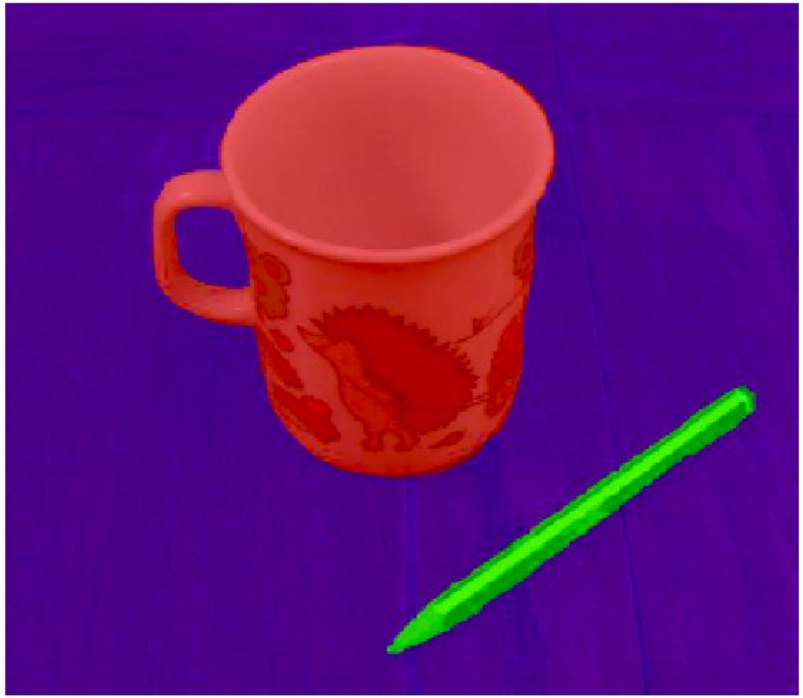
Image segmentation

J. Škovierová
julia.skovierova@cvut.cz
B-604b, Jugoslávských partyzánů 1580/3,
160 00 Prague 6, Dejvice

What is segmentation?

What is segmentation? Motivating picture



Image, courtesy Ondřej Drbohlav

What is image segmentation ?

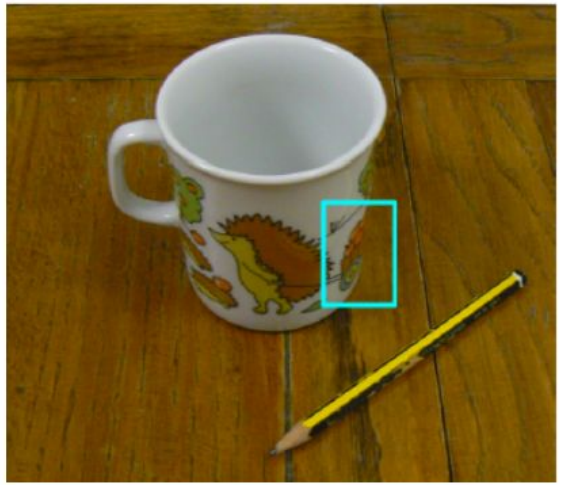
- **Segmentation**
  - dividing an image into meaningful regions
  - the bridge from raw image data to understanding what's in the scene.
- **Regions** (i.e., compact sets) represent spatial closeness naturally and thus are important building steps towards segmentation. Objects in a 2D image very often correspond to distinguishable regions.
- The **object** is everything what is of interest in the image (from the particular application point of view). The rest of the image is **background**.
- The approach is similar to that used in pattern recognition, i.e., **division of the image into set of equivalence classes.**

Motivation - areas of use

Application area	Purpose of Segmentation	Usually used Techniques	Example Output
Object Recognition	Identify objects and their boundaries	Region-based, deep learning	Segmented objects (e.g. person, chair)
Medical imaging	Detect organs or anomalies	Thresholding, Graph cuts, U-Net CNNs	Tumor/organ masks
Autonomous Driving	Real-time scene understanding	Semantic, Instance, Panoptic segmentation	Road, cars, sidewalks, pedestrians,...
Remote Sensing	Land cover classification	Clustering, Spectral analysis	Land use maps

Segmentation can be difficult

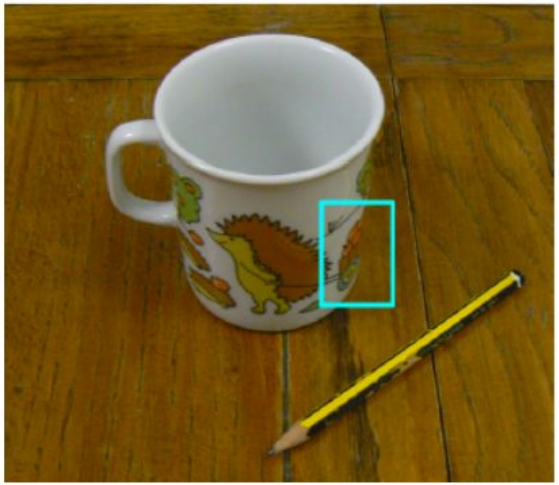
- It is difficult to find border between the cup and background in the indicated region because it does not differ in a local view.



Image, courtesy Ondřej Drbohlav

Segmentation can be difficult

- It is difficult to find border between the cup and background in the indicated region because it does not differ in a local view.
- Only knowledge of the cup semantics can solve the puzzle



Image, courtesy Ondřej Drbohlav

Image segmentation

- There is often no single answer how to segment.
- Segmentation is mostly based on rather ad hoc methods.
- There is no encompassing broad theory of segmentation. However, several recent theoretically grounded approaches have formulated segmentation as an optimization task (e.g., in a Markovian fields formalism).
- The special case of foreground vs. background segmentation is often met.
- Segmentation usually makes sense in a scope of a particular application.

Segmentation is application dependent

- Segmentation depends on an application, its semantics.
- Methods are not universally applicable to all images.
- Direct segmentation of the input image takes pragmatically into account semantic information about a particular application.
- This is the way how to bypass ill-posed problems in inverted tasks related to image formation physics.
- A vital role of a priori information:
  - Low-level: e.g., brightness, spatial coherence, color, texture, motion, . . .
  - Mid-level: object symmetries, proximity on larger scale, . . .

Bottom-up vs. top-down approach

- **Bottom-up**
  - “From pixels to perception”
  - Pixels belong together, because they look similar
  - Driven by local pixel data
  - No additional information is used
  - Uses just low-level information
  - Thresholding, Clustering, Mean shift, Graph cut
- **Top-down**
  - “From knowledge to detail”
  - Pixels belong together, because they come from same object
  - Prior knowledge / model
  - Semantic Segmentation (CNNs, Transformers)

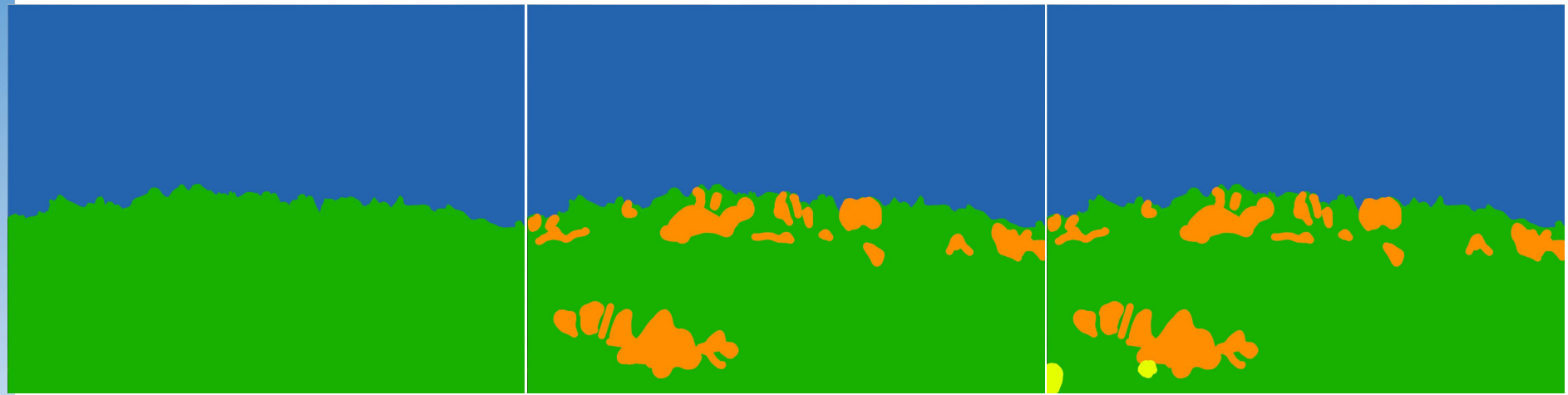
Segmentation by humans

- Task for humans: segment image

- Top-down approach
- Highly individual - different results
- Each human is using their own “top-down” approach based on their experiences, interests, tasks

Segmentation by humans

- Division into sky and earth only
- Added division of trees by color
- Added segmentation of animals



Humans vs Computational Approaches

- **Humans**
  - **Subjective:** Different observers may interpret object boundaries differently (especially in complex or abstract scenes).
  - Influenced by attention, context, and prior knowledge.
  - Example: One person might segment “forest” as one unit; another might separate trees, leaves, and sky.
- **Computational Segmentation**
  - Based on measurable image features: intensity, color, texture, edges, gradients.
  - Objective and reproducible.
  - Algorithms differ in their definition of similarity or boundary.
  - However, often less adaptive to semantic meaning unless aided by machine learning or deep networks.

Complete vs. partial segmentation (1)

- **Complete segmentation** -- divides an image into non-overlapping regions that match to the real world objects.
- Complete segmentation divides an image *R* into the finite number *S* of regions $R_1, \ldots, R_S$

$$R = \bigcup_{i=1}^{S} R_i\,, \qquad R_i \cap R_j = \emptyset\,, \qquad i \neq j\,.$$

- **Partial segmentation** -- it is possible to find only parts with semantic meaning in the image (e.g., regions, collection of edgels) which will lead to interpretation in later analysis.

