2D image segmentation based on spatial coherence

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz
also Center for Machine Perception, http://cmp.felk.cvut.cz

Outline of the lecture:

- Inspiration in Gestalt principles.
- Edge-based segmentation
- Region-based segmentation
- K-means clustering.
- Mean shift segmentation.

Segmentation, initial thoughts revisited

Goals of segmentation

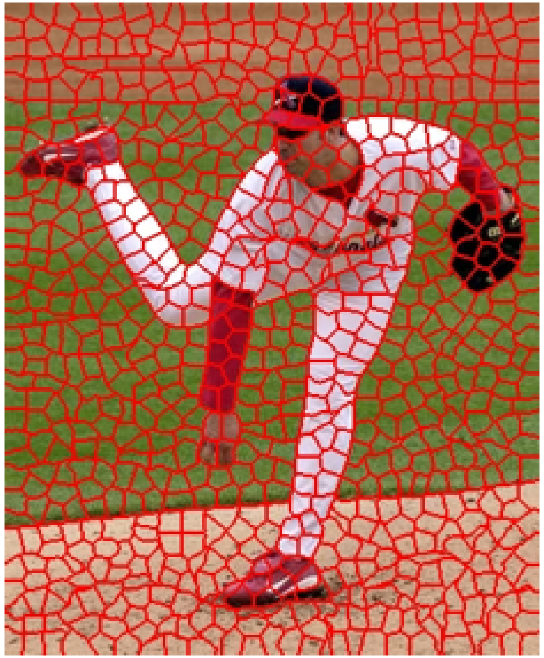
Separate image into coherent “objects”. Top-down or bottom-up process? Supervised or unsupervised?

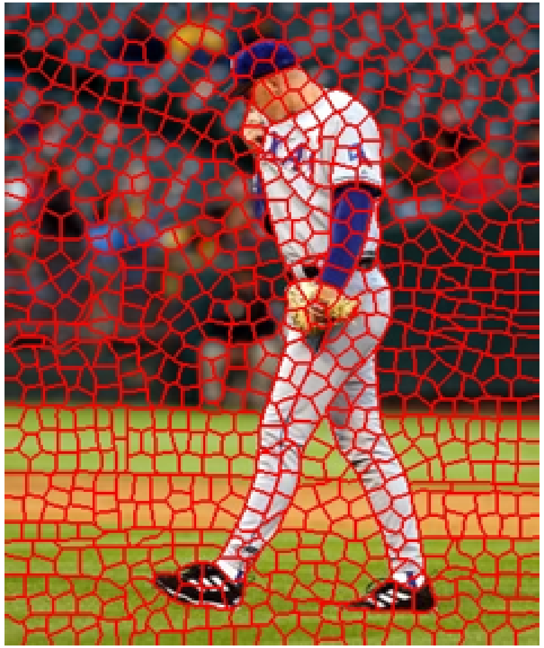
Ideas related to segmentation

- Collect together tokens that ‘belong together’. Group (cluster) them.
- Tokens are detectable entities in the image which can be grouped.
- Top-down segmentation: tokens belong together because they fit to the same object.
- Bottom-up segmentation: tokens belong together because they are locally spatially coherent.
- Top-down and bottom-up approaches are not mutually exclusive. They can often cooperate.

Superpixels for computational efficiency

- Group together similar-looking pixels for efficiency of further processing
- X. Ren and J. Malik. Learning a classification model for segmentation. ICCV 2003.





Gestalt principles of perceptual organization (1)

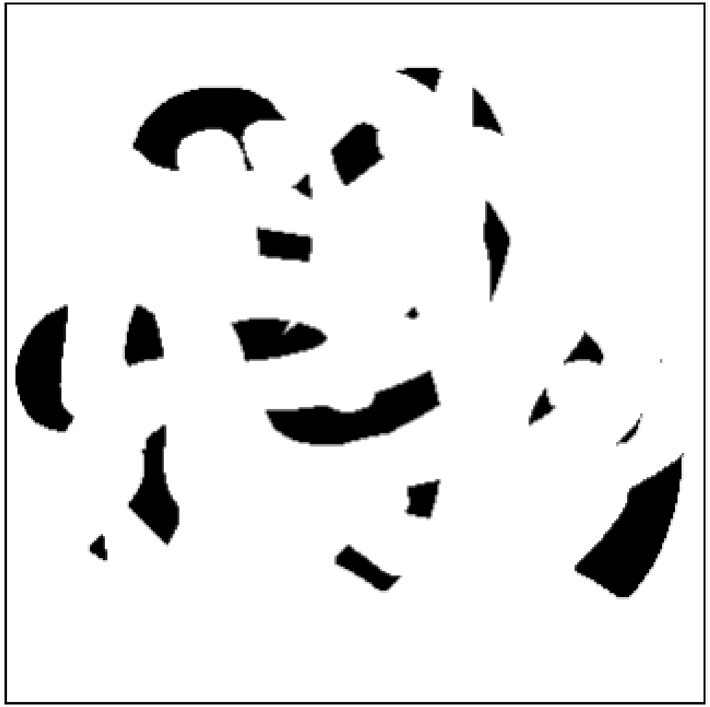
Perceptual organization refers to the mental and physiological steps that group parts of the visual world together to form objects.

- Founding publication by Max Wertheimer (born in Prague) in 1912 in Frankfurt a.M.
- Gestalt theory was meant to have general applicability; its main tenets, however, were induced almost exclusively from observations on visual perception.
- The overriding theme of the theory is that stimulation is perceived in organized or configurational terms (Gestalt in German means “configuration”).
- Patterns take precedence over elements and have properties that are not inherent in the elements themselves.

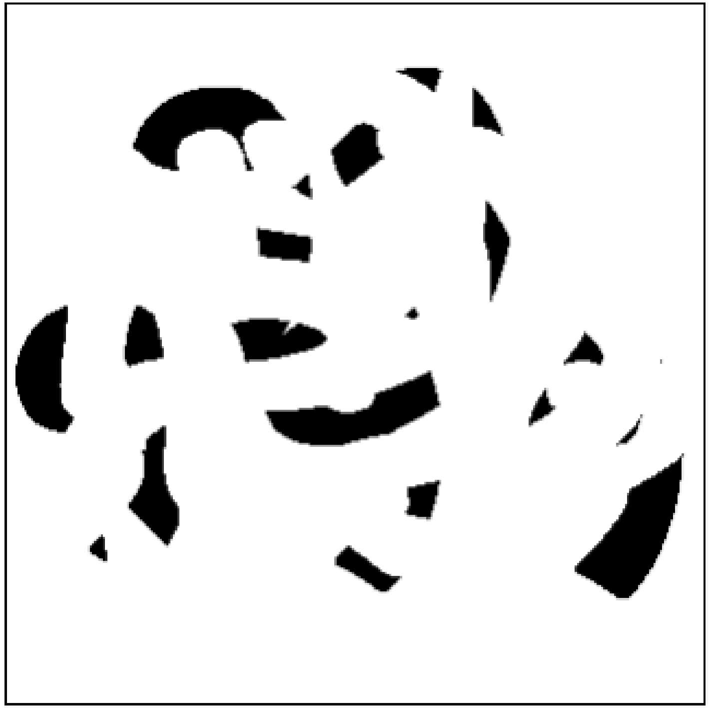
Gestalt principles of perceptual organization (2)

- Gestalt: a structure, configuration, or pattern of physical, biological, or psychological phenomena so integrated as to constitute a functional unit with properties not derivable by summation of its parts.
- "The whole is different from the sum of the parts".
- Rejected structuralism and its assumptions of atomicity and empiricism.
- Adopted a "holistic approach" to perception.

