Master in Computer Vision *Barcelona*

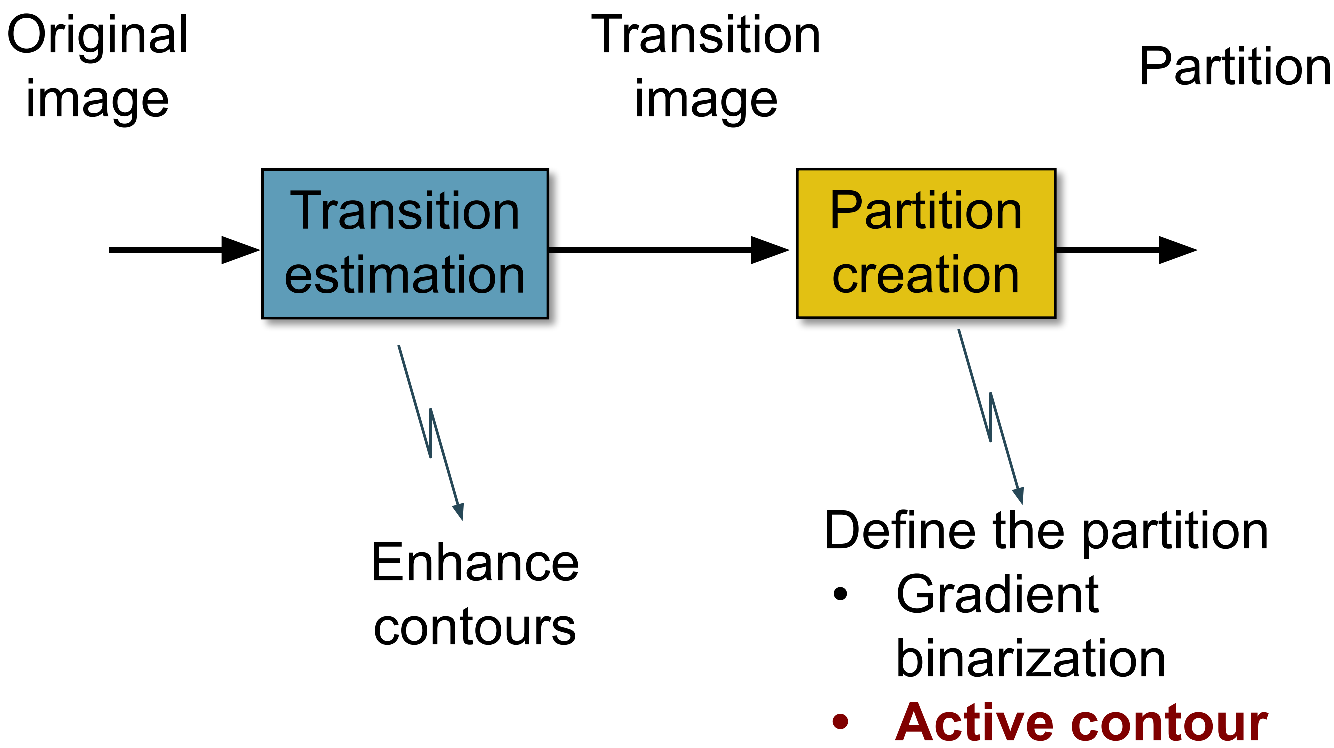
Module: Introduction to human and computer vision

Lecture 10: Grouping, segmentation and classification (II)

Lecturer: Ramon Morros

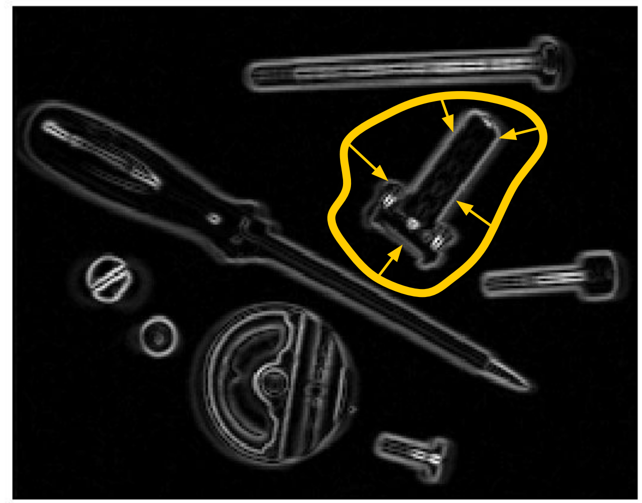
SEGMENTATION

Segmentation: Transition based



Transition based – Active contours

- Robust strategy: Active contours (Snakes)
  - **Evolution of a closed curve towards the points of high**



  - How to define the curve evolution?
  - How to implement it?
  - How to define the initial curve?

Transition based – Active contours

- Basic idea: Minimize the "length" of the curve
  - If s is the arc-length parametrization of the curve C:

$$\text{Length } = \oint ds$$

$$\text{Geodesic length } = \oint g(\nabla x)\, ds, \text{where g is function of } \nabla x$$

  - Assume g is low when the gradient is high and g∈[0,1], ex:

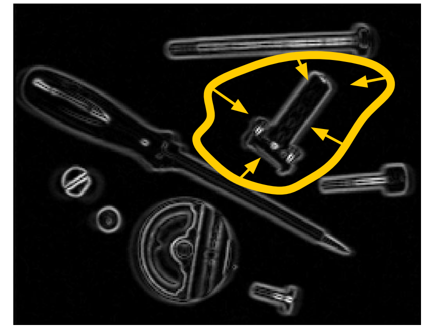
$$g(\nabla x) = \frac{1}{1 + \|\nabla x\|^p}$$

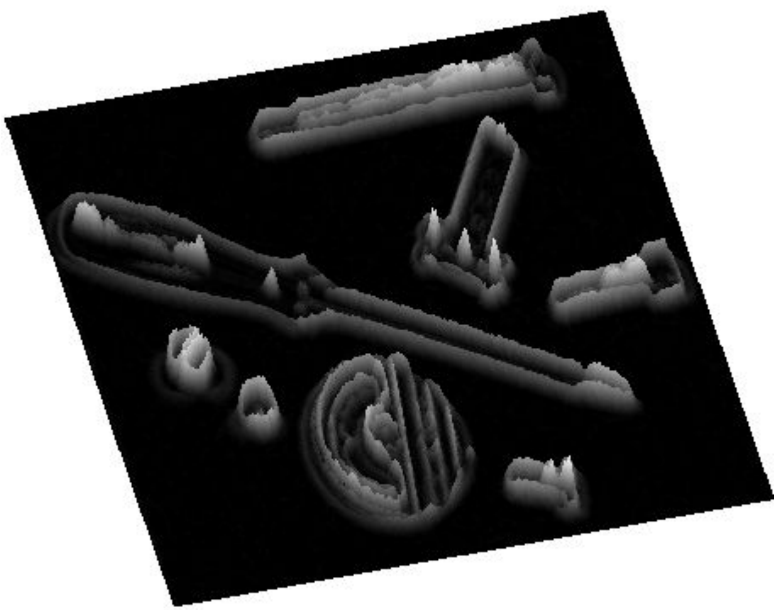
Transition based – Active contours

Find the curve $C(s)$ such that

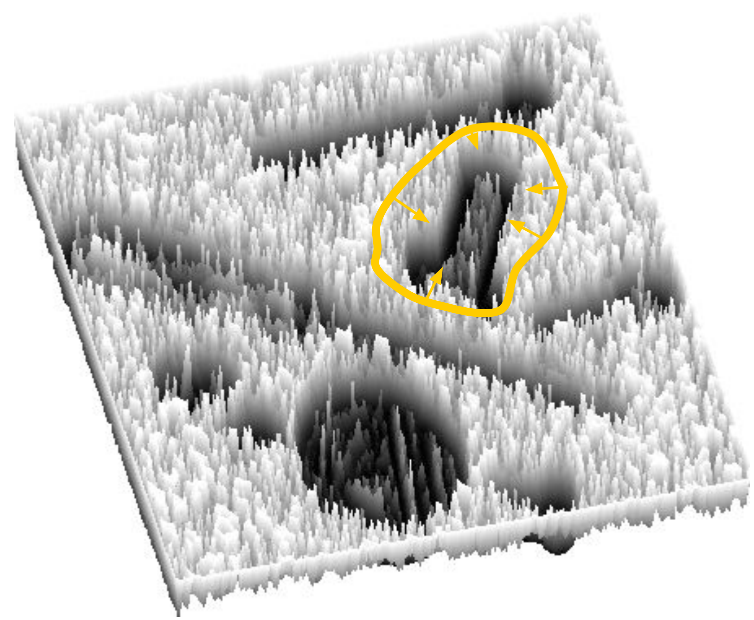
$$\oint g(\nabla x)\,ds \text{ is minimum}$$

Gradient:





3D view of ∇x



3D view of g(∇x)

Transition based – Active contours

- How to find the minimum of a function?:
  - Compute the gradient of the function:

$$\text{Minimum of } E(i) \quad \Rightarrow \quad \frac{dE(i)}{di} = 0$$

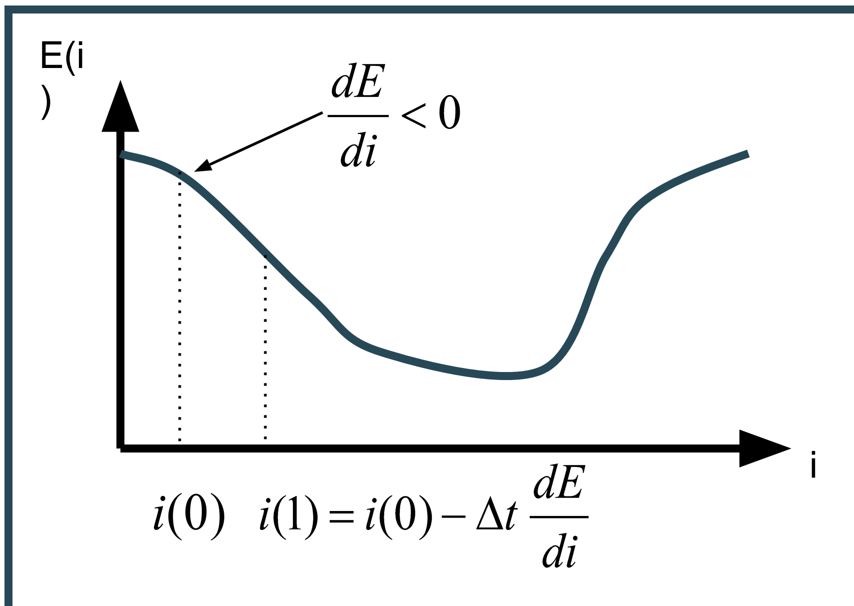
  - Gradient descent strategy: iterative search of i → i(t). It leads to the following PDE:

$$\boxed{\frac{\partial i(t)}{\partial t} = -\frac{\partial E(i)}{\partial i}}$$

In practice:

$$i(t) - i(t-1) \propto -\frac{\partial E(i)}{\partial i}$$

$$i(t) = i(t-1) - \mu \frac{\partial E(i)}{\partial i}$$



Transition based – Active contours

- Gradient descent strategy: iterative search of i → i(t). It leads to:

$$E(C)=\oint g(\nabla x)\,ds \Rightarrow \frac{\partial C}{\partial t}=-\frac{\partial E}{\partial C}=gk\vec{N}-\left\langle \nabla g,\vec{N}\right\rangle \vec{N}$$

Curve curvature

Curve normal

Transition based – Active contours

Brief reminder on differential geometry

- Continuous image model: i,j ∈ R
- Curve parametrization:

$$C(p)=\{(i(p),j(p)),\ p\in[a,b]\}$$

- Arc-length parametrization:

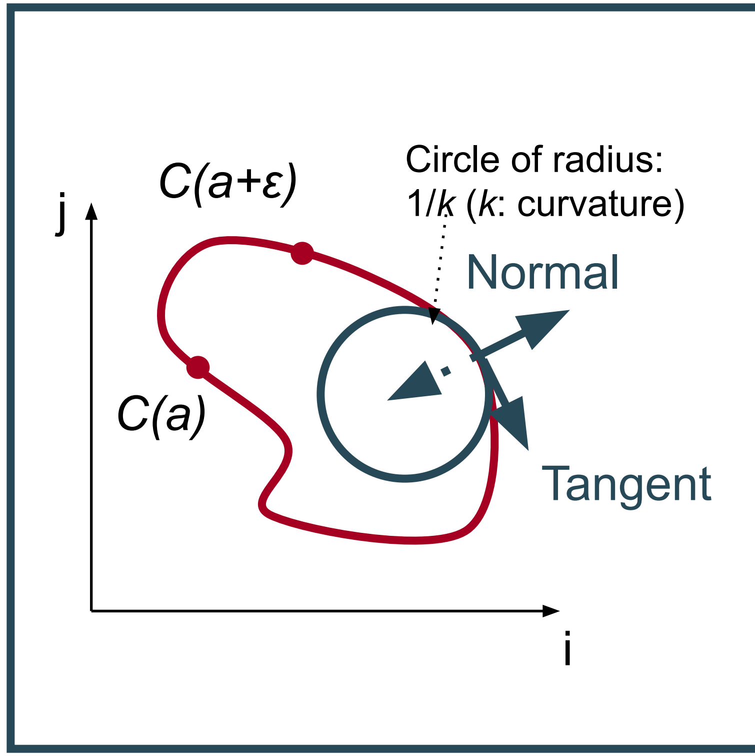
$$s \text{ such that } \left\|\frac{dC}{ds}\right\|=1$$

$$C(s)=\{(i(s),j(s)),\ s\in[0,Length]\}$$

- Tangent and Normal:

$$\vec{T}=\frac{dC}{ds}=\left(\frac{di(s)}{ds},\frac{dj(s)}{ds}\right)$$

$$k\vec{N}=\frac{d^2C}{ds^2}$$



$$\left\langle\frac{dC}{ds},\frac{dC}{ds}\right\rangle=1\Rightarrow\left|\begin{array}{l}\left\langle\frac{dC}{ds},\frac{d^2C}{ds^2}\right\rangle=0\\ \vec{T}\perp\frac{d^2C}{ds^2}=k\vec{N}\end{array}\right.$$

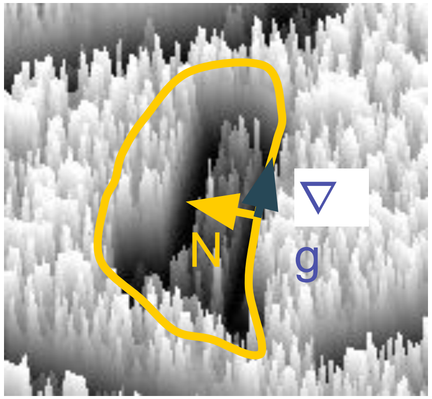
Transition based – Active contours

- Using as criterion the geodesic length:

$$E(C)=\oint g(\nabla x)\,ds \;\Rightarrow\; \frac{\partial C}{\partial t}=-\frac{\partial E}{\partial C}=gk\vec{N}-\langle \nabla g,\vec{N}\rangle \vec{N}$$

Internal force:
- Curvature motion
- Smoothing

External force:
- Force the curve evolution towards minima of g
- The evolution stops when the gradient of g is perpendicular to the curve normal



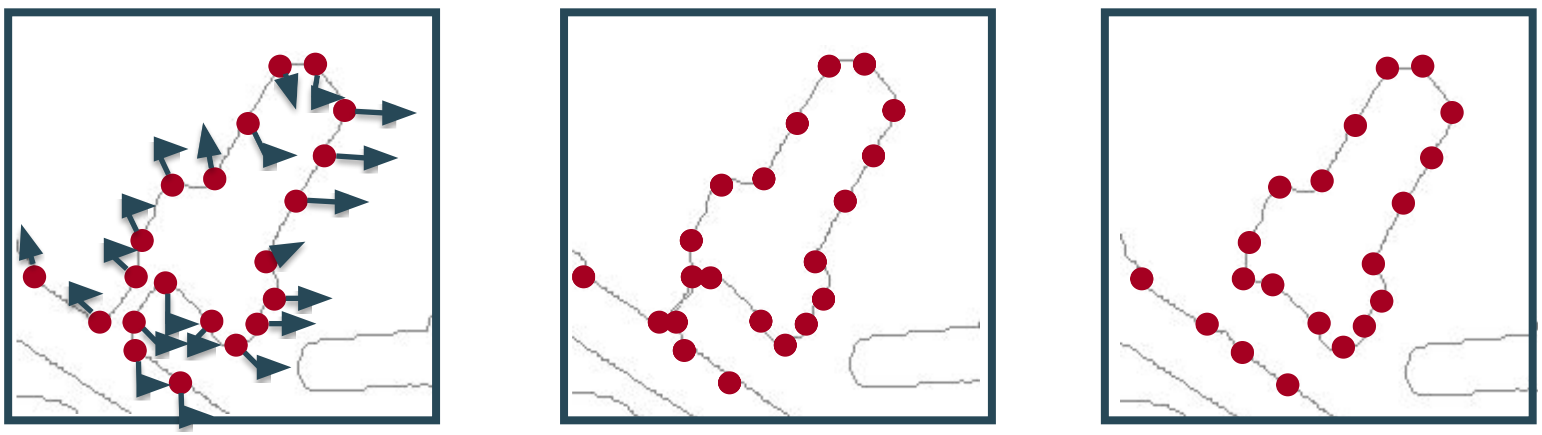
Note: Other (but similar) expressions are possible for the internal and external forces

In practice:

$$C(t)=C(t-1)+\mu(gk-\langle \nabla g,\vec{N}\rangle)\vec{N}$$

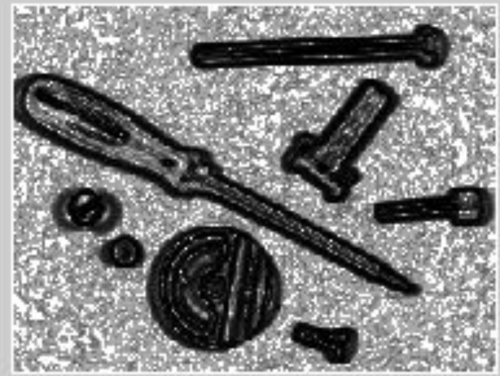
Transition based – Active contours

- Curve defined by control points:
  - Displace the control point following: $\left(gk - \left\langle \nabla g, \vec{N} \right\rangle\right)\vec{N}$
  - Resample the curve when it shrinks
  - Handle topological changes
- Other implementations are possible (for example: level set)

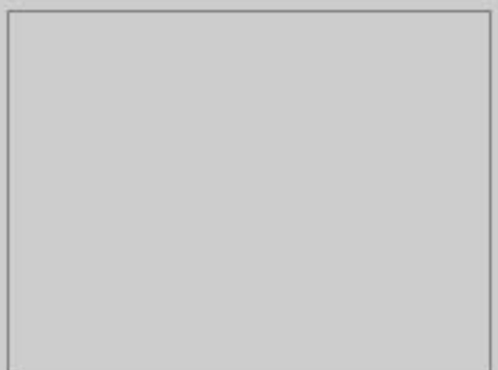


Transition based – Active contours

Original image x



$G(\nabla x)$



Initial contour

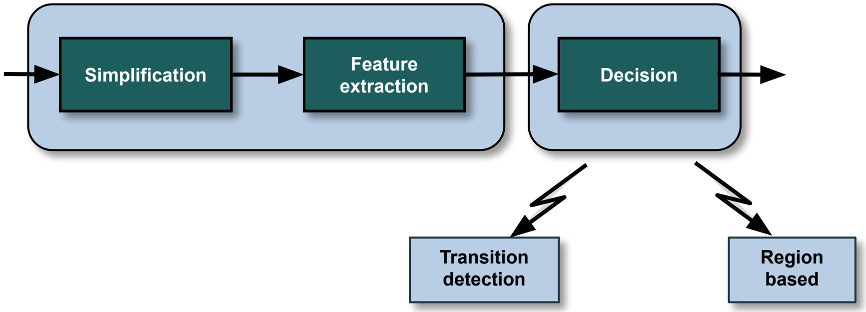
Final contour

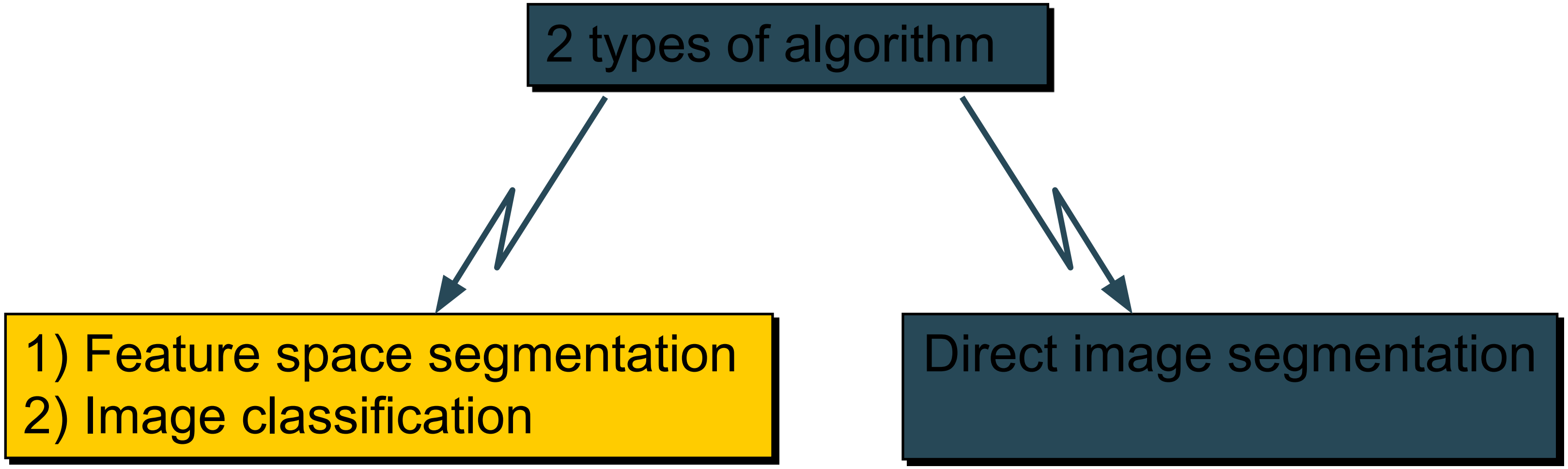
Segmentation: Transition based

- 3 main steps for segmentation:
  - Simplification, feature extraction and decision
  - Decision can be based on discontinuity (transition-based) or homogeneity (region-based)

