Fundamentals of Image Processing

▶ Lecture 8: Image segmentation ◀

Master of Computer Science
Sorbonne University

September 2022

Outline

Introduction

Optimization based methods

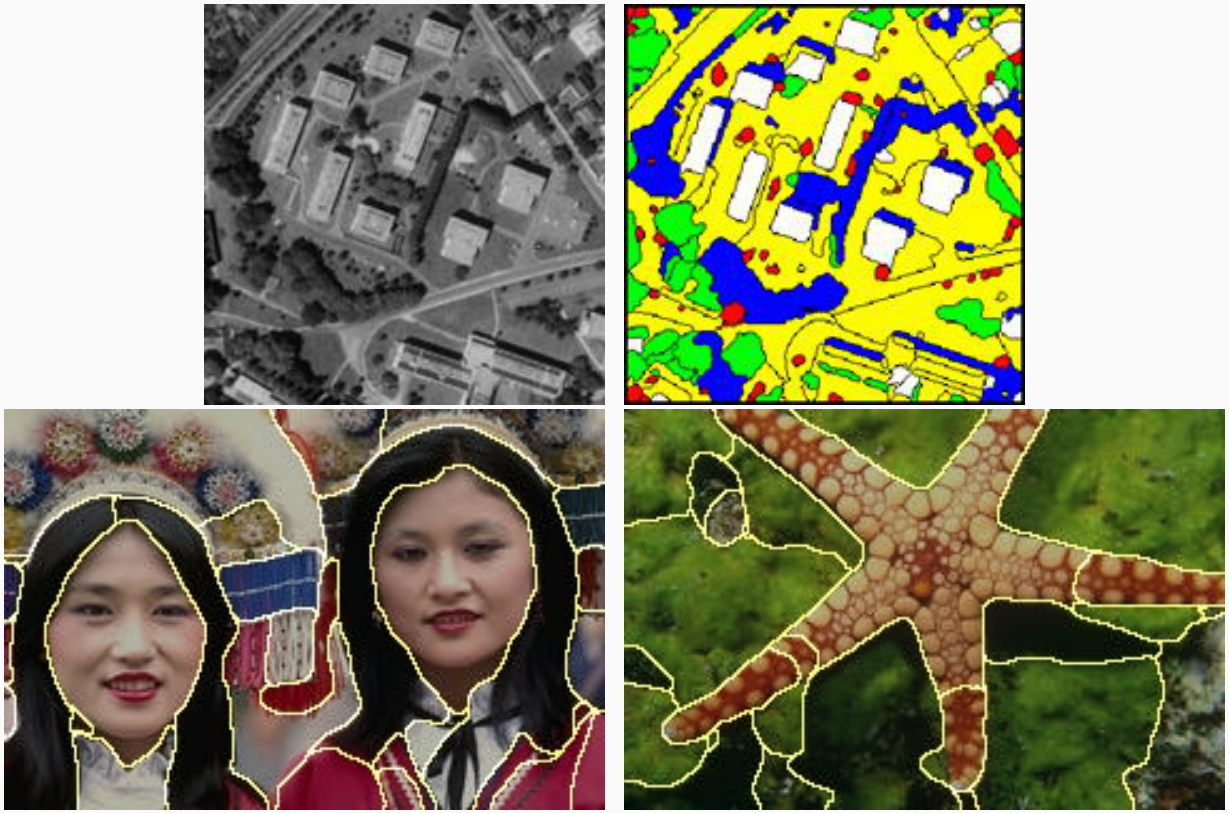
Clustering based methods

Region based methods

Definition

- Partition of the image domain into semantic objects or homogeneous regions (homogeneity according to various criteria such as color, texture, intensity...)
- Partition of image domain R: a set of k non empty subdomains R_i such that:
  - $R = \bigcup_{i=1}^{k} R_i$
  - R_i connected set (there exists a path between any two pixels of R_i, which is included in R_i)
  - $R_i \cap R_j = \emptyset, \forall i \neq j$
  - Homogeneity according to a predicate Q: $Q(R_i) = TRUE$ and $Q(R_i \cup R_j) = FALSE, \forall i \neq j$

Examples



Introduction

Links with lectures on "detection and characterization of image primitives"

- A region is an image primitive
- Given a region-based analysis algorithm
  1. Region detection $\Rightarrow$ segmentation
  2. Region description:
     - Characterization of shape, position, size...
     - Characterization of radiometric values...

Introduction

- Several types of methods:
  - Optimization point of view (general formalism)
    - Global and constrained minimization of a cost function comparing the original image and the segmented image $\Rightarrow$ **optimization based methods**
  - Radiometric point of view:
    - Histogram analysis, quantization... $\Rightarrow$ **clustering based methods**
  - Geometric/region point of view:
    - Region growing, split and merge... $\Rightarrow$ **Region based methods**
- Difficulties:
  - Issue of the detection robustness (repeatability)
  - Ill-posed problem: many acceptable solutions, highly context dependent
  - Often high computational cost

Outline

Introduction

Optimization based methods

Clustering based methods

Region based methods

Optimization based methods

- Principle:
  - Constrained minimization of a functional measuring the distance between the segmented and original images
  - Constraints can be related to the space of solutions (regular functions, piecewise constant functions...)
- Many admissible image partitions exist...
  How to choose?
- Additional constraints:
  - minimize partition cardinality
  - maximize the size of the smallest region
  - maximize the difference between adjacent regions
  - ...

Optimization based methods: energy functional

- Energy term for each region R_i: a weighted sum of two terms:

$$E(R_i) = E_I(R_i) + \lambda E_C(R_i)$$

- Energy term for the segmented image $R = \cup_i R_i$:

$$E_\lambda(R) = \sum_i E_I(R_i) + \lambda E_C(R_i)$$

- Data energy (or fidelity term): energy term describing the link between the image I (input data) and the segmented image R:
  - low value if the segmented image R is close to the initial image I
- Complexity (or internal) energy: does not depend on I, only R, low value if few regions, or regions with regular shape...
- λ regularization parameter

Optimization based-methods: two paradigms

$$E_\lambda = E_I + \lambda E_C$$

	Variational Mumford-Shah, 1989	Bayesian Geman-Geman, 1984
E_I	Data term	Likelihood potential
E_c	Regularization term	Prior potential
E_λ	Total energy	Posterior potential

Data term: Mumford and Shah model (simplified)

- Constant piecewise model: each region is represented by a constant, estimated as the mean of the region intensities
- Notations:
  - Let R be a region containing N pixels of values $X_1, \cdots, x_N$, elements of $\mathbb{R}^3$ (for color images), $X_i = (x_i^1, x_i^2, x_i^3)$.
  - Let m^1, m^2 , m^3 be the means of the R pixel values in each color channel, $m = (m^j)_{j=1,2,3}$
  - $\|.\|$ Euclidean norm
  - Data term: distance between I and the mean of R: $E_I(R) = \sum_{k=1}^{N} \|X_k - m\|^2 = \sum_{k=1}^{N} \sum_{j=1}^{3} (x_k^j - m^j)^2$
- A quick calculation:
  - Let V be the 3×3 variance/covariance matrix: $v(i,j) = \sum_{k=1}^{N} (x_k^i - m^i)(x_k^j - m^j)$
  - λ_j eigenvalues of V: $E_I(R) = \sum_{j=1}^{3} v(j,j) = \mathrm{Tr}(V) = \sum_{j=1}^{3} \lambda_j$

Other energies

- Alternative data energies:
  - Gaussian model
  - Other distributions of gray-level / color values.
- Complexity/regularity energy term E_C
  - Example: length of region R boundary

Optimization based-methods

Primal problem

$$\min D(I, x) \quad \text{s.t.} \quad C(x) \leq \epsilon$$

Of two models with equal complexity, the best model has the highest fidelity

Dual problem

$$\min C(x) \quad \text{s.t.} \quad D(I, x) \leq \lambda$$

Of two models with equal fidelity, the best model is the less complex

Two constrained problems equivalent from the Lagrange multipliers method point view:

$$\text{minimizing } \mu_1 C(x) + D(I, x) \Leftrightarrow \text{minimizing } \mu_2 D(I, x) + C(x)$$

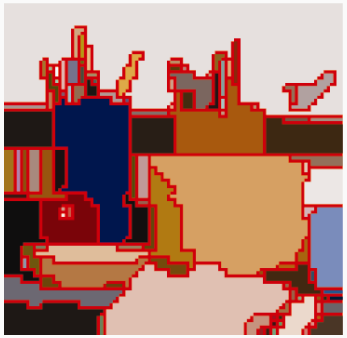
Here, with the previous notations: $C(x) = E_C(R)$, $D(I, x) = E_I(R)$

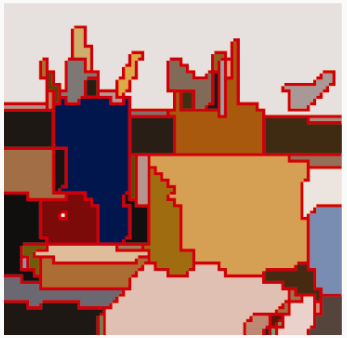
Optimization based-methods

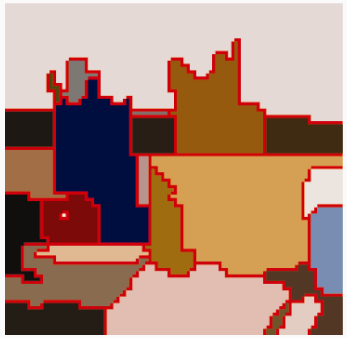
- Result of Mumford-Shah method with various values of λ:


Image

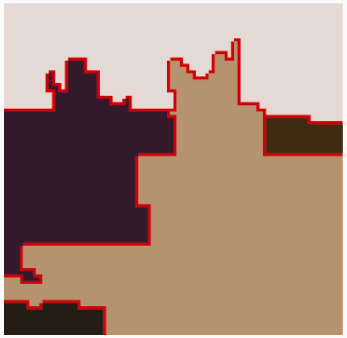

$\lambda = 0$


$\lambda = 0.1$


$\lambda = 0.2$


$\lambda = 0.5$


$\lambda = 0.8$


$\lambda = 0.9$


$\lambda = 1.2$

Outline

Introduction

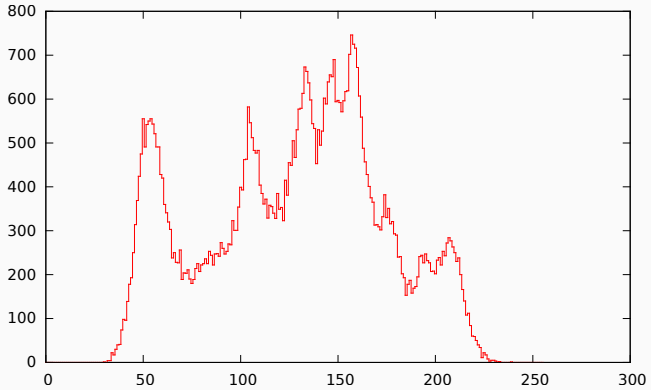
Optimization based methods

Clustering based methods

Region based methods

Histogram based methods

- Basic example: binary thresholding

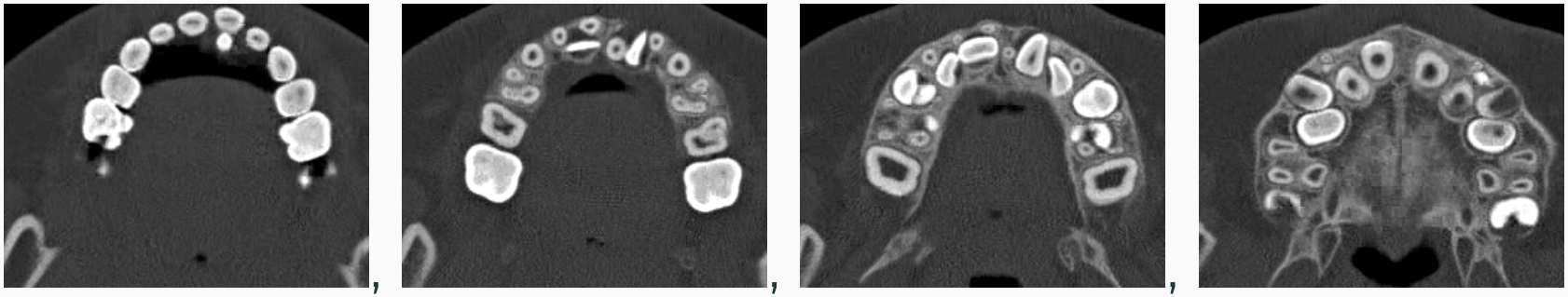


threshold S

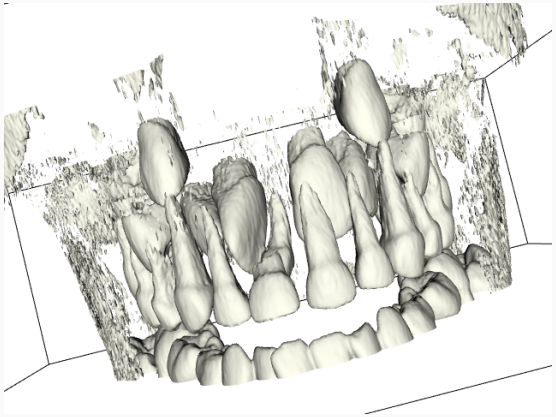
$$f(x,y) = 0 \quad \text{if } I(x,y) < S$$
$$f(x,y) = 255 \text{ otherwise}$$