Humans are good in grouping (1)

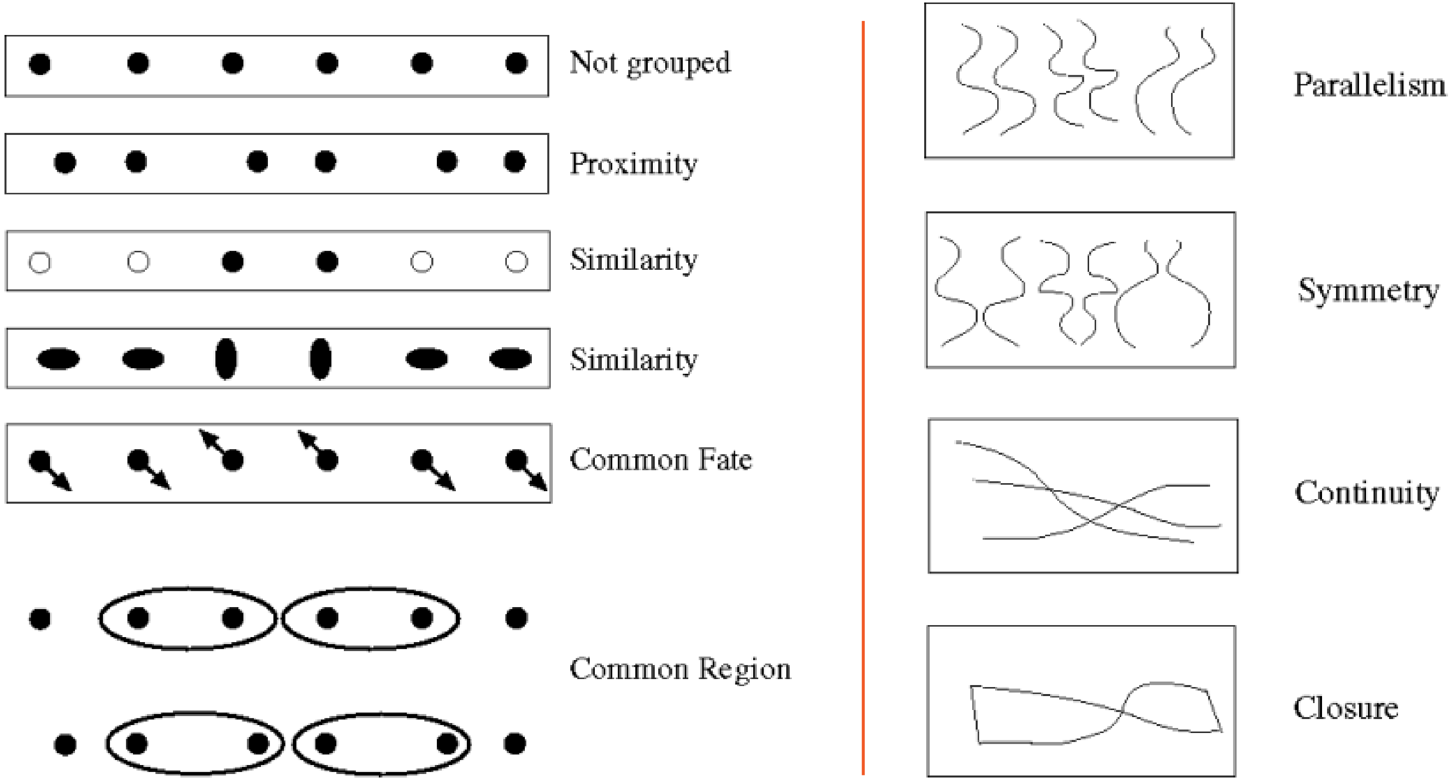


Humans are good in grouping (2)



Gestalt grouping principles

(1) Pragnanz (simplicity); (2) Similarity; (3) Good continuation; (4) Proximity; (5) Common region (e.g. similar brightness, color, contour); (6) Common fate; (7) Familiarity of meaningfulness.



Grouping is not always easy

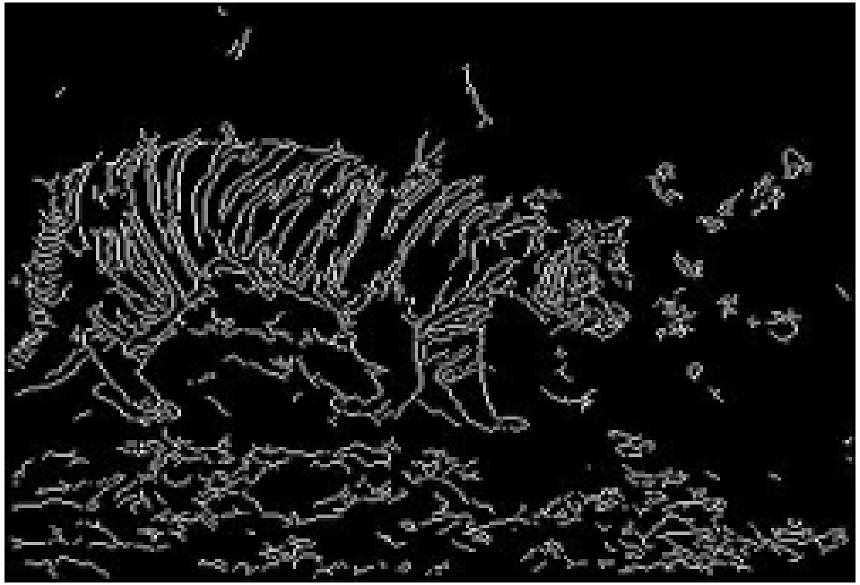
Picture by R. C. James

Contours, boundaries in segmentation

- Physical phenomena involved in the image formation give rise to collections of edgels, often called contours, in images.
  - Closed contours = boundaries of semantically meaningful regions.
  - Open contours = are usually parts of region boundaries with gaps due to noise or low contrast.
  - 'Other contours' can sometimes be found, e.g., corresponding to a surface discontinuity on the surface of a 3D object.
- Contours can be represented as an ordered list of edgels which is often modeled mathematically as a curve.

Edge-based image segmentation

- Edges by a gradient operator. Edgels are significant edges.



- Edgels linking and following by relaxation, voting, dynamic programming, ...
- Natural for grouping of curvilinear structures.
- Often leads to hard premature decisions.

Edge-based image segmentation (2)

- Edge detector is applied to find region boundary candidates.
- Thresholding of the edge magnitude is the simplest way how to get boundary candidates.
- Some iterative (possibly knowledge-based technique) is used to find boundaries.

Problems with edge-based segmentation:

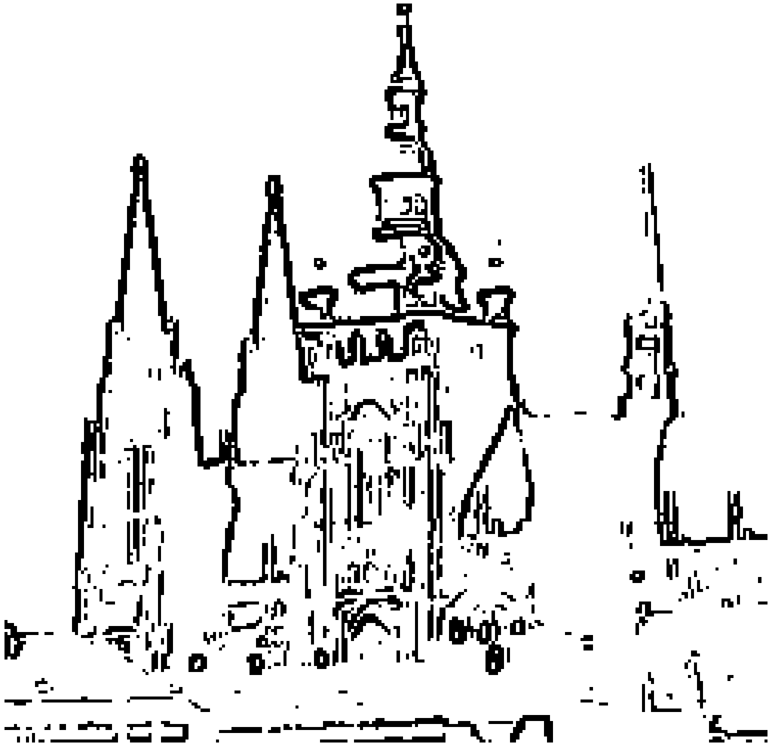
- It is difficult to obtain closed contours, i.e., boundaries of regions.
- The whole image need not be segmented.