Complete vs. partial segmentation (2)

- **To proceed from partial to complete segmentation** it is needed to explore the higher-level of information processing.
- This can be performed iteratively in a feed-back loop.
- The information about the semantics of the specific application is used.
- Partial segmentation reduces the amount of data which need to be processed.

- Examples of a complete 2D segmentation
  - Seek **contrast objects in a homogeneous background.** Intensity thresholding provides a silhouette corresponding to objects, e.g., printed characters, cell kernels, back-light illuminated details inspected in industry.

Observations (or features) and segmentation

- There is an important question: Which observations (features in the pattern recognition terminology) distinguish regions corresponding to different objects in the image?
- Examples of features: intensity, color, shape of the region, texture, motion in video, disparity in stereo imaging.
- **Primary features** are provided by the sensor directly (e.g., intensity, color, depth in the range camera, temperature in far infrared (thermal) camera).
- More complicated **secondary features** have to be calculated from primary features as: texture parameters, shape parameters of the region, mutual relations between regions, motion parameters in video, stereo disparity, etc.

Gestalt principles of perceptual organization

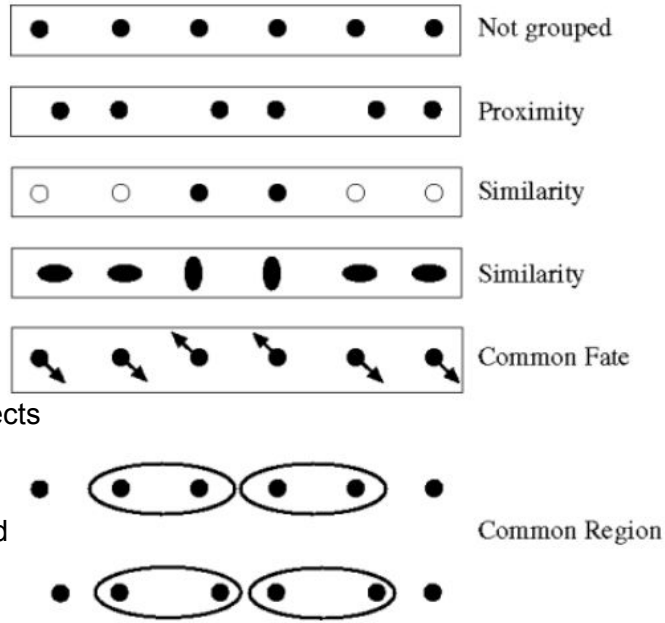
- Perceptual organization refers to the mental and physiological steps that group parts of the visual world together to form objects.
  - Founding publication by Max Wertheimer (born in Prague) in 1912 in Frankfurt a.M.
  - Gestalt theory was meant to have general applicability; its main tenets, however, 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.

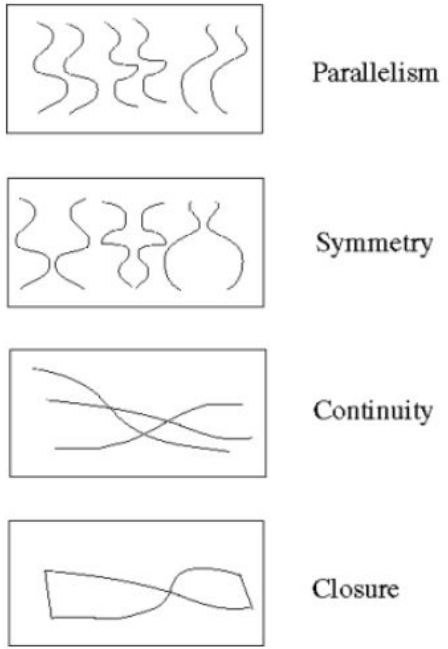
Gestalt principles of perceptual organization

- **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.

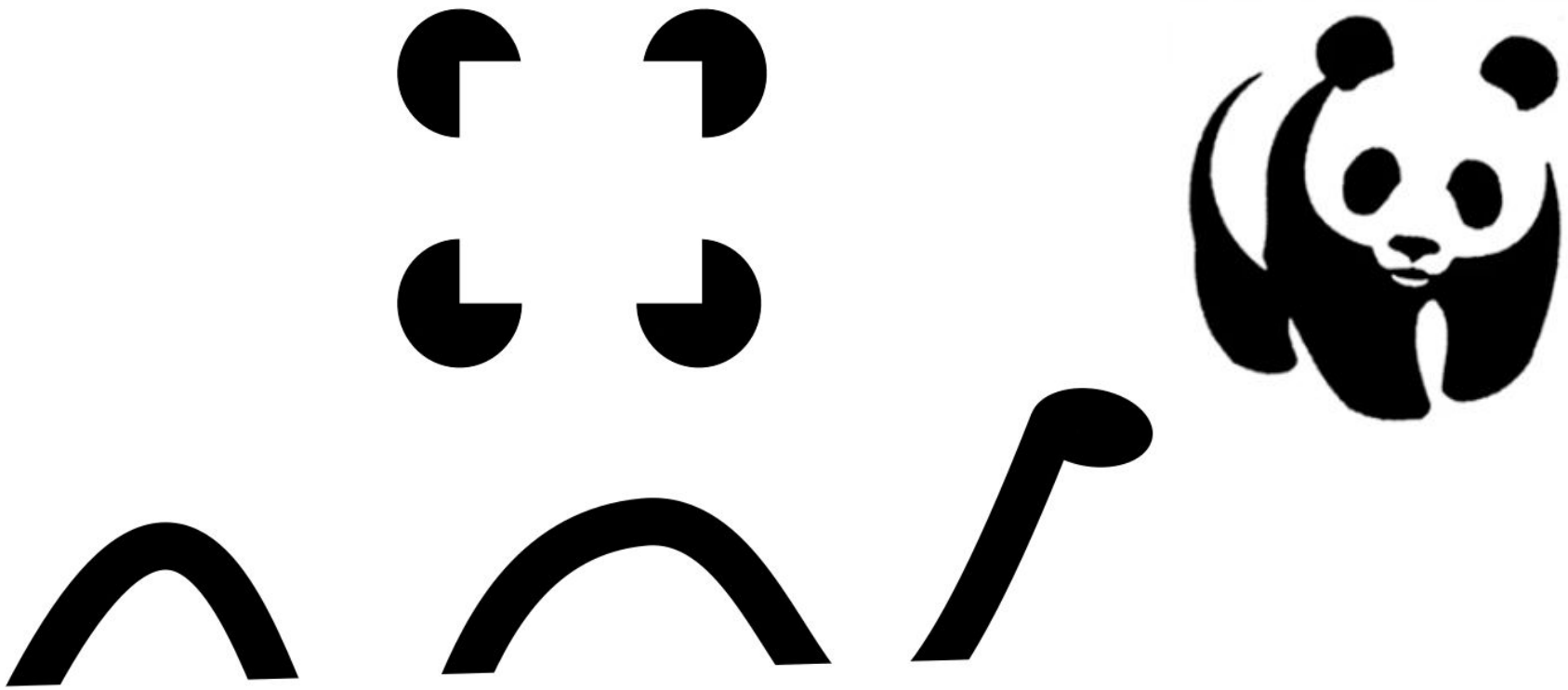
Gestalt grouping principles

- **Proximity**
  nearby elements form groups
- **Similarity**
  similar color/shape/texture = grouped
- **Continuity**
  humans prefer smooth lines
- **Closure**
  humans fill in gaps to form complete objects
- **Figure–Ground**
  separation of foreground and background
- **Common Fate**
  elements moving together are grouped together





Illusory contours



Grouping is not always easy

Picture by R. C. James

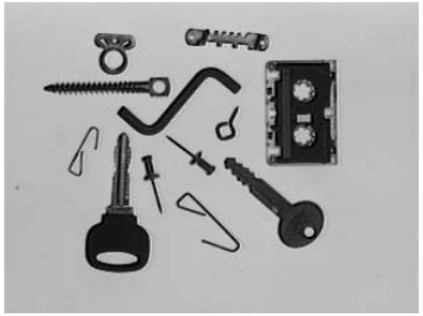
Thresholding

- Input image *f (i, j)*, output image *g(i, j)*.
- For each pixel *(i, j)*

$$g(i,j) = \begin{cases} 1 & \text{for} \quad f(i,j) \geq \text{Threshold}\,, \\ 0 & \text{for} \quad f(i,j) < \text{Threshold}\,. \end{cases}$$

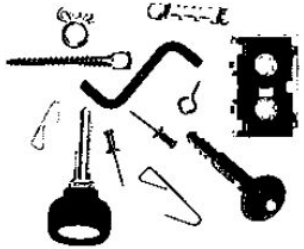
- + Simple technique, long time and more often used.
- + Easy in hardware, intrinsically parallel.
- – The threshold is a parameter which is difficult to adjust automatically in general.
- – Works only for subclass of images in which objects are distinct from background in intensity

Example, threshold influence

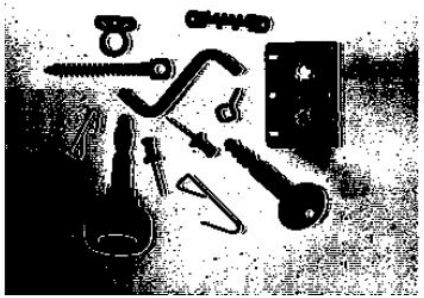


Original image.

Threshold segmentation.



Threshold too low.



Threshold too high.