Thresholding

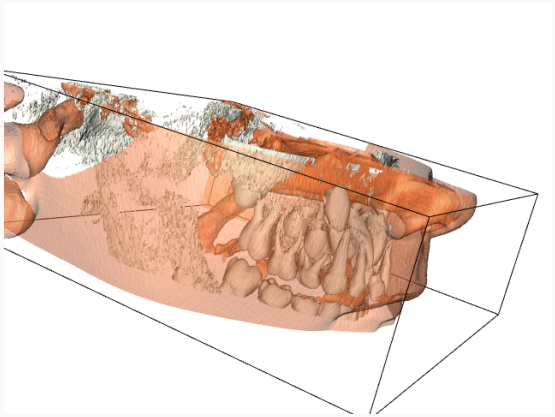
Choice of threshold value: supervised ... sometimes possible



(a) 4 slices of dental X-CT, dynamic range of 12 bits



(b) Threshold at 80: skull and teeth



(c) Threshold at 500: skin

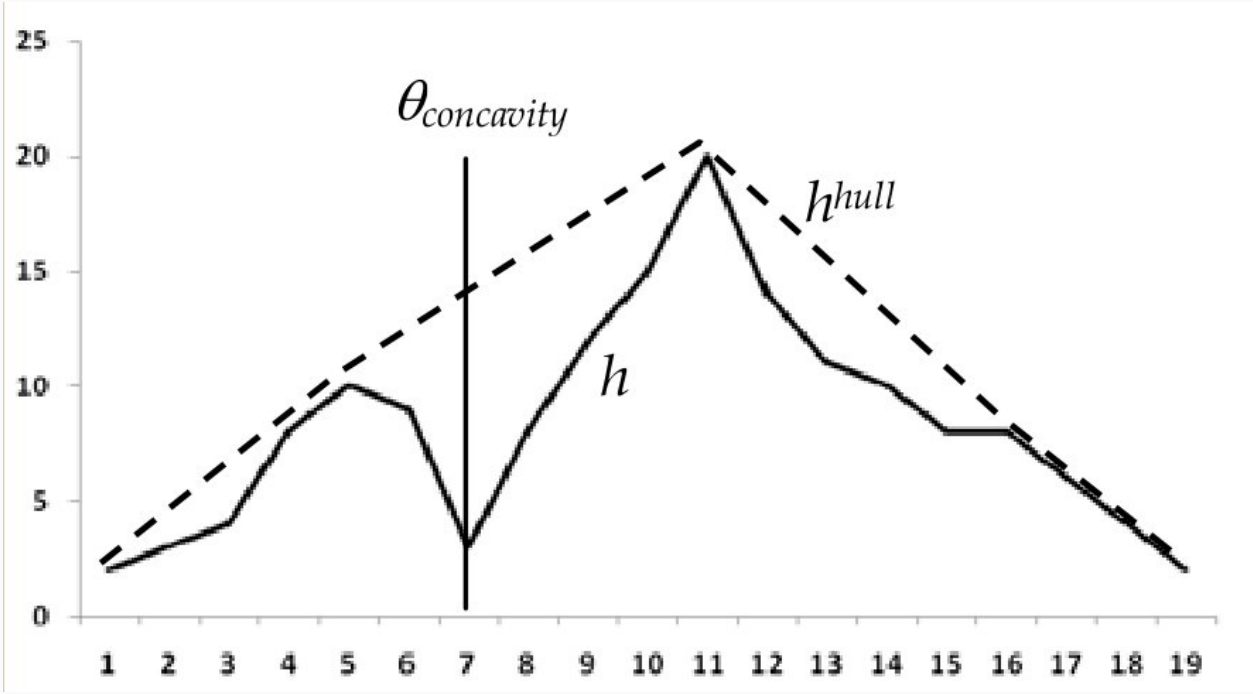
Threshold value selection

- *Survey over image thresholding techniques and qualitative performance evaluation*, Sezgin and Sankur, 2004: 40 methods, 6 categories:
  1. Histogram shape-based methods, where, for example, the peaks, valleys and curvatures of the smoothed histogram are analyzed.
  2. Clustering-based methods, where the gray-level samples are clustered in two parts as background and foreground object, e.g. modeled as a mixture of two Gaussians.
  3. Entropy-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image...
  4. Object attribute-based methods maximize a measure of similarity between the gray-level and the binarized images, such as fuzzy shape similarity, edge coincidence...
  5. Spatial methods use higher-order probability distributions and/or correlation between pixels.
  6. Local methods adapt the threshold value at each pixel according to the local image characteristics.

Thresholding

Example 1: concavity analysis (Rosenfeld, 1983)

- Let h be the image histogram
- h^{hull}, convex hull of h.
- Choice: $t = \underset{g}{\operatorname{argmax}} |h(g) - h^{\text{hull}}(g)|$



Example 2: Otsu method, 1979

- Find the threshold value t minimizing the **intra-class intensity variance** (sum of the variance of the two classes) of an image I with L levels intensity values
- Two classes:
  - $C_1(t) = \{(x, y) \mid 0 \leq I(x, y) < t\}$
  - $C_2(t) = \{(x, y) \mid t \leq I(x, y) < L\}$
- Intra-class variance:

$$\sigma^2_{intra}(t) = n_1(t)\sigma^2_1(t) + n_2(t)\sigma^2_2(t) \tag{1}$$

with

- n_1 and n_2 sizes of C_1 and C_2
- σ^2_1 and σ^2_2 intensity variances of C_1 and C_2

Thresholding

Otsu method

- Minimizing the intra-class variance means:
  - minimizing the variances of C_1 and C_2 weighted by their size (n_1 and n_2)
  - find the most homogeneous regions C_1 and C_2

Theorem (Redi *et al*, 1984)
The threshold value minimizing the intra-class variance is the limit of the following numerical scheme:

$$t_0 = \mu_0, \quad t_{k+1} = \frac{1}{2}(\mu_1(t_k) + \mu_2(t_k))$$

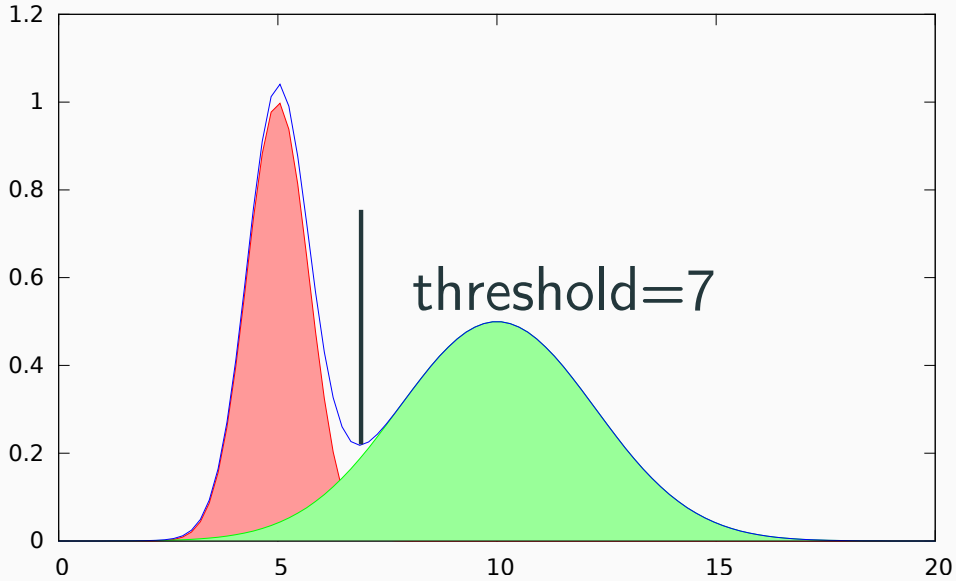
- Practically: the scheme is iterated until $|t_k - t_{k+1}| < \epsilon$, convergence is obtained after about 10 iterations for a 1-byte image

Otsu: example

Figure 1: MRI

Histogram based methods

- Choice of the threshold (a third method): statistical learning (Gaussian mixture)

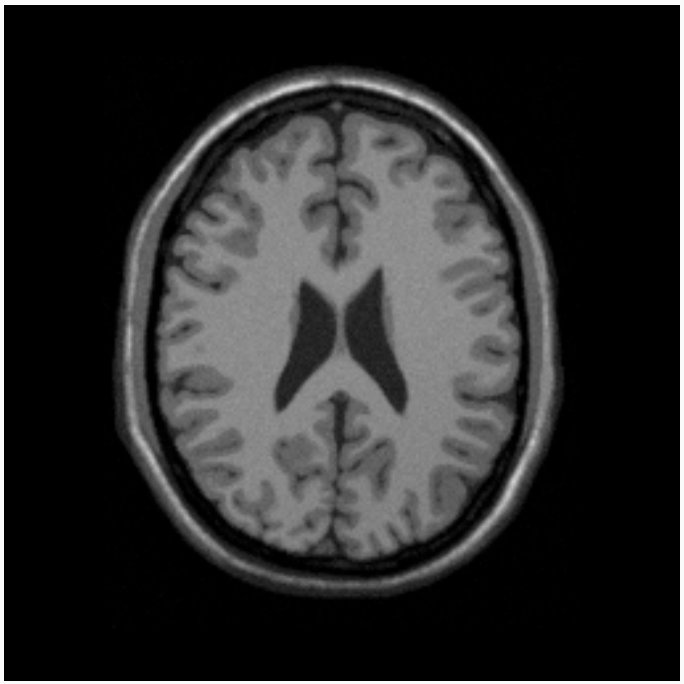


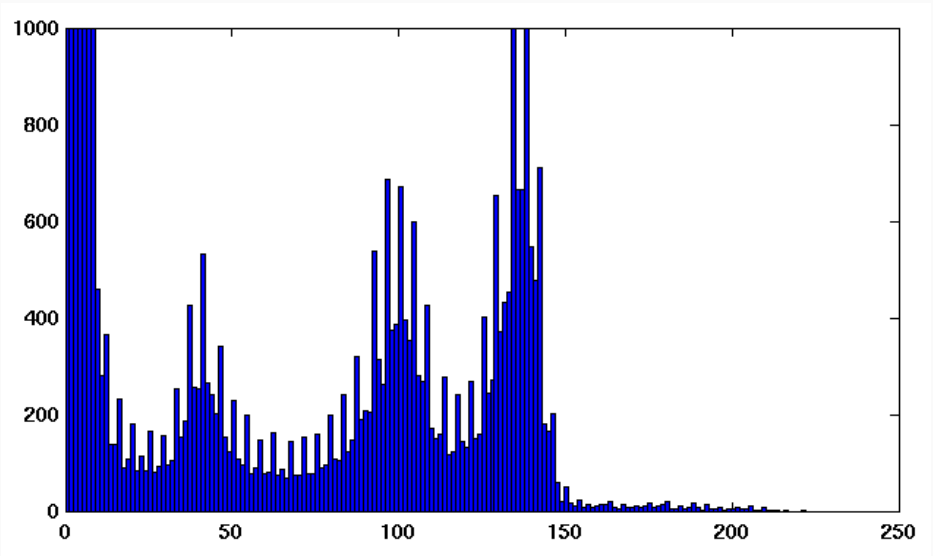
Histogram based methods

- Thresholding is not a segmentation method (no insurance to have connected sets): improvement with a spatial analysis
- Principle:
  1. Locate an isolated mode in the histogram
  2. Apply a thresholding to detect pixels belonging to this mode
  3. Among the connected sets matching this mode, select the biggest one
  4. Back to 1