Edge image thresholding (1)

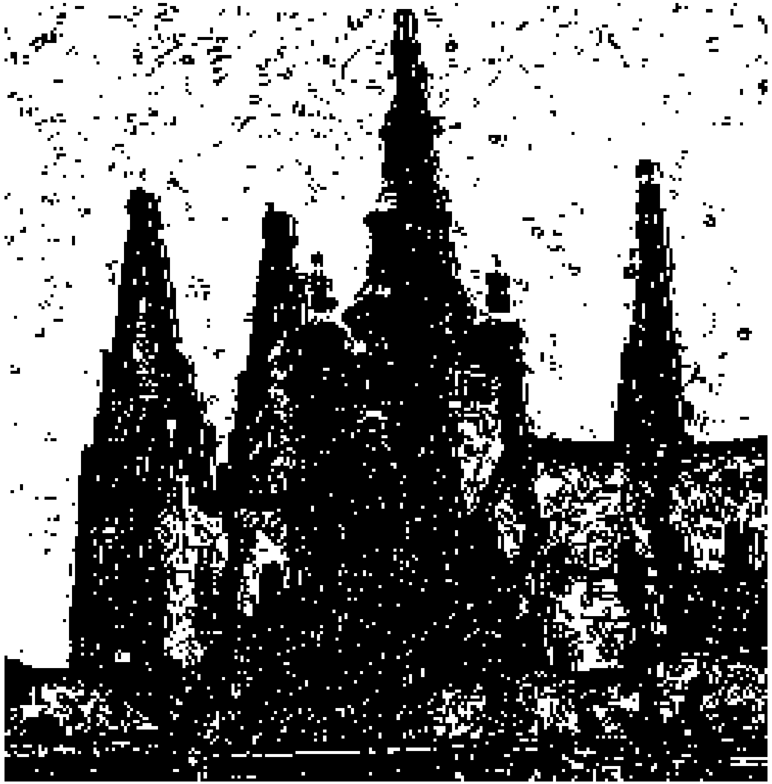
original image

edge image
(enhanced for display)

Edge image thresholding (2)

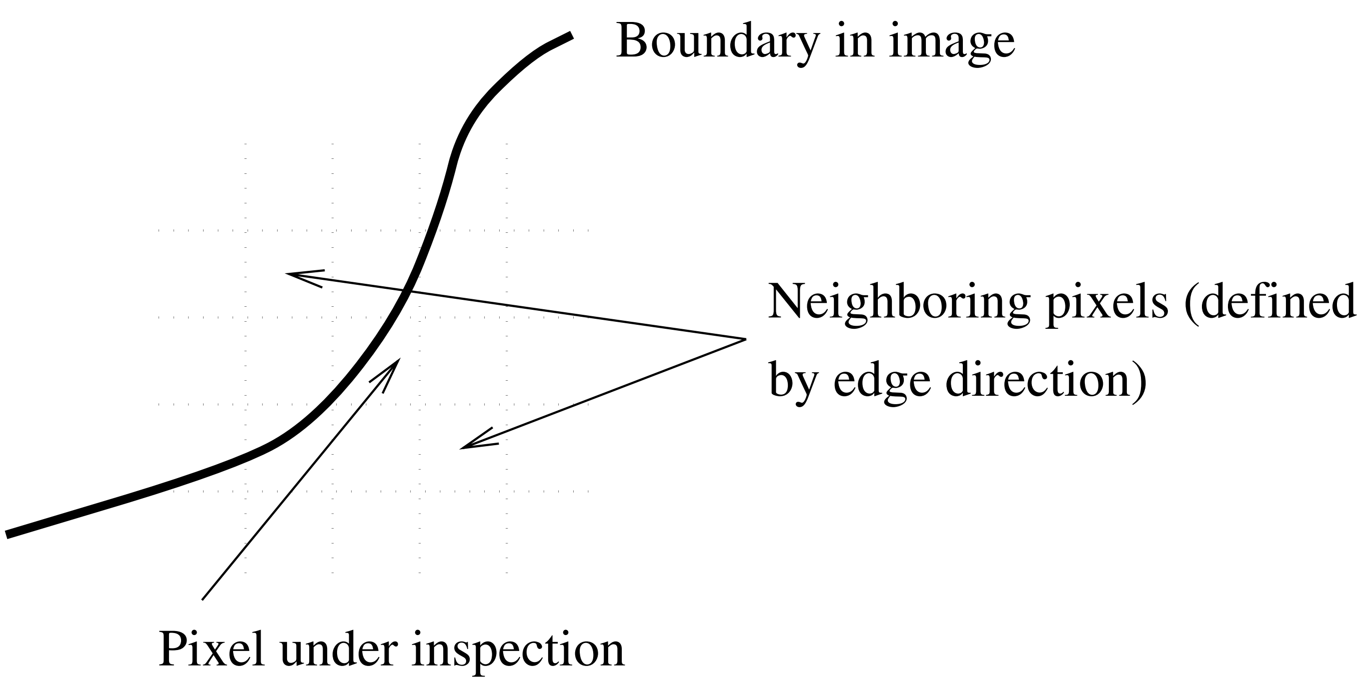


threshold 30



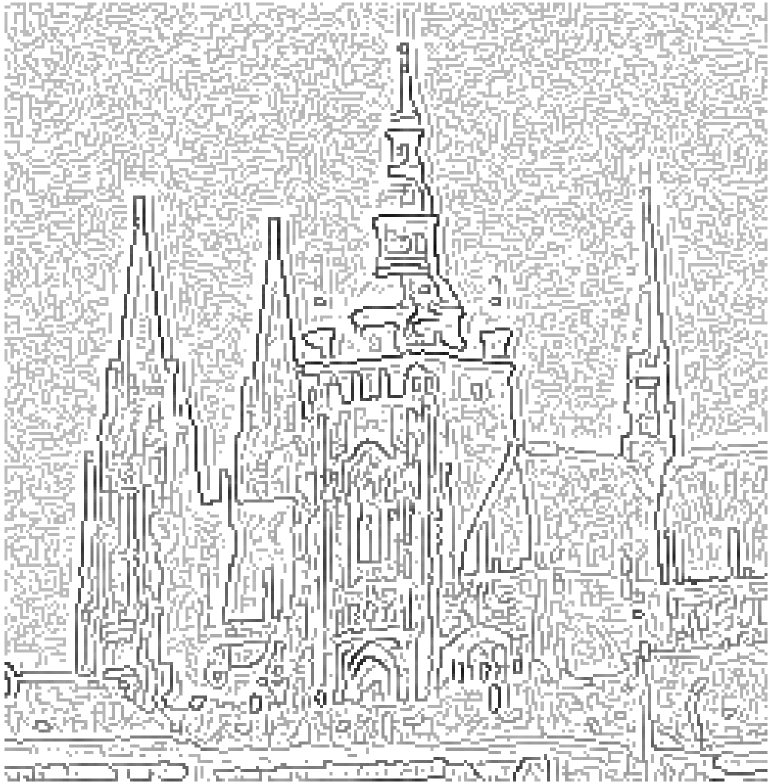
threshold 10 (too small)

Non-maximal suppression and hysteresis

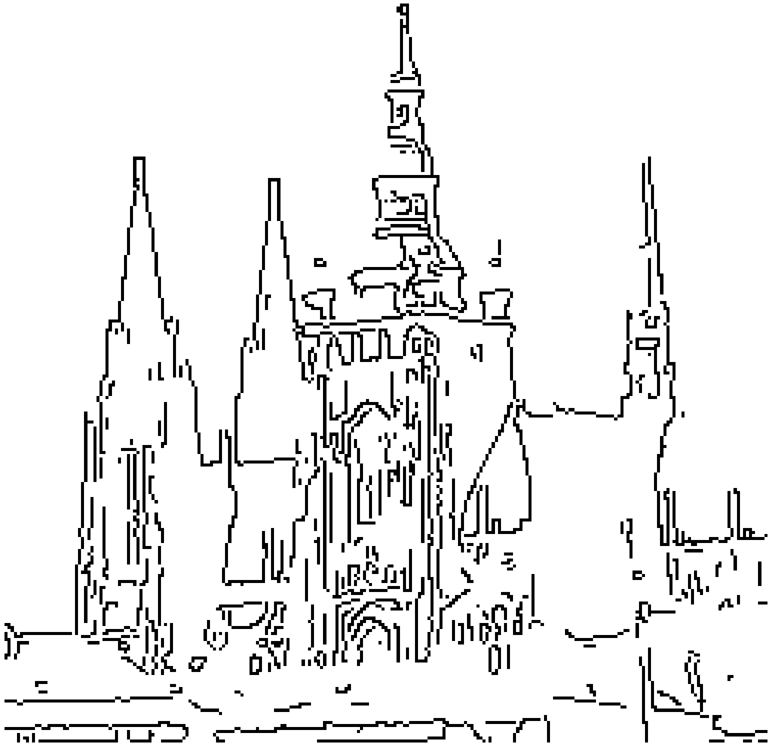


Pixels adjacent with respect to local edge information are inspected.

