Visión por Computadora I

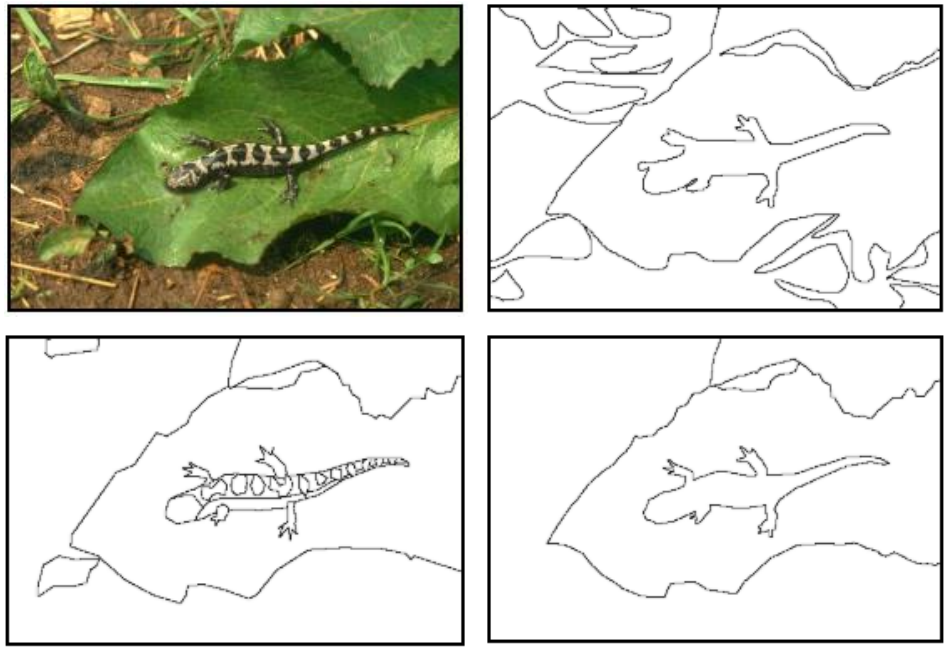
Ing. Andrés F. Brumovsky (abrumov@fi.uba.ar)

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

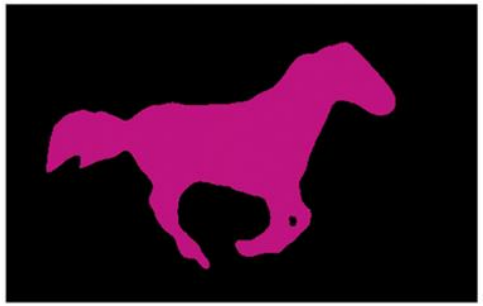
Laboratorio de Sistemas Embebidos -FIUBA

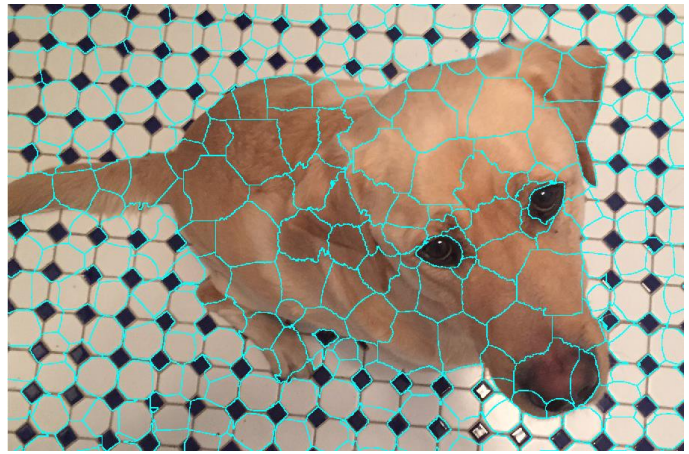
SEGMENTACIÓN

- La idea es "segmentar" la imagen en una cantidad de regiones.
- Clasificación a nivel de pixel
- Segmentación figura/fondo
- Superpíxels
- Por instancia o semántica



