Universidad Popular del Cesar

Tratamiento de Señales

Version 2022-1

Segmentación con Imágenes a Color

[Capítulo 2]

Dr. José Ramón Iglesias

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

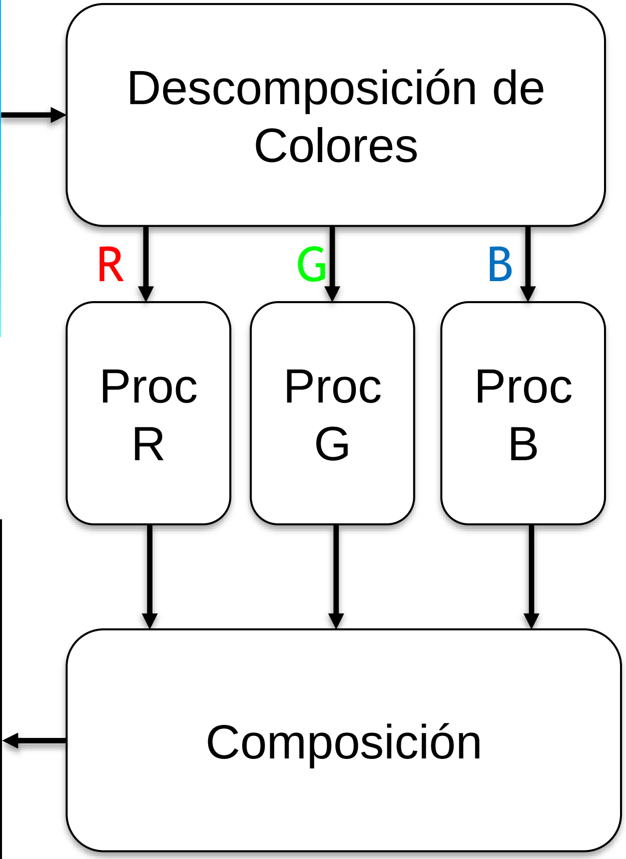
Tres métodos fundamentales:

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



Descomposición de Colores

R

G

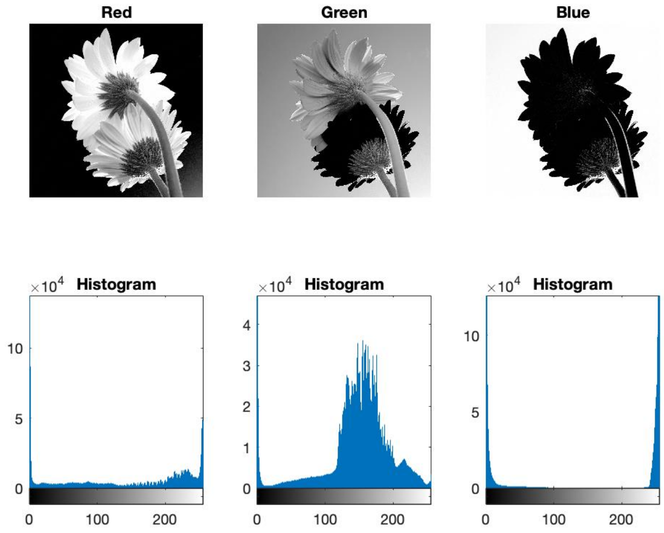
B

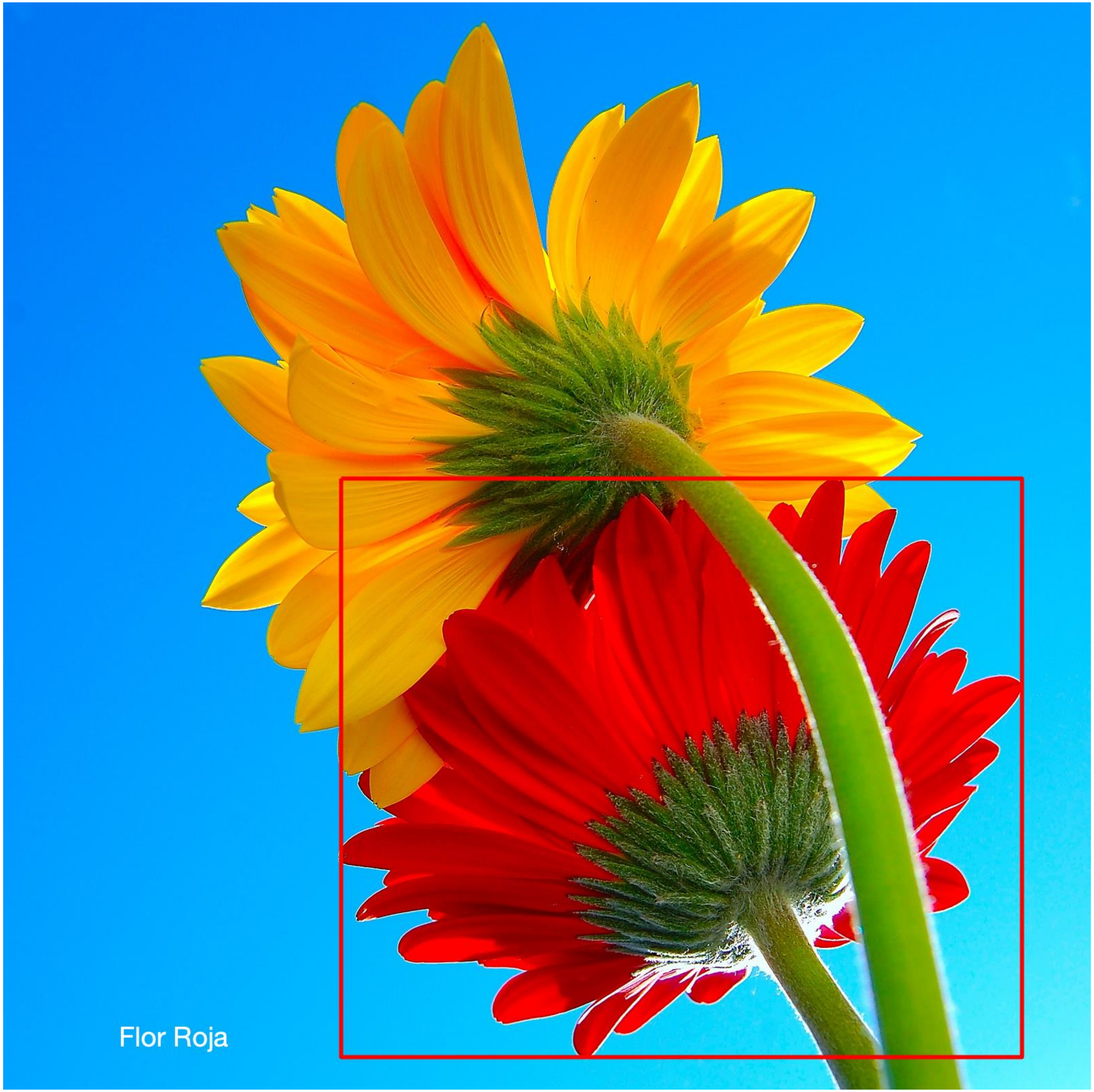
Proc R

Proc G

Proc B

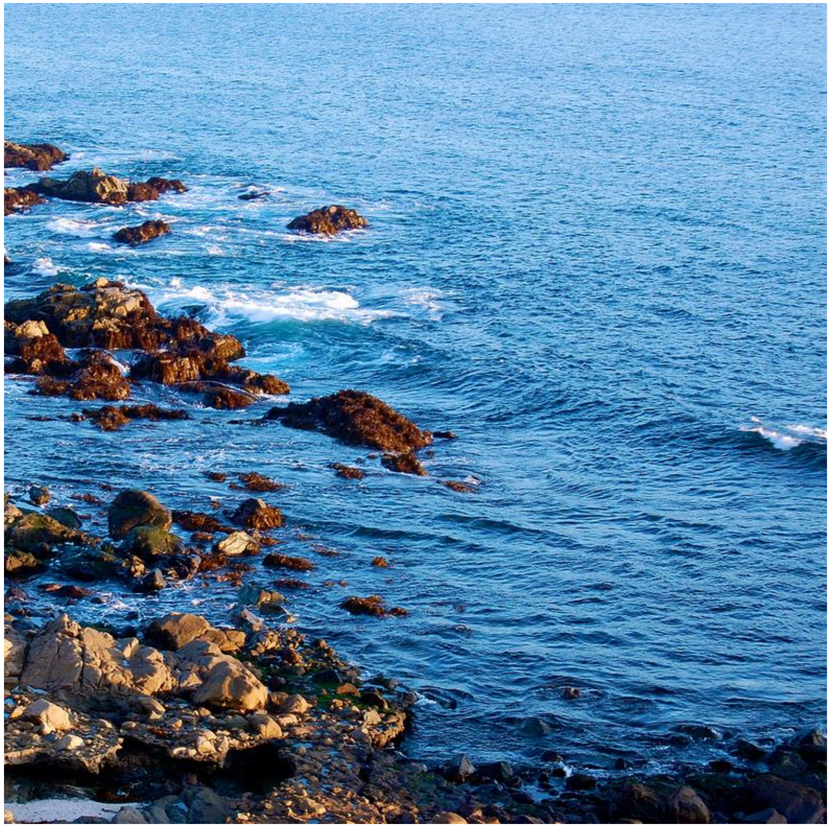
Composición

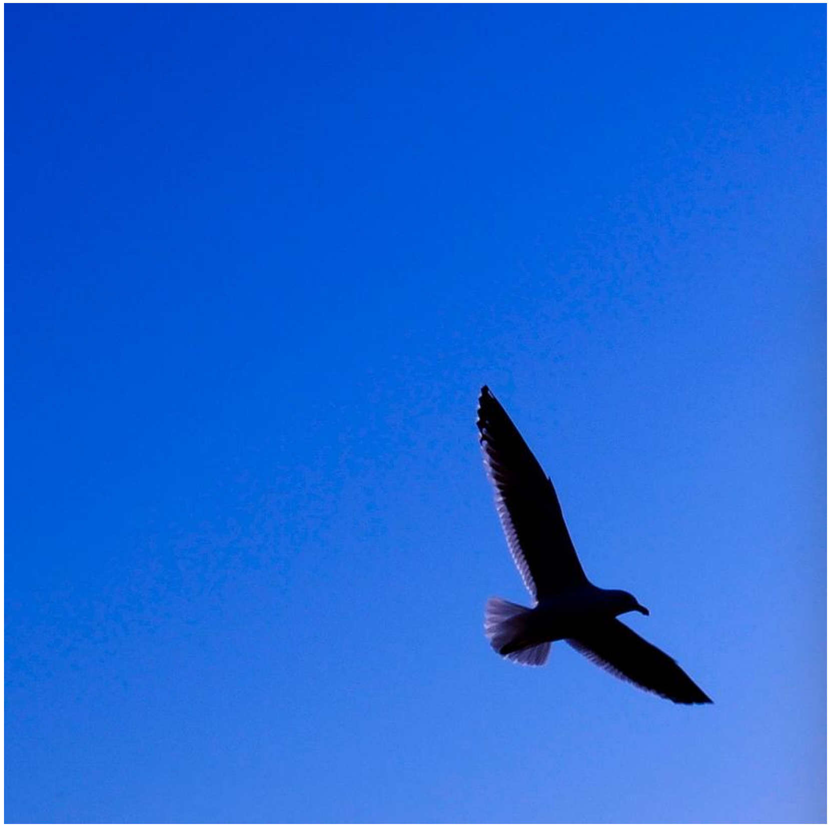


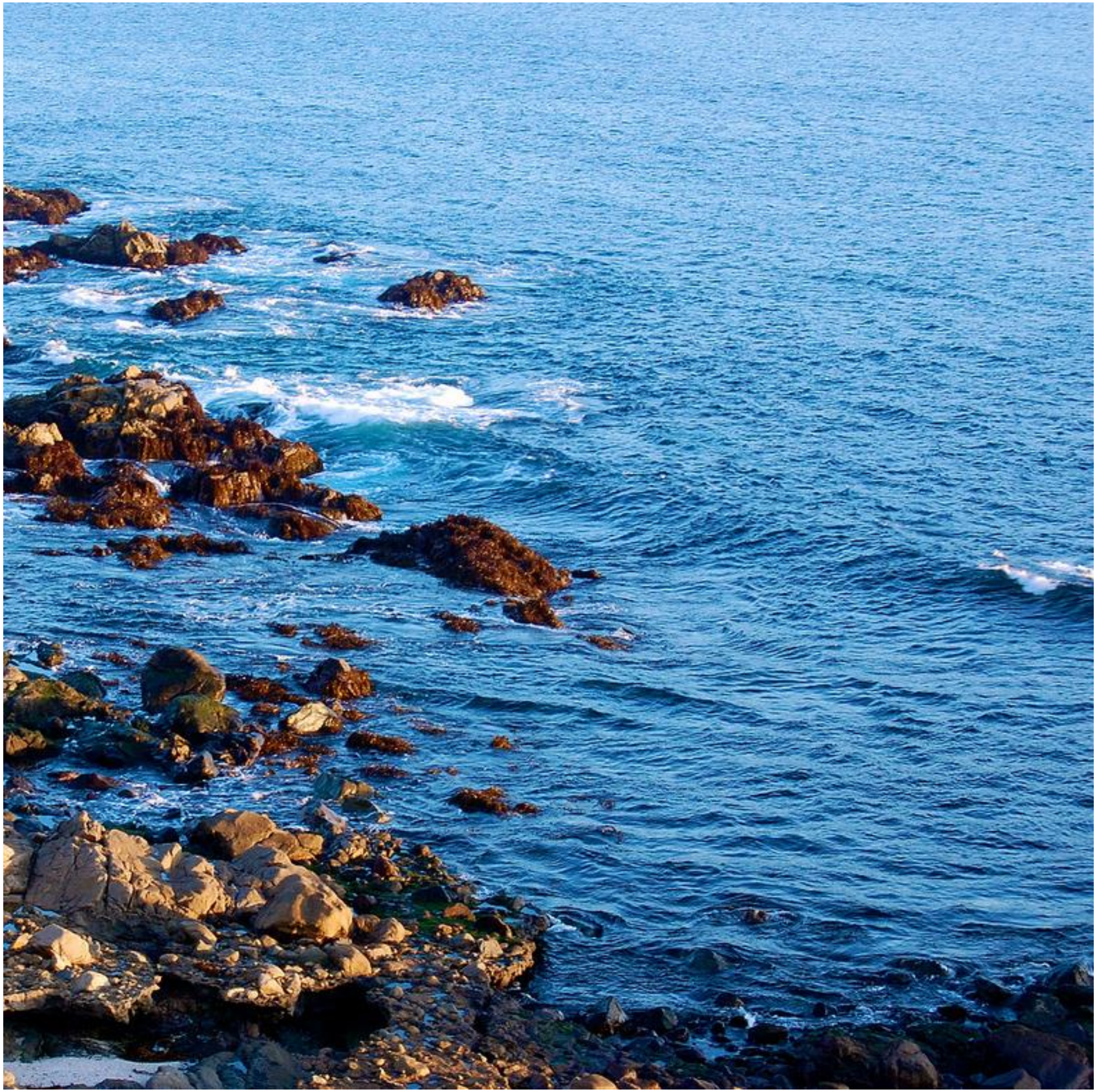


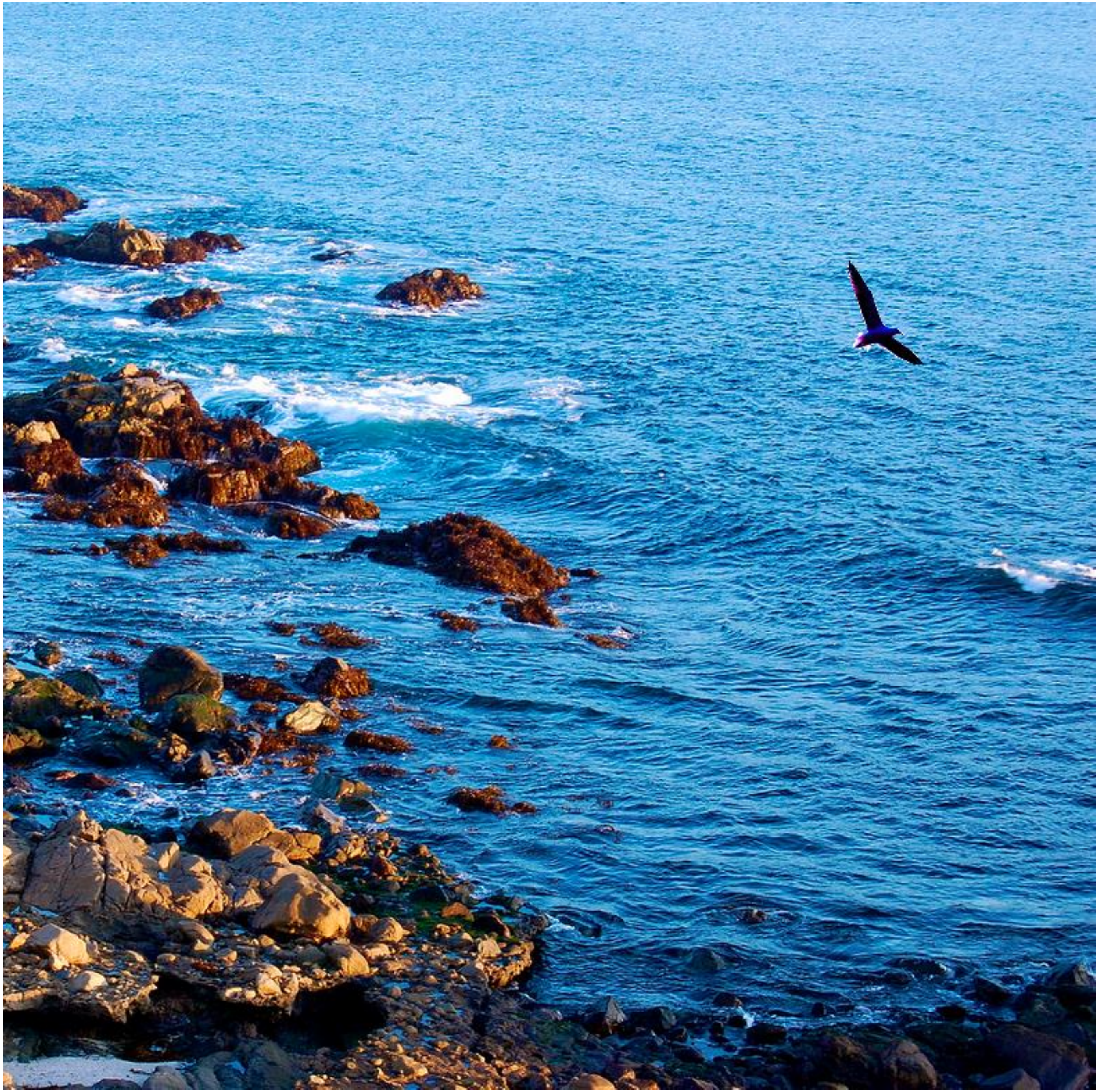
Flor Roja

Ejemplos Similares









Otros Ejemplos

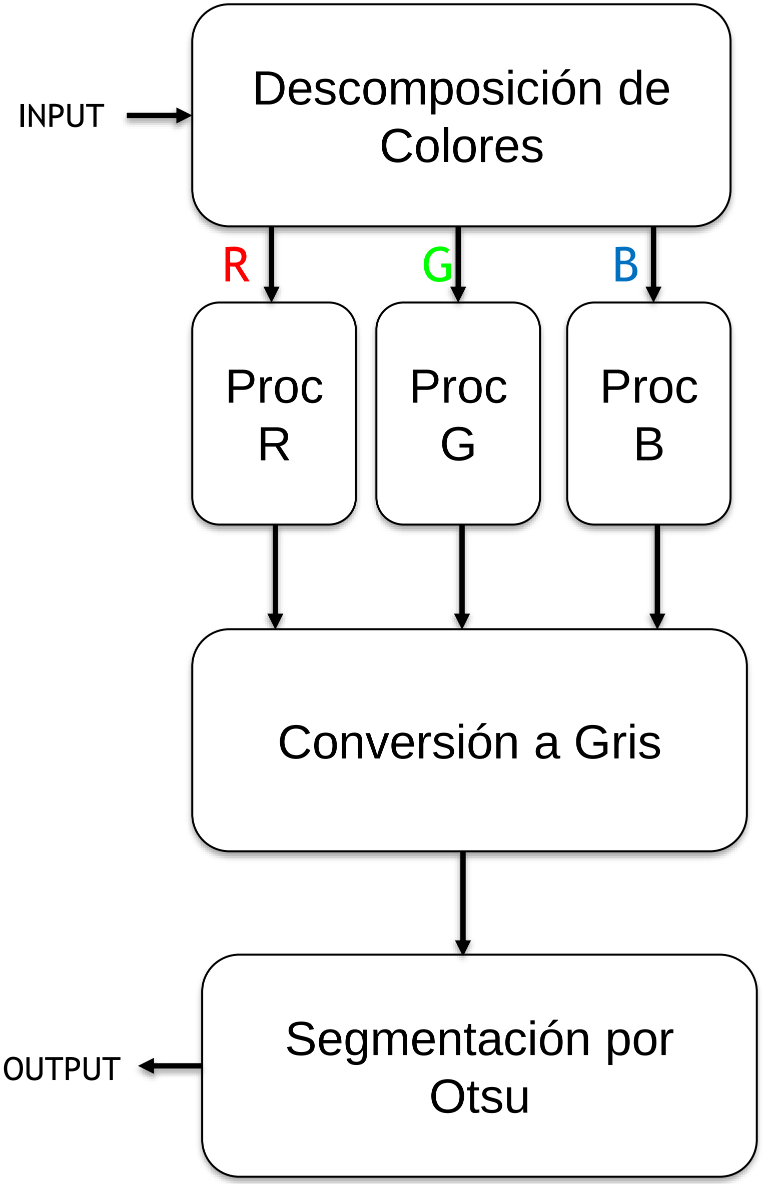
Tres métodos fundamentales:

1. Por canal de color

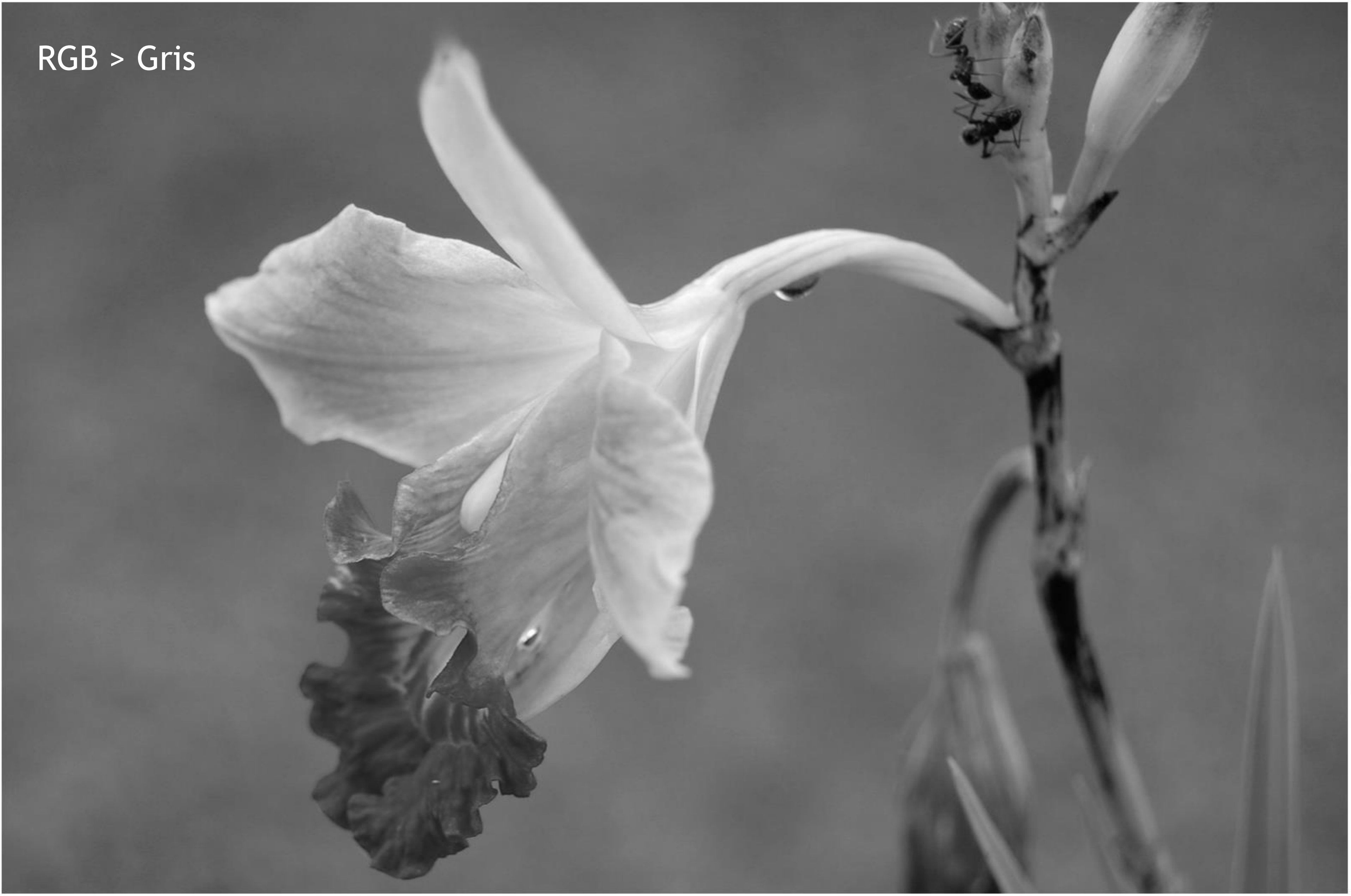
2. Adaptivo por umbrales

3. Clustering

2. Adaptivo por umbral (idea inicial)



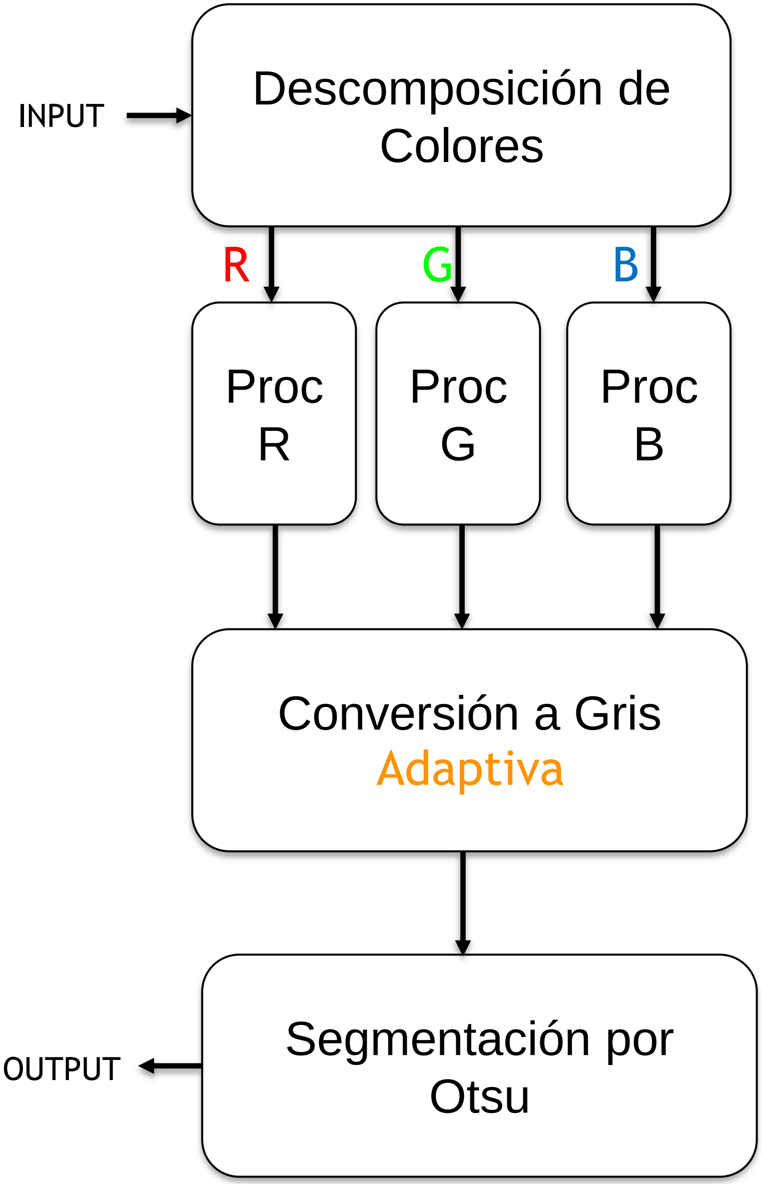
INPUT

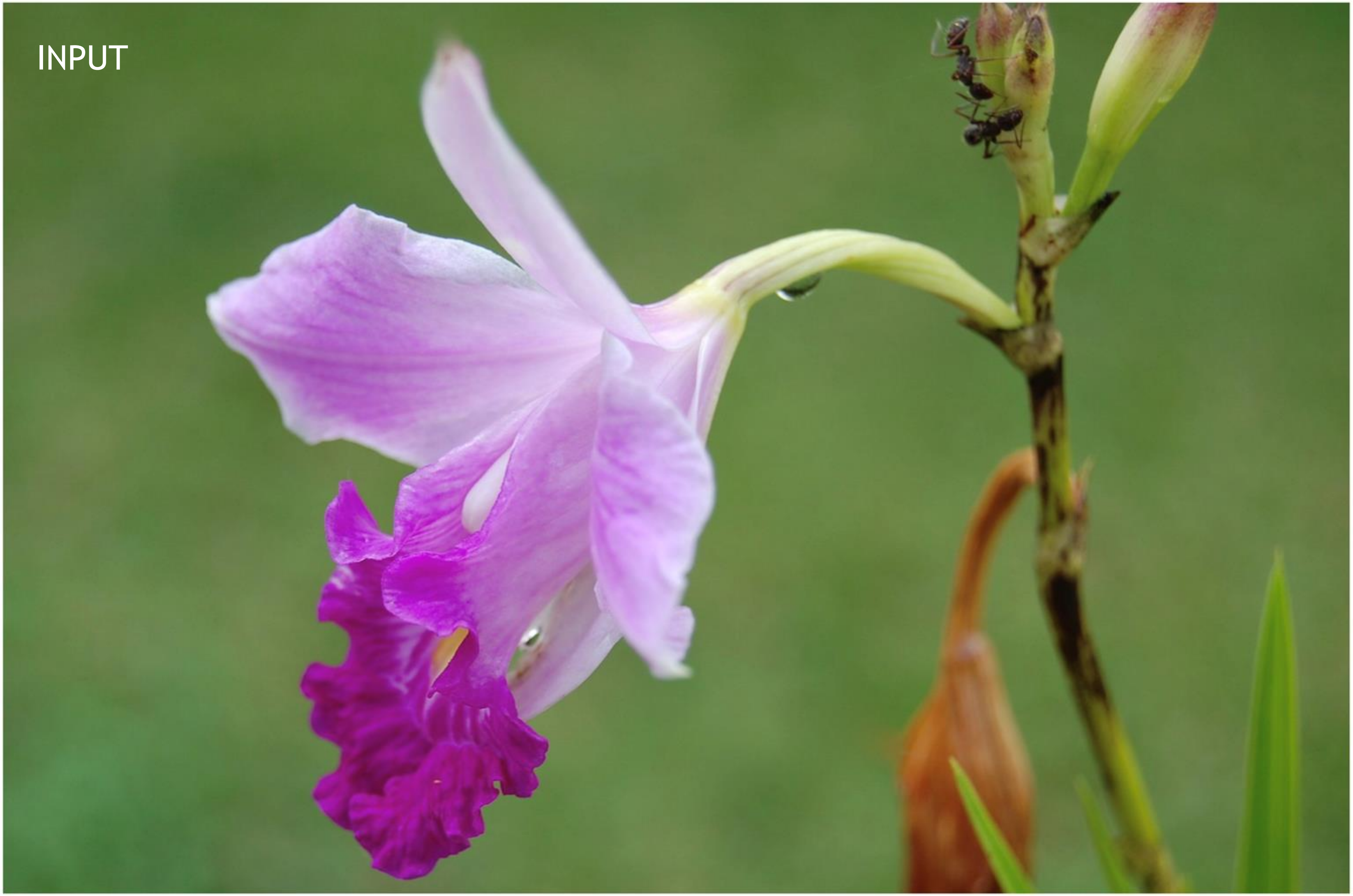


RGB > Gris

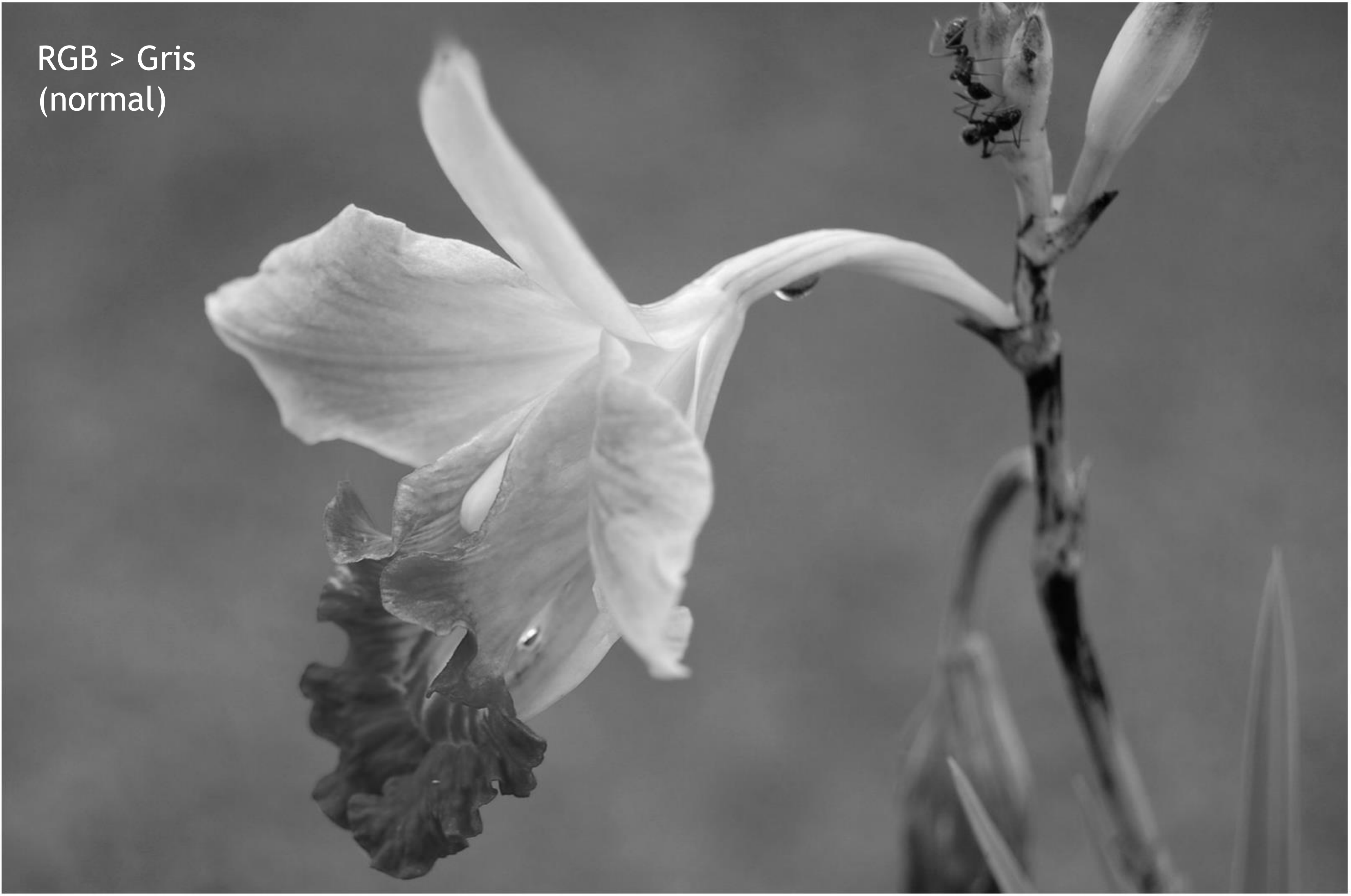
Otsu

2. Adaptivo por umbral

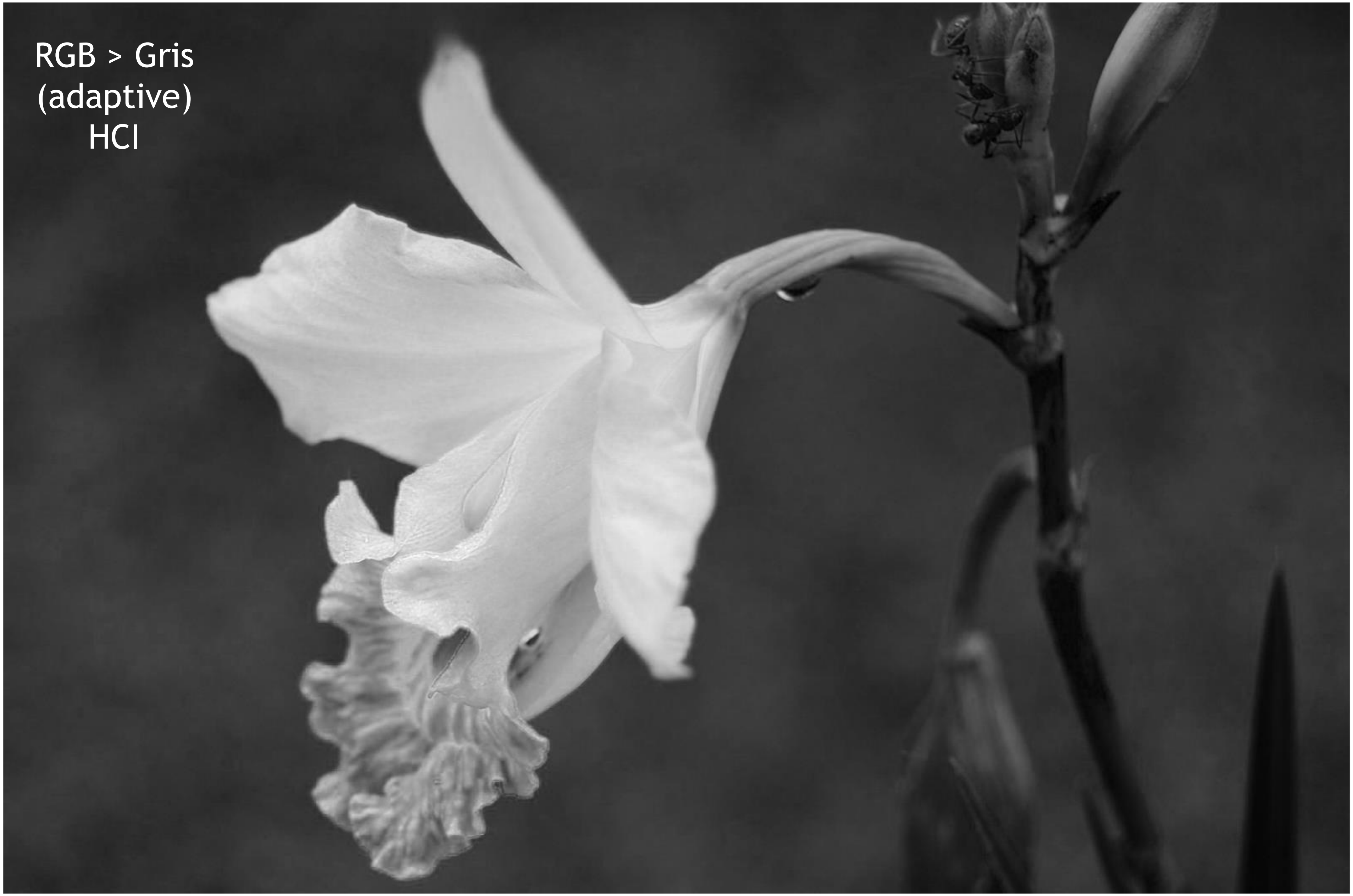




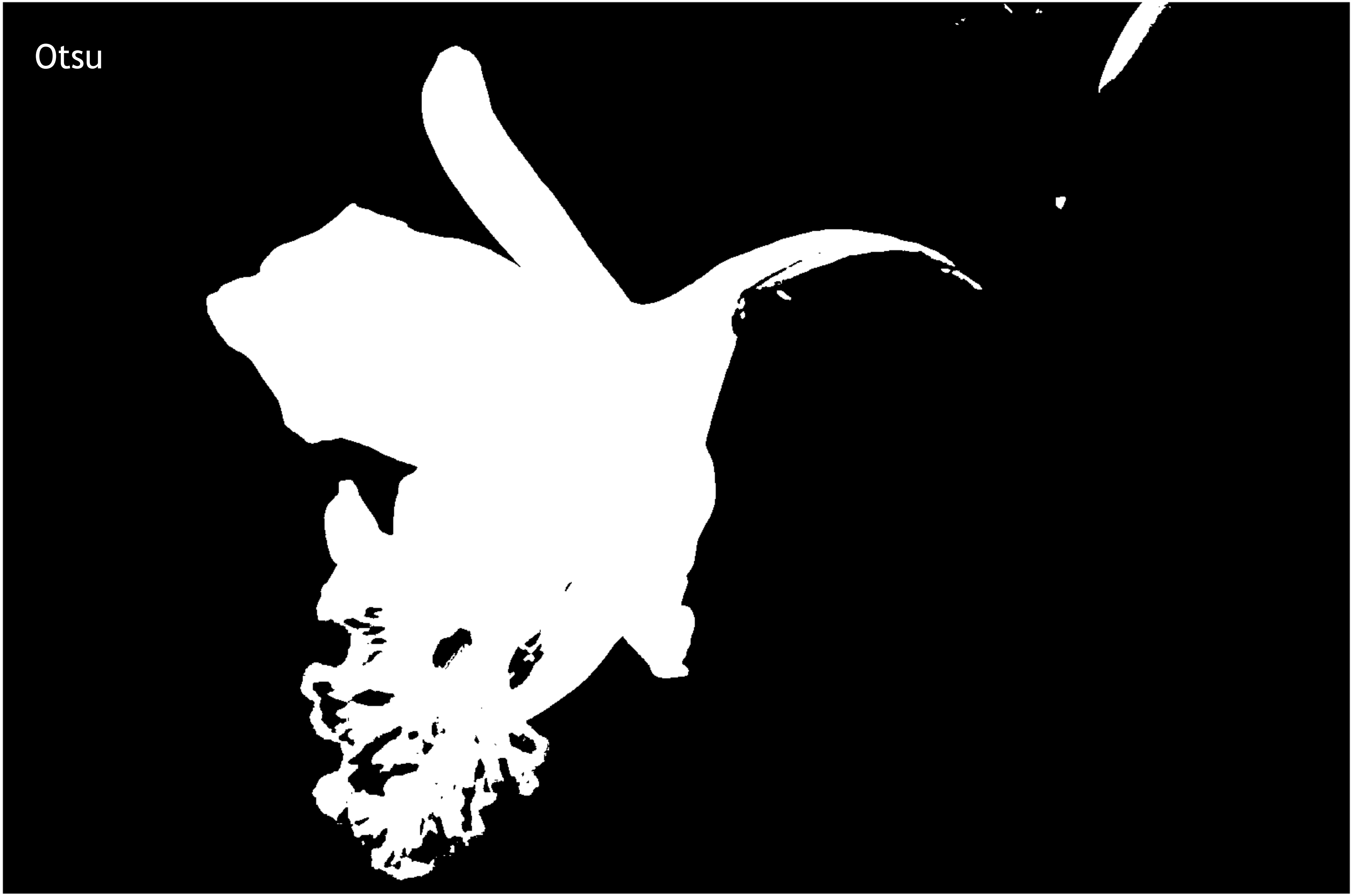
INPUT



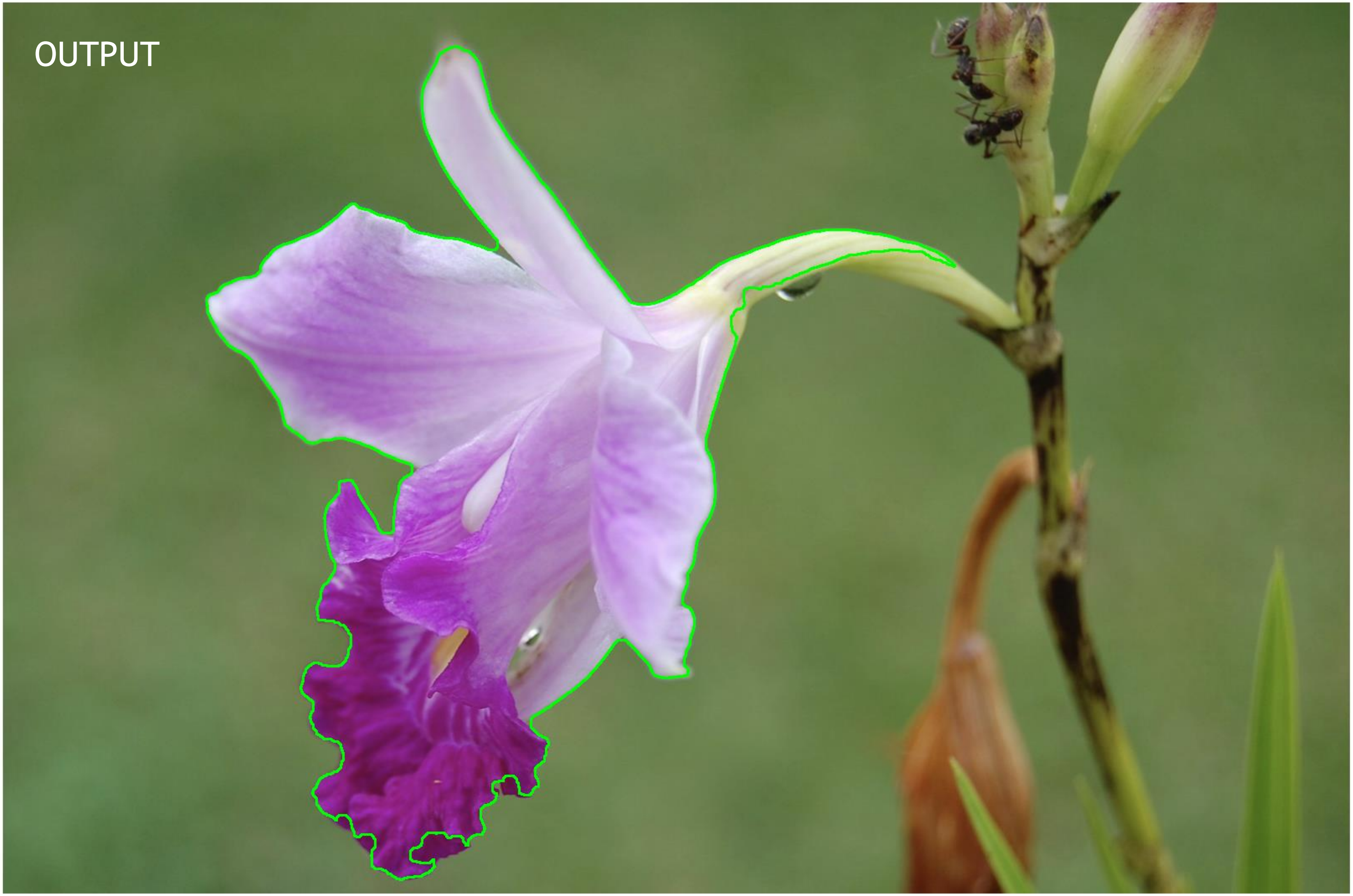
RGB > Gris
(normal)



RGB > Gris
(adaptive)
HCI

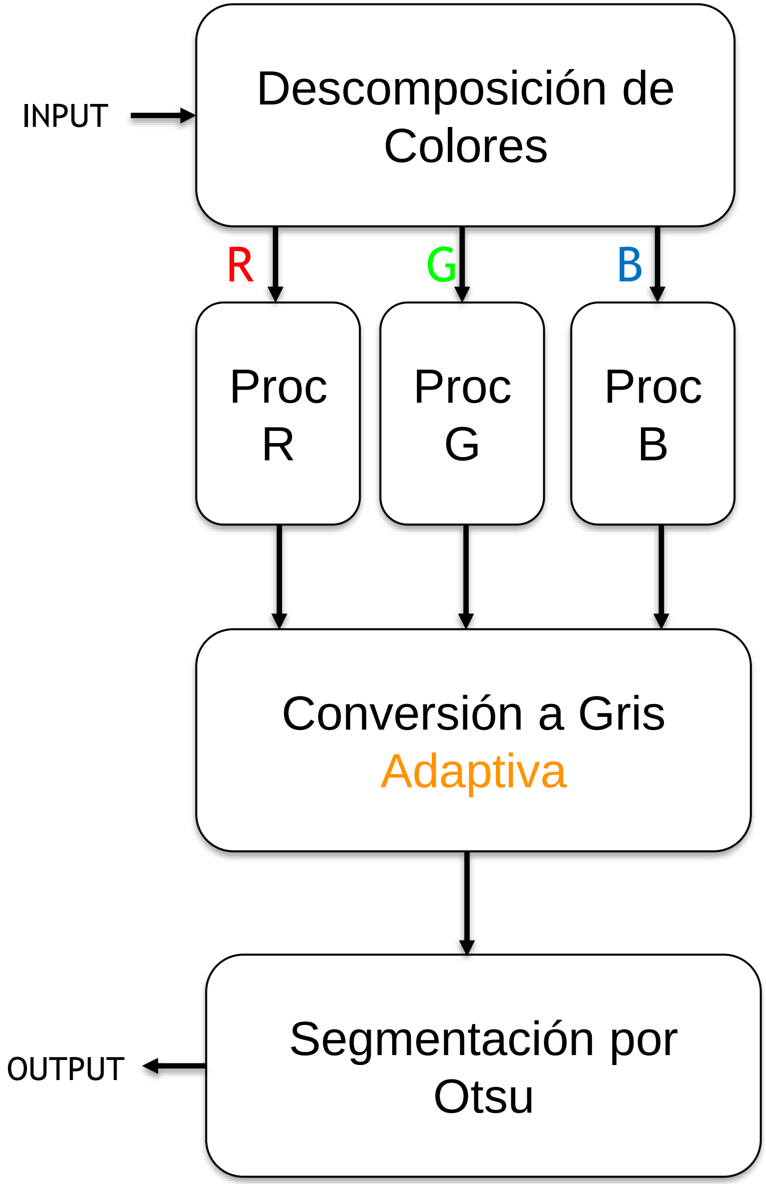

Otsu

Morfología



OUTPUT

2. Adaptivo por umbral


INPUT

Descomposición de Colores

R

G

B

Proc R

Proc G

Proc B

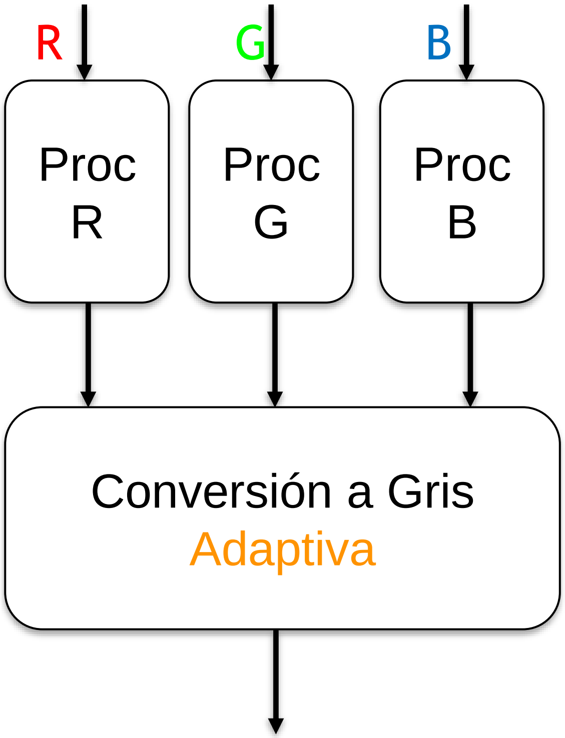
Conversión a Gris Adaptiva

Segmentación por Otsu

OUTPUT

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?