- Transition-based segmentation:
  - Estimate transitions: gradient (may be linear or not)
  - Decision:
    - **Thresholding**: difficult because of threshold definition, noise and open contour
    - **Active contours**:
      - Curve evolution to minimize the length “weighted” by the gradient
      - Gradient descent leads to a combination of internal (smoothing) forces and external force (attraction to data)

Segmentation: Homogeneity based





Homogeneity based – Feature space segmentation

Approach:

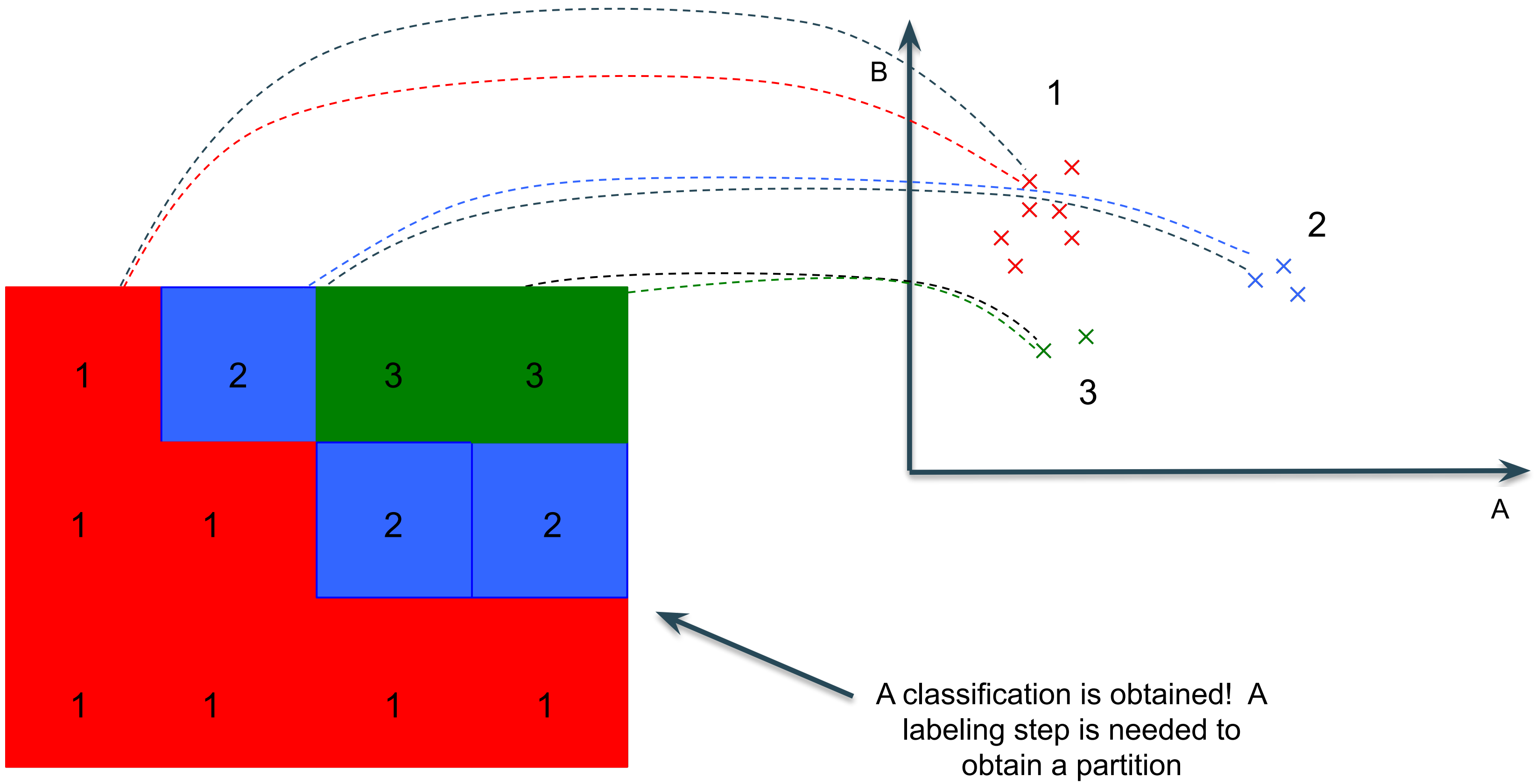
- Study the feature space
- Defines classes in the feature space
  (feature space segmentation)
- Region definition
  (image pixel classification)

Examples:

- Mono-dimensional feature space (gray level)
  - 2 classes
  - N classes
- Multidimensional feature space

Homogeneity based – Feature space segmentation

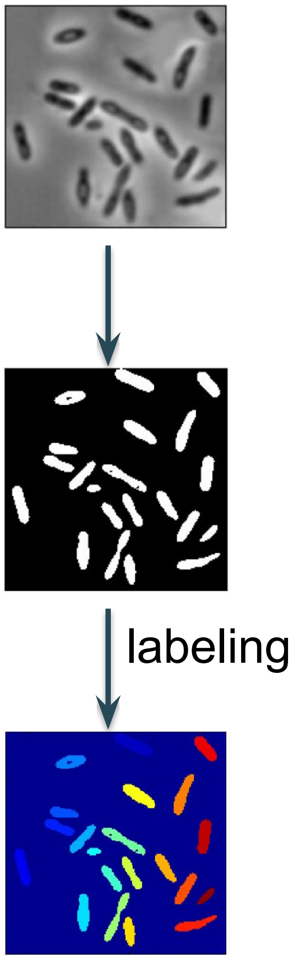
Example: bi-dimensional feature space



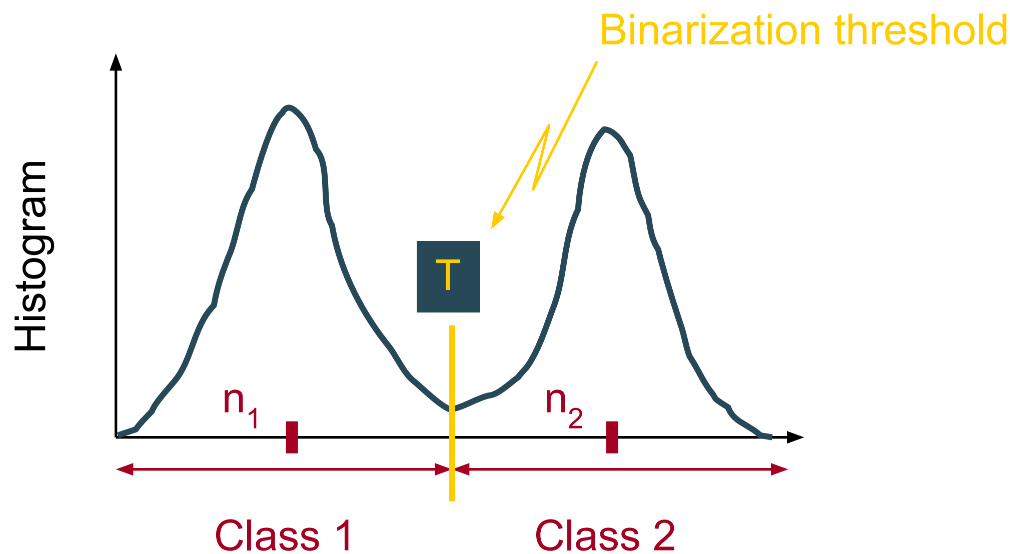
A classification is obtained! A labeling step is needed to obtain a partition

Homogeneity based – Feature space segmentation

Mono-dimensional feature space, 2 classes



- Study of the histogram (of gray level or a given feature)



- Image classification algorithm: Binarization

Homogeneity based – Feature space segmentation

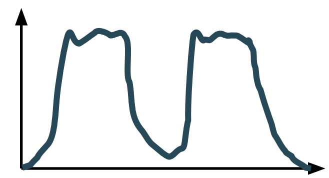
Mono-dimensional feature space, 2 classes: threshold

- If the class centers n_1 y n_2 are known (e.g.: histogram maxima)
  - Criterion: Intra class minimum variance

$$C = \underset{T}{Min}\Big[\sum_{0 \le n \le T} (n-n_1)^2 h(n) + \sum_{T < n \le M} (n-n_2)^2 h(n)\Big]$$

- If class centers are NOT known
  - ➛ Iterative estimation

$\hat{n}_1, \hat{n}_2 \rightarrow$ Fixed n_1, n_2 $\Downarrow$ Optimum T $\rightarrow$ Fixed T $\Downarrow$ Optimum n_1, n_2 $\rightarrow$ (back to Fixed n_1, n_2)



$$n_1 = \sum_{0 \le n \le T} nh(n) \Big/ \sum_{0 \le n \le T} h(n)$$

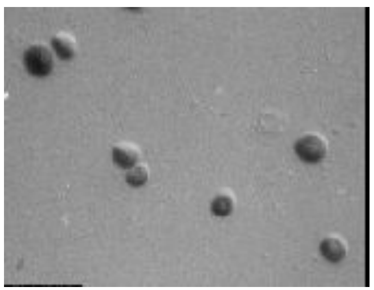
$$n_2 = \sum_{T < n \le M} nh(n) \Big/ \sum_{T < n \le M} h(n)$$

Mean of each class

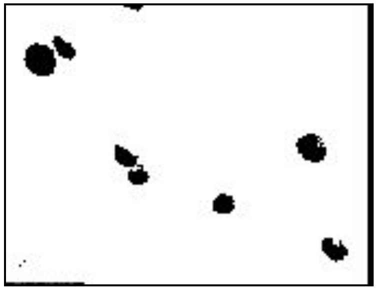
  - ➛ Local minimum

Homogeneity based – Feature space segmentation

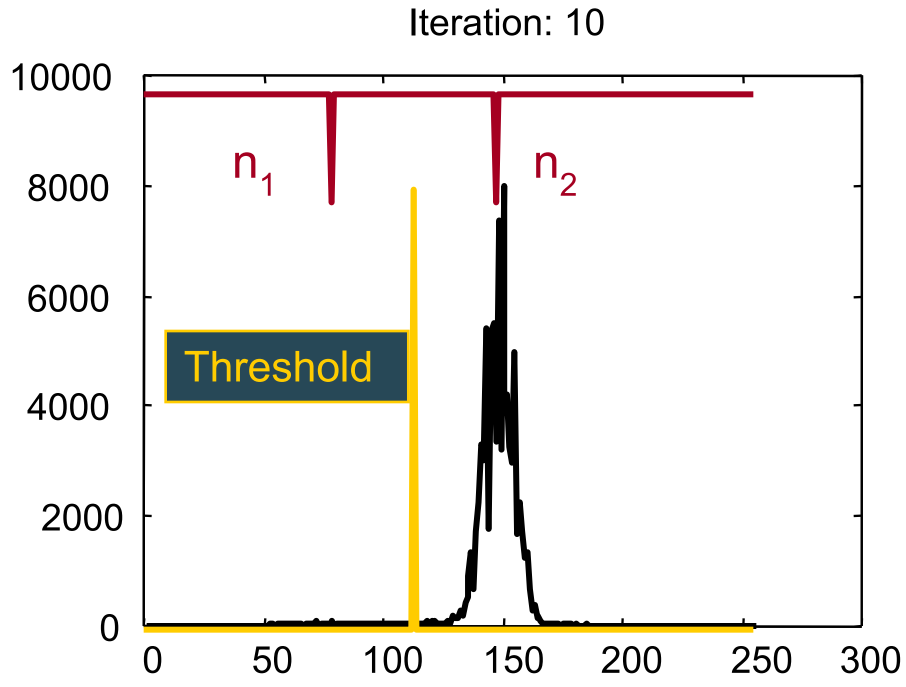
Mono-dimensional feature space, 2 classes: threshold



Original



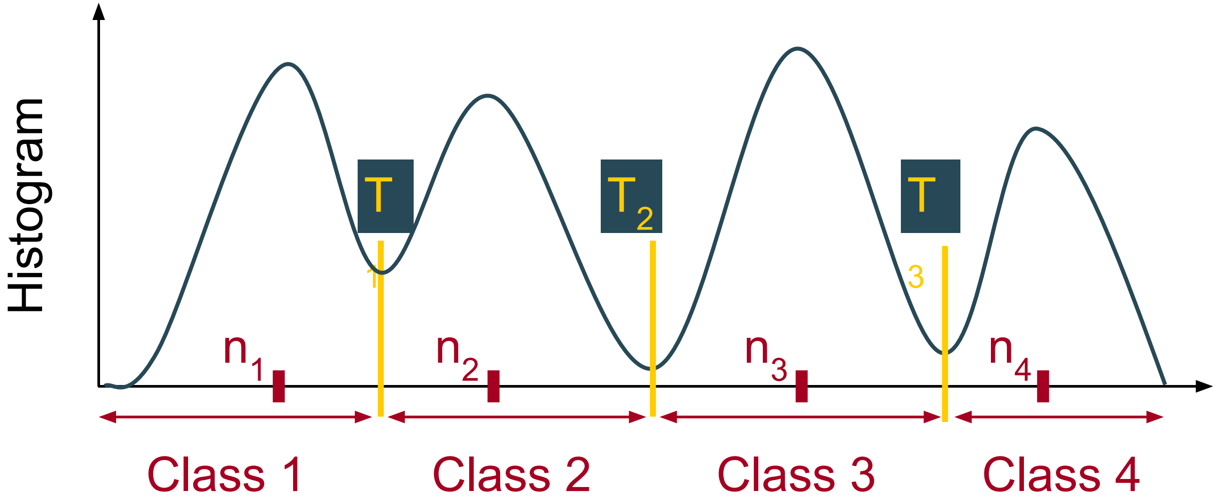
Binarized image



Homogeneity based – Feature space segmentation – K-Means

Mono-dimensional feature space, K classes

- Histogram analysis



- Classification algorithm: “K-means”

Homogeneity based – Feature space segmentation – K-Means

K-Means algorithm

$\hat{n}_1, \cdots, \hat{n}_K$ → Fixed $n_1,\ldots,n_k$ ⇓ Optimum $T_1,\ldots, T_{k-1}$ → Fixed $T_1,\ldots, T_{k-1}$ ⇓ Optimum $n_1,\ldots,n_k$ → (back to Fixed $n_1,\ldots,n_k$)

- All T combinations (may be complex!)
- Euclidean distance

$$T_1 = (n_1 + n_2)/2$$
$$\cdots$$
$$T_{k-1} = (n_{k-2} + n_{k-1})/2$$

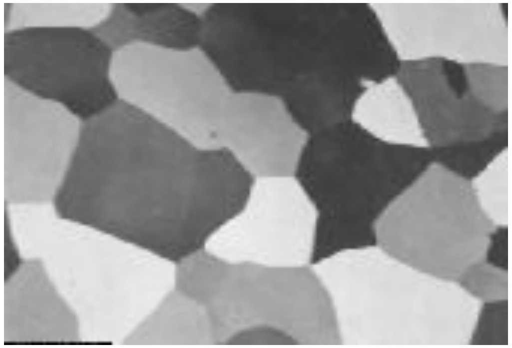
Individual class mean

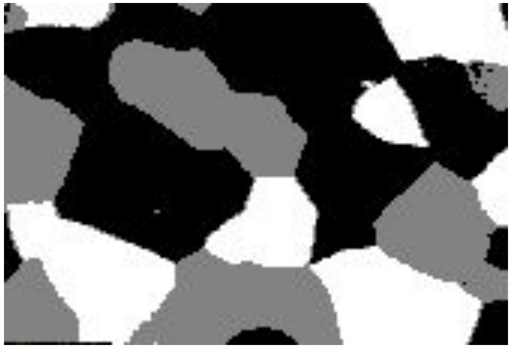
$$n_1 = \sum_{0 \le n \le T_1} nh(n) / \sum_{0 \le n \le T_1} h(n)$$
$$\cdots$$
$$n_K = \sum_{T_{k-1} < n \le M} nh(n) / \sum_{T_{k-1} < n \le M} h(n)$$

