IMAGE SEGMENTATION

Aziz M. Qaroush

Birzeit University

First Semester 2015/2016

Outline

- Element of Image Analysis
- Segmentation Methods
  - Edge Based Methods
  - Thresholding
  - Region Growing
  - Split and Merge
  - K-Means
  - Mean Shift
  - Spectral Clustering
  - Active Contour
  - ………

Element of Image Analysis

Preprocess

Image acquisition, restoration, and enhancement

↓

Intermediate process

Feature extraction & Image segmentation

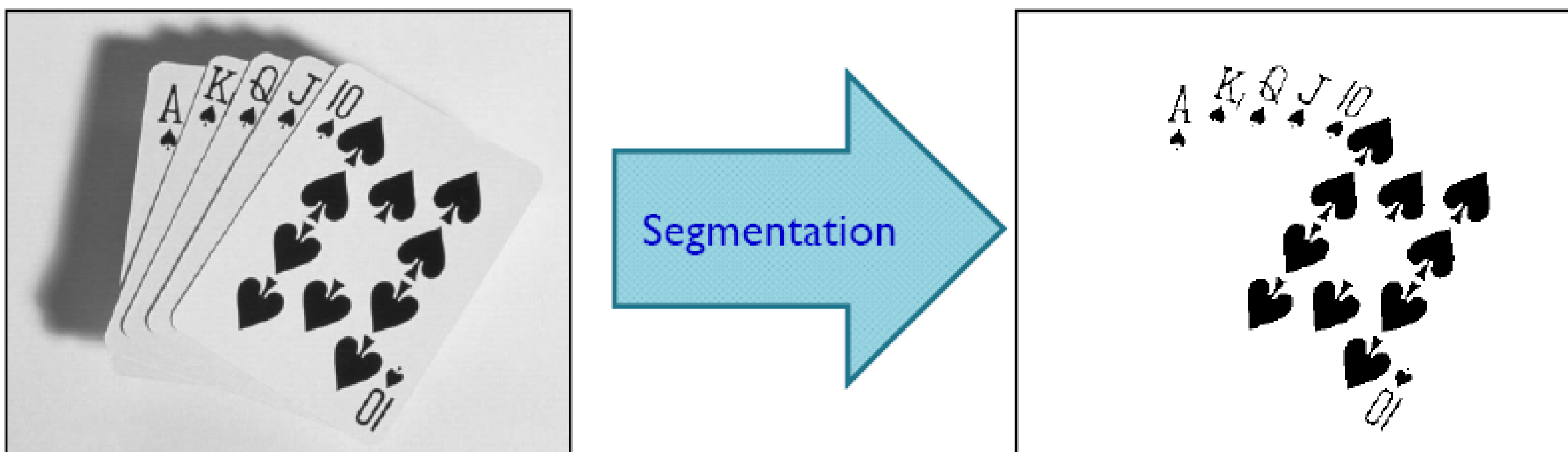
↓

High level process

Image interpretation and recognition

Fundamentals

- **What is image segmentation?**
  - The process of subdividing an image into its individual components/objects
  - A central operation in all machine vision applications
  - The level of subdivision depends on the problem in hand
  - It is nontrivial for complex images
  - The accuracy of the segmentation process determines the correct operation of computerized vision systems



Fundamentals

- Let R represents the entire spatial region occupied by the image. The segmentation processes subdivides R into n subregions $R_1, R_2, \ldots, R_n$ such that

(1) $\bigcup_{i=1}^{n} R_i = R$

(2) R_i is a connected set, i = 1,2,3,...,n

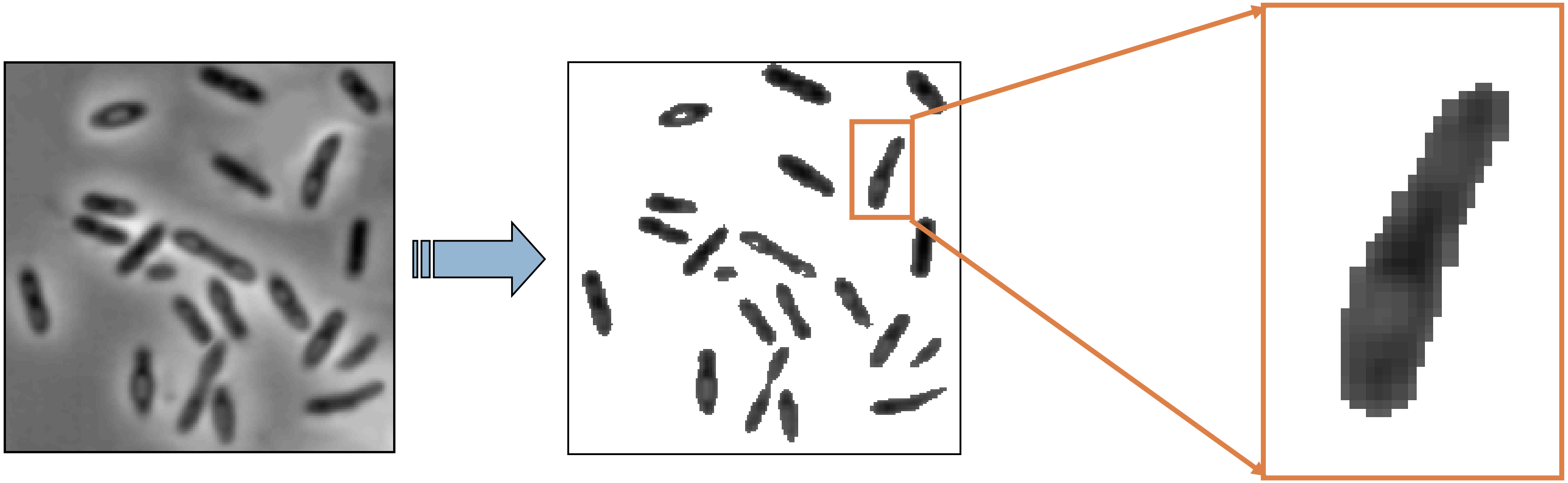
(3) $R_i \cap R_j = \varnothing$, for all i and j, i ≠ j