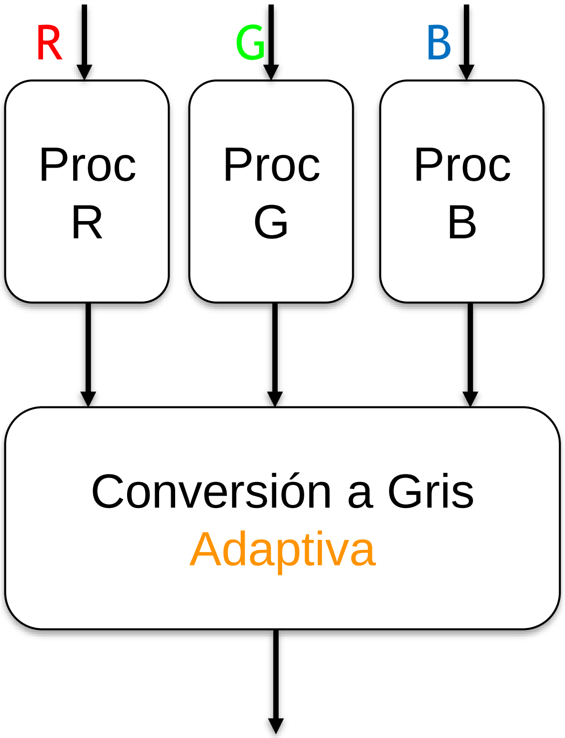
No

2. Adaptivo por umbral

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

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

$$\sigma_J^2(k_r, k_g, k_b) \rightarrow \max$$



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)

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Food Engineering 66 (2005) 353–360

JOURNAL OF FOOD ENGINEERING

www.elsevier.com/locate/jfoodeng

Segmentation of colour food images using a robust algorithm

Domingo Mery [a,*], Franco Pedreschi [b]

[a] *Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile,*
Av. Vicuña Mackenna 4860(183), Santiago de Chile, Chile

[b] *Departamento de Ciencia y Tecnología de Alimentos, Facultad Tecnologíca, Universidad de Santiago de Chile (USAH),*
Av. Ecuador 3769, Santiago de Chile, Chile