Berkeley Segmentation Data Set



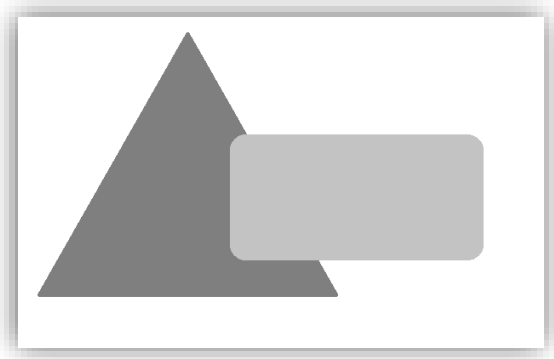


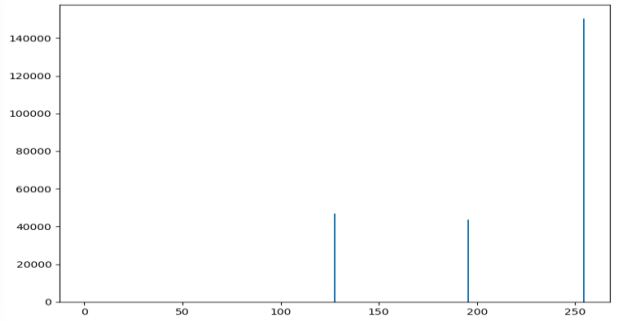
SEGMENTACIÓN

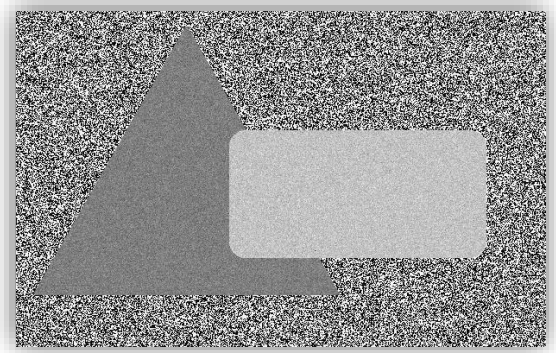
- Lo que buscamos es "encontrar clústers" a los que corresponda cada región.
- Los mejores clústers son los que minimizan las SSD entre todos los puntos y el centro del clúster más cercano

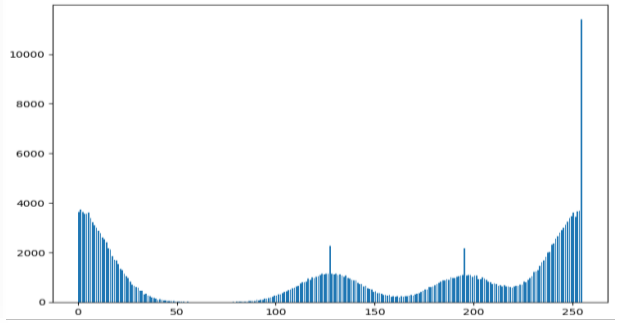
$$SSD = \sum_{clúster\ C_i} \sum_{p \in C_i} \|p_j - c_i\|$$

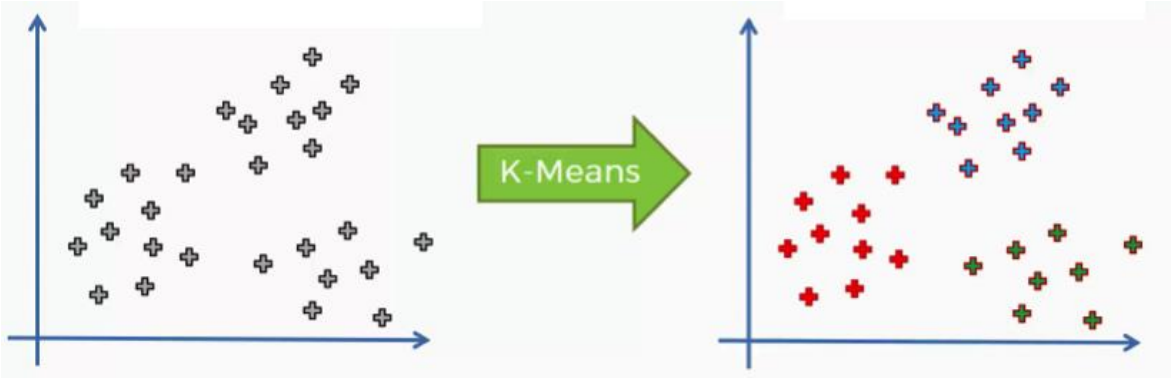
- Problema…el huevo o la gallina
  - Si conocemos los centros de los clústers podemos asignar los puntos que corresponden al mismo
  - Si conocemos las poblaciones podemos calcular cuáles son los centros del clúster
- Posible solución – K-means
  1. Inicializar arbitrariamente los centros de las poblaciones (elegir cuántos)
  2. 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)
  3. Dados los puntos de cada clúster encontrar el nuevo c_i (la media de la población)
  4. Si algún c_i se movió volver al paso 2