(4) $Q(R_i) =$ TRUE, for i = 1,2,3,...,n

(5) $Q(R_i \cup R_j) =$ FALSE, for any adjacent regions R_i and R_j

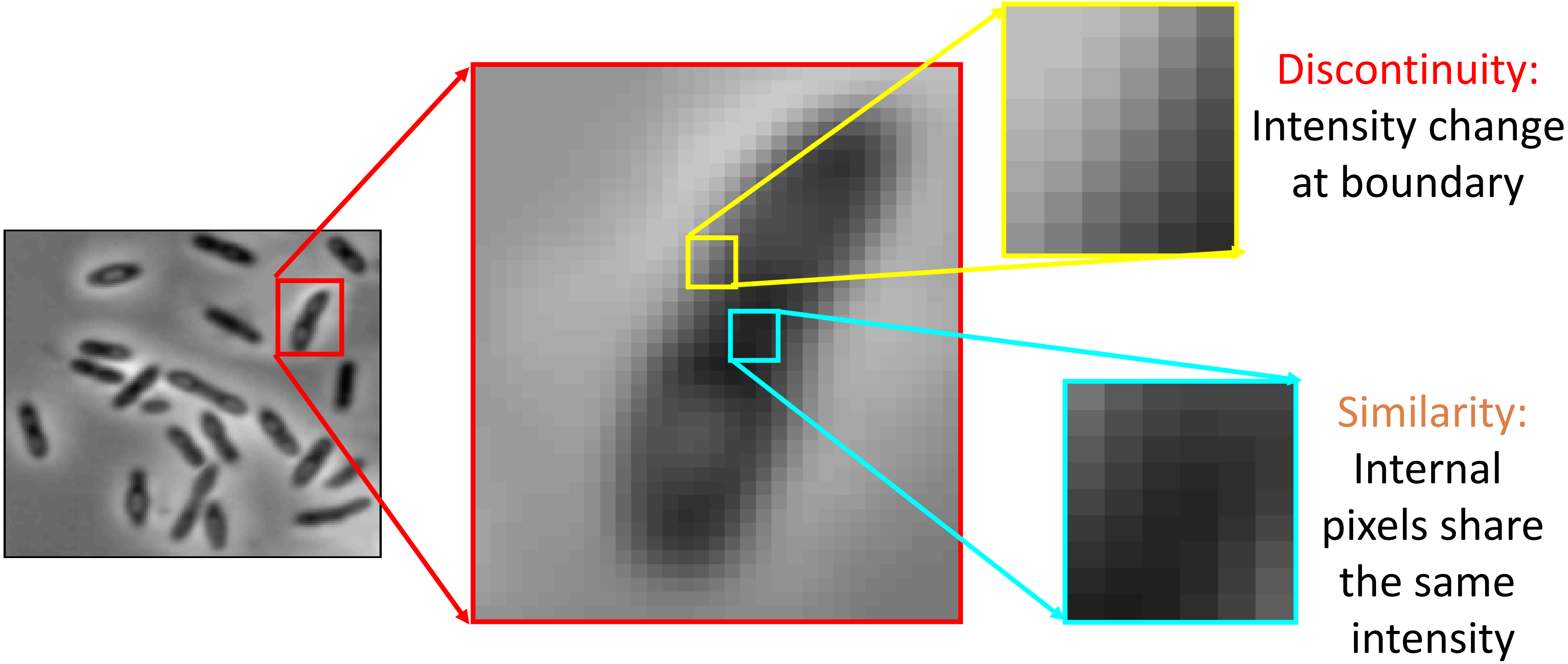
Fundamentals

- Importance of Image Segmentation
  - Image segmentation is used to separate an image into constituent parts based on some image attributes. Image segmentation is an important step in image analysis
    1. Image segmentation reduces huge amount of unnecessary data while retaining only importance data for image analysis
    2. Image segmentation converts bitmap data into better structured data which is easier to be interpreted

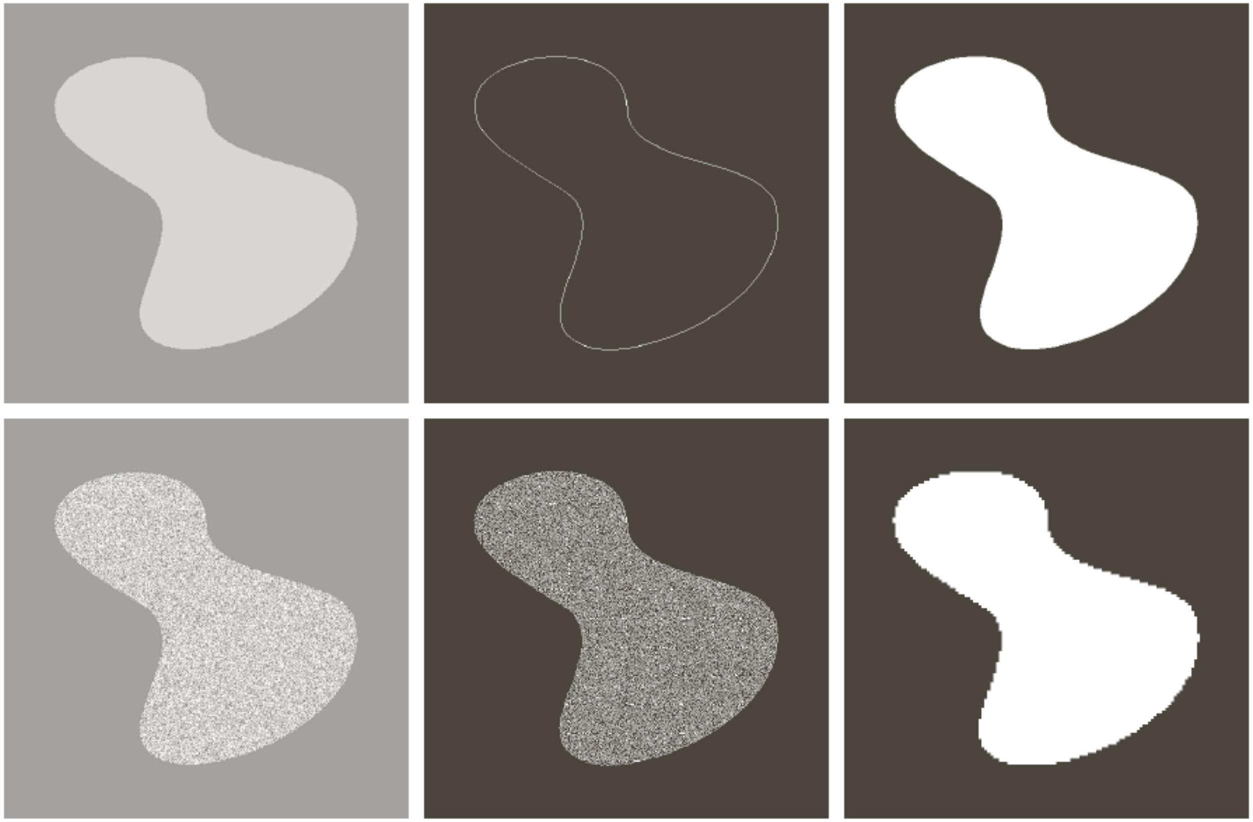


Fundamentals

- Image Attributes for Image Segmentation
  1. Similarity properties of pixels inside the object are used to group pixels into the same set.
  2. Discontinuity of pixel properties at the boundary between object and background is used to distinguish between pixels belonging to the object and those of background.



