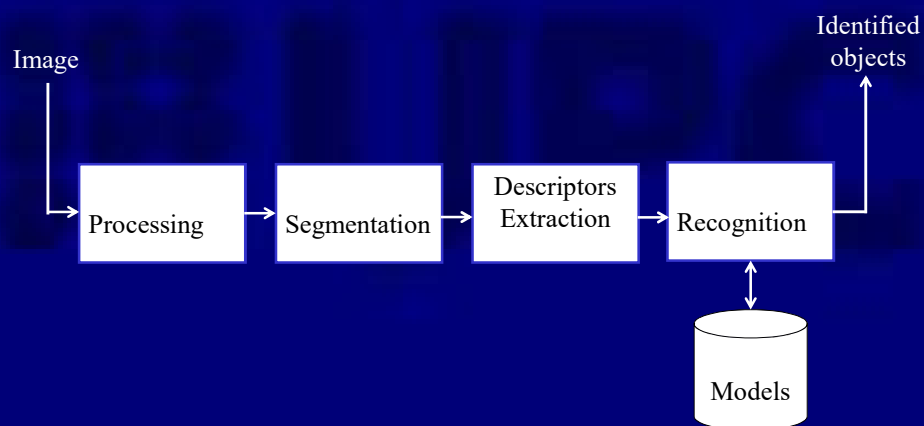


Segmentació de la imatge

A computer vision system



Introducció

- La segmentació consisteix en dividir la imatge en regions amb característiques similars

Quines característiques ?

Què vol dir similars ??

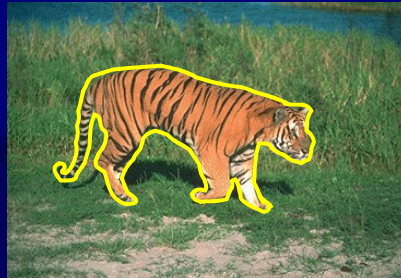
- Cada regió es representa com a una vora tancada
- Els problema sorgeix quan les imatges són sorolloses
- Diferents enfocs:
 - Binaritzat
 - Segmentació basada en contorns
 - Algorisme de Canny
 - Watershed
 - Segmentació basada en clustering
 - K-means



Introducció

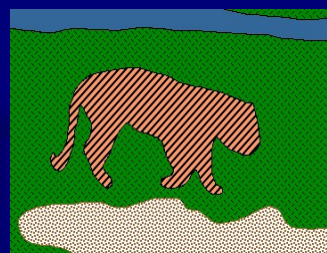


Extraure objectes



i això... com ho fem?

From images to objects



Què és el que defineix un objecte?

Problema subjectiu !!

Les lleis Gestalt treballen aquest tema:

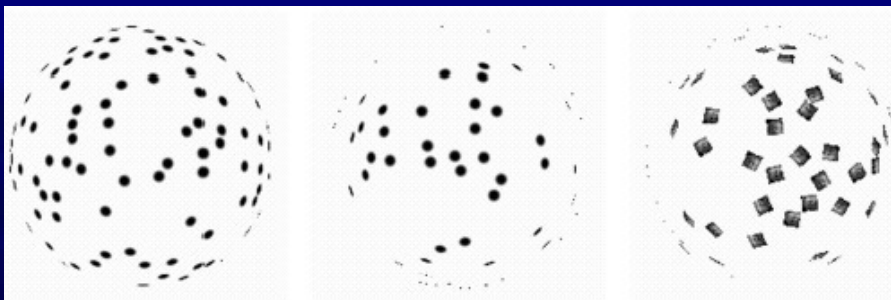
proximitat, similitut, continuïtat, tancament, common fate

Críteris d'homogeneitat ?









UPC

Críteris d'homogeneitat ?



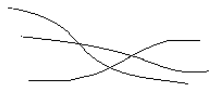



UPC

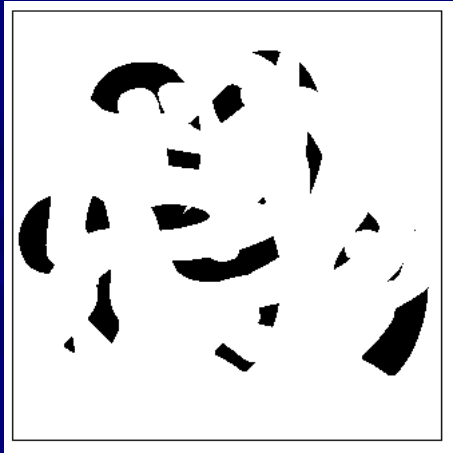
Críteris d'homogeneitat ?

	Not grouped
	Proximity
	Similarity
	Similarity
	Common Fate
	Common Region

Críteris d'homogeneitat ?

	Parallelism
	Symmetry
	Continuity
	Closure

Criteria d'homogeneity ?



UPC

Criteria d'homogeneity ?



UPC

Regions i contorns

Els **contorns** es troben buscant les diferències entre els valors de píxels adjacents.

Les **Regions** es troben buscant les similituts entre els valors de píxels adjacents.

- Per segmentar hem d'usar algun criteri per agrupar o separar píxels en regions:
 - Nivell de gris
 - Color
 - Textura
 - Profunditat
 - Etc.