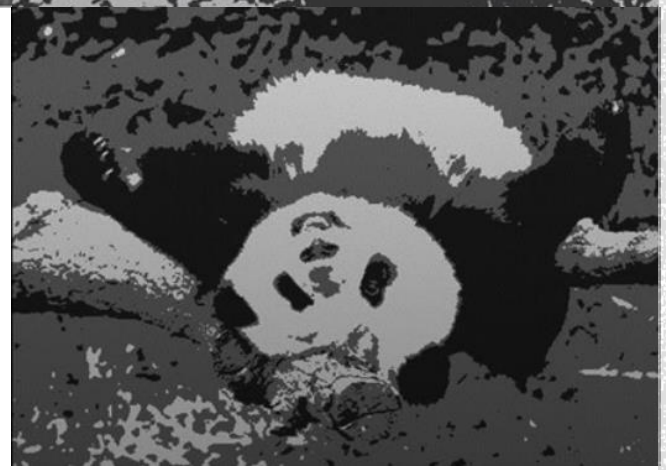


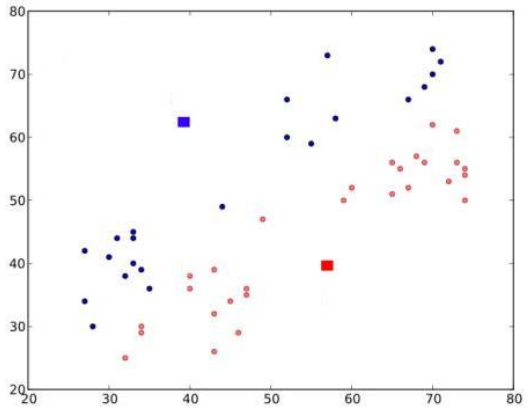


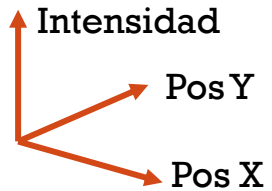


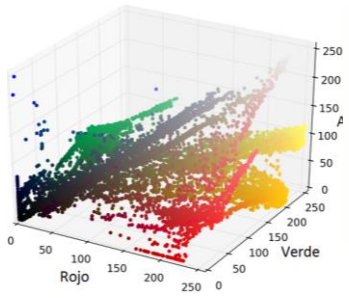










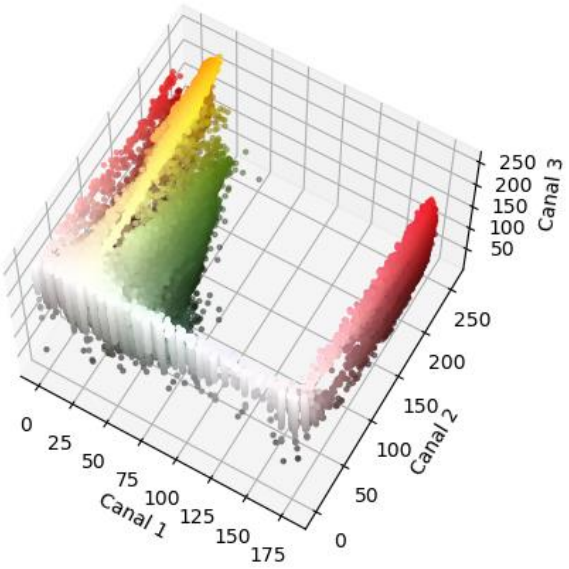


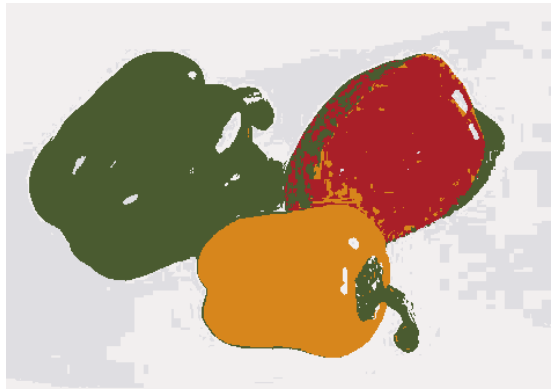
K-MEANS

- 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

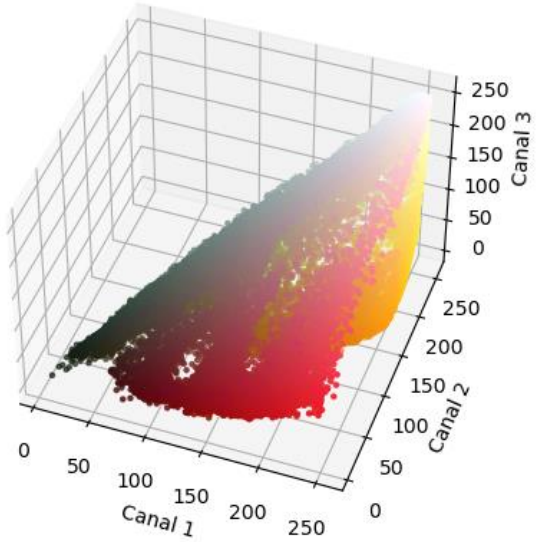
K-MEANS

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





Espacio: HSV



Espacio: RGB

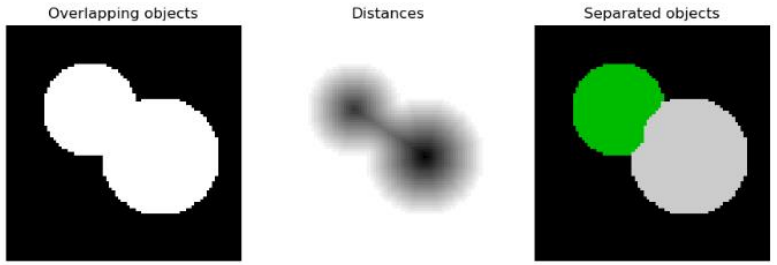
WATERSHED

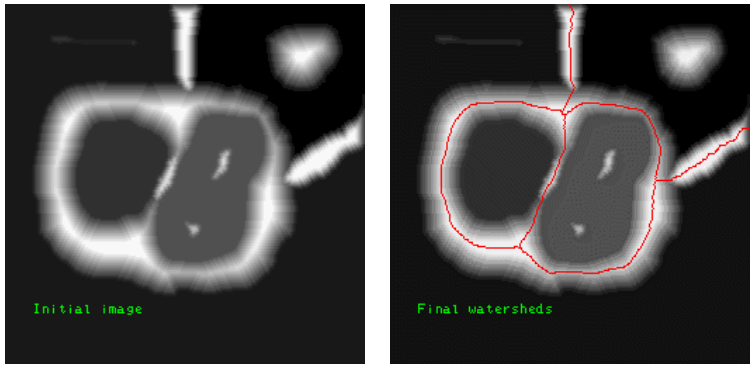
- La idea del método es segmentar la imagen en varios "cuencos" acumuladores.
- El método busca comenzar a inundar el paisaje de la imagen (pensada como superficie topográfica) en todos los mínimos locales y comenzar a etiquetar bordes a medida que las áreas (cuencos) de agua distinta comienzan a juntarse.
- 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**.

