

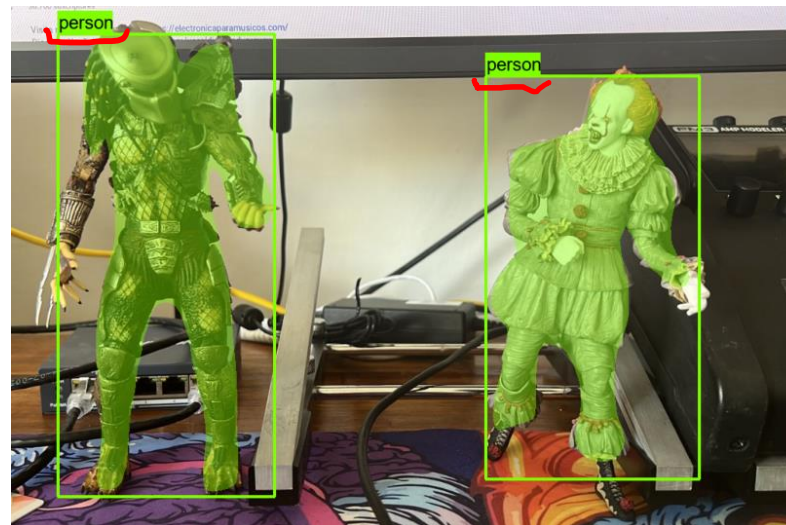
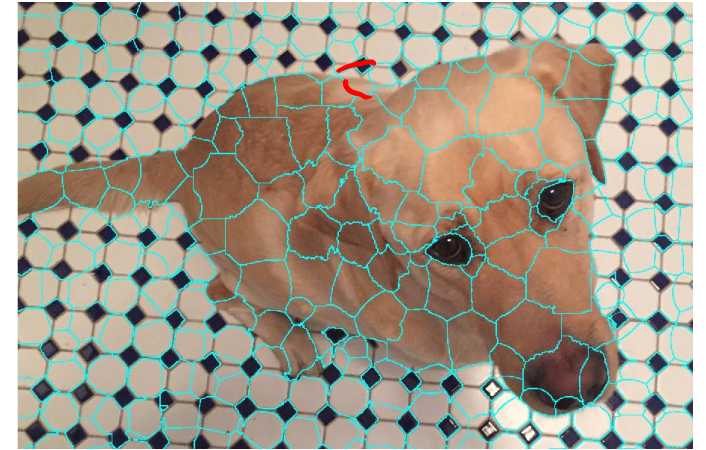
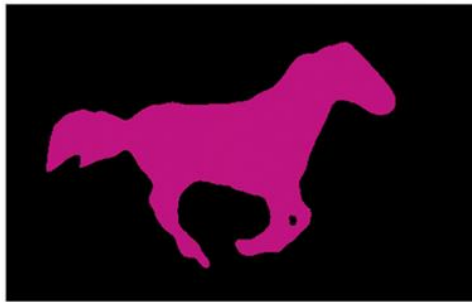
# Visión por Computadora I

Ing. Maxim Dorogov  
(mdorogov@fi.uba.ar)

Laboratorio de Sistemas Embebidos -FIUBA

# SEGMENTACIÓN

- La idea es “particionar” la imagen en una cantidad de regiones.
- Clasificación a nivel de pixel
- Segmentación figura/fondo
- Superpíxels
- Por instancia o semántica
- De contornos (watershed)



