

# Tema 9: Detección, extracción de características, segmentación y reconocimiento de objetos

## Parte I

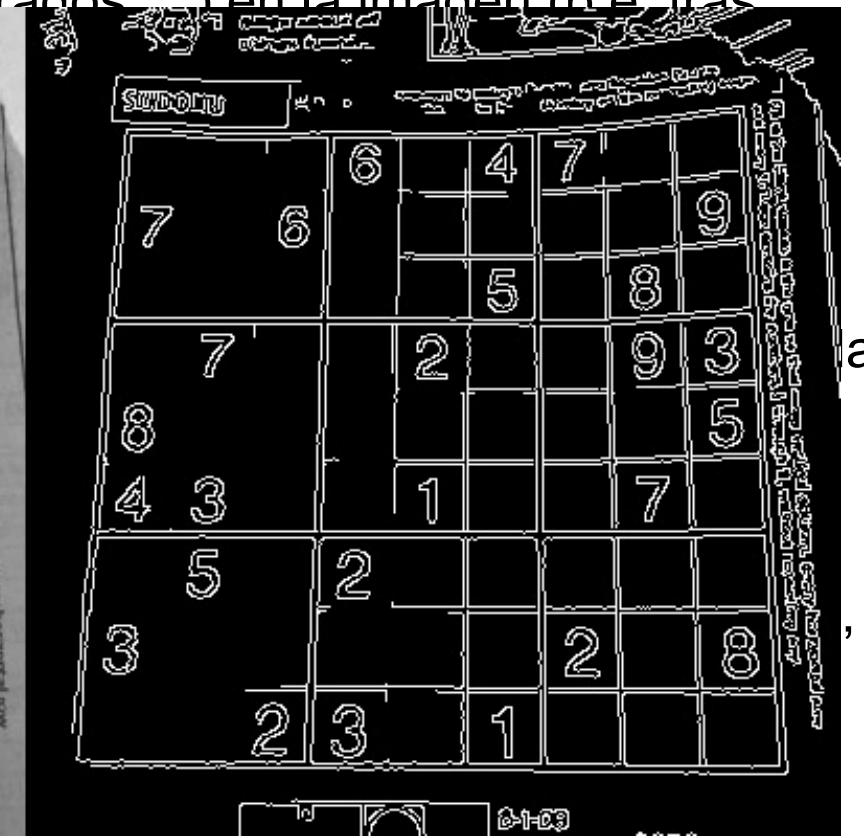
# Índice

- Transformada de Hough
- Características SIFT
- Segmentación de imágenes
  - Algoritmo de las K-medias
  - Segmentación basada en regiones



# Transformada de Hough

- Fundamentos:
  - Motivación:
    - Desarrollar técnicas que permitan identificar primitivas geométricas sencillas (líneas, círculos, cuadrados,...) en la imagen (p.e., trazos)
- M



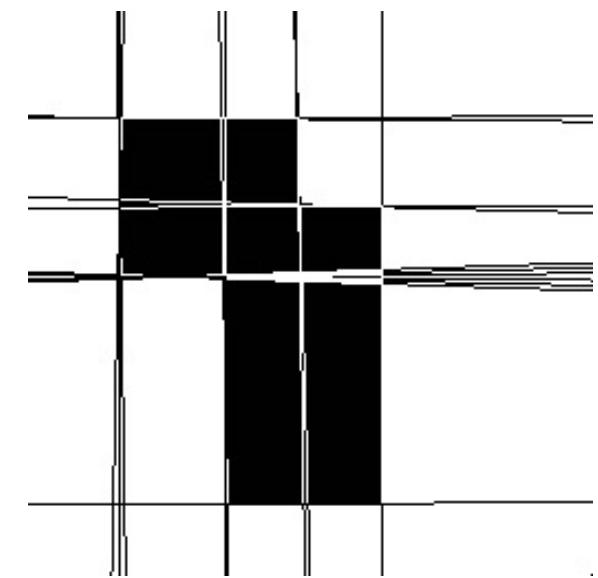
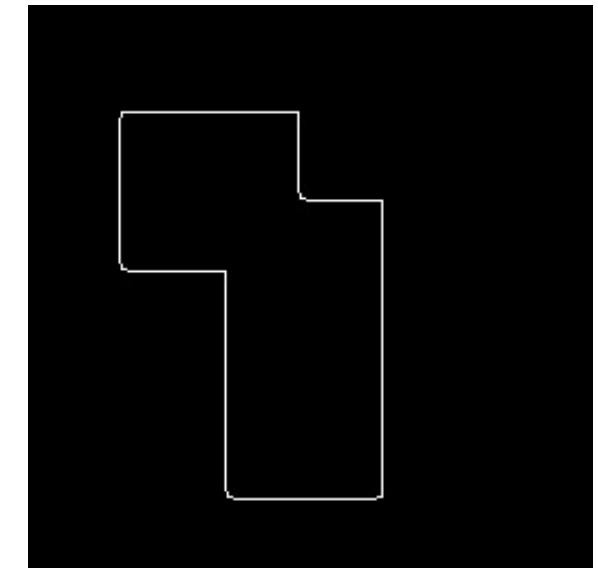
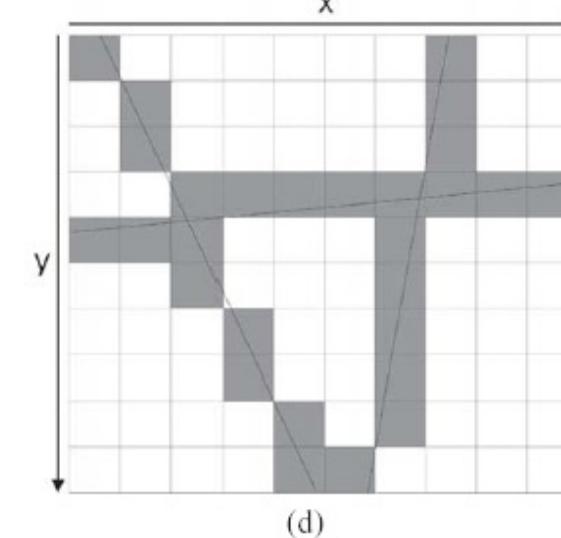
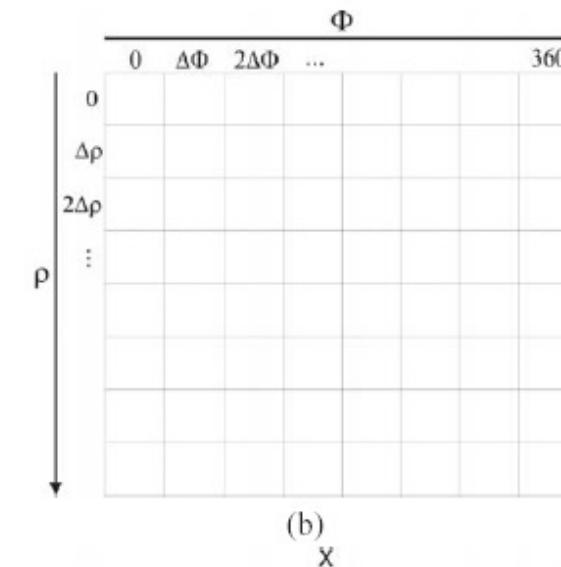
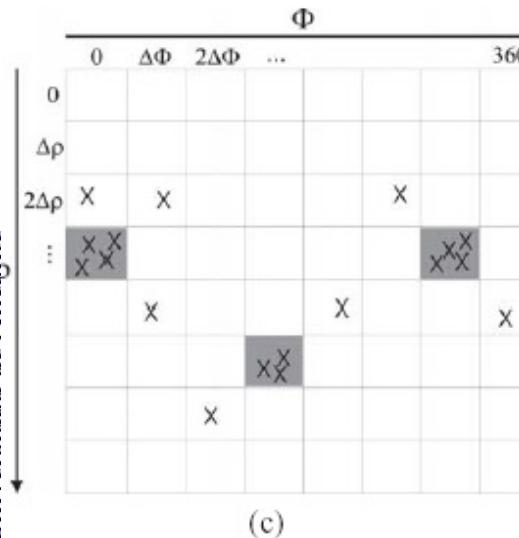
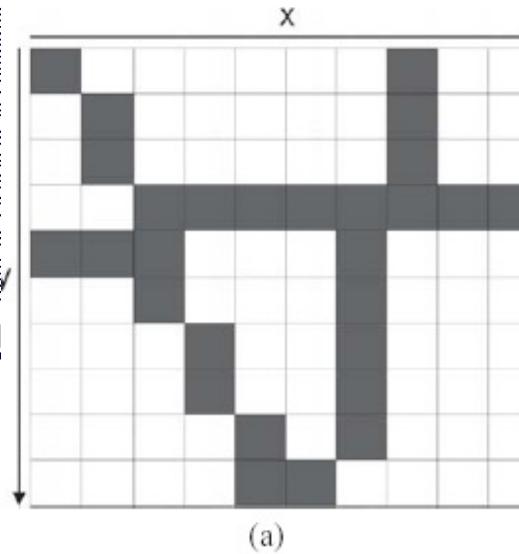
# Transformada de Hough





# Transformada de Hough

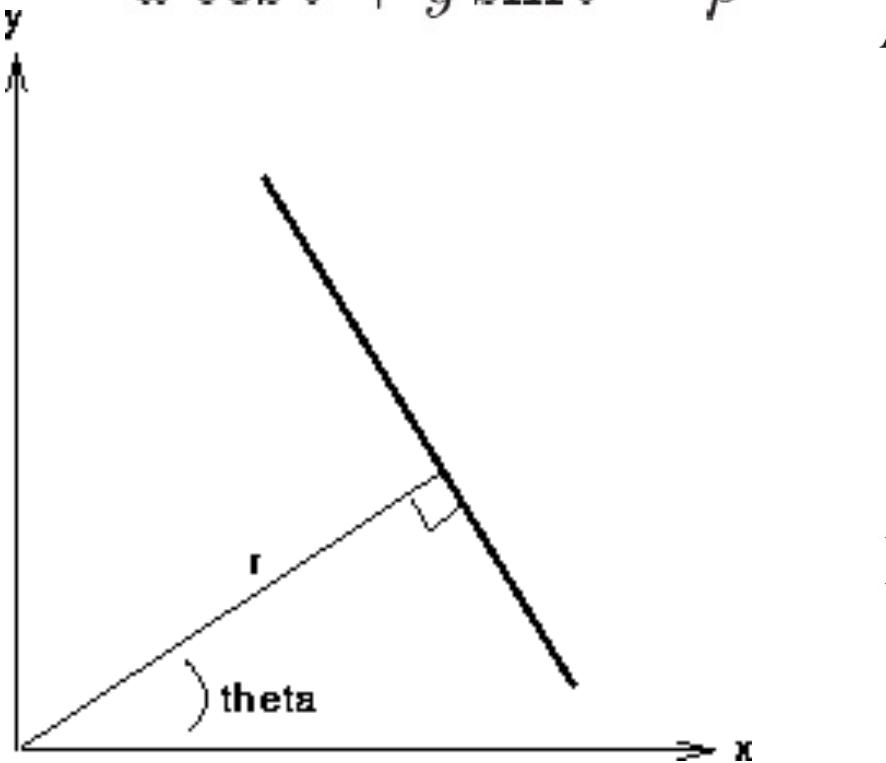




# Transformada de Hough

- Transformada para la recta:
  - Ecuación paramétrica:

$$x \cos \theta + y \sin \theta = \rho$$



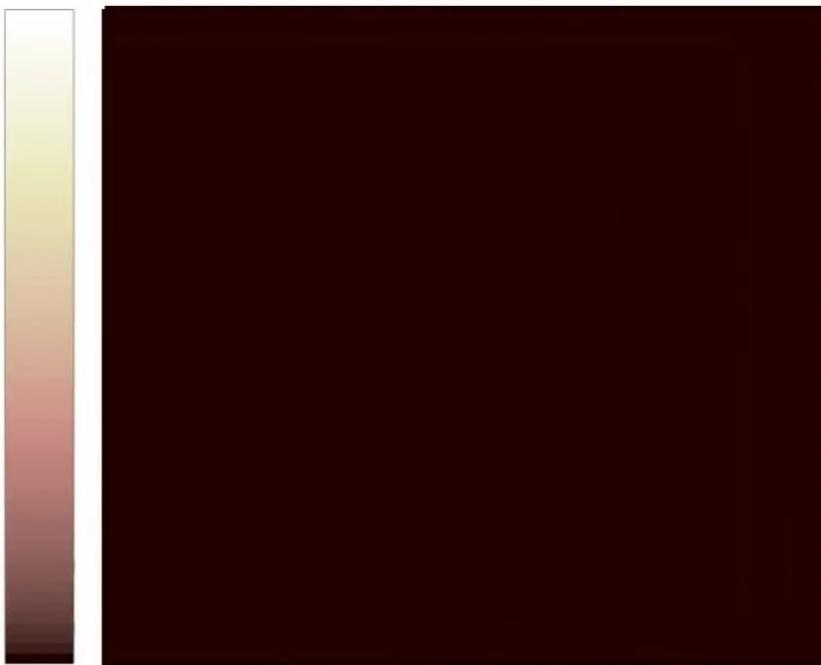
```

Algoritmo HOUGH_RECTA {
    Para cada  $(\rho, \theta)$ 
         $Votos_{\rho\theta} \leftarrow 0$ 
    Para cada pixel  $(x, y)$  de la imagen de entrada
        Para cada orientacion posible de las rectas  $\theta$ 
             $\rho \leftarrow x \cos \theta + y \sin \theta$ 
             $Votos_{\rho\theta} \leftarrow Votos_{\rho\theta} + 1$ 
    Devolver  $Votos$ 
}

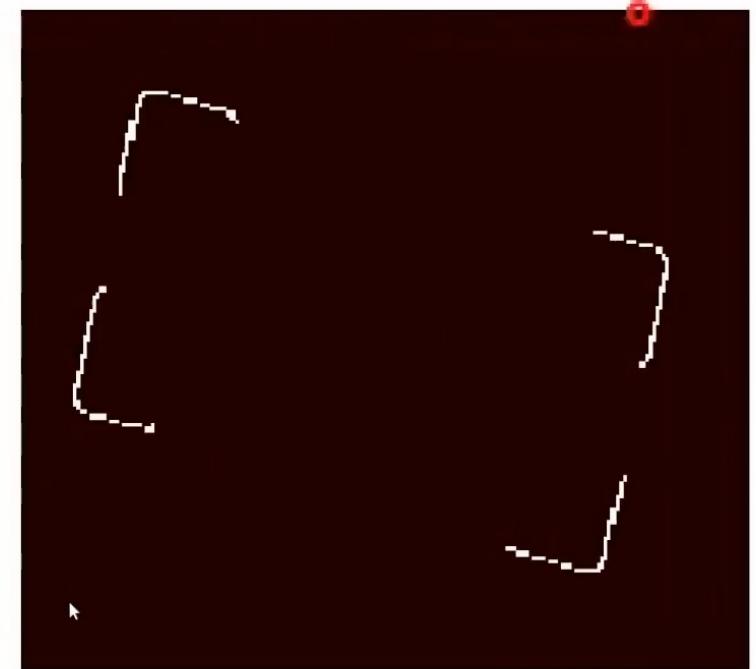
```