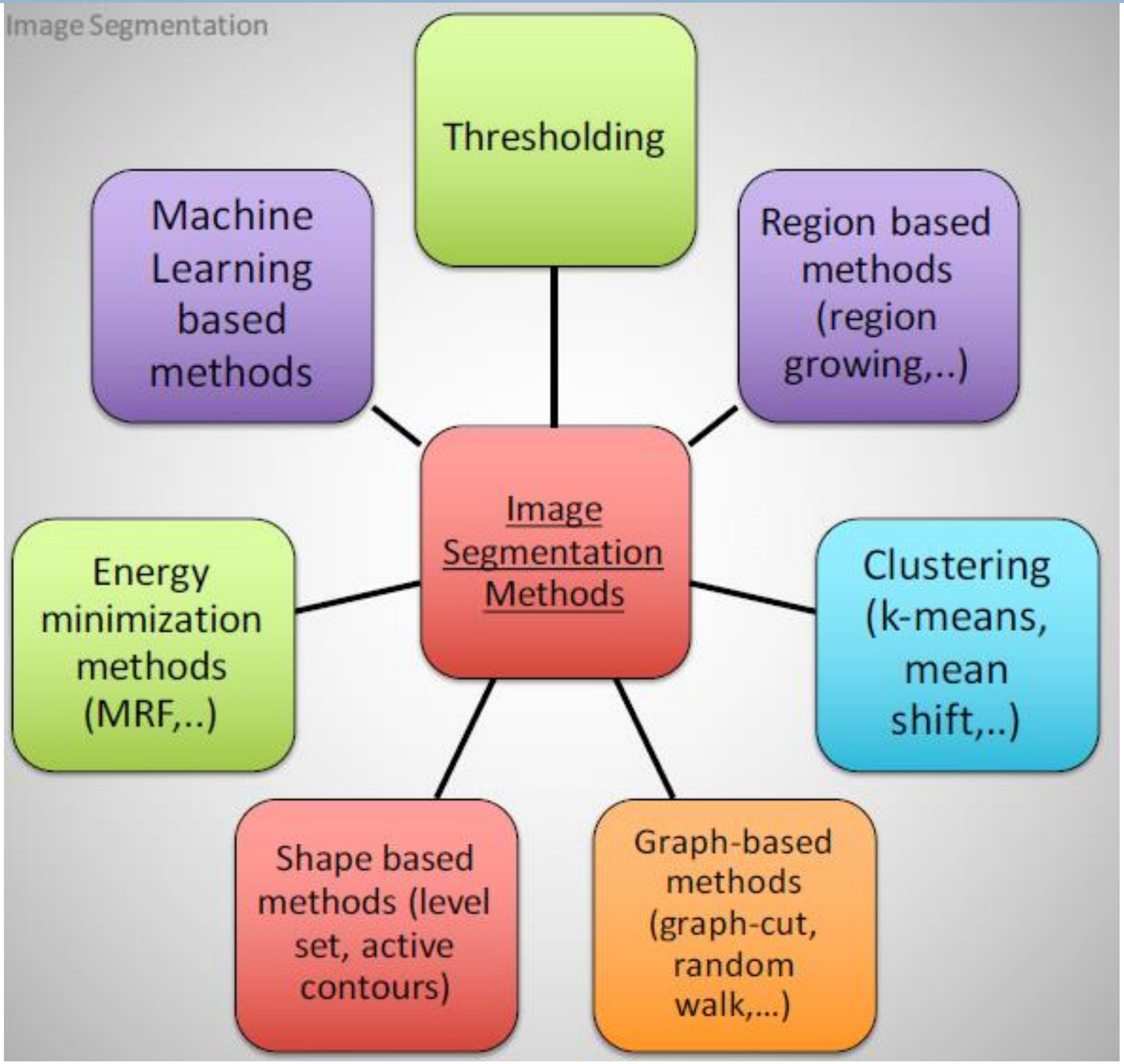
Fundamentals



a b c
d e f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Image Segmentation Techniques



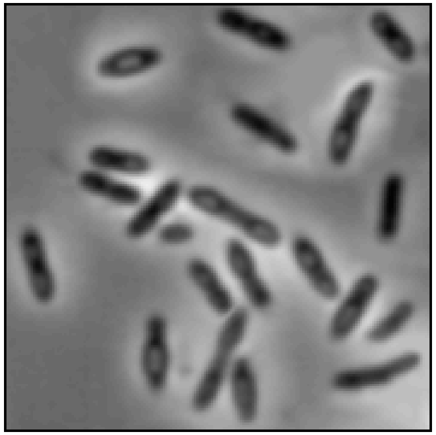
Edge Based Method

- Edge detection techniques transform images to edge images using the changes of grey tones in the images.
  - Edges are the sign of lack of continuity, and ending.
  - Edges are local changes in the image intensity and Edges occur on the boundary between two regions.
- Edge Detection Methods
  - Spatial Gradient Measurements on an image
    - Roberts Detection
    - Prewitt Detection
    - Sobel Detection
    - Canny
  - Edge Detection Soft Computer approaches
    - Fuzzy Logic Based Approach
    - Genetic Algorithm Approach
    - Neural Network Approach

Thresholding

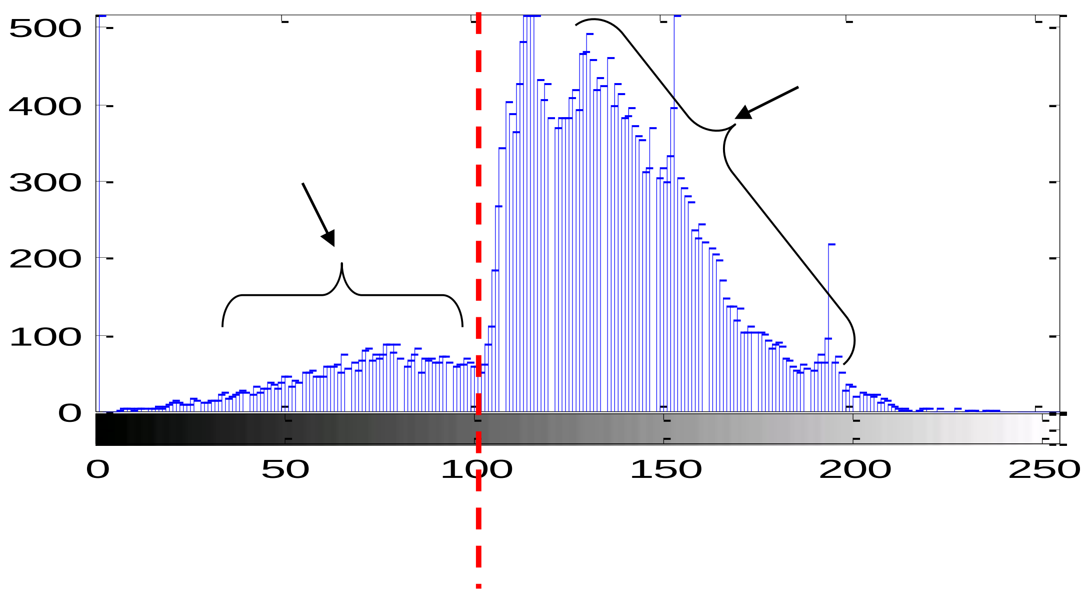
- Thresholding is based on partitioning the image into regions based on intensity values and/or properties of these values
- It is simple and computationally cheap
- In its very simple form, thresholding attempts to find an intensity value that separates the objects of interest from remaining of the image
- Usually this is performed by investigating the image histogram which may contain modes that represent the object(s) and the background