Segmentación por instancia: Mask R-CNN



UNET

**Dataset:** Berkeley Segmentation Data Set



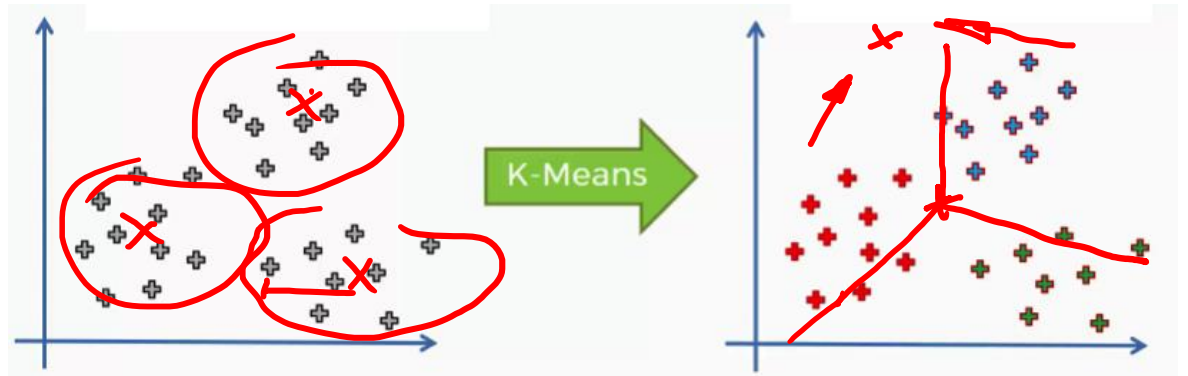
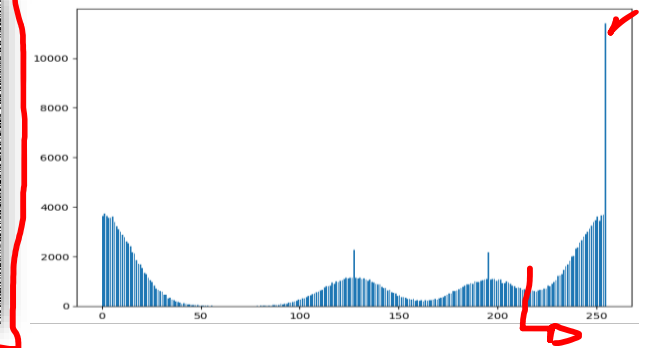
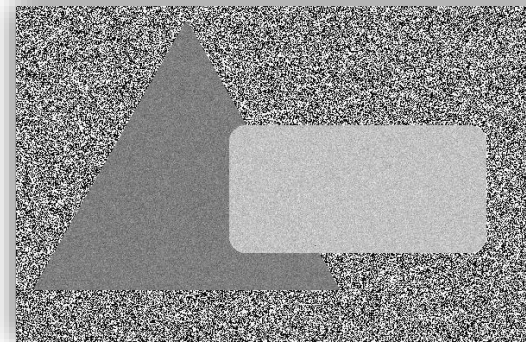
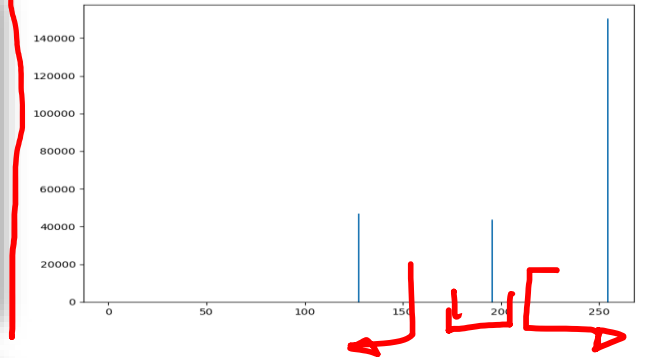
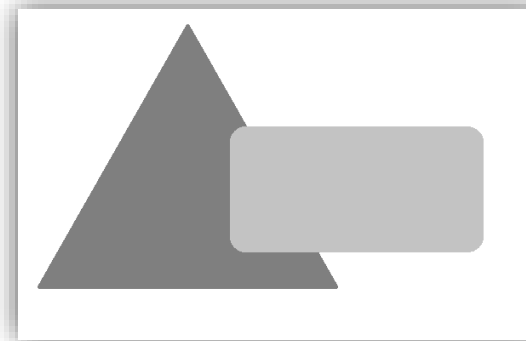


# SEGMENTACIÓN

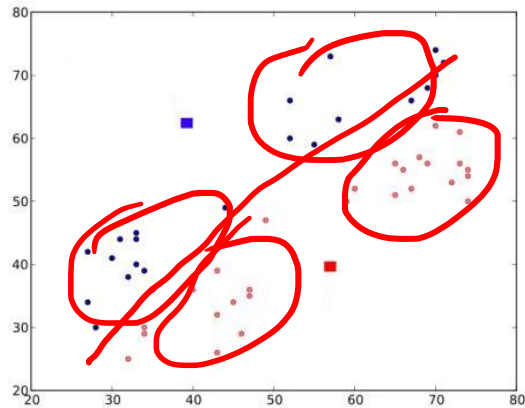
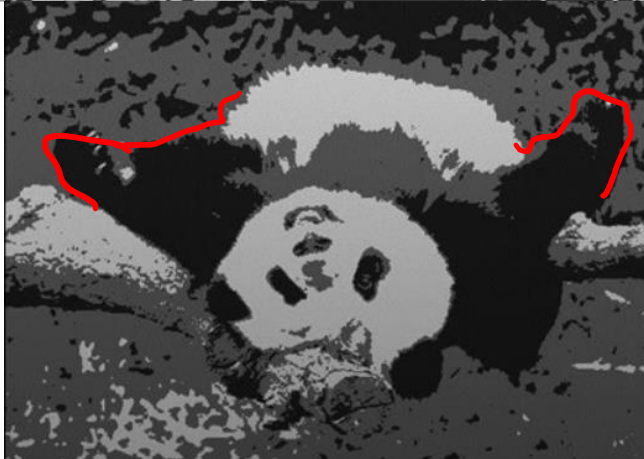
- Lo que buscamos es “encontrar clústers” a los que corresponda cada región.
- Los mejores clústers son los que minimizan las SSD entre todos los puntos y el centro del clúster más cercano

$$SSD = \sum_{\text{clúster } c_i} \sum_{p \in C_i} \|p_j - c_i\|$$

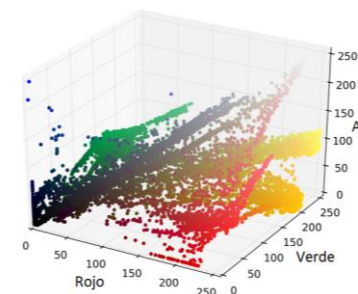
- Problema...el huevo o la gallina
  - Si conocemos los centros de los clústers podemos asignar los puntos que corresponden al mismo
  - Si conocemos las poblaciones podemos calcular cuáles son los centros del clúster
- Posible solución – K-means
  - Inicializar arbitrariamente los centros de las poblaciones (elegir cuántos)
  - Determinar los puntos correspondientes a cada clúster (para cada  $p_j$  encontrar el  $c_i$  más cercano y poner a  $p_j$  en el clúster  $c_i$ )
  - Dados los puntos de cada clúster encontrar el nuevo  $c_i$  (la media de la población)
  - Si algún  $c_i$  se movió volver al paso 2