Thresholding, modifications

- **Local adaptive thresholds**, e.g. divide an image into subimages and find threshold in each of them.
- **Band thresholding**, let *D* be a set of intensities, e.g. an interval of intensities

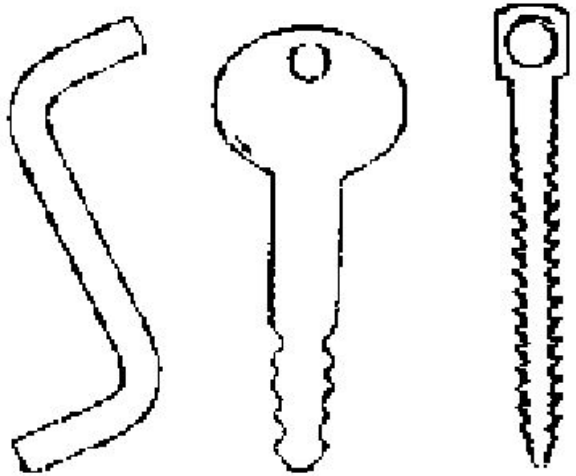
$$g(i,j) = \begin{cases} 1 & \text{for } f(i,j) \in D\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

- **Multiple thresholds**.
- **Semi-thresholding,** to suppress background, useful in cases in which image is analysed by a human

$$g(i,j) = \begin{cases} f(i,j) & \text{for} \quad f(i,j) \geq \text{Threshold}\,, \\ 0 & \text{for} \quad f(i,j) < \text{Threshold}\,. \end{cases}$$

Example. Border regions by the band thresholding

Original image.



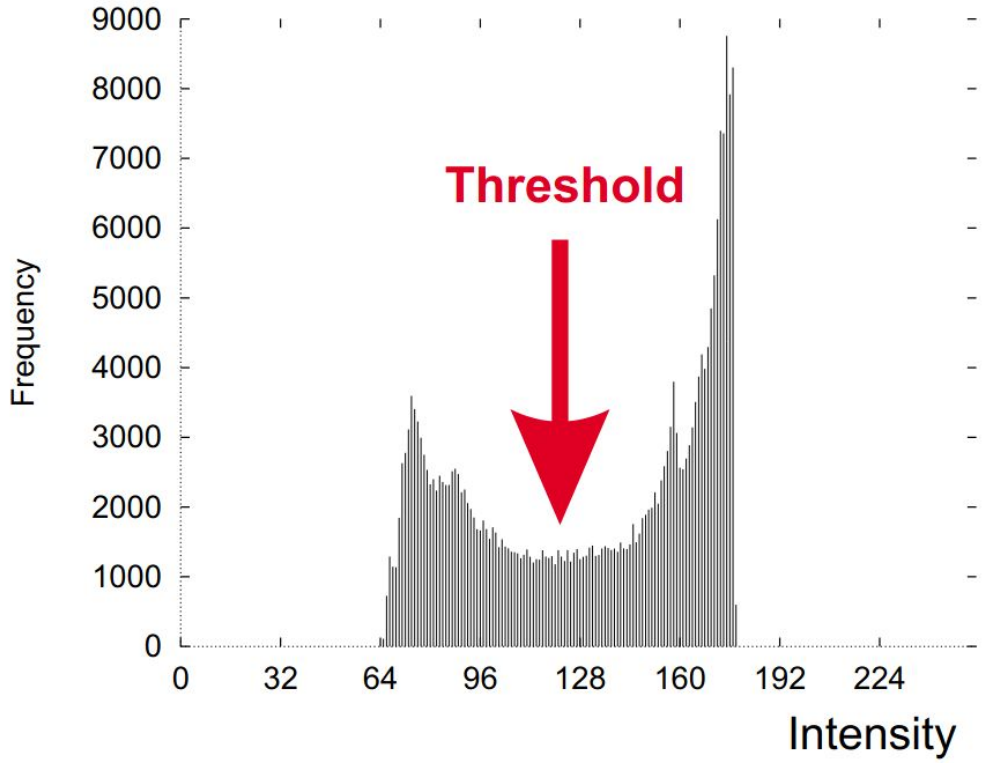
Border regions detected.

Automatic threshold detection, Use of histogram

- ***p*-tile thresholding,** if we know that the objects cover 1/p of the image, e.g. printed characters on a sheet ⟹ 1/p area of a histogram.

- **Histogram shape analysis,** distinct objects on background correspond to a bi-modal histogram. Find middle between the modes.

Automatically found threshold according to a bi-modal histogram



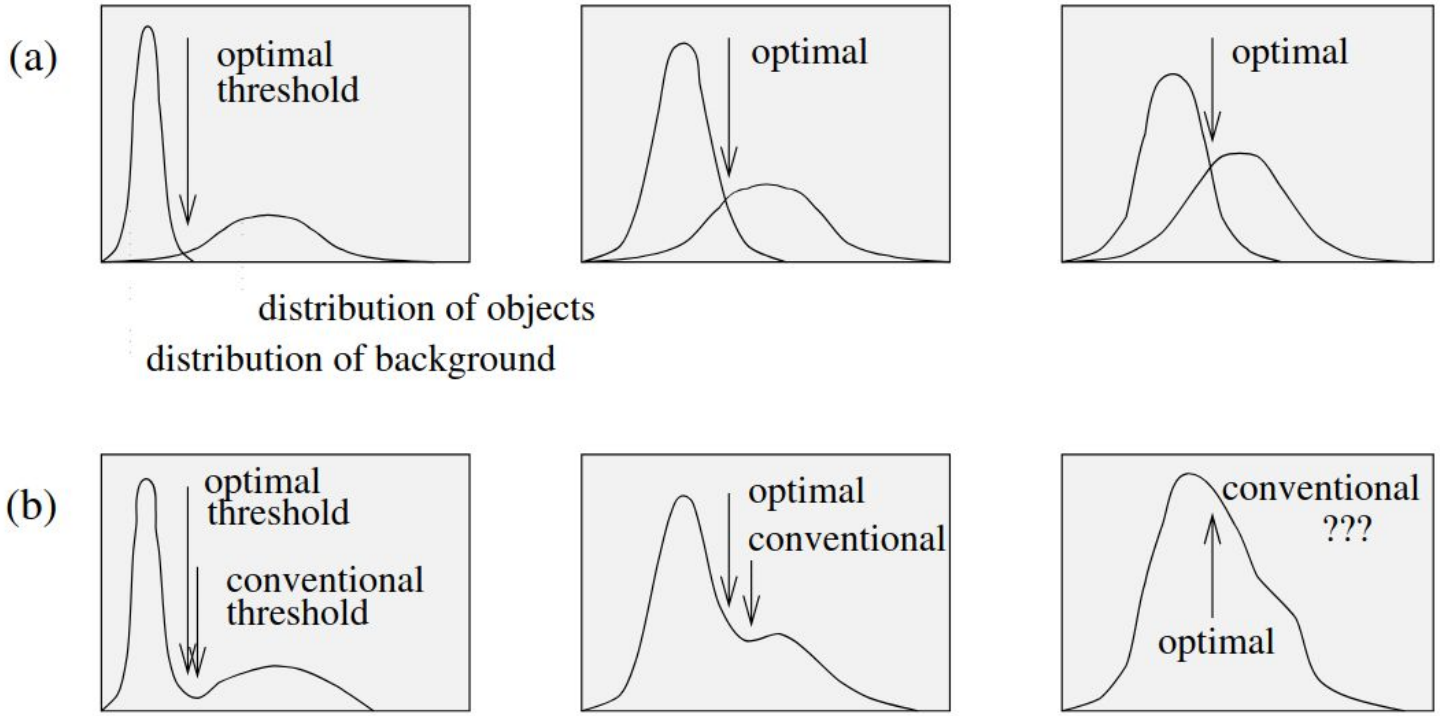
Smoothing of the ragged histogram

- Let $h_f(i)$ be a histogram which is ragged. Intensities $i = 0, \ldots, i_{max}$.
- Smoothing before finding the threshold automatically helps to avoid the problem with many local minima.
- The 1D smoothing using sliding averaging of the window size 2K + 1 is often used.
- The new histogram $h'_f(i)$ is calculated as

$$h'_f(i) = \frac{1}{2K+1} \sum_{j=-K}^{K} h_f(i+j), \quad i = K, \ldots, i_{\max} - K$$

Optimal thresholding by a mixture of Gaussians

- **Motivation:**



Local thresholds by mixture of Gaussian

- h_{region} – local histogram.
- h_{model} – approximation of a histogram by n Gaussian distributions,

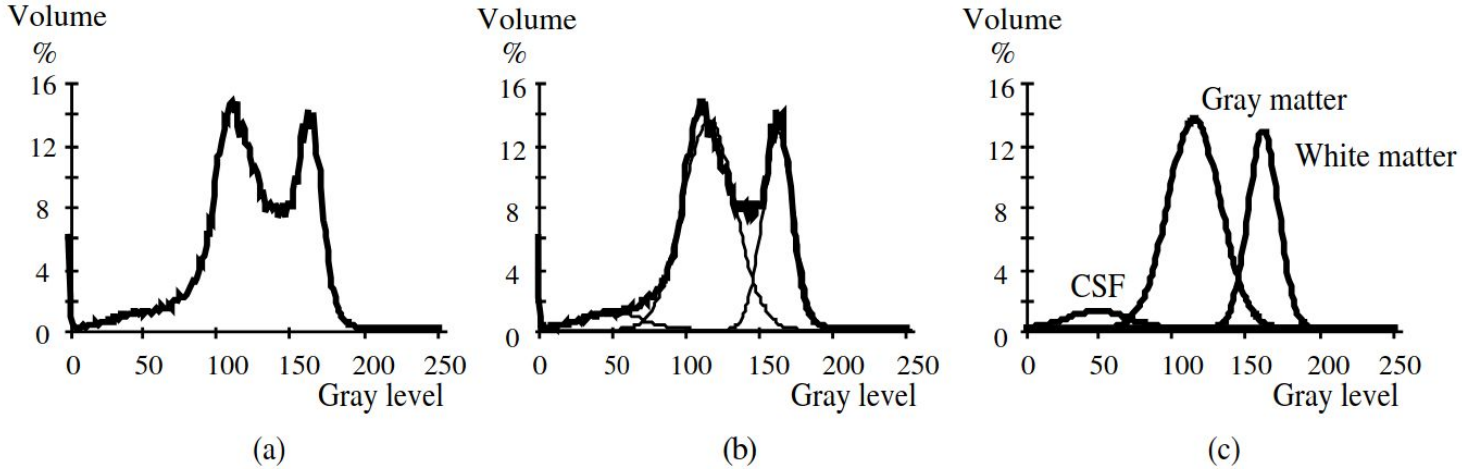
$$h_{\text{model}}(g) = \sum_{i=1}^{n} a_i e^{(-\frac{(g-\mu_i)^2}{2\sigma_i^2})}$$