# Transformada de Hough



input image



Vídeo obtenido de: <https://www.youtube.com/watch?v=4zHbl-fFIII&t=221s>

# Transformada de Hough

- Transformada del círculo:

- Ecuación paramétrica:  $(x - a)^2 + (y - b)^2 = r^2$
- Algoritmo:

**Algoritmo** HOUGH\_CIRCULO{

**Para** cada  $(a, b, r)$

$Votos_{abr} \leftarrow 0$

**Para** cada pixel  $(x, y)$  de la imagen de entrada

**Para** cada coordenada x del centro  $a$

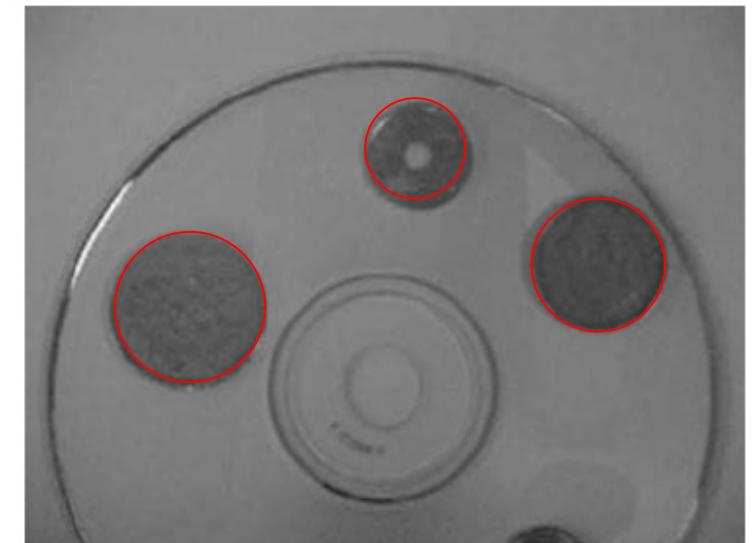
**Para** cada coordenada y del centro  $b$

$r \leftarrow \sqrt{(x - a)^2 + (y - b)^2}$

$Votos_{abr} \leftarrow Votos_{abr} + 1$

**Devolver**  $Votos$

}





# Índice

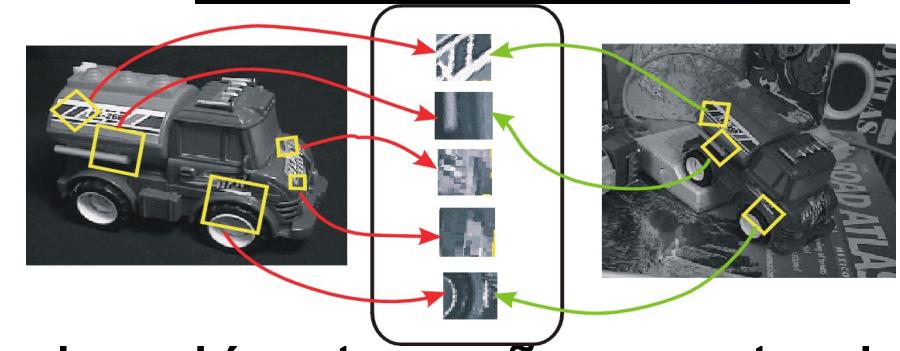
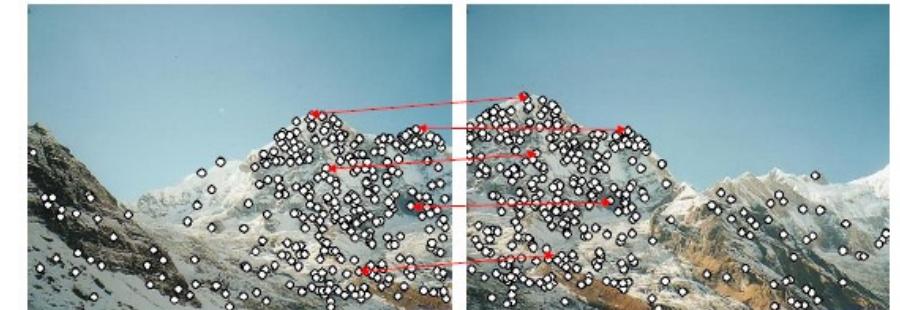
- Transformada de Hough
- Características SIFT
- Segmentación de imágenes
- Algoritmo de las K-medias
- Segmentación basada en regiones



# Extracción de características

- Las características de una imagen se puede entender como una representación “comprendativa” de la imagen, pero reducida
- Existen una serie de tareas en visión que necesitan extraer estas características
  - Reconocimiento de objetos
  - Reconstrucción 3D
  - Alineación de imágenes para crear panorama
  - Seguimiento de movimiento
  - Búsqueda en base de datos de imágenes
  - Aprendizaje automático
  - Etc.
- Estas características necesitarían ser identificadas de manera “única”

# Ejemplos



- Problemas: cambios de iluminación, tamaño, punto de vista, occlusiones, etc.
- <http://matthewwalunbrown.com/autostitch/autostitch.html>

# Solución

- Usar, además de la posición en la imagen, algún “descriptor” que identifique el punto de manera “única” (o casi única).
- El descriptor contendrá algún tipo de información de la imagen alrededor del punto.
- La misma característica, vista desde distintos puntos de vista, con distinta iluminación, con giros, etc. tendrá el mismo descriptor.
- Su cálculo debe ser eficiente.
- Existen varias alternativas: SIFT, SURF, BRIEF, MSER, etc.

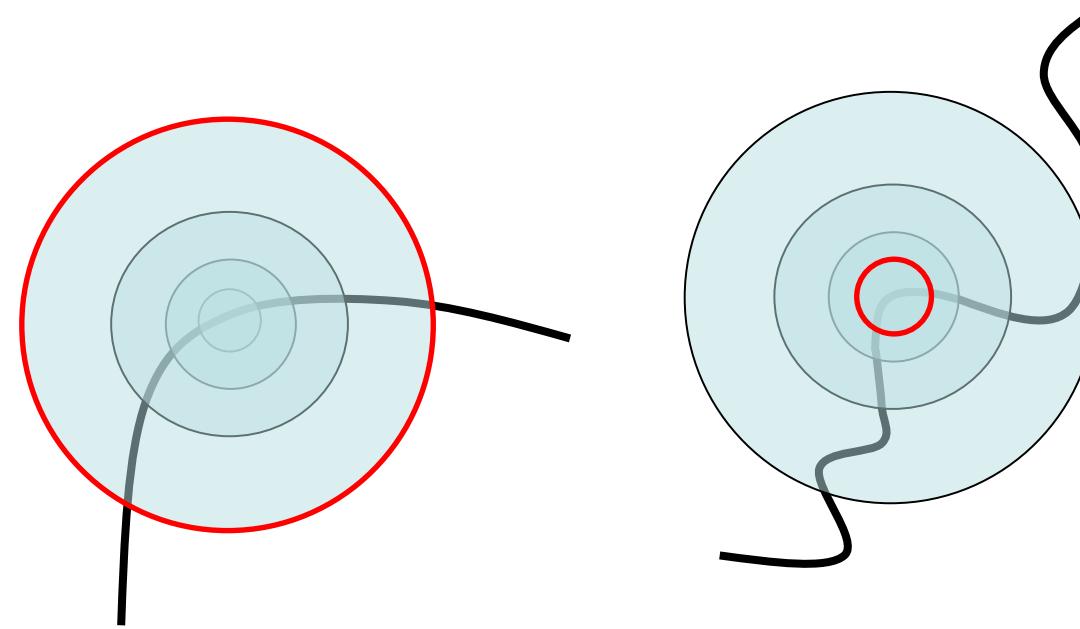
# SIFT

- David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004)
- SIFT: Scale Invariant Feature Transform
- Uno de los primeros en ser usados: el autor deja un ejecutable que puede ser usado. (<https://www.cs.ubc.ca/~lowe/keypoints/>)
- El algoritmo devuelve las características encontradas en la imagen
- Tiene dos pasos:
  - Encontrar la posición de los puntos (se podría usar Harris o cualquier otro)
  - Calcular el descriptor



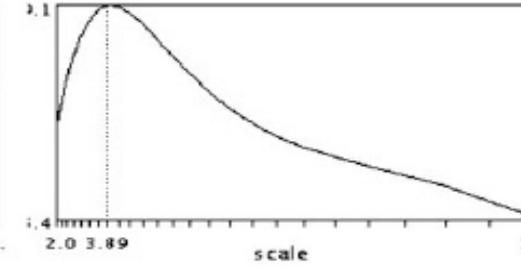
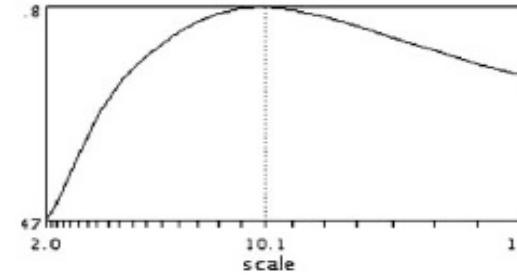
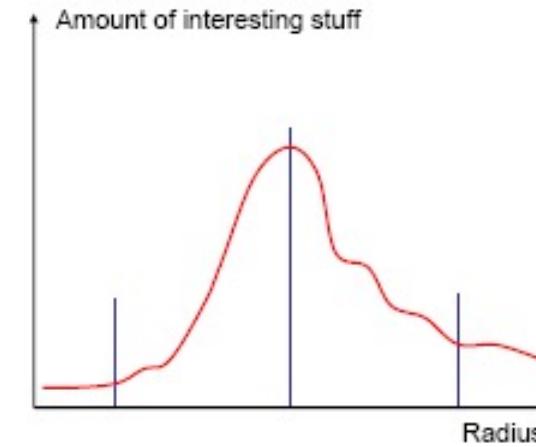
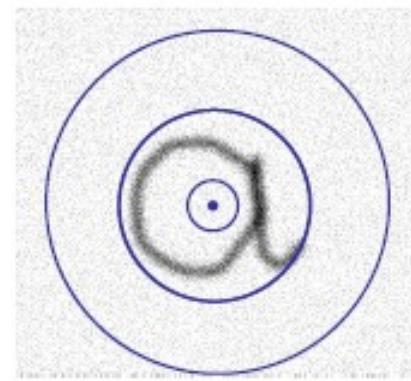
# Localización de las características

- Harris podría ser un buen candidato
  - ¿Por qué?
  - ¿Y en cuanto a la escala?
- La idea es intentar que el detector sea invariante a escala. Sift propone una detección multiescala

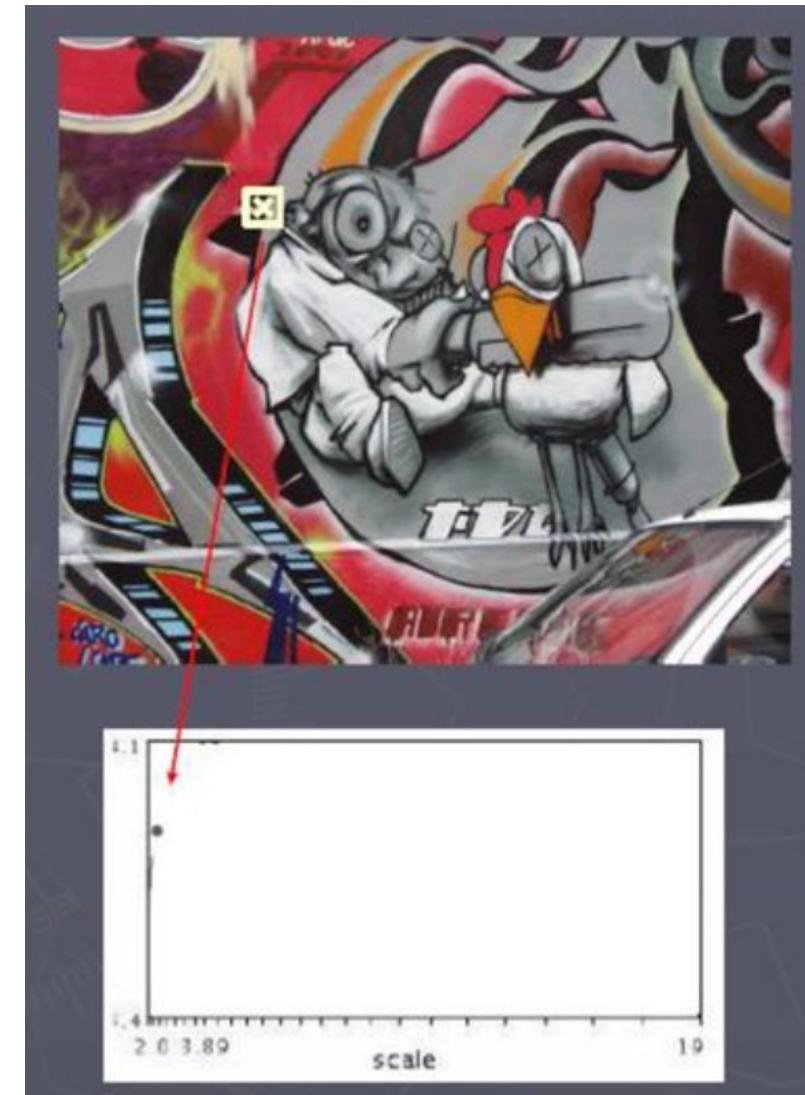


# Localización multiescala

- ¿Cómo se puede elegir la escala?