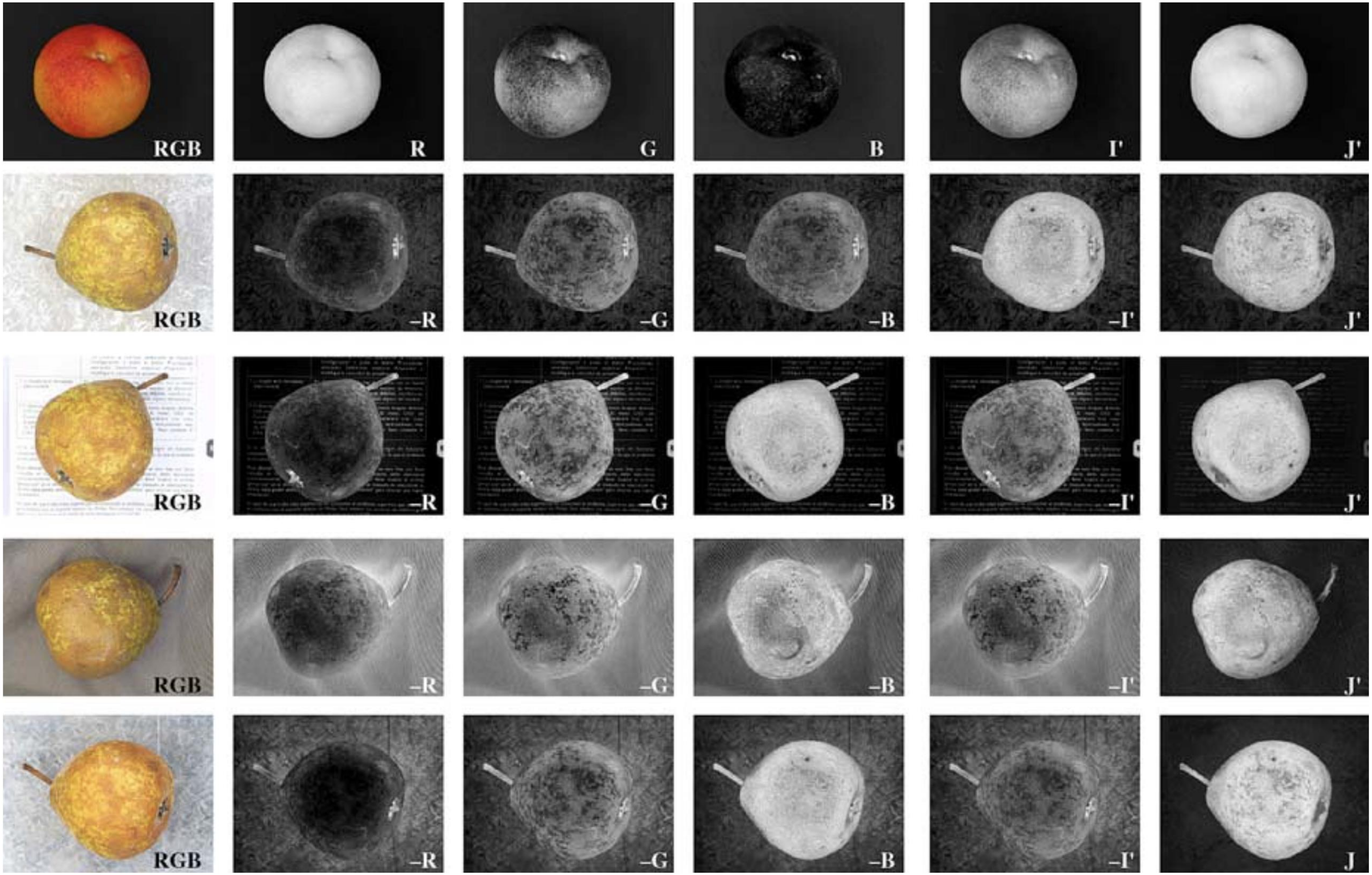
Received 14 October 2003; accepted 4 April 2004

Abstract

In this paper, a robust algorithm to segmenting food image from a background is presented using colour images. The proposed method has three steps: (i) computation of a high contrast grey value image from an optimal linear combination of the RGB colour components; (ii) estimation of a global threshold using a statistical approach; and (iii) morphological operation in order to fill the possible holes presented in the segmented binary image. Although the suggested threshold separates the food image from the background very well, the user can modify it in order to achieve better results. The algorithm was implemented in Matlab and tested on 45 images taken in very different conditions. The segmentation performance was assessed by computing the area A_z under the receiver operation characteristic (ROC) curve. The achieved performance was $A_z = 0.9982$.

Keywords: Image analysis; Image processing; Segmentation; Colour images

RGB R G B Gris HCI



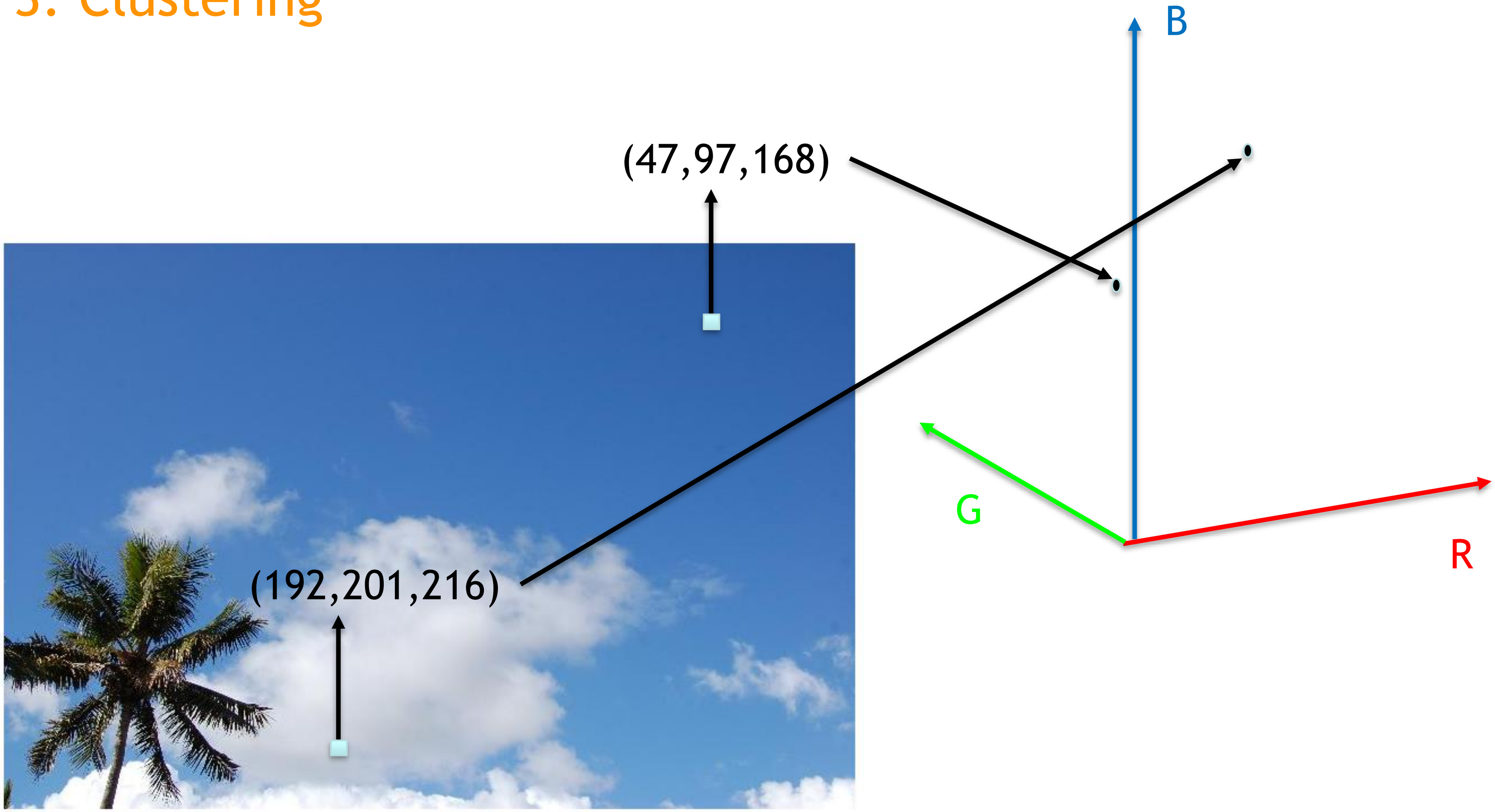
Tres métodos fundamentales:

1. Por canal de color

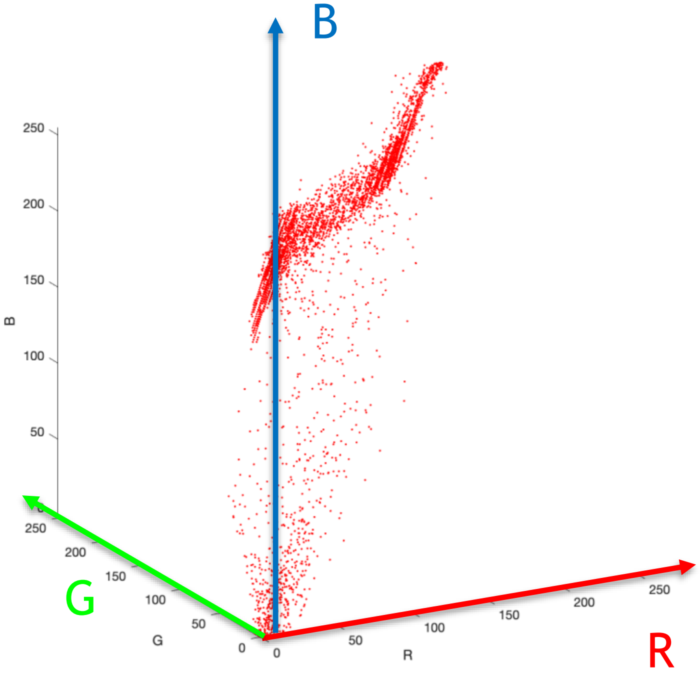
2. Adaptivo por umbrales

3. Clustering

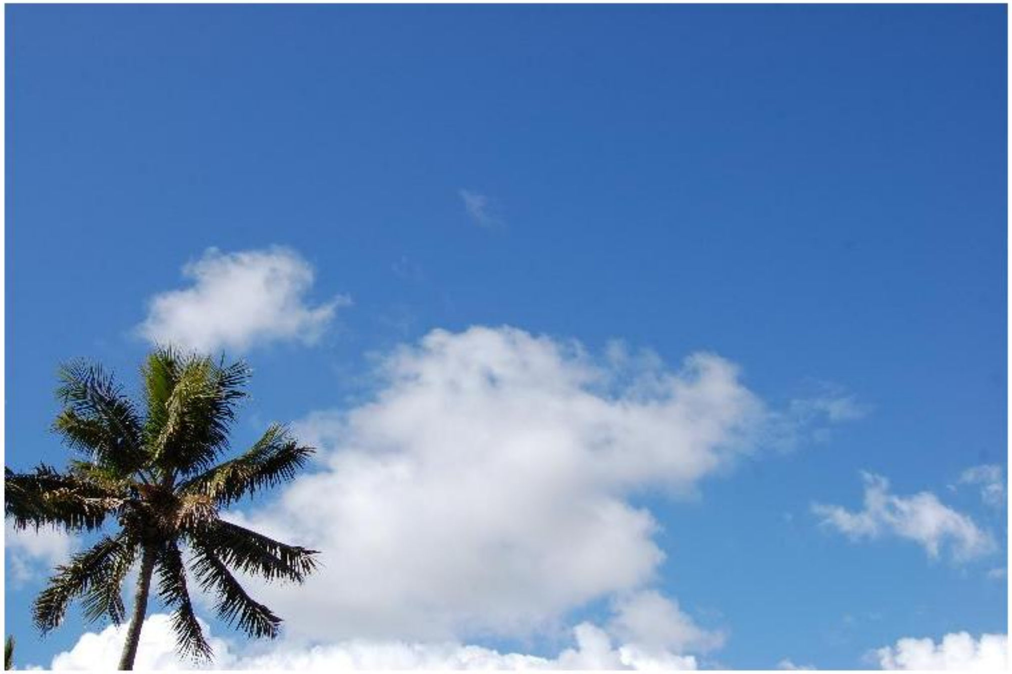
3. Clustering

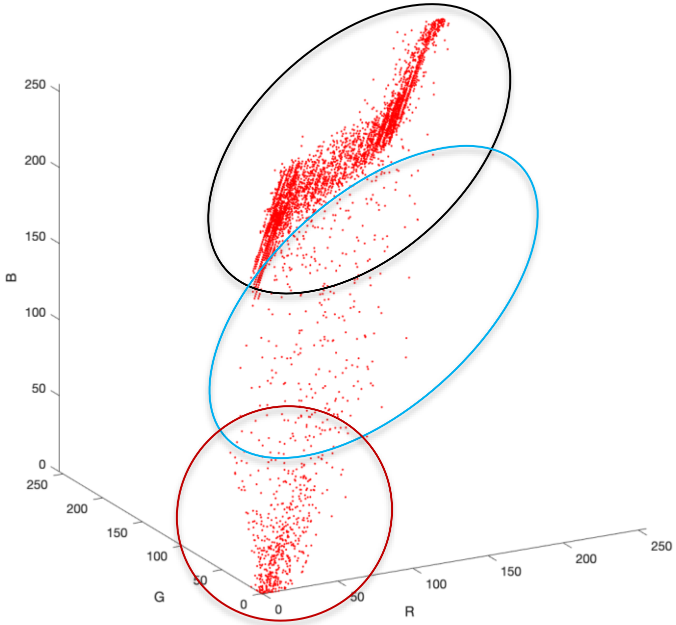


3. Clustering

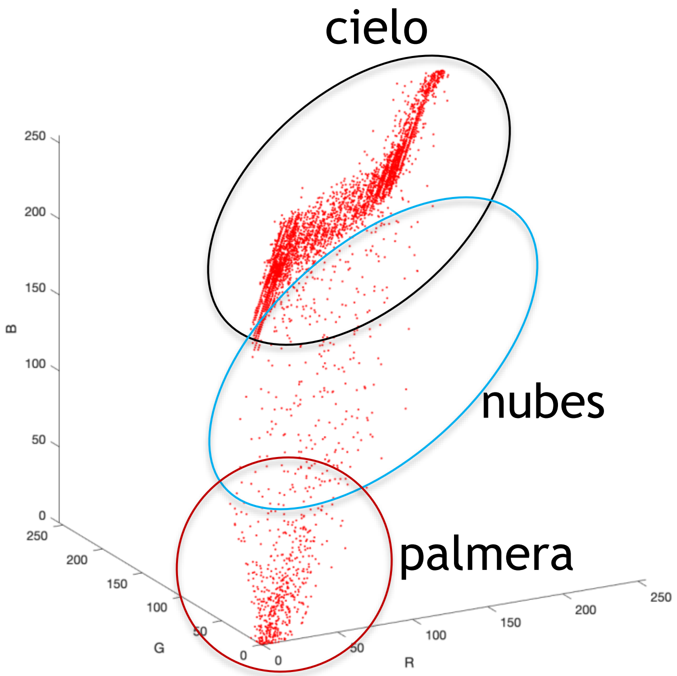


3. Clustering

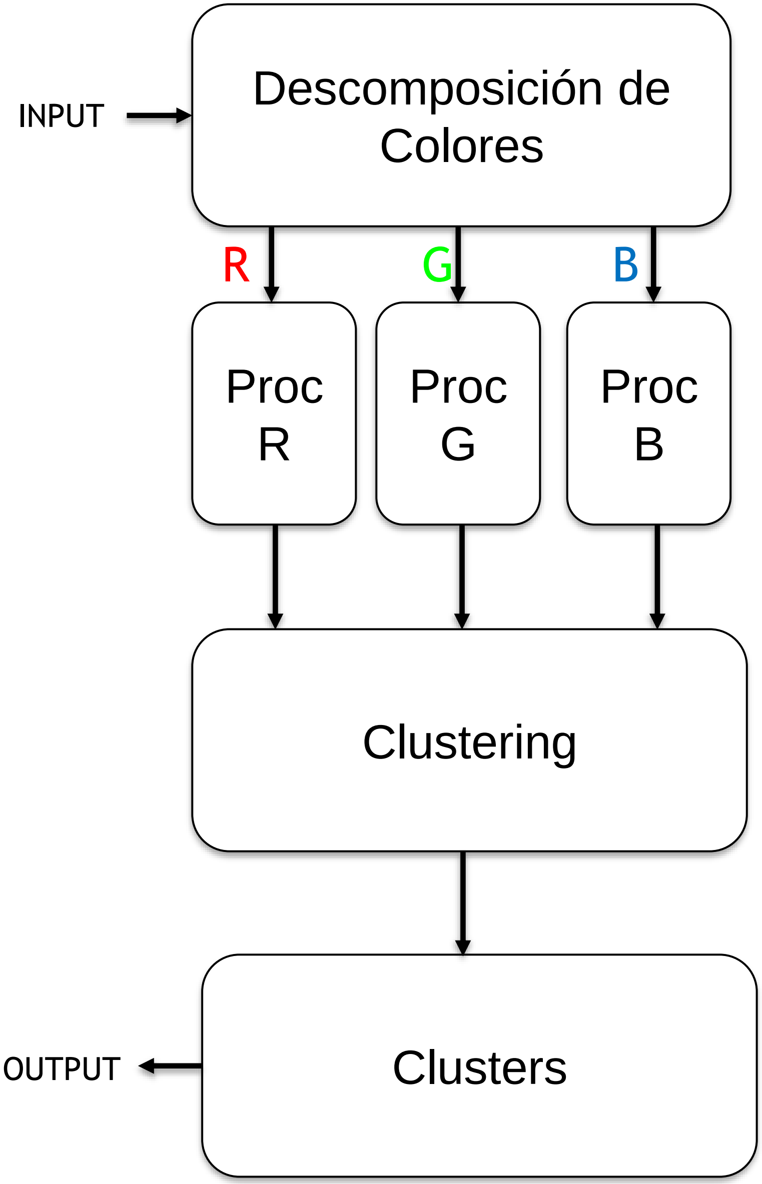




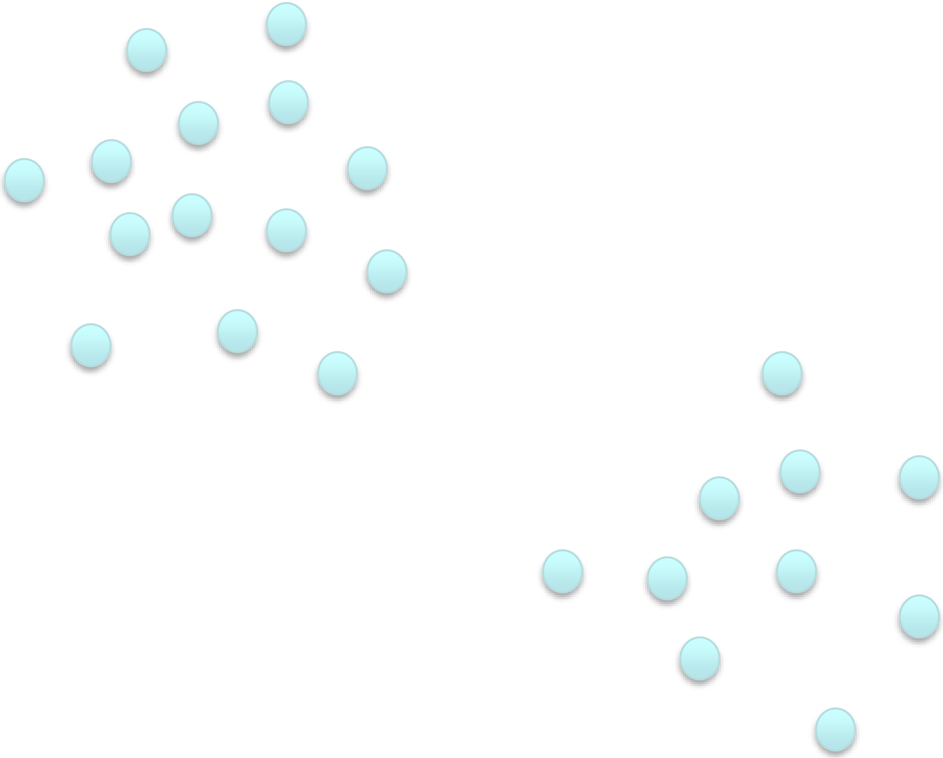
3. Clustering



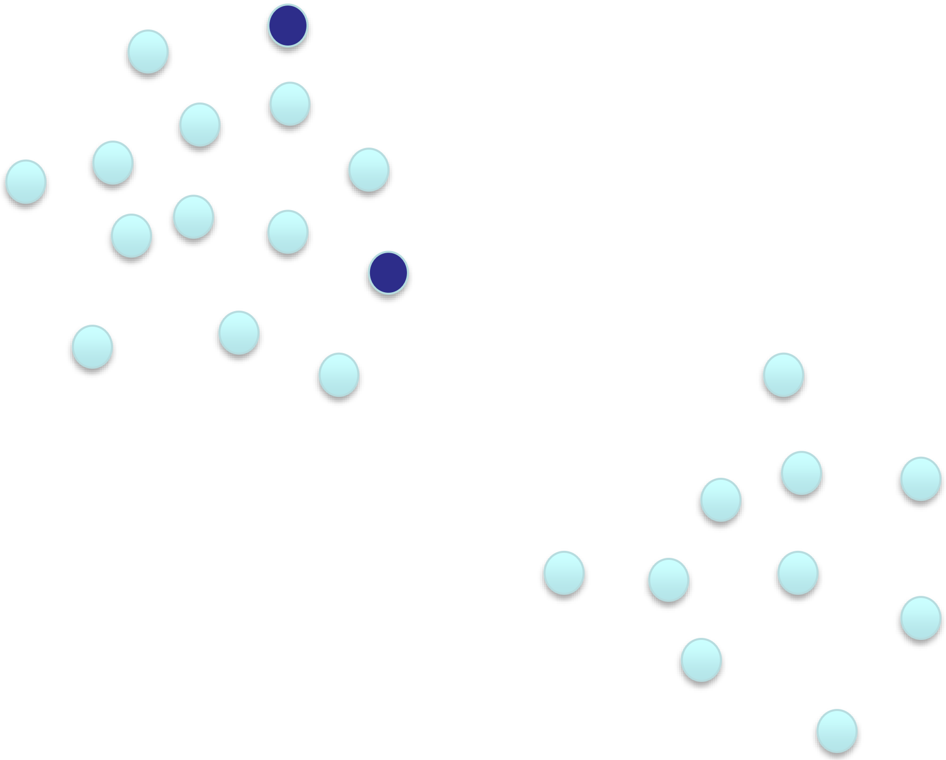
3. Clustering



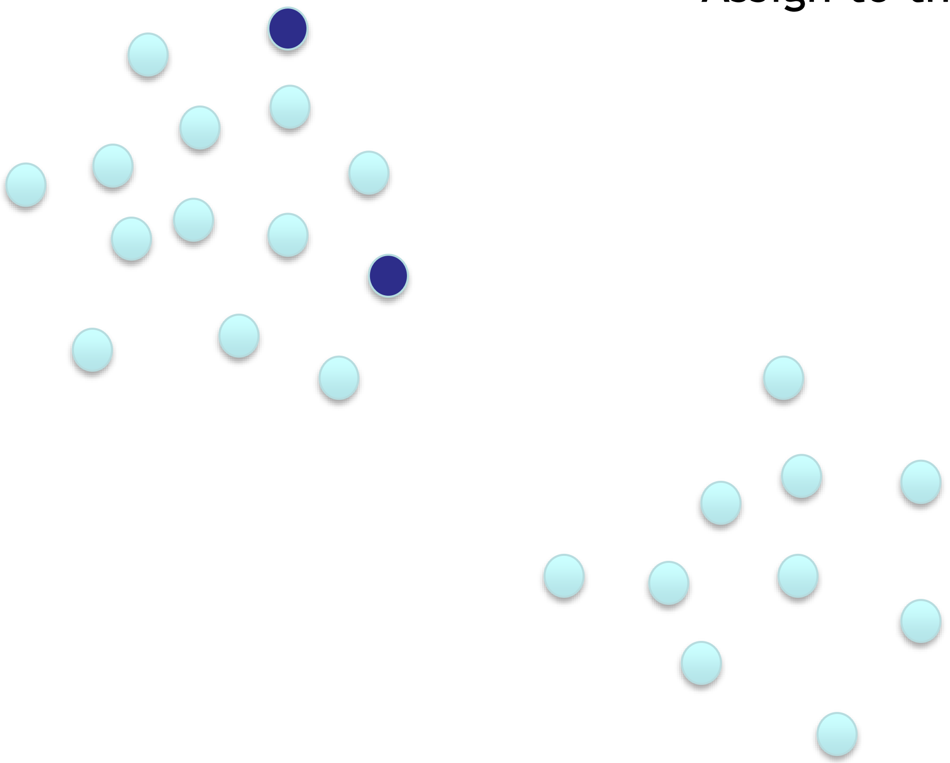
Clustering usando k-means

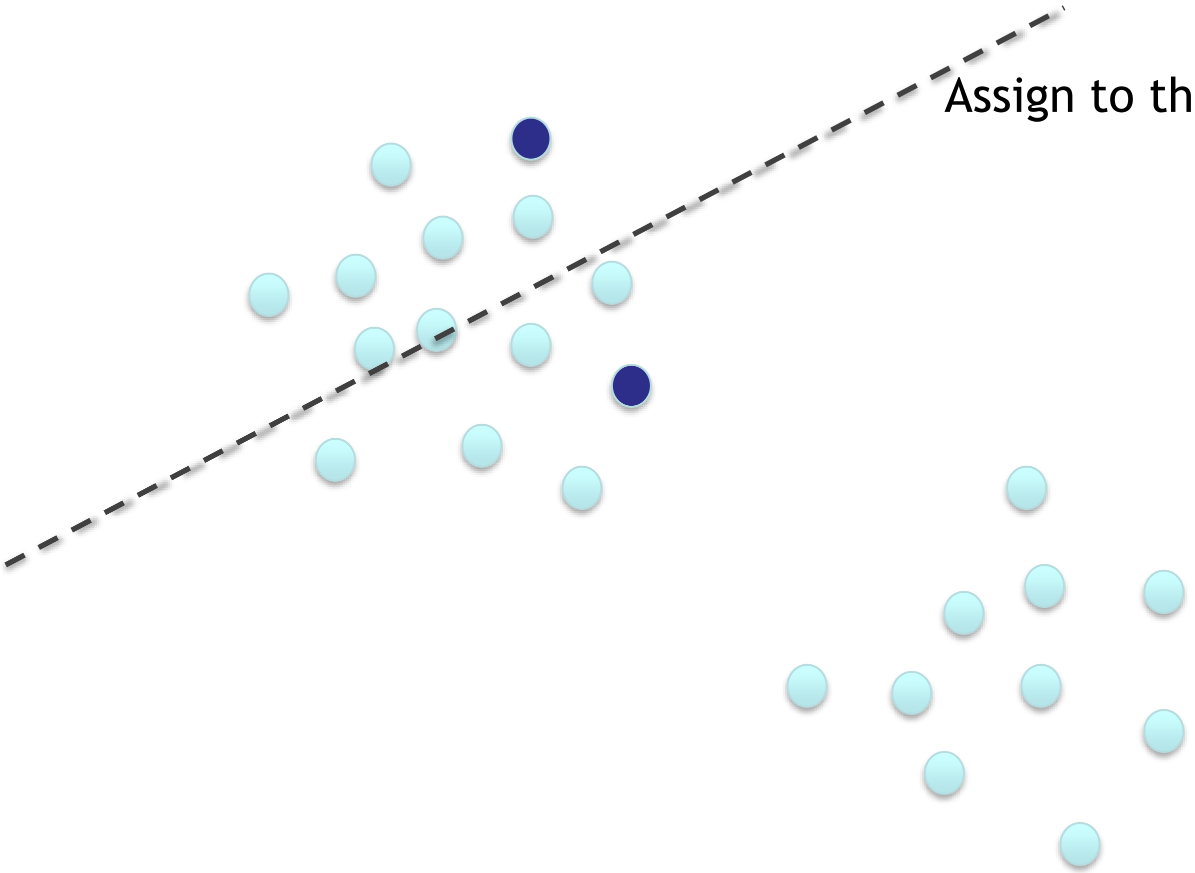


Choose random K=2 points (centroids)