Histogram based methods

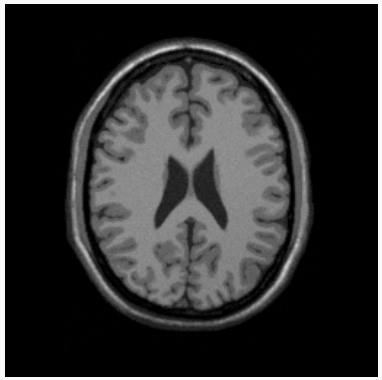
- Example (brain MRI image)

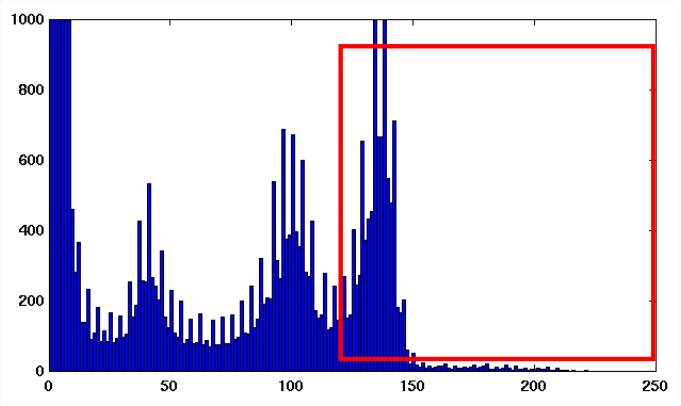




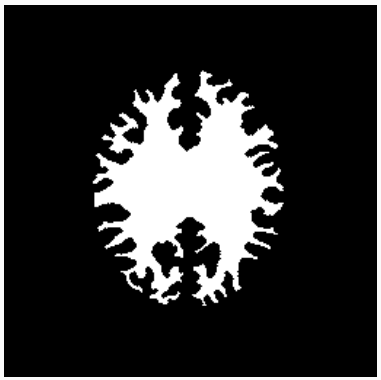
Histogram based methods

- Locate in the histogram a first mode:



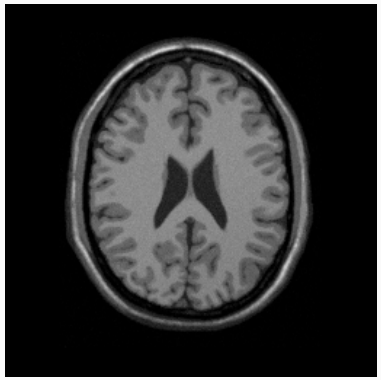


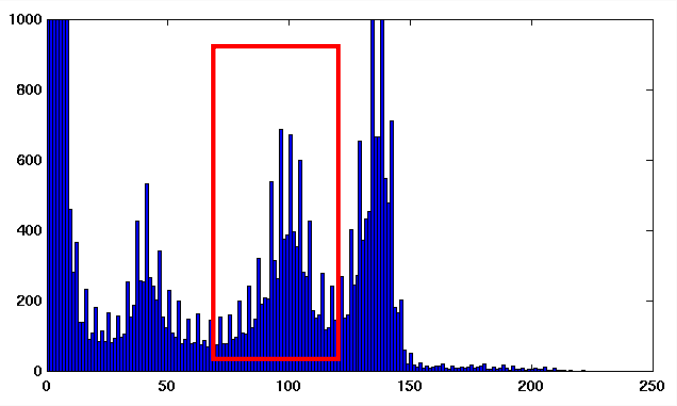
- Select the biggest connected set:



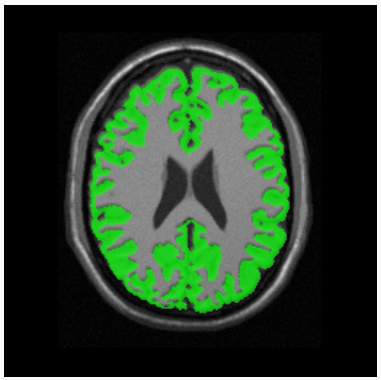
Histogram based methods

- Locate in the histogram a second mode:



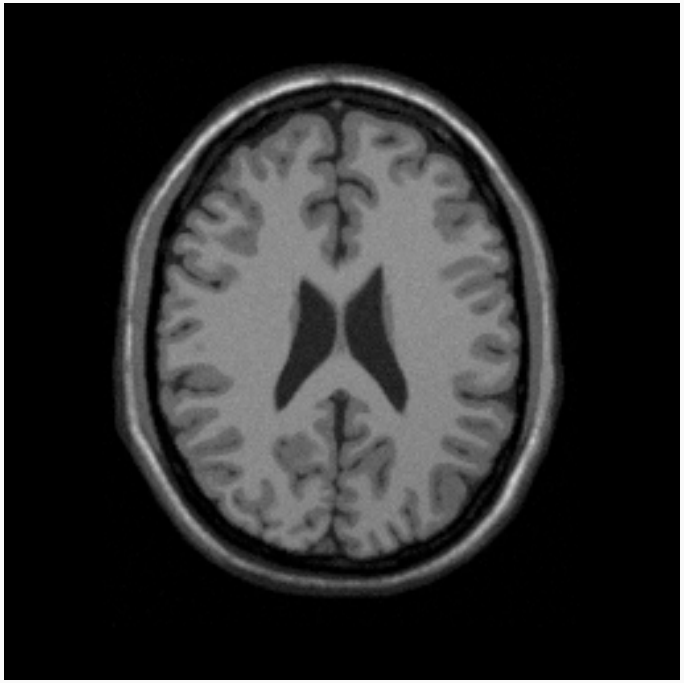


- Selection of the largest connected component

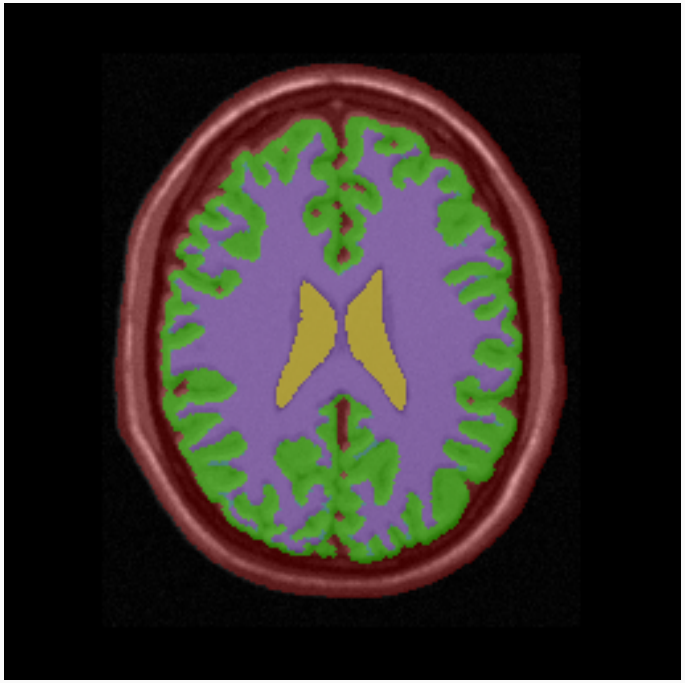


Histogram based methods

- Final result, after processing all modes:



original image



segmented image

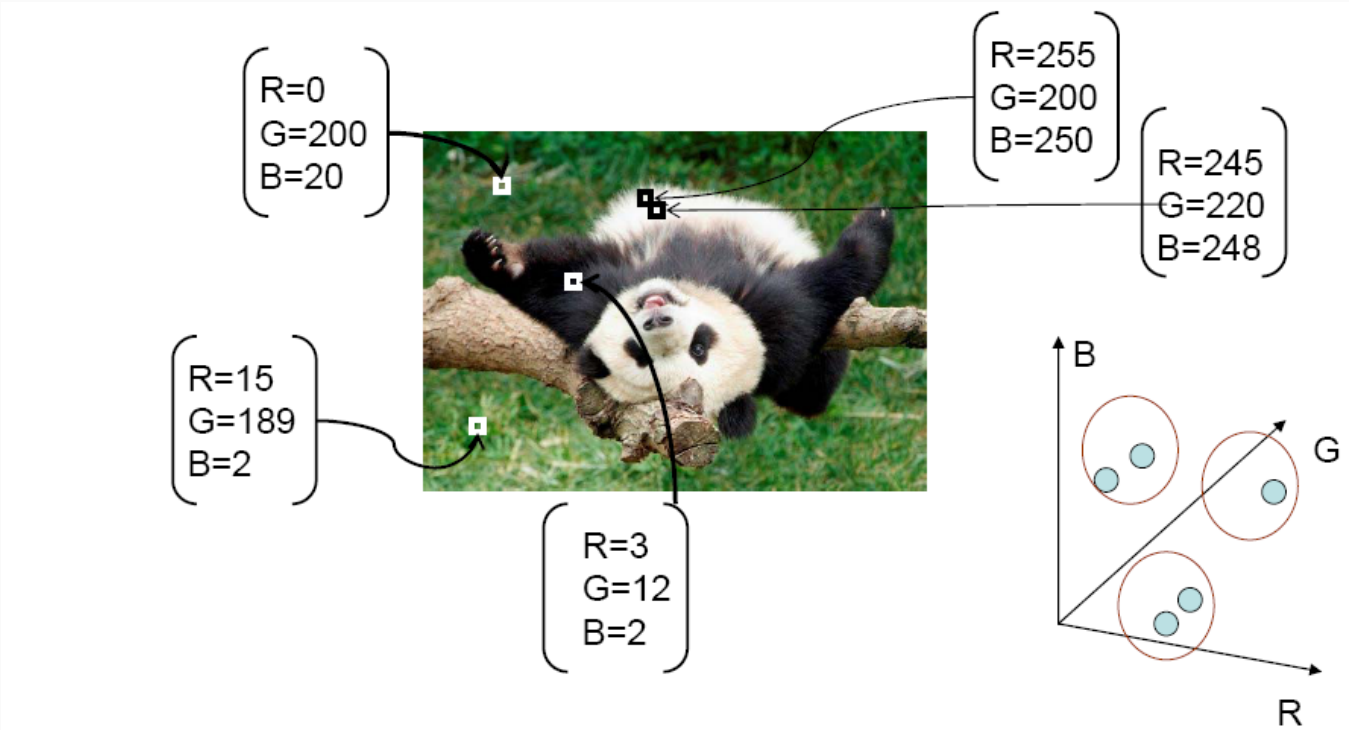
Histogram based methods

- Concluding remarks on histogram based methods:
  - Easy to code
  - Limited performance, may be efficient for homogeneous regions (as MRI)
  - No or few spatial information
- Extension if the pixel descriptor is a vector and not a scalar (intensity value)

  $\Rightarrow$ from histogram to clustering

Clustering based methods

- Multi-dimensional clustering:
  - Partition of vector values into homogeneous groups (clusters)



The k-means method

- k-means is an iterative algorithm with 4 steps:
  1. Choose k (number of clusters) and initialize centers $M_i, i = 1...k$
  2. (Re)assign each object or pixel O to the cluster C_i of center M_i such that dist(O, M_i) is minimal
  3. Update M_i for each cluster (as the average of values of O in C_i)
  4. Back to step 2 or stop if there is no more reassignement

  $\Rightarrow$ Finds a local minimum of $\sum_{i=1}^{k} \sum_{O \in C_i} \text{dist}(O, M_i)^2$