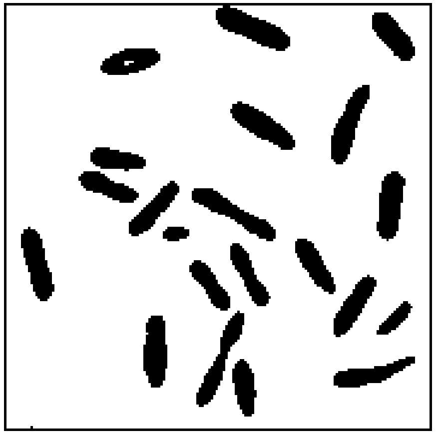
Thresholding



$$g(x,y) = \begin{cases} 1 & f(x,y) > T \\ 0 & f(x,y) < T \end{cases}$$

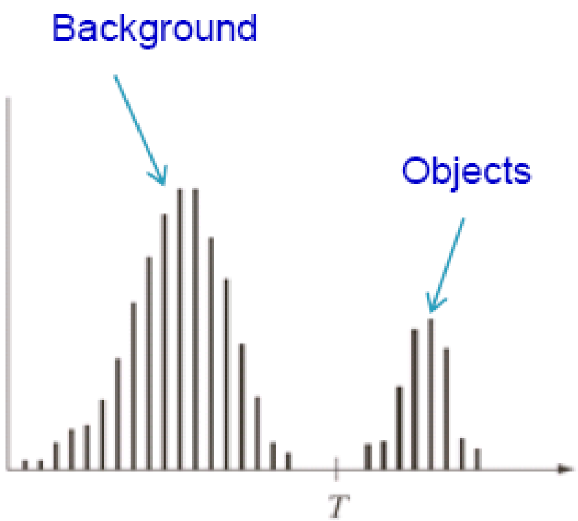
T = 102





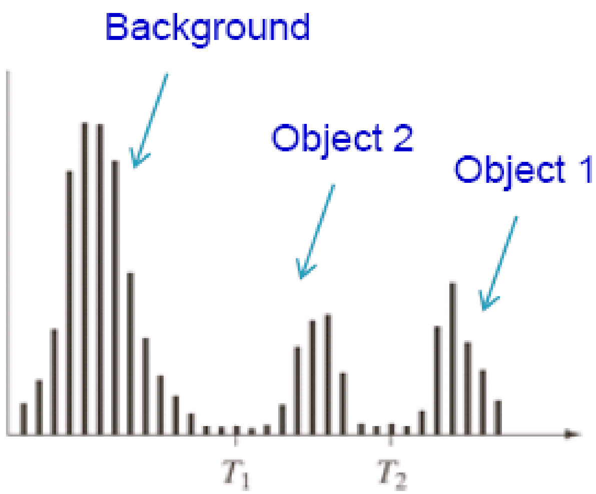
After thresholding

Thresholding



Histogram of an image with light objects and dark background . Segmenting the objects can be achieved by specifying T such that

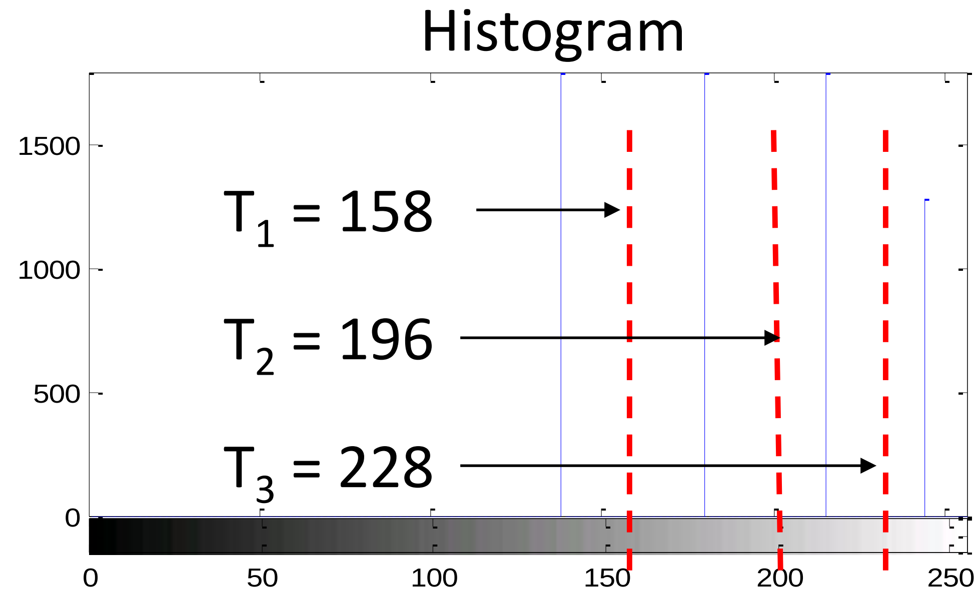
$$g(x,y)=\begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \le T \end{cases}$$



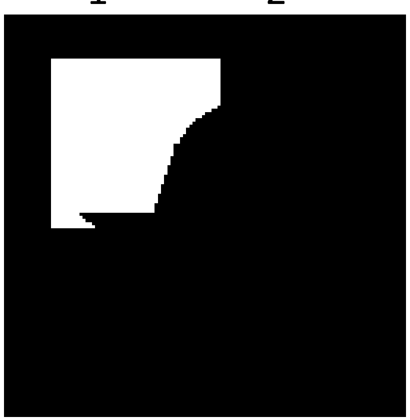
More challenging situation where we need to specify two thresholds in order to segment the object from background (Multiple Thresholding)

$$g(x,y)=\begin{cases} a, & f(x,y) > T_2 \\ b, & T_1 < f(x,y) \le T_2 \\ c, & f(x,y) \le T_1 \end{cases}$$

