INTRODUCTION TO IMAGE PROCESSING AND COMPUTER VISION

SEGMENTATION

Rafał Jóźwiak
R. Jozwiak@mini.pw.edu.pl
Faculty of Mathematics and Information Science
Warsaw University of Technology

SEGMENTATION

- selecting regions or objects
- autonomous (automatic) segmentation is one of the most difficult tasks in image processing and computer vision
- very important for other processing tasks (e.g. features extraction, objects classification or recognition etc.)
- segmentation – mid-level representation for higher level tasks

SEGMENTATION

- different levels of segmentation
  - very general – foreground (something important) and background (the rest of the image which is not important)
  - detailed - mutually exclusive regions (different regions) to which we can subsequently attach meaningful labels (different meanings)
- segmentation methods focus on objects/regions with some homogeneity within themselves or heaving some contrast between object/regions boundaries
- homogeneity and contrast measures can include such a features as gray level, color, texture etc.

SEGMENTATION

- generally a one „correct” segmentation for an image do not exist
- no quantitative image segmentation performance metric has been developed (gold-standard often is defined by expert)
- segmentation strongly depends on the task we are doing (what type of regions we want to identify, what we want to do with that segmented regions etc.)
- segmentation often requires combination of different segmentation techniques
- after segmentation, we can also analyze the shape of objects

SEGMENTATION METHODS

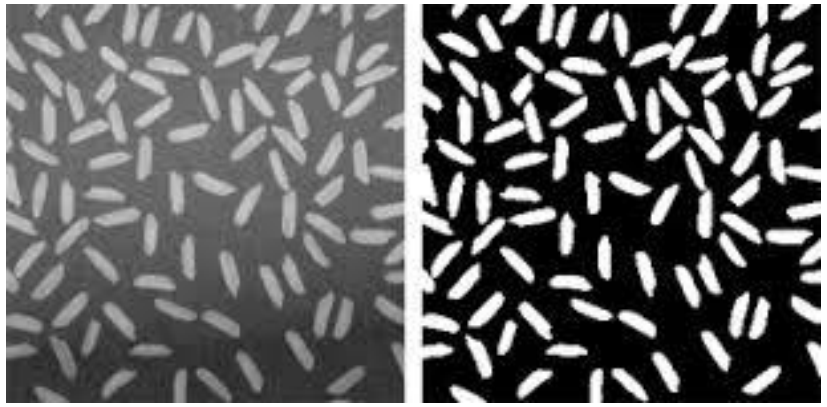
- several types of elementary segmentation methods (but not only):
  - pixel-based segmentation
    - use the gray values (colors) of the individual pixels
  - edge-based methods
    - detect edges and use edge information/follow edges
  - region-based methods
    - analyze regions homogeneity
  - model-based segmentation
    - utilize information about objects shape (shape should be known/assumed)

SEGMENTATION METHODS

- several types of elementary segmentation methods (other view):
  - automatic segmentation
    - without any human interaction
  - semi-automatic segmentation
    - some human interaction, e.g. initial point or shape
  - manual segmentation
    - performed by the human (with a small help of the computer)

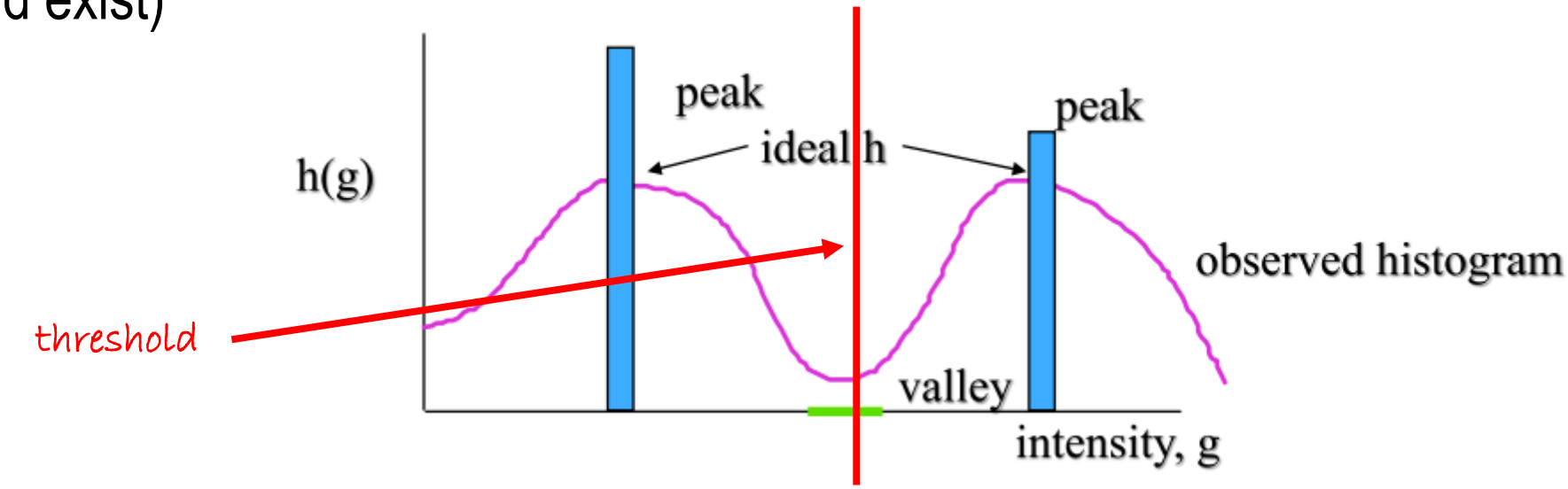
PIXEL-BASED SEGMENTATION -THRESHOLDING

- conceptually the simplest approach for segmentation
- selection of a single threshold for an image (object/background) – bilevel/binary thresholding
- we assumed that object pixels and the background pixels have nonoverlapping grey levels (colors) – luminance as a simple discriminating feature
- examples usage include handwritten and typewritten text, microscope biomedical samples



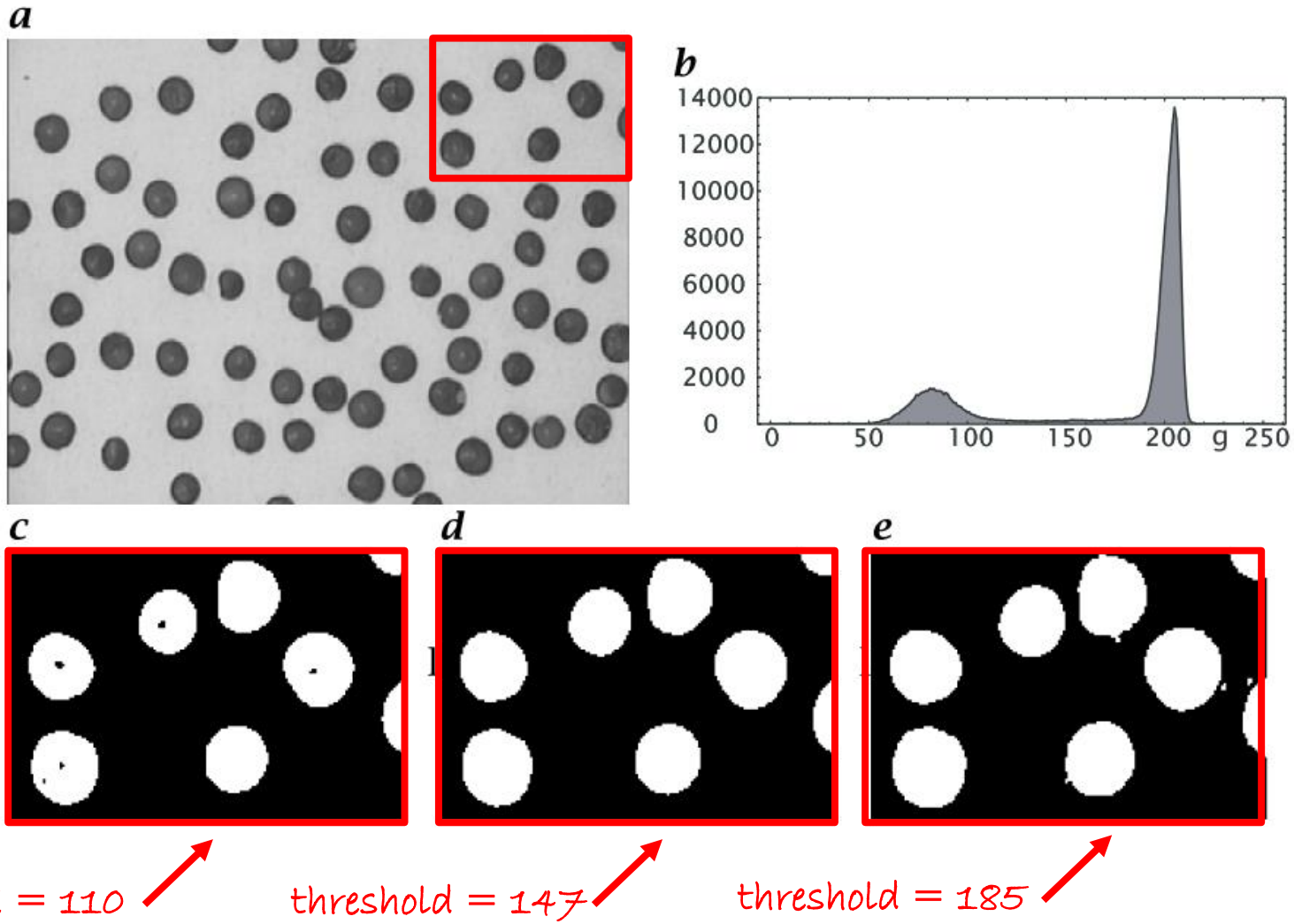
HISTOGRAM BASED THRESHOLDING

- histogram can be used for the purpose of threshold selection
- we assumed that histogram of some feature (e.g. pixel brightness in simplest case) shows a bimodal distribution with two distinct maxima
- we cannot expect that the probability for gray values between the two peaks will be zero (intermediate values filling the histograms in between the values for object and background exist)