Non-maximal suppression and hysteresis (2)



Non-maximal suppression



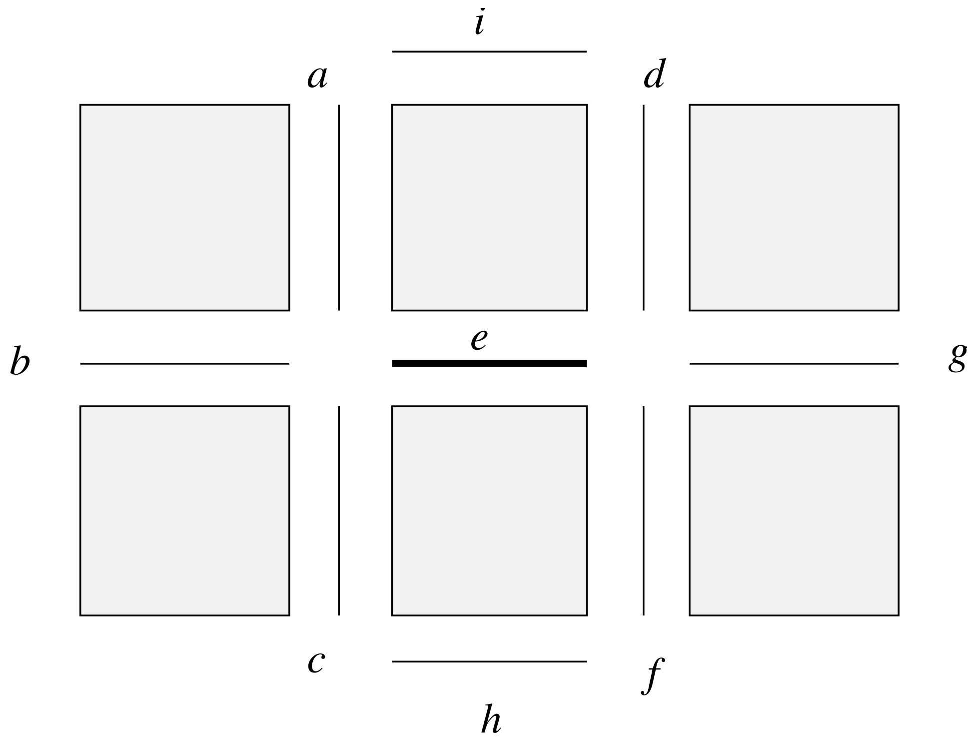
hysteresis
high threshold 70, lower 10

Edge relaxation

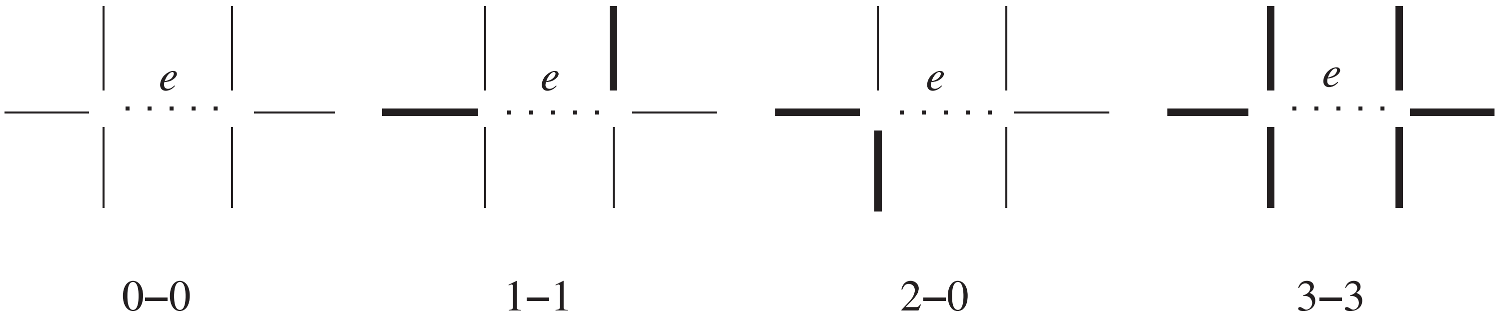
- Edge-based segmentation does not provide close borders as some parts of it are missing due to noise.
- Edge relaxation is a postprocessing step which closes gaps between border segments.
- It is an instance of a general algorithm called relaxation labelling.
- In edge relaxation, edge properties in the context of neighboring pixels are iteratively improved.
- The method will be illustrated on one example: relaxation based on crack edges (Hanson, Rieseman 1978).

Crack edge

- Introduces 'superresolution' inter-pixel property.
- There is a formal notion how to express it = cellular complexes.



Crack edge patterns

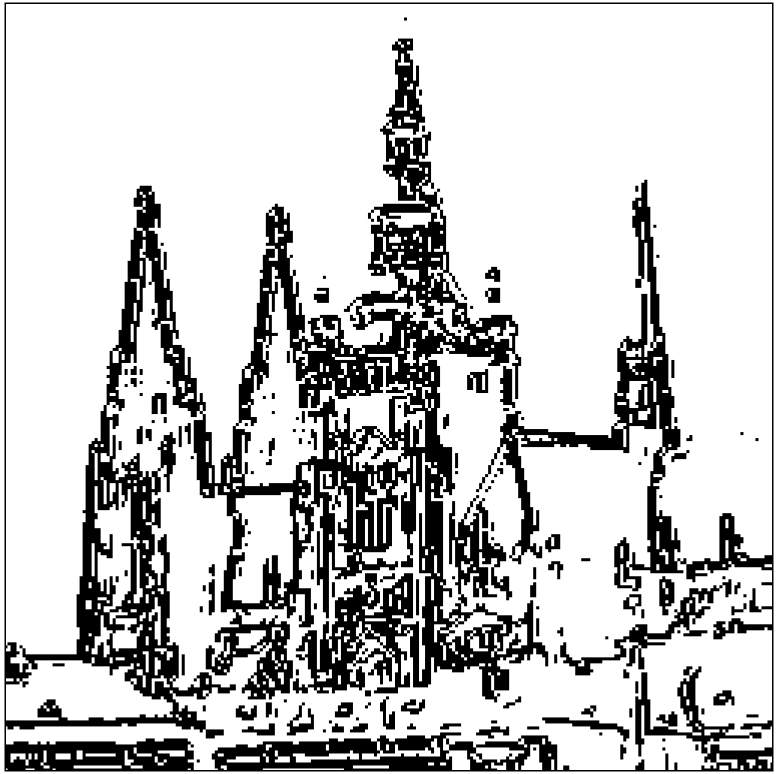


- **0-0** isolated edge – negative influence on the edge confidence
- **0-1** uncertain – weak positive, or no influence on edge confidence
- **0-2, 0-3** dead end – negative influence on edge confidence
- **1-1** continuation – strong positive influence on edge confidence
- **1-2, 1-3** continuation to border intersection – medium positive influence on edge confidence
- **2-2, 2-3, 3-3** bridge between borders – not necessary for segmentation, no influence on edge confidence

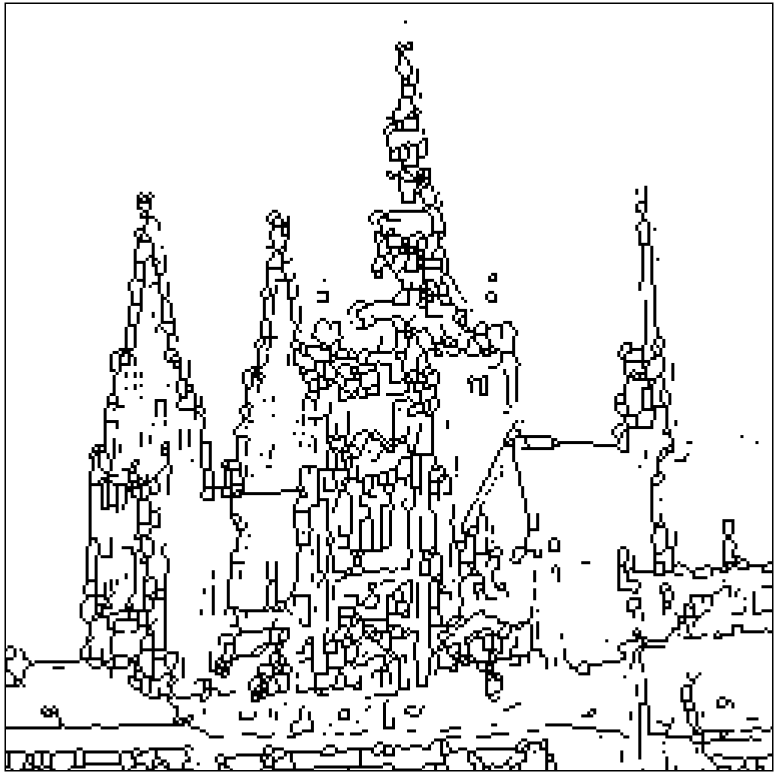
Edge relaxation algorithm

1. Evaluate a confidence $c^{(1)}(e)$ for all crack edges e in the image.
2. Find the edge type of each edge based on edge confidences $c^{(k)}(e)$ in its neighborhood.
3. Update the confidence $c^{(k+1)}(e)$ of each edge e according to its type and its previous confidence $c^{(k)}(e)$.
4. Stop if all edge confidences have converged either to 0 or 1. Repeat steps (2) and (3) otherwise.