Segmentació basada en contorns

- Els humans som molt sensibles als contorns
- El resultat de l'extracció de contorns no és una segmentació
- Cal un post-processat que combini els contorns en cadenes per a obtenir les vores
- Apareixen contorns que no es corresponen amb les vores: contorn no fi, soroll.
- No apareixen contorns que si es corresponen amb les vores



From edges to contours

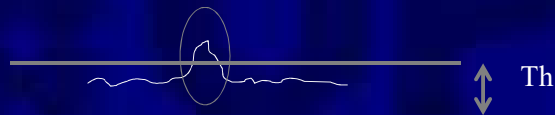
Recordem el problema...



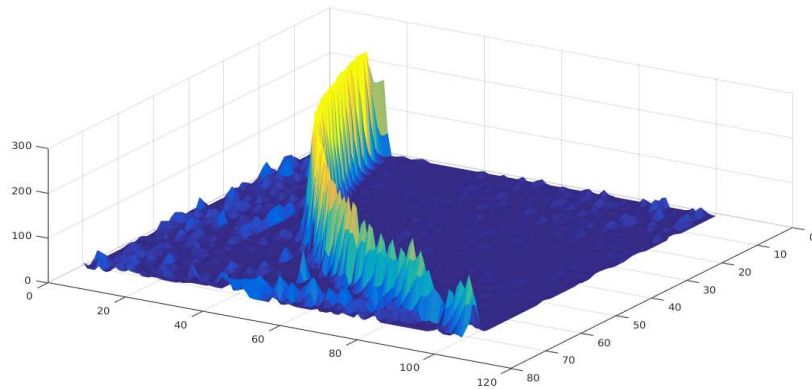
How do we “detect” edges?

E_s has large values at edges:

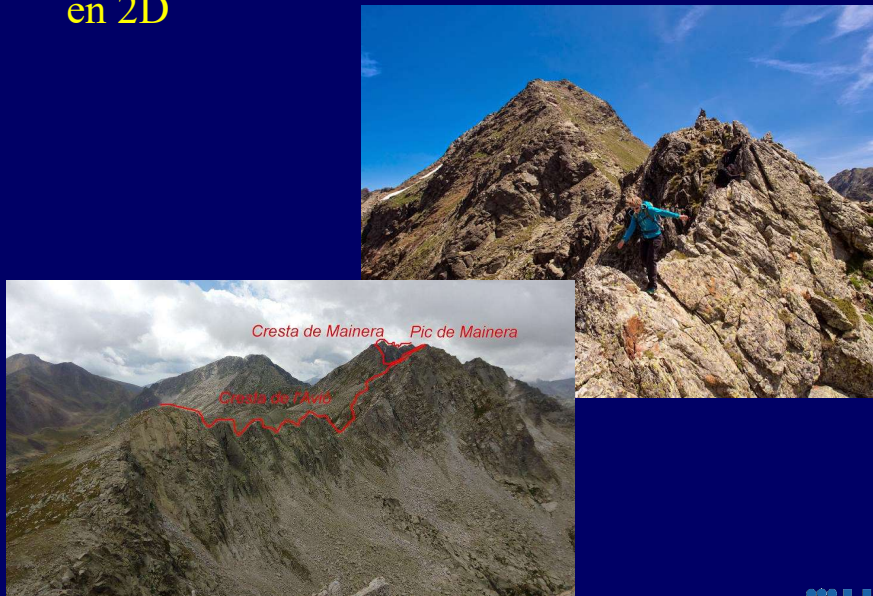
Find local maxima



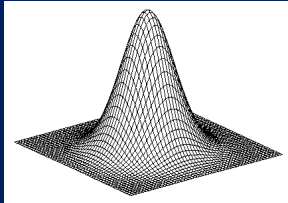
... but it also may have wide ridges around the local maxima (large values *around* the edges)



... no sembla fàcil buscar aquests màxims 'locals' en 2D

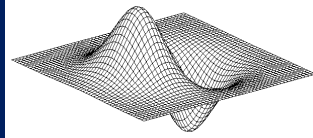


Representació 3D dels filtres



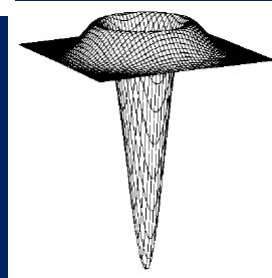
Gaussian

$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



Derivative of Gaussian

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$



Laplacian of Gaussian

$$\nabla^2 h_{\sigma}(u, v)$$

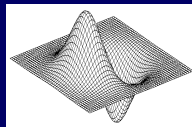
On ∇^2 és l'operador Laplacian: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$



L'operador de Canny

- Treballa amb la primera derivada de la gaussiana (filtrat gaussià + realçat per gradient)

-1. Realçat per DoG

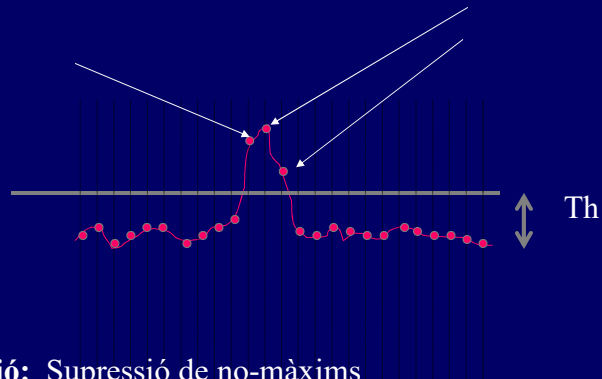


-2. Supressió dels no-màxims. Obtenim contorn fi.