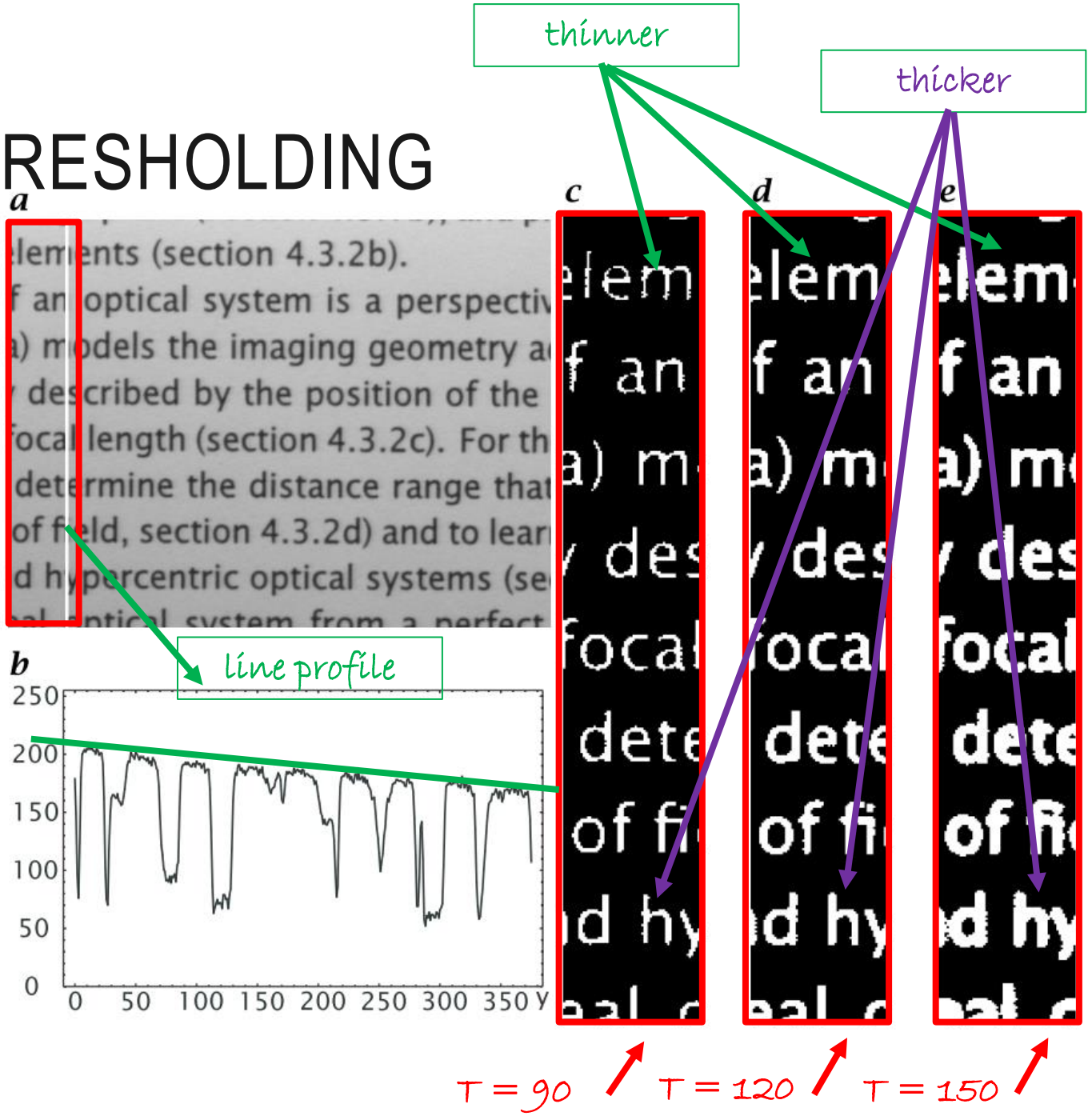
HISTOGRAM BASED THRESHOLDING

- if both the background and the object show rather uniform gray values a range of thresholds provide a quite good segmentation results



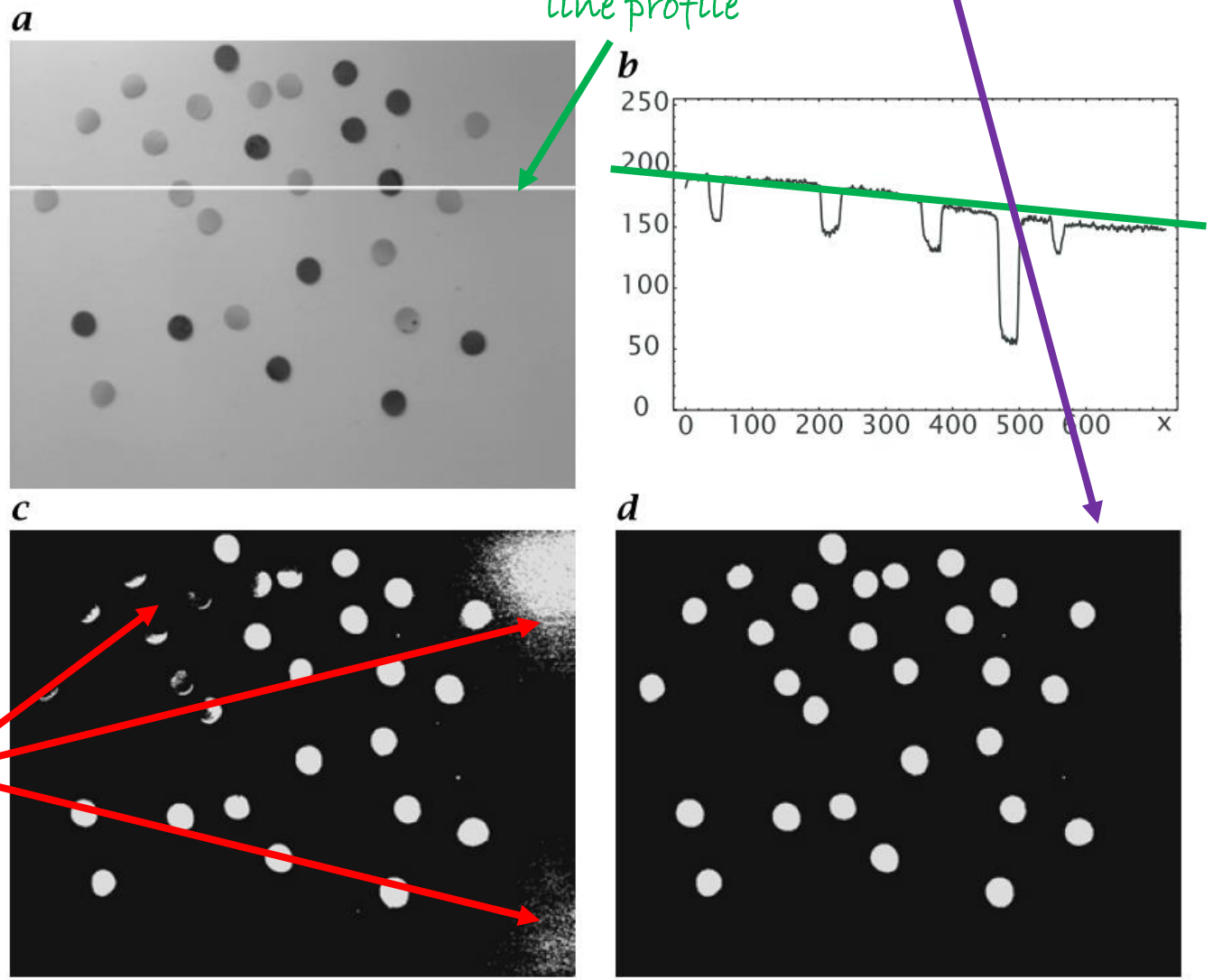
HISTOGRAM BASED THRESHOLDING

- problems occurs when the background is not uniform, or if objects with different gray values exist
- such a bias might be acceptable for some applications (e.g. recognition of letters as on example on the right) but affects object size and related parameters



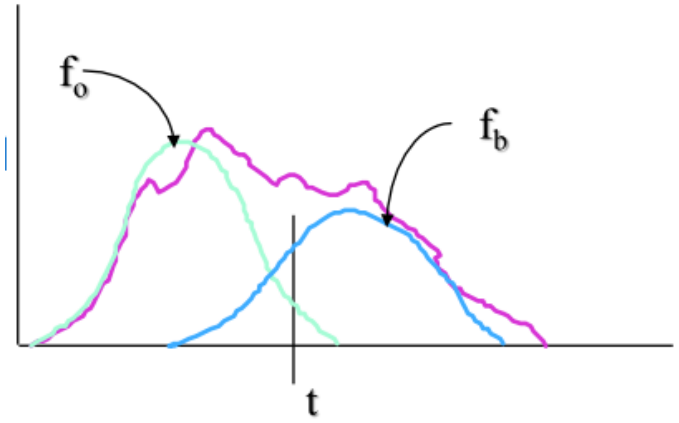
HISTOGRAM BASED THRESHOLDING

- some preprocessing task (e.g. inhomogeneous background/luminance correction, contrast correction/stretching etc.) might help with global thresholding
  - unfortunately not always works !



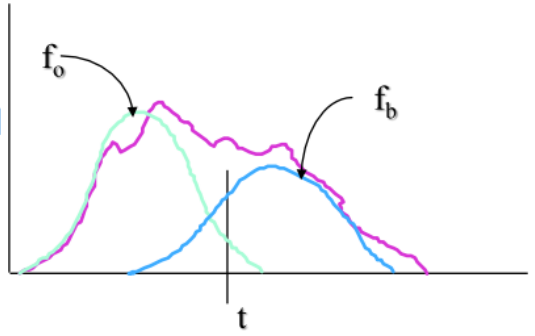
HISTOGRAM BASED THRESHOLDING

- selection of threshold should be done automatically (in ideal case)
- different techniques exist for automatic threshold selection
  - mean value (simples case)
  - iterative threshold selection
    - mean value as an initial threshold, than statistics for background (mean value of background pixels t_b) and objects (mean value of background pixels t_o) are calculated
    - new threshold is set as a $(t_b + t_o) / 2$
    - this is repeated until no change of threshold is detected
  - exploiting edges pixels
    - histogram is calculated only for pixels having a high Laplacian
  - sets of methods using modeling of histogram distributions
    - selecting the low point between two histogram peaks
    - a Gaussian distribution is assumed (Gaussian mixture)



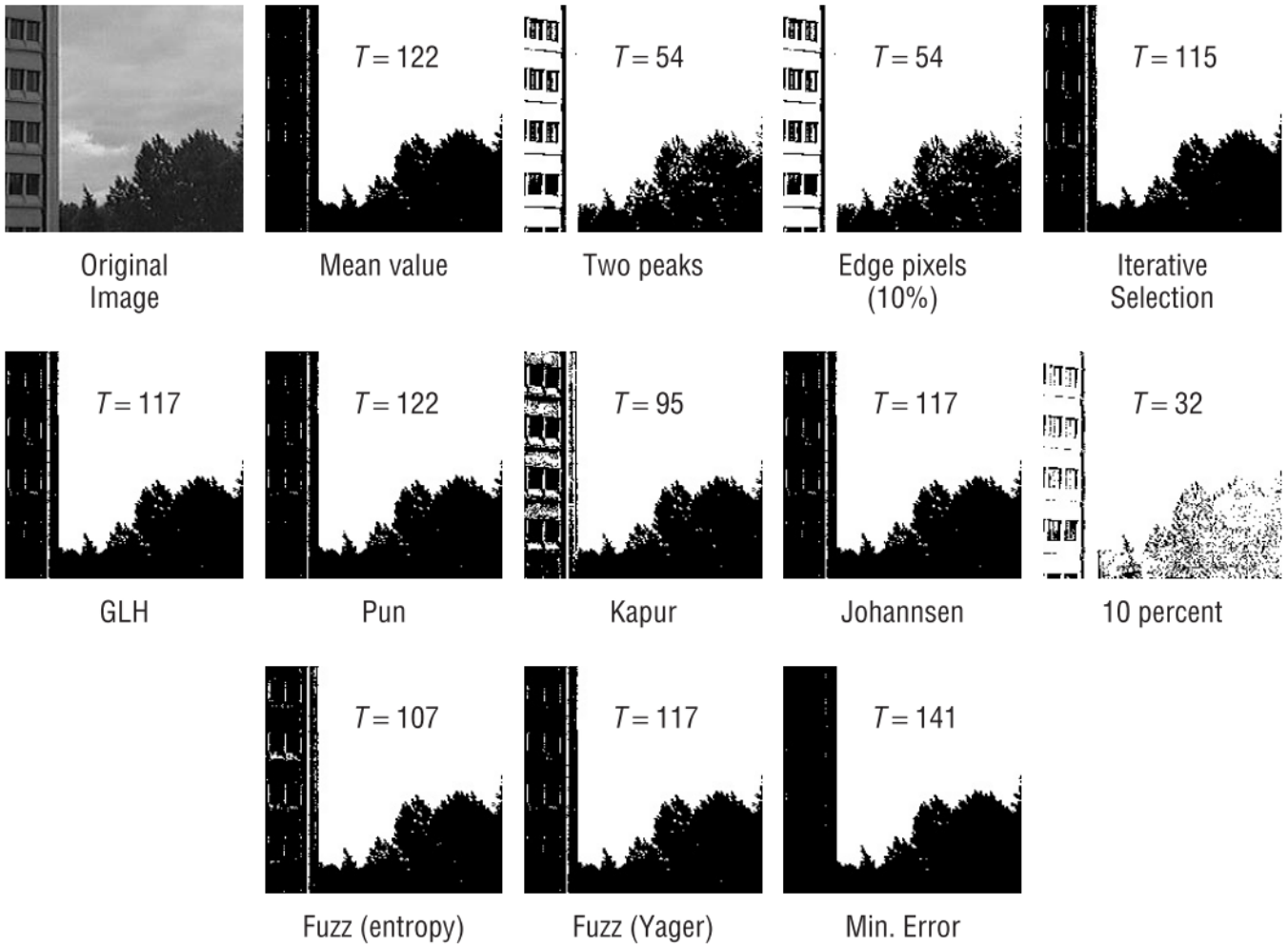
HISTOGRAM BASED THRESHOLDING

- sets of methods using modeling of histogram distributions
  - Otsu threshold - analysis of variance
    - object pixels and background pixels have different mean levels, and are random numbers drawn from one of two normal distributions
    - these distributions also have their own standard deviations and variances (variance is the square of the standard deviation)
    - an optimal (in some respects) threshold can be found by minimizing the ratio of the between-class variance to the total variance
  - using entropy
  - minimum error thresholding
  - using fuzzy sets