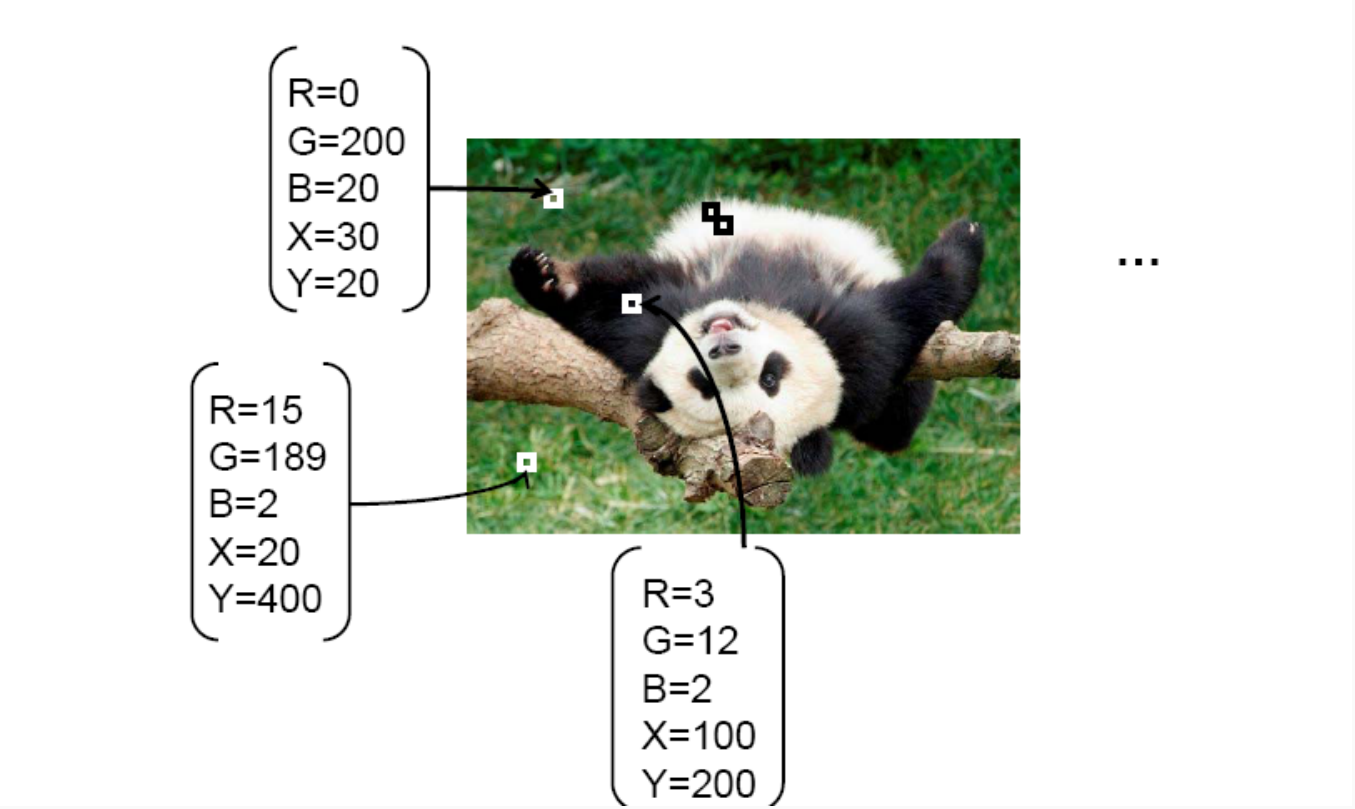
Segmentation based on clustering

- k-means clustering
  - No spatial consistency



Segmentation based on clustering

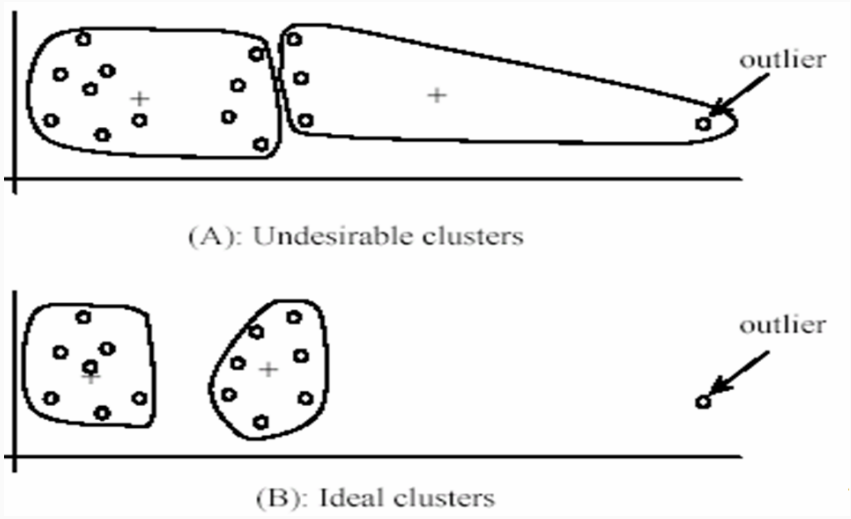
- To cluster pixels having similar features



Segmentation based on clustering

k-means :

- Pros:
  - Simplicity
  - Convergence insured (but only towards a local minimum)
- Cons:
  - k (# of clusters) has to be known, in practice this is unknown
  - Sensitive to initialization (initial centroid location)
  - Clusters are assumed spherical, distinct and approximatively of same size (in the feature space)



Mean shift algorithm (Comaniciu and Mee, 2002)

- Mean shift is an algorithm to search the modes (local maxima) of a distribution observed in the feature space

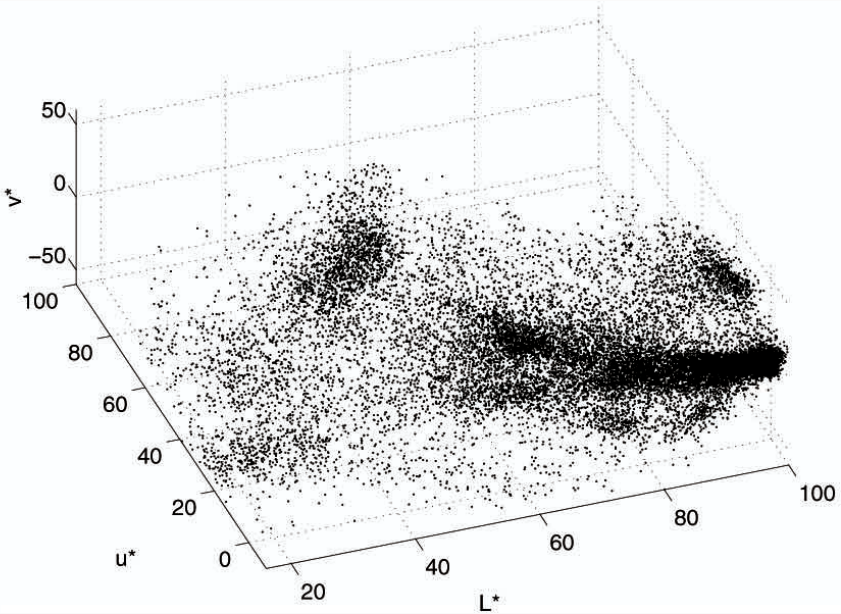


Image (left), space of features (L*u*v color coding, right)

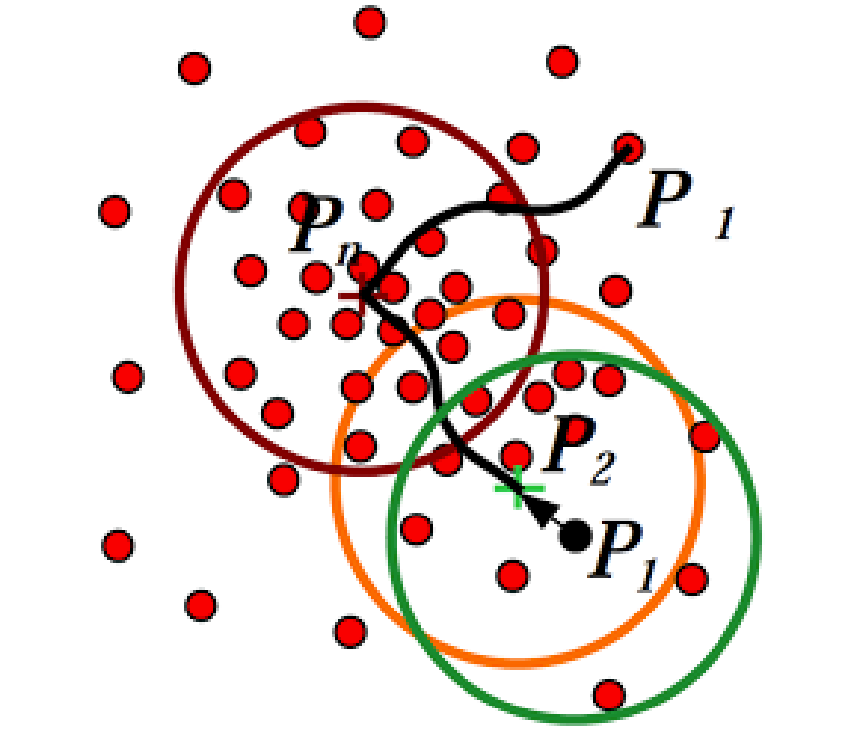
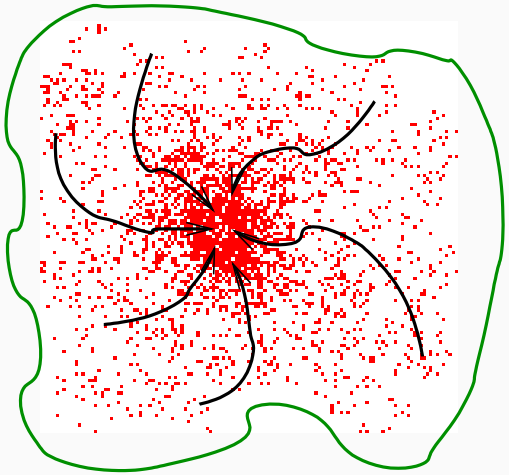


Figure 2: Principle of mean shift analysis: To find the cluster center for point P_1, repeatedly find the centroid of points inside a sphere (initially at P_1) and recenter the sphere on the centroid, until the sphere is stationary. (For Gaussian-kernel mean shift analysis, points further away from sphere centers are given exponentially decreasing weights in the centroid calculation.) This is an adaptive gradient ascent in the space of point densities.