- The minimized optimization criterion F ,

$$F = \sum_{g \in G} \left(h_{\text{model}}(g) - h_{\text{region}}(g) \right)^2$$

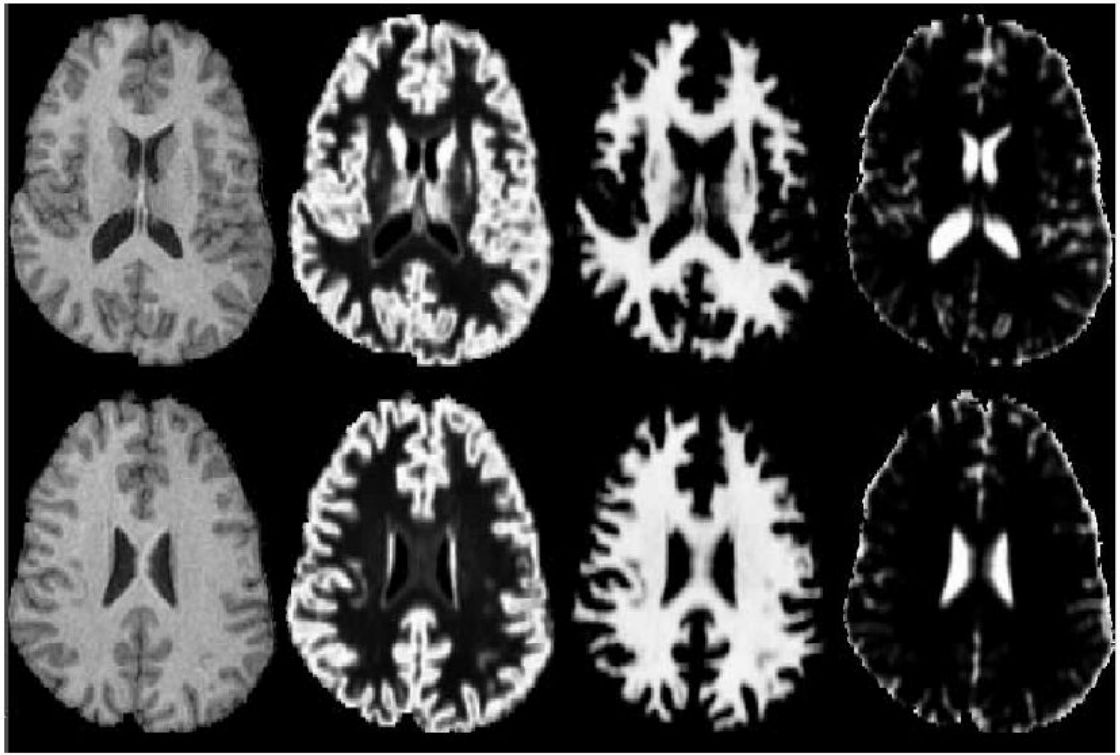
Example, Segmentation of the brain MR

- Input: T1-weighted NMR images.
- Desired classes: white matter, grey matter, celebro-spinal fluid (CSF)



Courtesy: Milan Šonka, University of Iowa.

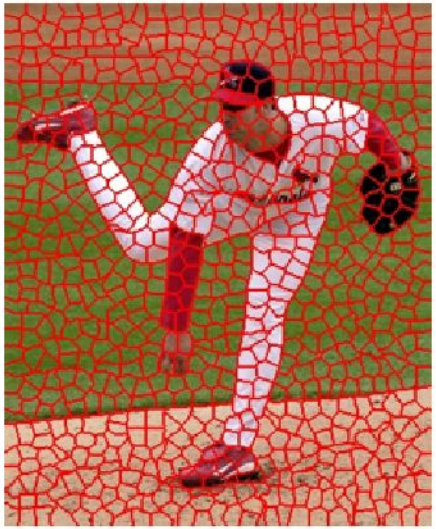
Brain MR, Segmentation result

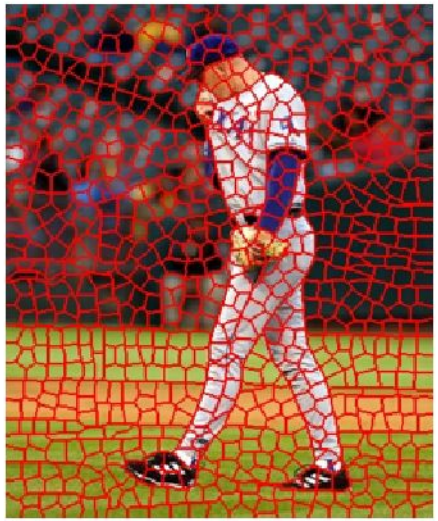


original gray matter white matter CSF

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.

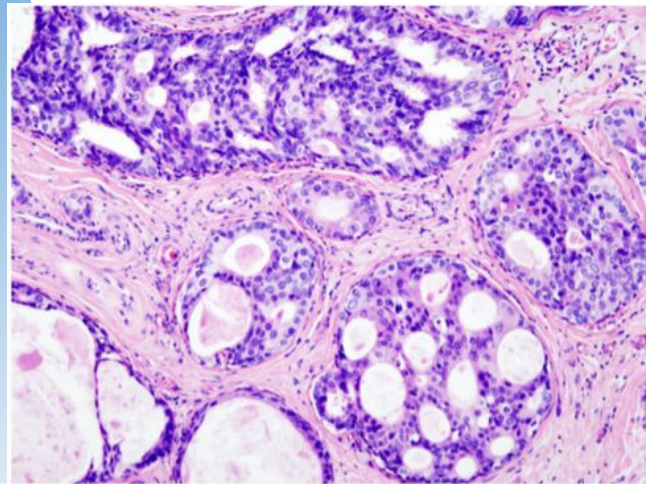




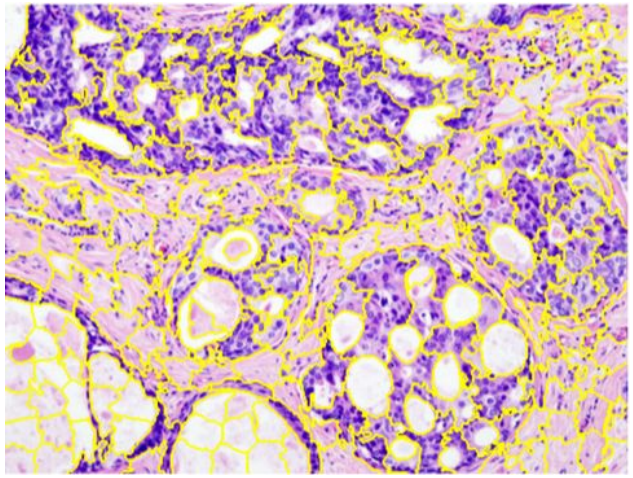
SLIC (Simple Linear Iterative Clustering)

- SLIC over-segments an image into compact, perceptually uniform regions (superpixels) by clustering in a 5-D space of color + position.
- Example : H&E breast histology

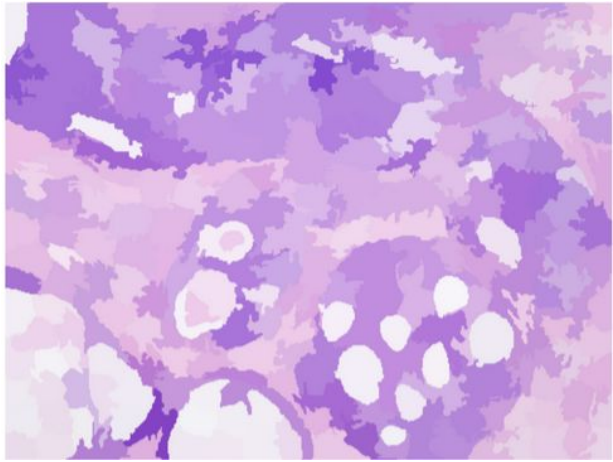
Input H&E breast histology



SLIC boundaries overlaid

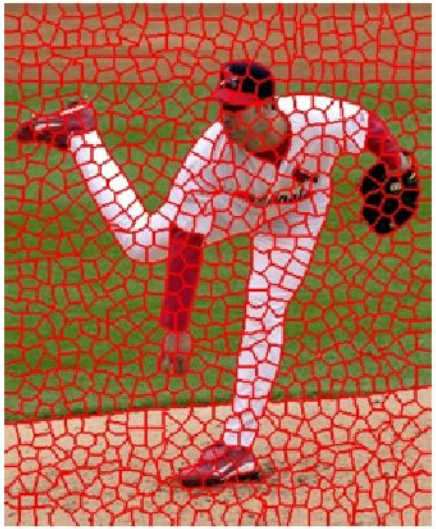


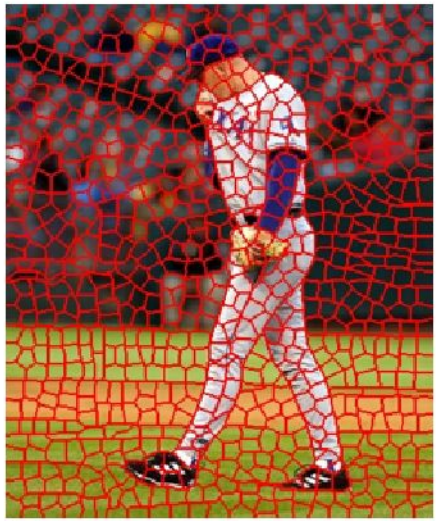
Region-averaged color (one color per superpixel)



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.