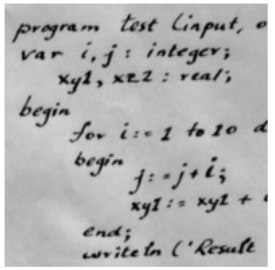


$$\frac{\sigma_b^2}{\sigma_t^2}$$

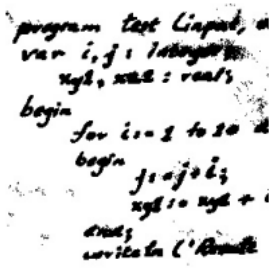
AUTOMATIC THRESHOLD SELECTION



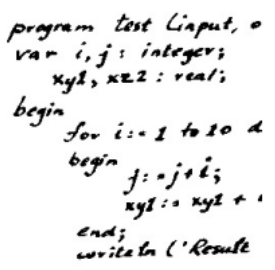
AUTOMATIC THRESHOLD SELECTION



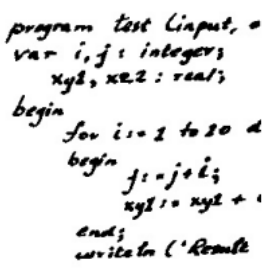
Original



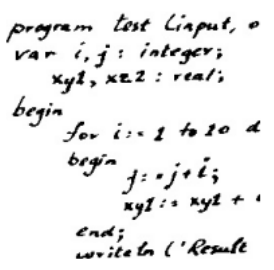
Mean value = 175



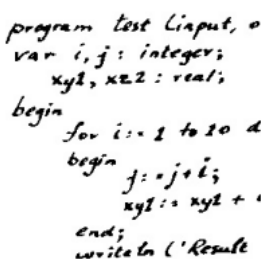
Two peaks = 132



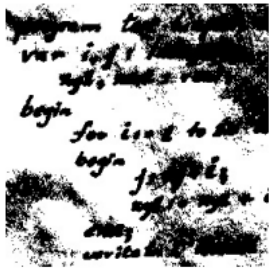
Edge
pixels = 156



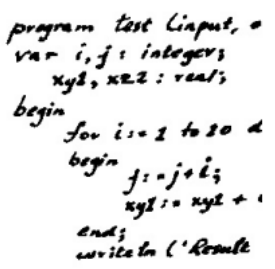
Iterative
Selection = 121



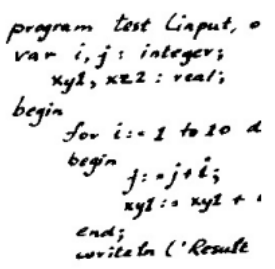
GLH = 122



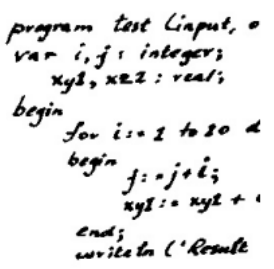
Pun = 184



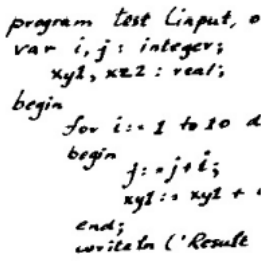
Kapur = 153



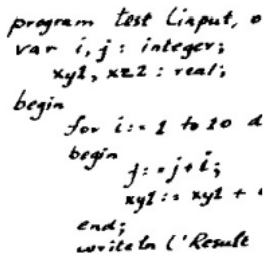
Johannsen = 132



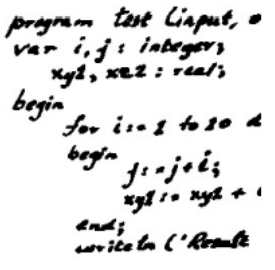
10 percent = 147



Fuzz
(entropy) = 131



Fuzz
(Yager) = 124



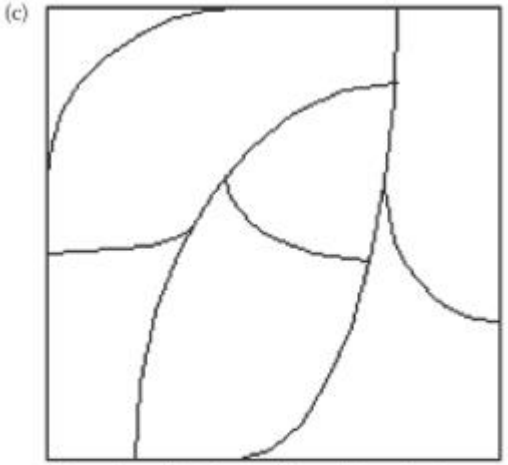
Min. Error = 166

HISTOGRAM BASED THRESHOLDING

- idea of global bilevel thresholding can be extended to multilevel thresholding
  - in the first stage the image is thresholder to separate brighter regions from darker regions (by locating a minimum between luminance modes of the histogram)
  - if these histograms are not unimodal, the parts are thresholder again (until the histogram of a part becomes unimodal)
  - several methods have been proposed for the selection of multilevel thresholds
- multilevel luminance thresholding concept can be also extended to the segmentation of color and multispectral images
  - segmentation is performed in separate color channels
  - different color spaces can be used (not only RGB)

EDGE-BASED METHODS

- edge-based segmentation is based on the fact that the position of an edge is given by an extreme of the first-order derivative or a zero crossing in the second-order derivative
- all we have to do is to search for local maxima in the edge strength and to trace the maximum along the edge of the object
- edge tracking
  - the maximum of the image is scanned line by line for maxima in the magnitude of the gradient
  - when a maximum is encountered, a tracing algorithm tries to follow the radient around the object until it reaches the starting point again
- Houg transform can be used for edge linking

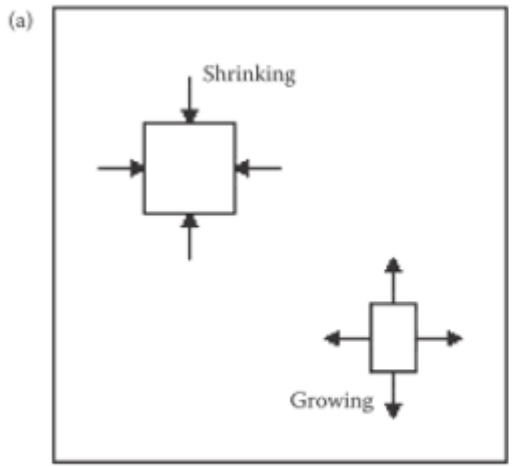


Boundary detection is often achieved using a differentiation operator to find lines or edges, followed by postprocessing to connect the points into borders.

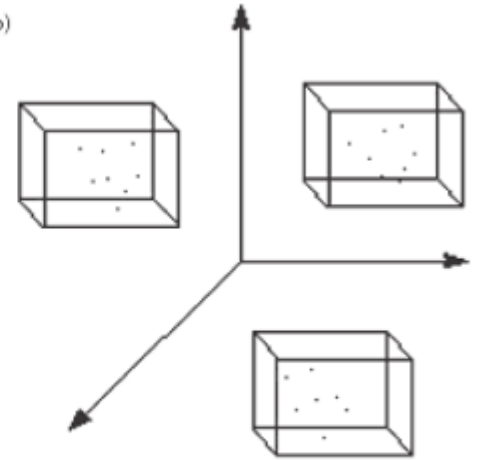
REGION-BASED SEGMENTATION

- region-based methods focus not on single pixel (judging solely on its gray value, independently of the context) but rather analyze pixel context utilizing the fact that an important characteristic of an object is its connectivity (relations between pixels in a local neighborhood)

- region-based segmentation can be divide into two main concepts:
  - grouping – identifying sets of coherent tokens/features in image (characterizing regions)
  - partitioning – dividing image into regions with coherent internal properties



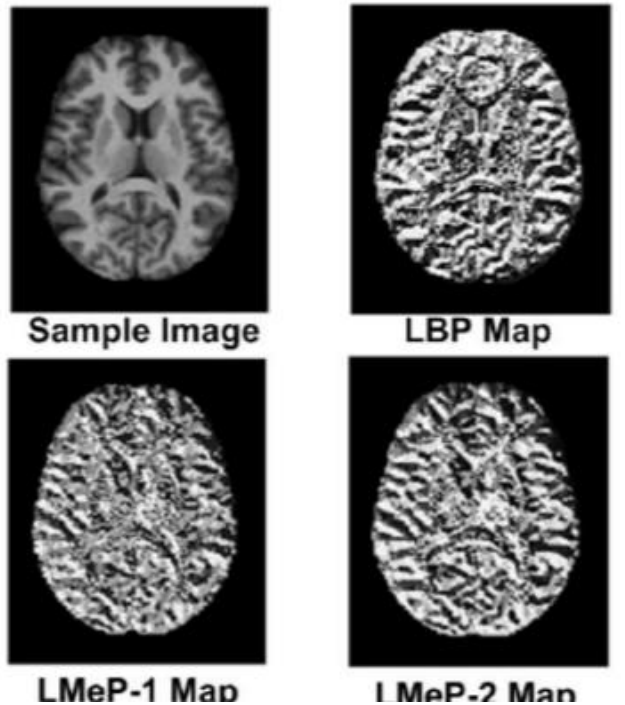
Region growing/shrinking is performed by finding homogeneous regions and changing them until they no longer meet the homogeneity criteria.



Clustering looks for data that can be grouped in domains other than the spatial domain.