-3. Binaritzat amb histèresi

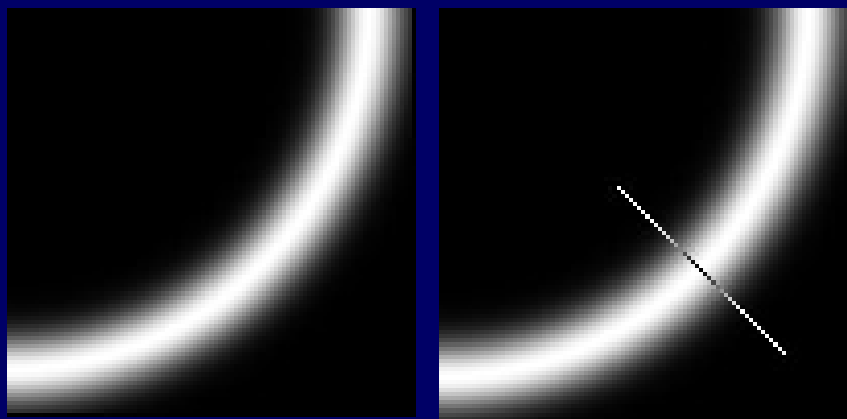


... i com soluciono el problema de trobar els màxims locals ???



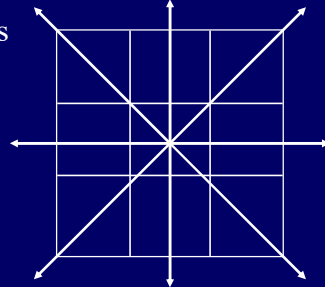
Solució: Supressió de no-màxims

Supressió de no-màxims de Canny



Supressió de no-màxims de Canny

1. Quantificar 8 possibles direccions



2. Comparar el píxel central amb els dos veïns en la direcció del gradient
3. Si el valor del píxel central és menor que algun dels veïns, marcar-lo per a esborrar
4. Esborrar tots els píxels marcats



Supressió de no-màxims de Canny



Imatge Original



Magnitud del
gradient

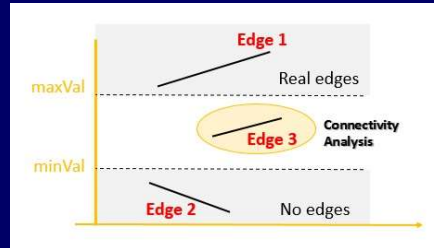


No-màxims
suprimits



L'operador de Canny. Binaritzat de contorns

- Binaritzat amb histèresi:



1. Marcar tots els píxels de contorn amb magnitud superior a T_H com a correctes
2. Marcar tots els píxels de contorn amb magnitud inferior a T_L com a incorrectes
3. Repassar cada píxel de contorn entre T_L i T_H
4. Si aquests píxels tenen un veí marcat com a correcte, els marquem com a correctes
5. Repetim els dos últims passos fins que el procés s'estabilitzi



L'operador de Canny. Binaritzat de contorns

Imatge
Original



Contorns
forts



Connectats!

forts + febles
connectats



Contorns
febles



courtesy of G. Loy

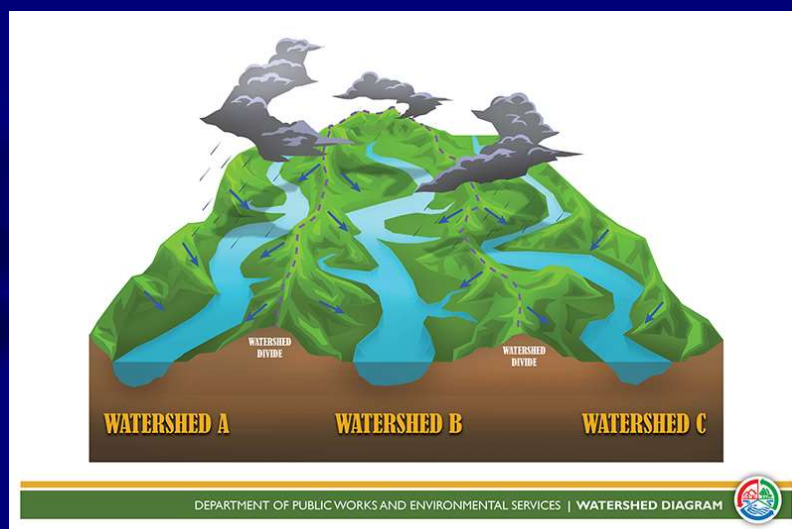


L'operador de Canny. Resultats



UPC

Watershed



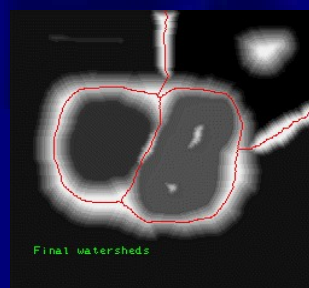
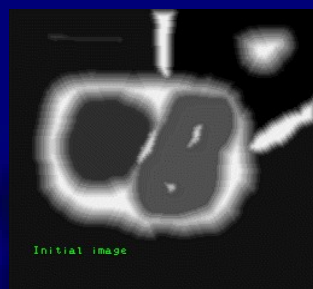
UPC