Edge relaxation, example (1)

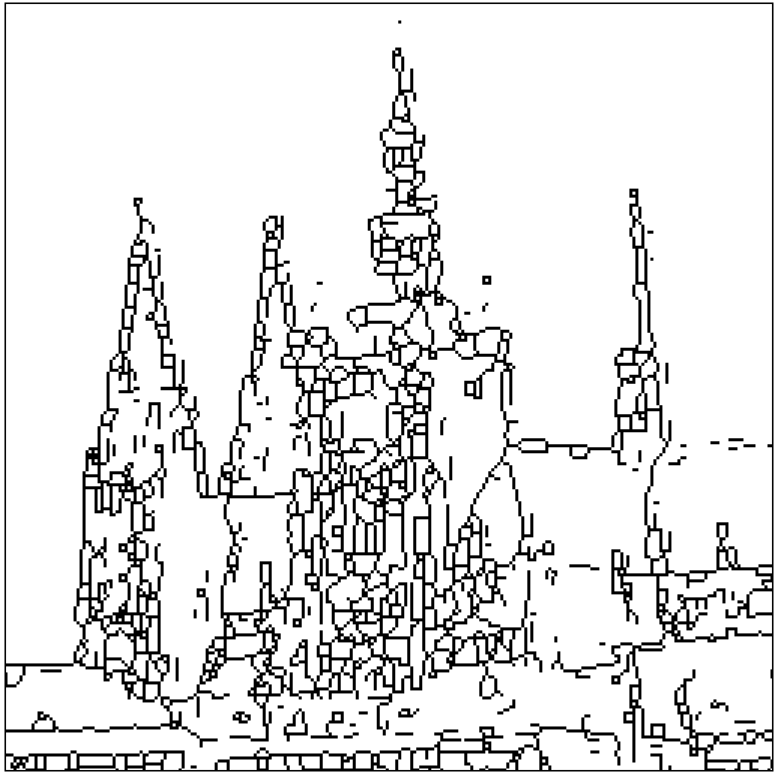


borders after 10 iterations



borders after thinning

Edge relaxation, example (2)



borders after 100 iterations
thinned

overlaid over original

Segmentation as clustering

- Image segmentation can be formulated as clustering which has been studied in statistics, pattern recognition or machine learning.
- Problem formulation: Assign a pixel to a cluster which is the most spatially adjacent and the most most homogeneous with respect to some criterion (e.g., brightness variance).
- There are several clustering methods. Seek unsupervised learning algorithms in pattern recognition. E.g., K-means, EM.

K-means clustering

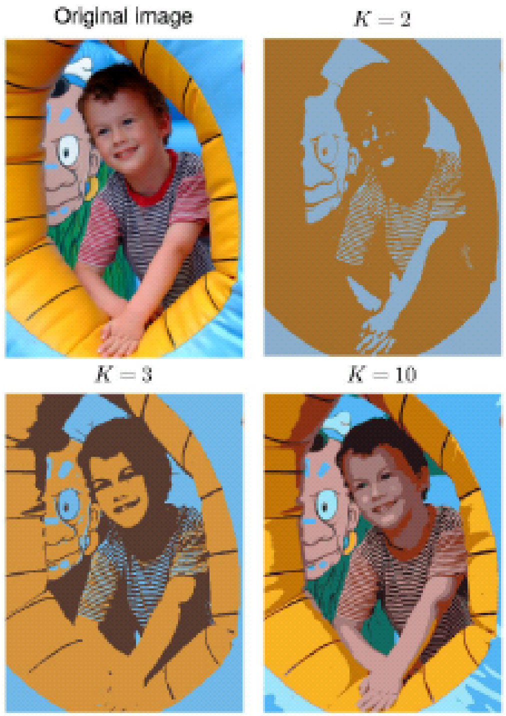
Lloyd's Algorithm

- Initialize the pixels to belong to a random region.
- Compute the mean feature vector in each region.
- Move a pixel to another region if this decreases the error function ($=$ total distance) J.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \left\| x_n - \mu_k \right\|^2 ,$$

  where n points to individual pixels, N is number of pixels, K is an a priori given number of clusters, $K < N$, x_n is a pixel value, μ_k is the cluster representative (mean point of a cluster).

- Iterate until pixels do not move any longer.



Author: Christopher Bishop

S. Lloyd, Last square quantization in PCM's. Bell Telephone Laboratories Paper (1957). In the journal much later: S. P. Lloyd. Least squares quantization in PCM. Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982.

K-means clustering (2)

- Converges to a local minimum of the error function (total distance) J.
- There are point sets, on which K-means takes superpolynomial time $\mathcal{O}(2^{\sqrt{n}})$ to converge.
  D. Arthur, S. Vassilvitskii (2006). How Slow is the k-means Method?. Proceedings of the 2006 Symposium on Computational Geometry.
- With $K = 2$ the K-means algorithm can be regarded as a thresholding with an automatically determined threshold.
- Pros: (a) Simple to implement; (b) Many implmentations available, e.g. in MATLAB, Python.
- Cons: (a) There is a need to know (or pick) the number of clusters K; (b) Sensitive to the initial choice of the clusters.; (c) Prefers spherical clusters; (d) Sensitive to outliers.

K-means clustering, a texture example

Image, courtesy Ondoej Drbohlav

- Feature used for clustering – the absolute value of 1st partial derivatives,

$$\left(\left| \frac{\partial I(x,y)}{\partial x} \right|, \left| \frac{\partial I(x,y)}{\partial y} \right| \right)$$

- Two clusters, $K = 2$.

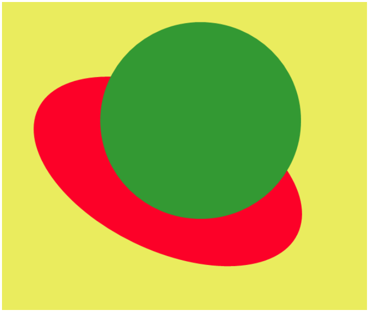
Segmentation by region growing/splitting

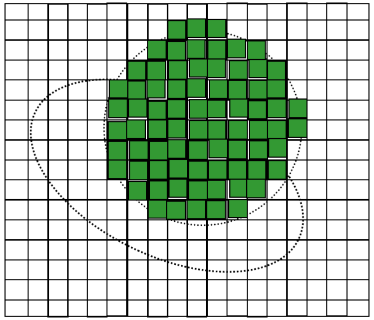
Region growing algorithm

1. Start with region R given by the seed K (one or several adjacent pixels).

2. For pixels $p \in K$ add their neighbors q into the region R provided they fulfil the similarity criterion between q and R.