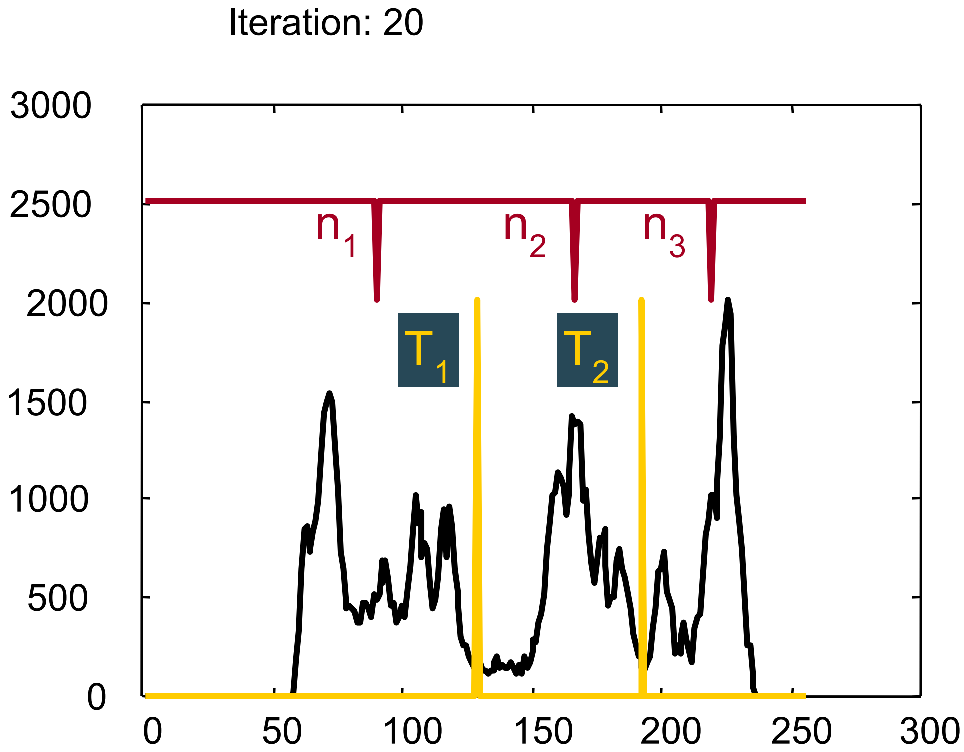
➛ Local minimum

Homogeneity based – Feature space segmentation – K-Means

K-Means example : 3 classes



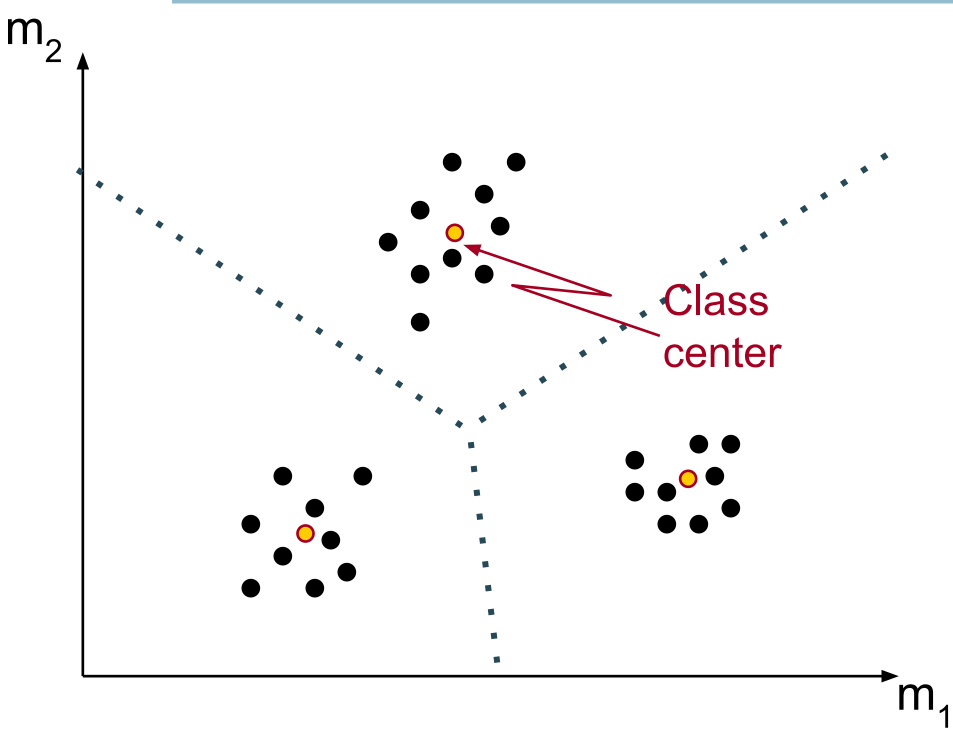




Homogeneity based – Feature space segmentation – K-Means

Multi-dimensional feature space, K classes

Dimension of the feature space: 2
Number of classes: 3



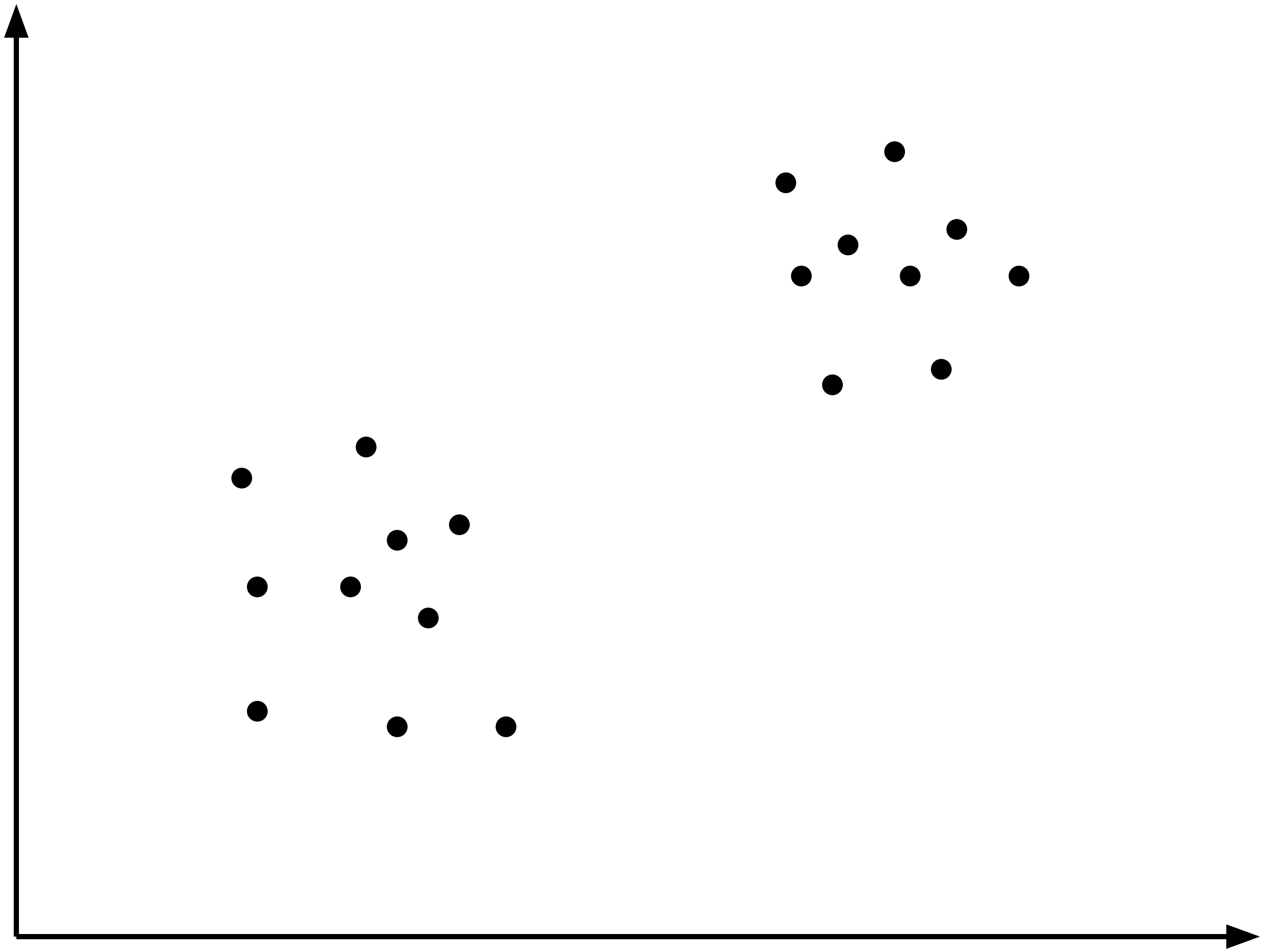
1. Initialize K classes. Compute the centers of each class
2. For each point:
   a. Compute the distances between the point and the class centers
   b. Assign the point to the closest class
3. Update the class centers
4. Repeat 2 & 3 until no change (in assignments or center values) is observed.

Max Lloyd algorithm

Complexity: O(n·k·d·i)
d: dimension, n: number of points, i: iterations

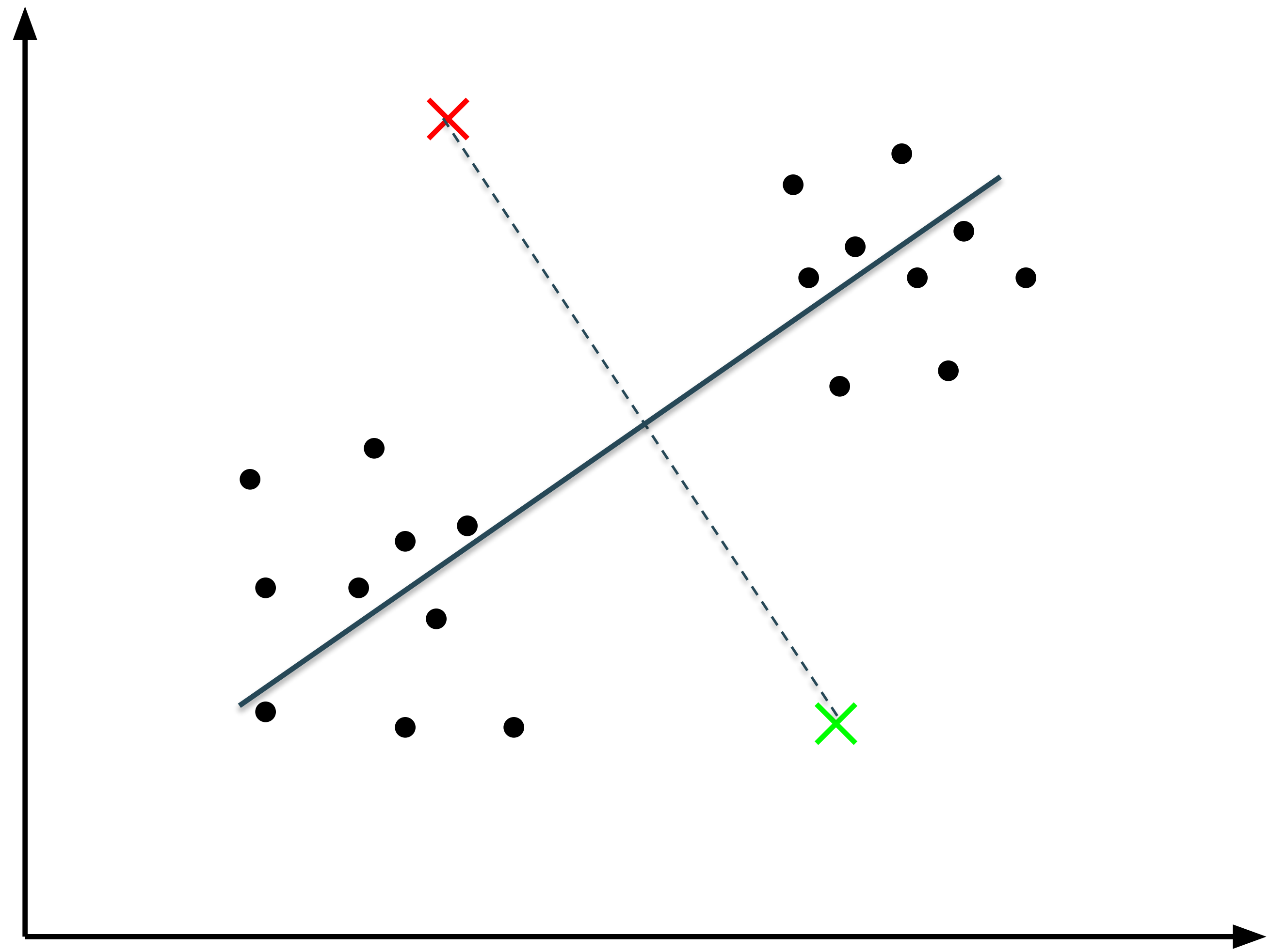
Homogeneity based – Feature space segmentation – K-Means

Example:



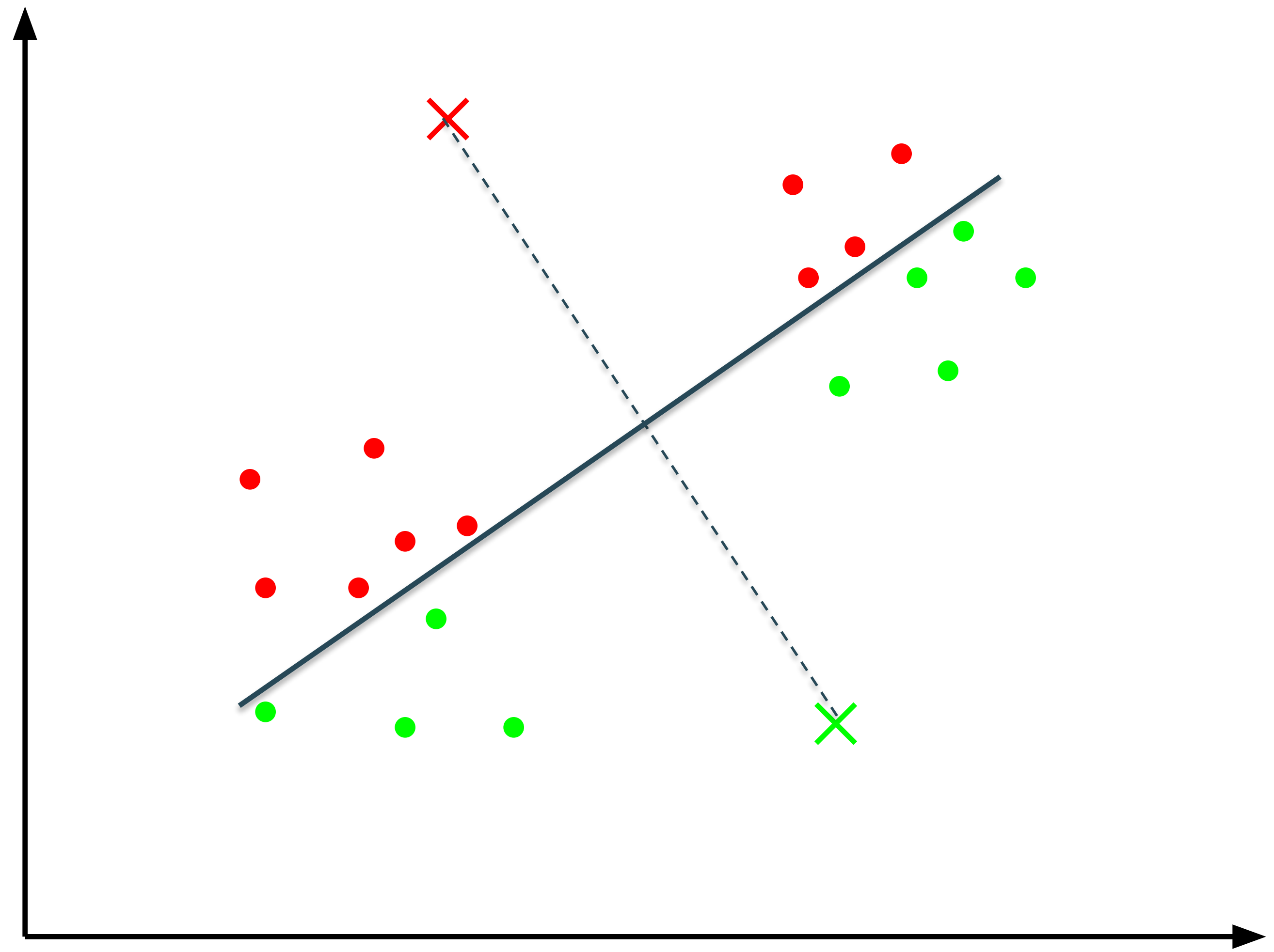
Homogeneity based – Feature space segmentation – K-Means

Example:



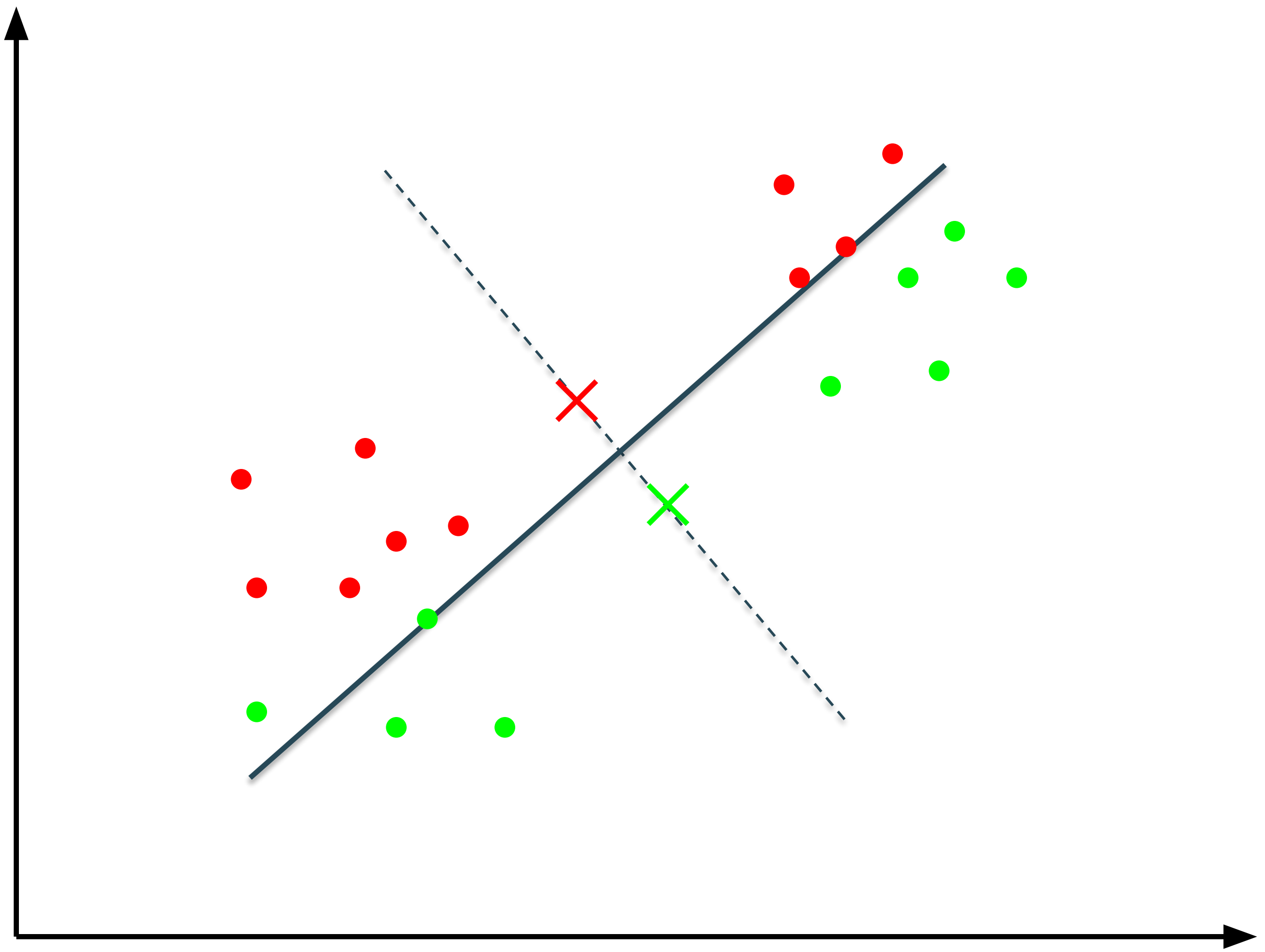
Homogeneity based – Feature space segmentation – K-Means

Example:



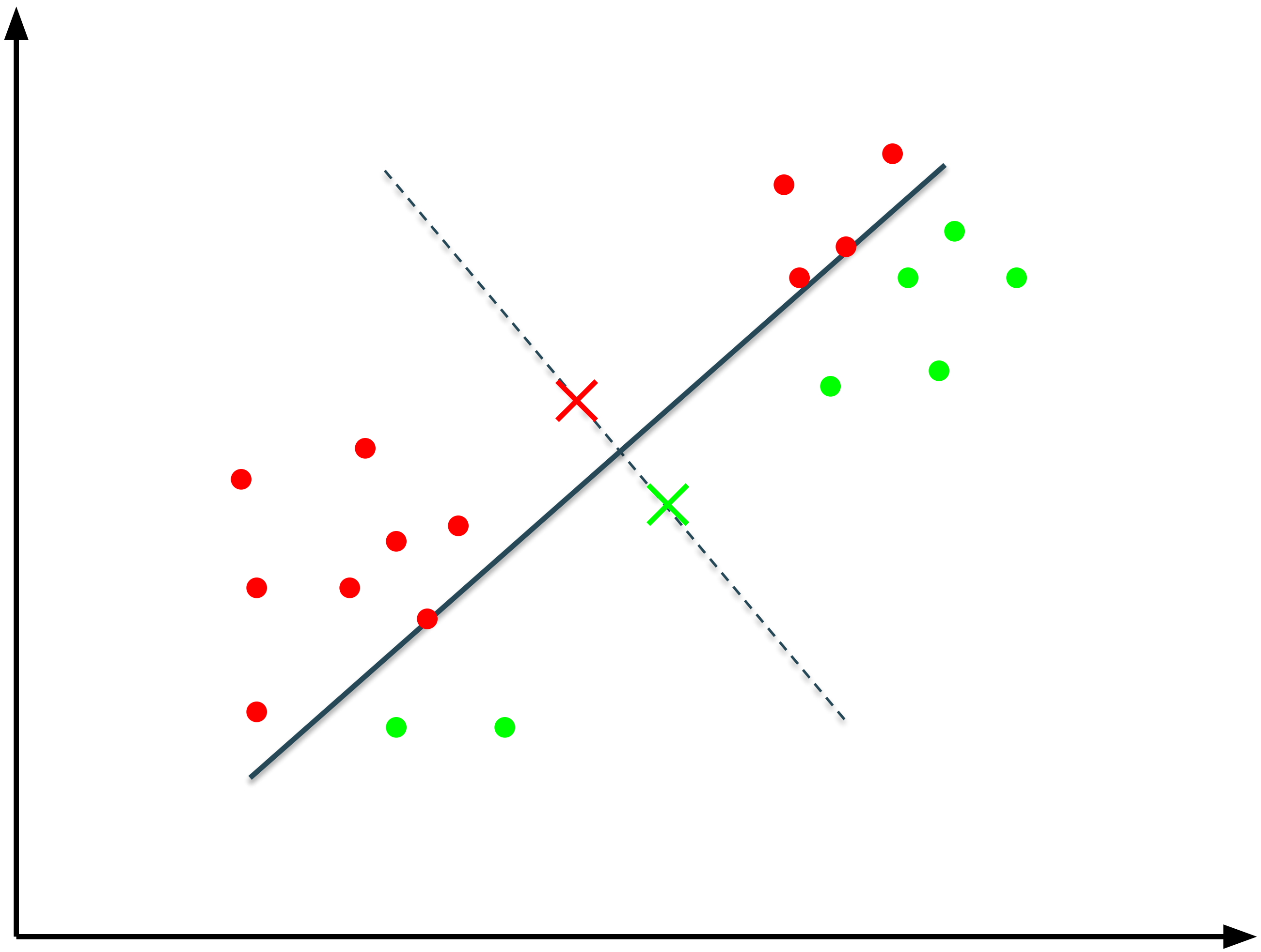
Homogeneity based – Feature space segmentation – K-Means