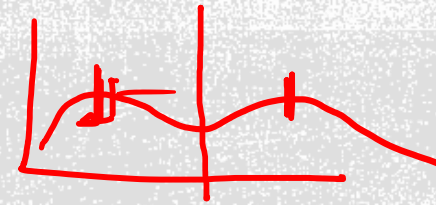




Intensidad  
Pos Y  
Pos X



# K-MEANS



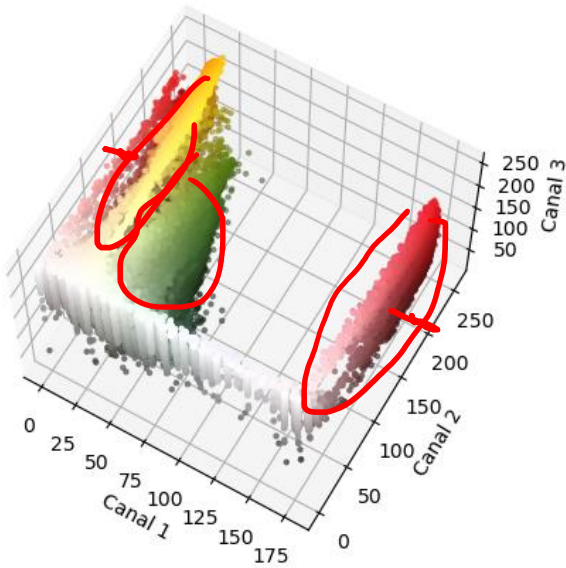
- Hay que indicar cuántos grupos (clústers) se busca segmentar
- ✓ Hay métodos para intentar detectar cuántos grupos "hay" en la imagen, pero el método fundamentalmente asume que ya lo sabemos.
- ✓ El agrupamiento se puede pensar como una cuantización del espacio de características (en el caso del panda, niveles de gris)
- ✓ En lugar de agrupar intensidades de gris, podemos trabajar con el espacio de color
- También podríamos agrupar en un espacio de características (features) que involucre la posición espacial. Por ejemplo, en escala de grises (Intensidad + Pos X + Pos Y)
- Pros
  1. Método simple ✓
  2. Converge a mínimos locales de la función de error
- Cons
  1. Consumo de memoria ✓
  2. Necesidad de elegir K ✓
  3. Sensible a la inicialización ✓
  4. Sensible a outliers
  5. Solo encuentra dentro de una región esférica del espacio de características ✓

E.M.

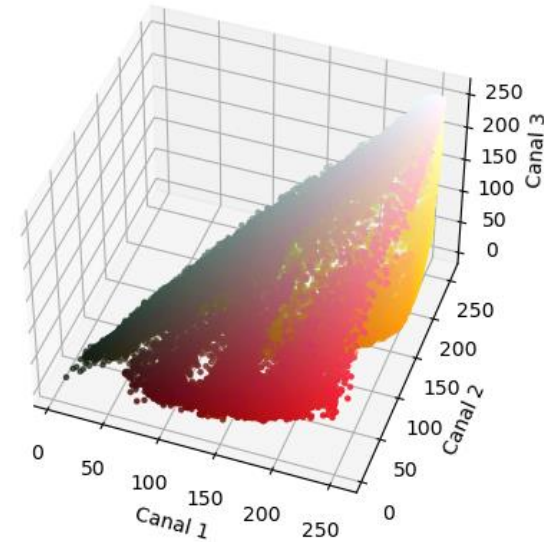
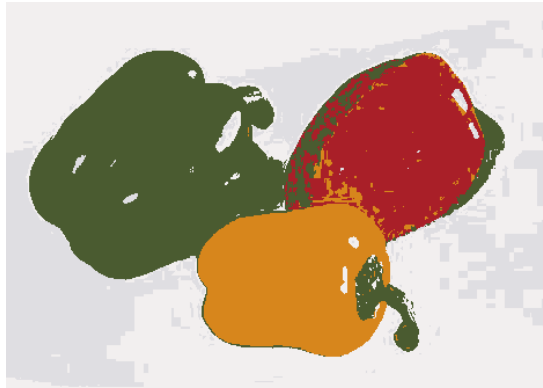


# K-MEANS

Ejemplo: Segmentación de una misma imagen representada con diferentes espacios de color y su distribución en el espacio de características.