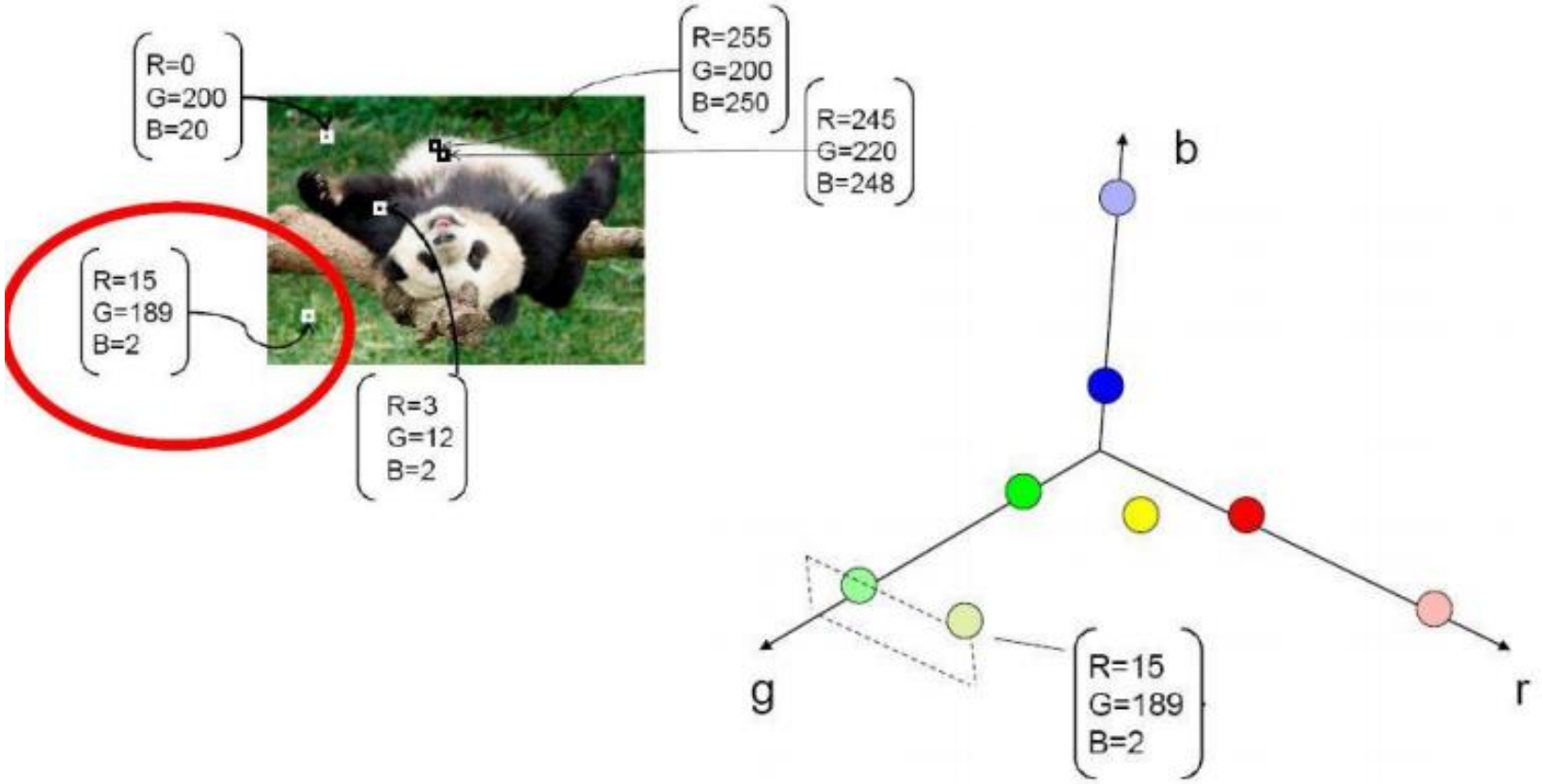
REGION-BASED SEGMENTATION

- in region-based segmentation we often work with feature image for segmentation
- features are calculated for each pixel within a small local context (depending an the mask size)
  - at the edges of the objects, where context includes pixels from both the object and the background, features cannot be easily calculated and used
  - often feature computation and segmentation are performed alternately (to solve that problem)
- in simplest case original image can be threated as a feature space
  - brightness or color is than essential image feature



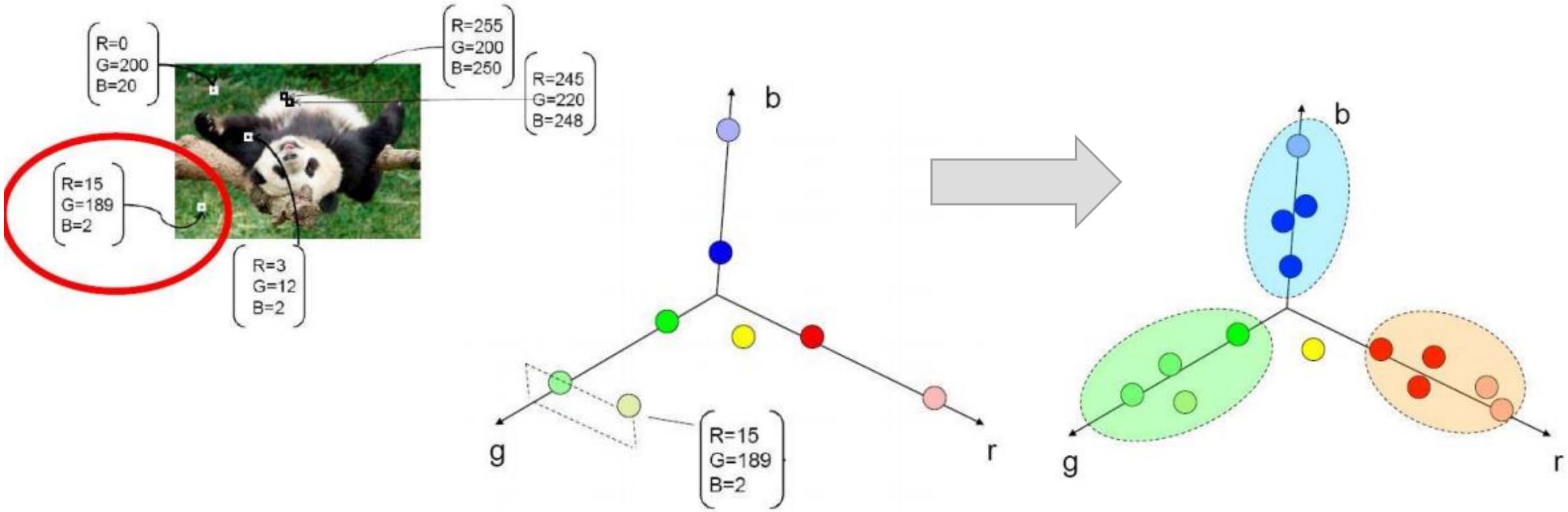
REGION-BASED SEGMENTATION - CLUSTERING

- clustering in RGB space (RGB as a simple feature space example)



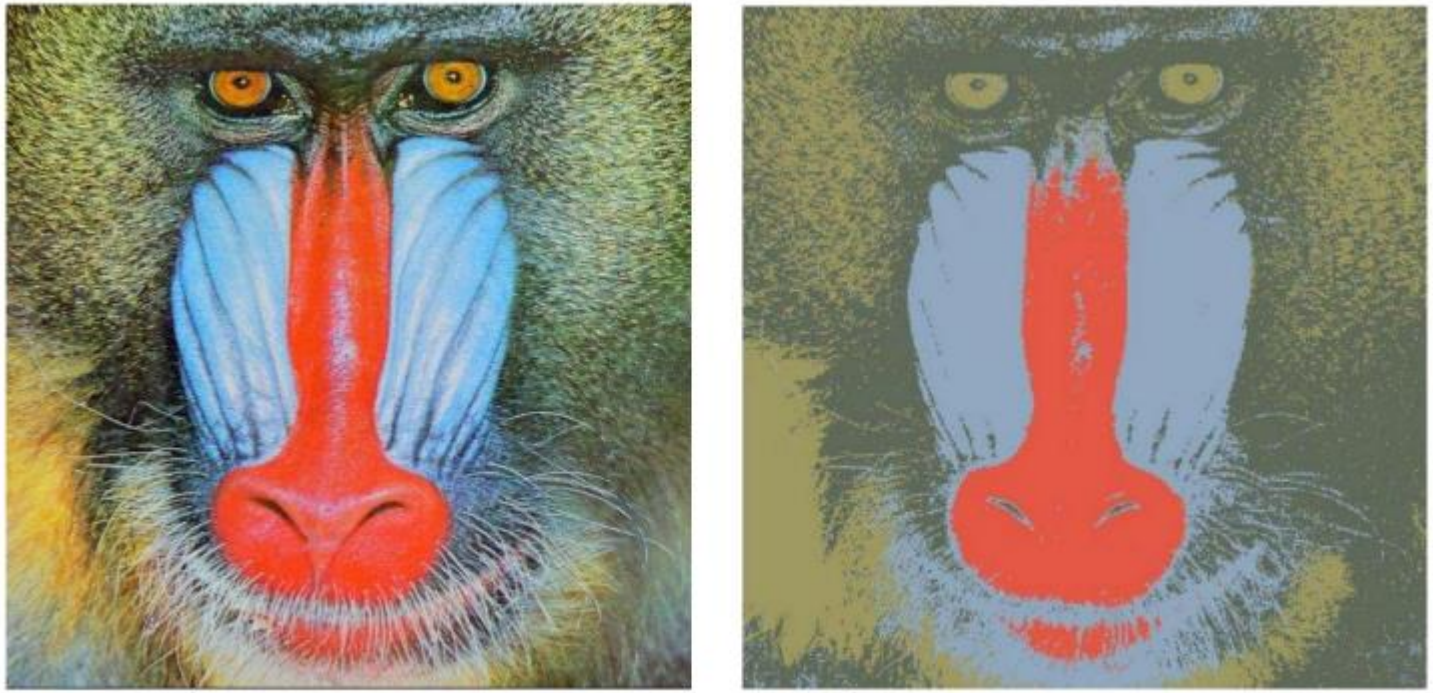
REGION-BASED SEGMENTATION - CLUSTERING

- clustering together tokens with high similarity (small distance in feature space)



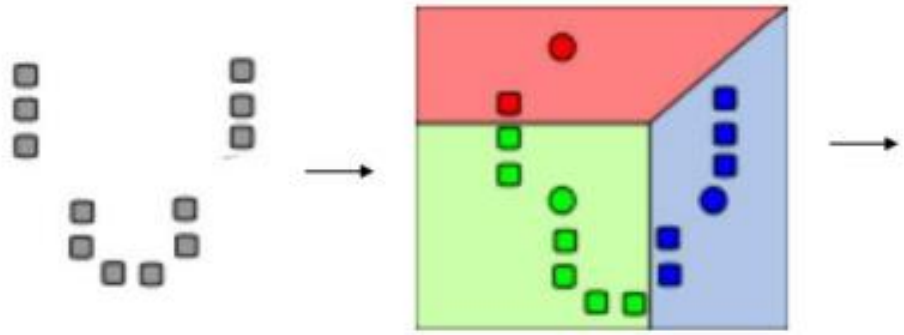
REGION-BASED SEGMENTATION - CLUSTERING

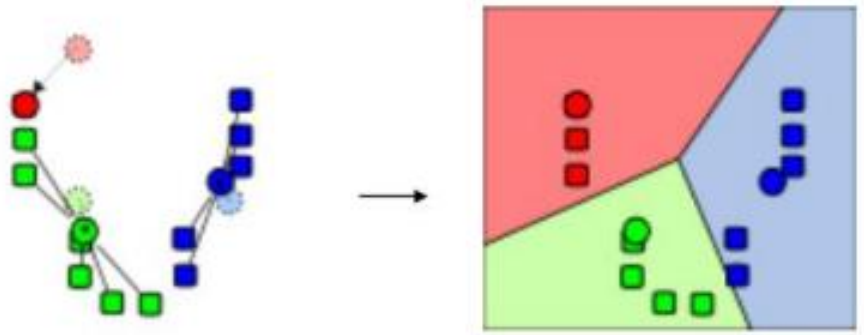
- clustering together tokens with high similarity (small distance in feature space)



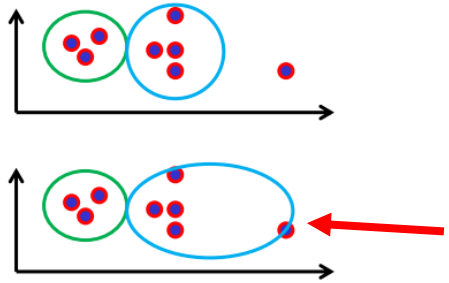
REGION-BASED SEGMENTATION - K-MEANS

- choose a fixed number of clusters
- choose cluster centers and point-cluster allocations to minimize error
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute (update) best cluster centers
  - if no mean has changed more than some ε, stop