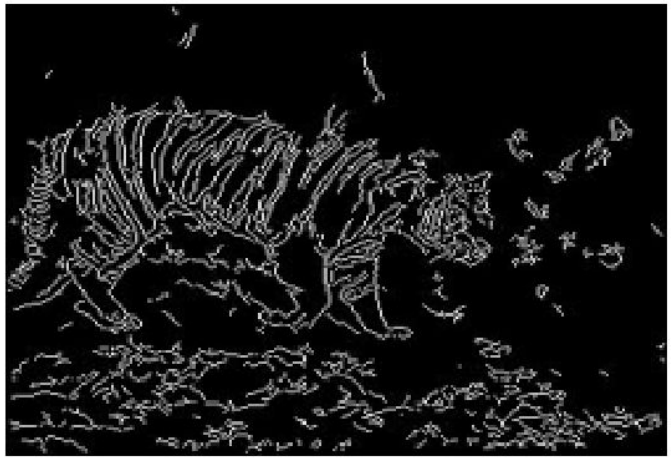


Contours, boundaries in segmentation

- 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.

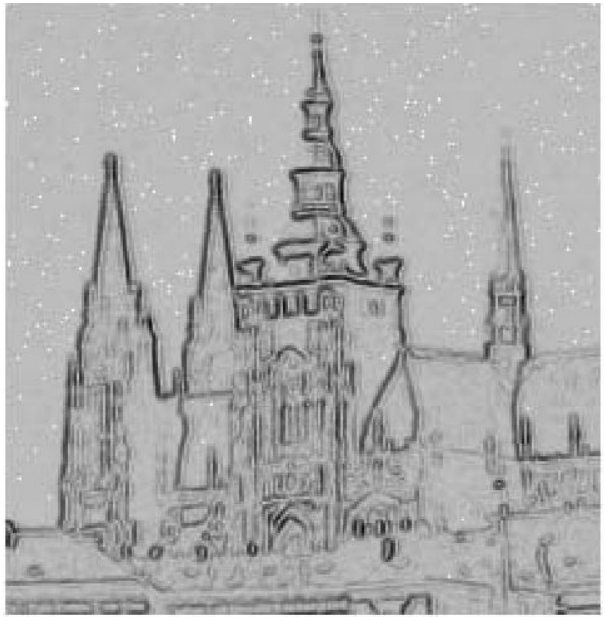
- Problems with edge-based segmentation:
  - It is difficult to obtain closed contours, i.e., boundaries of regions.
  - The whole image need not be segmented.

Edge-based image segmentation (3)

- **Edge detection**
  - Canny, Sobel, Laplacian
- **Link edges into contours**
  - connects neighboring edge pixels and forms closed contours
- **Fill or label enclosed regions**

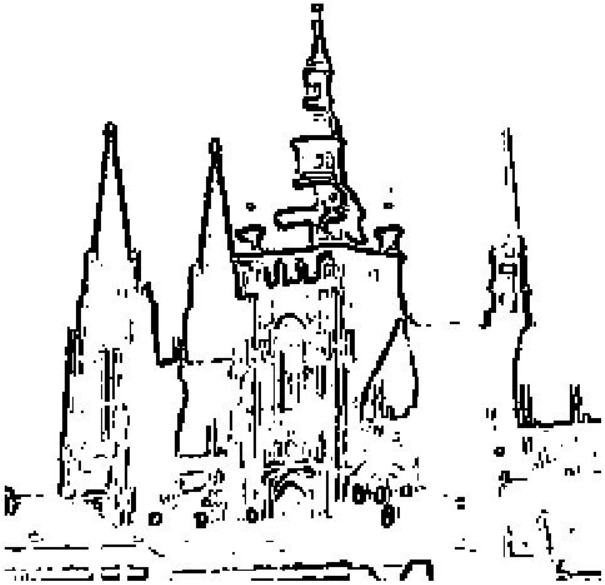
Edge image thresholding (1)

original image

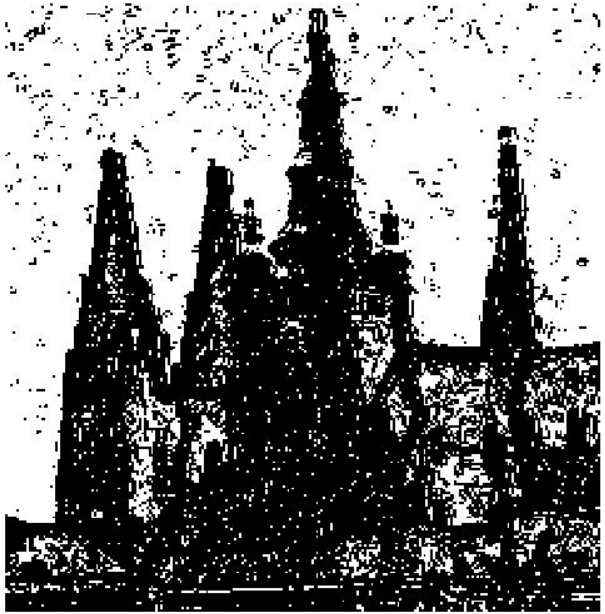


edge image
(enhanced for display)

Edge image thresholding (1)

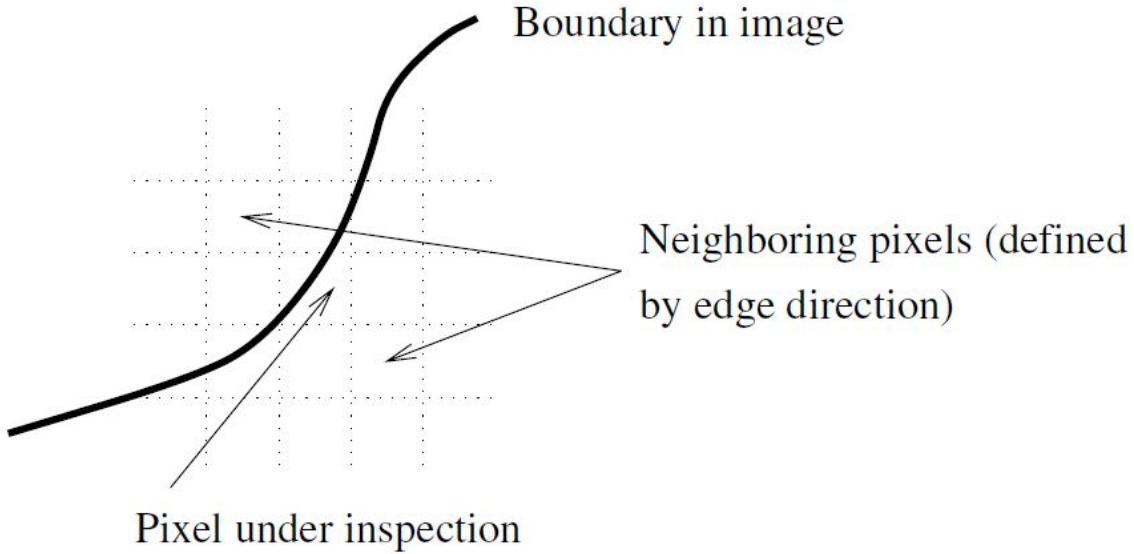


threshold 30



threshold 10 (too small)

Non-maximal suppression and hysteresis



Pixels adjacent with respect to local edge information are inspected.

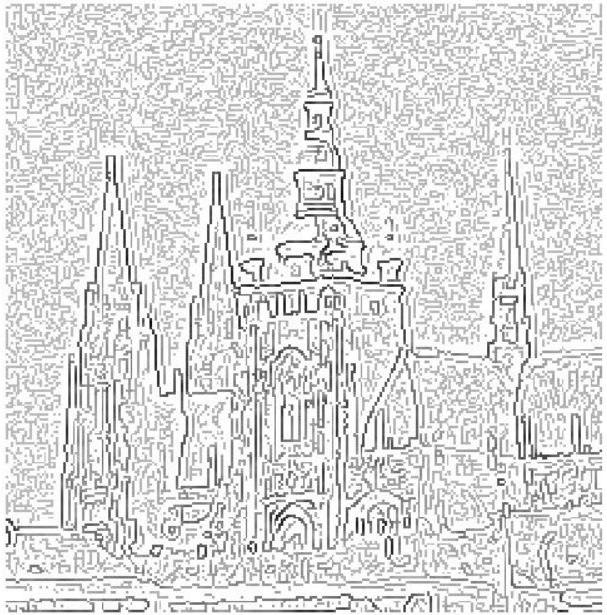
Non-maximal suppression

- Goal
  - To make detected edges **thin and precise** by keeping only the strongest edge pixels (local maxima).
- Algorithm
  - For each pixel, find its **gradient magnitude** and **gradient direction** (e.g., horizontal, vertical, or diagonal).
  - Compare the pixel's gradient magnitude with the magnitudes of its **two neighbors along the gradient direction**:
    - If the pixel's value is **not greater** than both neighbors, set it to **zero** (suppress it).
    - If it **is greater**, keep it — it's a local maximum (likely a true edge).

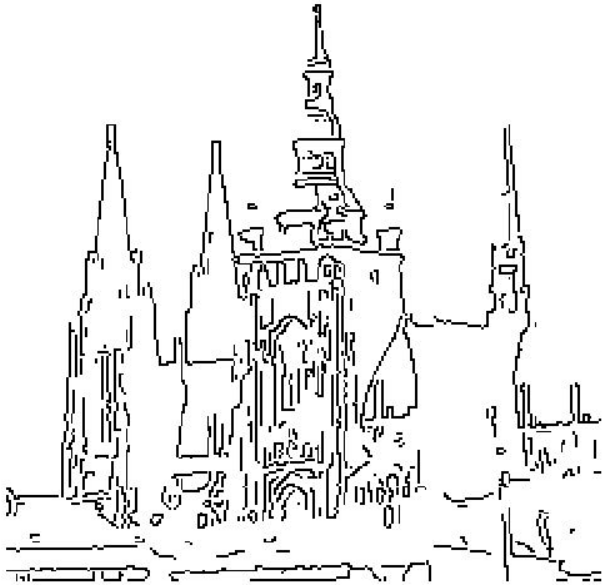
Hysteresis

- Goal
  - To decide which of the remaining edge pixels are **real edges** and which are **noise** — using two thresholds instead of one.
- Algorithm
  - Mark all pixels with gradient magnitude **above T_high** as **strong edges**.
  - Mark pixels with magnitude **between T_low and T_high** as **weak edges**.
  - Remove all others (below T_low).
  - For each **weak edge**, keep it **only if it's connected to a strong edge** (8-neighbor connectivity).