Segmentació per watershed

- Eina per a segmentació morfològica
- La imatge multinivell s'interpreta com una superfície topogràfica
- Descripció intuïtiva: inundació
 1. Representar la imatge com un relleu de plàstic
 2. Foradem el plàstic en els mínims locals de la imatge (o en els markers)
 3. Submergim el plàstic en aigua progressivament
 4. L'aigua que entra per cada forat s'etiqueta amb l'identificador del marker corresponent
 5. Quan aigües de diferents procedències es troben, aixequem una pared per a evitar que es barregin
 6. L'algorisme segueix fins que tot ha estat anegat
 7. Finalment les parets representen les línies de watershed
 8. La imatge queda particionada en regions



watershed



Watershed. Exemple

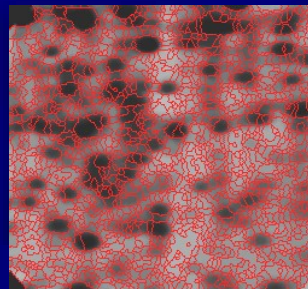
- Metodologia per segmentar (1)
 1. Obtenir la imatge gradient
 2. watershed a la imatge gradient a partir dels seus mínims



UPC

Watershed. La sobresegmentació

- Metodologia per segmentar (1)
 1. Obtenir la imatge gradient
 2. watershed a la imatge gradient a partir dels seus mínims



- Problema: Sobre-segmentació deguda al soroll o a petites irregularitats en la imatge gradient

UPC

watershed amb markers

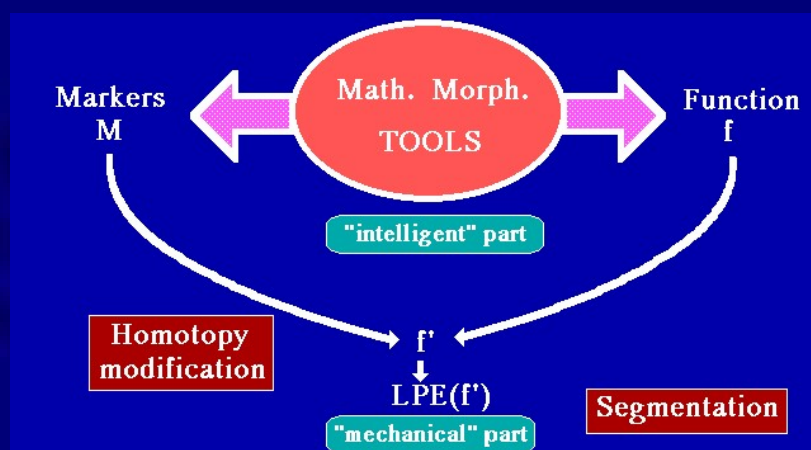
- Metodologia per segmentar (2)

1. Obtenir la imatge gradient
2. Buscar markers, un per cada objecte a segmentar
3. watershed a la imatge gradient a partir dels markers



- Ara, el problema de la segmentació consisteix en trobar els markers

Watershed. El paradigma

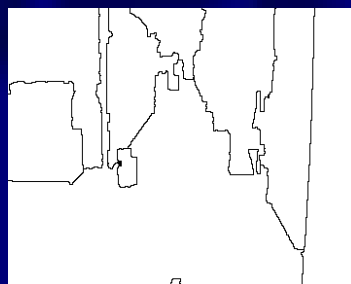
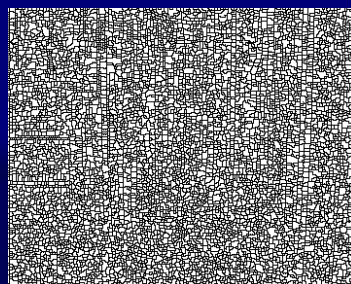


watershed amb markers

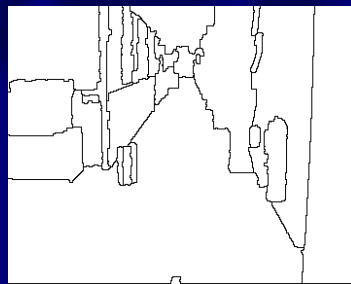
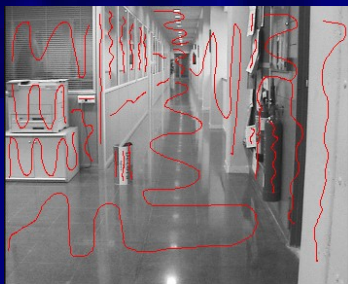
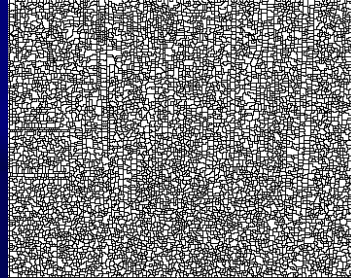
- Segmentació de carreteres. Els markers s'han col·locat manualment



watershed amb markers



watershed amb markers



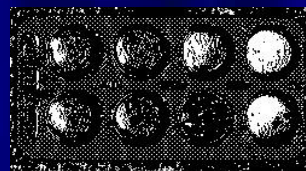
UPC

Obtenció automàtica de markers: extrems regionals

- Un màxim (mínim) regional és una regió connexa on tots els píxels veïns tenen un valor estrictament menor (major).
- Les imatges reals tenen masses màxims i mínims regionals.
- Cal filtrar per a reduir el nombre de màxims i mínims.
- Els extrems regionals trobats solen ser bons markers per al watershed.