Example:



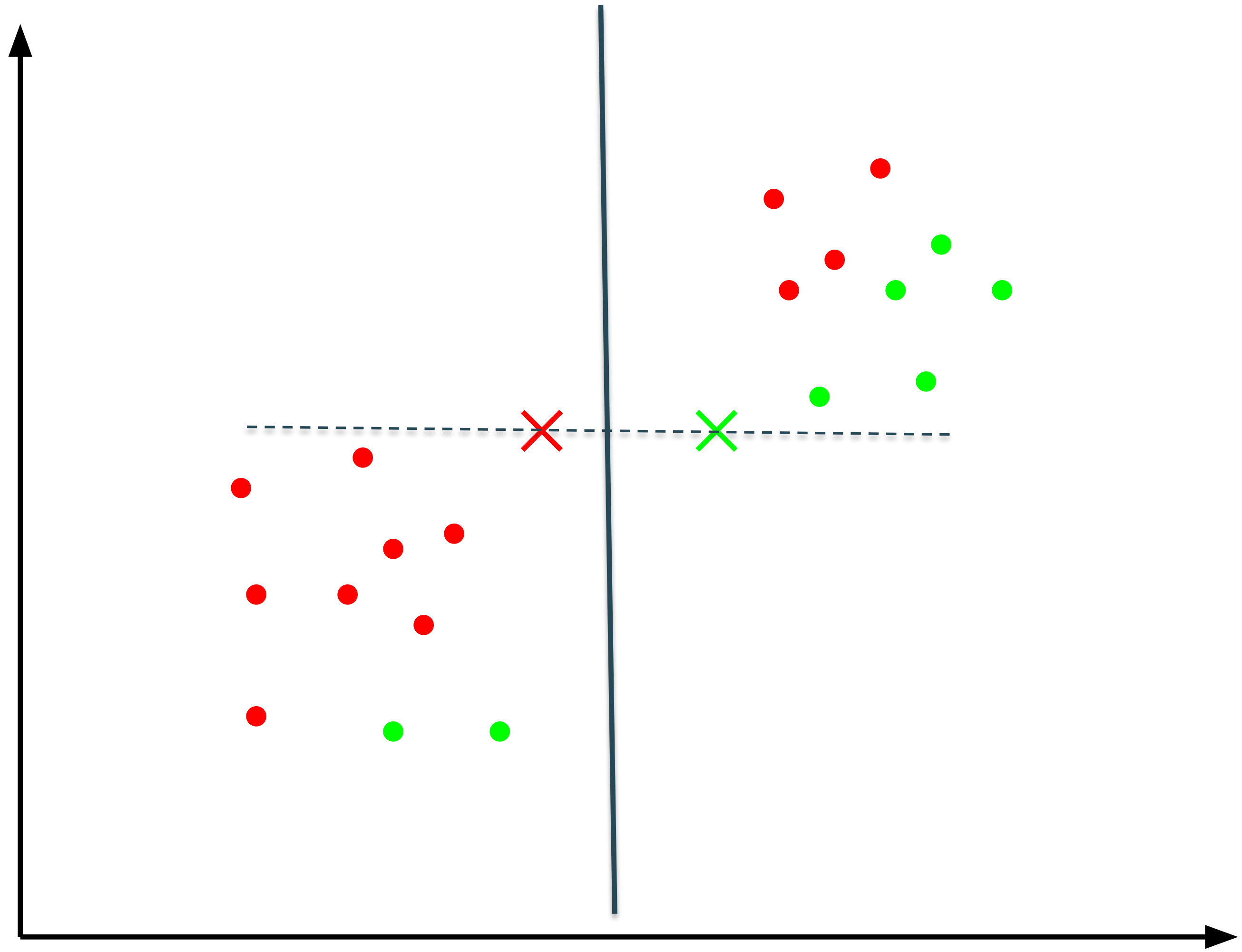
Homogeneity based – Feature space segmentation – K-Means

Example:



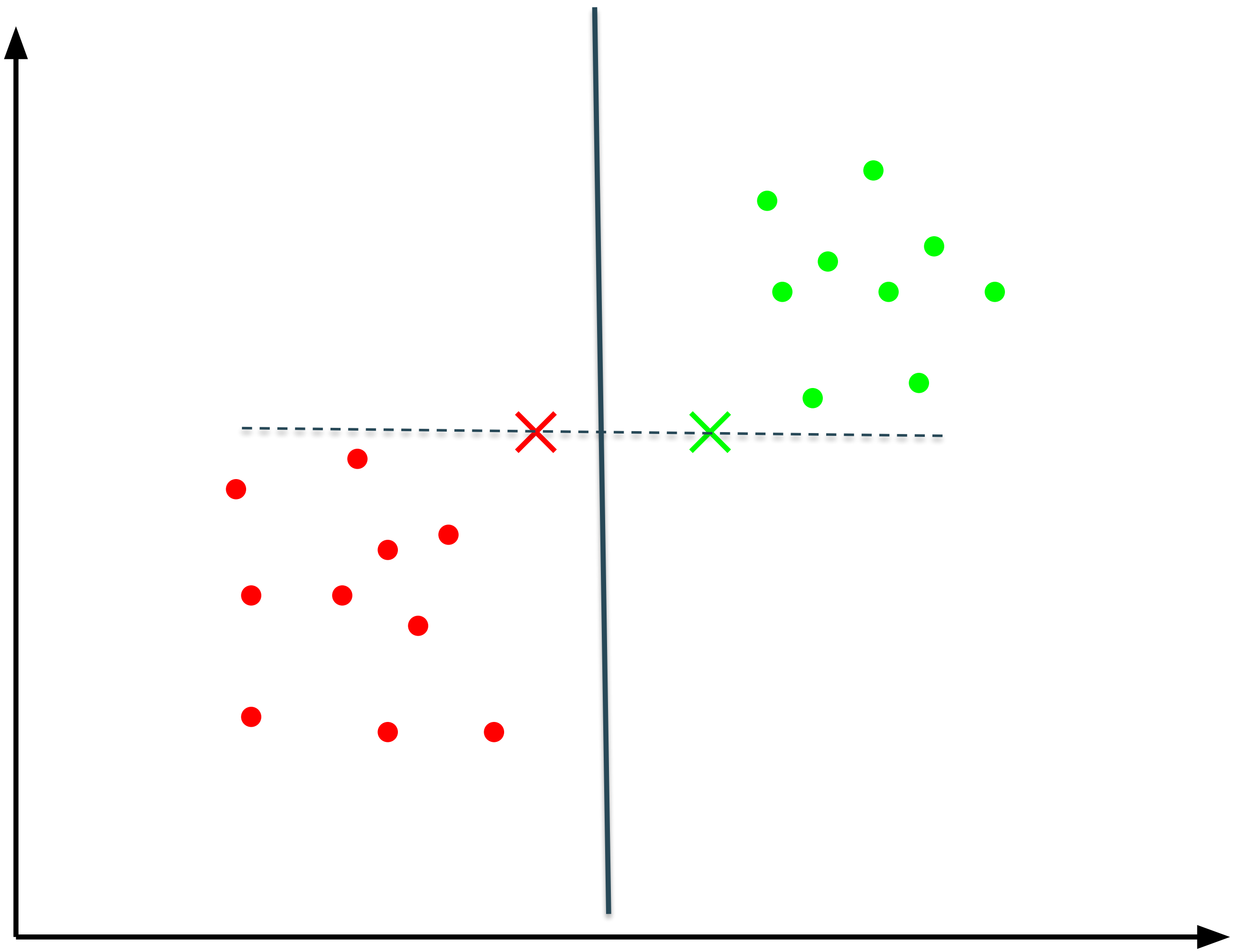
Homogeneity based – Feature space segmentation – K-Means

Example:



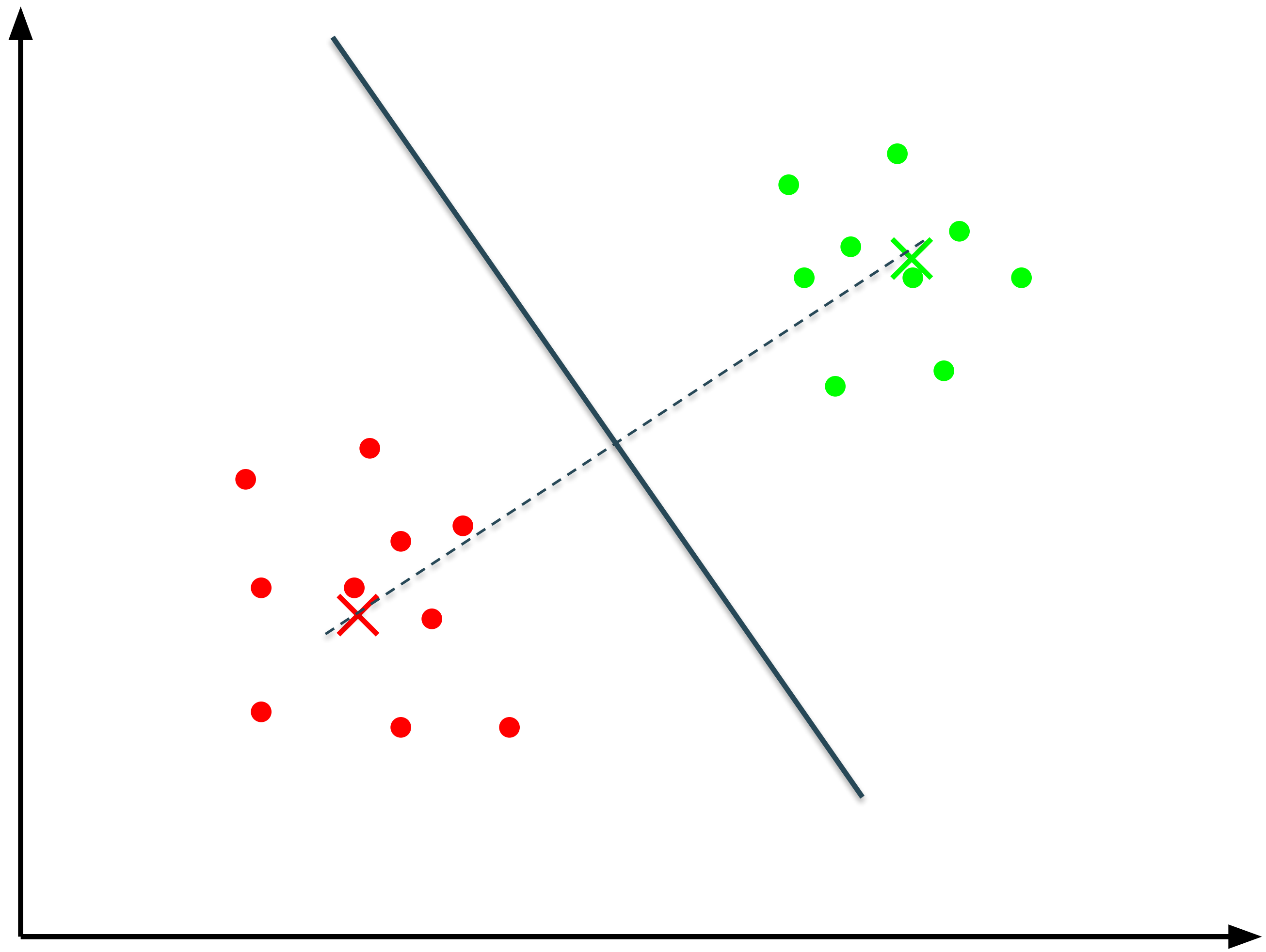
Homogeneity based – Feature space segmentation – K-Means

Example:



Homogeneity based – Feature space segmentation – K-Means

Example:



Homogeneity based – Feature space segmentation – K-Means

- Convergence is guaranteed … but may be to a local minimum → application optimized strategies can be added
- The straightforward implementation of such method may be too slow → there are proposed optimizations, especially for the classification step.
- Other distances are also possible (K-Medoids).
- Hard assignment of data points to clusters: small shift of a data point can flip it to a different cluster → can be solved by a soft assignment based on a probabilistic approach.

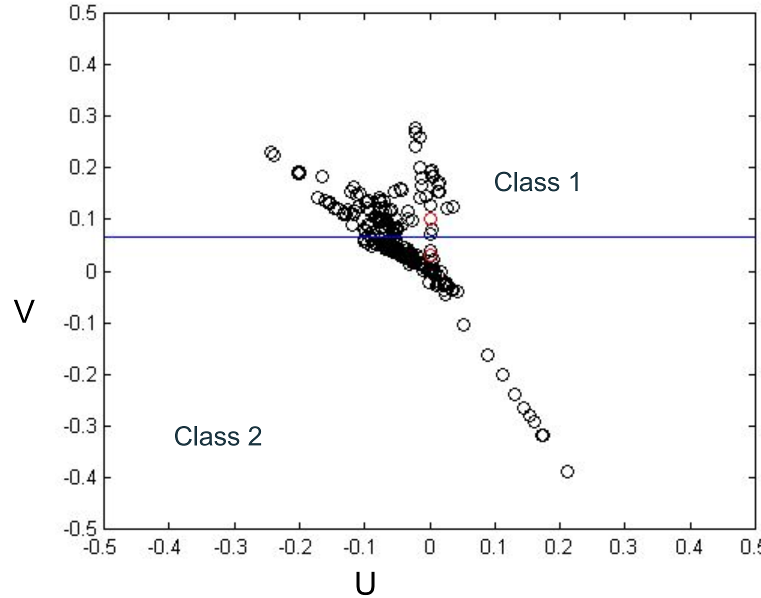
Gaussian Mixture Models

Slide credit: Dr. Antonio M. López (UAB)

Homogeneity based – Feature space segmentation – K-Means

Example: Face (skin) detection

Original image



U=B-Y component

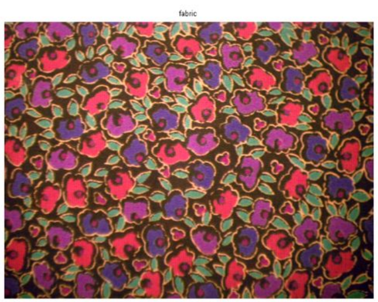
V=R-Y Component

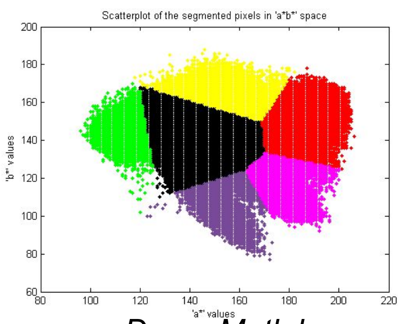
Binarization

Homogeneity based – Feature space segmentation – K-Means

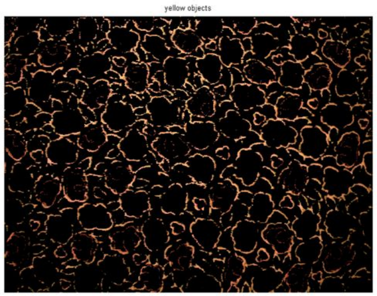
Example: Color segmentation

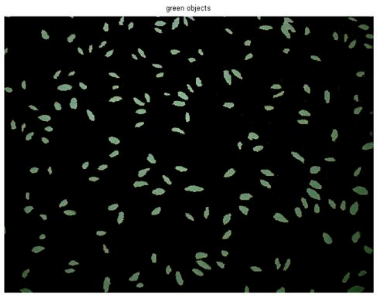
Goal: Segment the regions corresponding to the N dominant color of the image
Color space: CIE Lab

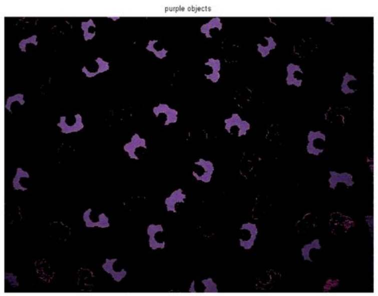




Demo Matlab (ipexfabric)







Homogeneity based – Feature space segmentation – K-Means

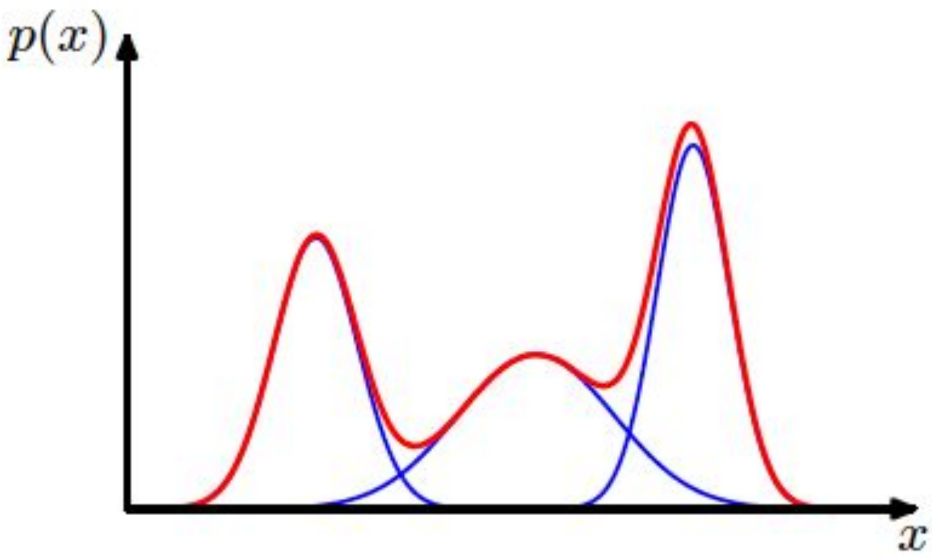
Example: Chroma Key

- Used in TV, cine or photography
- Consist in changing the scene background
- Background is generally blue or green. It is segmented and the visible background pixels are substituted by a different background.
- Possible issues: incoherent background/scene illumination or shadows

Homogeneity based – FS segmentation : Mixture Models

- In k-means, pixels are clustered using hard assignments
  - Each pixel goes to closest cluster center
  - A probabilistic approach, where each pixel helps estimating more than one cluster, may be more robust

→Probabilistic Mixture Model



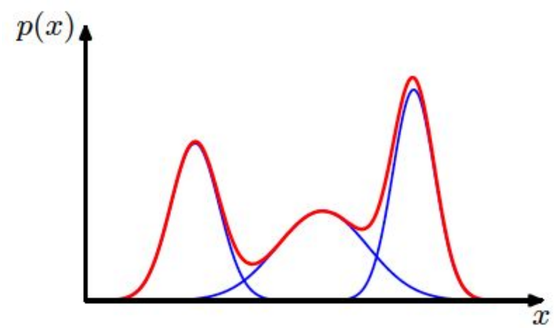
Homogeneity based – FS segmentation : Mixture Models

- Distributions can be approximated as a linear combination of basic densities

$$p(x) = \sum_{k=1}^{K} \pi_k p(x \mid k) = \sum_{k=1}^{K} P(k) p(x \mid k)$$

Mixing parameters

Densities



Where:

- **Mixing parameters**: can be seen as prior probabilities or normalized positive weights.
- **Densities** $p(x \mid k)$ are class-conditional densities, with *k* the discrete random variable of the class

Slide credit: Dr. Antonio M. López

Homogeneity based – FS segmentation : Mixture Models

- The $p(x \mid k)$ are, in fact, parametric densities: $p(x \mid \theta_k)$
- A widely used approach is to assume that all the $p(x \mid k)$ from the same family of densities
- In the limit, even smooth densities can be approximated
- Multivariate gaussians are a very popular choice

- Applications:
  - Probability density estimation
  - **Clustering: every component is a cluster center**

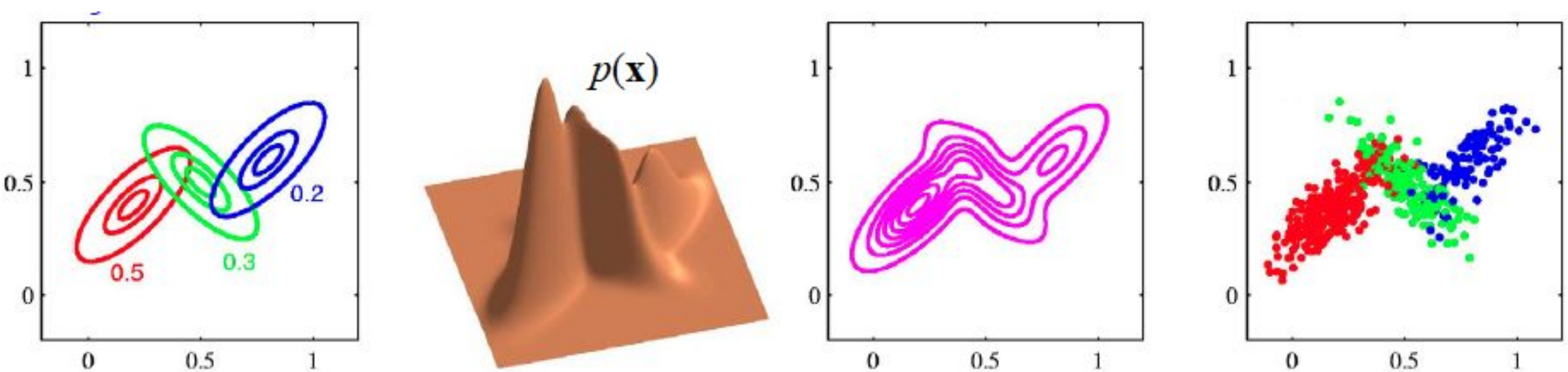
Homogeneity based – FS segmentation : GMM