Thresholding

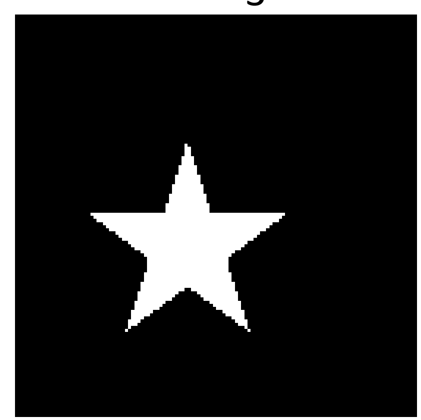


$T_1 < P < T_2$



$T_2 < P < T_3$

$P > T_3$

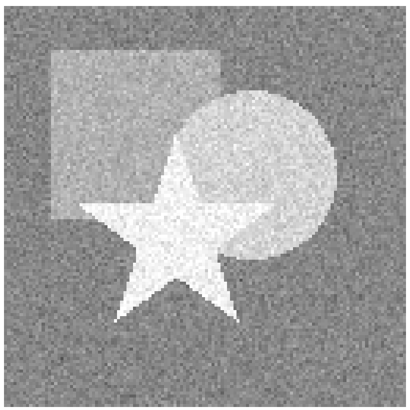


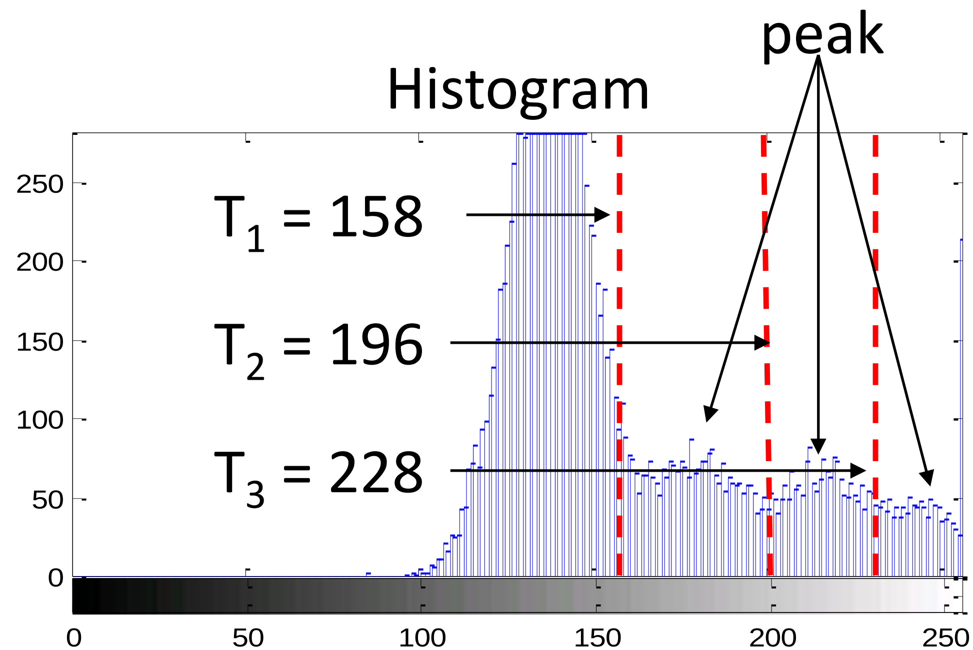
Thresholding

- **Thresholding Approaches**
  - Global – specify one or more fixed threshold values based on histogram shape
  - Local or regional thresholding – threshold values T at any point (x,y) is dependent on the properties of neighborhood around (x,y)
  - Dynamic or adaptive – threshold value depends on neighborhood properties and spatial coordinates as well
- **The success of intensity thresholding depends on**
  - The separation between the peaks in the histogram
  - The noise content in the image (the histogram modes broaden as noise increases)
  - The relative size of objects and background
  - The uniformity of illumination source
  - The uniformity of the reflectance properties of the image

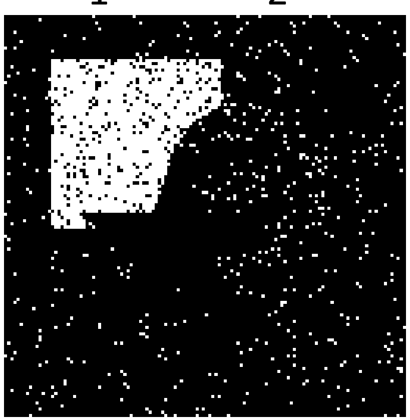
Thresholding - Effect of Noise

Image degraded by Gaussian noise (s =12)