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



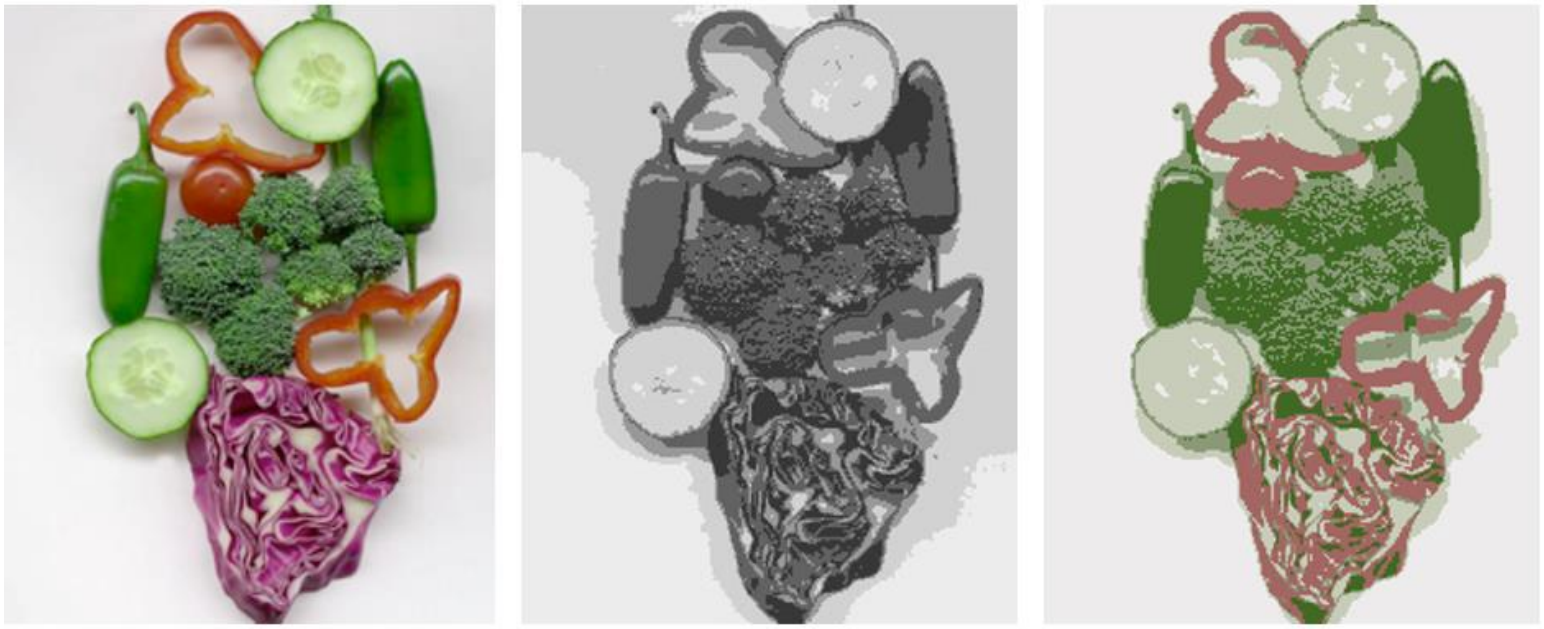
sensitive to outliers

REGION-BASED SEGMENTATION - K-MEANS

Image

Clusters on intensity

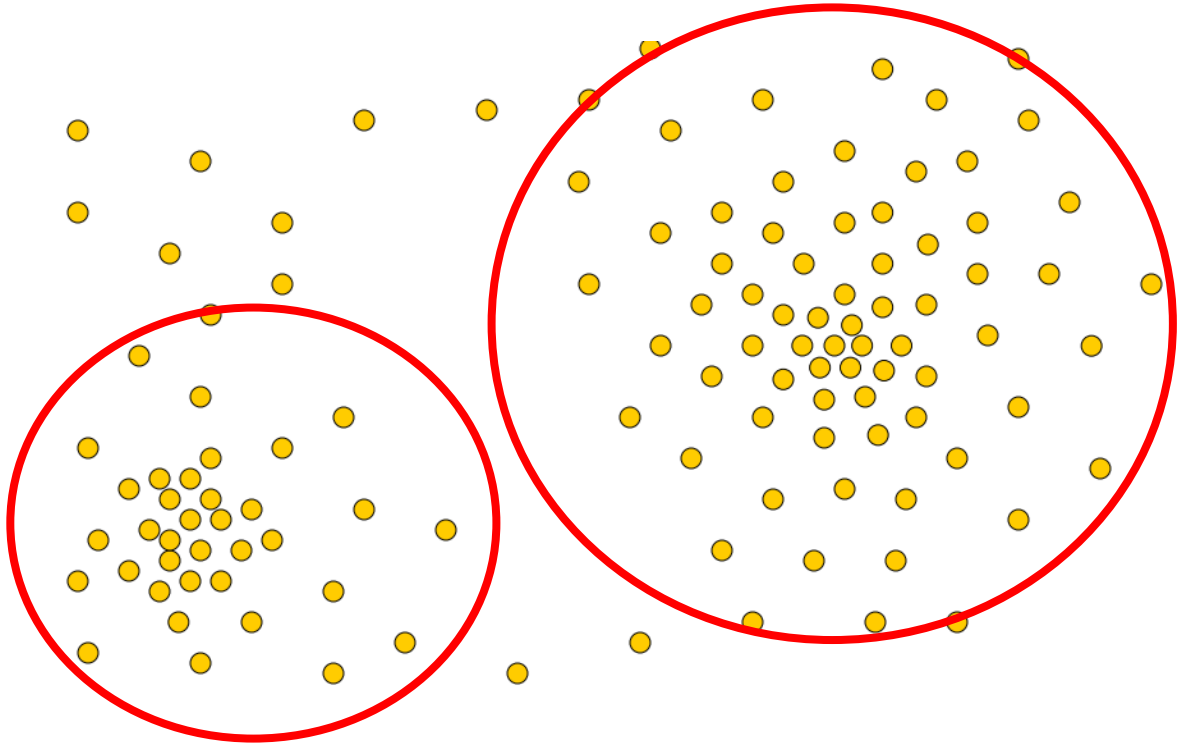
Clusters on color



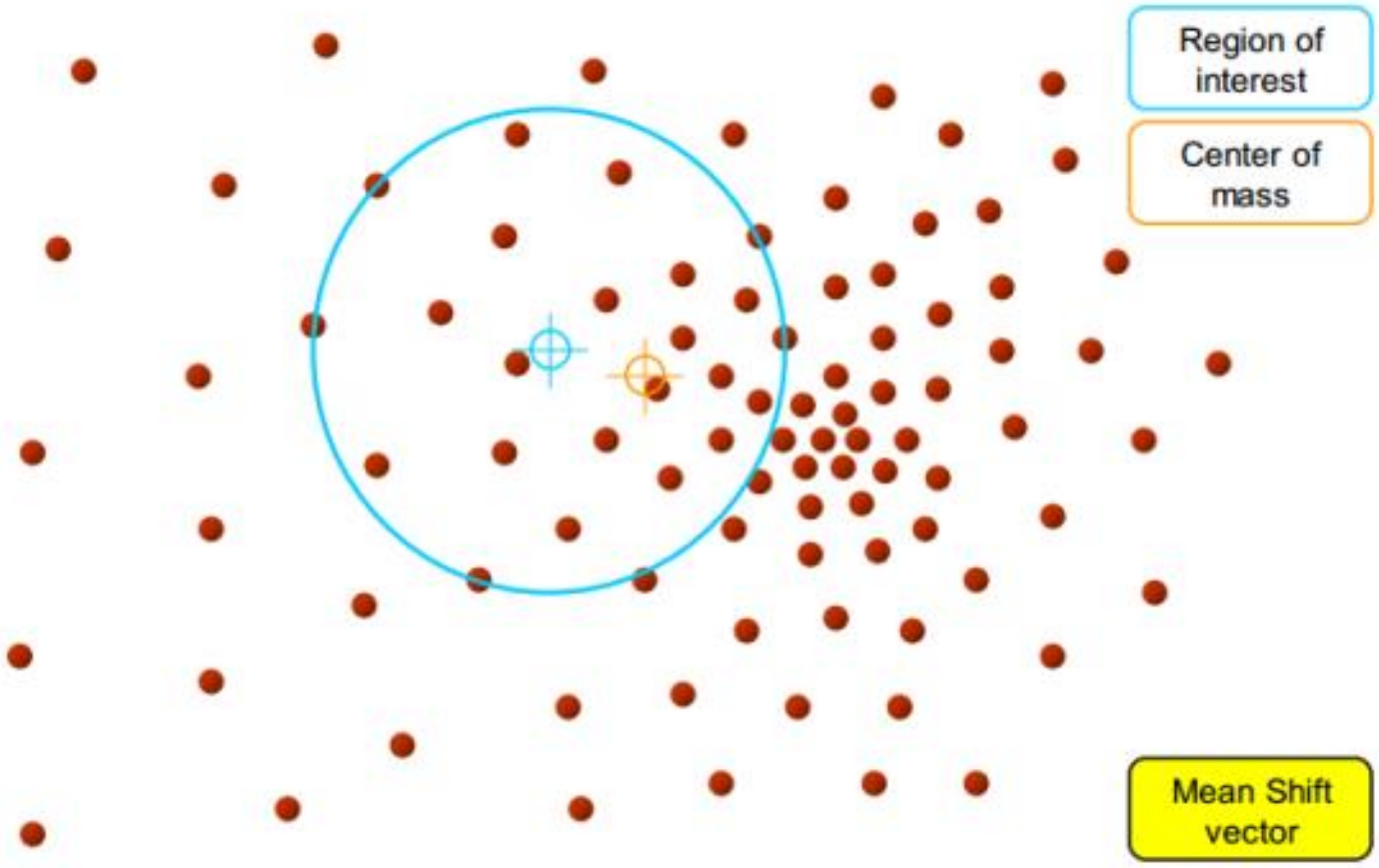
K-means clustering using intensity alone and color alone