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).

Fill/label enclosed regions

- **Region filling** (also called **flood filling**) is the process of **filling the interior** of a **closed boundary** (like the edge of an object) with a particular label or color — so that we can define a *region* inside that boundary.
- Flood Fill Algorithm
  - Input:
    - binary edge map
    - A seed point inside the region to be filled
  - 4-connection
  - 8-connection

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.

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

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

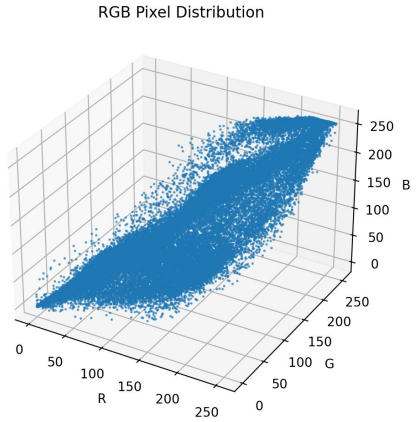
- Iterate until pixels do not move any longer.

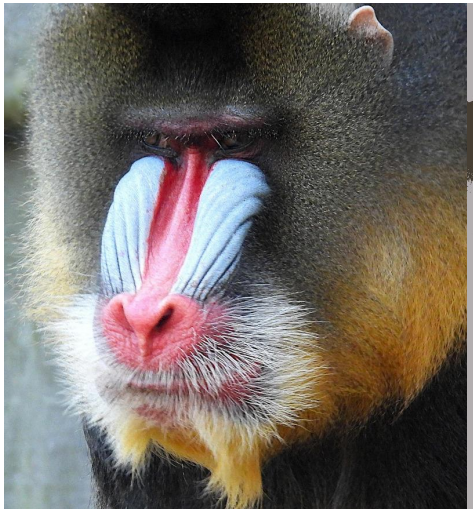
S. Lloyd. Last square quantization in PCM's. Bell Telephone Laboratories Paper (1957). In the journal much later: S.P. Lloyd. Least squares quantization in PCM. Spec. issue on quantiz., IEEE Trans. Inform. Theory, 28:129–137, 1982.

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 $O(2^{\sqrt{n}})$ 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 (3)





Original

k=2

k=8

k=16

K-means example - texture

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

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

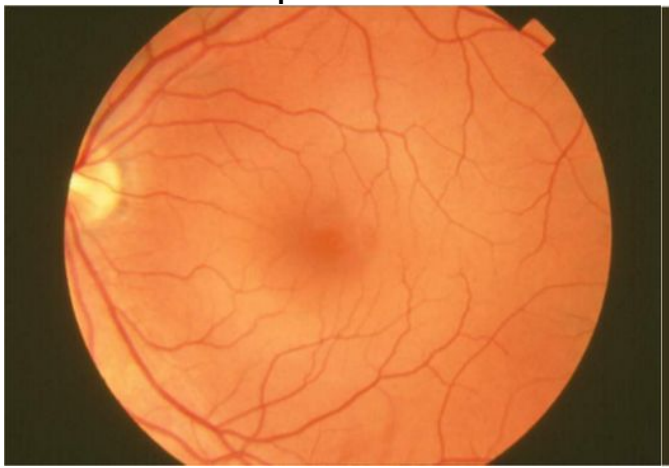
- Two clusters, K = 2

Image, courtesy Ondoej Drbohlav

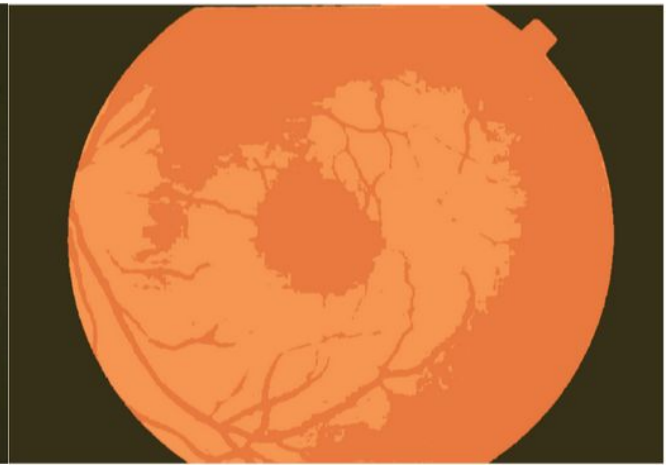
ROBOTIC PERCEPTION GROUP

K-means example - fundus

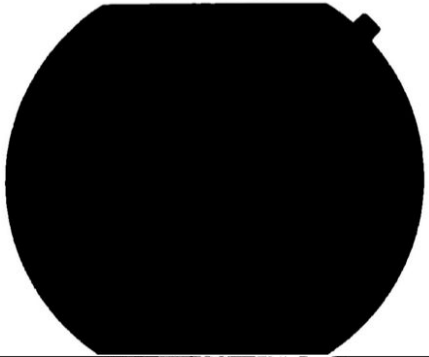
Input

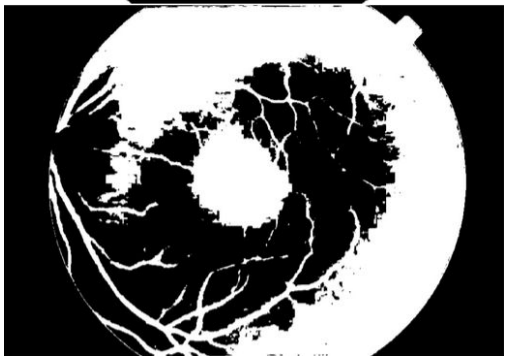


K=3



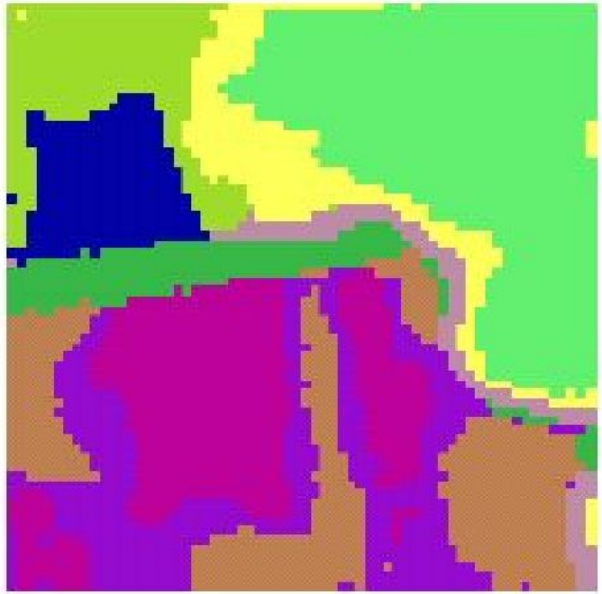
Masks:





Segmentation - remote sensing

- Segmentation of an aerial image of the sea coast



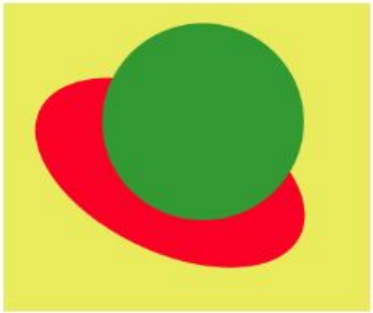
Segmentation by region growing/splitting

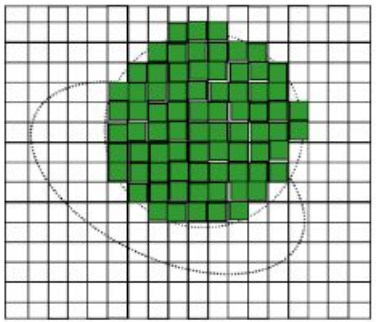
- Region growing algorithm

  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

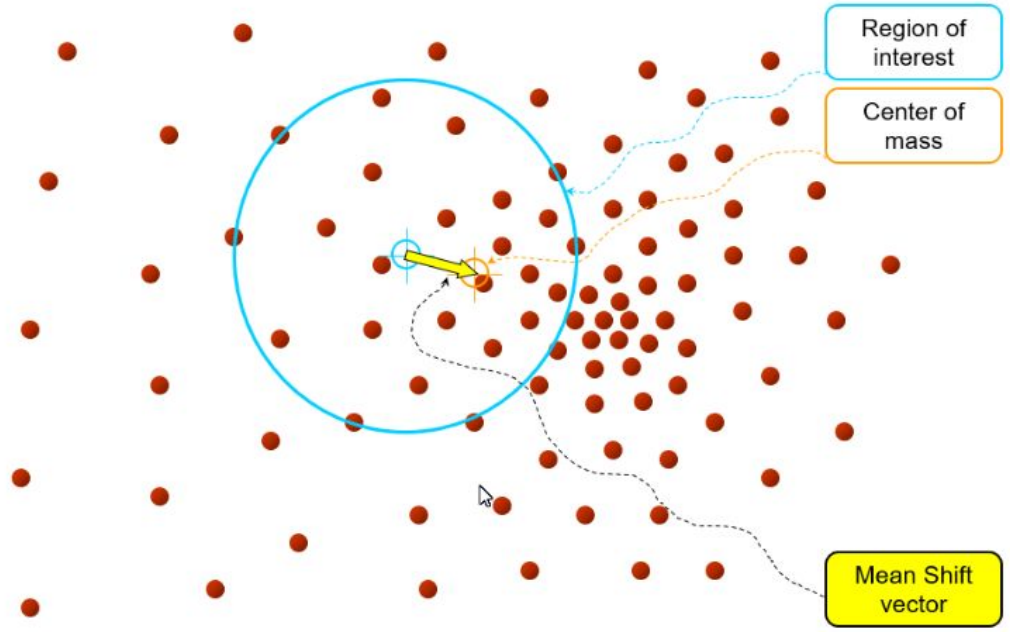
- Estimation of the density gradient - Fukunaga K.: Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.
- Sample mean of local samples points in the direction of higher density. It provides the estimate of the gradient.
- Mean shift vector *m* of each point *p*