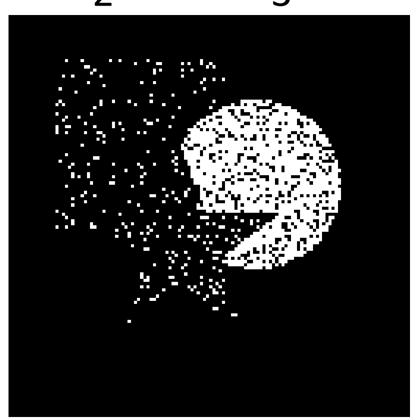




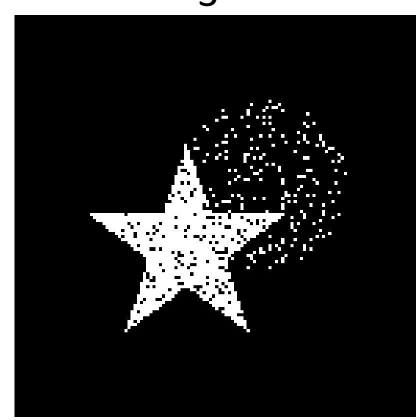
$T_1 < P < T_2$



$T_2 < P < T_3$



$P > T_3$



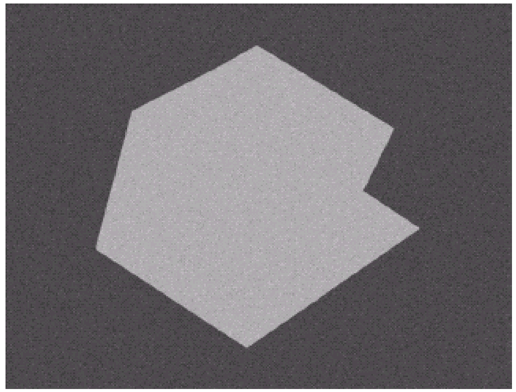
Thresholding - Effect of Illumination

An image can be expressed as

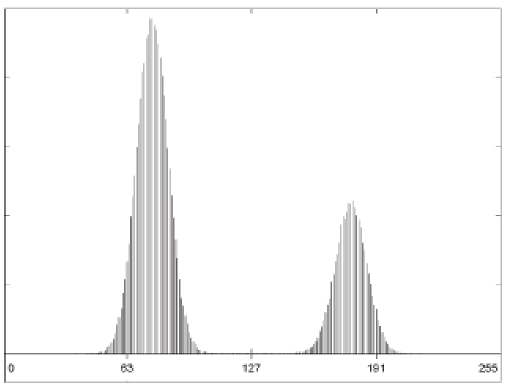
$$f(x,y) = i(x,y)r(x,y)$$

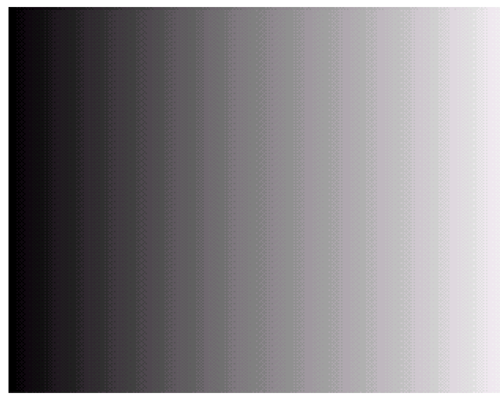
$i(x,y)$ = illumination component
$r(x,y)$ = reflectance component

Reflectance Function $r(x,y)$



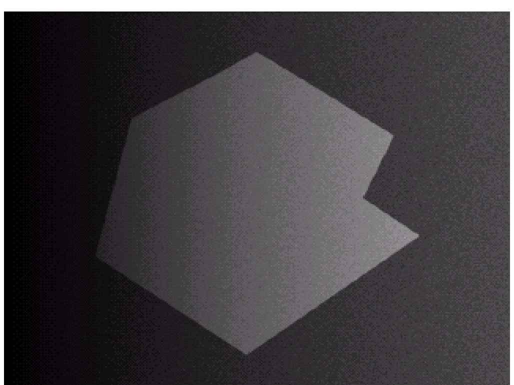
Histogram





Illumination Function $i(x,y)$

$f(x,y)$



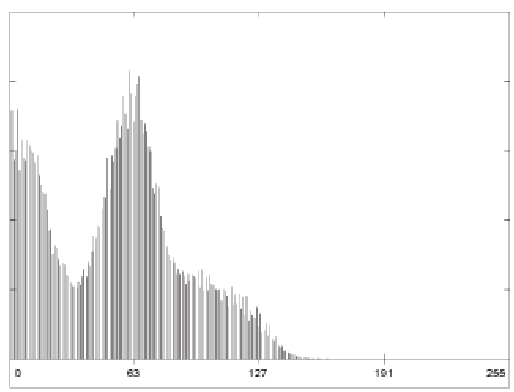
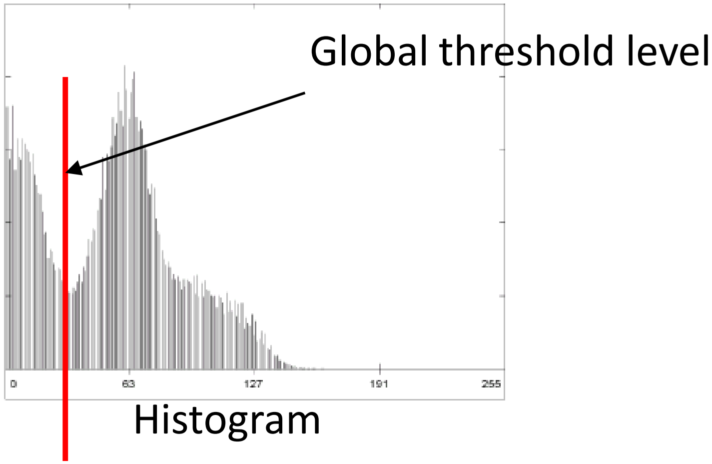


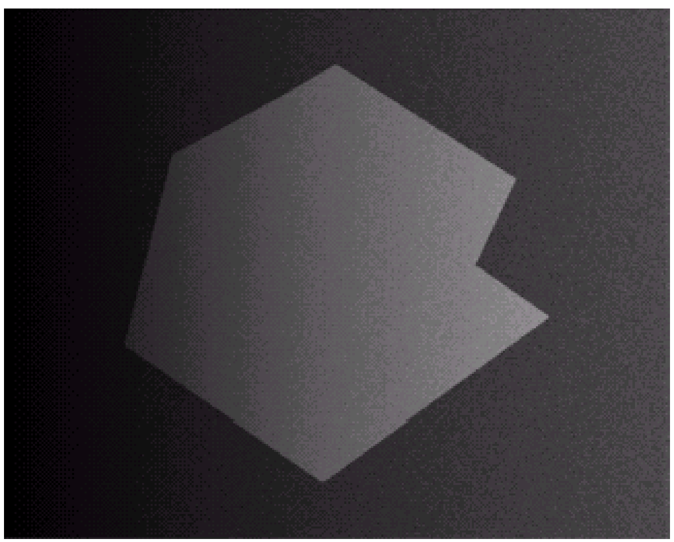
Image histogram

Thresholding - Effect of Illumination



Histogram

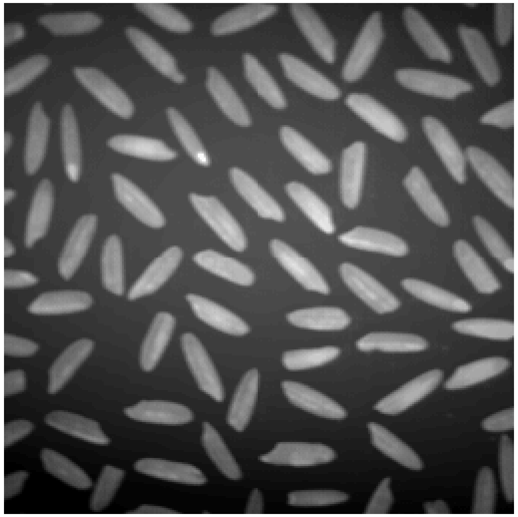
Global thresholding of nonuniform illumination image can cause huge errors!