3. Repeat step 2 until nothing changes.





Mean shift

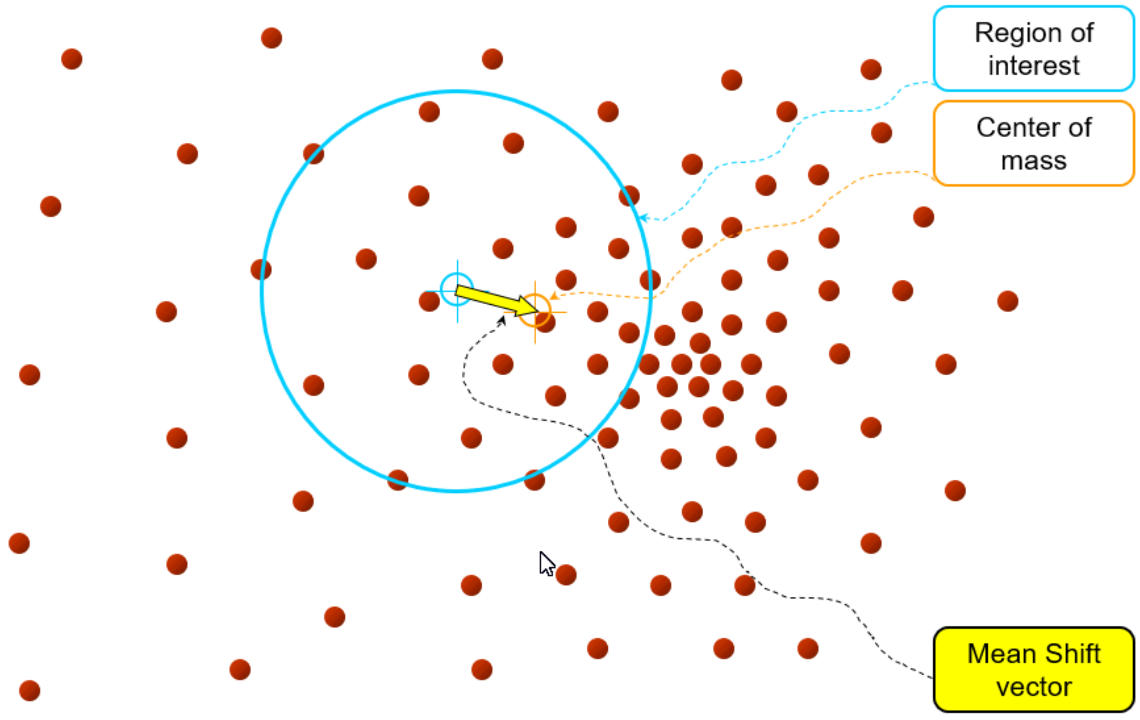
- Estimation of the density gradient - Fukunaga K.: Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.
- Sample mean of local samples points in the direction of higher density. It provides the estimate of the gradient.
- Mean shift vector m of each point p

$$m = \sum_{i \in \text{window}} w_i \left(p_i - p \right) , \quad w_i = \text{dist}(p, p_i)$$

- Based on the assumption that points are more and more dense as we are getting near the cluster "central mass".

Mean shift algorithm (1)

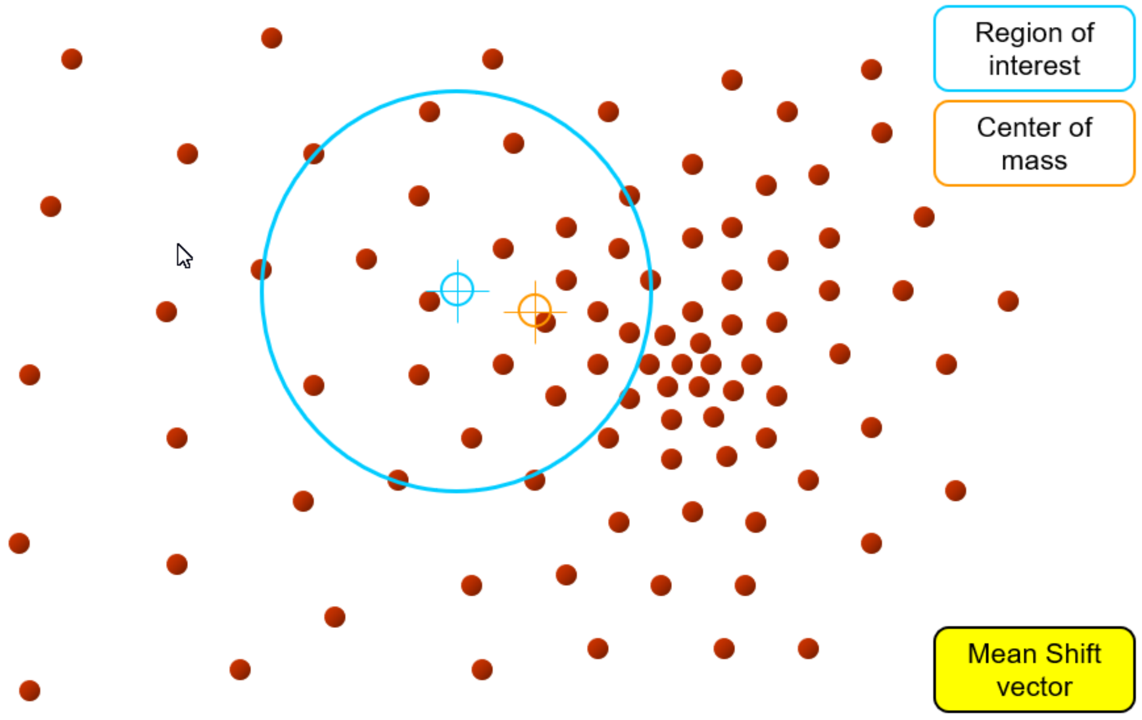
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

Mean shift algorithm (2)

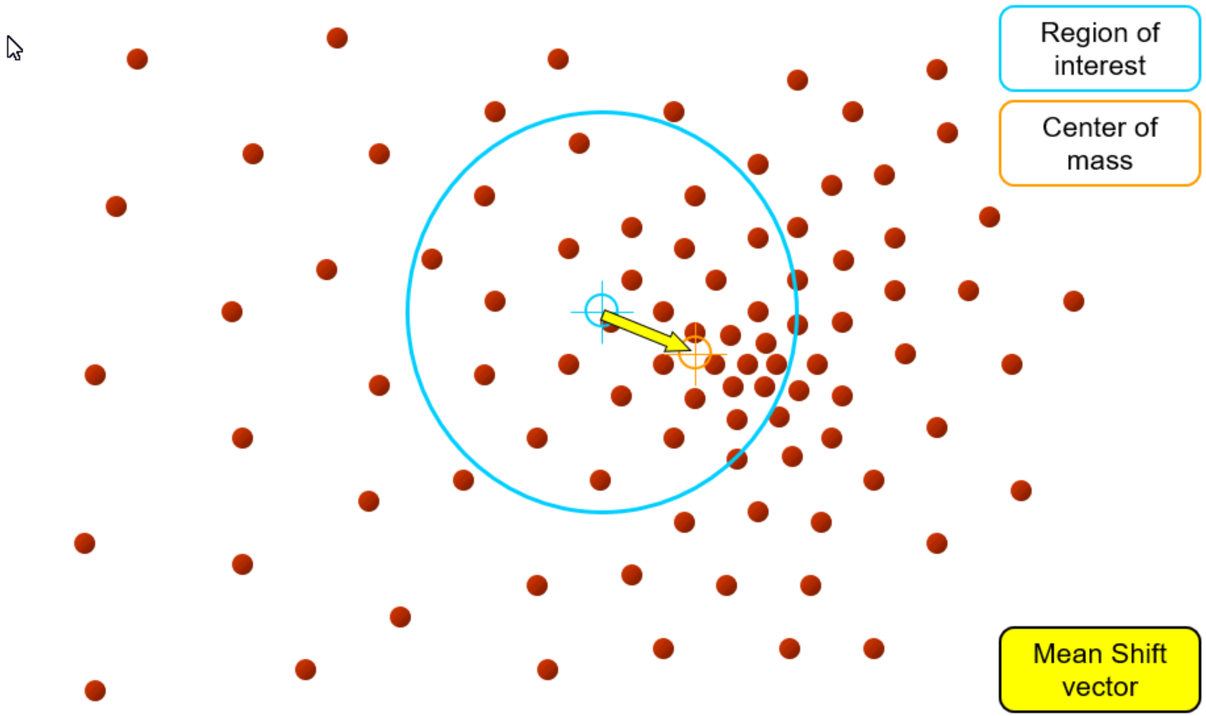
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

Mean shift algorithm (3)