Espacio: HSV



Espacio: RGB



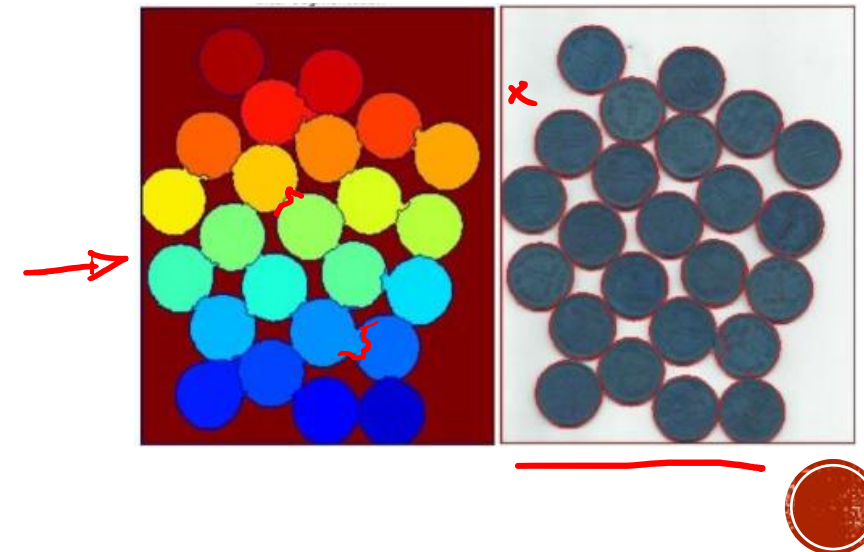
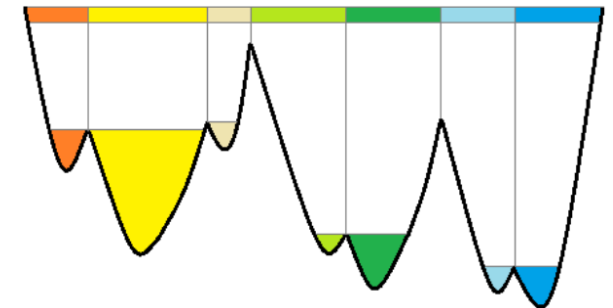
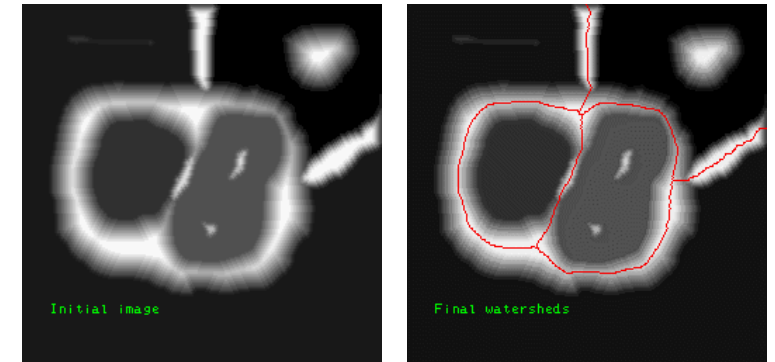
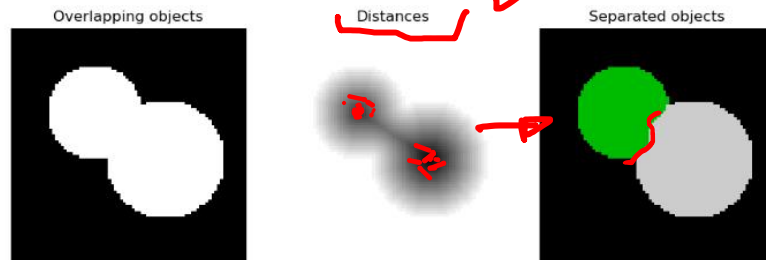


# WATERSHED

- La idea del método es segmentar la imagen en varios “cuencos” acumuladores.
- El método busca comenzar a inundar el paisaje de la imagen (pensada como superficie topográfica) en todos los mínimos locales y comenzar a etiquetar bordes a medida que las áreas (cuencos) de agua distinta comienzan a juntarse.
- Estos bordes se transforman en barreras (para que el agua de distintos colores no se termine mezclando) y se sigue inundando el paisaje hasta tapar todos los picos.
- Se podría utilizar este método aplicado a una imagen de magnitudes de gradiente (con suavizado previo) de manera de separar regiones homogéneas de crestas. De esta manera se podría utilizar también con imágenes color.
- La utilización de esta aproximación suele conducir a sobre segmentación por lo que una mejora al algoritmo consiste el agregado de marcadores (semillas) → **Marker Based Watershed**.

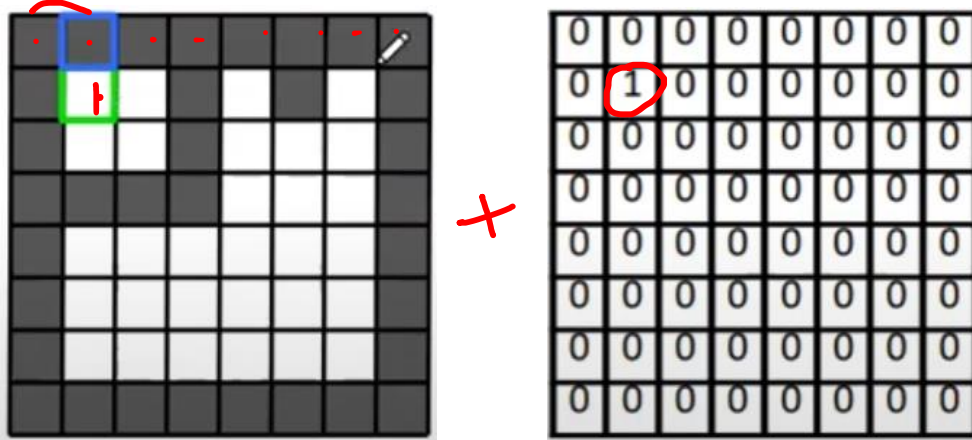
La indicación de marcadores puede ser:

- De manera interactiva por el operador, con un click de mouse.
- De manera automática, a través de una binarización, análisis morfológico y análisis de blobs.