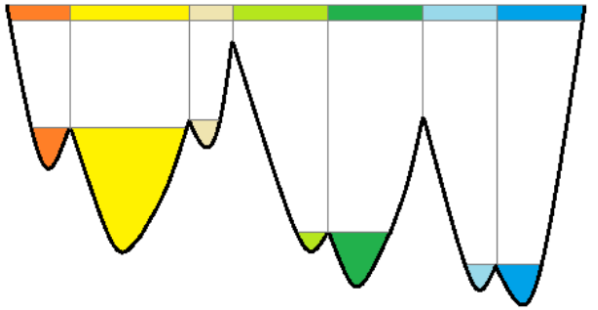
  La indicación de marcadores puede ser:

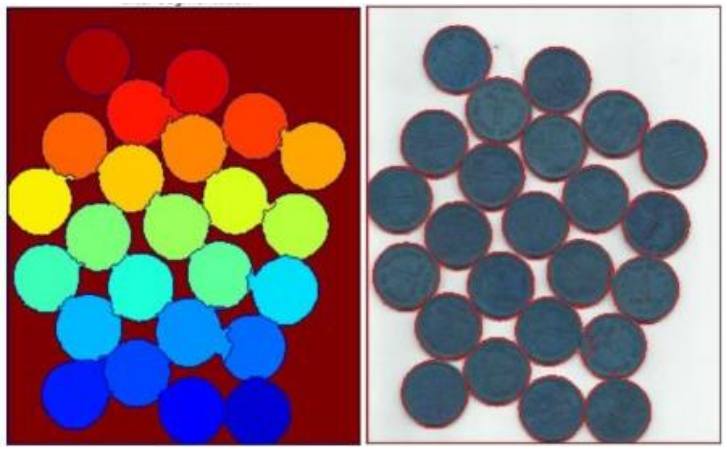
  1. De manera interactiva por el operador, con un click de mouse.
  2. De manera automática, a través de una binarización, análisis morfológico y análisis de blobs.

- Dado que los bordes obtenidos de esta manera suelen ser lo que el algoritmo de **contornos activos** sigue, usualmente se utiliza el algoritmo de watershed para precomputar dicha segmentación.



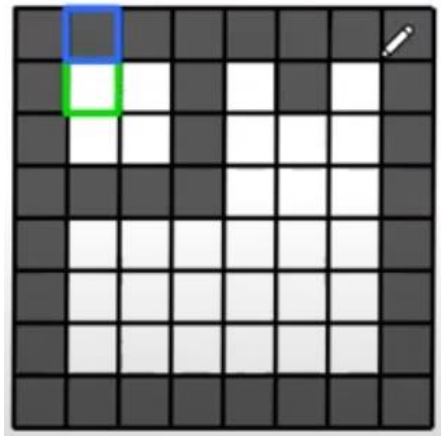






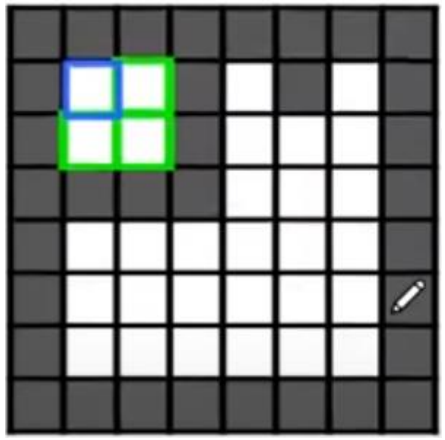
FLOOD FILLING (SOLO IMÁGENES BINARIAS)

1.



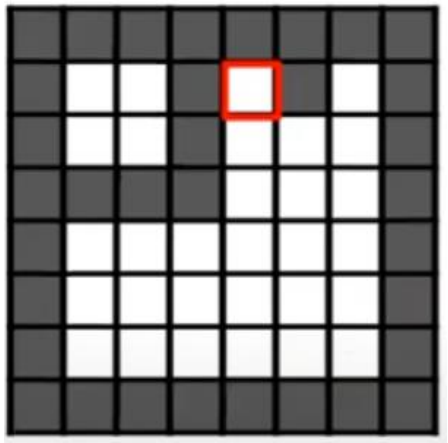
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2.



0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

3.