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

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

Mean shift (1)

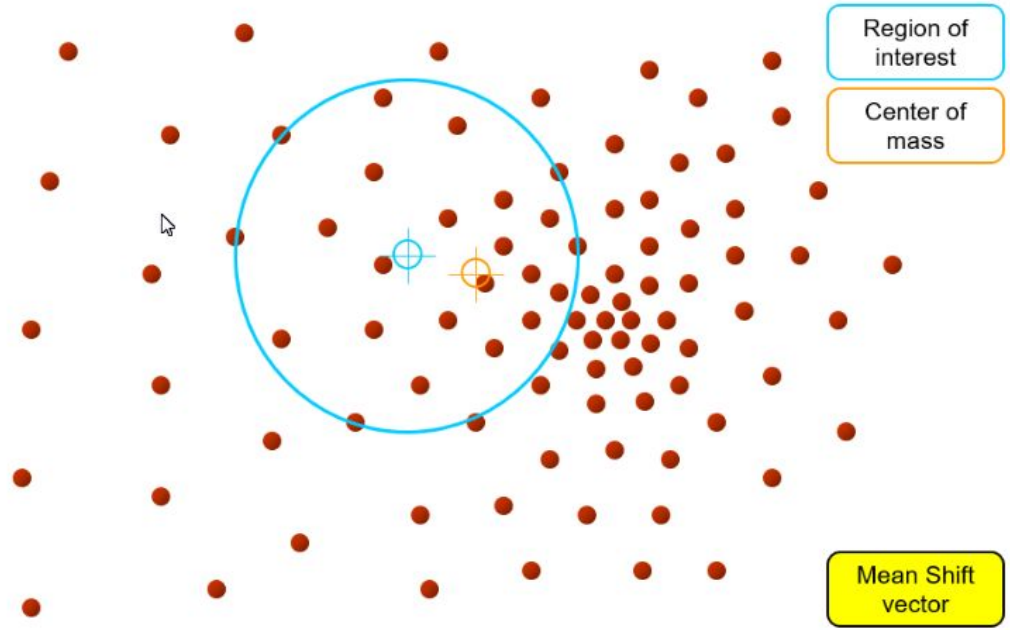
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (2)

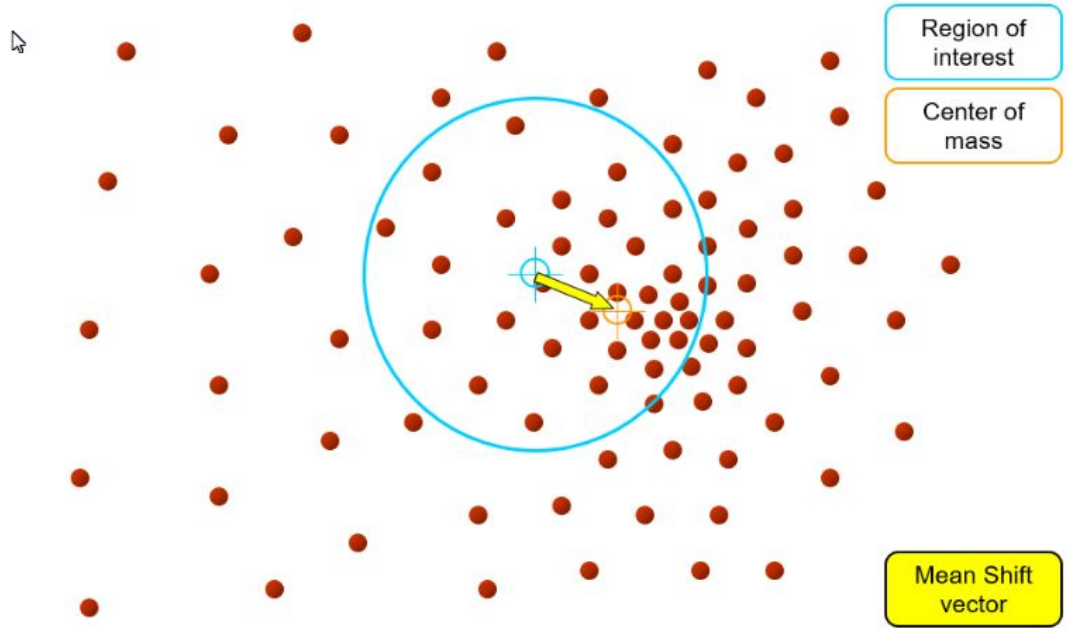
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (3)

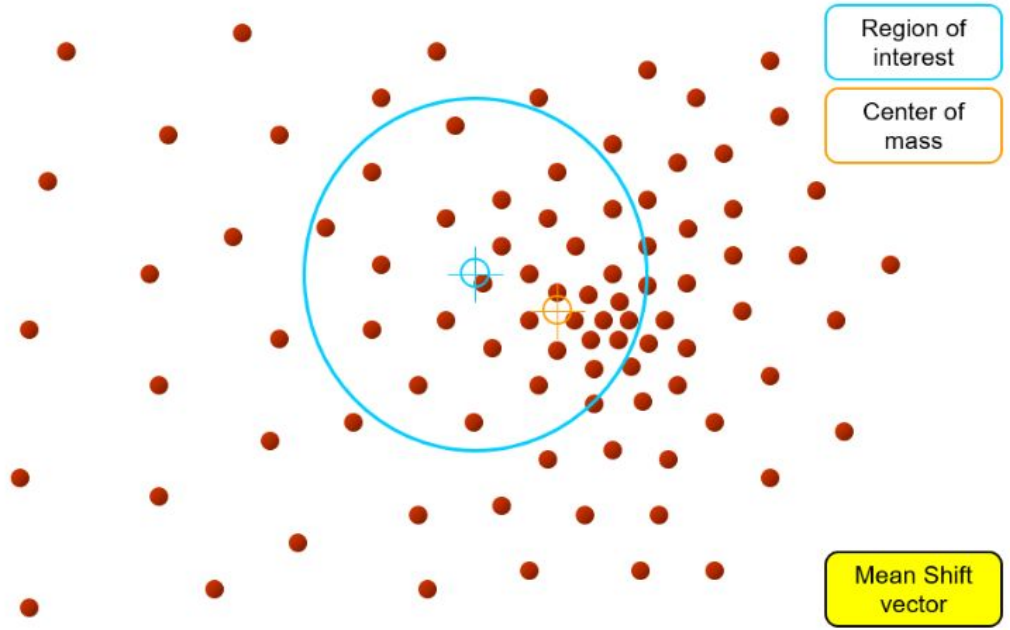
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (4)

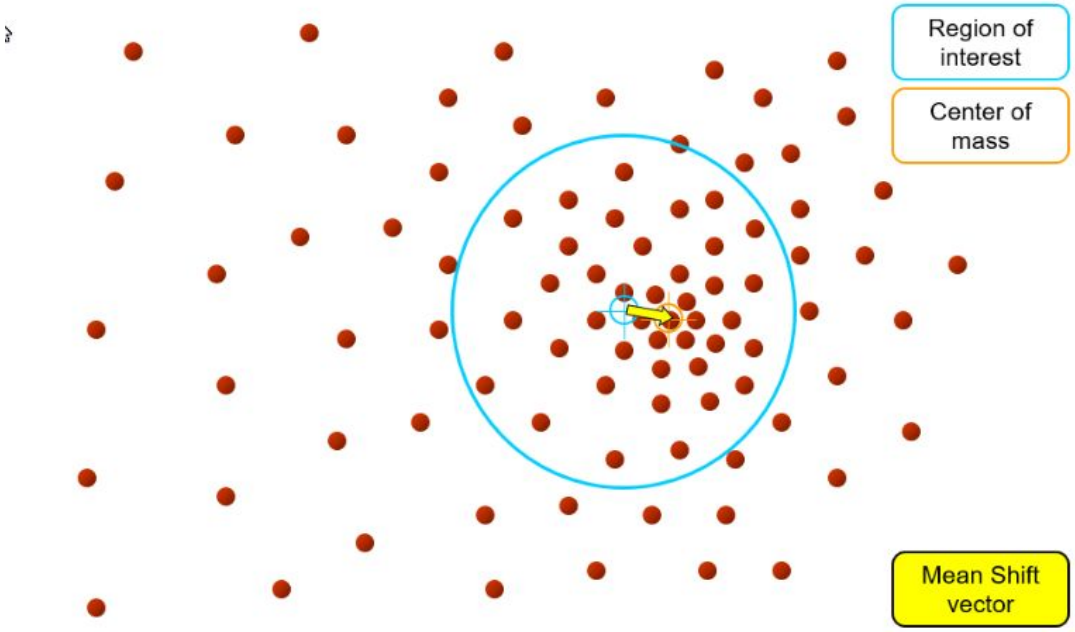
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (5)