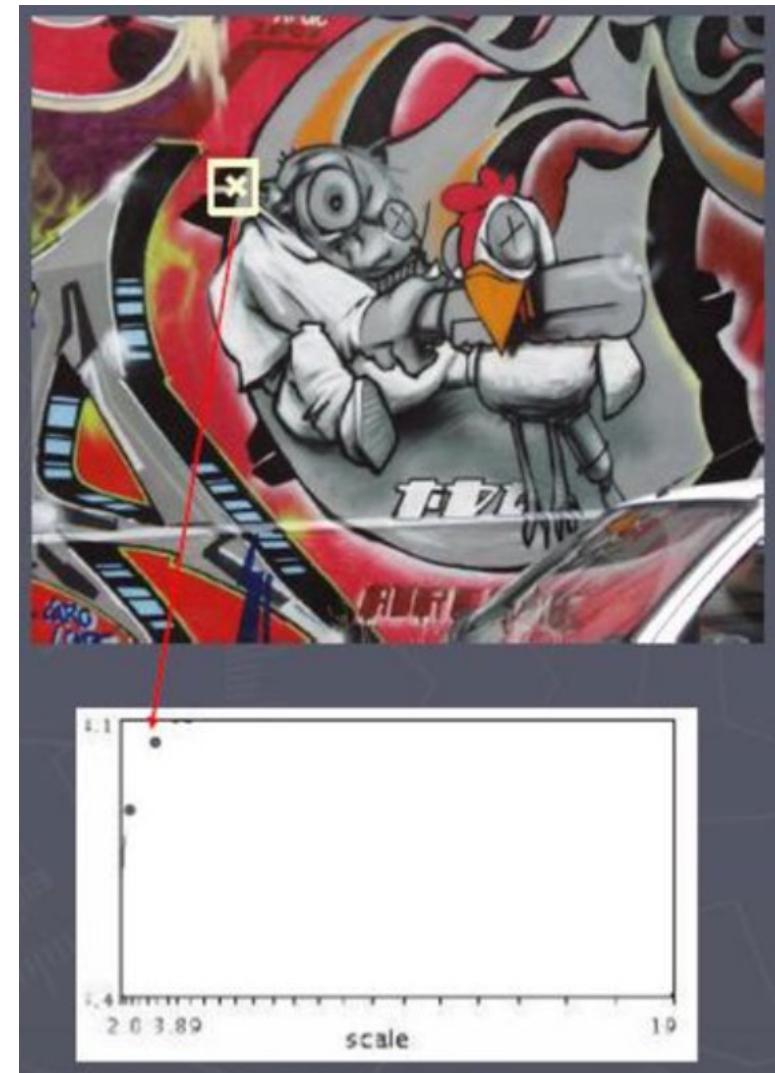


# Localización multiescala



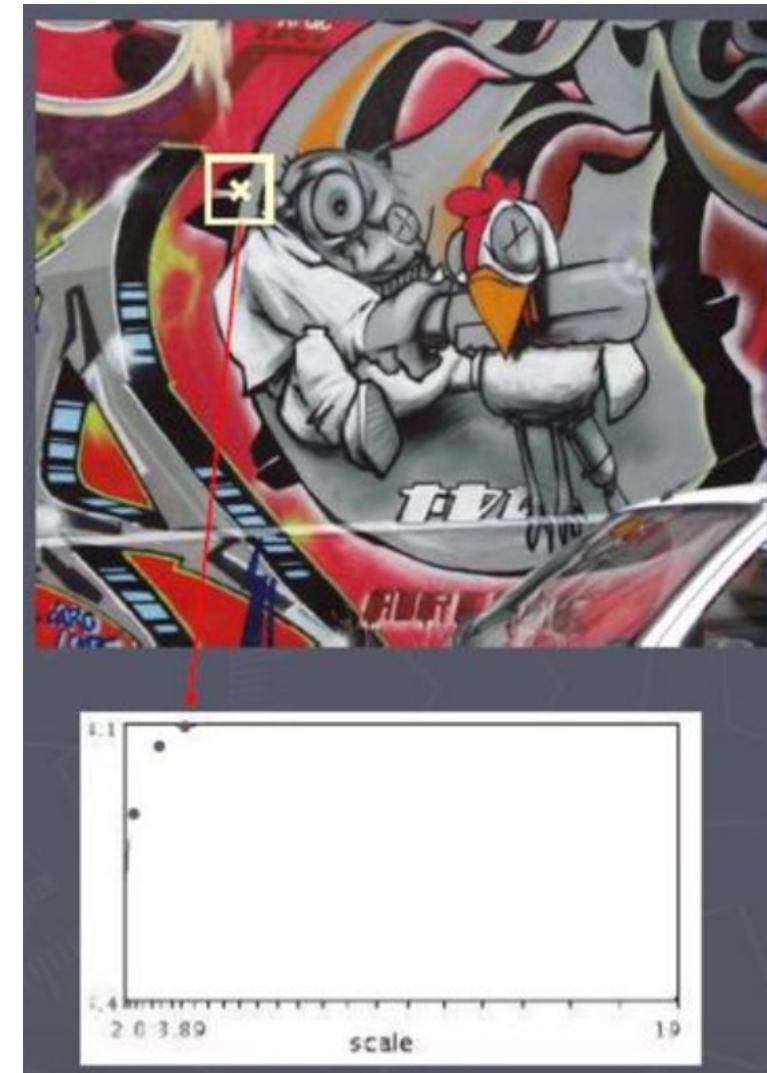
Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala



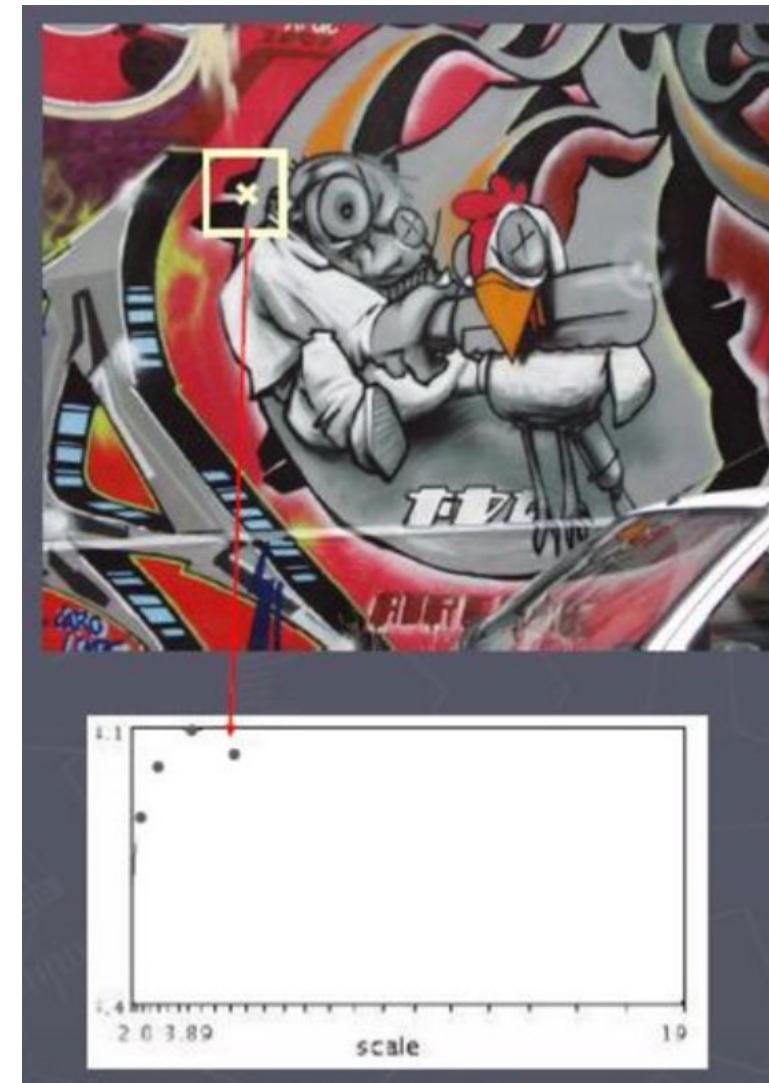
Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala



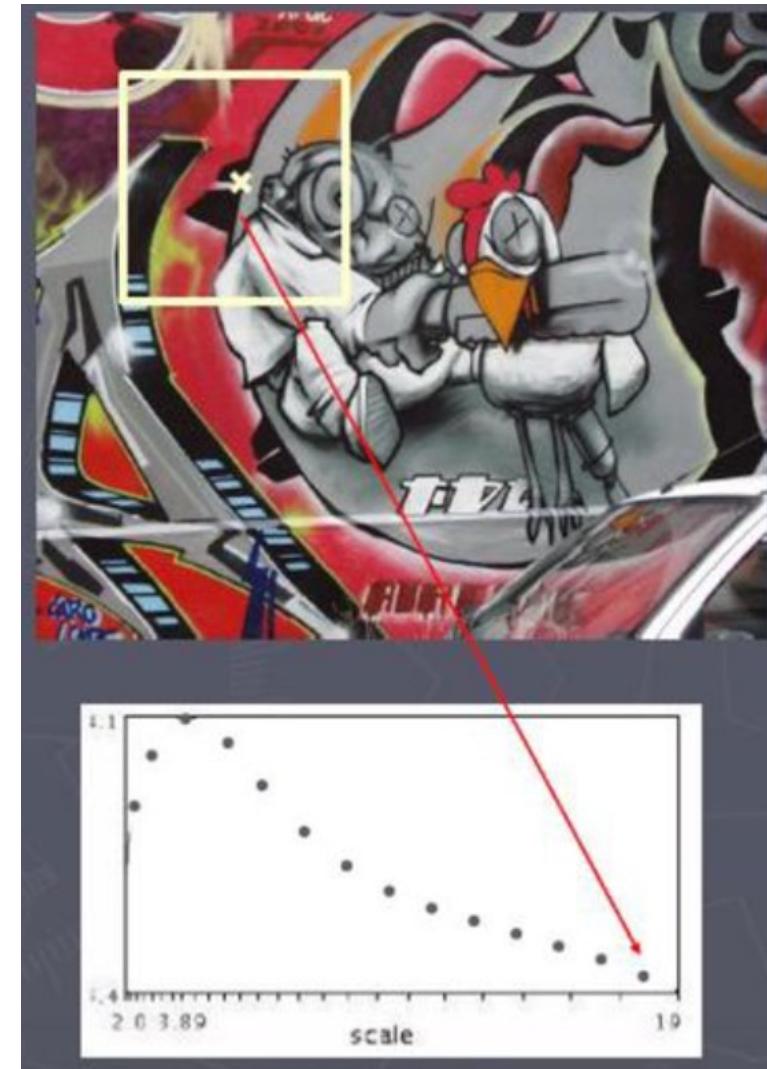
Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala



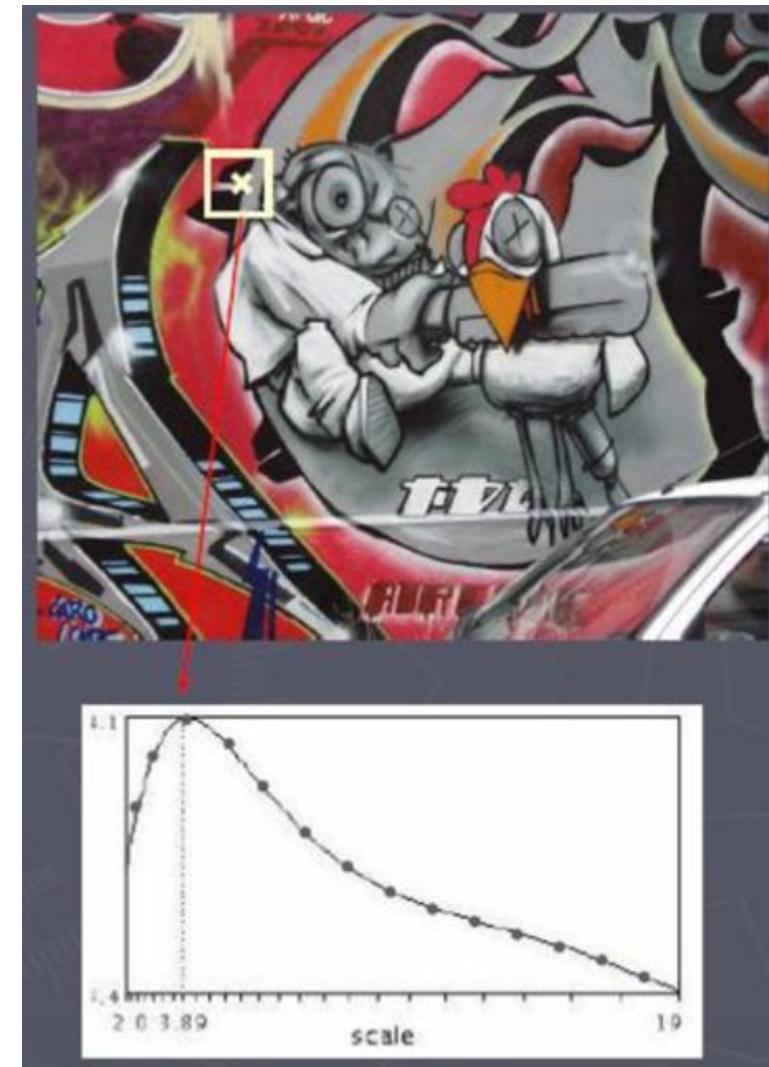
Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala



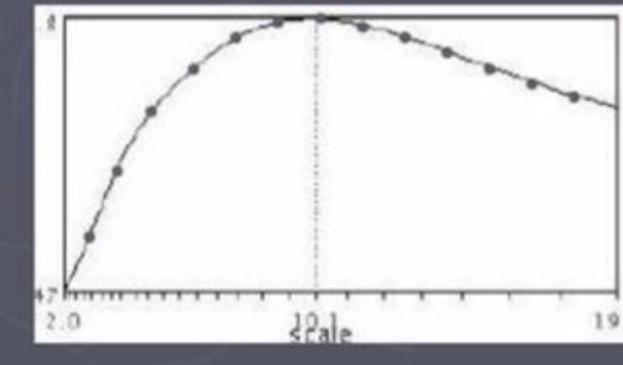
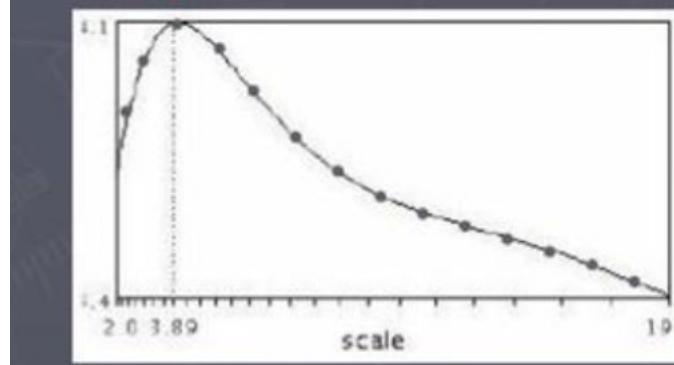
Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala



Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

# Localización multiescala

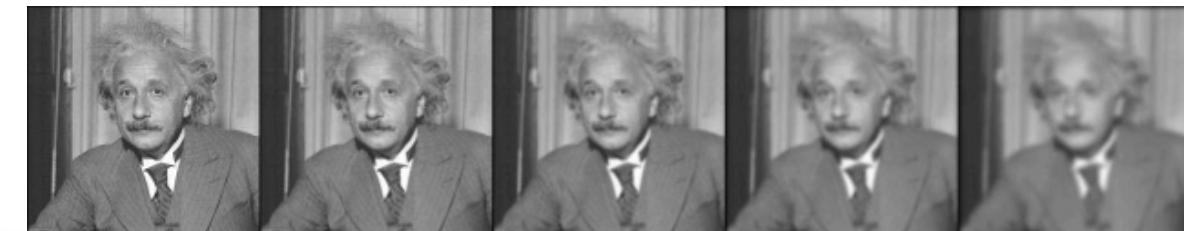
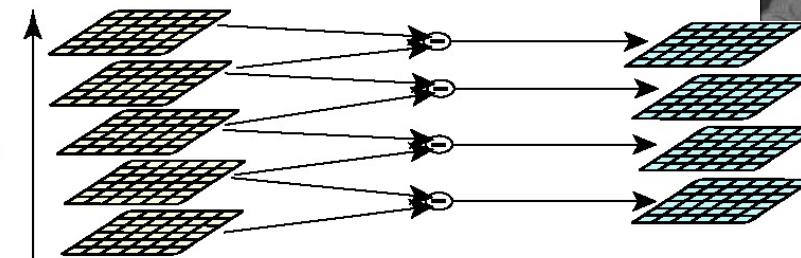


Imágenes obtenidas de: <https://slideplayer.com/slide/16981160/>

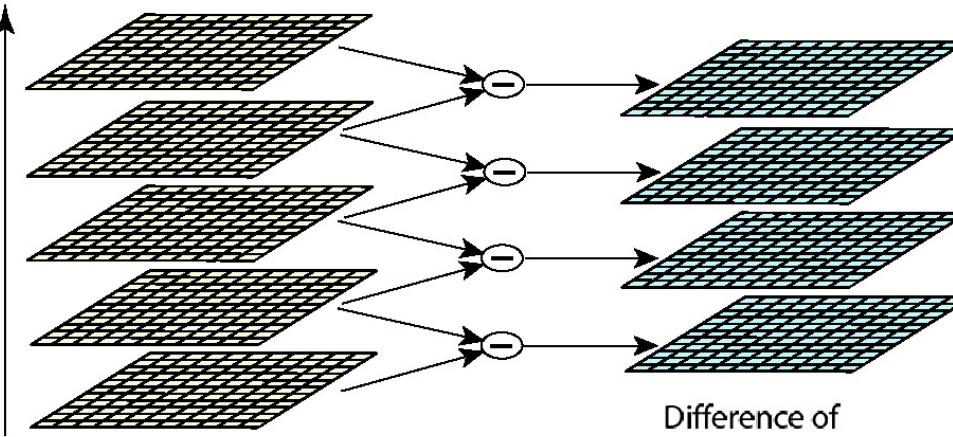
# Localización multiescala

- La localización en la multiescala se consigue con una diferencia de gaussianas (DoG)

Scale  
(next octave)



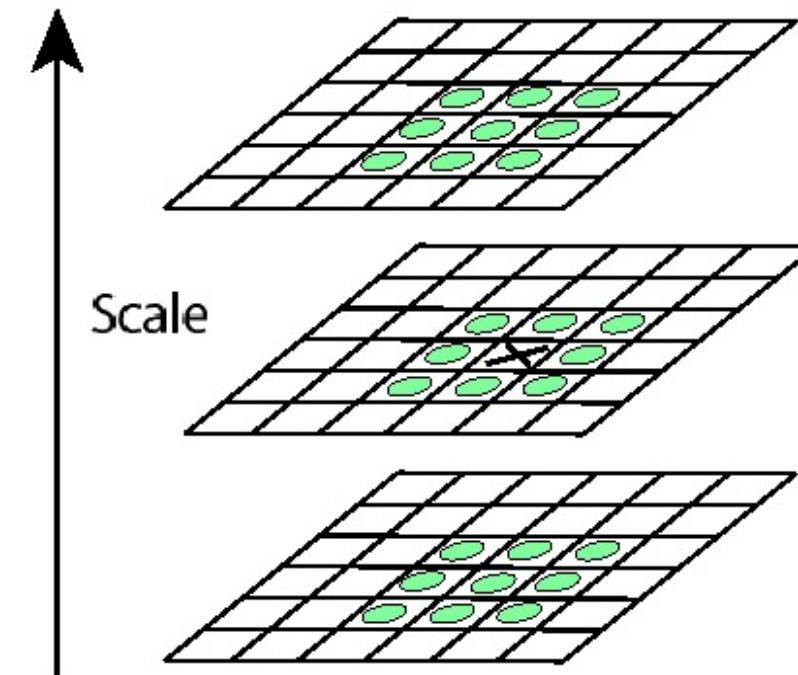
Scale  
(first octave)



Difference of  
Gaussian (DOG)

# Localización

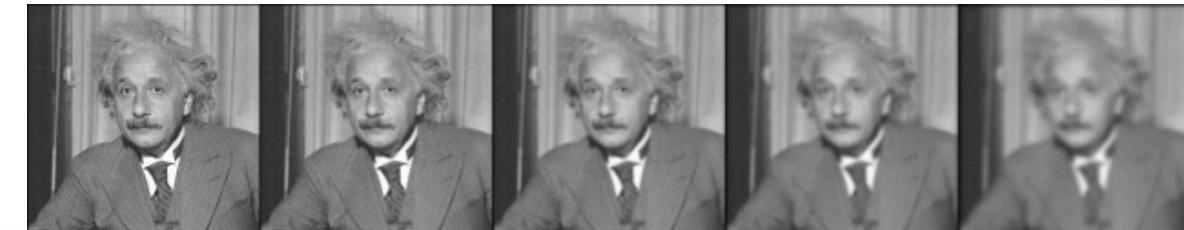
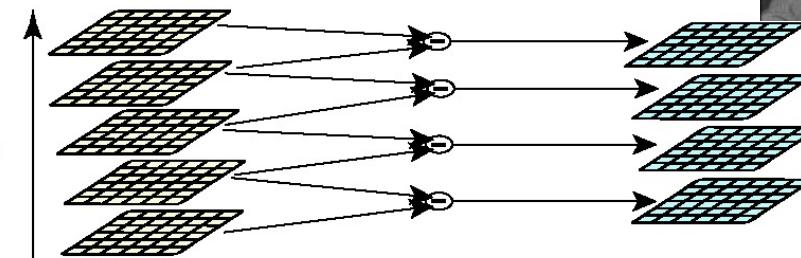
- Hay que encontrar un mínimo o máximo entre escalas



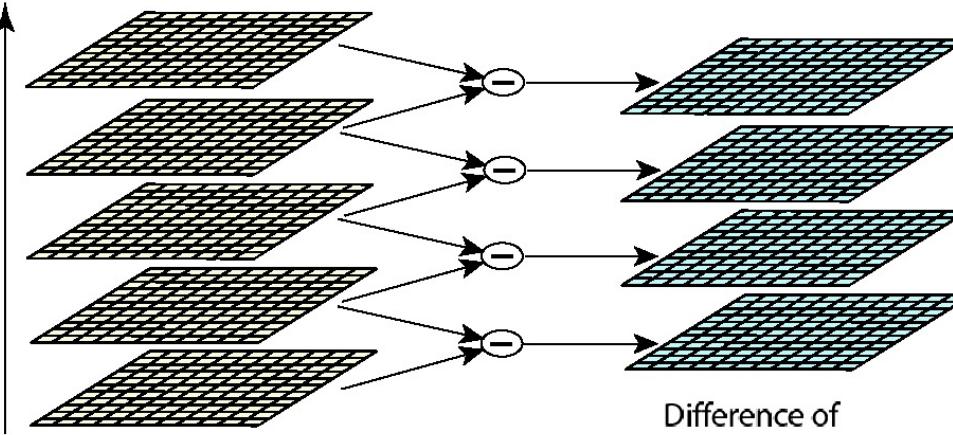
# Localización multiescala

- La localización en la multiescala se consigue con una diferencia de gaussianas (DoG)

Scale  
(next octave)



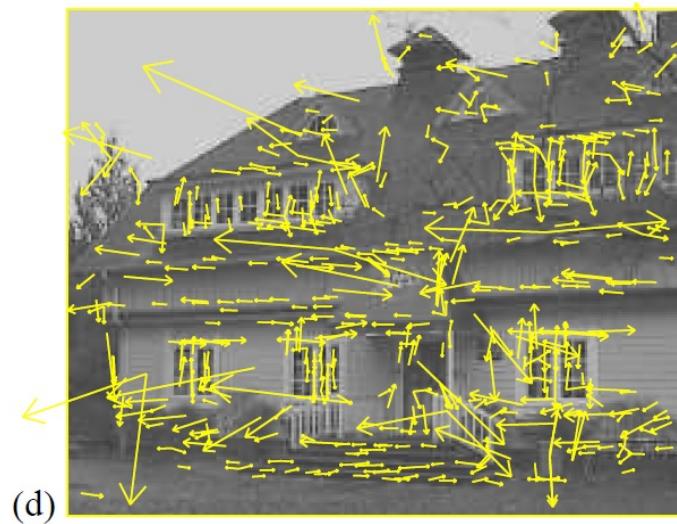
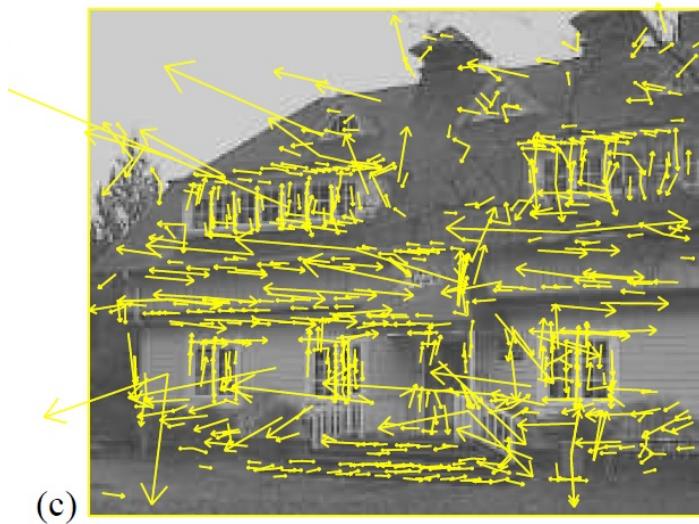
Scale  
(first octave)



Difference of  
Gaussian (DOG)

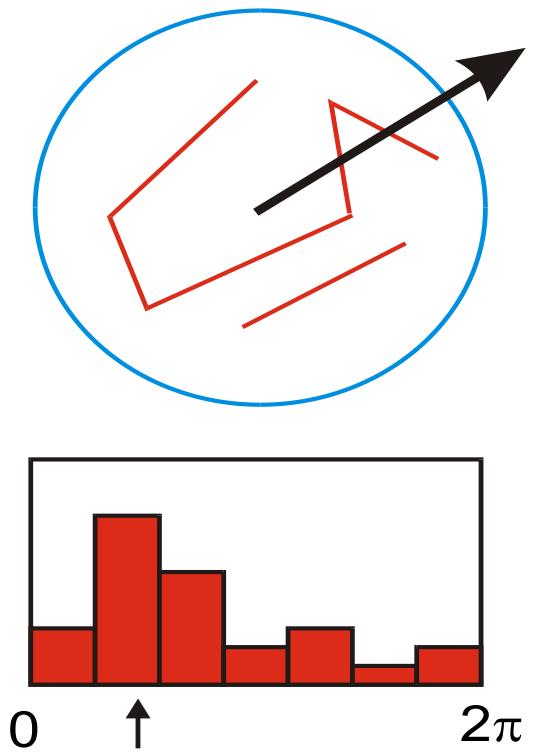
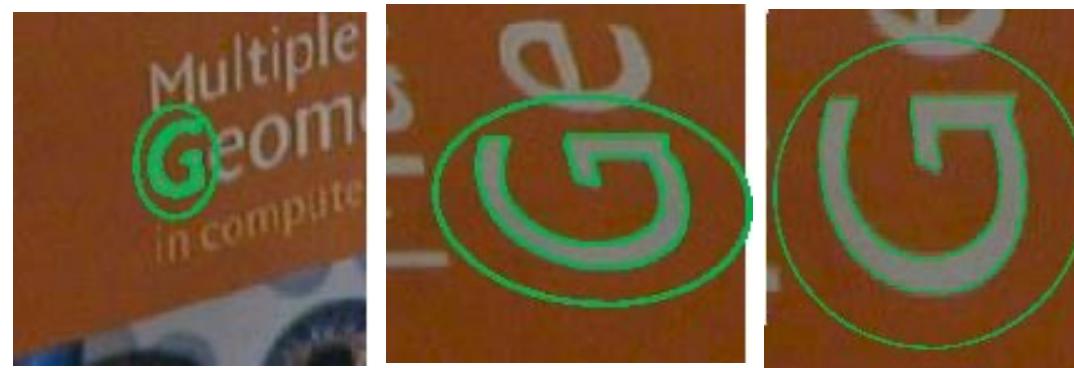
# Eliminación de puntos inestables

- Se encuentran muchos puntos, pero muchos de ellos son “inestables”
- Se eliminan aquellos:
  - Con un contraste bajo
  - Que se encuentran sobre una arista
- Iniciales: 832. Finales: 536



# Orientación de la característica

- Es necesario encontrar cómo está orientada la característica
- Se encuentra el histograma de orientaciones **ponderado en la escala calculada** y se calcula la orientación dominante