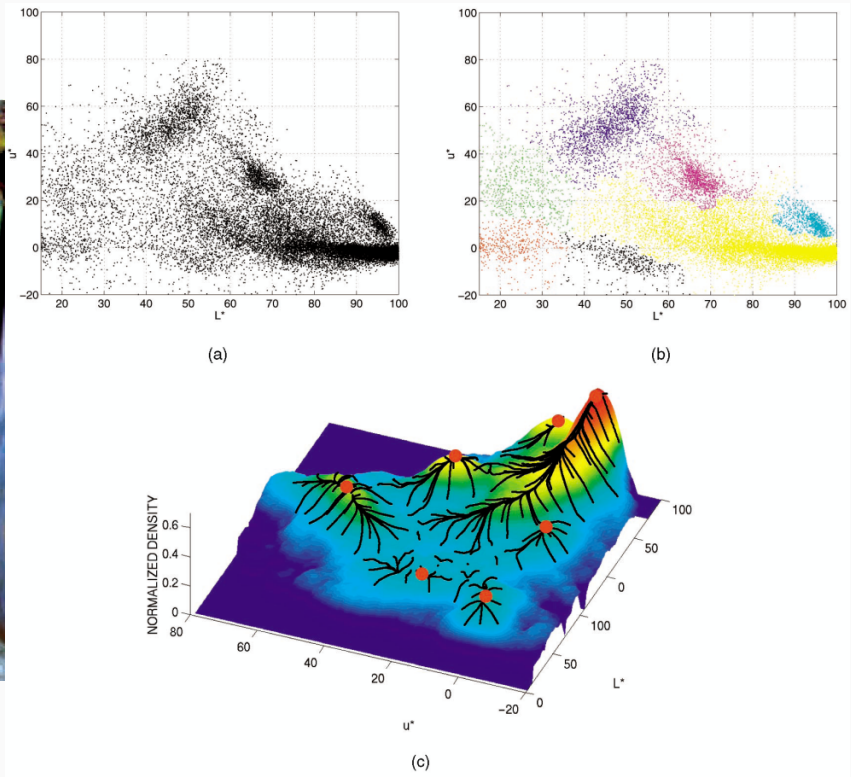
Mean shift clustering

- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode

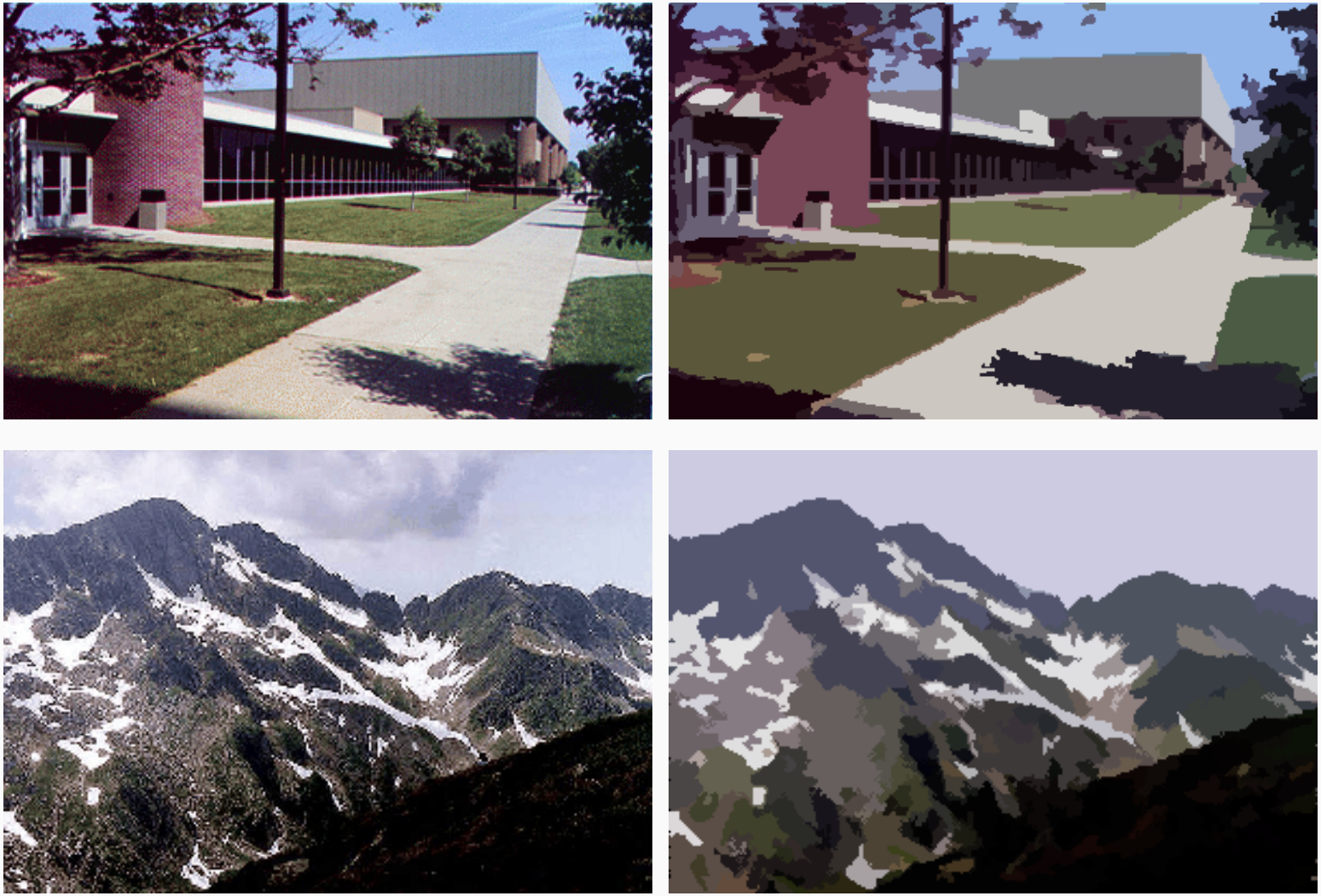


Mean shift clustering/segmentation

- Find features (color, gradients, texture, *etc.*)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same "peak" or mode



Mean shift segmentation results



Mean shift segmentation results

Mean shift pros and cons

- Pros
  - Does not assume spherical clusters
  - Just a single parameter (window size)
  - Finds variable numbers of modes
  - Robust to outliers
- Cons
  - Output depends on window size
  - Computationally expensive
  - Does not scale well with the dimension of feature space

Outline

Introduction

Optimization based methods

Clustering based methods

Region based methods

Region based segmentation

- Two main approaches:
  1. Fusion, region growing: from a pixel or seed region to the whole image
     - Clustering of pixels adjacent to a region according to a given homogeneity criterion
     - Example of criterion: similarity of the targeted pixel color with the average color of the region
  2. Division, region split (from the whole image to the image pixels)

Region based segmentation: principle

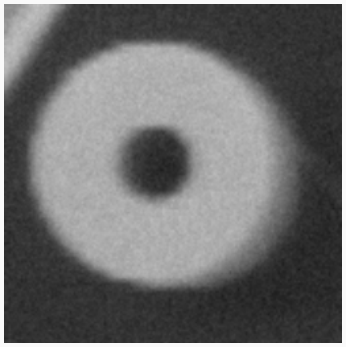
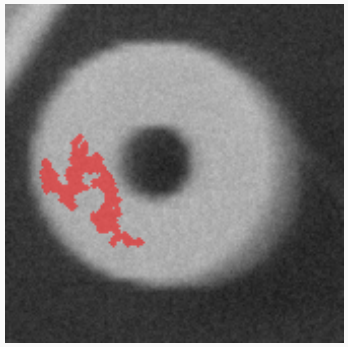
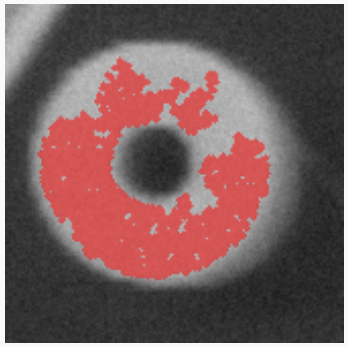
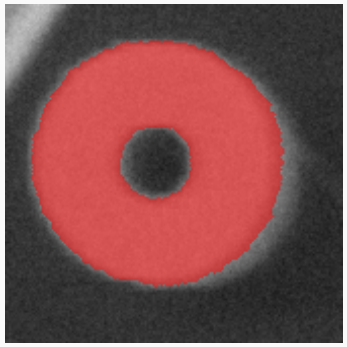
1. Initialization of a region R_0 to a unique pixel or a group of pixels (seed):
   - supervised choice
   - automatic choice (for instance a region of low contrast)
2. Neighbor pixels of R_0 satisfying an homogeneity criterion are added to the region
3. Iterate 2. until convergence

Region based segmentation

- Region growing examples:

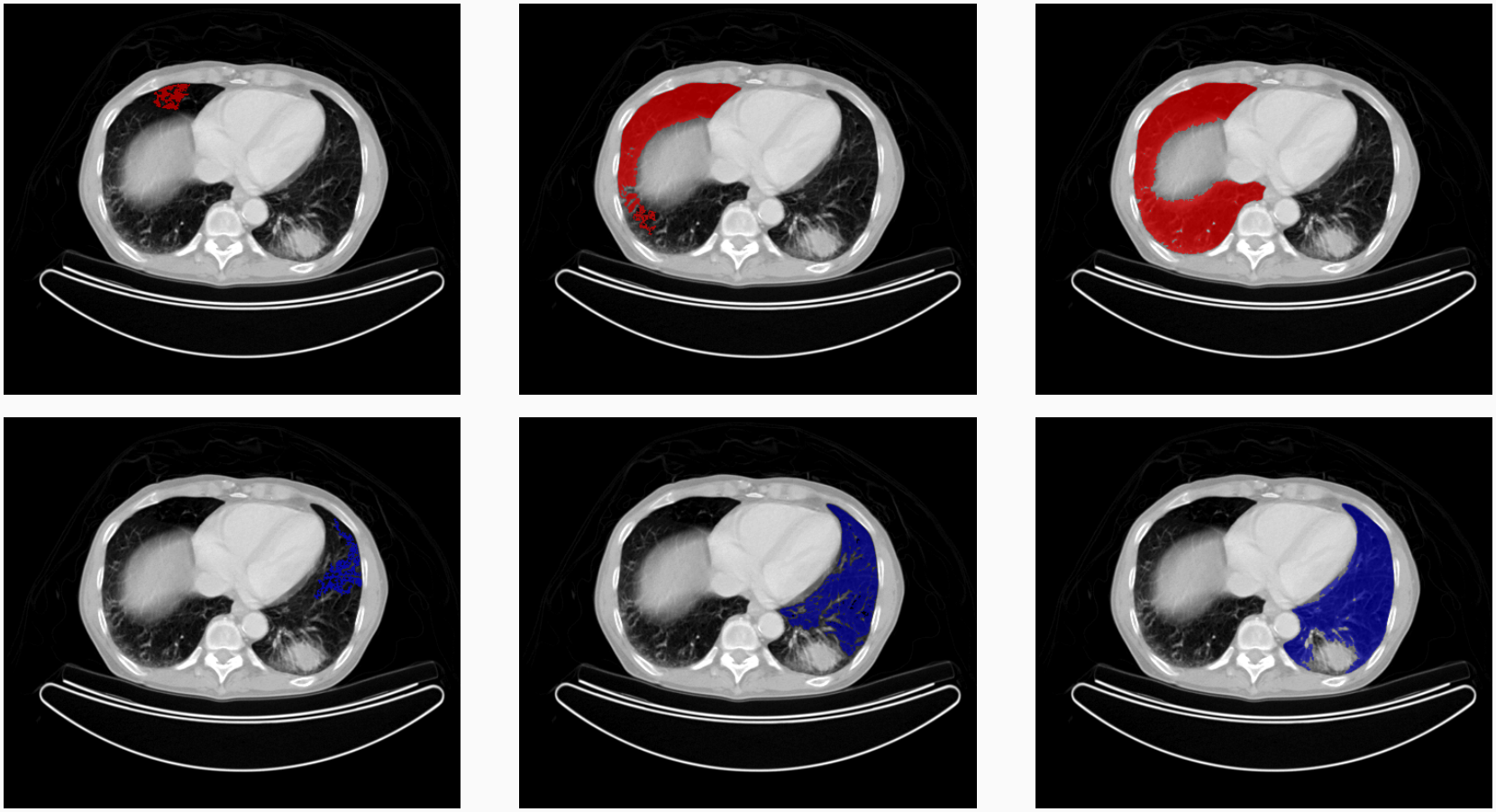



seed growing final region

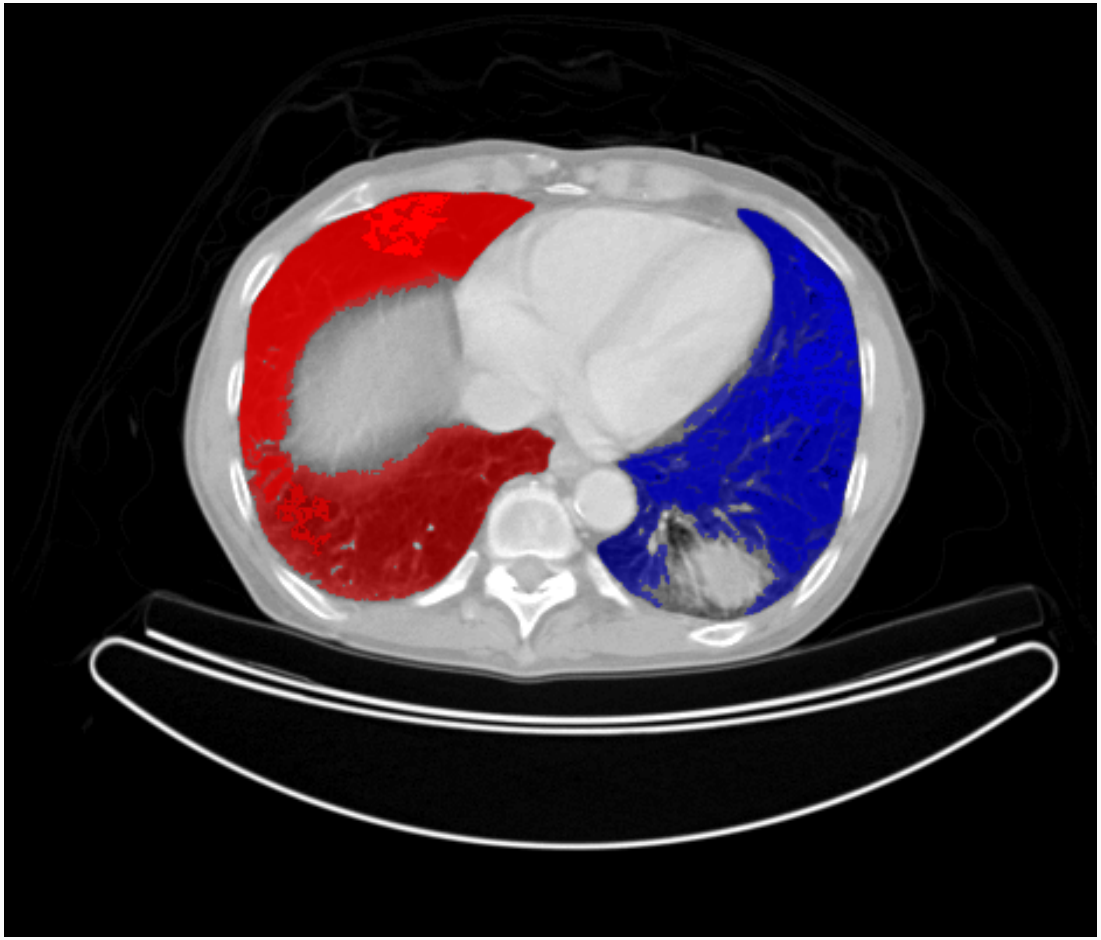
Region based segmentation

- Region growing examples:



Region based segmentation

- Final result

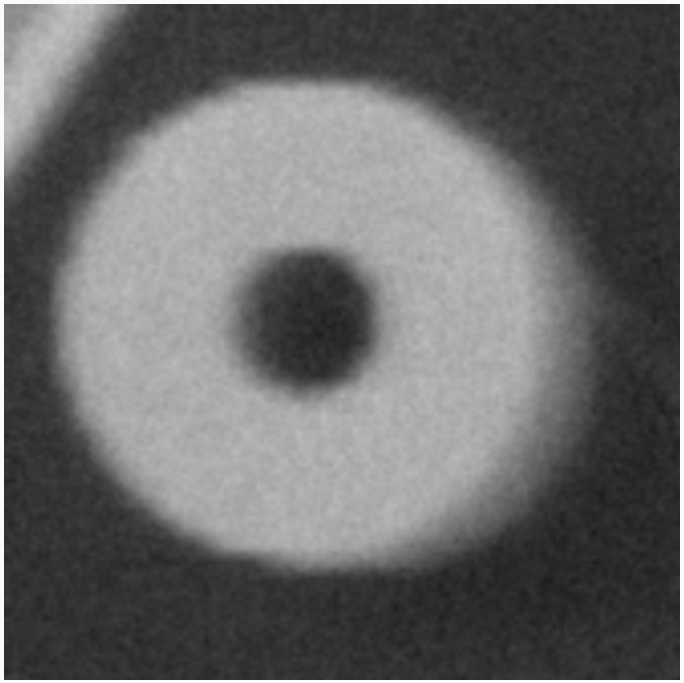


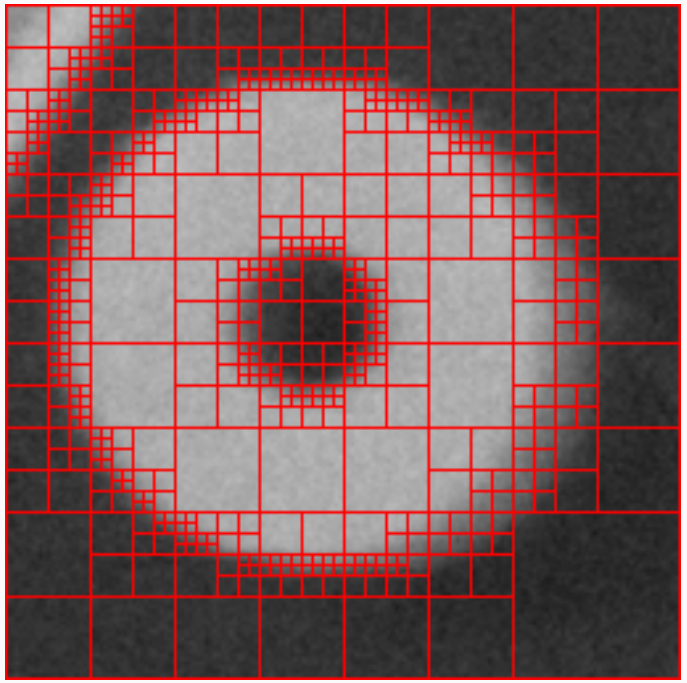
Region based segmentation

- Concluding remarks on region growing:
  - Performances are highly dependent on the initialization (seeds)
  - Highly dependent on the visit order of pixels
  - Easy to code
  - Fast

Region splitting

- Goal of splitting methods: divide the image domain automatically (given an homogeneity criterion) into a set of homogeneous regions



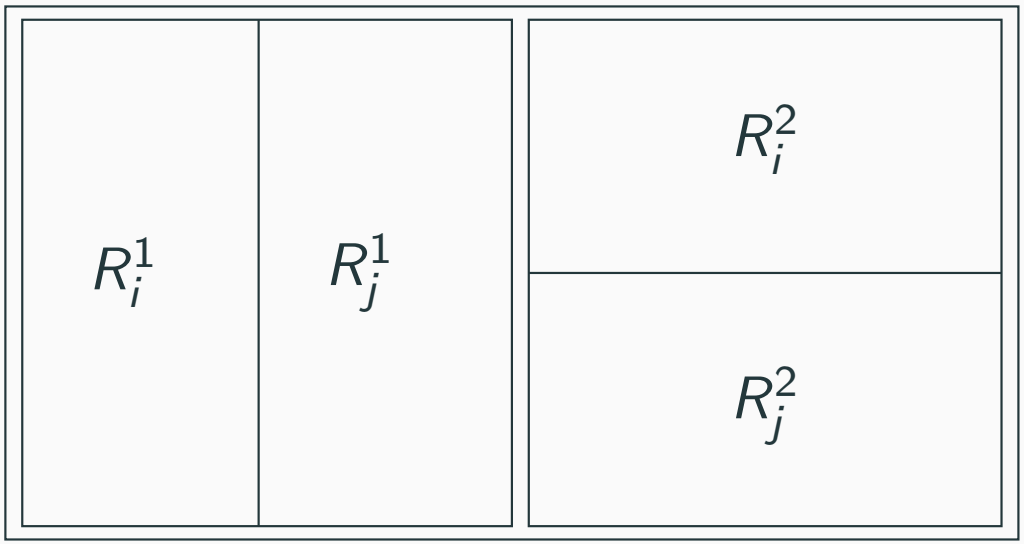


Region splitting – Principle

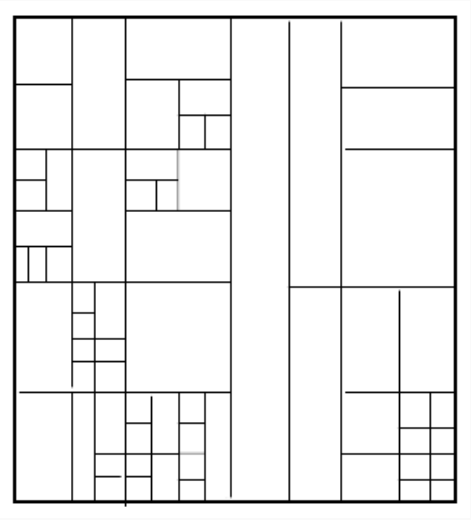
1. Initialization: $R = I$ the whole image
2. Determination of several candidate divisions δ producing new regions R_i^δ
3. Choice of subdivision according to the following process: for each set $\{R_i^\delta\}$ the number of homogeneous sub-regions is counted. The one that provides the most homogeneous regions is retained
4. Back to 2. for all non homogeneous sub-regions

Region based segmentation

- Region splitting
- Example of region splitting with two partitions:



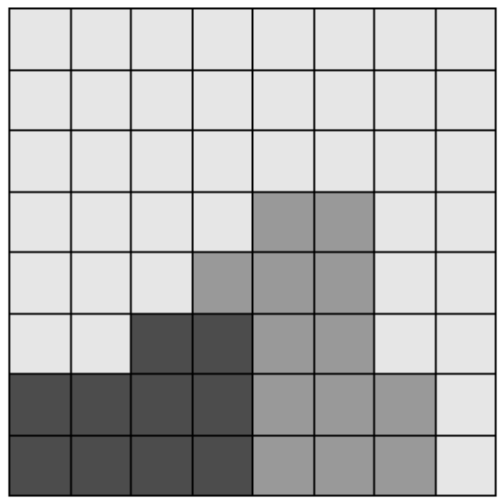
Original image



Partitioned image

Region based segmentation

- Region splitting:
  - Example with a quad-tree partition
  - Here, a unique way to split a region into 4 quadrants
  - Start with the entire image as a unique region corresponding a the root of the quad-tree

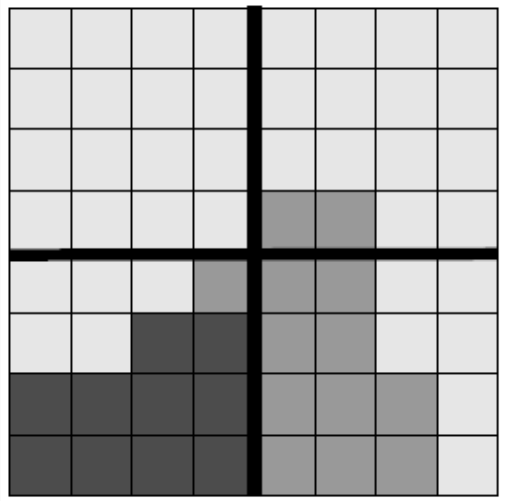


Original image

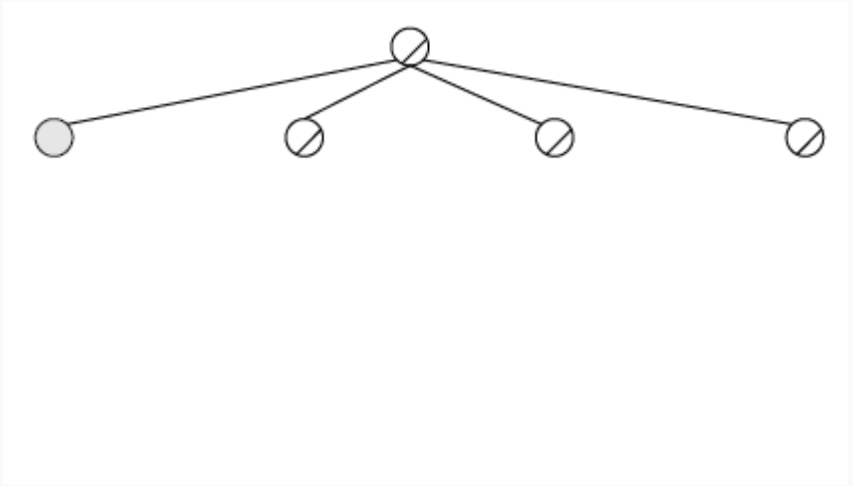
Corresponding quad-tree

Region based segmentation

- Region splitting:
  - Example with a quad-tree partition
  - Here, a unique way to split a region into 4 quadrants
  - Recursively: each non homogeneous region is subdivided into 4 smaller quadrants leading to 4 new leaves F in the quad-tree



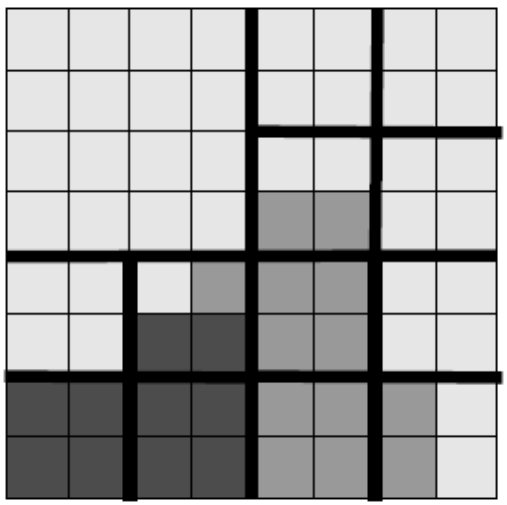
Original image



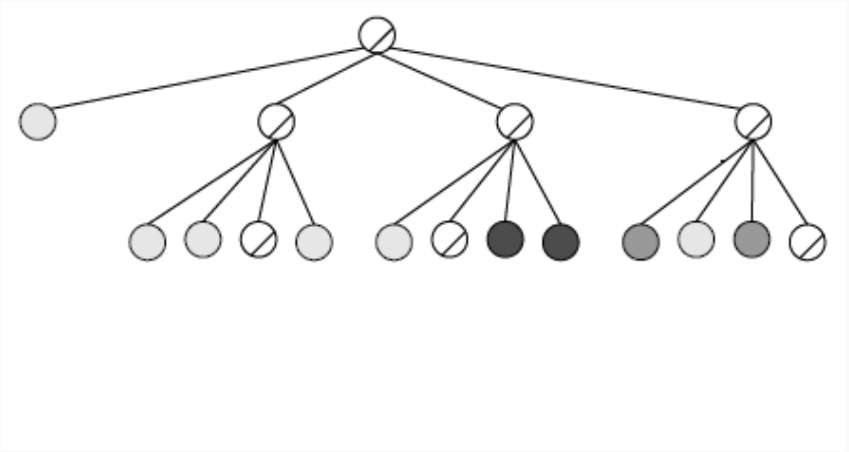
Corresponding quad-tree

Region based segmentation

- Region splitting:
  - Example with a quad-tree partition
  - Here, a unique way to split a region into 4 quadrants
  - Recursively: each non homogeneous region is subdivided into 4 smaller quadrants leading to 4 new leaves F in the quad-tree