Original



Extrems regionals

UPC

Obtenció automàtica de markers: extrems regionals

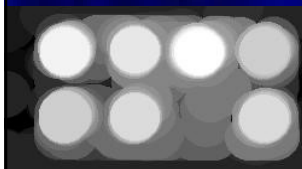
- Les tècniques de filtratge es basen en:
 - contrast: h-max, h-min
 - forma: Opening
 - mida: AreaOpening



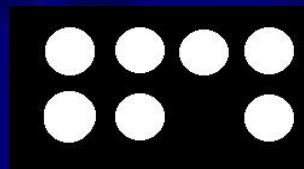
Original



Extrems regionals



Opening



Extrems regionals de la imatge filtrada

Obtenció automàtica de markers: SKIZ

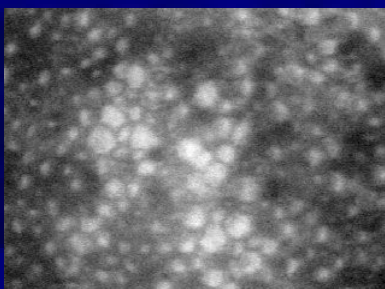
- Skeleton by Influence Zones
- Zona d'influència : Conjunt de píxels d'una imatge binària que estan més propers a una component connexa que a la resta
- Vores de les zones d'influència : SKIZ



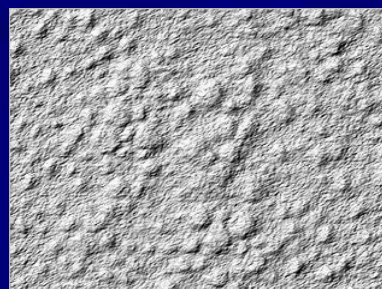
S'implementa mitjançant un watershed sobre la transformada de distància

watershed amb obtenció automàtica de marques

- Volem segmentar les cel·les d'una imatge de la cornea.
- La imatge és molt sorollosa

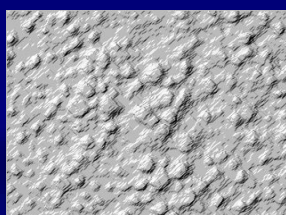


Imatge Original

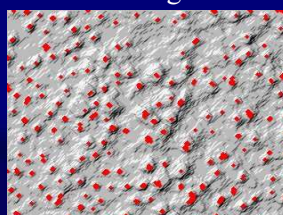


Imatge en relleu

- Per a eliminar soroll apliquem un filtre open-close
- Per trobar els markers busquem els màxims regionals

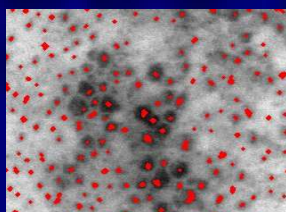


filtrada

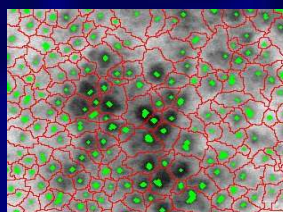


màxims regionals

- Per a trobar el marker del fons fem el SKIZ

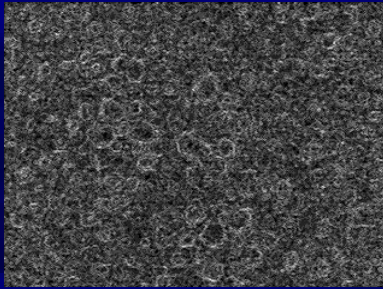


markers de les cel·les

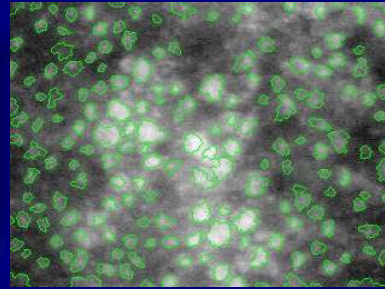


markers de cel·les i fons

- Segmentem la imatge amb watershed sobre la imatge gradient usant els markers obtinguts



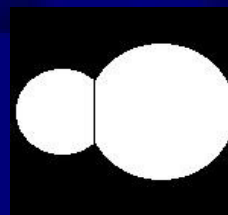
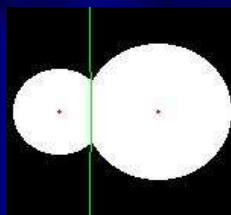
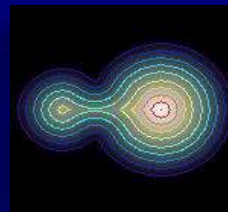
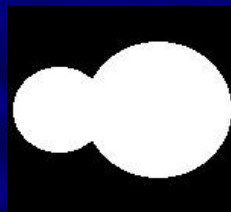
Imatge gradient