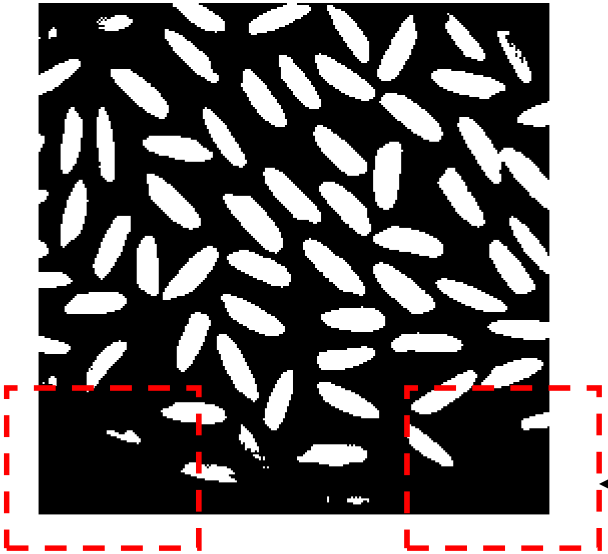


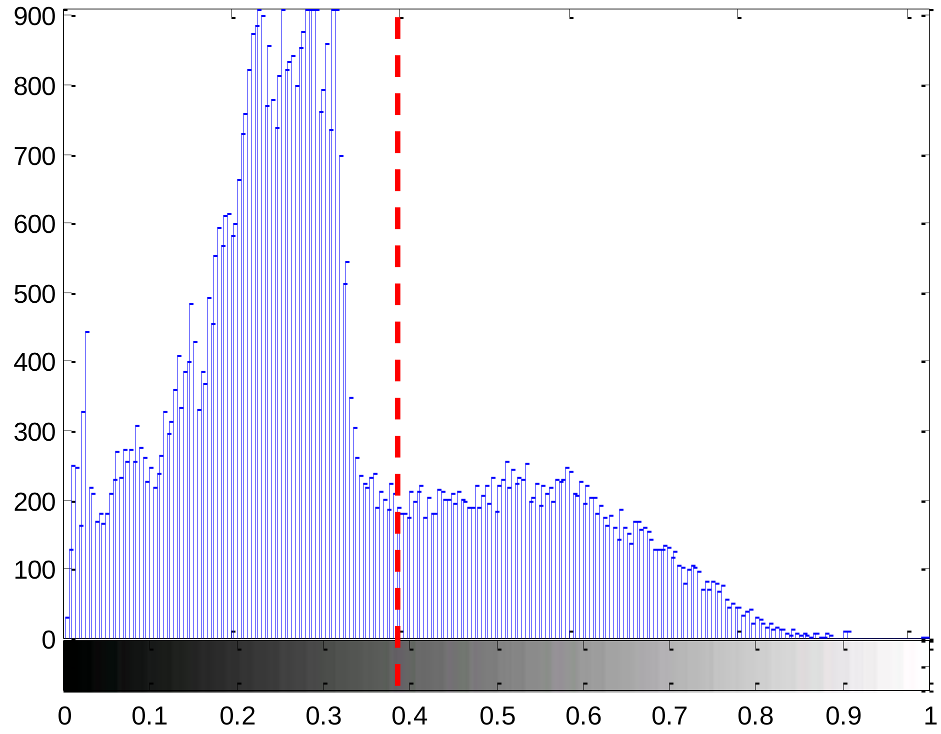
Nonuniform illumination image

Global thresholding result

Thresholding - Effect of Illumination







T=0.4

Error

Basic Global Thresholding

- When the intensity distribution of objects and background is sufficiently enough, it is possible to use a single global threshold value; a very common situation in practice
- To find such global threshold **automatically**

1. Select an initial estimate for the global threshold value, T
2. Segment the image using T to produce two groups of pixels; G1 containing pixels with intensity values > T and G2 with intensity values ≤ T
3. Compute the mean intensity value for each group, m_1 and m_2, and define the new threshold by

   $$T = (m_1 + m_2)/2$$

4. Repeat steps 2 and 3 until the change in T is smaller than a specified value ΔT

Basic Global Thresholding

- **Example**

Original

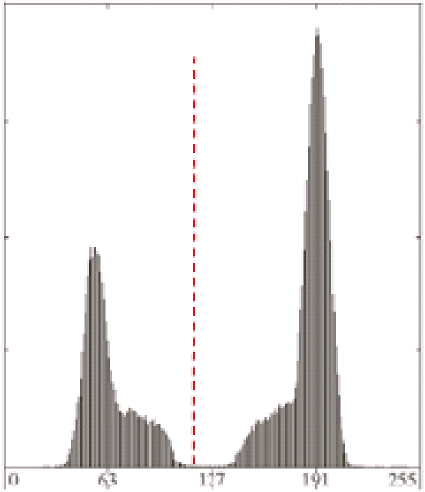


Image Histogram

Segmentation Result

Initial threshold = mean intensity of image

Optimal threshold = 125.4

Three iterations

Optimal Global Thresholding

- Views the thresholding problem as a statistical-decision problem with the objective minimizing the average error of miss-classifying pixels to two or more groups
- The method requires the knowledge of
  - The probability density function of the intensity levels of each class
  - The probability that each class occurs in a given applications
- Such requirements are usually hard to obtain in practice !
- Alternatively, we consider optimal global thresholding using Otsu's method
  - The method is optimal in the sense that maximizes the between-class variance
  - It is based on computations performed on the histogram of an image; an easily obtainable 1-D array

Otsu's Optimal Global Thresholding

- Let {0,1,2,..L-1} denote L distinct levels in MxN image . with normalized histogram whose components are $p_i = n_i/MN$
- Suppose we threshold the image with a threshold T(k) = k, 0<k<L-1, to produce two classes C_1 and C_2, where C_1 is the set of pixels with intensity levels in [0,k] and C_2 is the set of pixels with intensity levels in [k+1,L-1]
- Based on this threshold, the probability that a pixel belongs to C_1 (the probability of class C_1 occurring) is

$$P_1(k) = \sum_{i=0}^{k} p_i$$

and for C_2 is

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

Otsu's Optimal Global Thresholding

- The mean intensity value m_1 of pixels assigned to C_1 is

$$m_1(k) = \sum_{i=0}^{k} i \times P(i / C_1) = \sum_{i=0}^{k} i \times P(C_1 / i) P(i) / P(C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} i \times p_i$$

- Similarly, the mean intensity m_2 value of pixels assigned to C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} i \times P(i / C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i \times p_i$$

- The average intensity of the entire image is given by

$$m_G = \sum_{i=0}^{L-1} i \times p_i$$

- The cumulative mean up to level k is $m(k) = \sum_{i=0}^{k} i \times p_i$

- We can easily verify

$$P_1 m_1 + P_1 m_2 = m_G$$
$$P_1 + P_1 = 1$$

Otsu's Optimal Global Thresholding

- In order to evaluate the goodness of the threshold at level k we use the metric (separability measure)