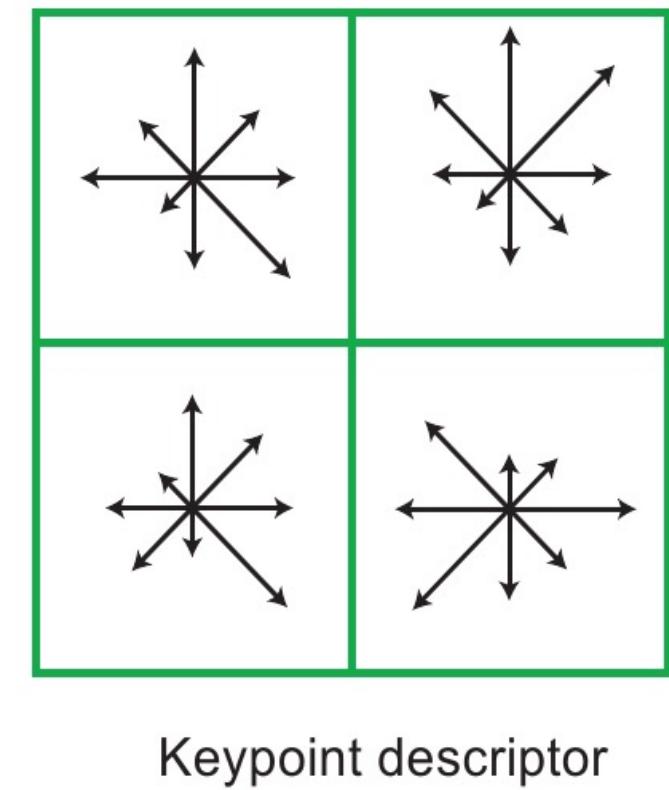
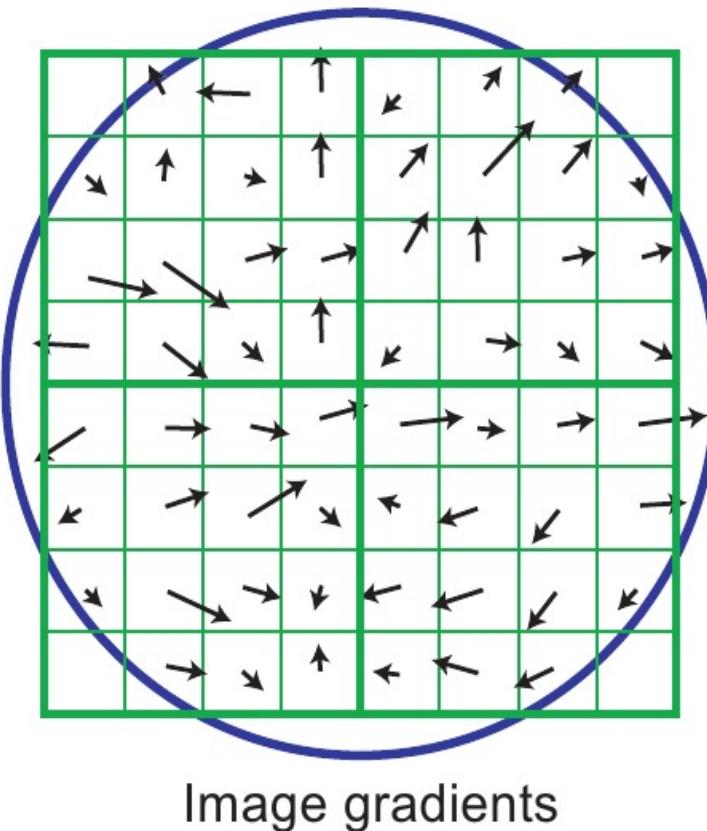
# Descriptor

- Hasta el momento hemos calculado:
  - Localización del punto.
  - Escala.
  - Orientación.
- Debemos encontrar el descriptor que nos permita comparar entre las características
- Se escoge una cierta vecindad del punto a la escala determinada



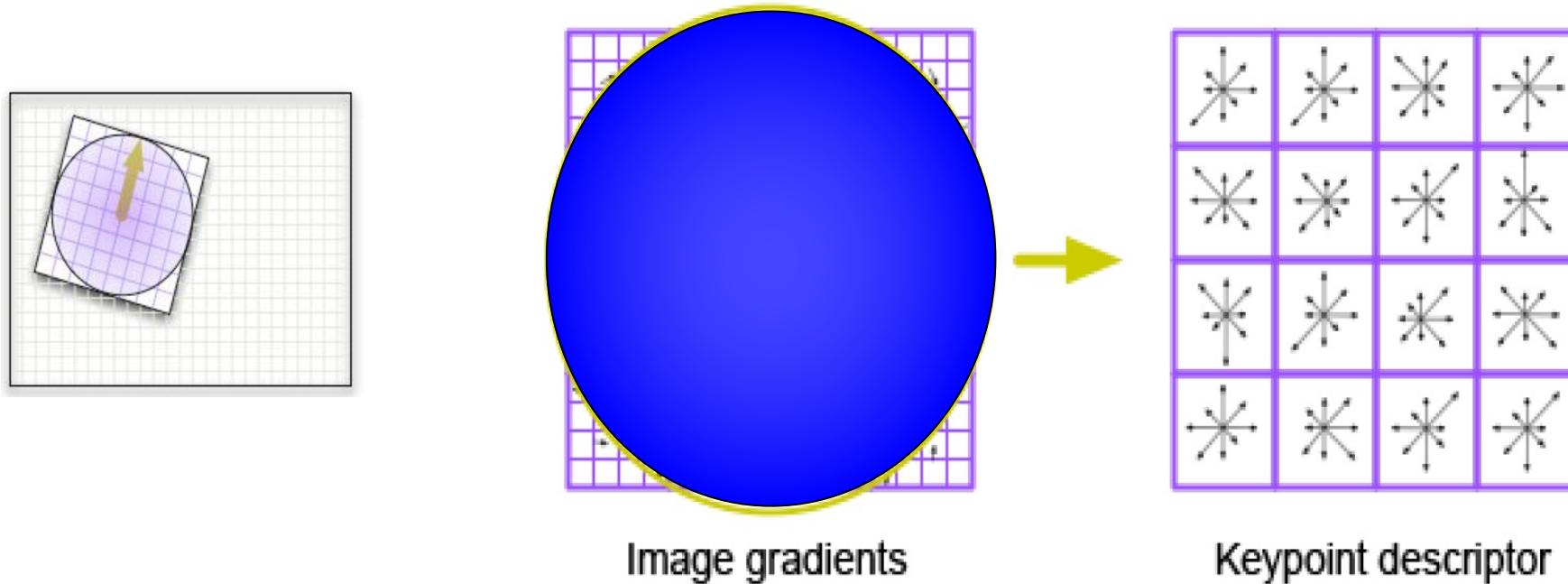
# Descriptor

- Se realiza un histograma de orientaciones alrededor del punto.
- Se calcula un histograma de 8 posibles orientaciones

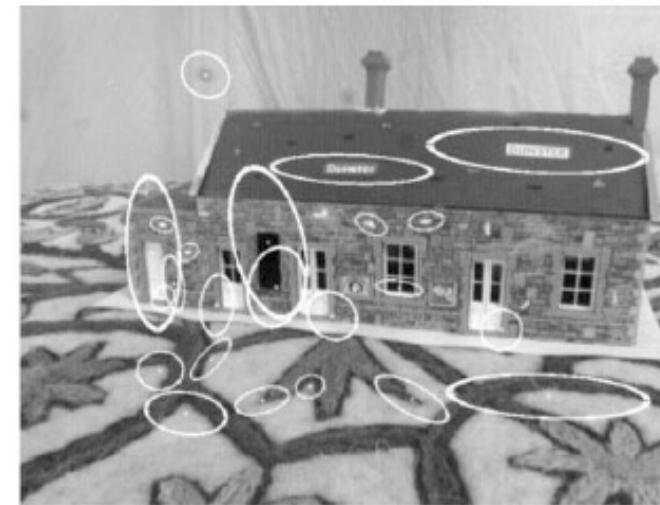
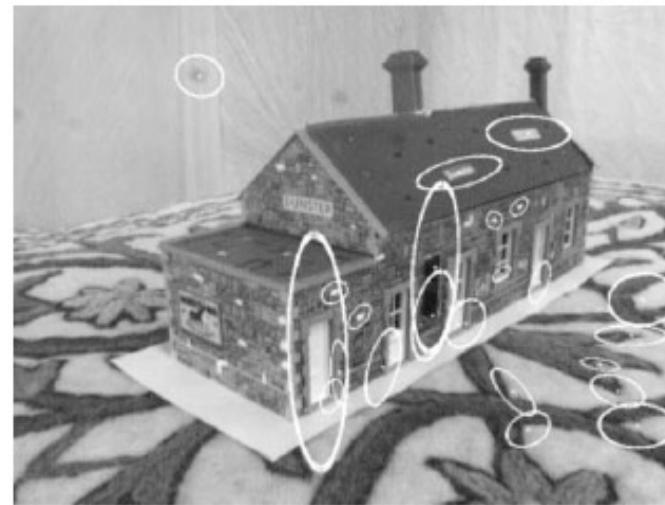


# Descriptor

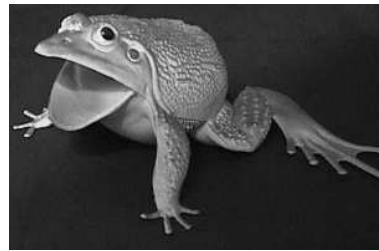
- Se realiza un histograma de orientaciones alrededor del punto.
- Ventana 16x16 particionada en 4x4
- Se calcula un histograma de 8 posibles orientaciones
- Descriptor:  $4 \times 4 \times 8 = 128$



# Ejemplos



# Reconocimiento en oclusión



# Otro ejemplo



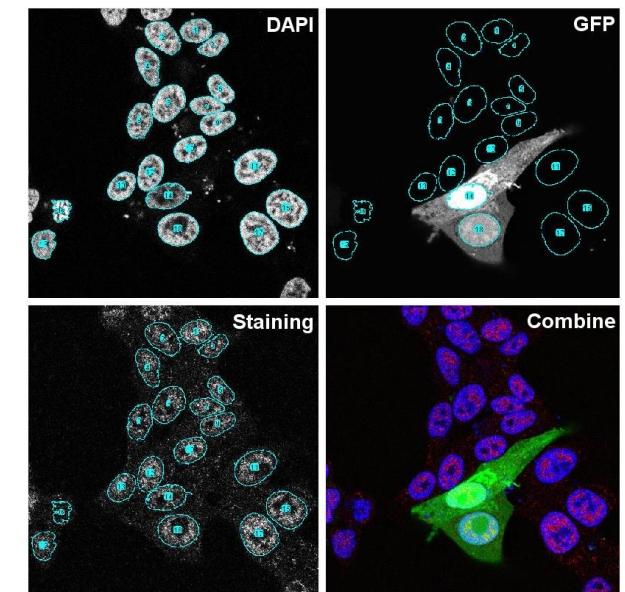
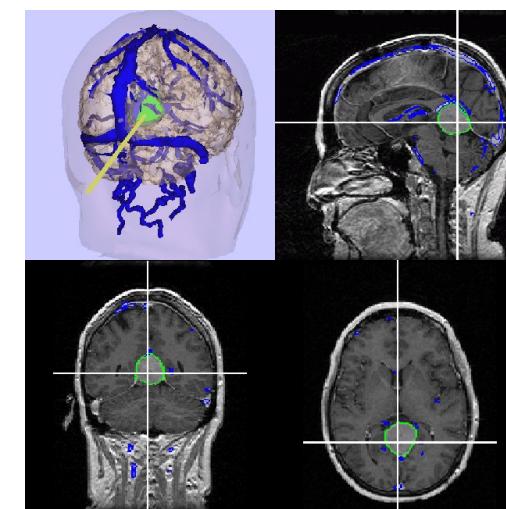
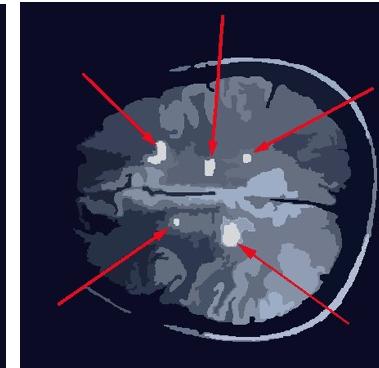
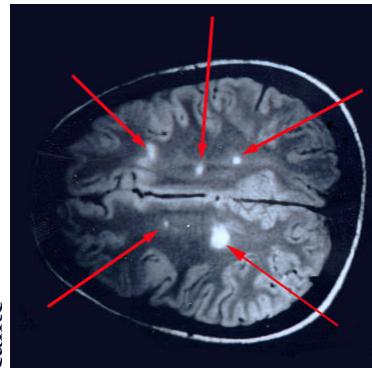
Extracción de características, segmentación y reconocimiento de objetos

# Índice

- Transformada de Hough
- Características SIFT
- Segmentación de imágenes
  - Algoritmo de las K-medias
  - Segmentación basada en regiones

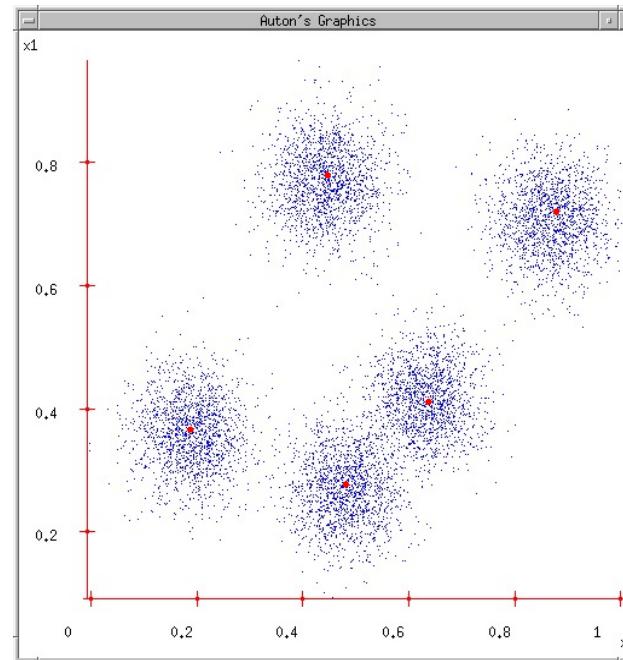
# Segmentación de imágenes

- La segmentación de imágenes es el proceso de extraer zonas de la imagen con el mismo color/nivel de gris/textura para identificarlas automáticamente



# Algoritmo de las K-medias

- Encuentra las medias de las distribuciones (*clusters*) (sirve para cualquier problema de aprendizaje)
- Es necesario conocer el número de distribuciones (*clusters*) distintas existentes