Good choice for the components → Gaussians

$$\boxed{p(x\,|\,\theta)=\sum_{k=1}^{K}\pi_k p(x\,|\,\theta_k)} \qquad 0\le\pi_k\le 1,\quad \sum_{k=1}^{K}\pi_k=1$$

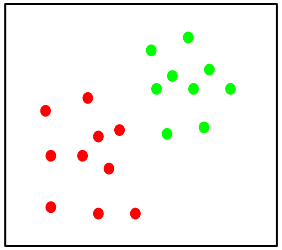
$$p(x\,|\,\theta_k)=N(x\,|\,\mu_k,\Sigma_k)\propto\exp\left(\frac{-(x-\mu_k)\Sigma_j^{-1}(x-\mu_k)}{2}\right)$$

Gaussians can be seen as oriented “blobs” in feature space



Homogeneity based – FS segmentation : GMM

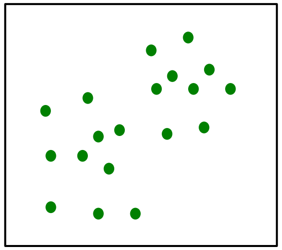
- Mixture parameter estimation:
  - Labeled data: we know which component generates each Gaussian → Easy!!



$$\pi_k = \frac{n_k}{N}, \quad \mu_k = \frac{1}{n_k}\sum_{i|l_i=k} x_i, \quad \Sigma_k = \frac{1}{n_k}\sum_{i|l_i=k}(x_i-\mu_k)(x_i-\mu_k)^T$$

Maximum Likelihood (ML) estimates

  - Unlabeled data: the assignments are not known
    - Guessed based on the current mixture distribution estimate
    - Soft assignments (posterior probabilities)



Homogeneity based – FS segmentation : GMM

- Maximum Likelihood Estimation

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \{L(\theta \mid x)\} \quad where\ L(\theta \mid x) = P(x \mid \theta)$$

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \{\log(L(\theta \mid x))\}$$

Analytically easier

Data set: $\{x_i\}$

$$p(x \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta)$$

<u>Incomplete data Log-Likelihood</u>

$$\sum_i \log p(x_i \mid \theta) = \sum_i \log\left(\sum_k \pi_k \underbrace{p(x_i \mid \mu_k, \Sigma_k)}_{N(x \mid \mu_k, \Sigma_k)}\right)$$

- We do not know which component generates each gaussian → incomplete data or hidden variables

Homogeneity based – FS segmentation : GMM

- To maximize $\log(L(\theta \mid x))$, we can derive with respect to the parameters and equal to zero. We obtain:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k(x_i) \cdot x_i$$

$$N_k = \sum_{i=1}^{N} \gamma_k(x_i) \cdot x$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k(x_i) \cdot (x_i - \mu_k) \cdot (x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$\gamma_k(x_i) = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

Posterior probability of *k* given $\boldsymbol{x}_i$, or in other words, responsibility that the Gaussian *k* takes for explaining the observation $\boldsymbol{x}_i$.

Slide credit: Dr. Antonio M. López

Homogeneity based – FS segmentation : GMM

- $\gamma_k(x_i)$ represents the posterior prob. of *k* given x_i
  $\rightarrow$ responsibility that gaussian *k* takes for explaining observation x_i

$$p(x \mid k) \sim N(x \mid \mu_k, \Sigma_k)$$

$$p(k \mid x) = \frac{P(k)p(x \mid k)}{p(x)} = \frac{\pi_k N(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x \mid \mu_j, \Sigma_j)} \Rightarrow \gamma_k(x_i) = p(k \mid x_i)$$

$$p(x) = \sum_{k=1}^{K} \pi_k p(x \mid \theta_k)$$

Slide credit: Dr. Antonio M. López

Homogeneity based – FS segmentation : GMM

Interpretation

$$\gamma_k(x_i) = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k(x_i) \cdot x_i$$

Weighted mean of all the points in the data set

$$N_k = \sum_{i=1}^{N} \gamma_k(x_i)$$

Effective number of points assigned to cluster **k**

Covariance as weighted variances of the data wrt to each component estimated mean

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k(x_i) \cdot (x_i - \mu_k) \cdot (x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Average responsibility that each component takes for explaining the data

Notice: if **K=1** then $\gamma_1(x_n) = 1$. This corresponds to the ML estimation of the parameters of a single Gaussian kernel.

Slide credit: Dr. Antonio M. López

Homogeneity based – FS segmentation : GMM

- If K > 1, there is no closed form solution

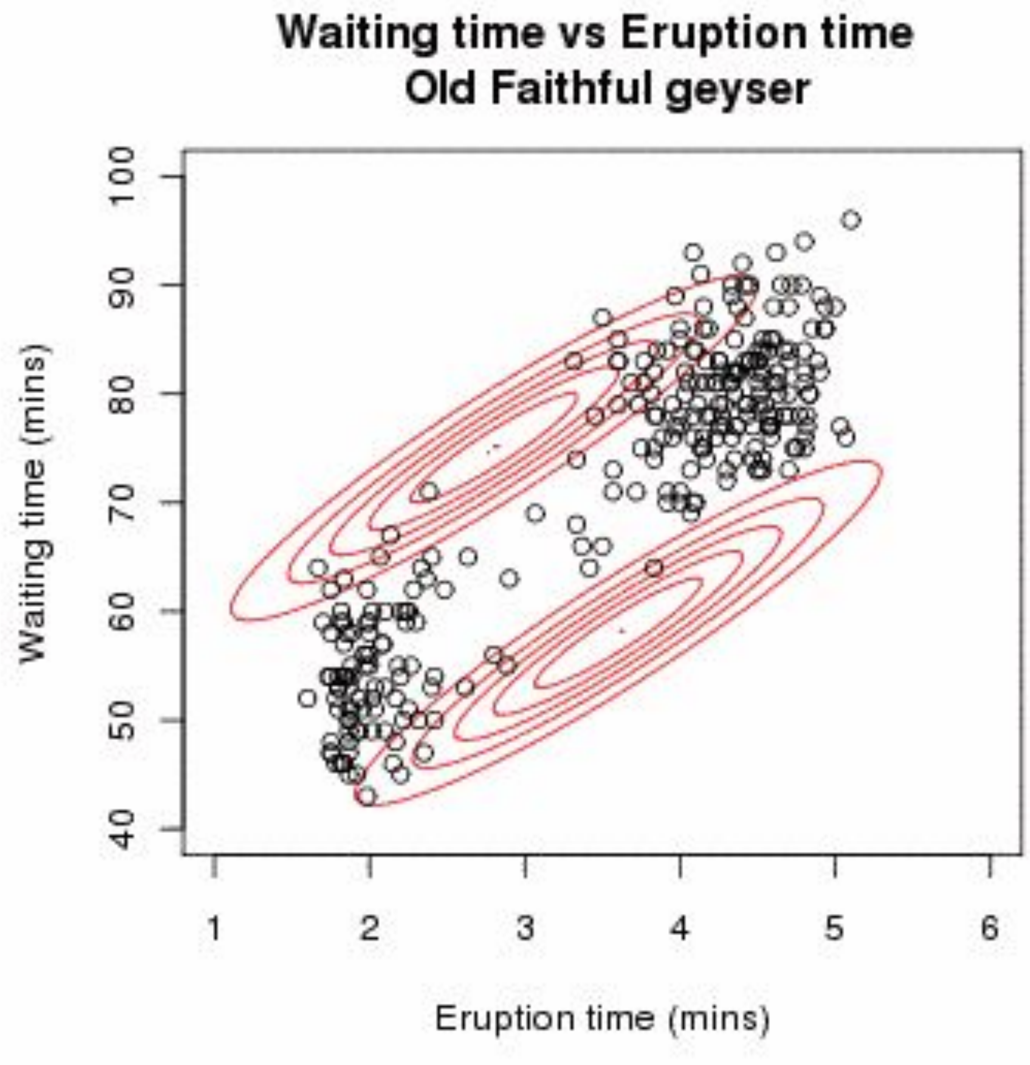
$$\{\pi_K, \mu_k, \Sigma_k\} \text{ depend on } \{\gamma_k\}$$
$$\{\gamma_k\} \text{ depend on } \{\pi_k, \mu_k, \Sigma_k\}$$

- Recursive procedure:
  1. Start: Given K, provide the initial parameters: $\{\pi_k, \mu_k, \Sigma_k\}, \quad k = 1 \ldots K$
  2. Classify: given $\{\pi_k, \mu_k, \Sigma_k\}$ compute the $\{\gamma_k\}$
  3. Re-center: Given $\{\gamma_k\}$, compute the $\{\pi_k, \mu_k, \Sigma_k\}$
  4. Repeat 2-3 until convergence

→ This can be seen as Expectation Maximization (EM) framework!
Classification: **E**xpectation step
Re-centering: **M**aximization step (ML estimation)

Slide credit: Dr. Antonio M. López

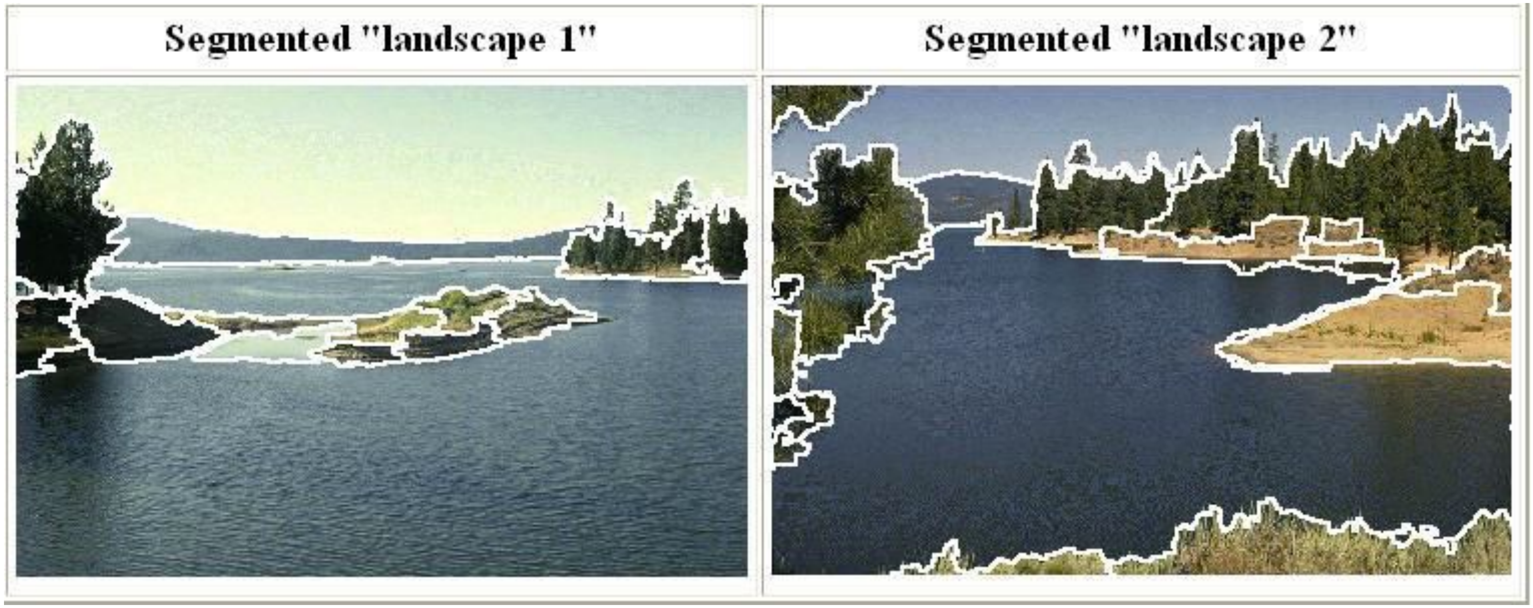
Homogeneity based – FS segmentation : GMM



http://en.wikipedia.org/wiki/Expectation_maximization

Homogeneity based – FS segmentation : Mean shift

- Technique for clustering-based segmentation
- It seeks nodes (local maxima of density) in feature space

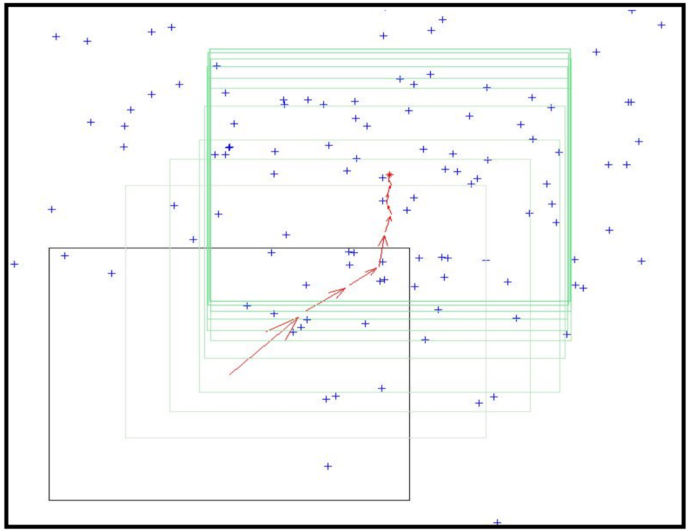


D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Homogeneity based – FS segmentation : Mean shift

Mean-shift

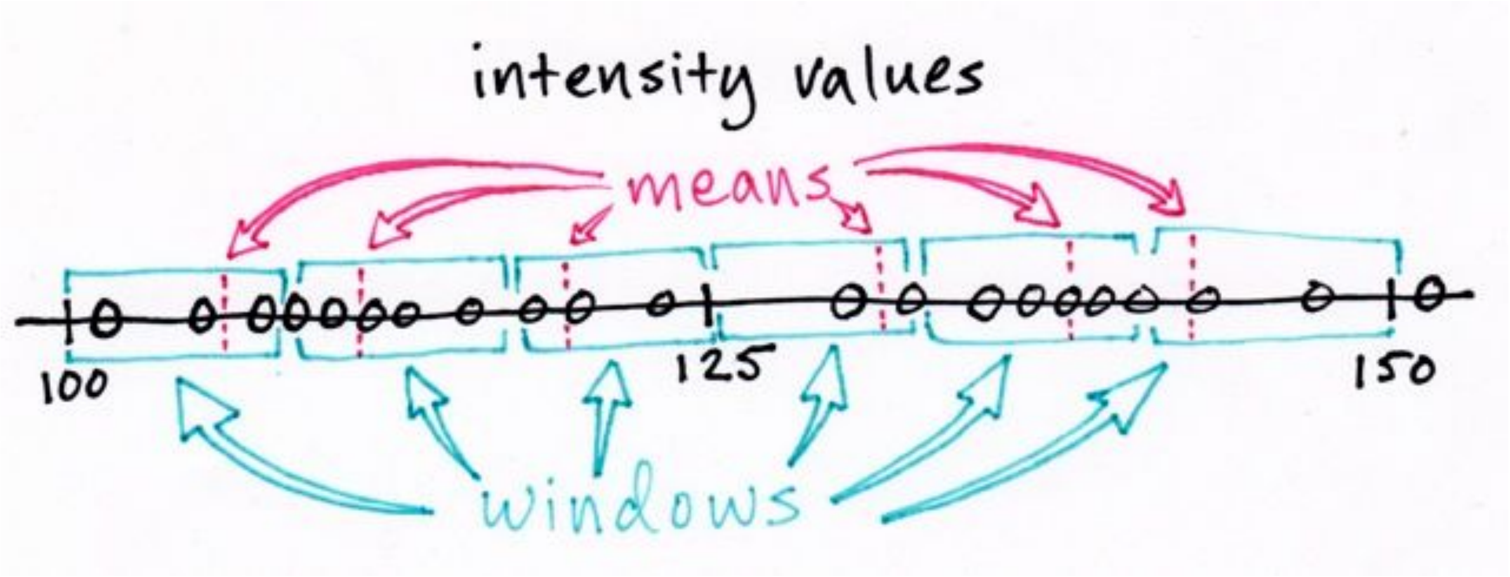
1. Select a search window size
2. Define a set of initial points for the search windows
3. Compute the mean location (centroid) of the data inside the window
4. Re-center the window at position computed in 3.
5. Iterate 3. and 4. until convergence