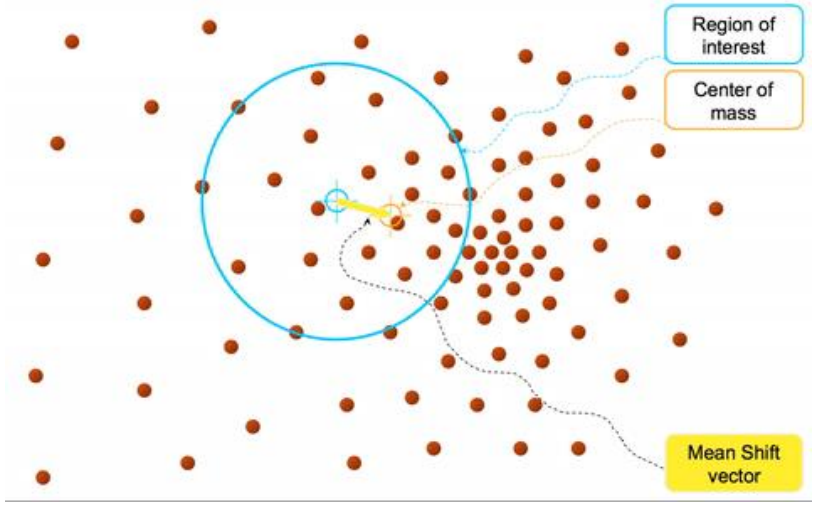


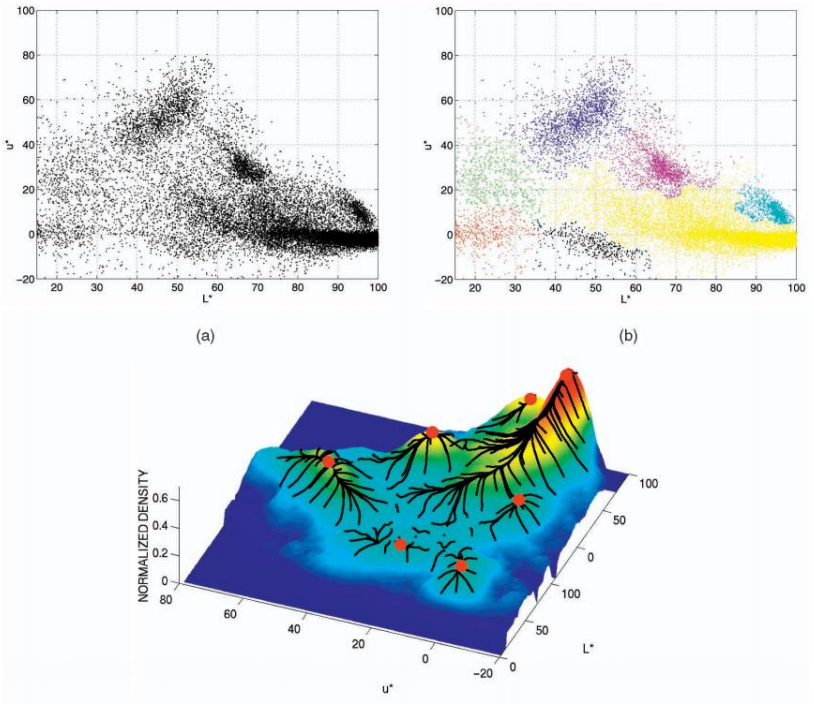
0	0	0	0	0	0	0	0
0	1	1	0	2	0	2	0
0	1	1	0	2	2	2	0
0	0	0	0	2	2	2	0
0	2	2	2	2	2	2	0
0	2	2	2	2	2	2	0
0	2	2	2	2	2	2	0
0	0	0	0	0	0	0	0