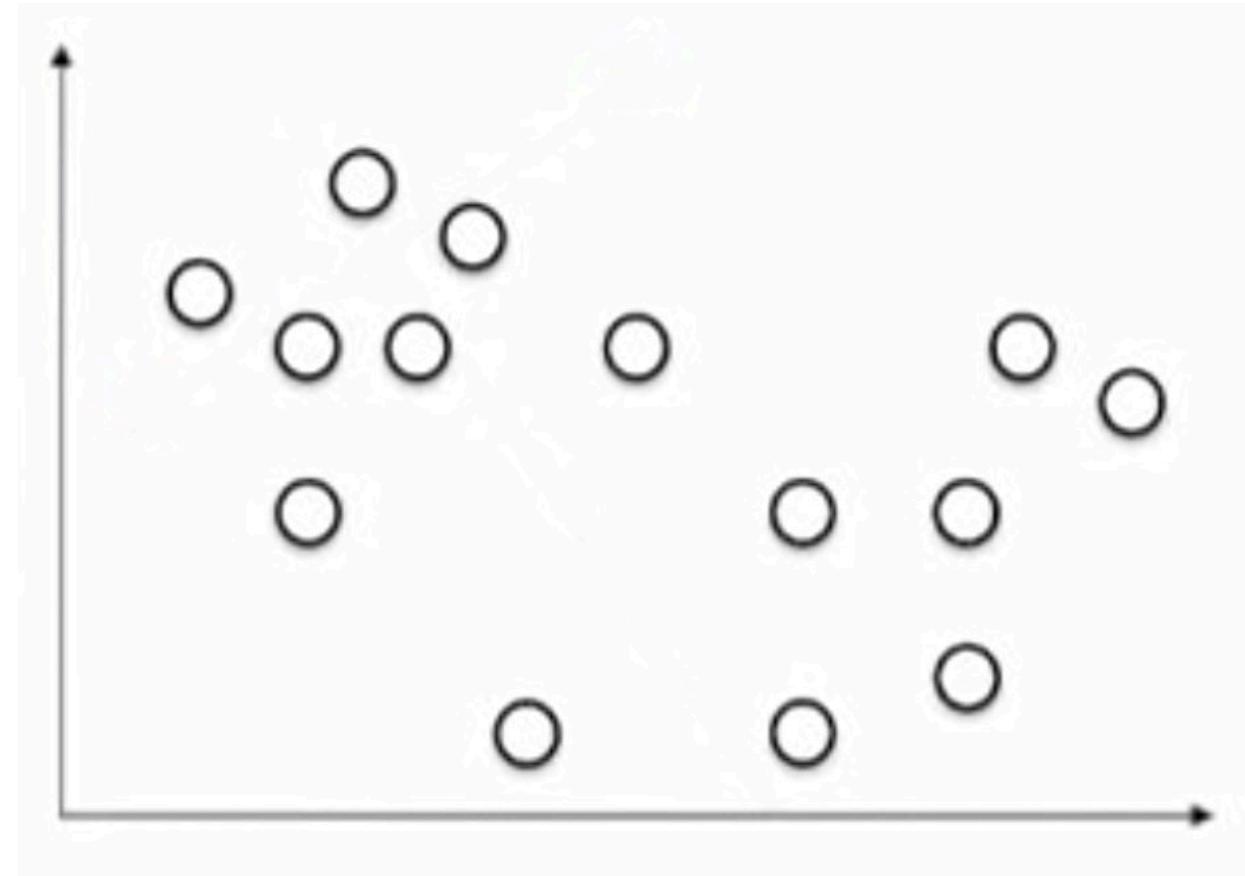


# Algoritmo de las K-medias

- 1) Inicialización: se buscan  $k$  puntos (medias)
- 2) Hacer
  - 1) Pertenencia: para cada punto, se encuentra la distribución (*cluster*) más cercana (el  $k$  más cercano)
  - 2) Cálculo de las nuevas medias
- 3) Hasta que no haya cambios

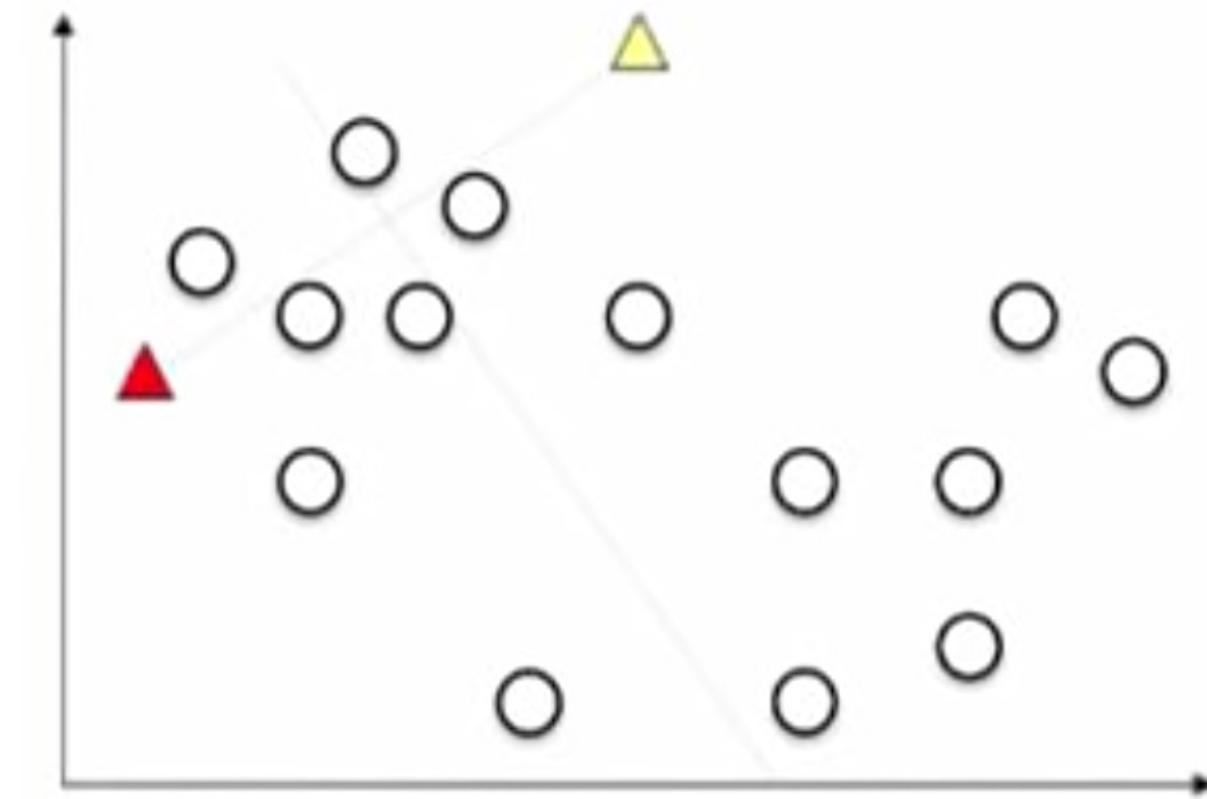


# Algoritmo de las K-medias



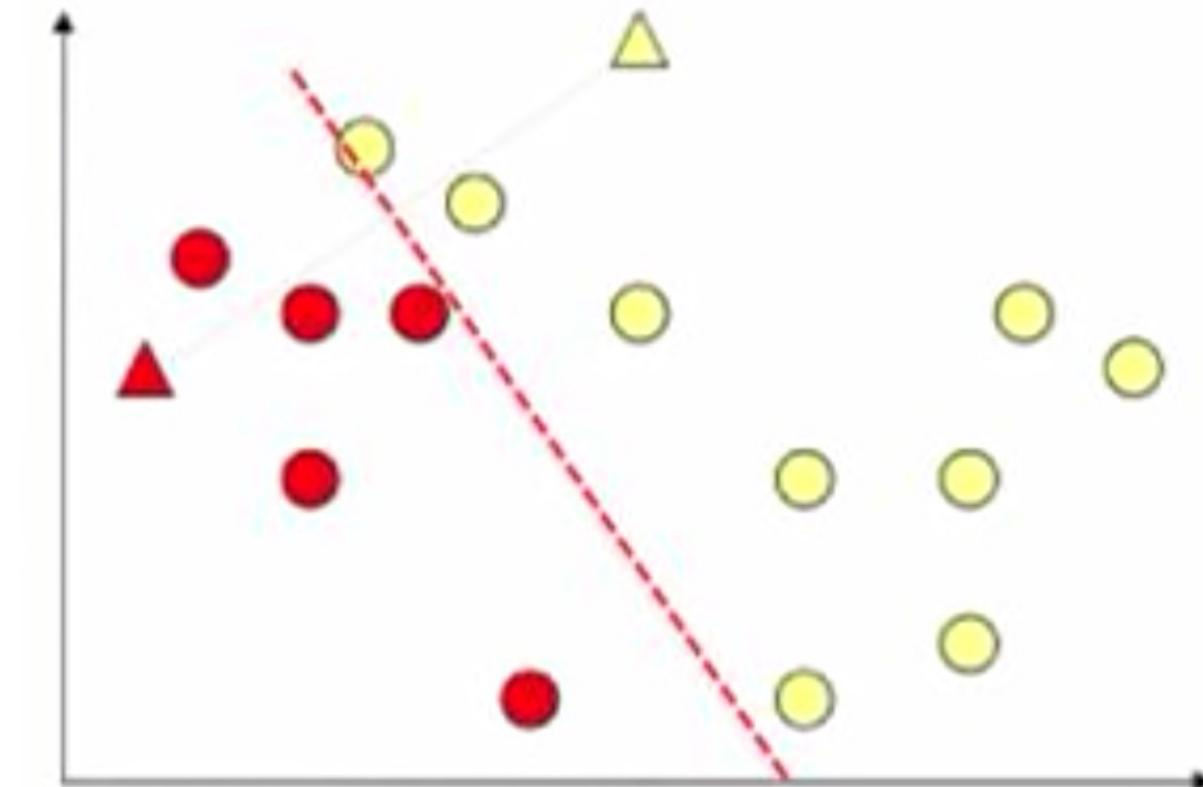
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



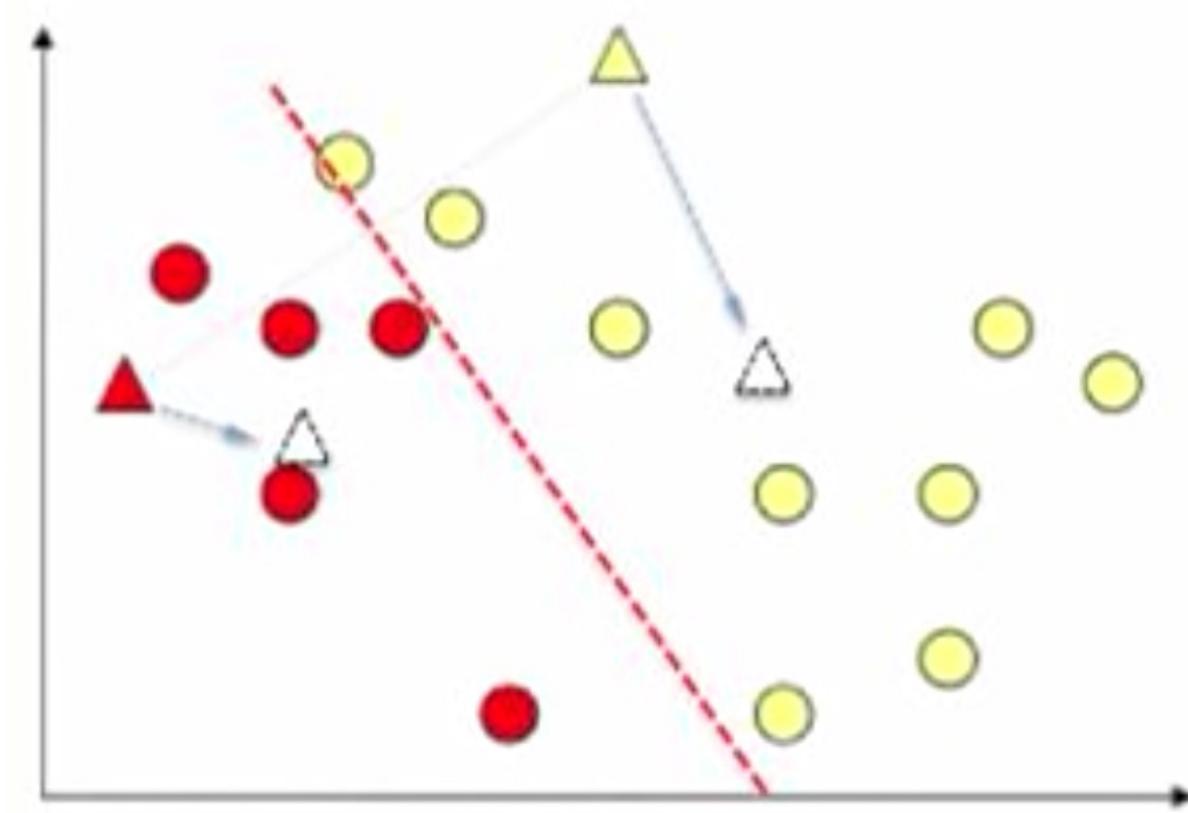
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



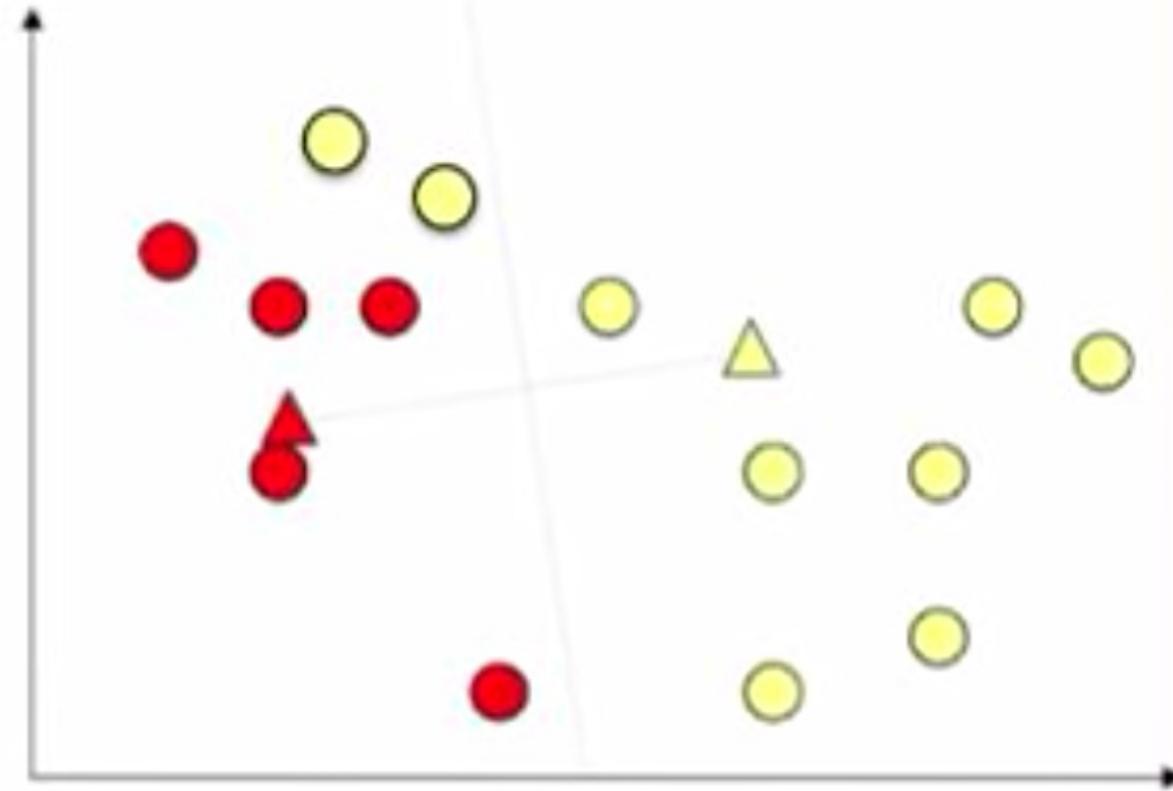
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