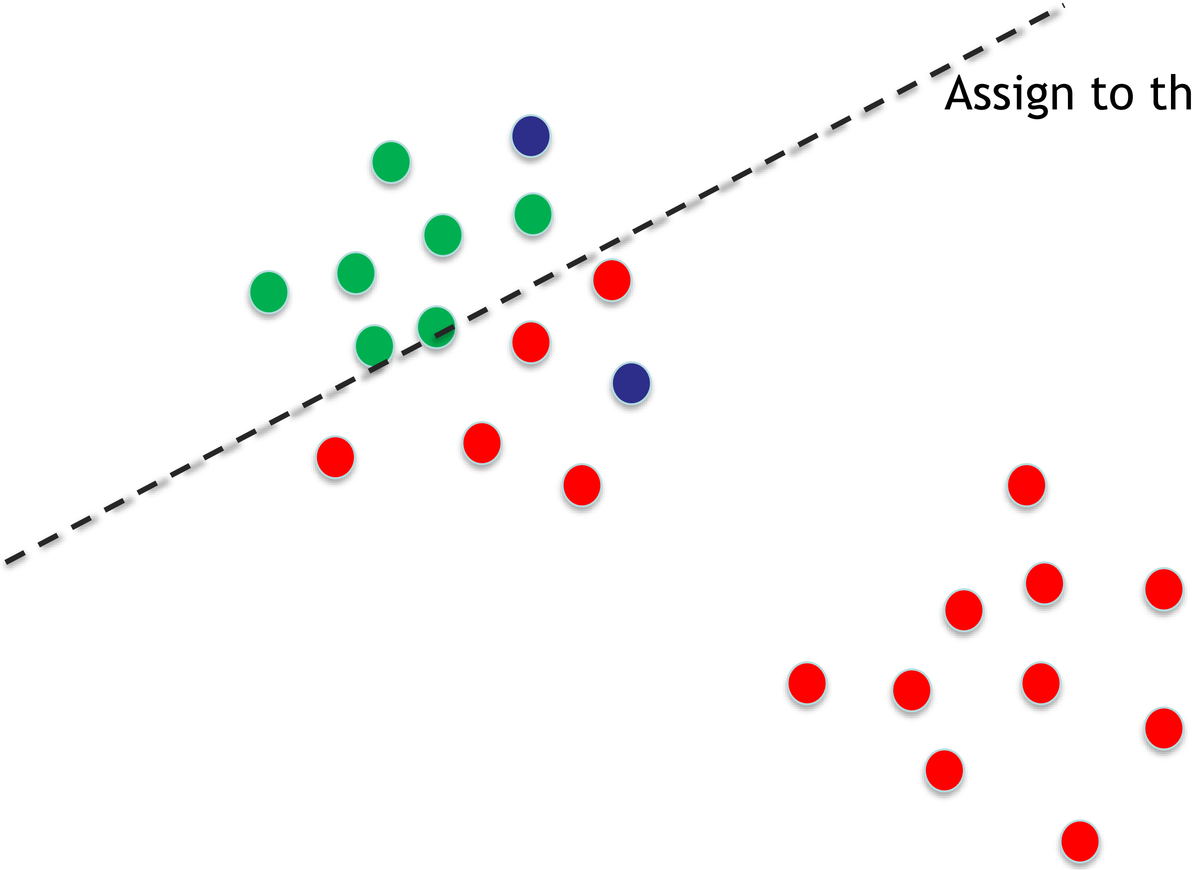


Assign to the nearest centroid

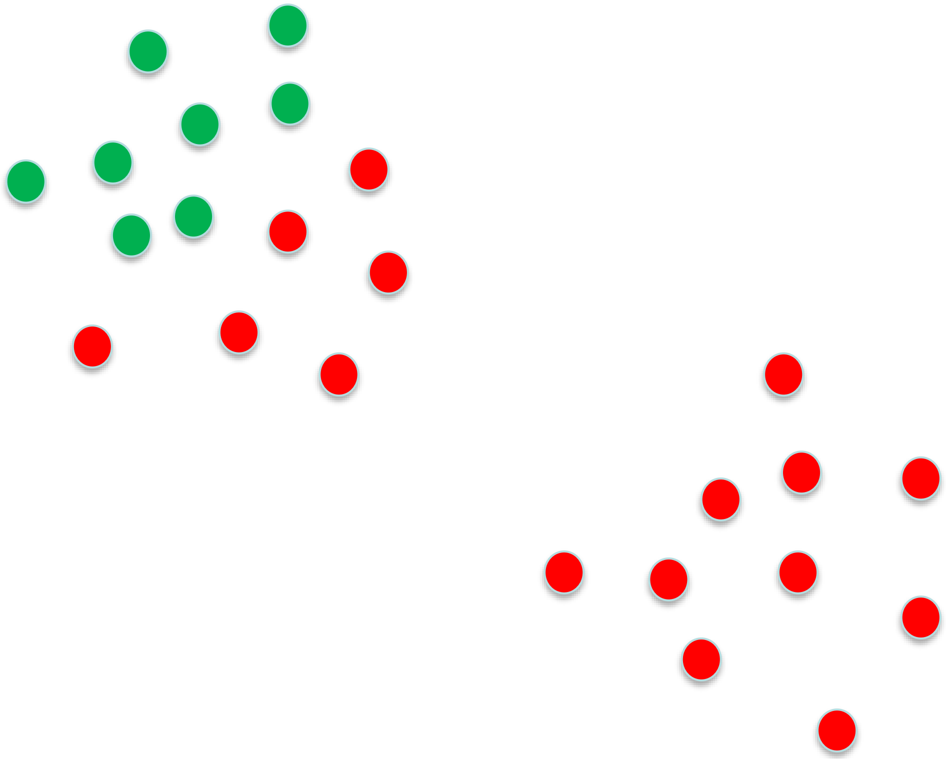




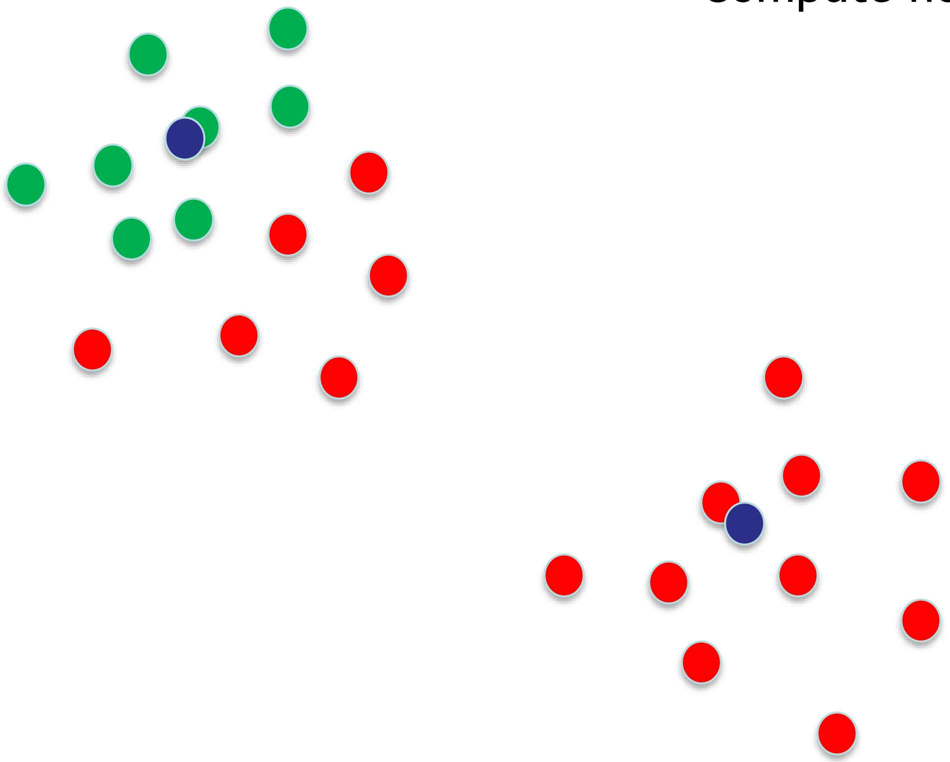
Assign to the nearest centroid

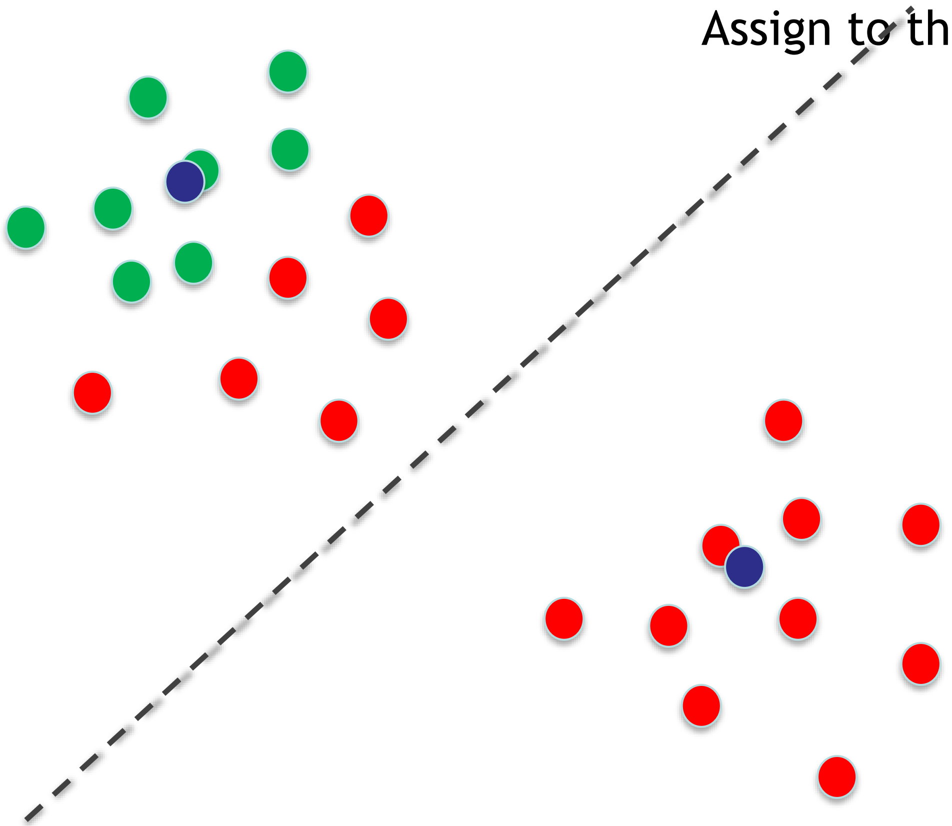


Assign to the nearest centroid



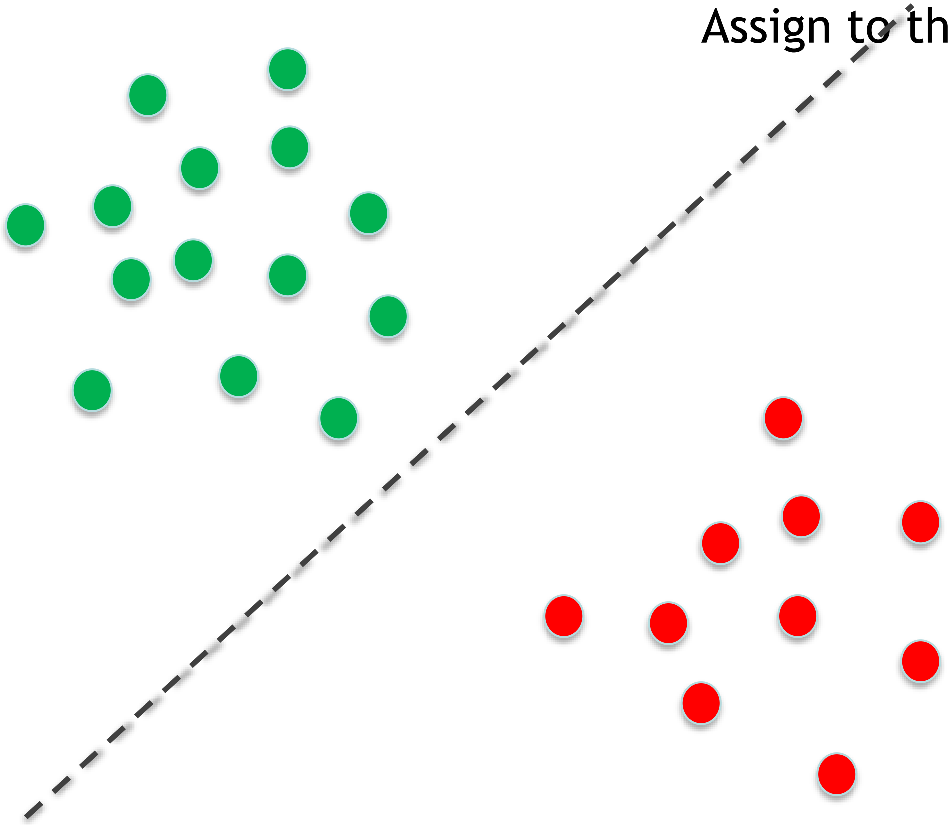
Compute new centroids



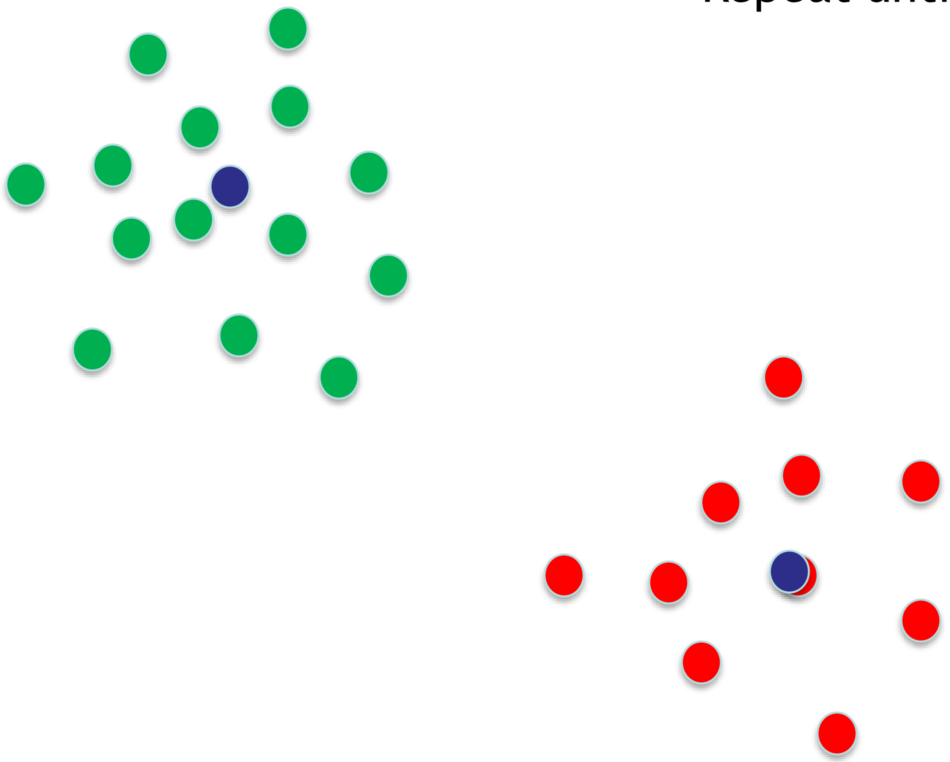


Assign to the nearest centroid

Assign to the nearest centroid

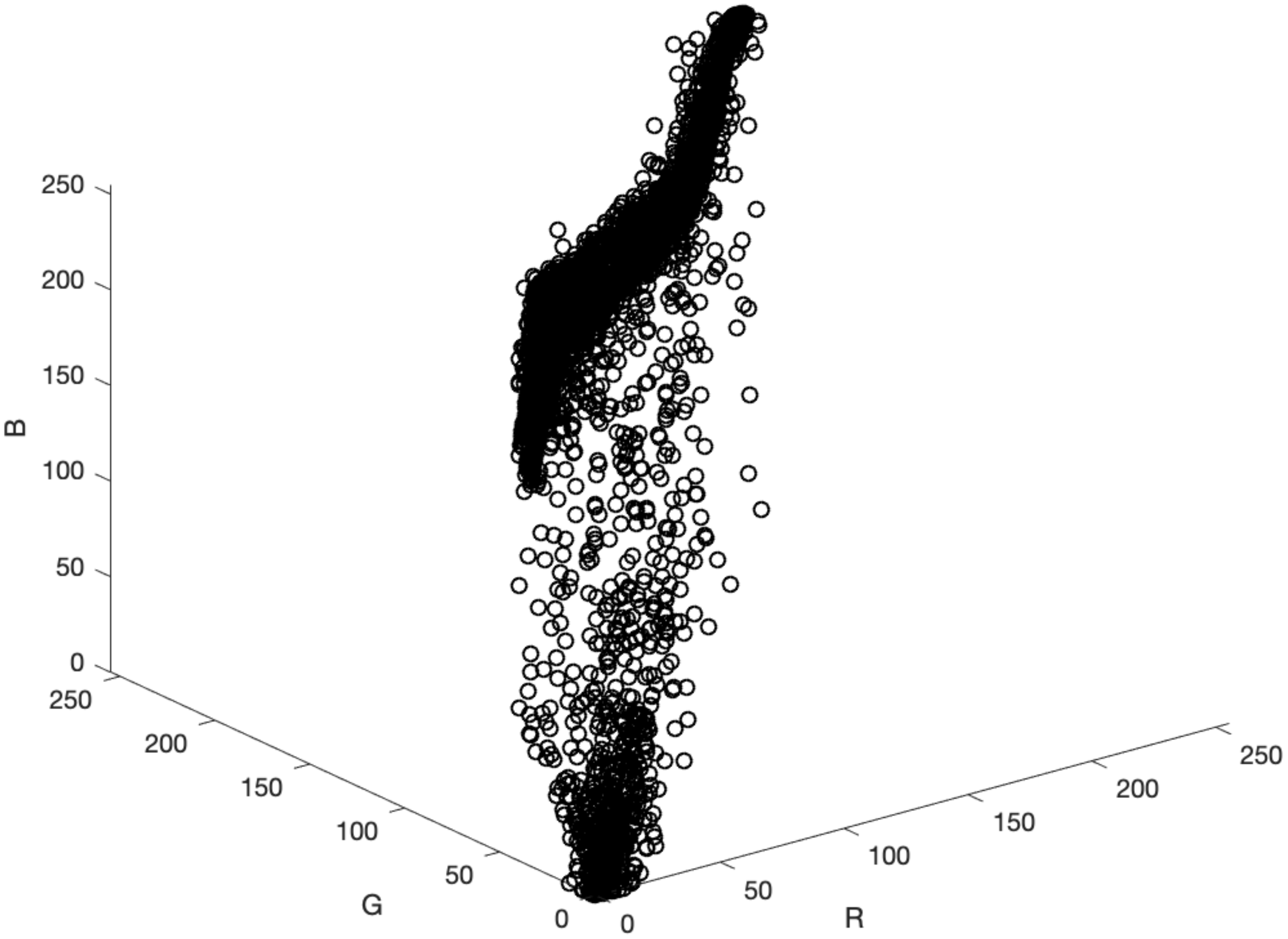


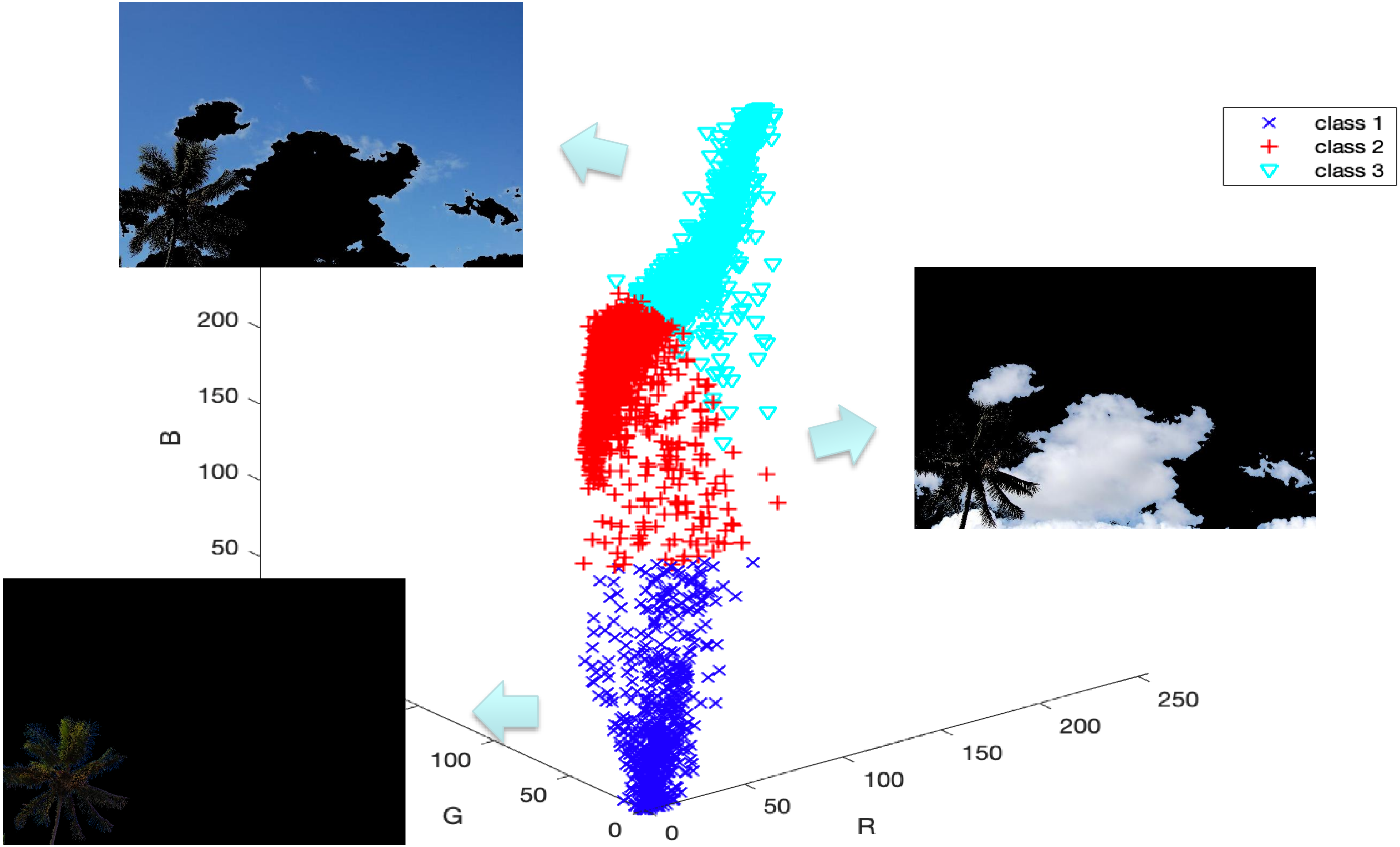
Repeat until convergence

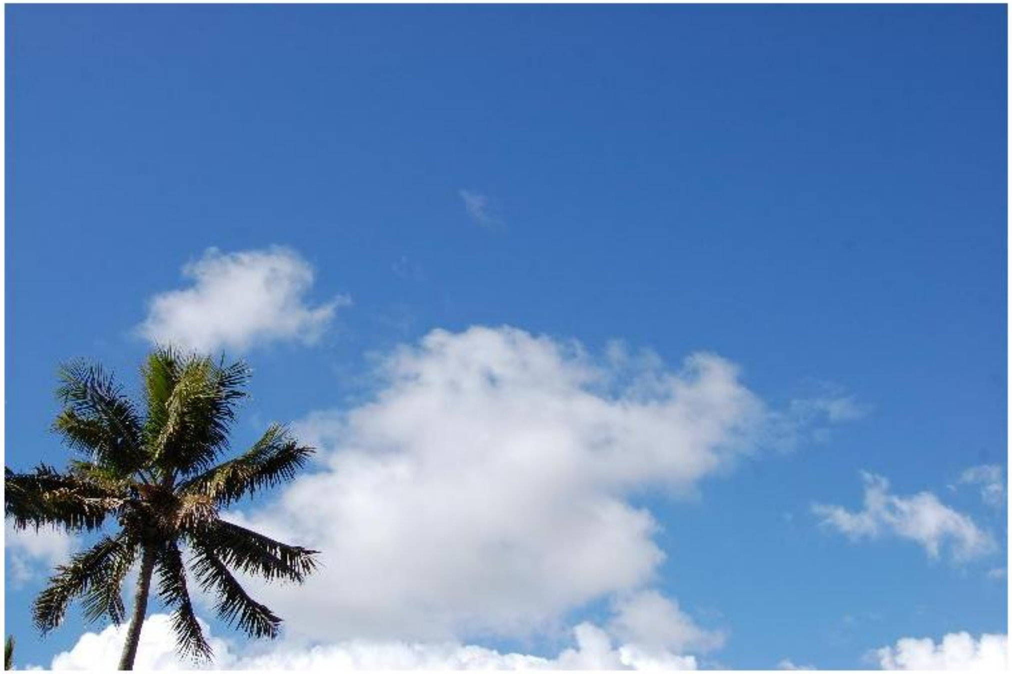


Algorithm K-Means:

```
1. Input Data $X = \{x_1, x_2, \dots x_N\}$ and number of clusters K
2. Centroids $\{c_1, c_2, \dots c_K\}$ = random K points of X