REGION-BASED SEGMENTATION – MEAN SHIFT

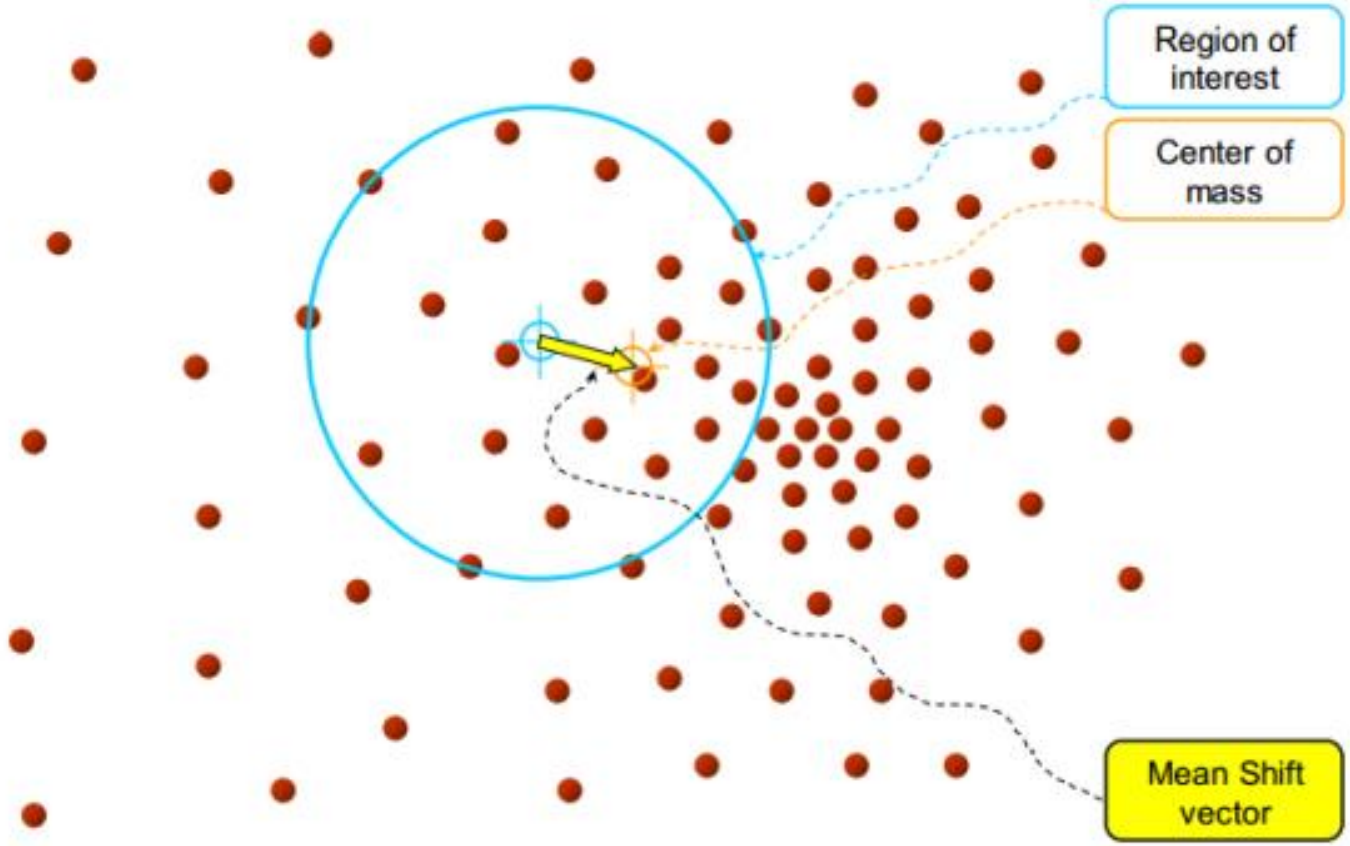
- segmentation as finding places with high density in feature space
- idea:
  - clusters are places where data points tend to be close together
  - assume data are IID (independent, identically distributed) samples from probability distribution
  - find local maxima in this probability distribution