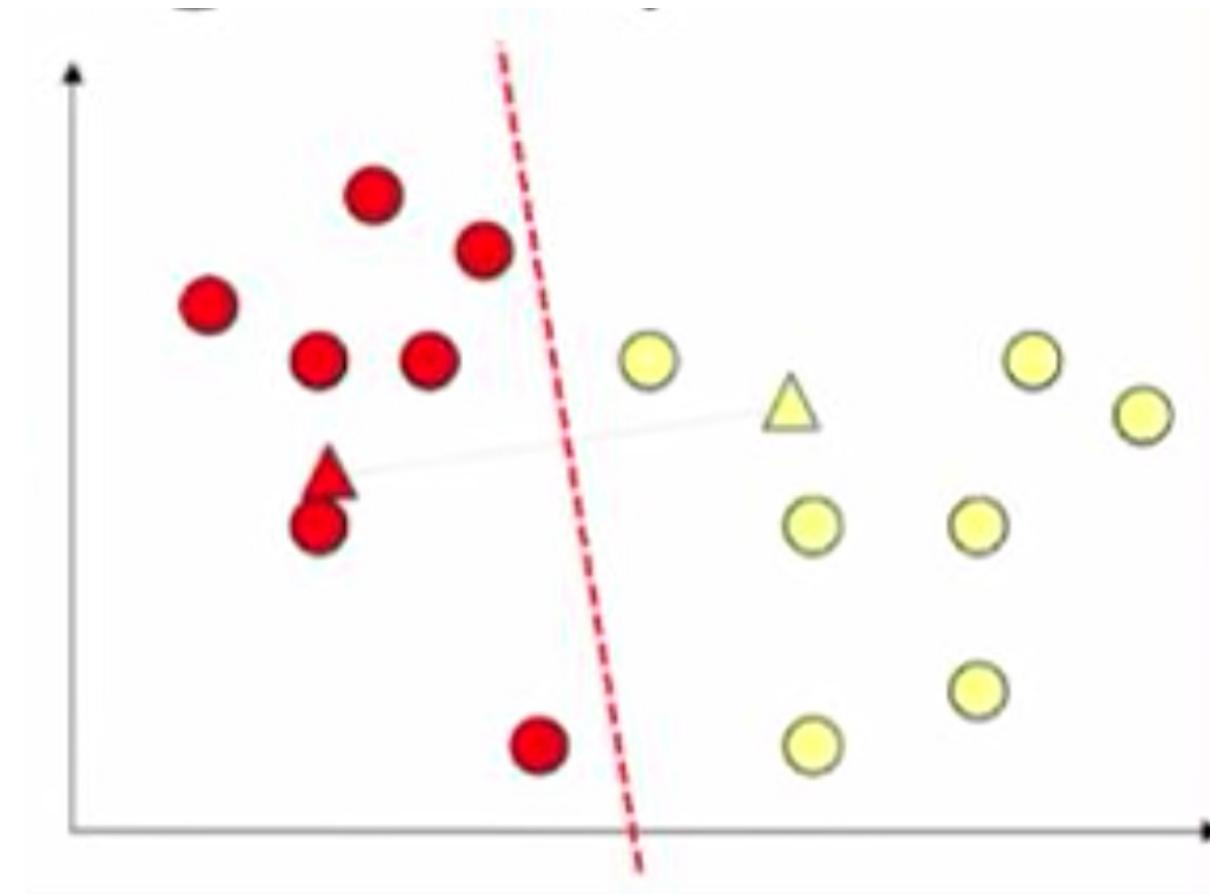
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



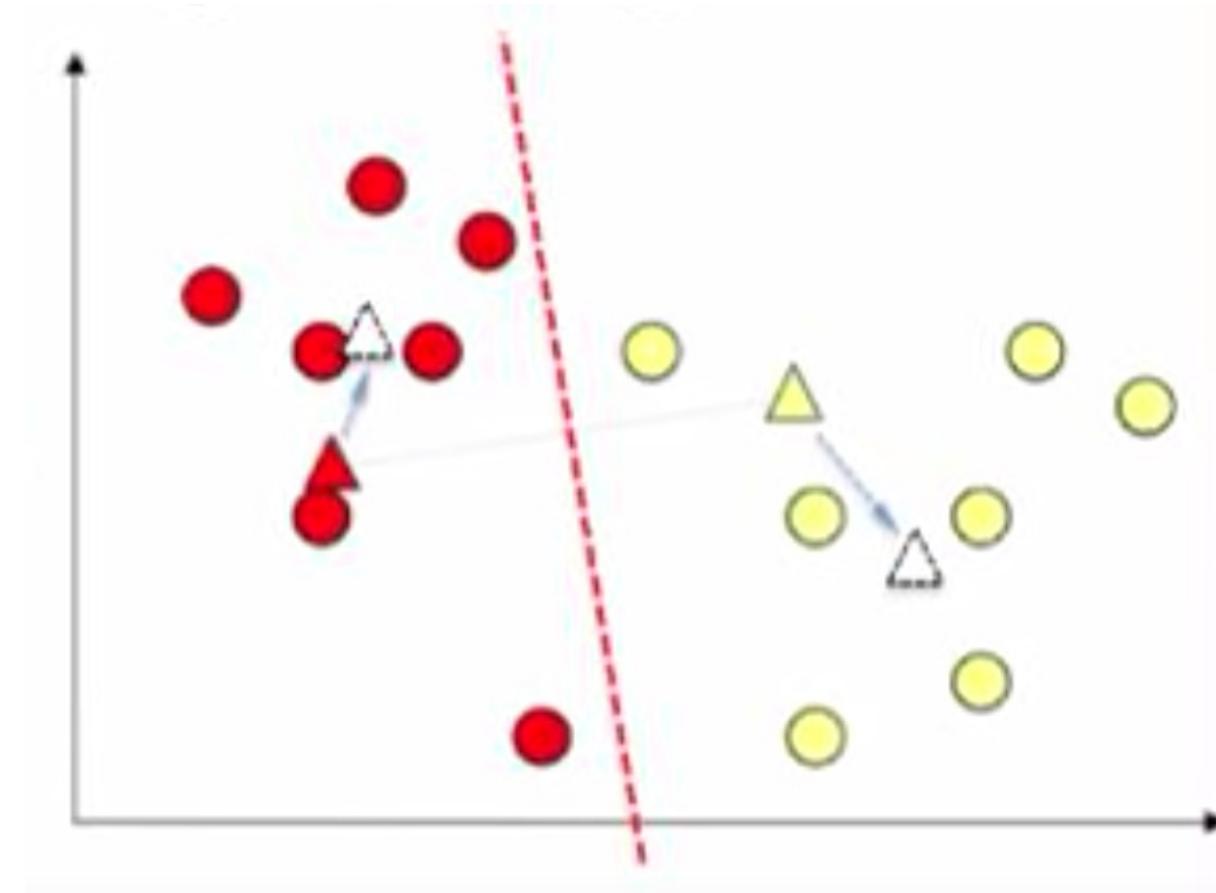
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



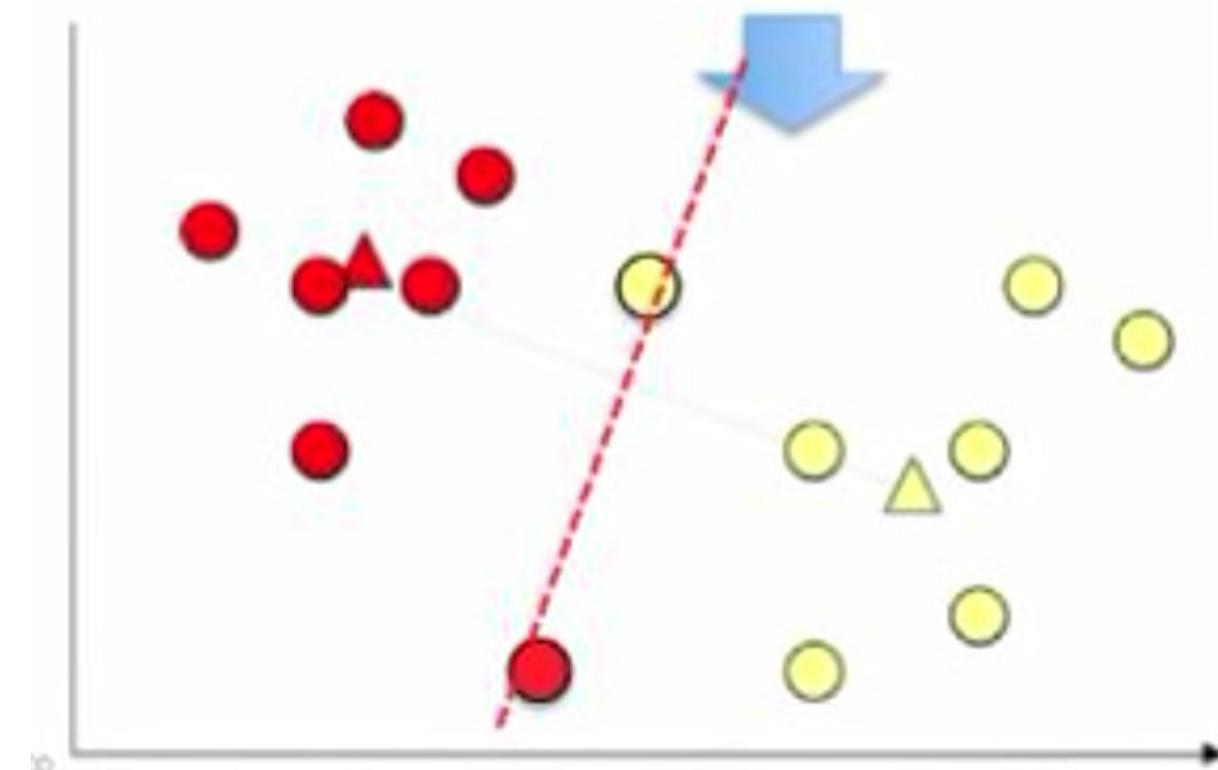
Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Algoritmo de las K-medias



Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

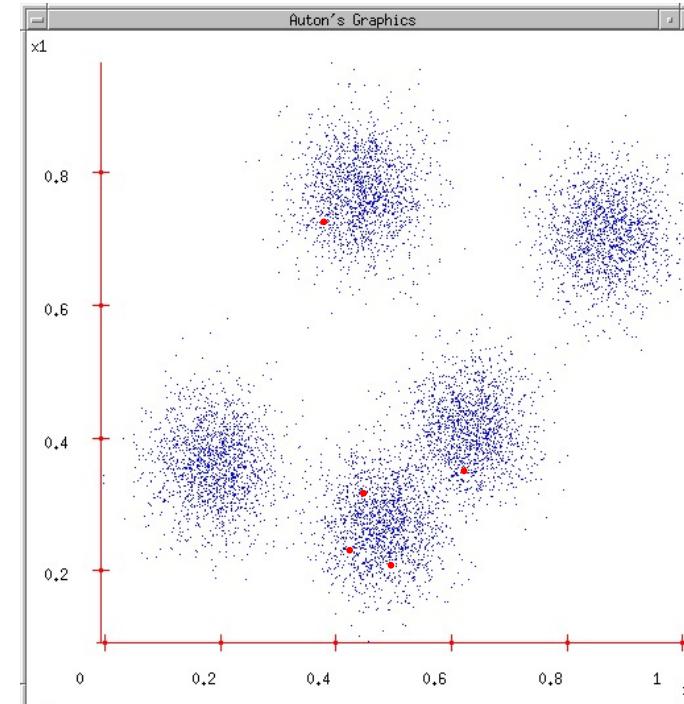
# Algoritmo de las K-medias



Imágenes obtenidas de: [https://www.youtube.com/watch?v=\\_aWzGGNrcic](https://www.youtube.com/watch?v=_aWzGGNrcic)

# Inicialización

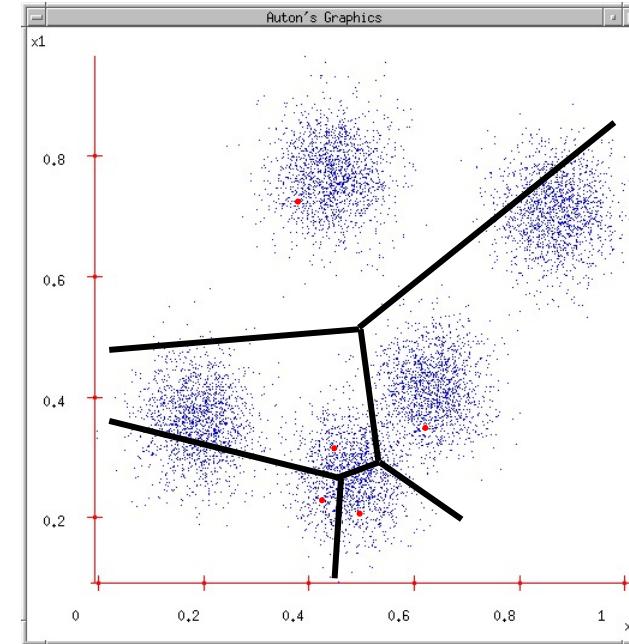
- Para elegir la inicialización, podemos:
  - Asignar las medias de manera aleatoria.
  - Redistribuir las medias de manera uniforme
  - Usar alguna heurística (por ejemplo, en secuencias de imágenes, usar las medias resultado anteriores)



# Cálculo de la pertenencia

- Dependerá del problema:
  - Distancia euclídea: el  $k$  más cercano
  - Probabilidad de pertenencia a un cluster: usando su media y varianza

$$P(x) = \frac{1}{\sqrt{\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right)$$