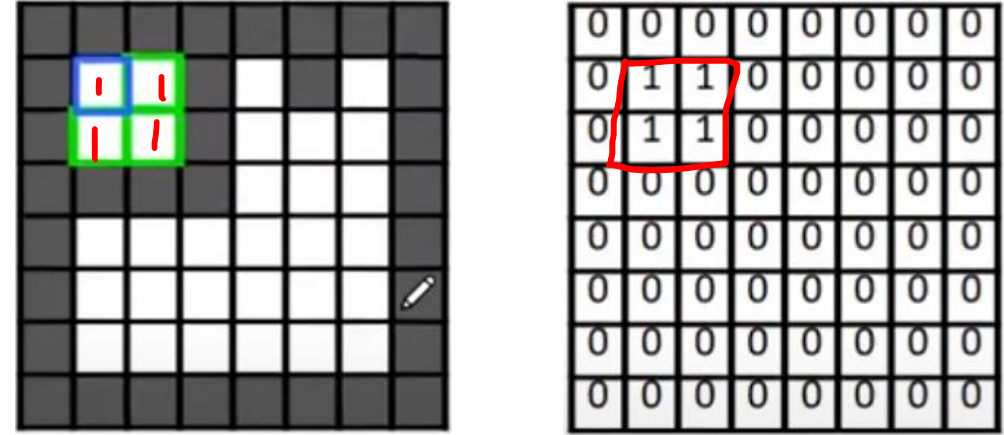


# FLOOD FILLING (SOLO IMÁGENES BINARIAS)

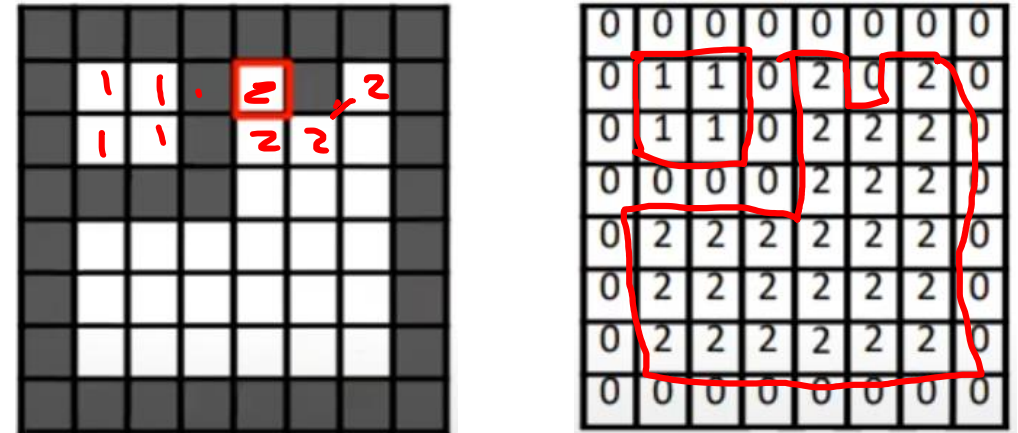
1.



2.



3.



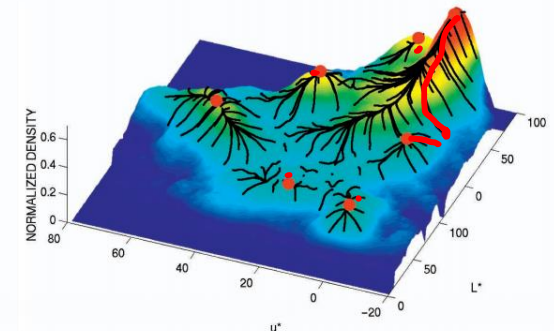
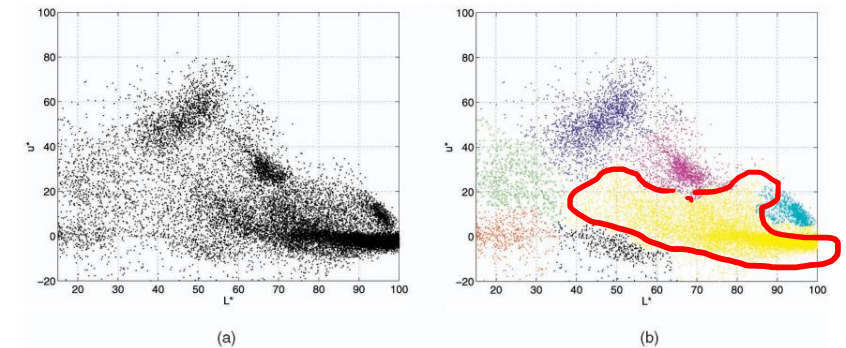
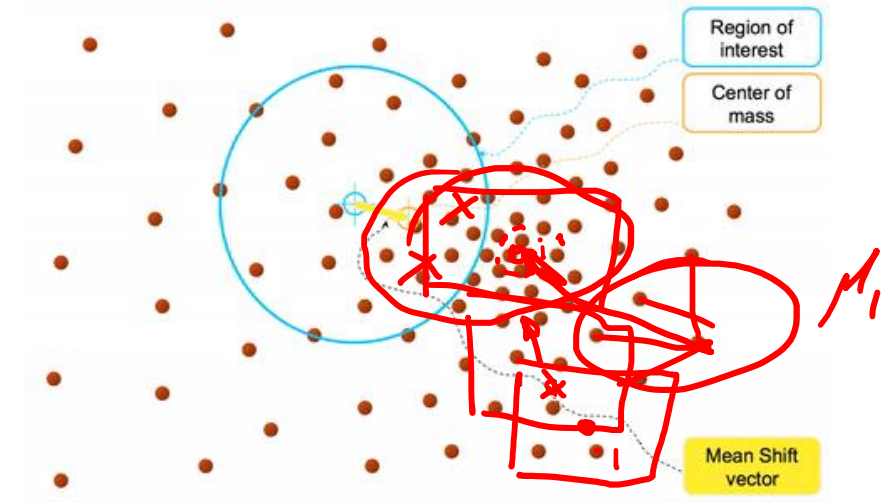
1. Encuentro un pixel que no es fondo
2. Si el pixel no tiene etiqueta asignada y no es vecino de otro pixel etiquetado: Le asigno una nueva etiqueta.
3. Si sus vecinos no tienen etiqueta les asigno la misma que al pixel central (siempre y cuando no sean fondo)
4. Me desplazo a otro pixel y repito desde paso 1 hasta que no queden elementos sin etiquetar