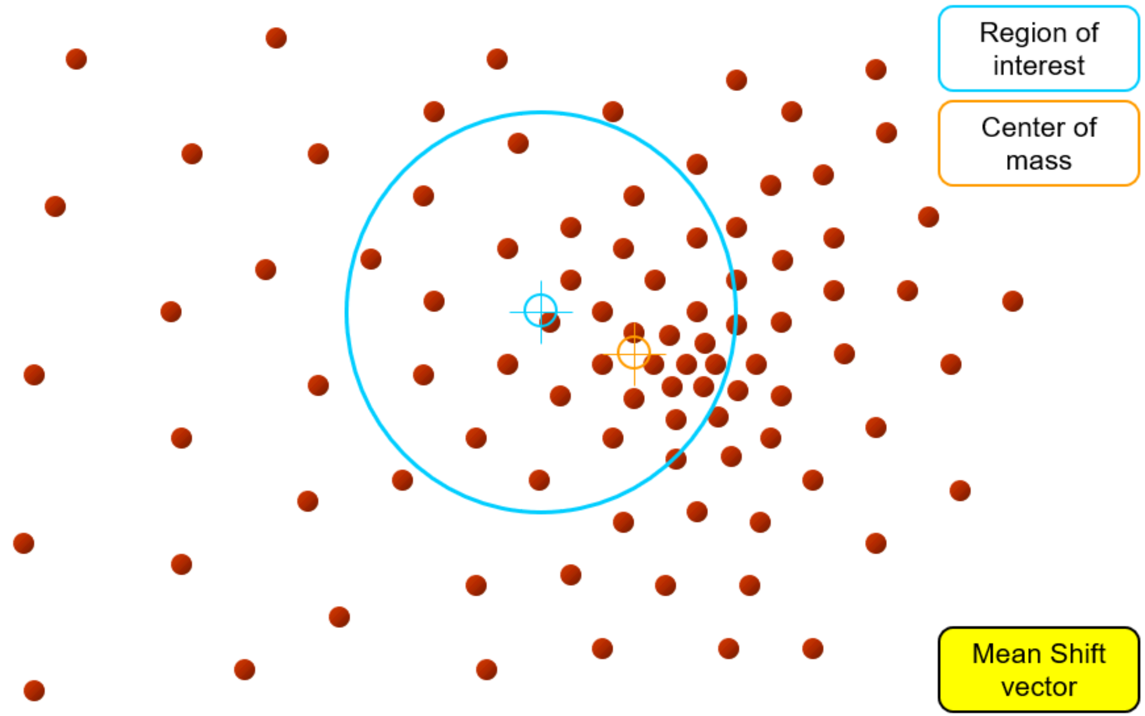
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

Mean shift algorithm (4)

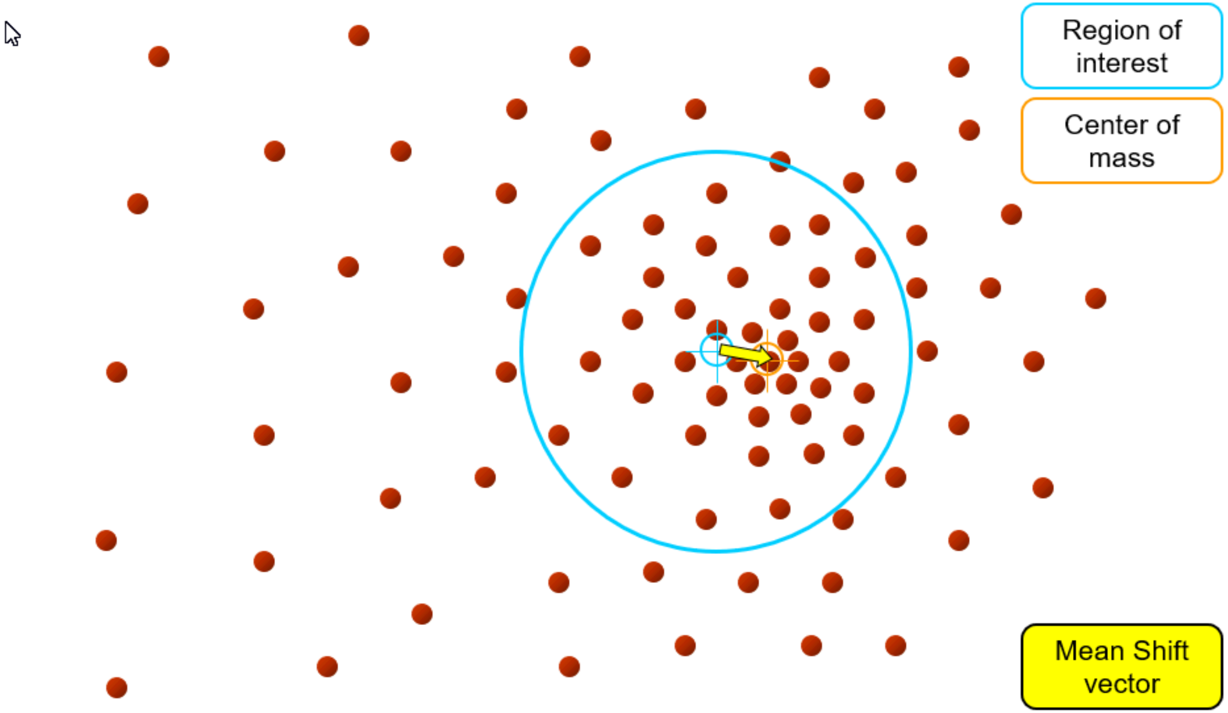
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

Mean shift algorithm (5)

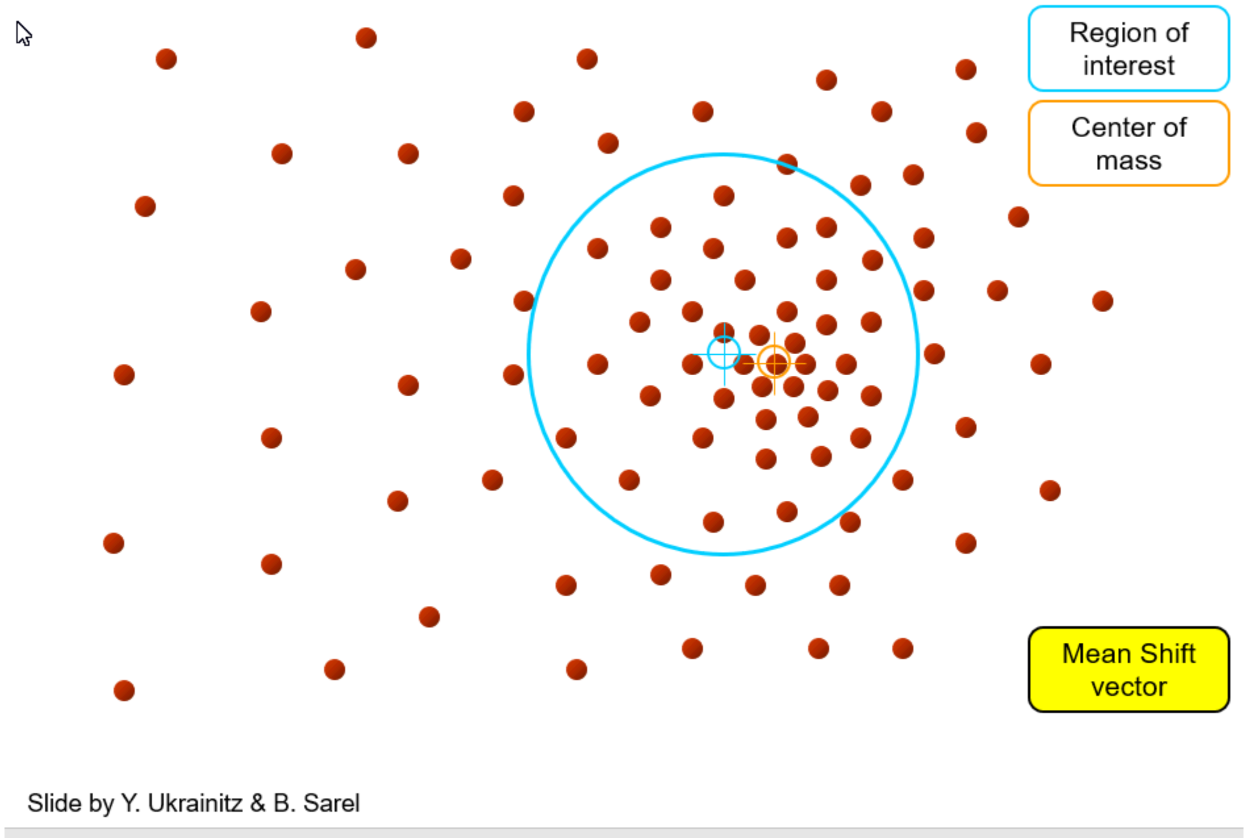
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