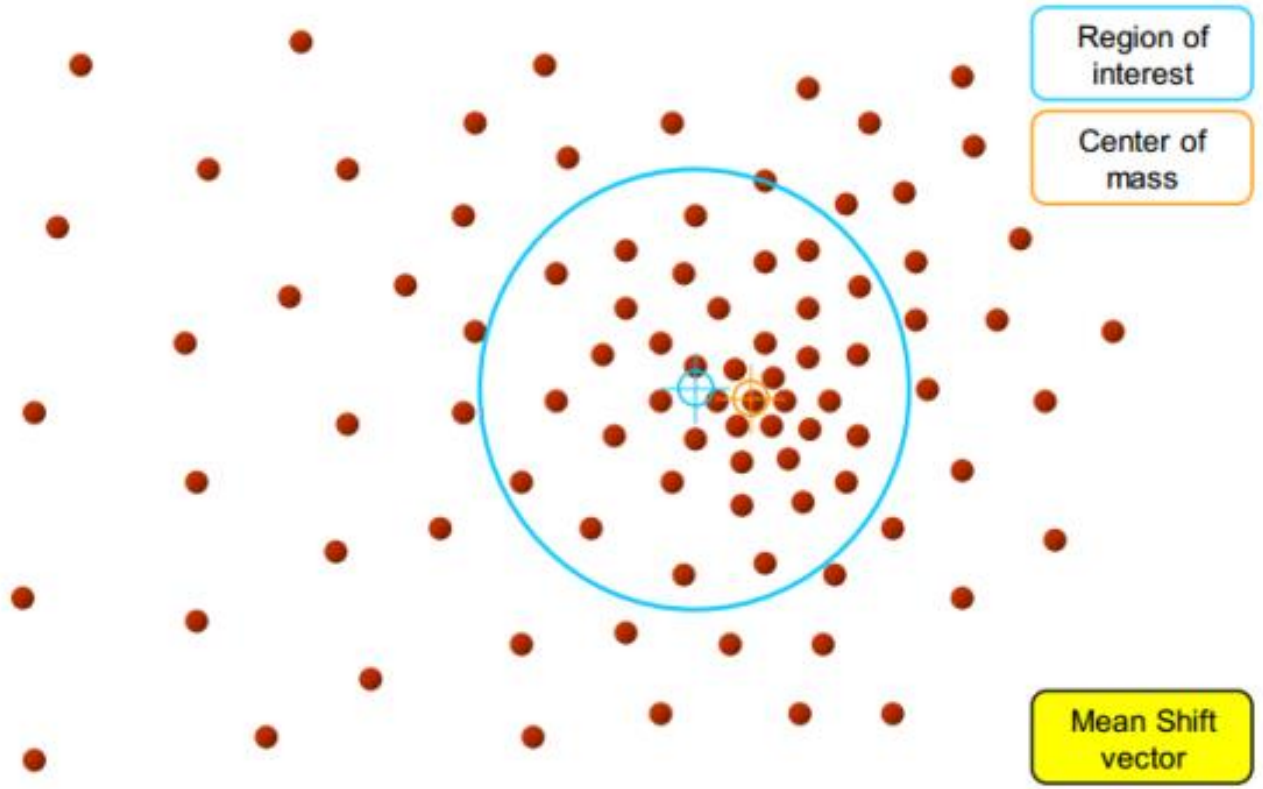
REGION-BASED SEGMENTATION – MEAN SHIFT



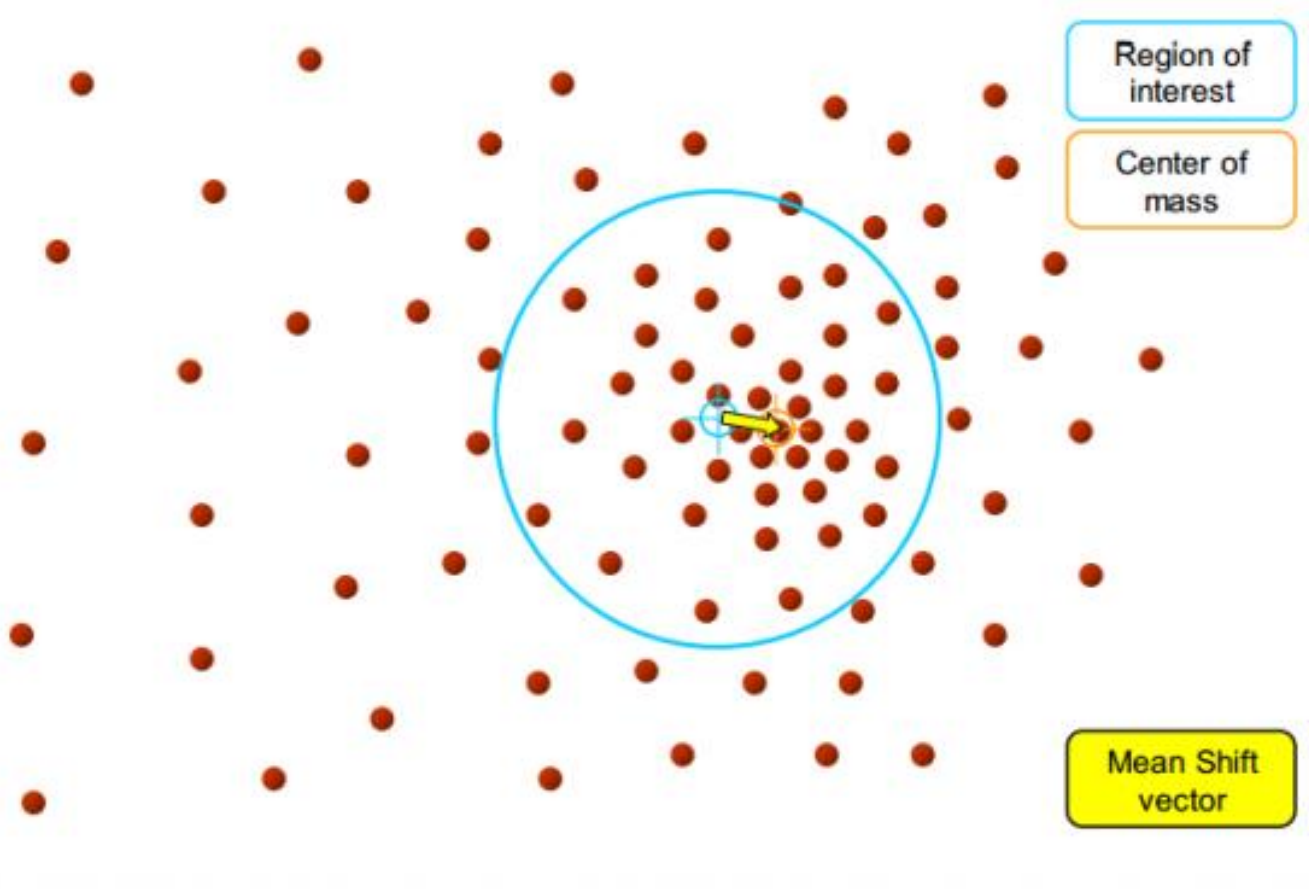
REGION-BASED SEGMENTATION – MEAN SHIFT



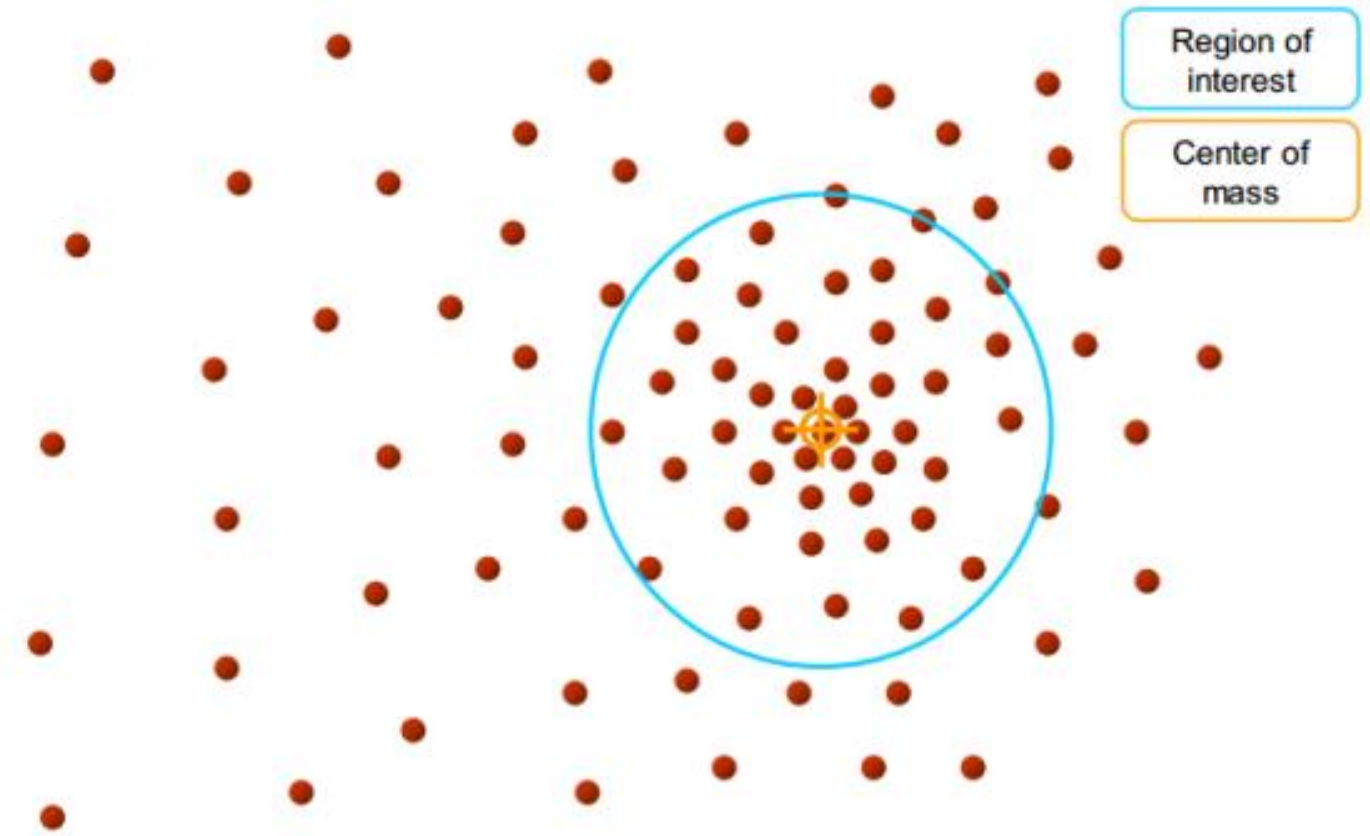
REGION-BASED SEGMENTATION – MEAN SHIFT



REGION-BASED SEGMENTATION – MEAN SHIFT

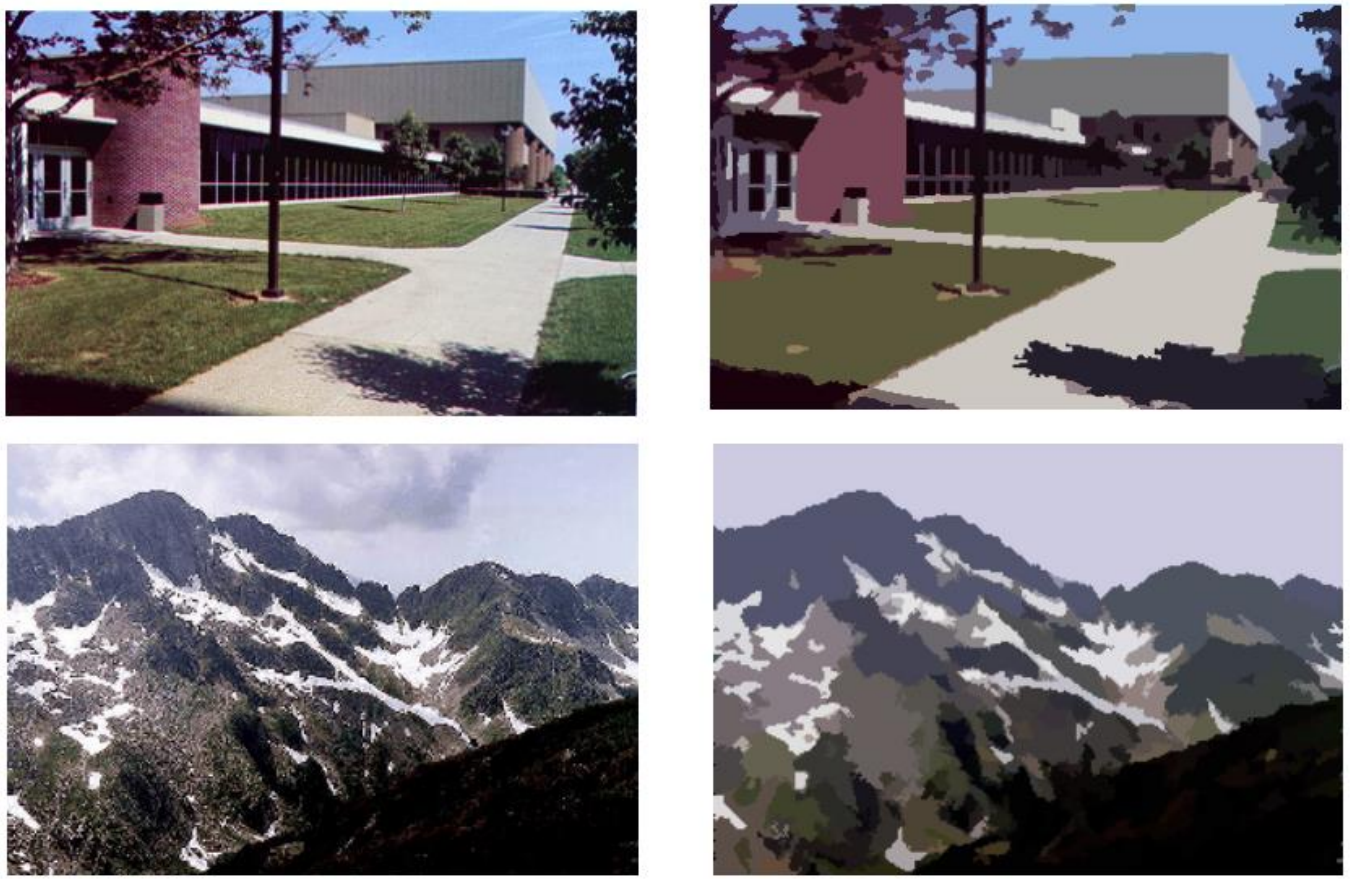


REGION-BASED SEGMENTATION – MEAN SHIFT



REGION-BASED SEGMENTATION – MEAN SHIFT

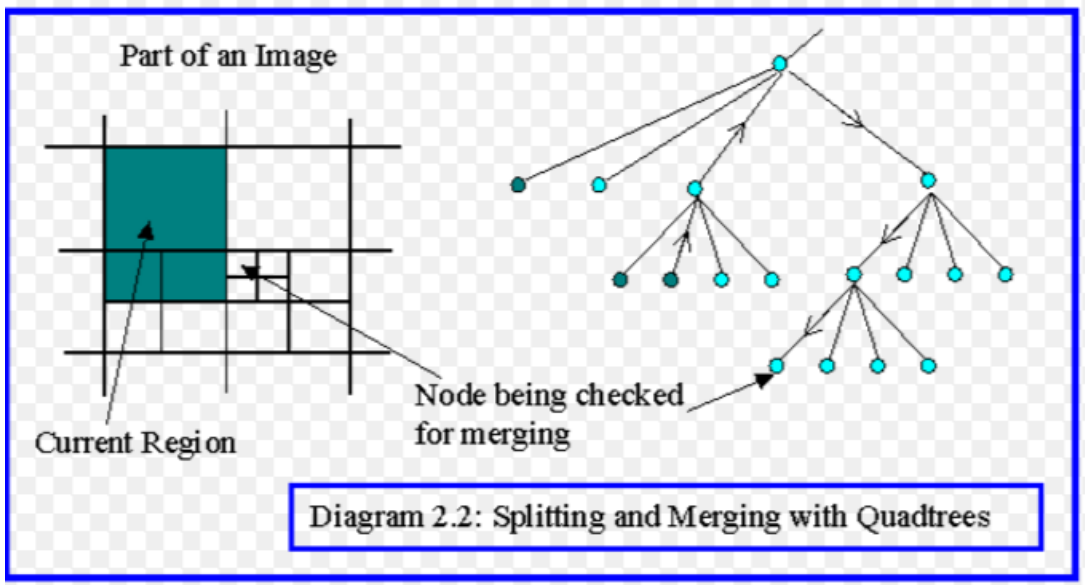
Mean shift segmentation results



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

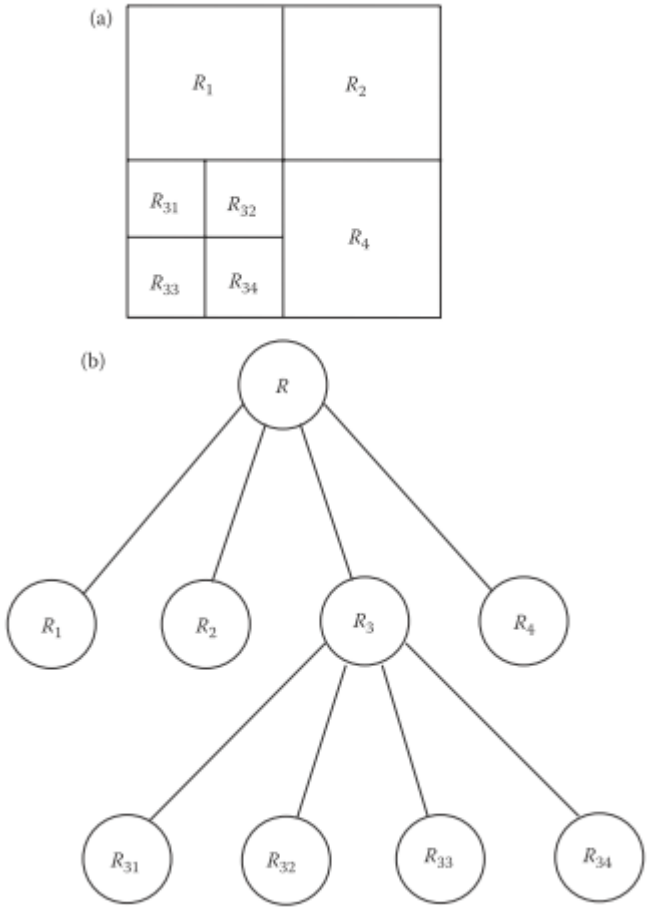
REGION-BASED SEGMENTATION – SPLIT AND MERGE

- split and merge, are referred to as state space techniques and use graph structures to represent the regions and their boundaries
  - quadtree is used as a data structure - makes the splitting and merging of regions easy
  - various split and merge algorithms have been described



REGION-BASED SEGMENTATION – SPLIT AND MERGE

- general algorithm:
  - define a homogeneity test
    - this involves defining a homogeneity measure, which may incorporate brightness, color, texture, or other application-specific information, and determining a criterion the region must meet to pass the homogeneity test
- split the image into equal sized regions
- calculate the homogeneity measure for each region
- if the homogeneity test is passed for a region, then a merge is attempted with its neighbor(s); if the criterion is not met, the region is split
- continue this process until all regions pass the homogeneity test



REGION-BASED SEGMENTATION – SPLIT AND MERGE

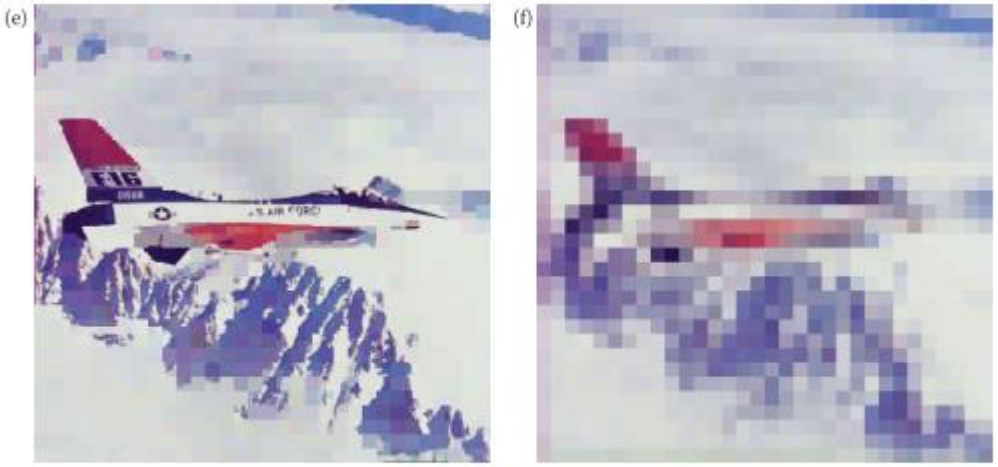


FIGURE 4.3-2
Segmentation with split and merge algorithm, various homogeneity criteria. The original image is 512 × 512 pixels, and the *Entry level* parameter was set to 6. (a) Original image, (b) Local mean versus global mean, (c) Local standard deviation versus global mean, (d) Variance with *Threshold* = 25, *Percentage* = 0.7 (70%), (e) Weighted gray-level distance with *Threshold* = 25, (f) Texture homogeneity with *Similarity* = 50, and *Pixel distance* = 2.