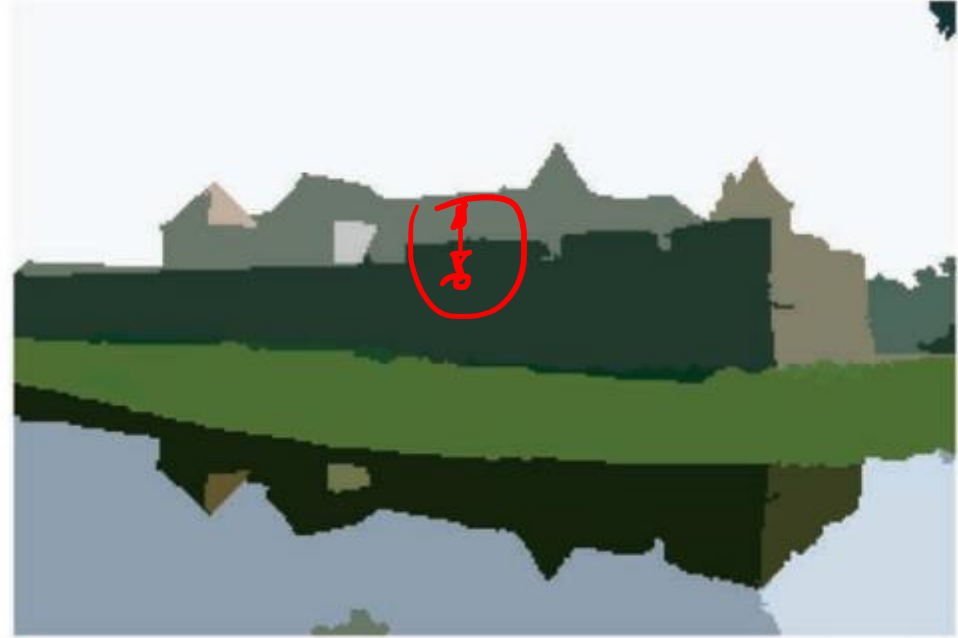
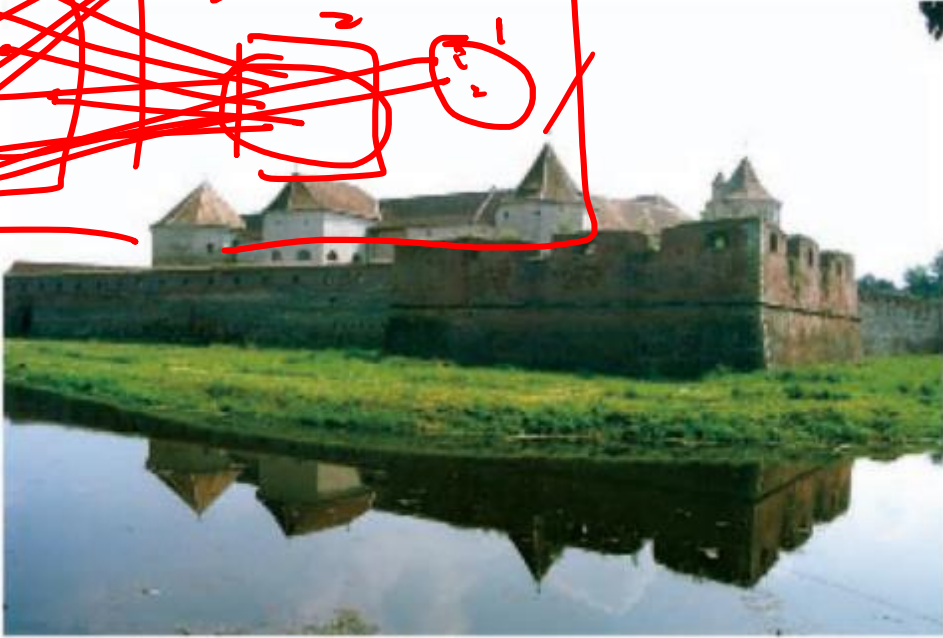
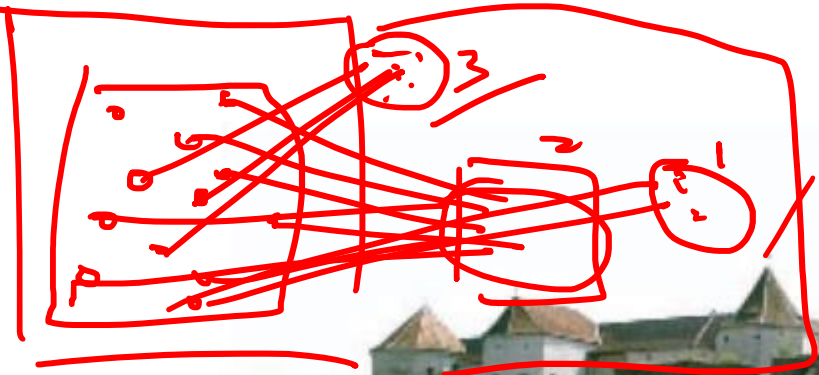
# MEAN SHIFT