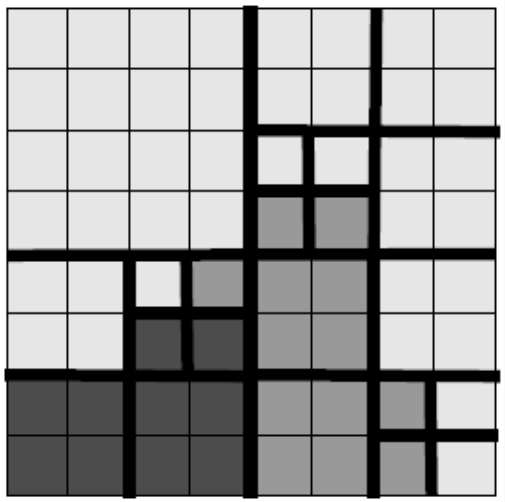
Original image



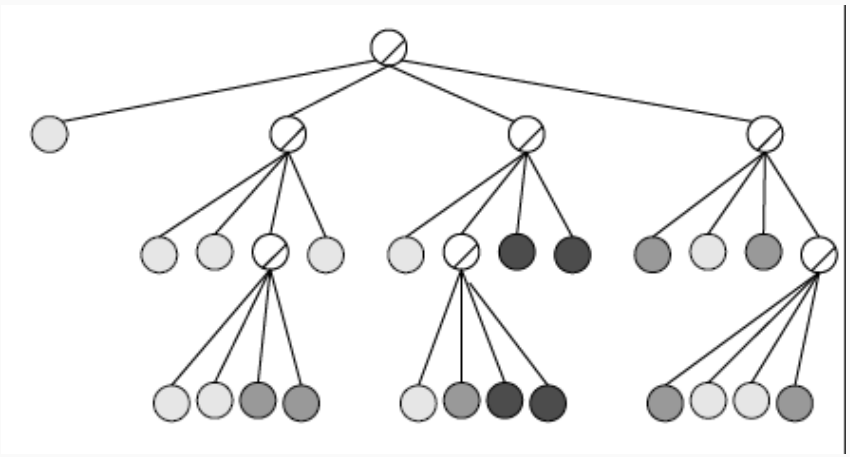
Corresponding quad-tree

Region based segmentation

- Region splitting:
  - Example with a quad-tree partition
  - Here, a unique way to split a region into 4 quadrants
  - The algorithm stops when there are no more non homogeneous regions/leaves to divide



Original image



Corresponding quad-tree

Region spliting – Properties

- The partition geometry has a strong impact on the final segmentation
- For instance, the quad-tree splitting leads to square regions
- Other partitions are possible (triangle, hexagonal...)
- The choice of partition geometry may be driven by the shape of regions to segment in the image
- Not invariant by translation
- Extension to 3-D is easy

Hybrid methods

- How to combine region splitting and region merging?
- Idea: apply a merging algorithm not on pixels but on homogeneous regions detected by a splitting algorithm
- Example: the split and merge algorithm

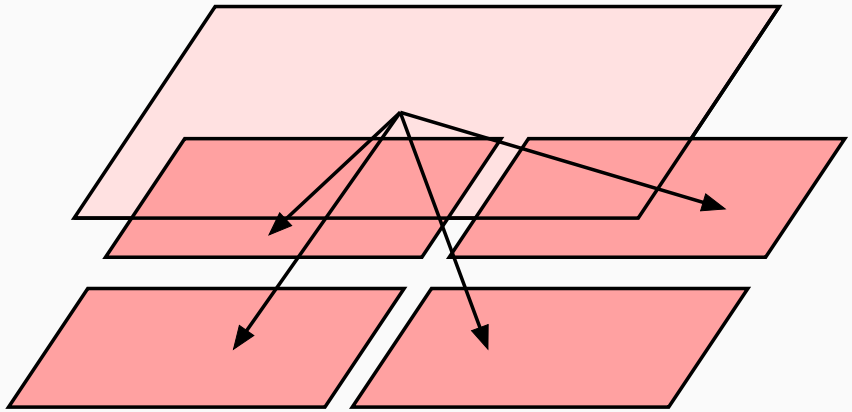
Split and merge

- Algorithm: given an initial segmentation
  1. define a graph of adjacency: each vertex is an homogeneous region and each edge between two vertices indicates that the two corresponding regions are spatially neighbors
  2. define a similarity criterion applying on an edge
  3. sort the list of edges
  4. merge the two best candidates (remove one of the two vertices of the graph and update the edges)
  5. back to 3.

Region based segmentation (Split & merge)

- Split & Merge (Horowitz, Pavlidis (1976), Dubuisson, Jain (1993)) :
  - Stage 1: Create homogeneous regions (split)
  - Stage 2: Merge the homogeneous regions

Split algorithm: a quad-tree



Region based segmentation (Split & merge)

Example with a homogeneity criterion based on low variance

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

initial image

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

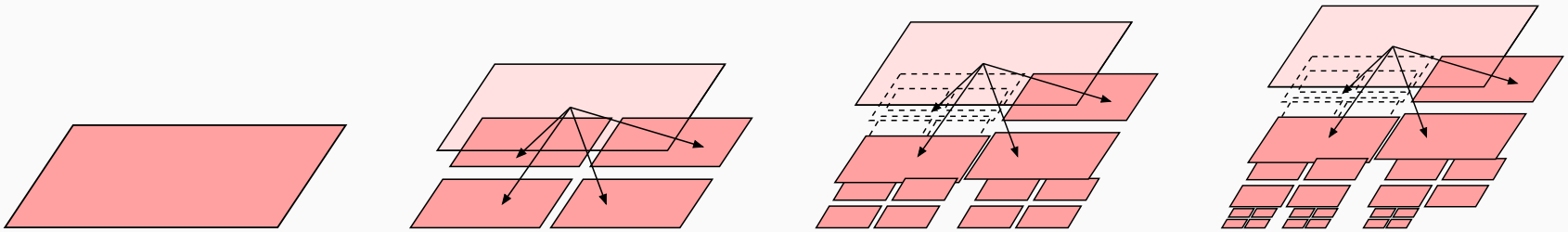
split 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 2

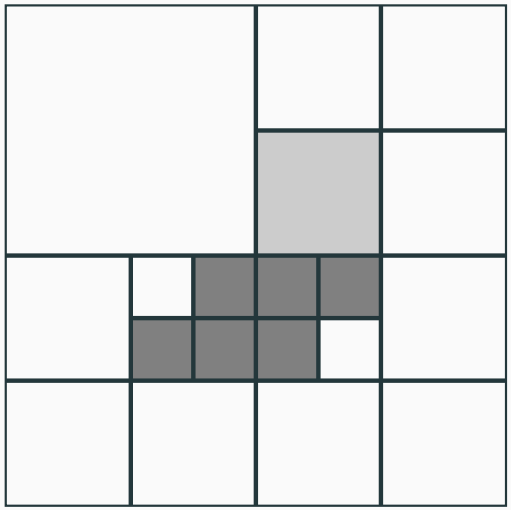
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

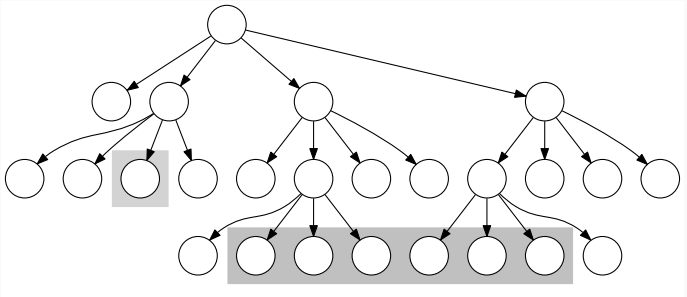
split 3

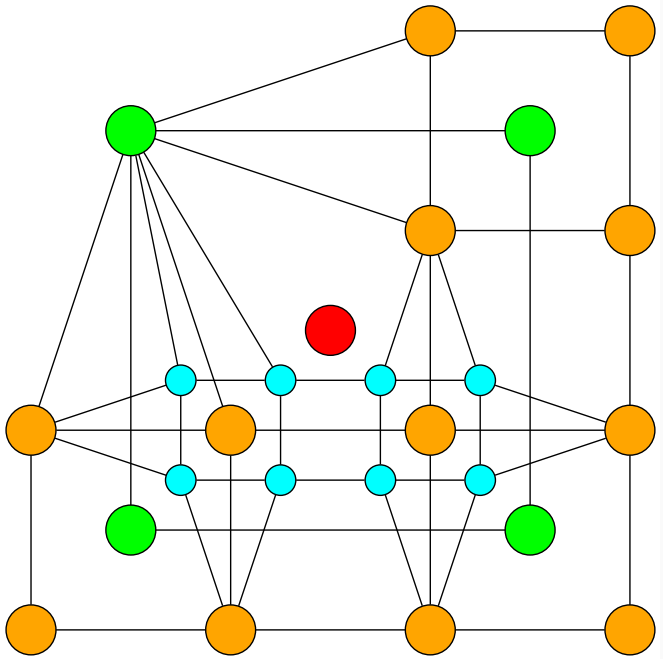


Region based segmentation (Split & merge)

- Determination of the *Region Adjacency Graph* (RAG)
  It may be obtained during the quad-tree recursion
- Edges = carry a measure of homogeneity difference

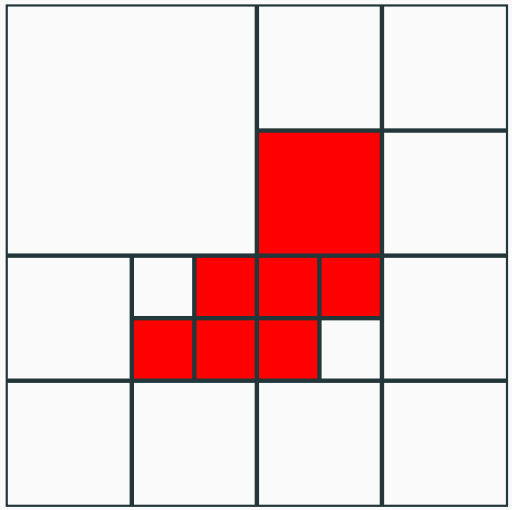


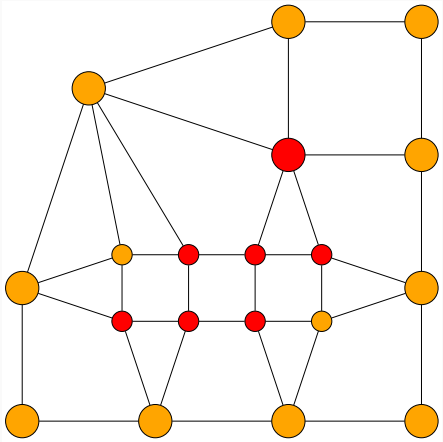




Region based segmentation (Split & merge)

- MERGE:
  - Each vertex of the RAG is examined
  - If one neighbor of this vertex has a low homogeneity difference, the two vertices merge in the RAG
  - When no more merge can happen, the algorithm stops





the distance (in terms of homogeneity) between regions is given by the corresponding edge in the RAG

Region based segmentation (Split & merge)

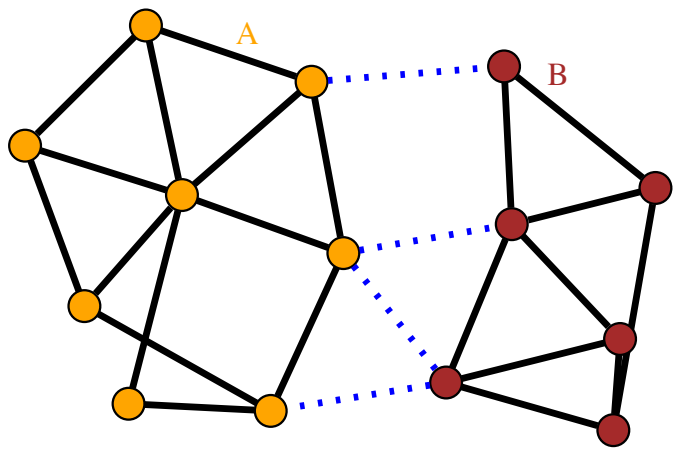
original image

split & merge

Region based segmentation

- Conclusion on graph merging
  - The final result is highly dependent of the visit order of the regions (see step 3. of the split and merge algorithm). Depending on the context, it could be useful to first process the smallest regions.
  - Easy to extend in 3-D
- Other approach: recursive partition of the RAG