Slide credit: Gary Bradski & Sebastian Thrun

Homogeneity based – FS segmentation : Mean shift

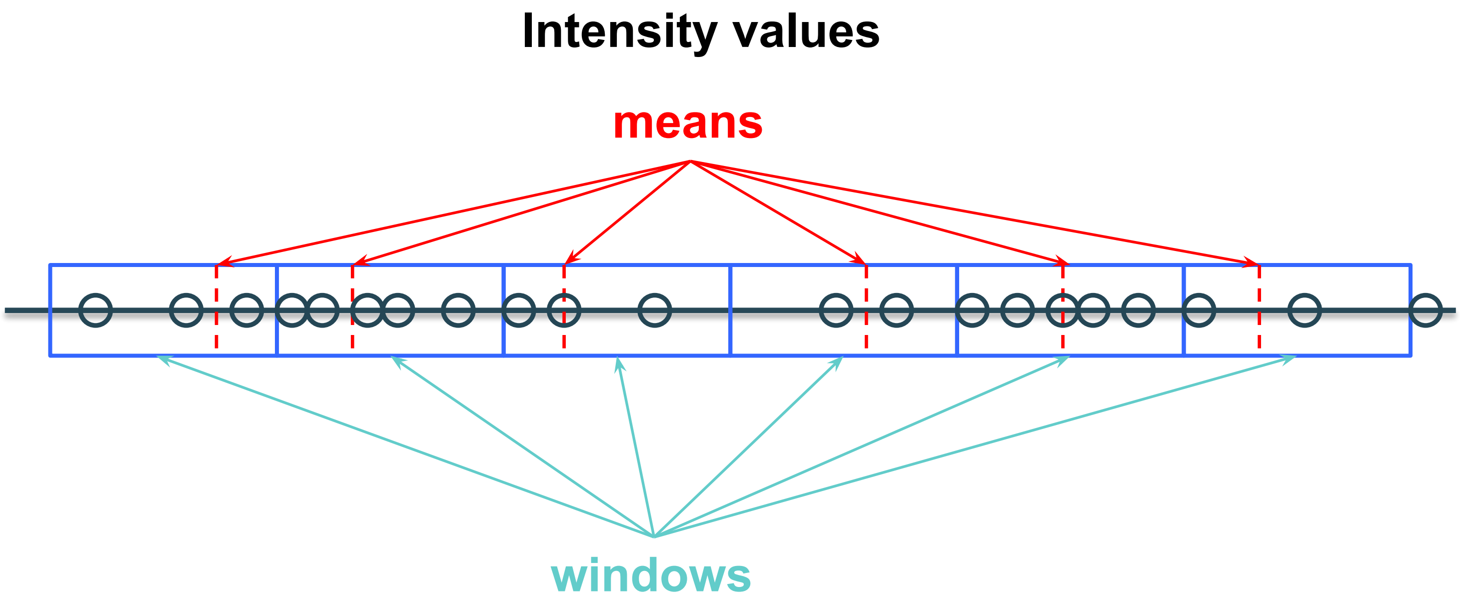
1D example



Source: Mike Laielli – benderseye.com

Homogeneity based – FS segmentation : Mean shift

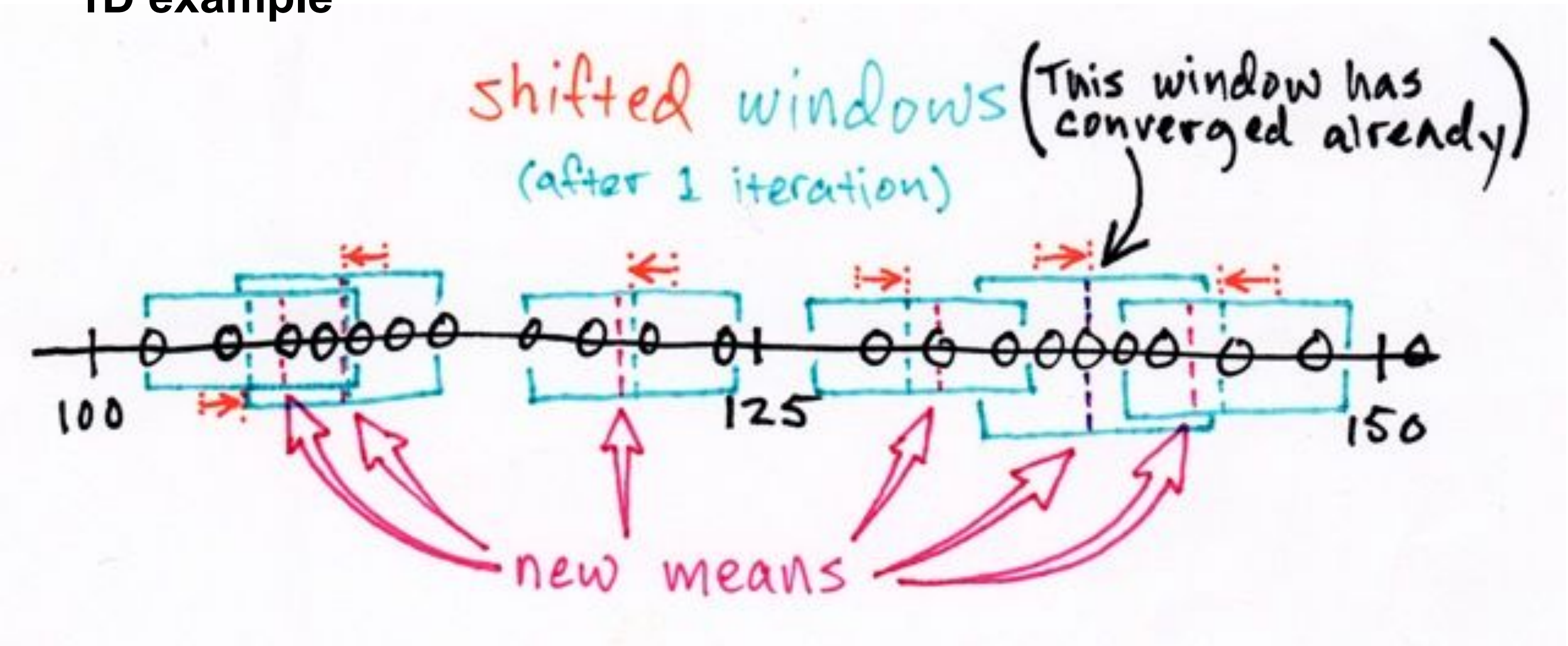
1D example



Source: Mike Laielli – benderseye.com

Homogeneity based – FS segmentation : Mean shift

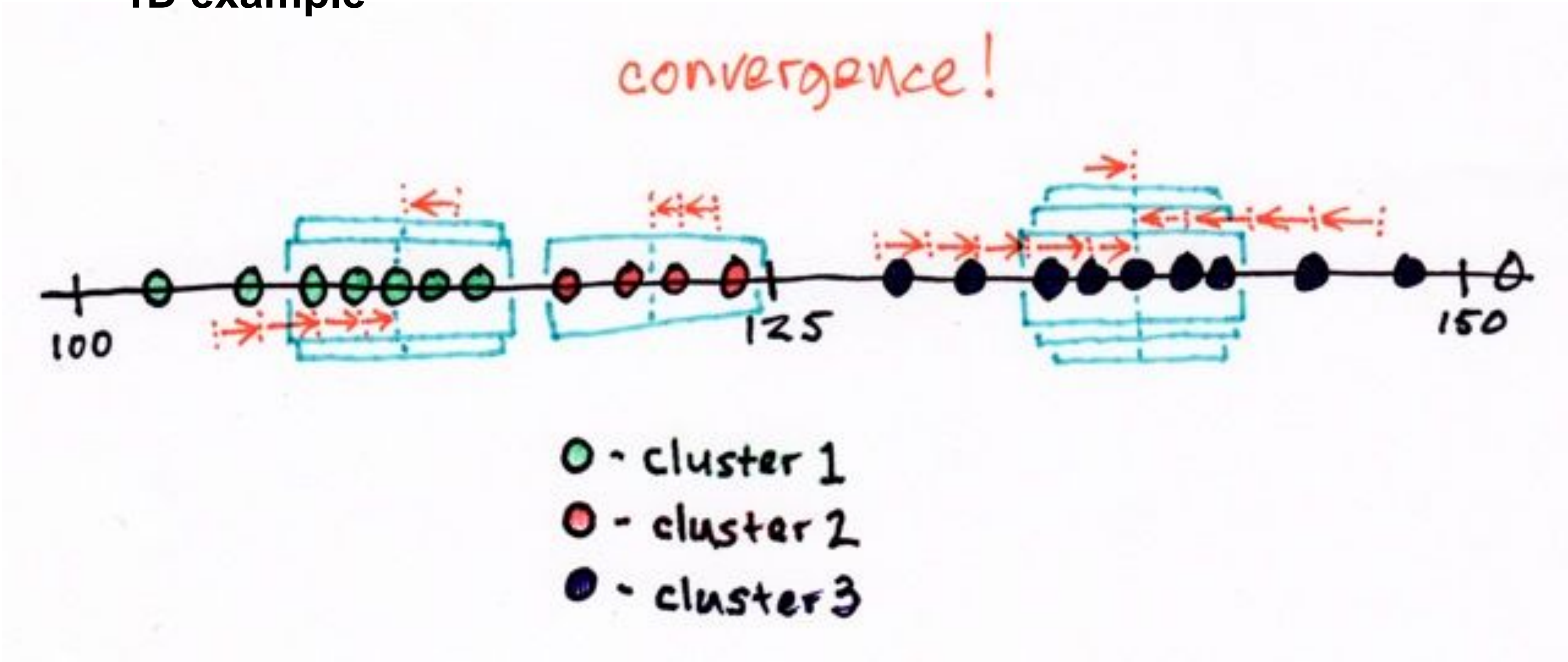
1D example



Source: Mike Laielli – benderseye.com

Homogeneity based – FS segmentation : Mean shift

1D example

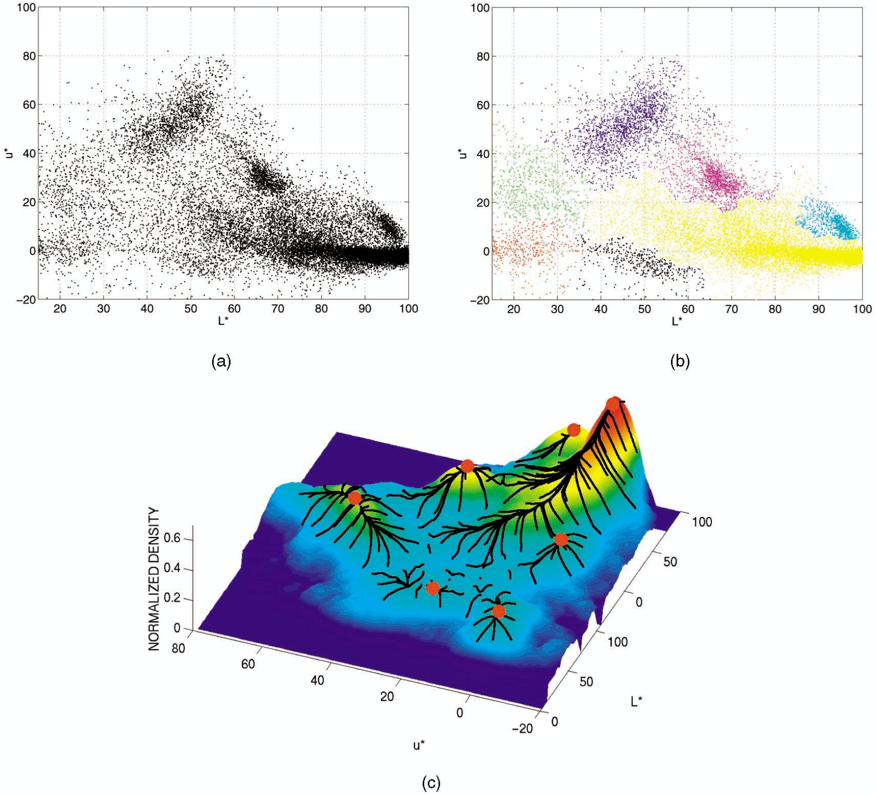


Source: Mike Laielli – benderseye.com

Homogeneity based – FS segmentation : Mean shift

Procedure:

- Find features: (color, gradient, etc.)
- Select window size. Initialize windows at individual feature points
- Perform mean-shift for each window until convergence
- Merge windows that end near the same mode.



(a)

(b)

(c)

Homogeneity based – FS segmentation : Mean shift

Pros

- Does not assume spherical clusters
- Just a single parameter (window size)
- Finds variable number of modes
- Robust to outliers

Cons

- Output depends on window size
- Computationally expensive
- Does not scale well with dimension of feature space (curse of dimensionality)

$O(Tn^2)$

T: Number of iterations
N : number of points in the dataset

Segmentation - FS segmentation : Conclusions

Strong point:

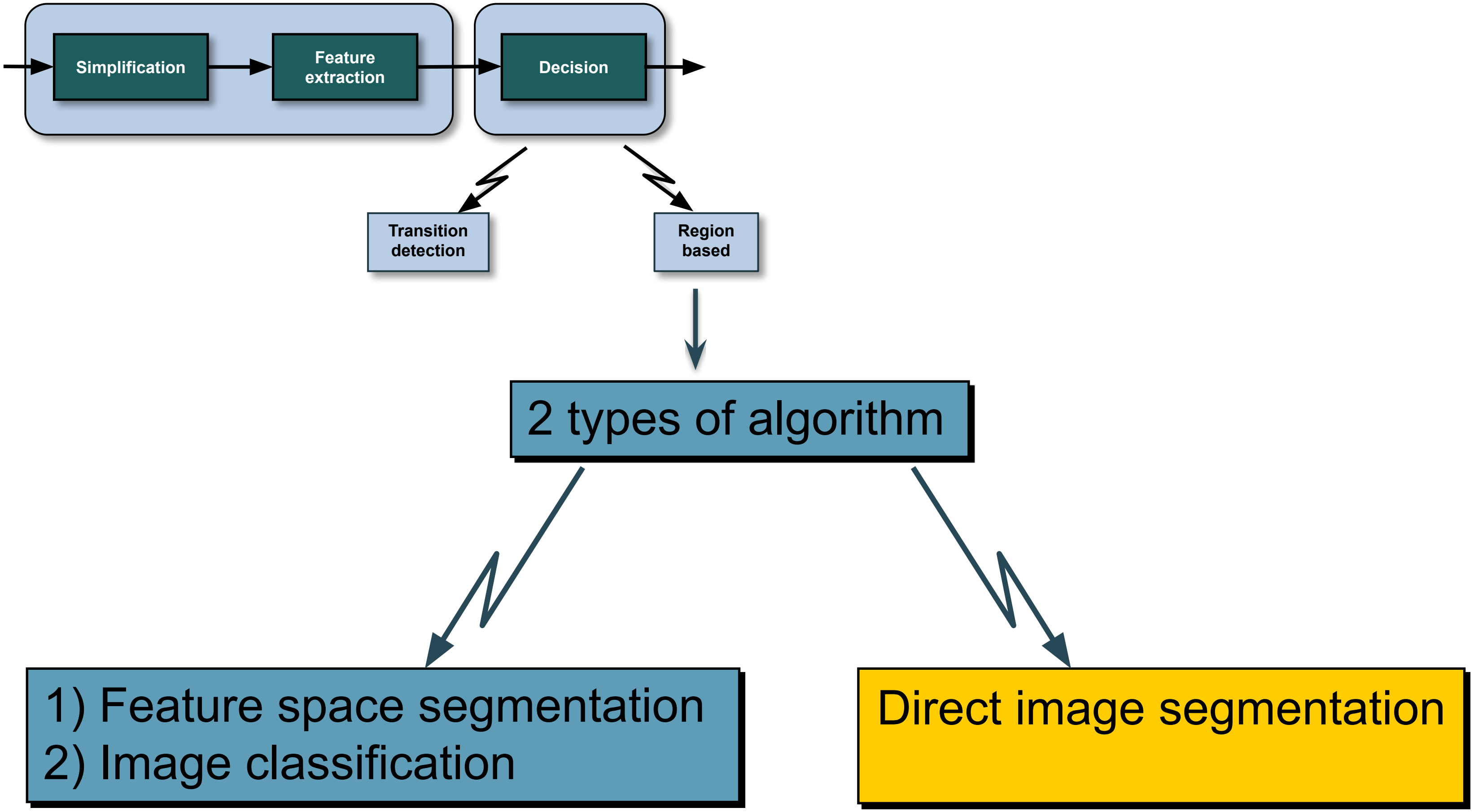
- Simple

Drawback:

- No control of the space connectivity

➢ May result on large number of **small isolated** regions

Segmentation: Homogeneity based

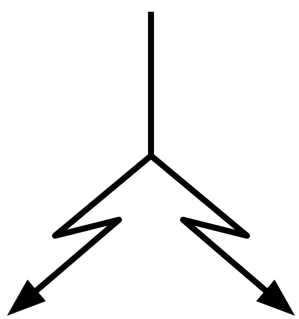


Region Splitting, merging, growing

Homogeneity based – Region splitting, merging & growing

Strategy:

- Create an initial partition
- Define an optimization criterion
- Optimize (modify) the partition (criterion minimization)



Split **Merge**

Regions

Create / remove regions
Shift contours

Homogeneity based

Segmentation: Create a partition of the image such that

$$\begin{cases} C(R_i) = \mathrm{T}rue, \quad \forall i \\ C(R_i \cup R_j) = \text{ False}, \qquad \forall i, j \end{cases}$$

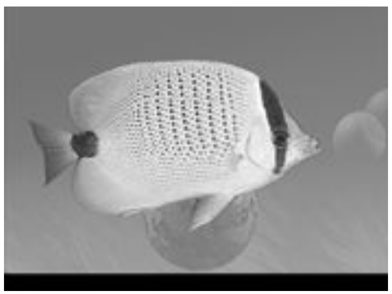
Regions should group pixels which are homogenous.

How to estimate the homogeneity?

- Define a region model
- Measure the variability of the pixels with respect to the model

Homogeneity based: region model

$x(i,j)$



R_n

$M_n(i,j)$

0 Order Polynomial (Mean)

$$M_n(i,j) = \sum_{i,j \in R_n} x(i,j)$$

1st order polynomial

$$M_n(i,j) = ai + bj + c$$

2nd order polynomial

$$M_n(i,j) = ai^2 + bj^2 + cij + di + ej + f$$

Homogeneity based: measure of variability

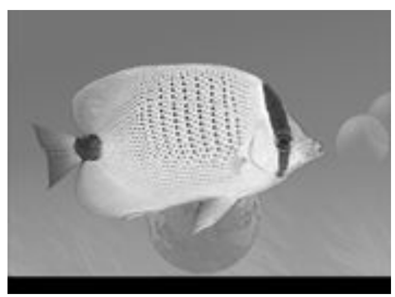
Classical example: Mean Squared Error

$$C_T = \frac{1}{N} \sum_{n} \sum_{i,j \in R_n} [x(i,j) - M_n(i,j)]^2$$

R_n: Region
i,j: Position
N: Pixel num.

Gray level (or color)

Region model:
- Constant (mean)
- Polynomial, ...



0 Order polynomial

But is this enough?

Segmentation: Homogeneity based

4 Examples:

- Split & Merge (top-down)
- Region merging (bottom-up)
- Region growing (seeds)
    - Watershed

Homogeneity based – Split & Merge

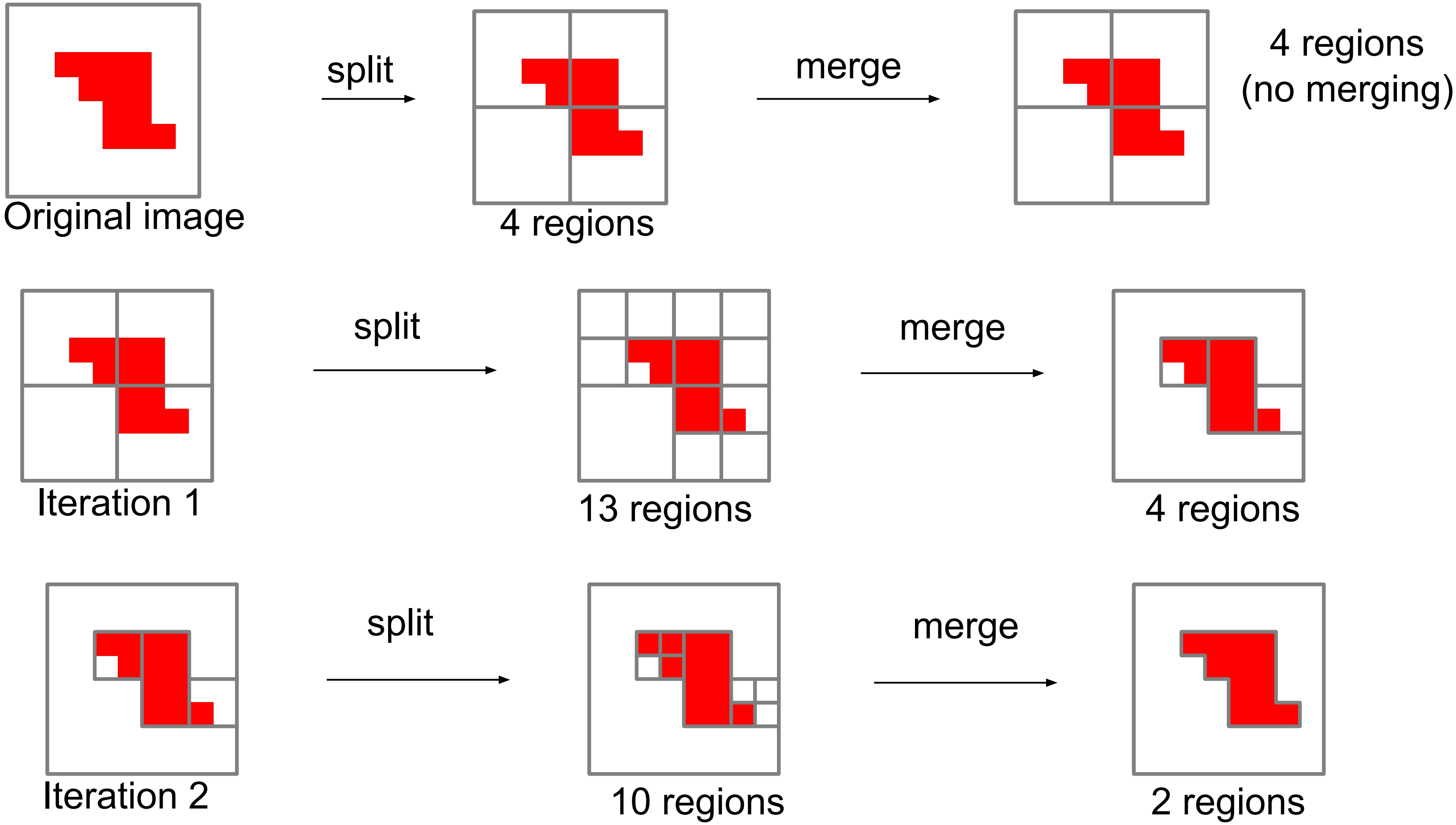
1) Initial partition: Image = 1 region (top-down)
2) Classical criterion, C: Mean squared error with respect to a 0, 1st or 2nd order model

---

"Split":
If $C(R_i)>T_0$ ➛ Geometric splitting

"Merge":
If $C(R_i \cup R_j)<T_1$ ➛ Merge R_i and R_j

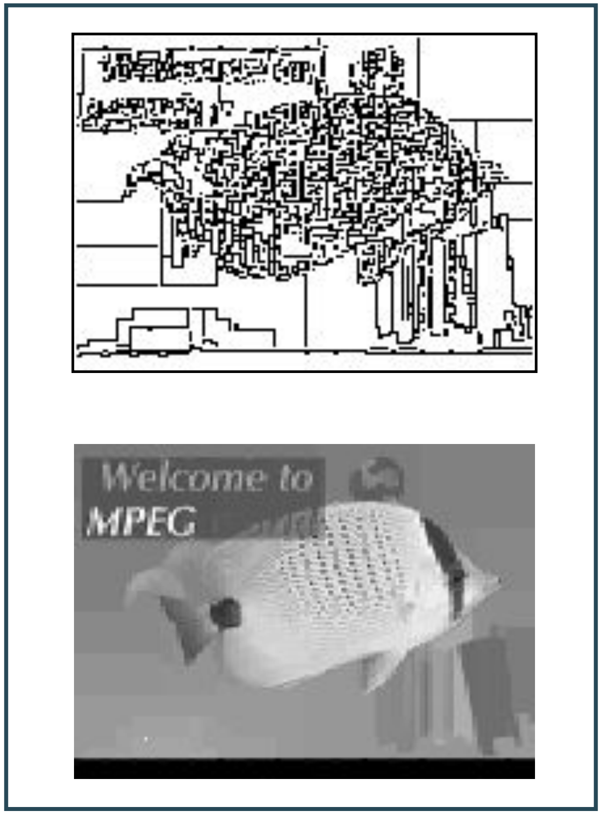
Homogeneity based – Split & Merge



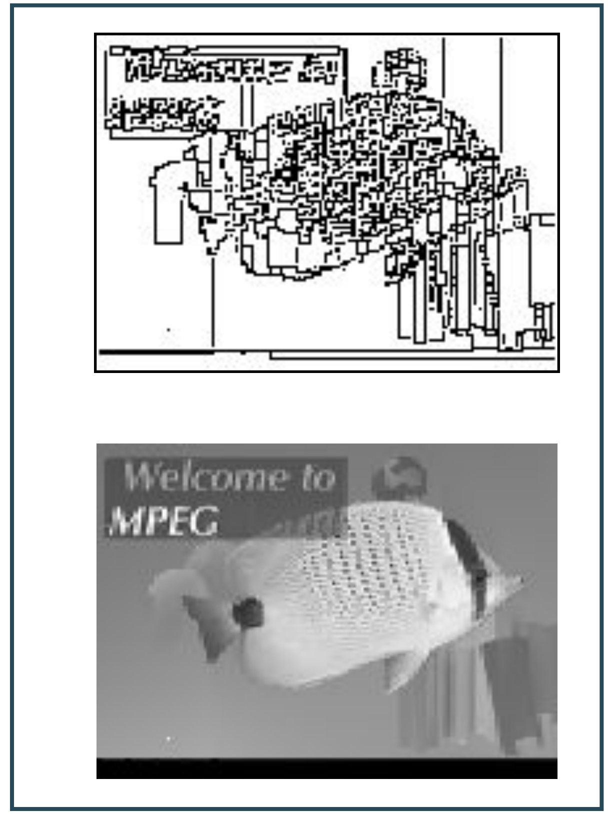
Split → If region is not homogeneous, split in 4
Merge → If neighboring regions are homogenous, Merge them