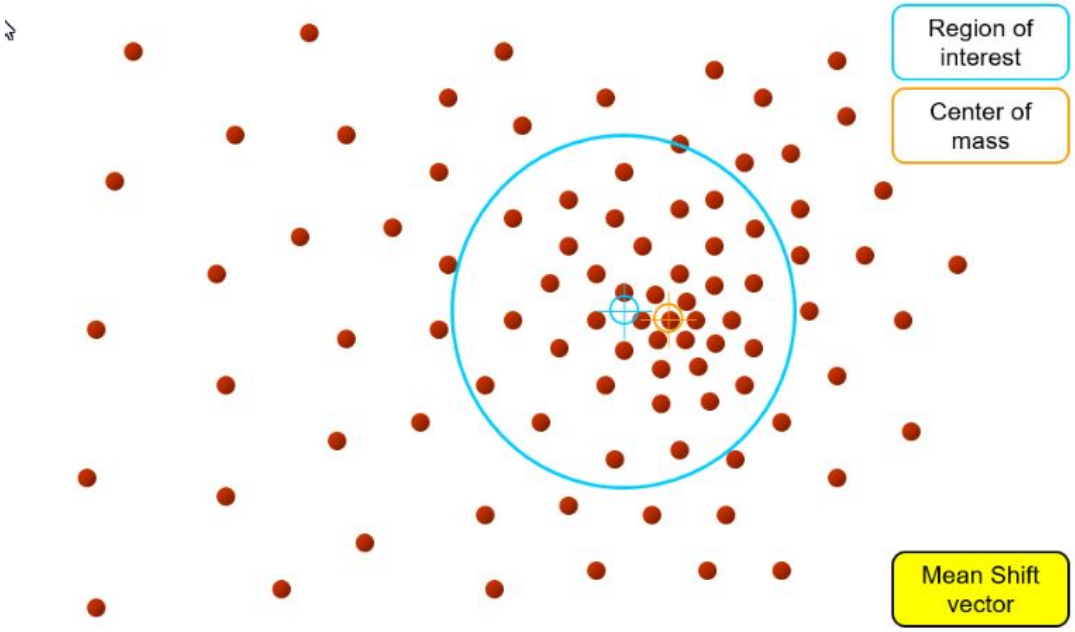
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (6)

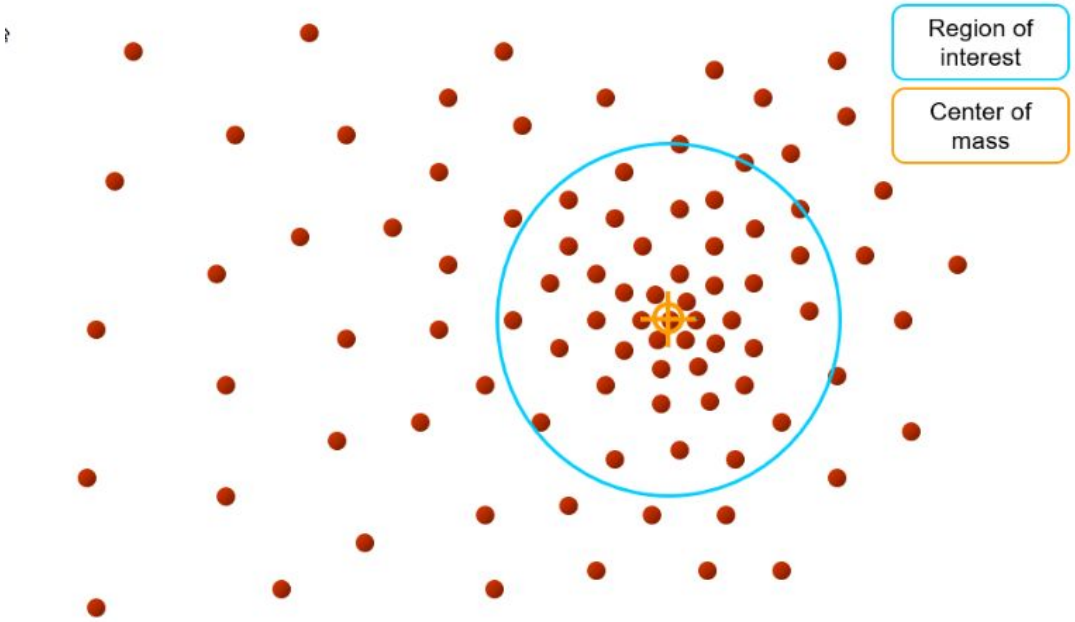
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



Slide by Y. Ukrainitz & B. Sarel

Mean shift (7)

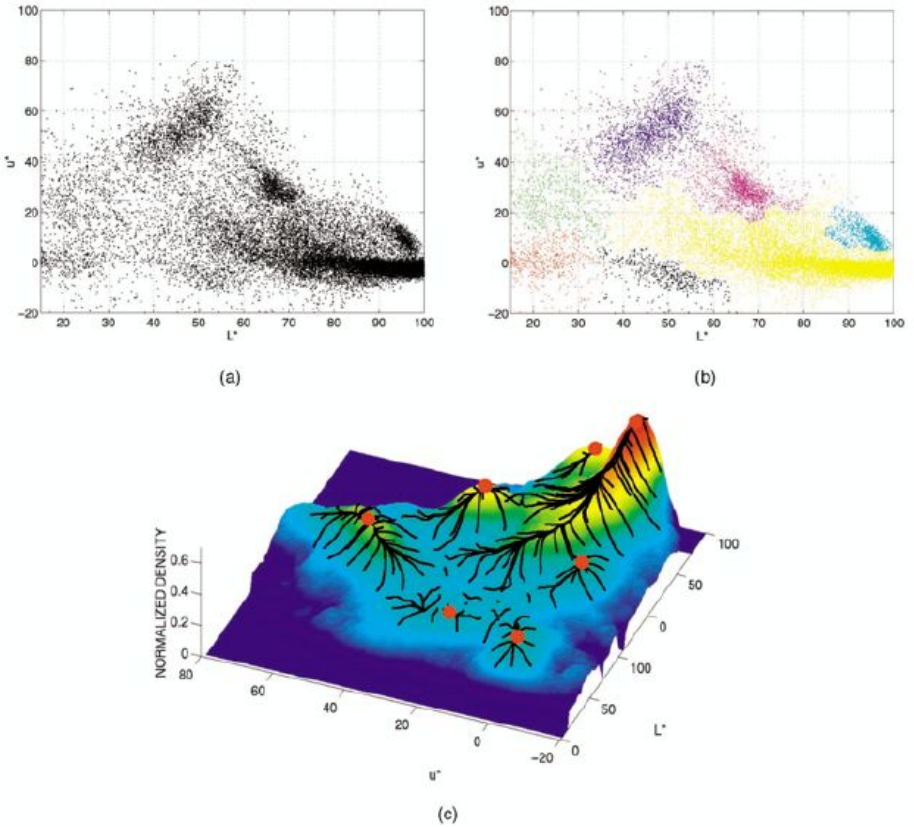
- Input: points in the Euclidean (feature) space.
- Determine a search window size (usually small).
- Choose the initial location of the search window.
- Compute the mean location (centroid of the data) in the search window.
- Center the search window at the mean location computed in the previous step.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



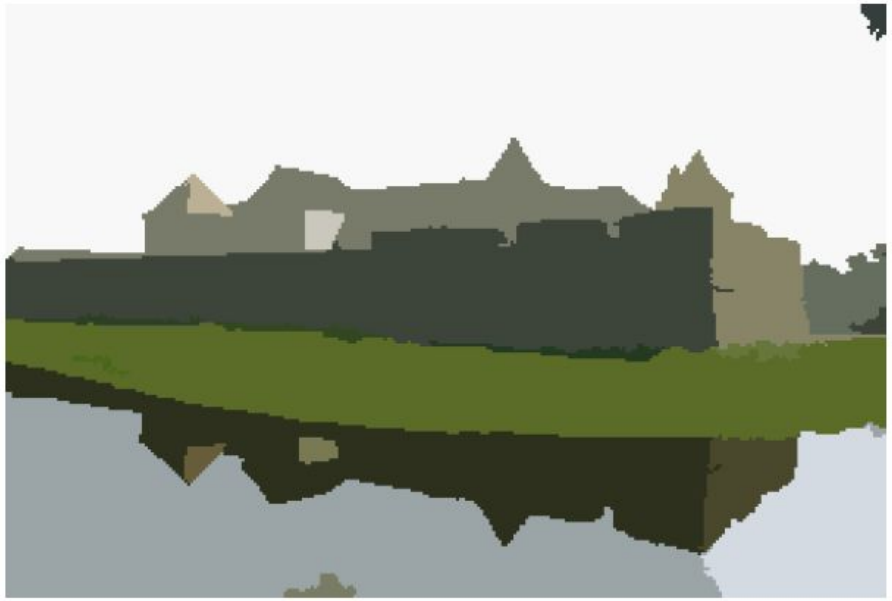
Slide by Y. Ukrainitz & B. Sarel

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.



Mean shift image segmentation, Example 1



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

Mean shift image segmentation, Example 2

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

Semantic vs. Instance vs. Panoptic Segmentation

- **Semantic Segmentation**
  - Classifies each pixel into a category (e.g., sky, tree, road).
  - No distinction between individual instances of the same class.
  - Example: all cars = one “car” region.
- **Instance Segmentation**
  - Identifies each individual object separately.
  - Combines detection + segmentation (e.g., Mask R-CNN).
  - Example: distinguishes car #1 vs car #2.
- **Panoptic Segmentation**
  - Unifies both:
    - Semantic segmentation for amorphous “stuff” (sky, road, grass)
    - Instance segmentation for countable “things” (person, car, tree).

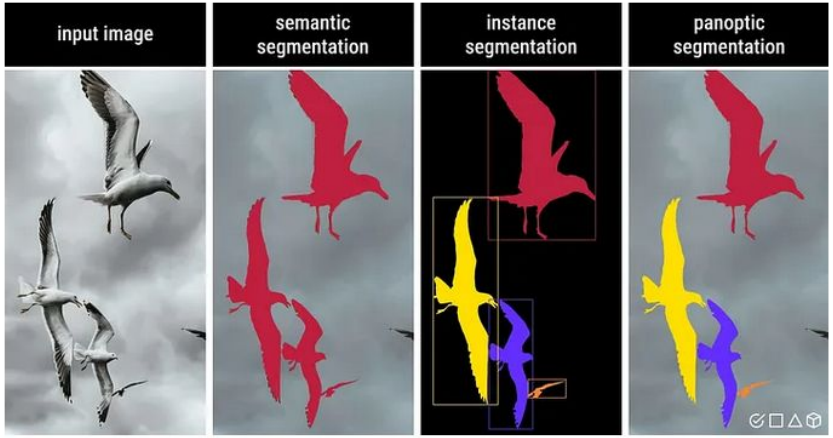


Image source