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



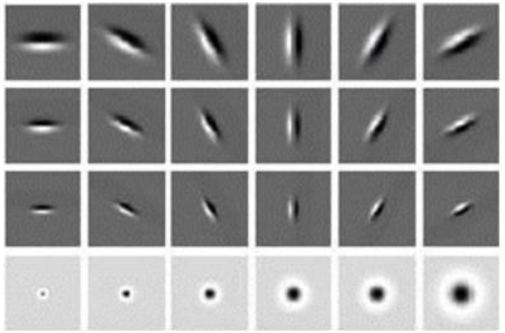


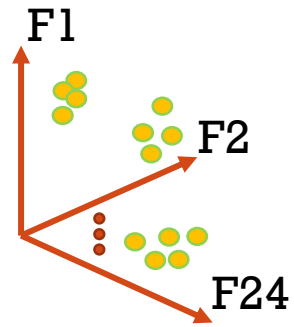
MEAN SHIFT

- **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

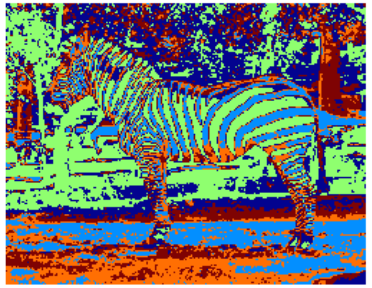




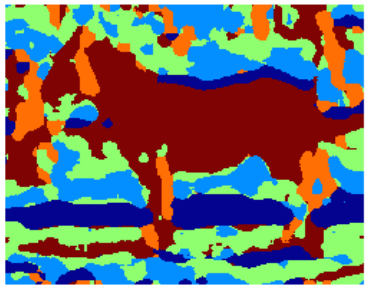
Banco de filtros (24)