Homogeneity based – Split & Merge

0 order model



2nd order model



Split & Merge:

- Simple
- Simple initial partition
- Global view

- Pure geometrical split
- Contour are not always natural

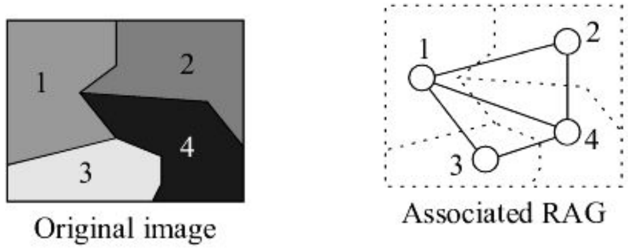
Segmentation: Homogeneity based

4 Examples:

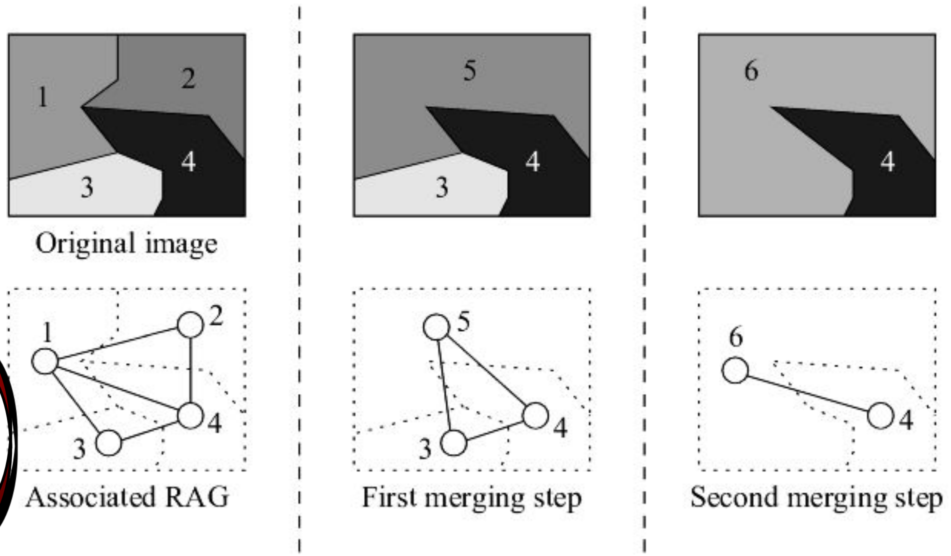
- Split & Merge (top-down)
- Region merging (bottom-up)
- Region growing (seeds)
    - Watershed

Homogeneity based – Region merging

- Region Adjacency Graph (RAG)
  - Node: regions of the image
  - Edge: Neighbor relationship



- Iterative merging process:
  - Define an initial partition and its RAG (nodes: regions, edges: neighboring region similarity)
  - Find the pair of most similar neighboring regions
  - Merge them
  - Update the edges



Homogeneity based – Region merging

- Initial partition:
  - Each pixel is an individual region
  - Over-segmentation
- Similarity between regions:
  - Approximation of Mean Squared Error:

$$C_{color}(R_1,R_2) = N_{R1}\left\| M_{R1} - M_{R1\cup R2} \right\|_2^2 + N_{R2}\left\| M_{R2} - M_{R1\cup R2} \right\|_2^2$$

  - Contour length variation:

$$C(R_1,R_2) = \alpha C_{color}(R_1,R_2) + (1-\alpha) C_{cont}(R_1,R_2)$$

$$C_{cont}(R_1,R_2) \approx -\text{Length of common contour}$$

- Stopping criterion:
  - Given number of regions
  - PSNR resulting from modeling the regions of the partition

Individual pixels

M_{R_1} M_{R_2} $M_{R_1 \cup R_2} = (M_{R_1} + M_{R_2})/2$

$$(M_{R_1} - M_{R_1 \cup R_2})^2 + (M_{R_2} - M_{R_1 \cup R_2})^2$$

Homogeneity based – Region merging

Region number (50)

22,00 dB [50 regions] | 29,50 dB [50 regions] | 22,13 dB [50 regions] | 32,29 dB [50 regions]

PSNR (26dB)

[26 dB] 273 regions | [26 dB] 13 regions | [26 dB] 347 regions | [26 dB] 7 regions

Slide credit: Verónica Vilaplana

Segmentation: Homogeneity based

4 Examples:

- Split & Merge (top-down)
- Region merging (bottom-up)
- Region growing (seeds)
  - Watershed

Homogeneity based – Region growing

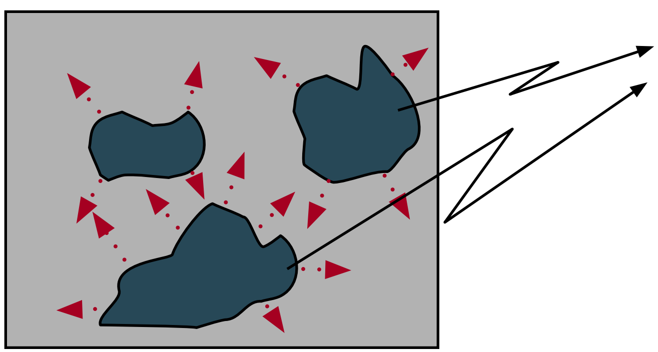
Basic strategy:

- Initial segmentation: Partial segmentation

$$\bigcup_i R_i \neq E$$

$$E \setminus \bigcup R_i = \text{Uncertainty zone}$$

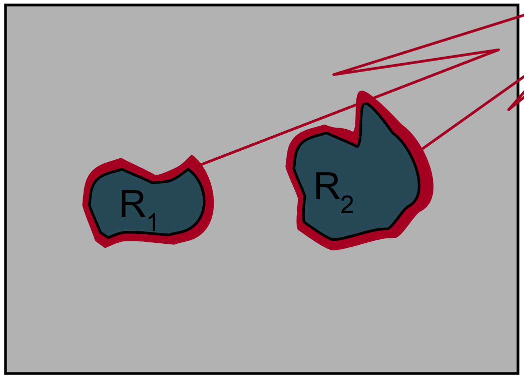
- Progressive elimination of the uncertainty zone



Initial regions:

- Regions interior
- “Safe” areas
- Application dependent

Homogeneity based – Region growing



Pixels from the uncertainty area that are connected to regions: $\{p_j^{R_i}\}$

Assign iteratively the "closest" pixel

$p_j^{R_i}$ is assigned to R_i if:

$$C\{R_1,\ldots,\{p_j^{R_i} \cup R_i\},\ldots,R_k\} < C\{R_1,\ldots,\{p_m^{R_n} \cup R_n\},\ldots,R_k\}$$

$$\forall (n,m) \neq (i,j)$$

$\Rightarrow$ Merging between regions and pixels from the uncertainty area

Homogeneity based – Region growing

Criterion: Local estimation of the variation of the criterion

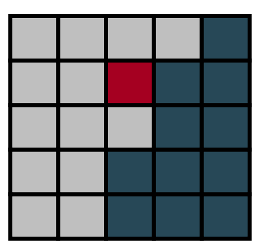
For each potential merging between p_0 and R_i, compute:

$$\Delta C = \alpha \, (p_0 - M_{Ri})^2 + (1-\alpha)\, \Delta contour$$

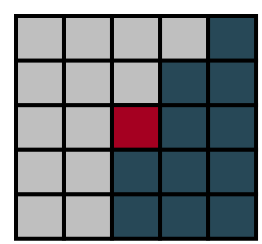
Pixel gray level

Region mean

Variation of the contour length



$\Delta contour = 3-1 = 2$



$\Delta contour = 2-2 = 0$

Homogeneity based – Region growing

Original image | Initial regions | α = 1 contour=9800

α = 0.75 cont.=8000 | α = 0.50 cont.=6300 | α = 0 cont.=3500

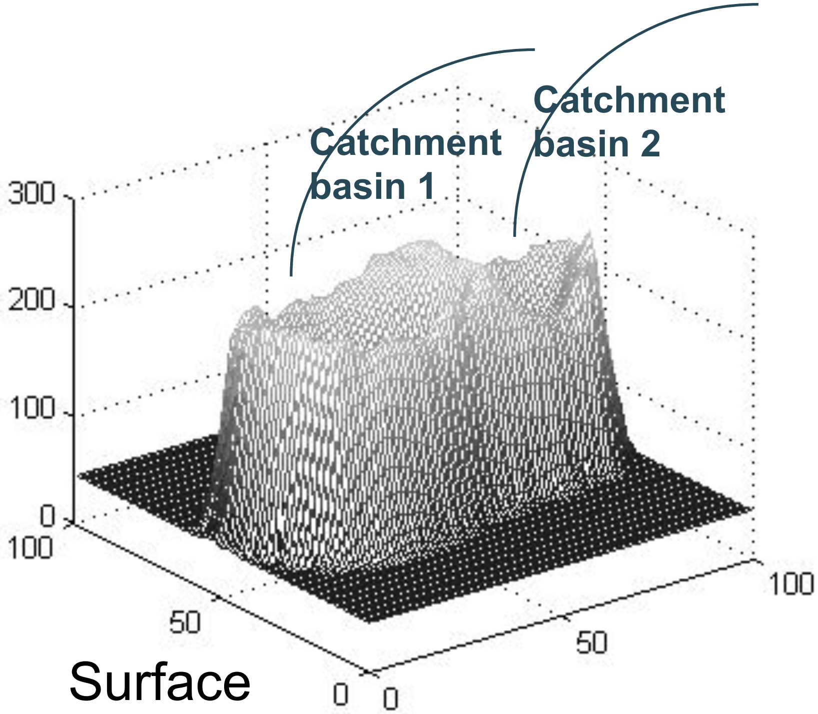
Homogeneity based – Region growing : Watershed

Watershed: Morphological approach to image segmentation through region growing

Image ➛ Topographic surface

Gray level ➛ Height

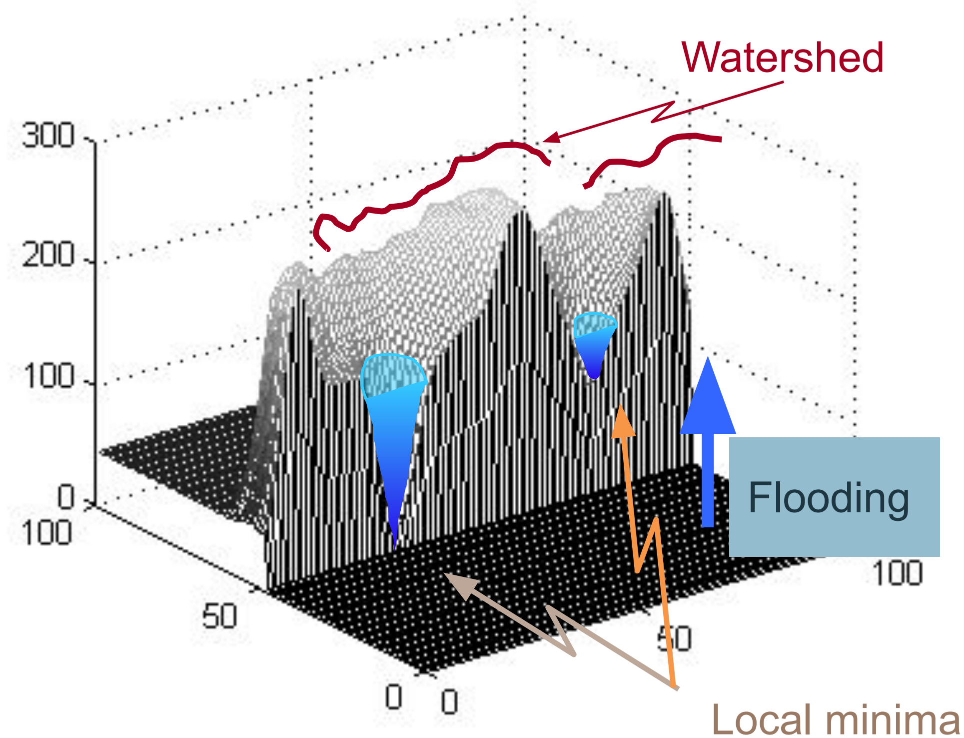
Original image



Surface

Homogeneity based – Region growing : Watershed

<u>Watershed</u>: locations where 2 (or more) catchment basins meet

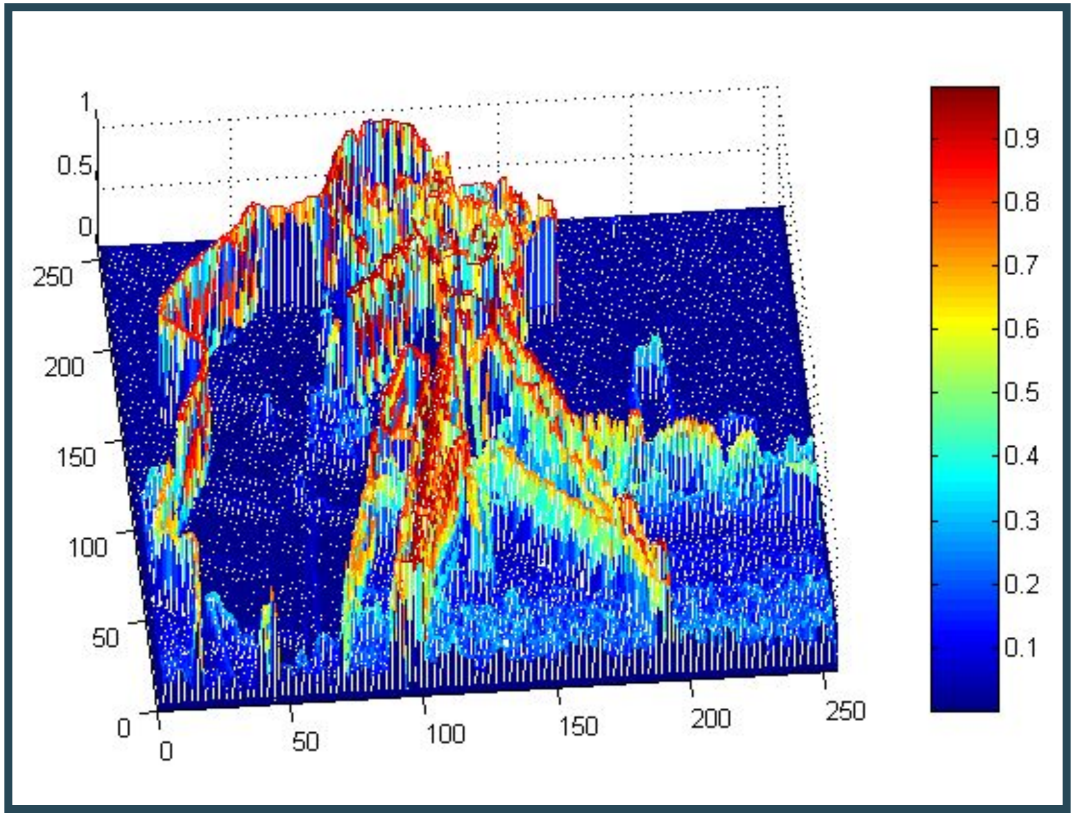


Each pixel is assigned to the catchment basin assigned to its minimum
➛ Flooding algorithm

Homogeneity based – Region growing : Watershed

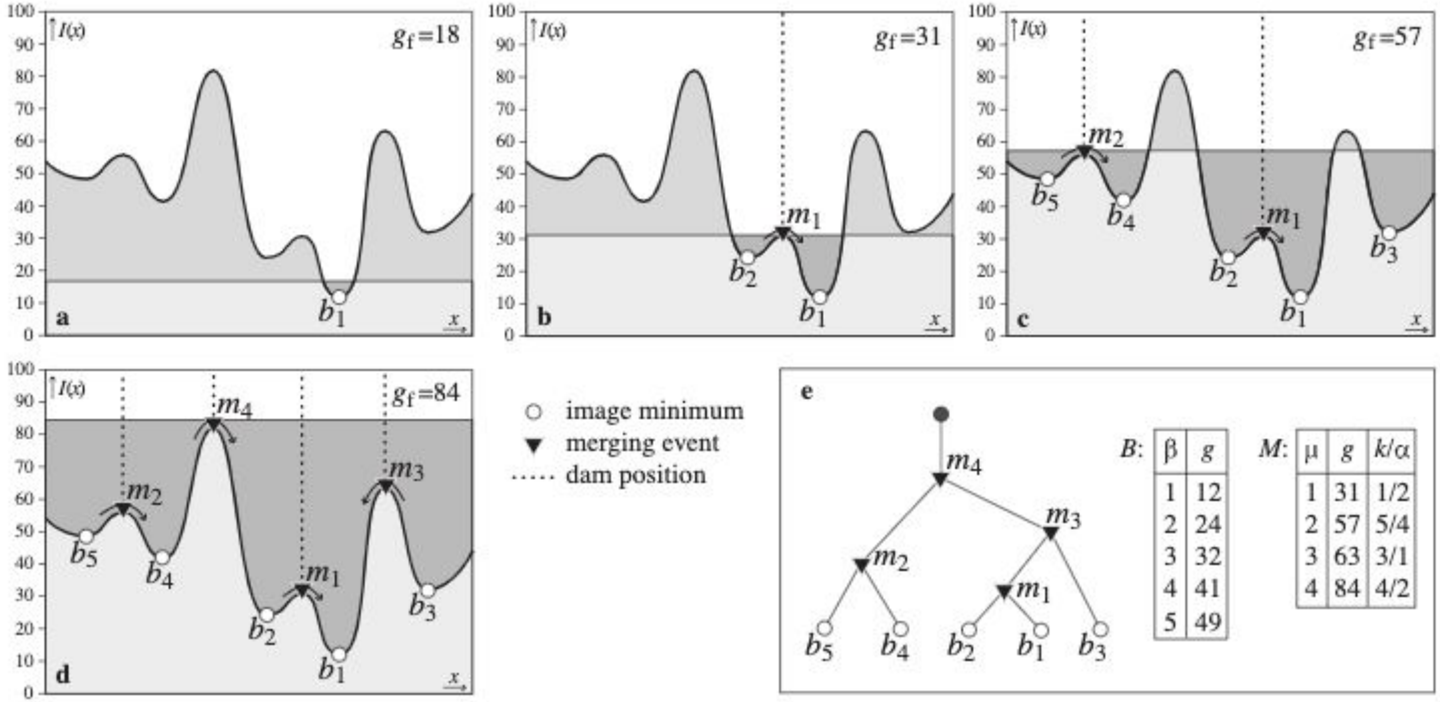
Original image

Gradient



➛ image contours = watershed of gradient

Homogeneity based – Region growing : Watershed



β	g
1	12
2	24
3	32
4	41
5	49

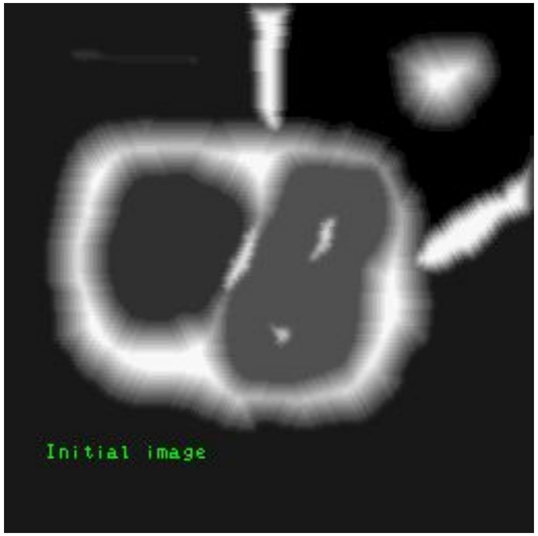
μ	g	k/α
1	31	1/2
2	57	5/4
3	63	3/1
4	84	4/2