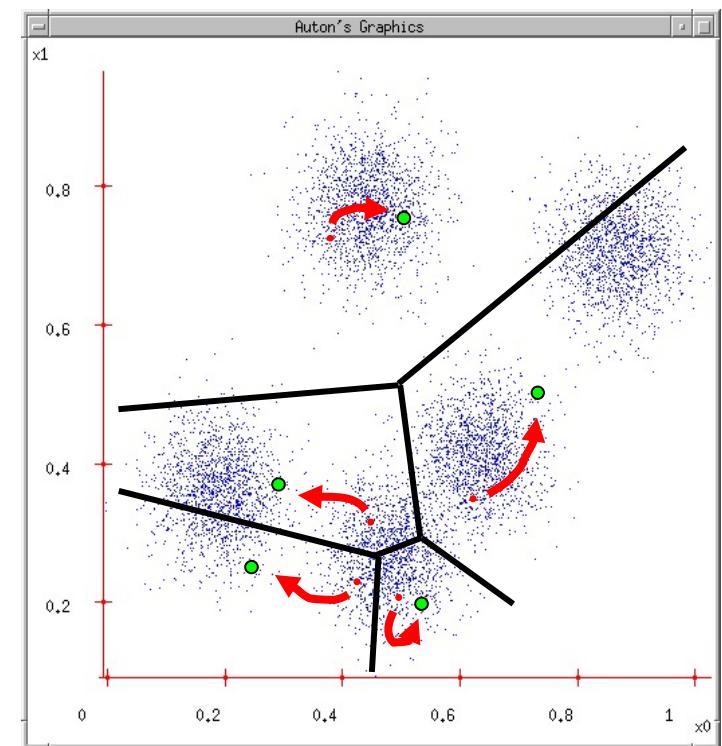


# Cálculo de las nuevas medias

- Para cada cluster, recalcular su media (y varianza, si es el caso), usando todos los puntos que pertenecen a ese cluster

- Si es distancia euclídea:  $\mu_{cluster} = \frac{1}{n} \sum_{i=1}^n x_i$

- Si es probabilidad:  $\mu_{cluster} = \frac{\sum_{i=1}^n p(x_i) x_i}{\sum_{i=1}^n p(x_i)}$



# K-medias para segmentación de imágenes

- Imágenes de gris: se usan los valores de gris, no las posiciones de los píxeles.
- Imágenes de color:
  - Usar directamente los valores R,G,B
  - Convertir a H,S,B y usar la componente H



K=2 63ms



K=4 353ms



K=8 1100ms



probabilidad K=4 112ms

# Características de las K-medias

- Es necesario indicar K
- Una mala inicialización puede llevar más tiempo
- Puede no encontrar la solución más óptima
- En su modo probabilístico:
  - Se puede aplicar a cualquier tipo de distribución (texturas)
  - Se pueden usar distintas medidas de distancia a distribución (Mahalanobis, Kullback-Leiber)
- Se conoce también como: segmentación EM, Fuzzy K-means

# Índice

- Transformada de Hough
- Características SIFT
- Segmentación de imágenes
  - Algoritmo de las K-medias
  - Segmentación basada en regiones

# Segmentación basada en regiones

- El objetivo es encontrar regiones de la imagen homogéneas según algún criterio.
  - Hay dos maneras:
    - Crecimiento de regiones (*region growing*): empezamos con regiones pequeñas (semillas) y las hacemos crecer o bien las mezclamos, usando un criterio de similaridad.
    - Partición de regiones (*region splitting*): empezamos con regiones grandes (incluso toda la imagen) y las vamos dividiendo usando un criterio de homogeneidad.
  - Existen métodos que combinan las dos maneras.

# Crecimiento de regiones

- Haralick and Saphiro “Image segmentation techniques” Computer Vision, Graphics and Image Processing. V. 29 (1). 1985
- Asume que una región es un conjunto de píxeles conectados
- Sea  $R$  una región de  $N$  píxeles

$$\mu = \frac{1}{N} \sum_{x_i \in R} I(x_i)$$

$$\sigma^2 = \sum_{x_i \in R} (I(x_i) - \mu)^2$$

- Tenemos que definir un test de similaridad para saber si un nuevo píxel  $y$ , adyacente a algún píxel de  $R$ , se puede añadir a la región



# Crecimiento de regiones: test de similaridad

- Podemos definir el siguiente test:
  - Calculamos  $T$ :
$$T = \sqrt{\frac{(N-1)N}{(N+1)} \frac{(y-\mu)^2}{\sigma^2}}$$

N: número de elementos que forman la región  
 $\mu$ : valor medio de la región  
 $\sigma^2$ : varianza de la región
- Si  $T$  es lo suficientemente pequeño, añadimos  $y$  a  $R$
- Si se añade, debemos actualizar la media y varianza:

$$\mu_{nueva} = \frac{N\mu + y}{(N+1)} \quad \sigma_{nueva}^2 = \sigma^2 + (y - \mu_{nueva})^2 + N(\mu_{nueva} - \mu)^2$$

- Si no se añade, empezamos a calcular una **nueva** región con dicho píxel

# Crecimiento de regiones: semillas

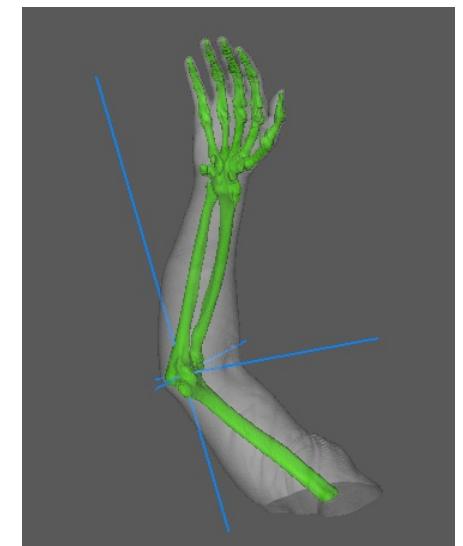
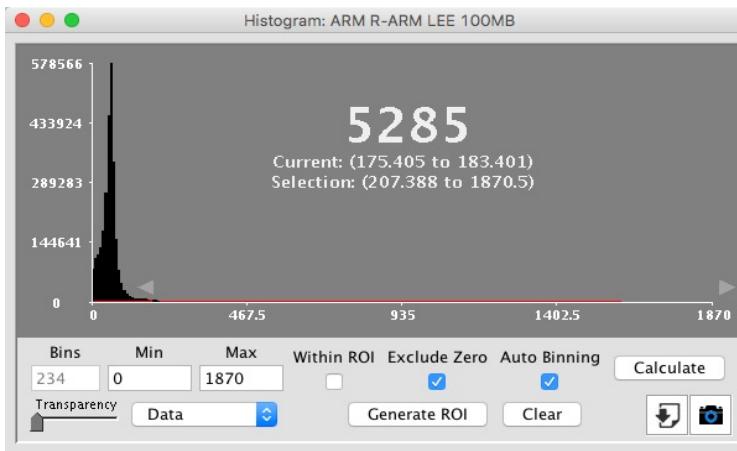
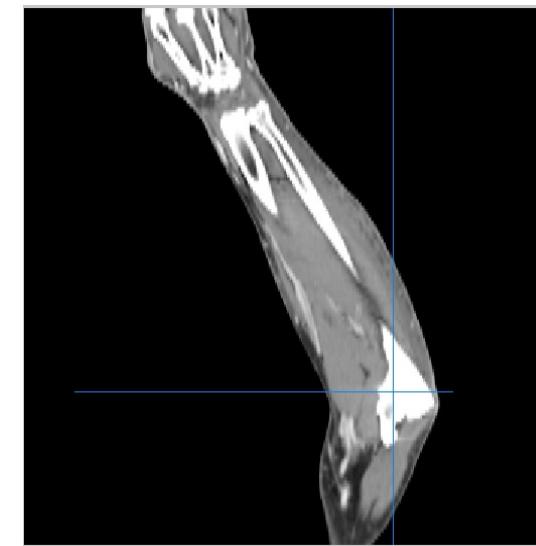
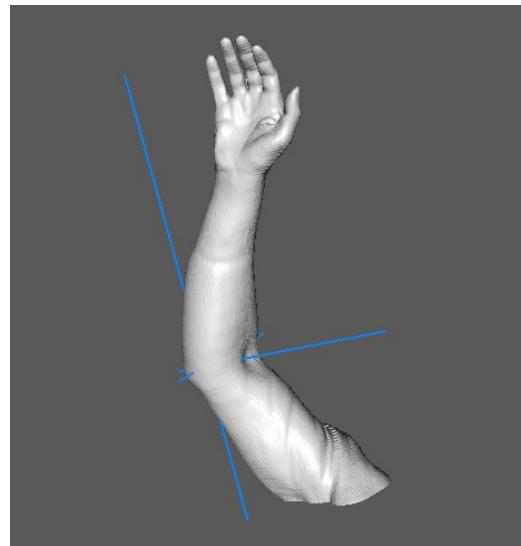
- El método anterior lo podemos hacer empezando por cualquier píxel de la imagen
- Sin embargo, es mejor lanzar un conjunto de “semillas”, es decir, lanzar varios puntos de partida
- Para ello, se puede hacer lo siguiente:
  - Se aplica un detector de aristas
  - Los puntos cuyo valor de magnitud de gradiente estén próximos a cero, serán “valles”, es decir, puntos dentro de regiones
  - Usaremos esos puntos para hacer una “inundación” del valle (los usamos como “semillas” del método anterior)



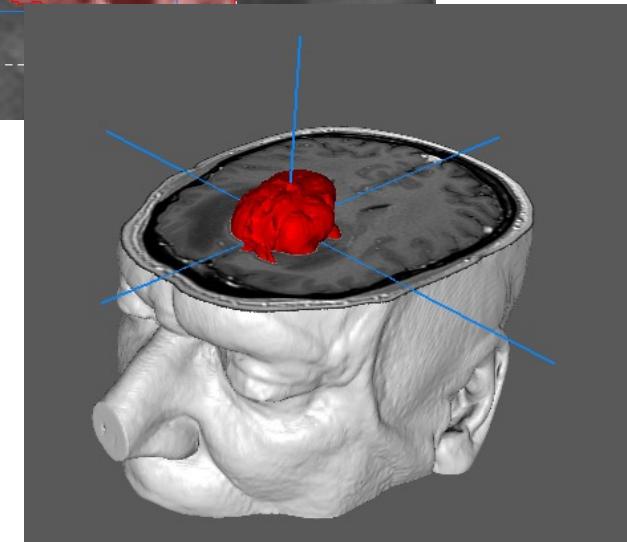
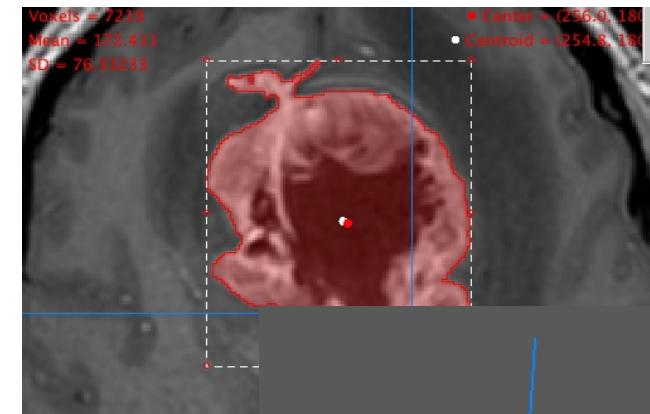
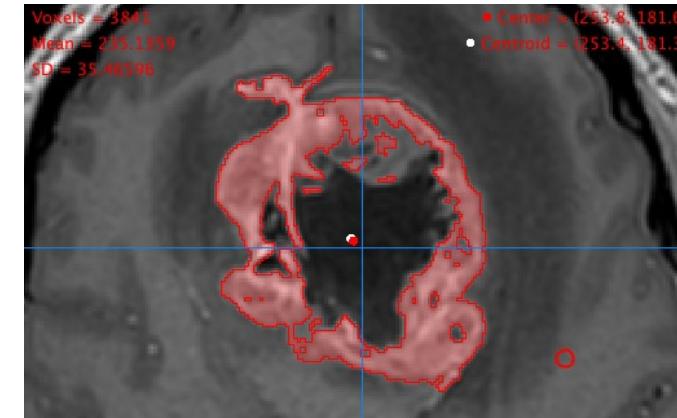
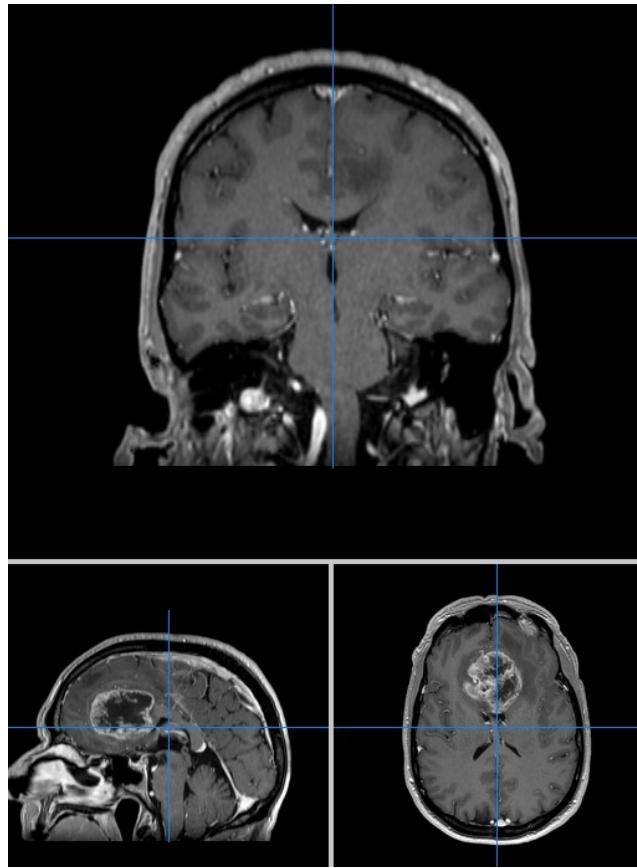
# Partición de regiones

- Este proceso es el inverso del anterior.
- Partimos una única región (toda la imagen) y usamos algún criterio de homogeneidad para partir la región.
  - Si se cumple el criterio, no se sigue dividiendo.
  - Si no se cumple, se divide.
- Se puede usar el histograma para dicho criterio: ver si existen “valles”, que pueden servir a su vez para dividir la región en otras más pequeñas.
- Conlleva una complejidad mayor, al tener que manejar alguna estructura de datos adicional.

# Ejemplos Segmentación



# Ejemplos Segmentación





# Referencias

- Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 (January, 1972)
- Transformada de Hough: [http://en.wikipedia.org/wiki/Hough\\_transform](http://en.wikipedia.org/wiki/Hough_transform)
- K-medias: [http://en.wikipedia.org/wiki/K-means\\_clustering](http://en.wikipedia.org/wiki/K-means_clustering)