3. For each data point $x_i$
4.        Compute distance $d_{ij} = d(x_i,c_j)$  $i=1,\dots,N$, $j=1,\dots K$
5.        Assign $x_i$ to the nearest centroid: $y_i = \mathrm{argmin}_j\{d_{ij}\}$
6. Compute the new centroids of each cluster
              $c^*_j = \mathrm{mean}(x_i)$  for $y_i = j$
7.  if $c^*_j \neq c_j$ then $c_j = c^*_j$  go to step 3
8.  Output: $\{c^*_1, c^*_2, \dots c^*_K\}$ and $y_i$ for $i=1,\dots,N$
```

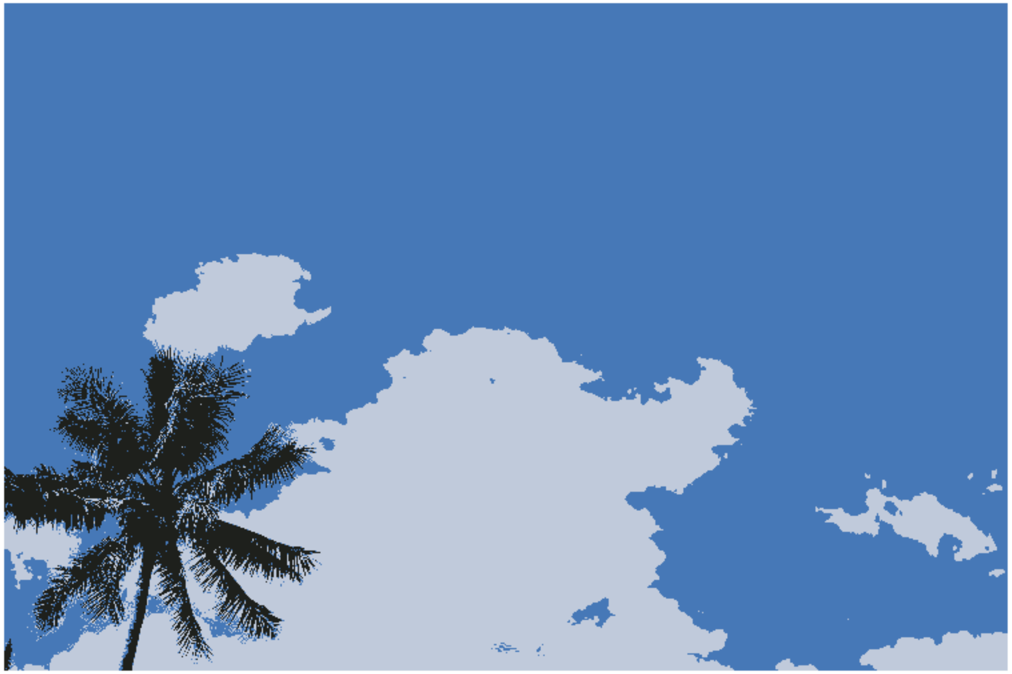

B
250
200
150
100
50
0
G
250
200
150
100
50
0
R
0
50
100
150
200
250


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



INPUT

OUTPUT



Tres métodos fundamentales:

1. Por canal de color

2. Adaptivo por umbrales

3. Usando clustering