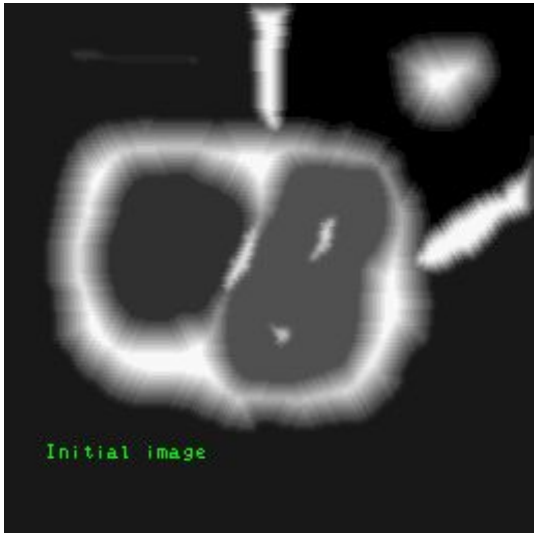
Illustration of hierarchical watershed transform on a continuous 1d function I(x) interpreted as a landscape. The landscape is sequentially flooded from bottom to top. **a:** At a flooding height g_f = 18, a single basin b_1 exists. **b:** At g_f = 31, a second isolated minimum b_2 and the lowest separating ridge have been detected. At the ridge point, a candidate for basin merging m_1 is registered and stored for future evaluation. **c:** At higher flooding heights, more and more image minima and merging procedures are registered. **d:** At g_f = 84 the flooding is complete. **e:** All (atomic) basins and merging candidates have now been detected and are ordered in a hierarchical tree. Relevant information, i.e., the respective gray levels g[β] and g[μ] and merging candidates k[μ] and α[μ], are stored in the basin and merging tables B and M

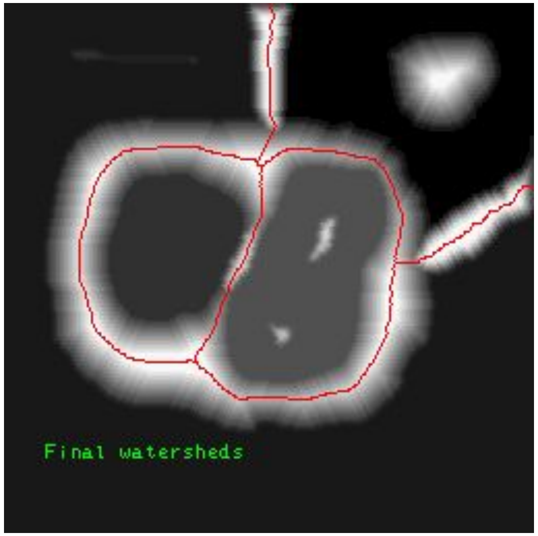
Horst K. Hahn and Heinz-Otto Peitgen, "IWT – Interactive Watershed Transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional grayscale images", Proc. Medical Imaging, SPIE 5032, San Diego, Feb 2003

Homogeneity based – Region growing : Watershed

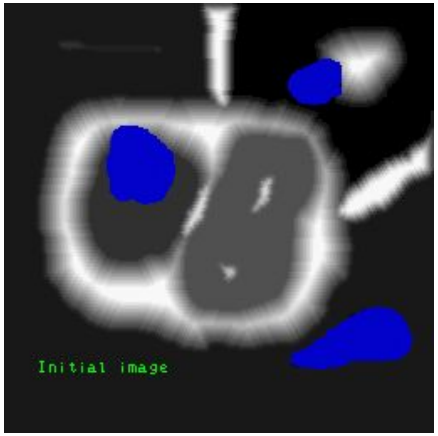
- Initial regions: Local minima of the image
- Growing process: The pixel with the gray level that is the closest to the flooded level is assigned to its corresponding catchment basin.







Homogeneity based – Region growing : Watershed



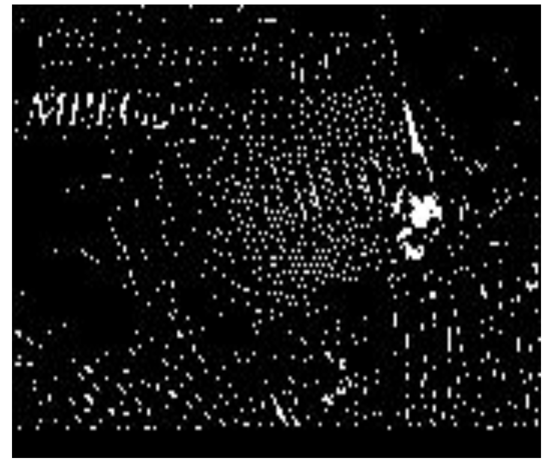
Original image with marker

Segmentation

Markers: initial partition

Homogeneity based – Region growing : Watershed

Origina
l



703 max.

Maxim
a

1055 max.

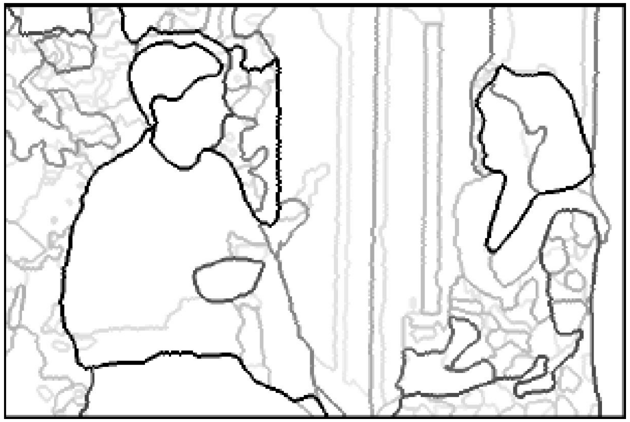
Homogeneity based – Region growing

Advantages:	Drawbacks:

• Precise contours
• Control the region connectivity
	• Need of a partial segmentation
	  • Markers
	  • Definition of a “safe” area

Mixed transition/homogeneity – gPb-OWT-UCM

- Ultrametric contour maps (UCM)
  - Transforms a contour map into a hierarchy of regions
  - Iterative merging using *Ultrametric* dissimilarities



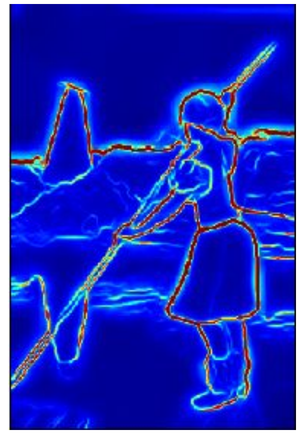
Mixed transition/homogeneity – gPb-OWT-UCM

Original image

- Local cues
- Global cues

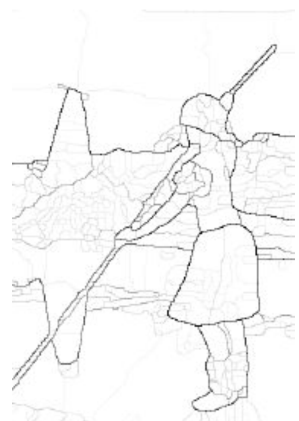
gPb contour probabilities

Contour image

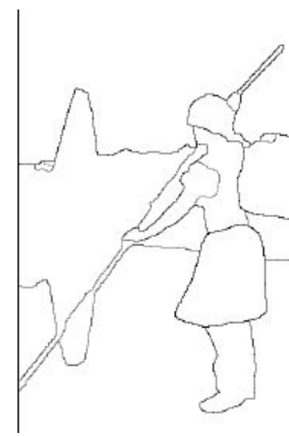


Oriented Watershed Transform

gPb: D. Martin, C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color and texture cues,” PAMI, 2004.



Iterative merging



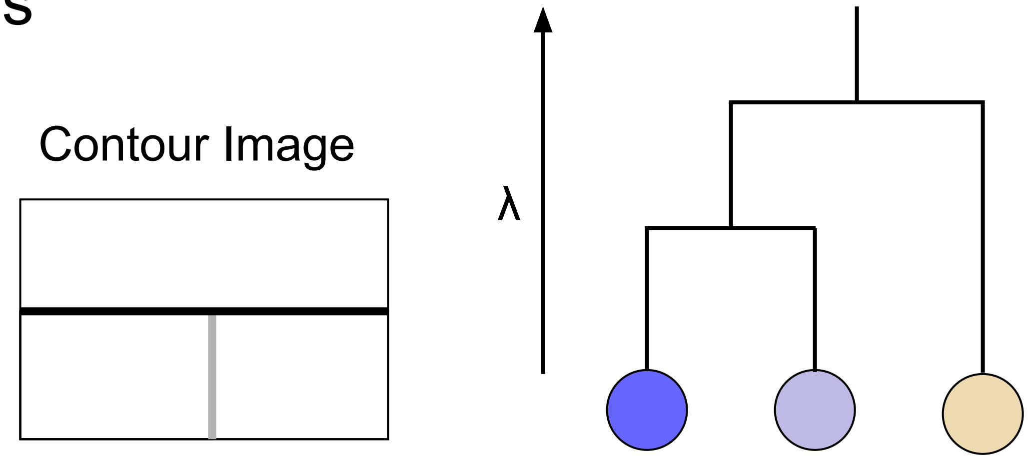
P. Arbelaez, M. Maire, C. Fowlkes, J. Malik. **Contour Detection and Hierarchical image Segmentation**. IEEE Trans. on PAMI , 2011.

Source: Hsin-Min Cheng

Mixed transition/homogeneity – gPb-OWT-UCM

- Defines a duality between closed, non-self-intersecting weighted contours and a hierarchy of regions
- The base level of this hierarchy respects even weak contours and is thus an over-segmentation of the image.
- Upper levels of the hierarchy respect only strong contours, resulting in an under-segmentation.
- Moving between levels offers a continuous trade-off between these extremes

Contour Image



Mixed transition/homogeneity – gPb-OWT-UCM

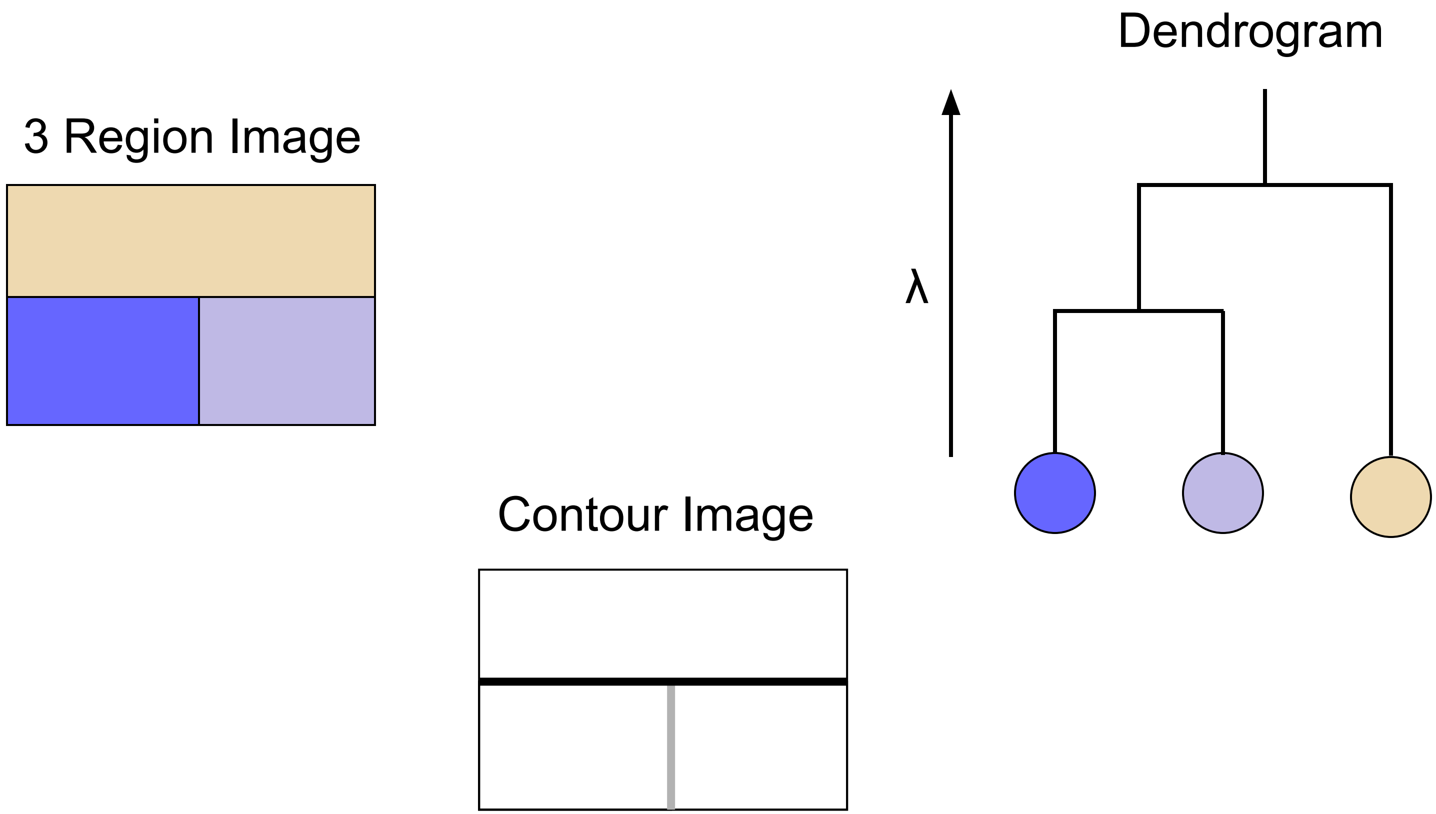
- Define an initial graph G = (P_0; K_0; $W(K_0)$)
  – The nodes are the regions, P_0
  – The links are the arcs K_0 separating adjacent regions
  – The weights $W(K_0)$ are a measure of dissimilarity between regions
- Dissimilarity between two adjacent regions is defined as the average strength of their common boundary in K_0

Algorithm:

1) Select minimum weight contour:
   $C^* = \mathrm{argmin}_{C \in \mathcal{K}_0} W(C)$.
2) Let $R_1, R_2 \in \mathcal{P}_0$ be the regions separated by C^*.
3) Set $R = R_1 \cup R_2$, and update:
   $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \backslash \{R_1, R_2\} \cup \{R\}$ and $\mathcal{K}_0 \leftarrow \mathcal{K}_0 \backslash \{C^*\}$.
4) Stop if $\mathcal{K}_0$ is empty.
   Otherwise, update weights $W(\mathcal{K}_0)$ and repeat.

Mixed transition/homogeneity – gPb-OWT-UCM

- Hierarchical segmentation



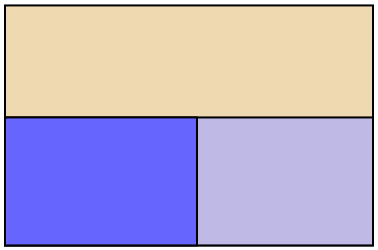
Source: Pablo Arbeláez

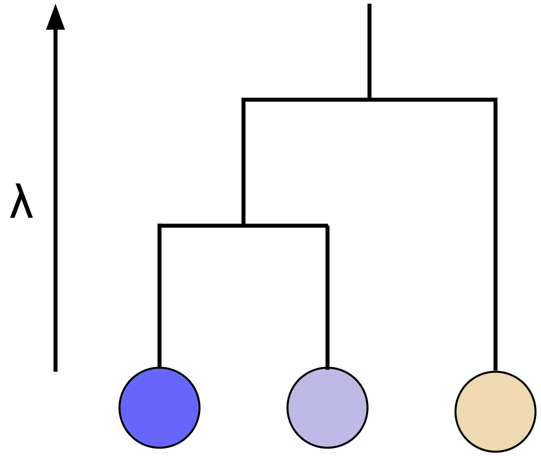
Mixed transition/homogeneity – gPb-OWT-UCM

- Ultrametric

$$D(R_1, R_2) \leq \max\{D(R_1, R), D(R, R_2)\}$$

The union R_{12} of two regions R_1 and R_2 must have >= distance to adjacent region R than either R_1 or R_2

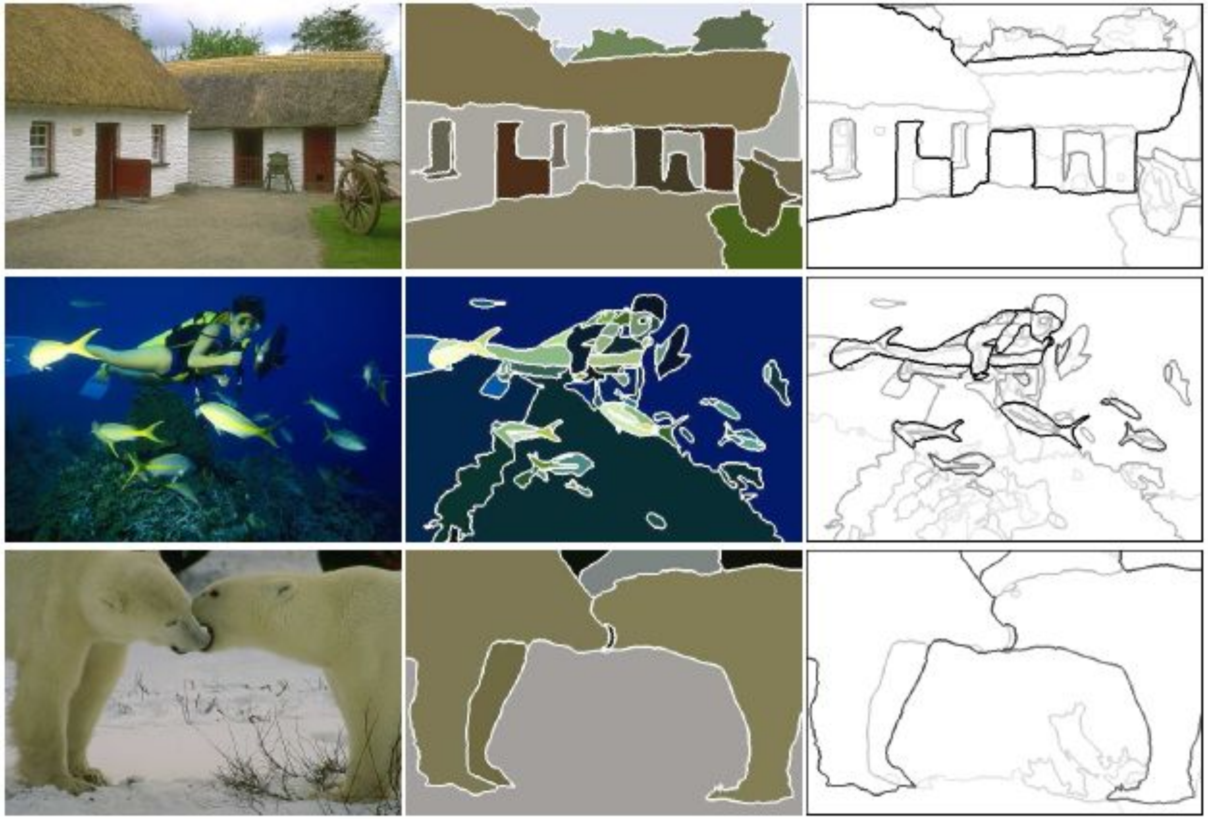




Source: Pablo Arbeláez

Mixed transition/homogeneity – gPb-OWT-UCM

Results



Source: Pablo Arbeláez

Fast UCM implementation (MCG Pre-trained, Matlab/C++):
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/mcg/

Segmentation: 2019 state of the art

- Research focused on semantic segmentation, instance segmentation

  - Mostly deep learning

    Semantic segmentation: popular architectures:
    https://towardsdatascience.com/semantic-segmentation-popular-architectures-dff0a75f39d0

    A 2019 Guide to Semantic Segmentation:
    https://heartbeat.fritz.ai/a-2019-guide-to-semantic-segmentation-ca8242f5a7fc

- Classical methods still used:
  - Max-tree representation

    Yongchao Xu, Edwin Carlinet, Thierry Géraud, Laurent Najman. Hierarchical Segmentation Using Tree-Based Shape Spaces. IEEE Transactions on PAMI, 2017, 39 (3)

    Salembier P, Liesegang S, López-Martínez C. Ship Detection in SAR Images Based on Maxtree Representation and Graph Signal Processing. IEEE Transactions on Geoscience and Remote Sensing. 2019;57(5)

Segmentation: Summary