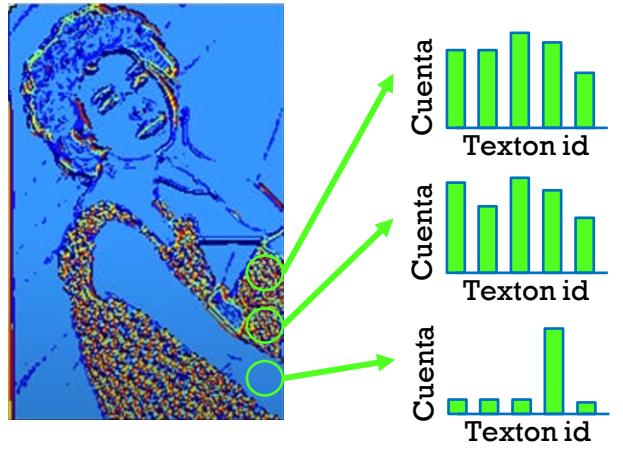
Original image

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

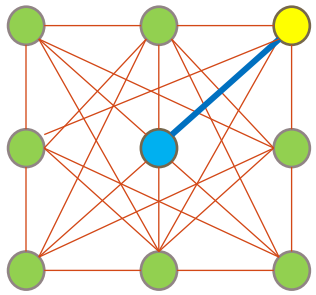


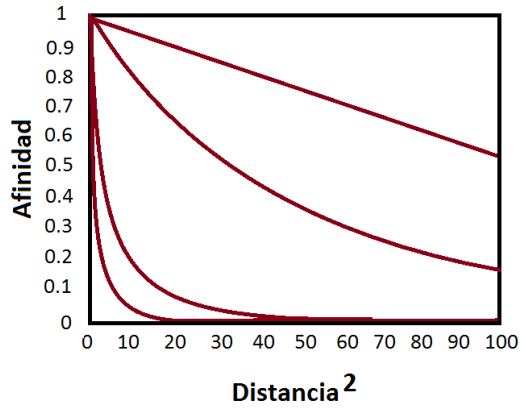
k-means (k=5), feature: texton

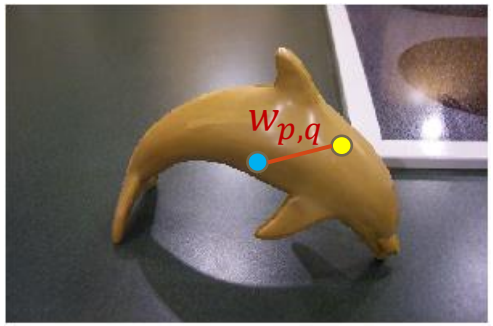


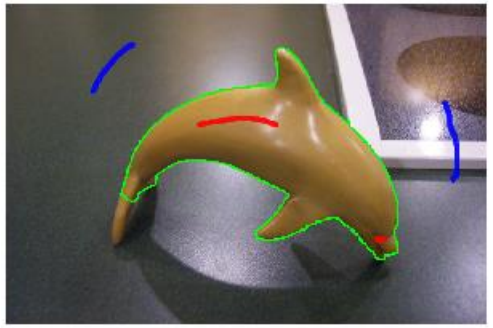


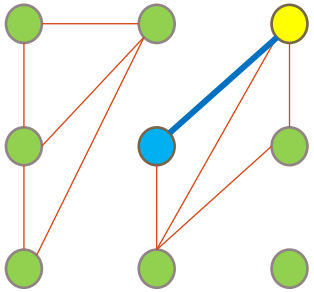
GRAPH CUT









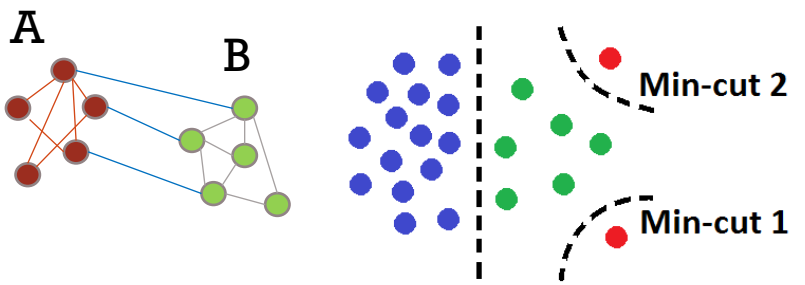


- La segmentación de imágenes puede realizarse pensando a la imagen como un grafo (en relación a la conexión de sus píxeles)
- La idea es generar con cada píxel un grafo conectado completamente
  1. Un nodo (vértice) para cada píxel
  2. Un enlace para cada par de píxeles $\langle p, q \rangle$
  3. Un peso $w_{p,q}$ para cada enlace (basado en su "afinidad": color, posición, etc.)
- La afinidad puede medirse de manera estándar de la siguiente manera:

$$aff(\overline{x_p}, \overline{x_q}) = exp\left(-\frac{1}{2\sigma^2} dist(\overline{x_p}, \overline{x_q})^2\right)$$

- El parámetro σ habla de la proximidad relativa a la afinidad de los píxeles
- Luego, la idea es "cortar" los enlaces con menos peso (menor afinidad)
  1. Píxeles similares quedarán en las mismas conexiones
  2. Píxeles distintos quedarán en conexiones distintas

GRAPH CUT NORMALIZADO



- La idea de Graph Cut es, removiendo algunas conexiones tener gráficos desconectados.
- El costo del corte se define como la suma de los pesos de los enlaces cortados

$$corte\ (A,B) = \sum_{p\in A, q\in B} w_{p,q}$$

- Existen algoritmos que eligen los mejores cortes (min-cut, max-flow, etc)