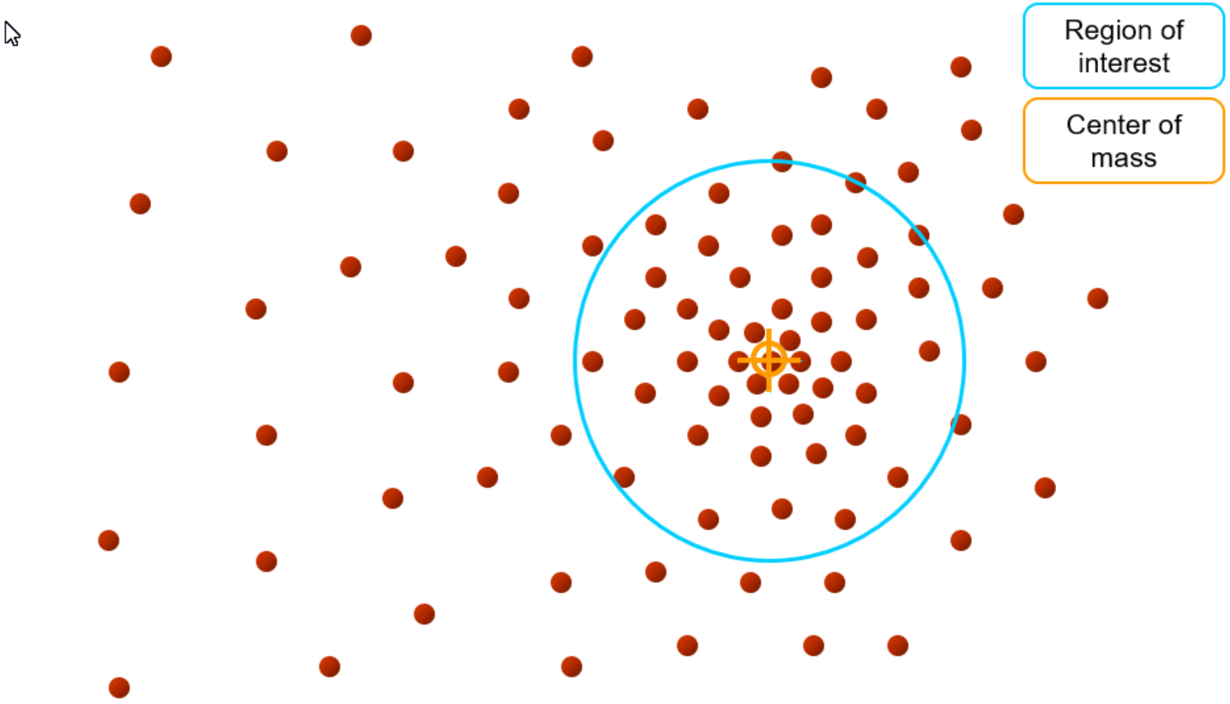
Mean shift algorithm (6)

- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Mean shift algorithm (7)

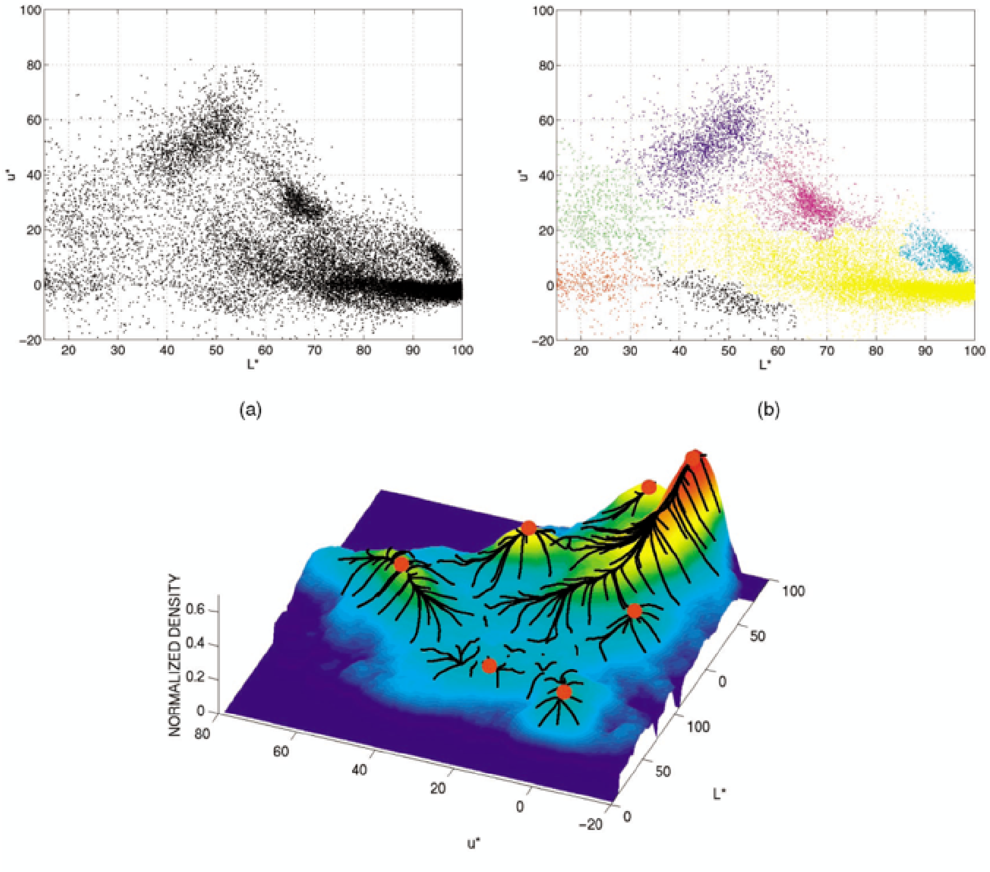
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.
- Repeat until convergence.



Slide by Y. Ukrainitz & B. Sarel

Mean shift image segmentation algorithm

1. Convert the image into tokens (via color, gradients, texture measures etc).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same 'peak' or mode.
5. The data these merged windows traversed are clustered together.

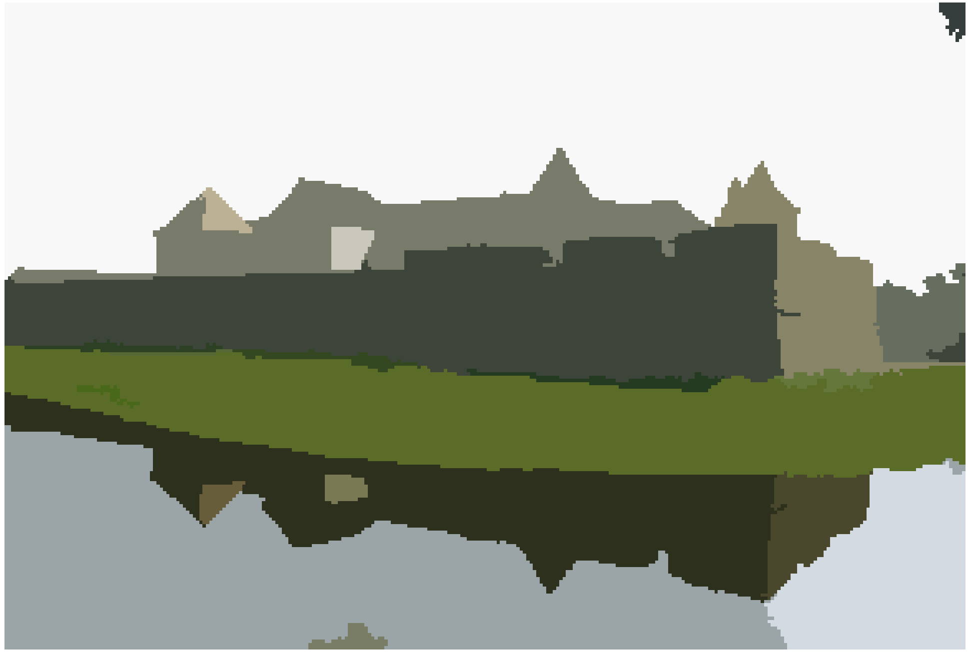


(a)

(b)

(c)

Mean shift image segmentation, Example 1



http://www.caip.rutgers.edu/∼comanici/MSPAMI/msPamiResults.html

Mean shift image segmentation, Example 2

Mean shift image segmentation, Example 3

http://www.caip.rutgers.edu/∼comanici/MSPAMI/msPamiResults.html

Mean shift; pros and cons

Pros

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

Cons

- The output depends on the window size
- Computationally expensive
- Does not scale favorably with the dimension of the feature space