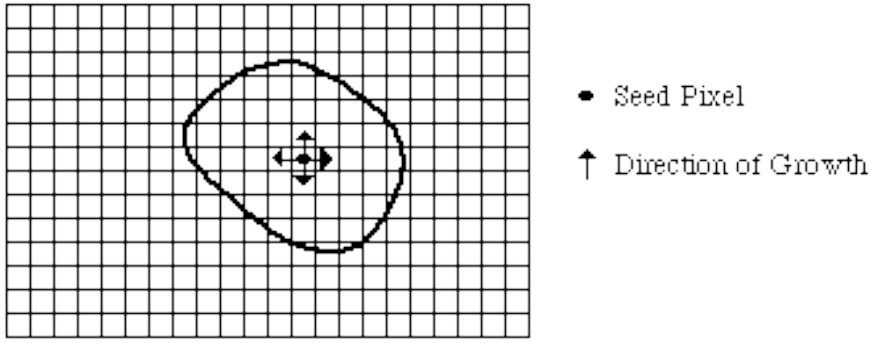
REGION-BASED SEGMENTATION – REGION GROWING

- region growing

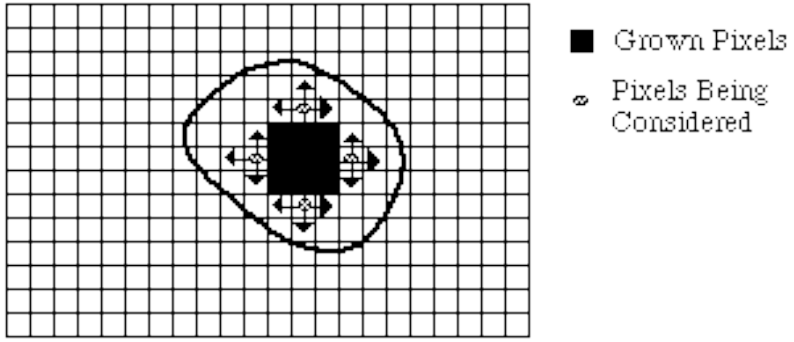
*a) Chose or determined a group of seed pixel which can correct*ly represent the required region;

b) Fixed the formula which can contain the adjacent pixels in the growth;

c) Made rules or conditions to stop the growth process



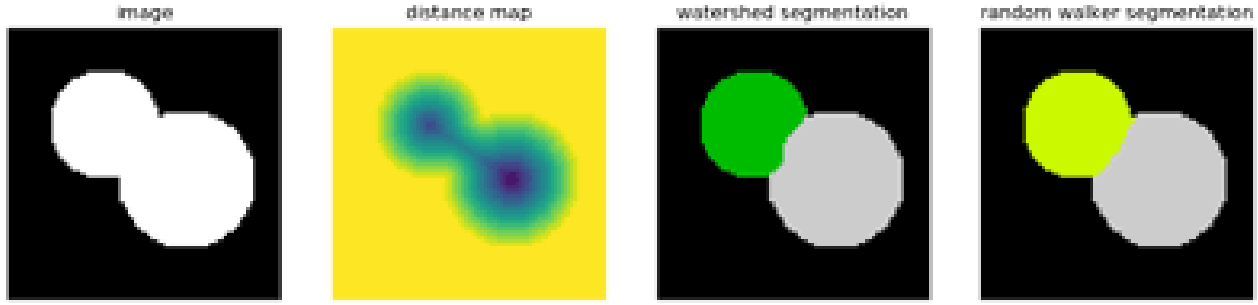
(a) Start of Growing a Region



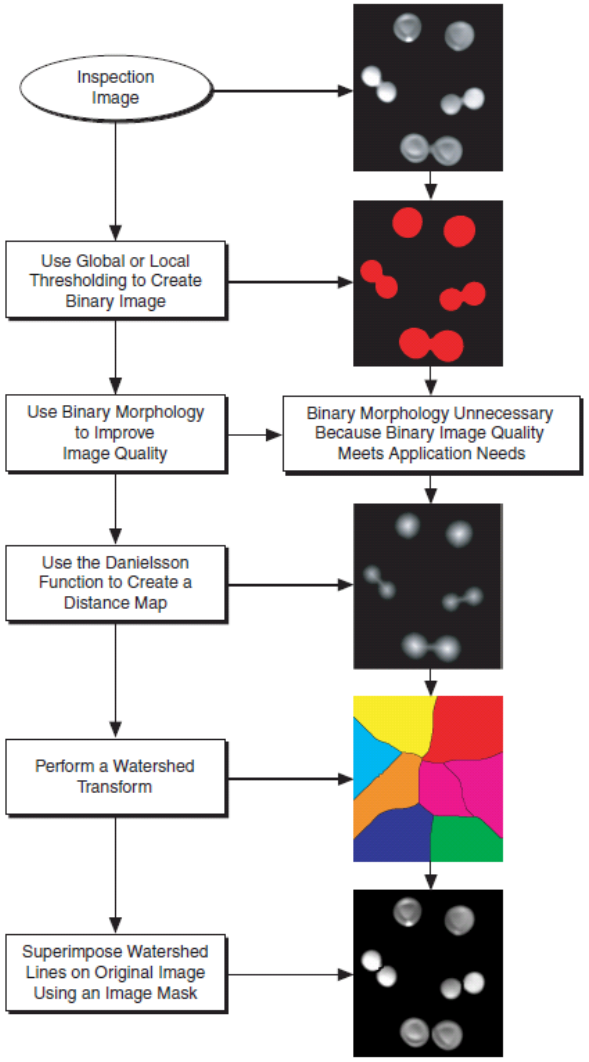
(b) Growing Process After a Few Iterations

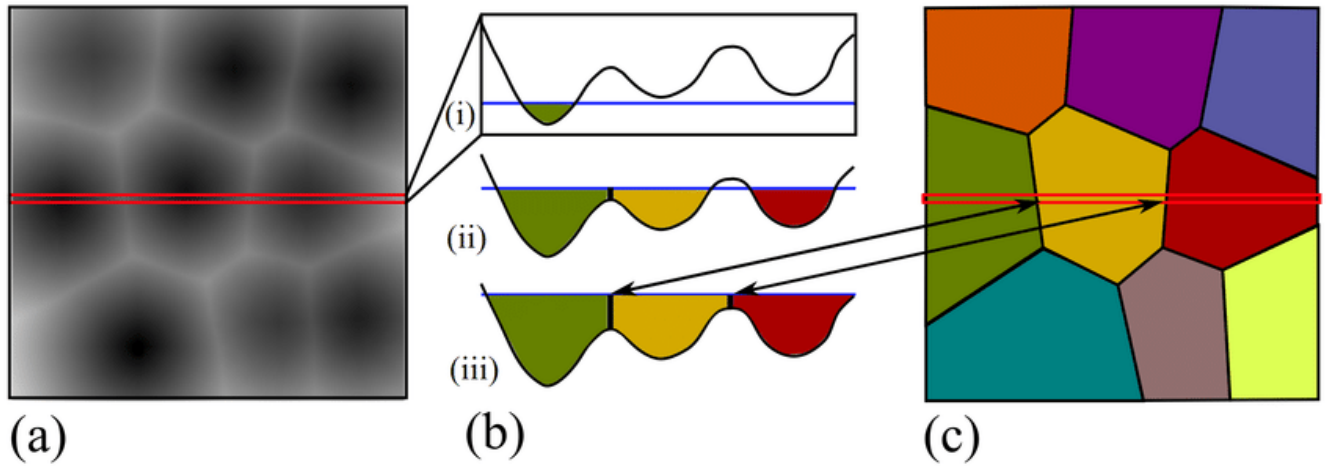
WATERSHED SEGMENTATION

- classical algorithm used for segmentation- separating different, often overlapping objects
- we stasrt with user-defined markers (e.g. some local maximas, but not always)
- watershed algorithm treats pixels values as a local topography (elevation)
- algorithm floods basins from the markers until basins attributed to different markers meet on watershed lines (objects usually meet)

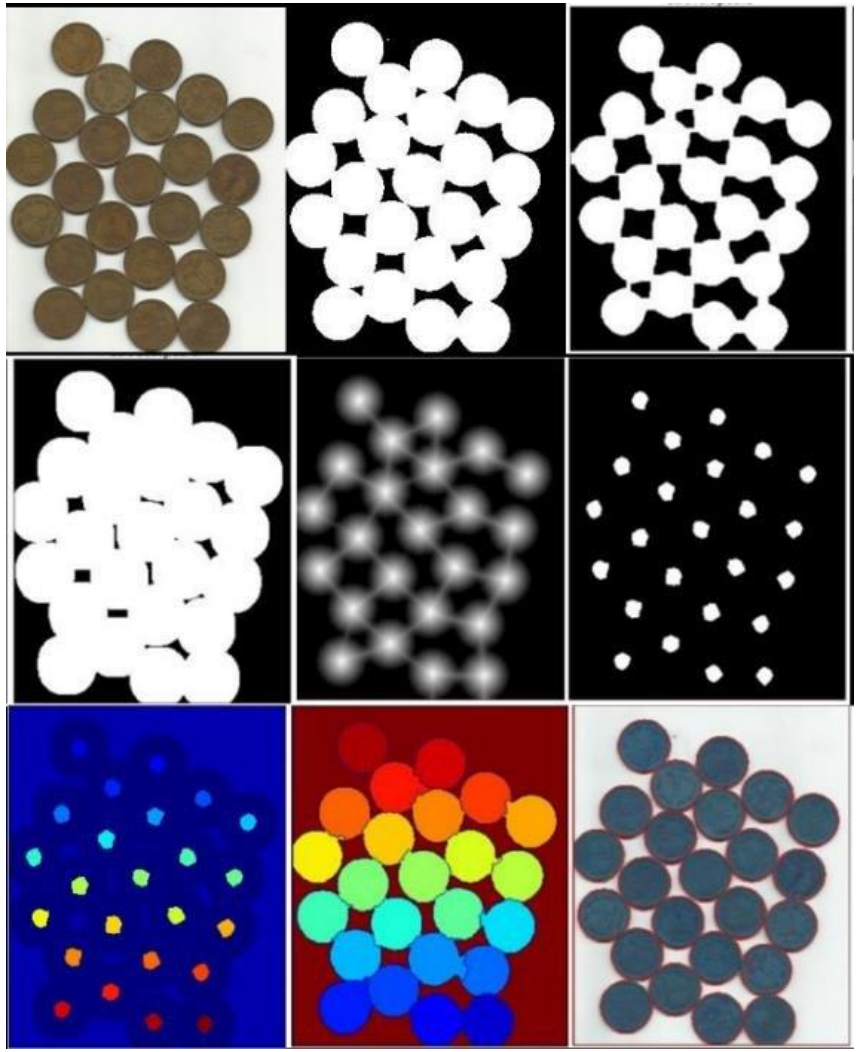


WATERSHED SEGMENTATION





WATERSHED SEGMENTATION



https://docs.opencv.org/master/d3/db4/tutorial_py_watershed.html