- En K-means
  - Necesitamos conocer la cantidad de grupos (clústers)
  - Somos sensibles a las condiciones de inicialización
  - Asumimos una distribución esférica alrededor de las características
- Busca los “modos” o máximos locales de las densidades en el espacio de características
  1. Supone una distribución de probabilidad en algún espacio de características (color, gradientes, textura, ubicación, etc.)
  2. Toma una región de interés (normalmente pesada por una gaussiana)
  3. Inicializa las ventanas en cada punto de característica individual (en cada píxel)
  4. Calcula el centro de masa de esa región
  5. Mueve el centro a la nueva región (a través del “mean shift vector”)
  6. Luego de la convergencia une las ventanas (píxeles) que terminan cerca del mismo pico o modo de la función distribución

$P \times r$







# MEAN SHIFT

## Pros

1. Encuentra automáticamente los puntos de atracción ✓
2. Solo precisa elegir un parámetro (el tamaño de ventana) ✓
3. No asume que la imagen se encuentre dividida en clústers (regiones) ✓
4. Técnica genérica para encontrar múltiples modos (picos) ✓

## Cons

1. Selección del tamaño de ventana
2. No escala bien con el aumento de dimensiones en el espacio de características

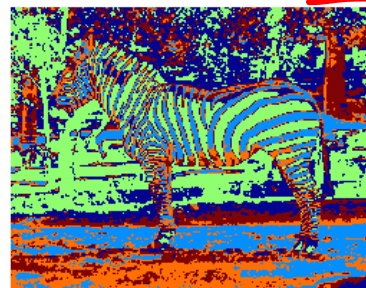


# SEGMENTACIÓN POR TEXTURAS

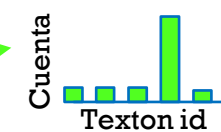
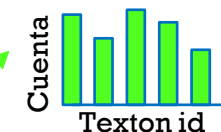
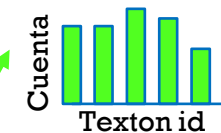
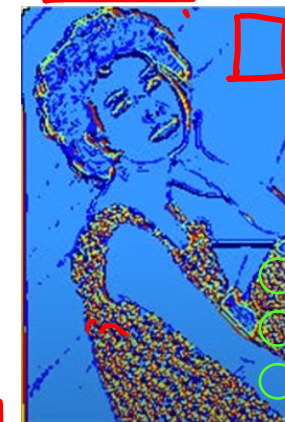
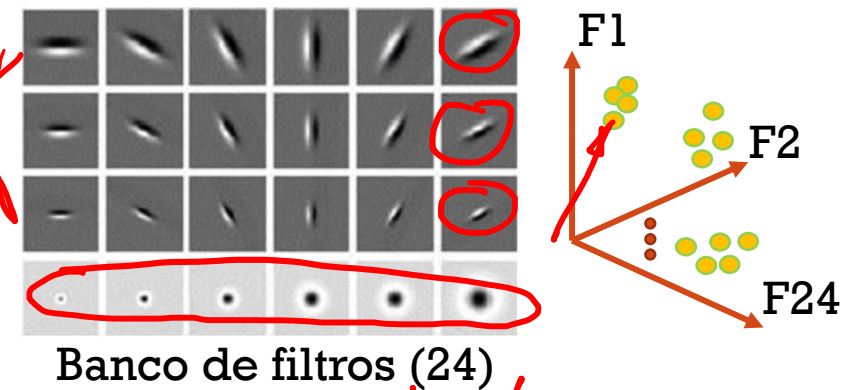
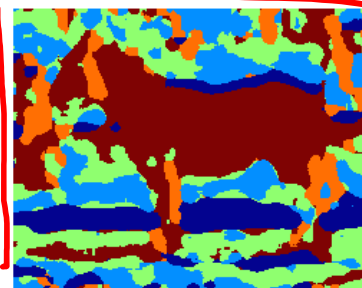
- Hay casos en que el espacio de características no puede basarse en colores, intensidades o ubicación.
- En los casos de la derecha es evidente que las regiones serían más fácilmente separables utilizando texturas.
- La idea es agrupar dos veces seguidas
  1. En el espacio de características al aplicar los filtros vamos a agrupar (por ejemplo con K-means) y nos vamos a referir a cada grupo como "texton" (Béla Julesz, 1981)
  2. Describimos texturas en ventanas como función de un histograma de textons y agrupamos en el espacio de histogramas
- Otra opción es usar descriptores LBP (Local Binary Pattern), también agrupando espacios de histogramas.



k-means (k=5), feature: rgb



k-means (k=5), feature: texton





# OTROS MÉTODOS DE SEGMENTACIÓN

## Graph Cut ("GrabCut": interactive foreground extraction using iterated graph cuts)

- La segmentación de imágenes puede realizarse pensando a la imagen como un grafo (en relación a la conexión de sus píxeles)
- La idea es generar con cada píxel un grafo conectado completamente
  - Un nodo (vértice) para cada píxel
  - Un enlace para cada par de píxeles  $\langle p, q \rangle$
  - Un peso  $w_{p,q}$  para cada enlace (basado en su "afinidad": color, posición, etc.)
- Luego, la idea es "cortar" los enlaces con menos peso (menor afinidad)
  - Píxeles similares quedarán en las mismas conexiones
  - Píxeles distintos quedarán en conexiones distintas