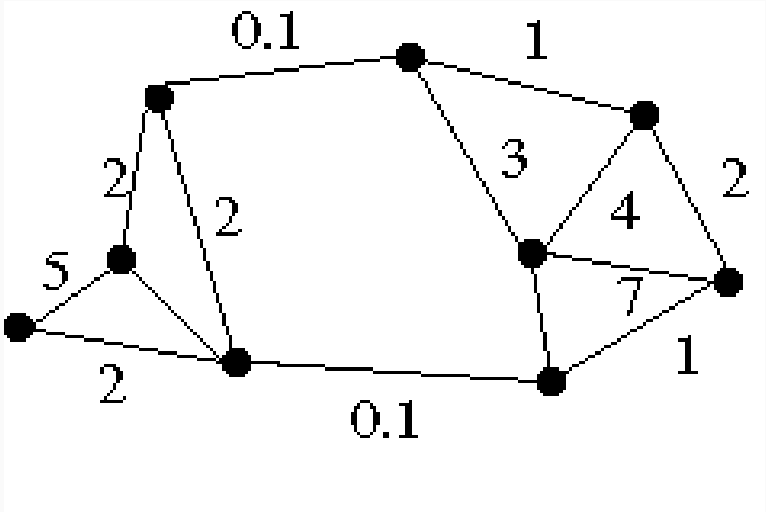
The graph cut algorithm

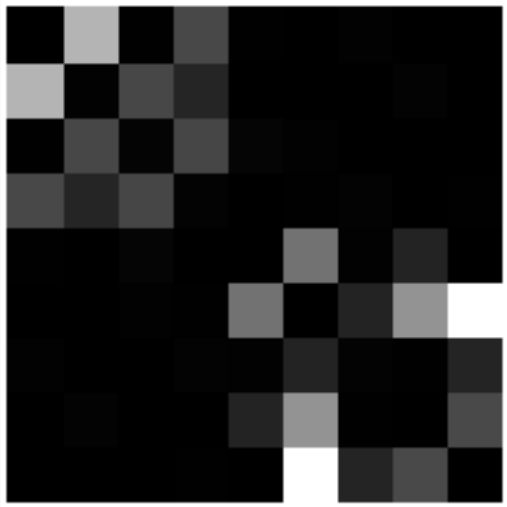


- Find a set of edges that cuts the graph into two sub-graphs A and B
- Cost function: sum of weights of cut edges
- A cut provides a segmentation
- What is a "good" cut and how to find it?

Minimum cut

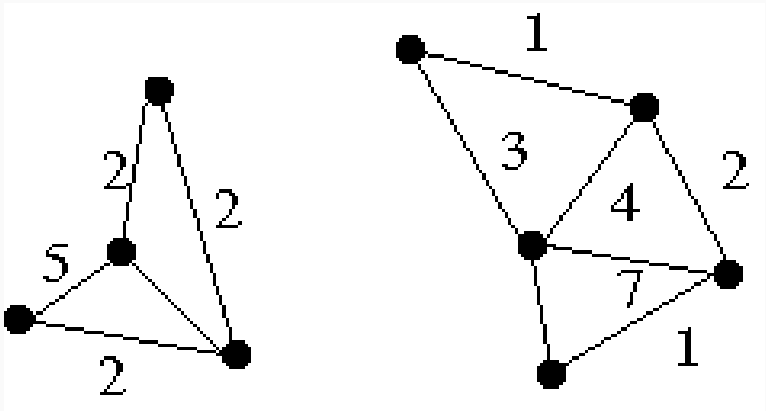
- Segmentation = choose the cut of minimal cost
  - Some efficient algorithms exist

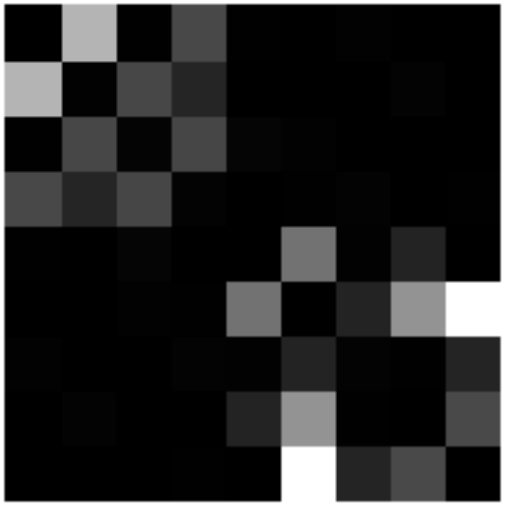




Minimal cut

- Segmentation = choose the cut of minimal cost
  - Some efficient algorithms exist
  - The graph cut can be iterated on sub graphs



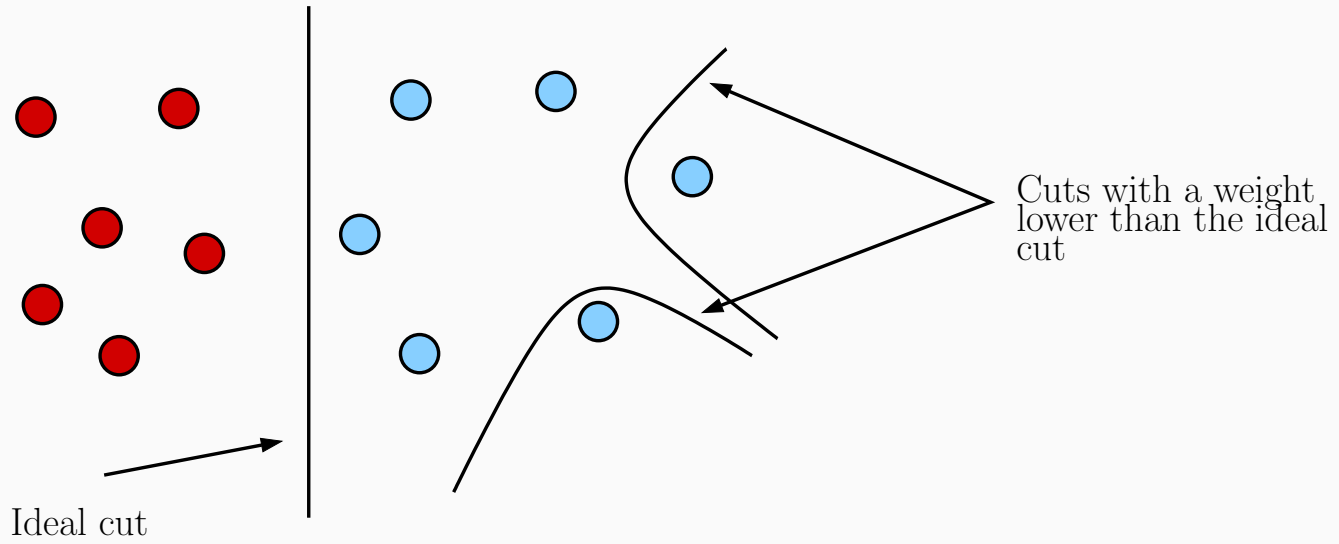


Cost function

- $G = (V, E)$
- $W(u, v)$ matrix of weights between adjacent vertices u and v, the cost of a cut is:

$$W(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

- Issue: the *minimal cut* tends to select cuts with elements of small size



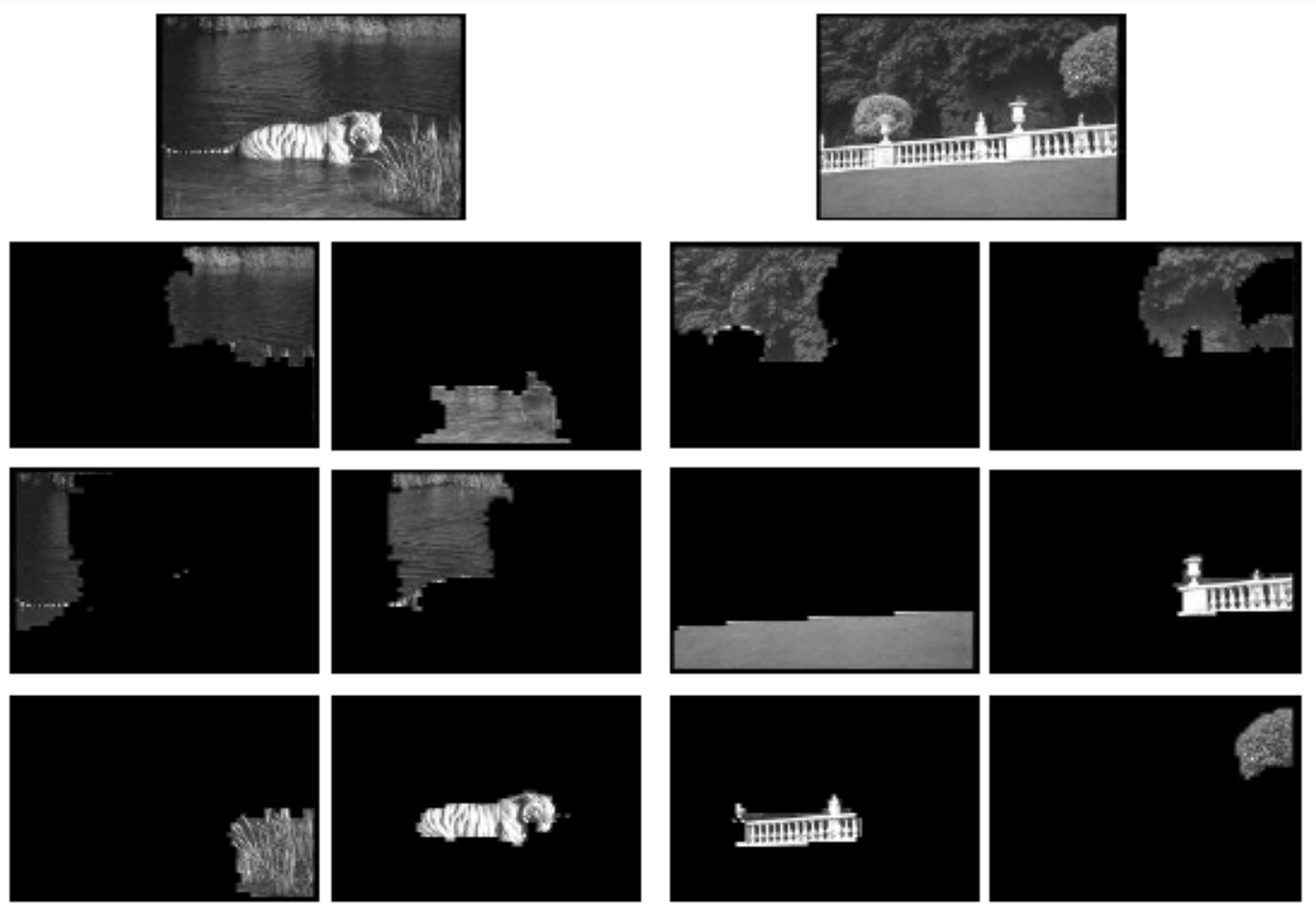
Normalization of cost function

- To fix this issue, one considers a normalized cost N_W :

$$N_W(A, B) = \frac{W(A, B)}{W(A, V)} + \frac{W(A, B)}{W(B, V)}$$

- Numerical implementation uses a matrix diagonalization
- See: J. Shi and J. Malik. *Normalized cuts and image segmentation*. PAMI 2000

Examples



Segmentation: conclusion

- Segmentation is one of the most important tasks in computer vision
- A huge amount of algorithms exist in the literature, this lecture is not an exhaustive review! Other main classes of methods will be presented during the second year the Master IMA:
  - Active contours, geodesic contours and other variational methods
  - Mathematical morphology based methods
  - Markov random fields
  - Multi-scale segmentation
  - and off course, segmentation based on machine learning such as deep convolutional neural networks, SVM, random forest, and so on...
- There nothing is such as one best algorithm: just an ad-hoc solution for a given class of images and problems