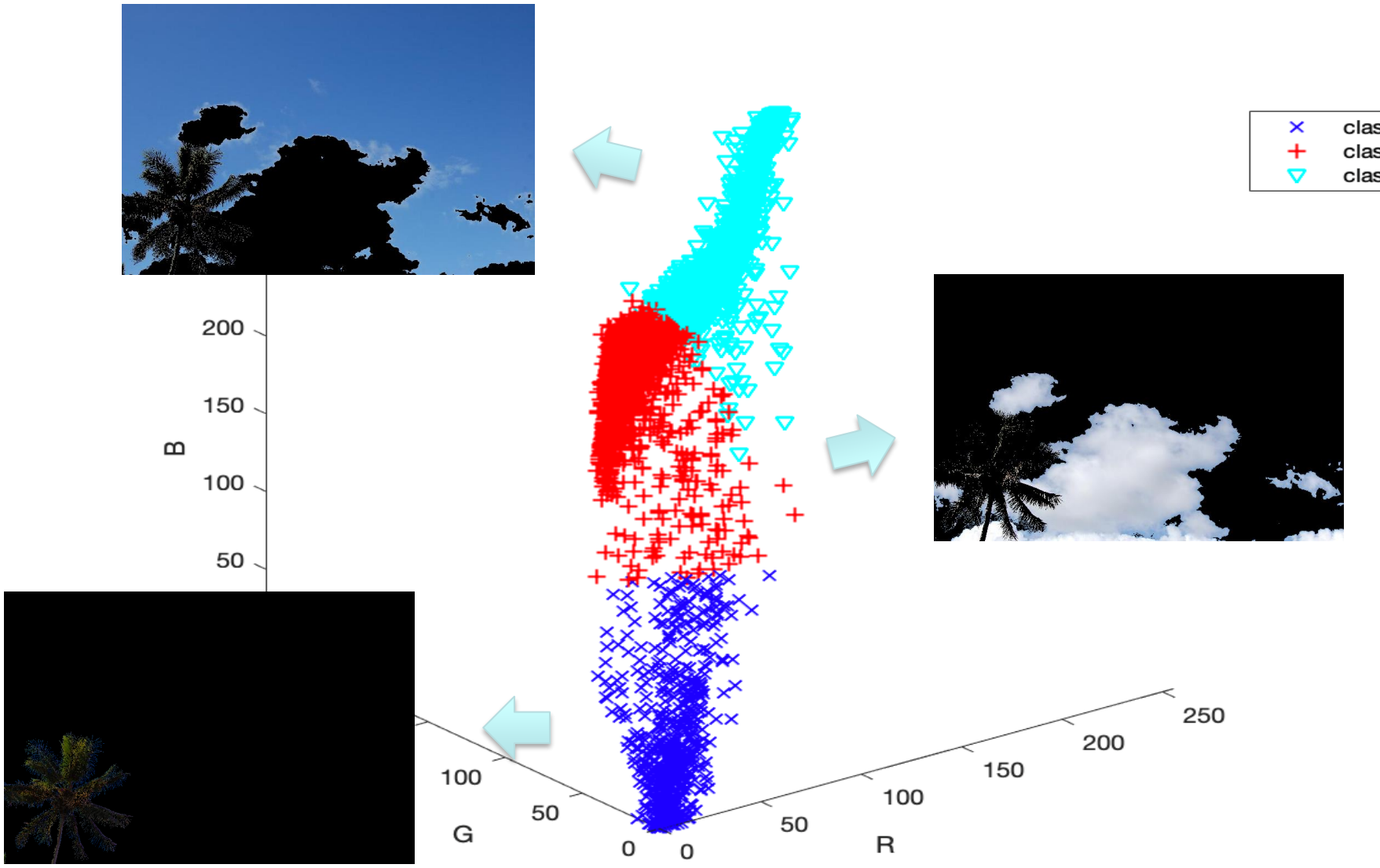
5. Assign x_i to the nearest centroid: $y_i = \text{argmin}_j\{d_{ij}\}$

6. Compute the new centroids of each cluster
   $c^*_j = \text{mean}(x_i)$ for $y_i = j$

7. if $c^*_j \neq c_j$ then $c_j = c^*_j$ go to step 3

8. Output: $\{c^*_1, c^*_2, \ldots c^*_K\}$ and y_i for $i=1,\ldots,N$


B
250
200
150
100
50
0
G
250
200
150
100
50
0
R
0
50
100
150
200
250


class 1
class 2
class 3
B
200
150
100
50
G
100
50
0
0
50
100
150
200
250
R

INPUT

OUTPUT

Tres métodos fundamentales:

1. Por canal de color

2. Adaptivo por umbrales

3. Usando clustering