Esta familia de métodos requiere de "asistencia" para marcar las regiones a segmentar, en opencv esta implementado como cv.grabCut ✓

### Pros:

- Presenta un encuadre basado en grafos genérico
  - Por tanto existe la flexibilidad de elegir una función que calcule los pesos (afinidades) entre los nodos (y disociarla de la función de corte)

### Cons:

- El tiempo de cómputo puede ser elevado (matrices de afinidad de  $n \times n$  con  $n$  el número de píxeles)
- Requiere de asistencia humana, o artificial para marcar el objeto y ajusta la segmentación.



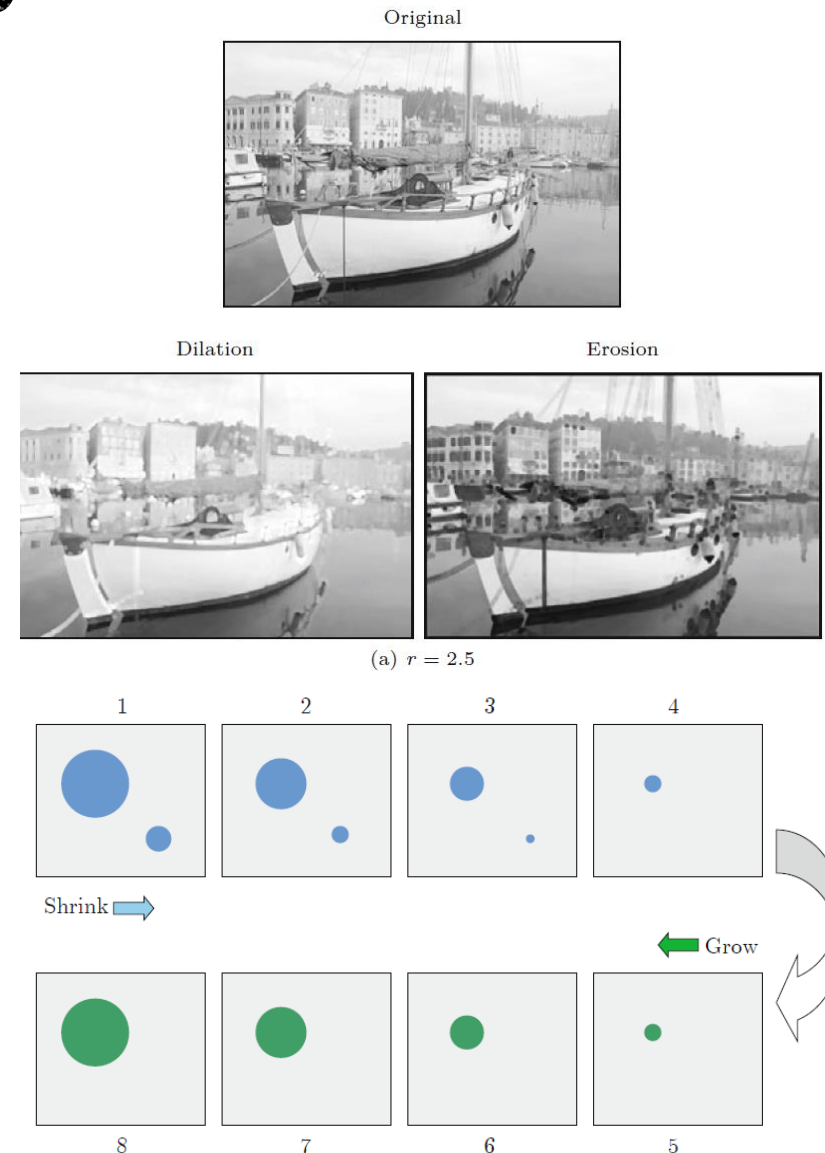
# PROCESAMIENTO MORFOLÓGICO

- Son operaciones no lineales que se aplican a imágenes binarias (se pueden generalizar para color o escala de grises) y modifican la estructura de los objetos presentes.
- Cada operación requiere de dos elementos (además de los parámetros propios de cada caso en particular):
  - Imagen a procesar
  - Kernel o elemento estructurante
- Las operaciones morfológicas se resumen a superponer el elemento estructurante con la imagen (similar a *template matching*, filtrado convolucional, etc...) y decidir si el pixel central se mantiene como objeto, valor 255, o se convierte en fondo, valor 0.

## Operaciones elementales:

- **Erosión:** Si todos los pixeles debajo del kernel valen 1 entonces el pixel central sobrevive, caso contrario se convierte en pixel de fondo, (se pone en cero)
- **Dilatación:** Si al menos 1 pixel debajo del kernel es distinto de cero entonces el pixel se pone en 1.

Estas operaciones se resuelven de manera eficiente mediante operadores lógicos AND y OR.





# PROCESAMIENTO MORFOLÓGICO

Algunas operaciones mas usadas, realizadas con un kernel cuadrado de 5x5:



Imagen original



Erosión



Dilatación



Cierre: Dilatación y erosión



Apertura: Erosión y dilatación



Gradiente: Dilatación - erosión