Imatge segmentada

Separació de touching-blobs

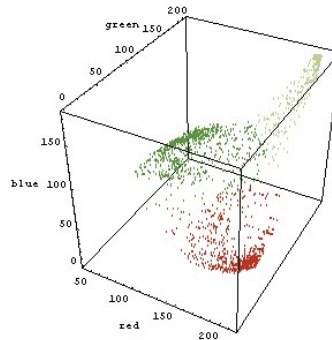
- Transformada de distància + watershed
- Els màxims de la transformada de distància seran els markers
- Watershed sobre la inversa de la transformada de distància



K-Means Clustering



■ RGB vector



K-means clustering minimizes
$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of } i\text{'th cluster}} \|x_j - \mu_i\|^2 \right\}$$

K-means clustering

K-means clustering algorithm

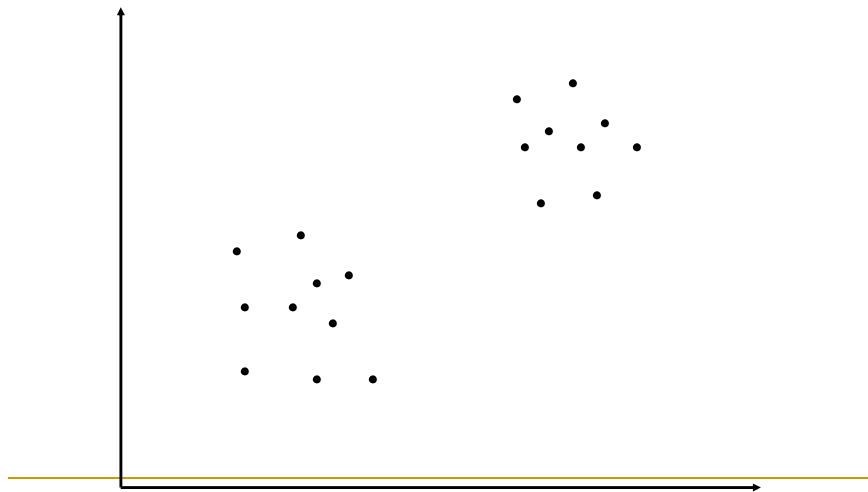
1. Randomly initialize the cluster centers, c_1, \dots, c_K
2. Given cluster centers, determine points in each cluster
 - For each point p , find the closest c_i . Put p into cluster i
3. Given points in each cluster, solve for c_i
 - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Properties

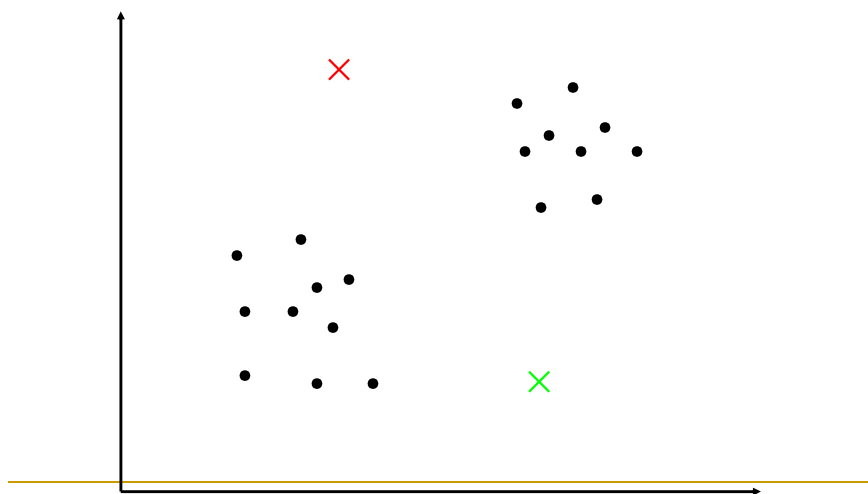
- Will always converge to *some* solution
- Can be a “local minimum”
 - does not always find the global minimum of objective function:

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

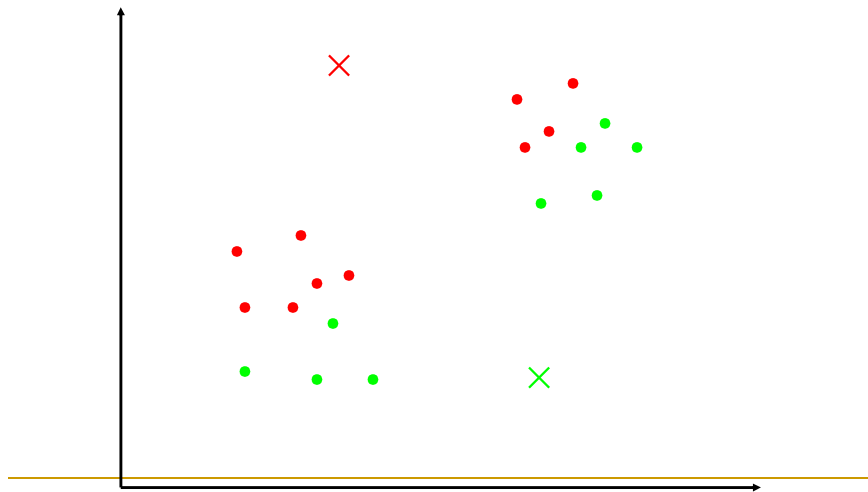
K-Means Clustering



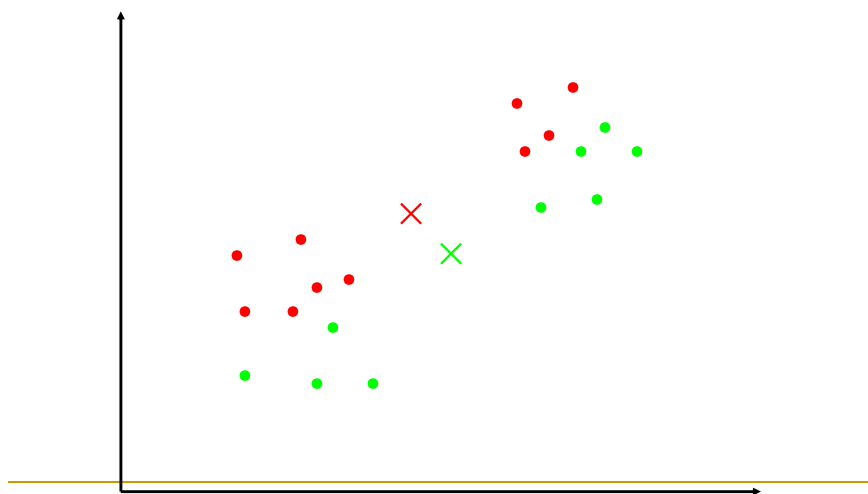
K-Means Clustering



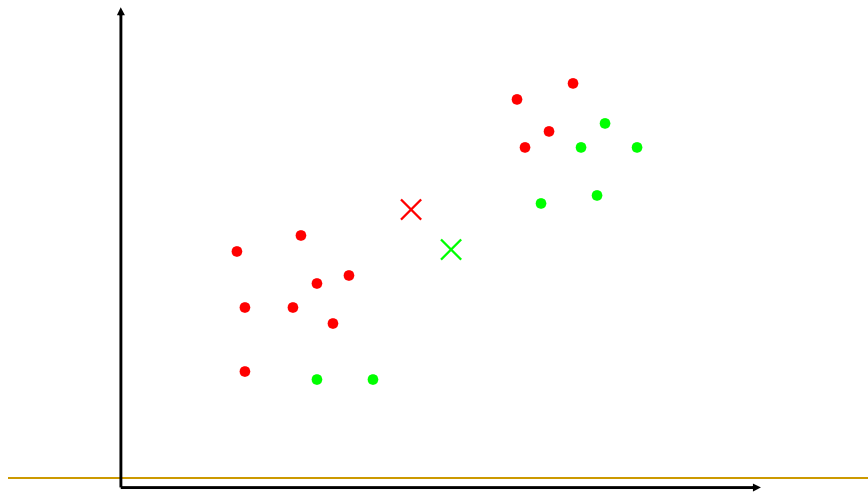
K-Means Clustering



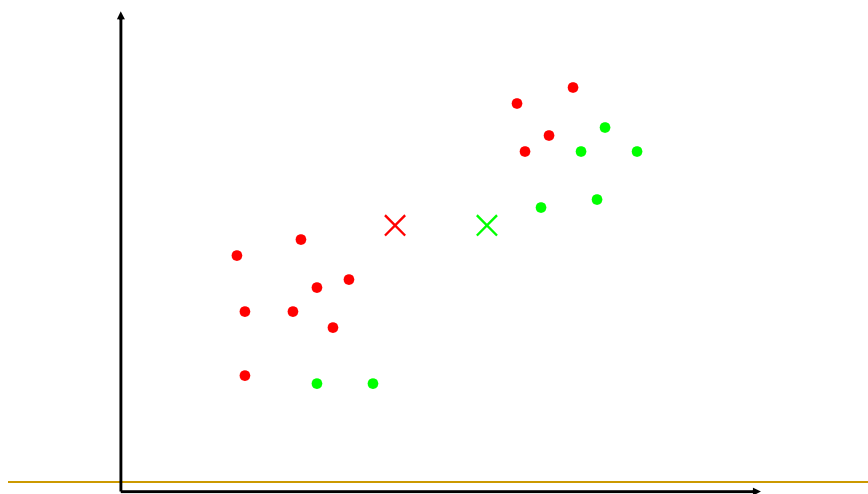
K-Means Clustering



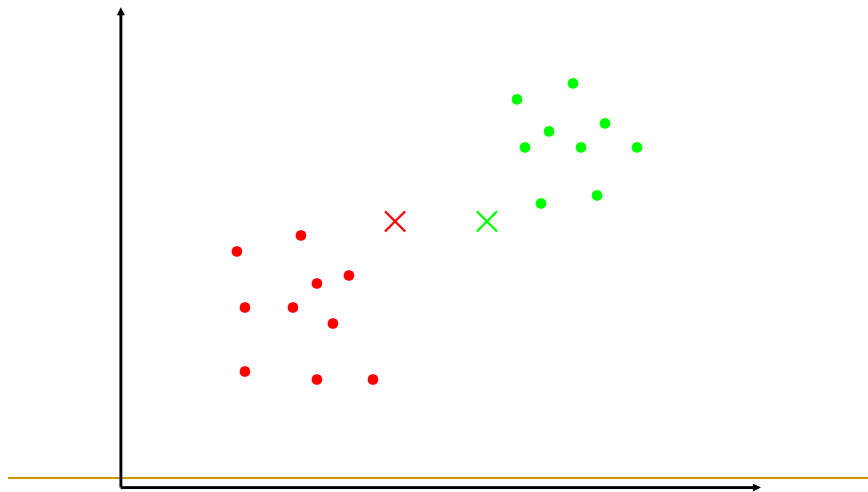
K-Means Clustering



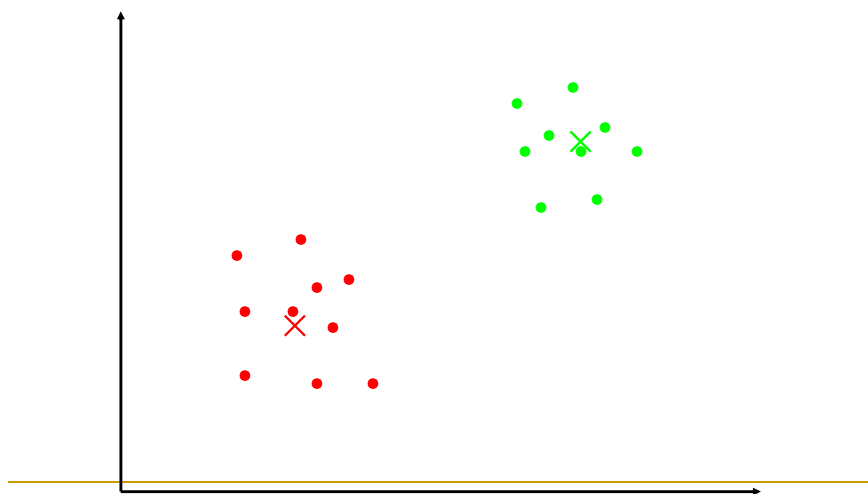
K-Means Clustering



K-Means Clustering



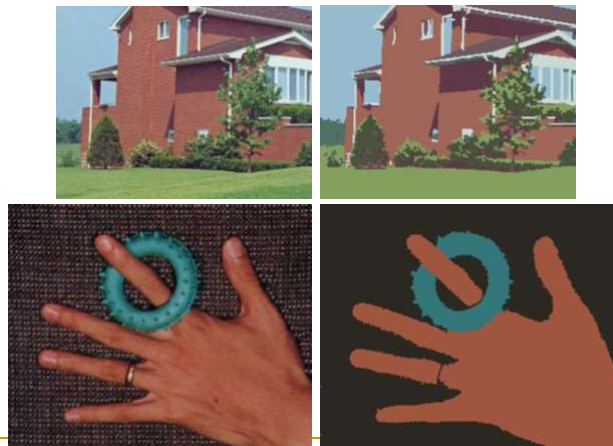
K-Means Clustering



Clustering



■ Example

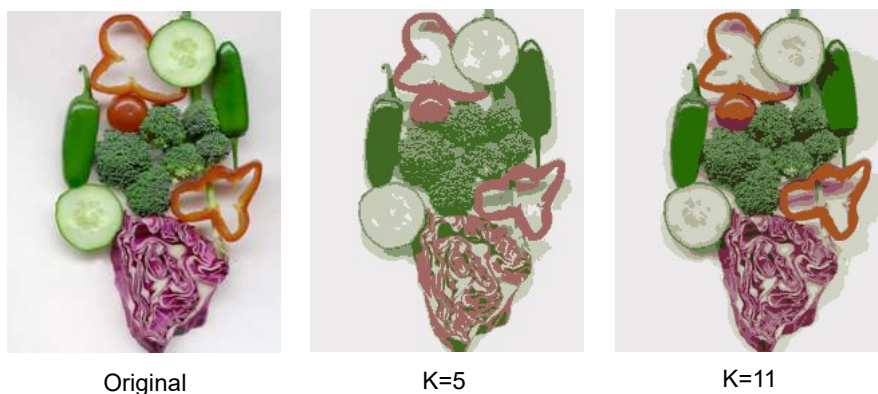


D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.

K-Means Clustering



■ Example

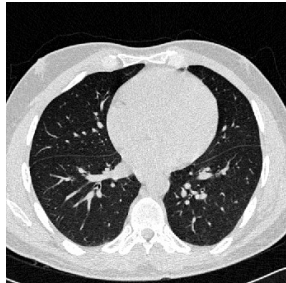


Original

K=5

K=11

K -means Image Segmentation



An image (I)



Three-cluster image (J) on gray values of I

Note that K -means result is “noisy”

K -means: summary

- Algorithmically, very simple to implement
- K -means converges, but it finds a local minimum of the cost function
- Works only for numerical observations
- K is a user input; alternatively BIC (Bayesian information criterion) or MDL (minimum description length) can be used to estimate K
- Outliers can cause considerable trouble to K -means