- Problema con min-cut
  - El peso de un corte es proporcional al número de enlaces en el corte
  - Tiende a producir componentes pequeños aislados
- Corte normalizado: Arregla la propensión a recortar pequeñas áreas

$$Corte\ Norm(A,B) = \frac{corte\ (A,B)}{asoc(A,V)} + \frac{corte\ (A,B)}{asoc(B,V)}$$

$$Asoc(A,V) = suma\ de\ todos\ los\ pesos\ que\ tocan\ A$$

- Pros
  1. 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)
  2. No requiere un modelo de la distribución de datos (solo la función de distancia)
- Cons
  1. El tiempo de cómputo puede ser elevado (matrices de afinidad de nxn con n el número de píxels)
  2. En grafos densamente conectados hay muchos cómputos a realizar → Se puede resolver con una representación en autovalores/autovectores…pero con cierta complejidad del algoritmo.
  3. El corte normalizado tiene preferencia por particiones del gráfico "balanceadas". Si de hecho existen regiones pequeñas a segmentar el algoritmo va en contra de eso.

TP5

- Para una imagen de su elección:
  1. Construir un espacio de características basado en color
  2. Construir un espacio de características basado en color y posición. Ejemplo: $\bar{X} = (R, G, B, x, y)$ (pueden experimentar con otros espacios de color o usar escala de grises)
  3. Graficar la distribución de características para ambos espacios
  4. Obtener la imagen posterizada y las mascaras correspondientes a cada clase, utilizando k-means con asignación aleatoria de centroides, para ambos espacios de características
  5. Comparar todos los resultados obtenidos