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

where σ_G^2 is the variance of the entire image and σ_B^2 is the between-class variance which is defined as

$$\begin{aligned}\sigma_B^2(k) &= P_1(k)(m_1(k)-m_G)^2 + P_2(k)(m_2(k)-m_G)^2 \\ &= P_1(k)P_2(k)(m_1(k)-m_2(k))^2 \\ &= \frac{(m_G P_1(k)-m(k))^2}{P_1(k)(1-P_1(k))}\end{aligned}$$

- For optimal segmentation, we need to find the intensity level k that maximizes η as it implies larger seperability between classes

$$\eta(k^*) = \max_{0\le k\le L-1} \sigma_B^2(k)$$

Otsu's Optimal Global Thresholding

- **Notes**
  - As the difference between m_1 and m_2 increases, σ_B^2 increases to indicate better seperability
  - The definition of the metric η assumes that $\sigma_G^2 > 0$. The variance of an image could be zero if it contains one intensity level only
  - In order to find the optimal threshold k*, we have to compute σ_B^2 for all integer values of k and then pick the value that maximizes the between-class variance σ_B^2
  - Once the optimal threshold is found, we simply threshold the image into two classes using

$$g(x,y) = \begin{cases} 1, & f(x,y) > k^* \\ 0, & f(x,y) \le k^* \end{cases}$$

Optimal Global Thresholding

- **Summary of Otsu's Algorithm**

1) Compute the normalized histogram of the input image and denote its components by p_i , i = 1,2,…,L-1
2) Compute the cumulative sum $P_i(k)$, for k = 0,1,2,…,L-1 using

$$P_1(k)=\sum_{i=0}^{k} p_i$$

3) Compute the cumulative means, m(k), for k = 0,1,2,…,L-1 using

$$m(k)=\sum_{i=0}^{k} i\times p_i$$

4) Compute the global intensity mean m_G using

$$m_G=\sum_{i=0}^{L-1} i\times p_i$$

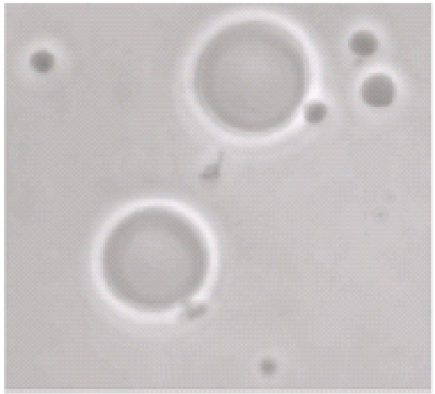
5)
6) Compute the between-class variance $\sigma_B^2(k)$ for k = 0,1,2,…,L-1 using

$$\sigma_B^2(k)=\frac{(m_G P_1(k)-m(k))^2}{P_1(k)(1-P_1(k))}$$

7) Pick Otsu's threshold k* that gives the maximum value for $\sigma_B^2(k)$
8) Obtain the separability measure at k* using $\eta(k^*)=\frac{\sigma_B^2(k^*)}{\sigma_G^2}$

Optimal Global Thresholding

- **Example:** segmentation of the molecules from background



Polymersome cells image

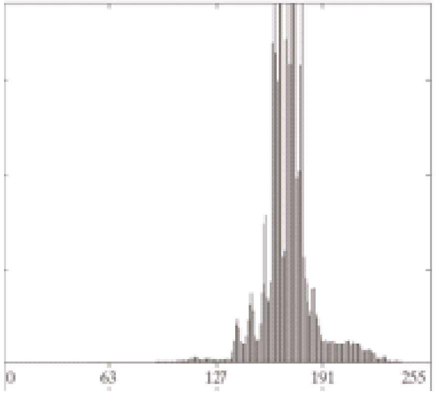


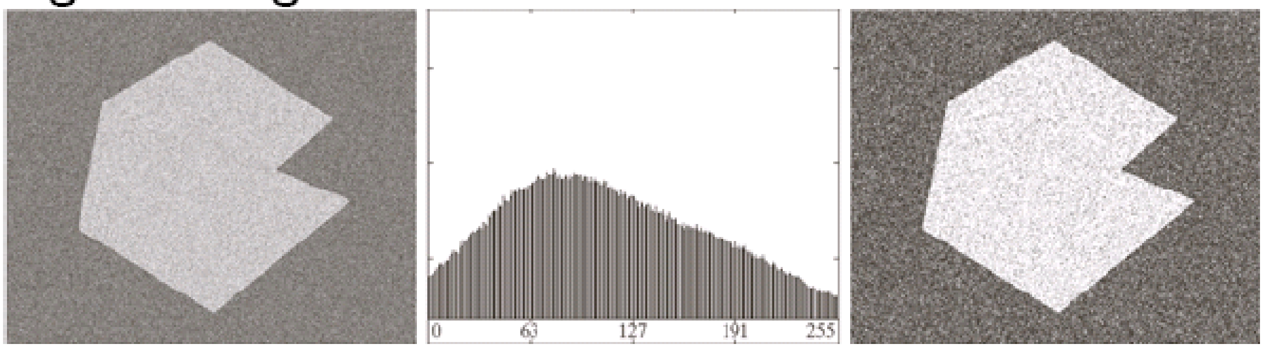
Image Histogram

Segmentation using basic global thresholding T* = 169

Segmentation using Otsu's global thresholding k* = 181

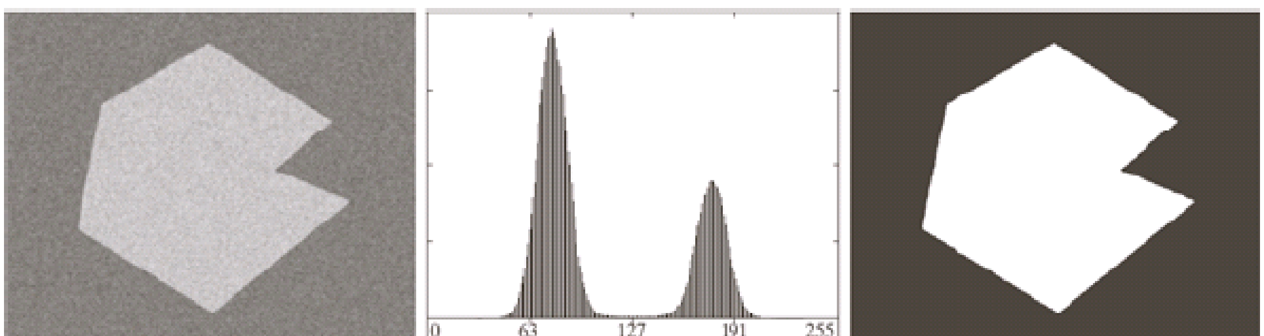
Enhancing Global Thresholding by Smoothing

- It was noted earlier that the presence of noise highly affect the result of thresholding
- One approach to improve thresholding result is to smooth the original image



Segmentation Fails

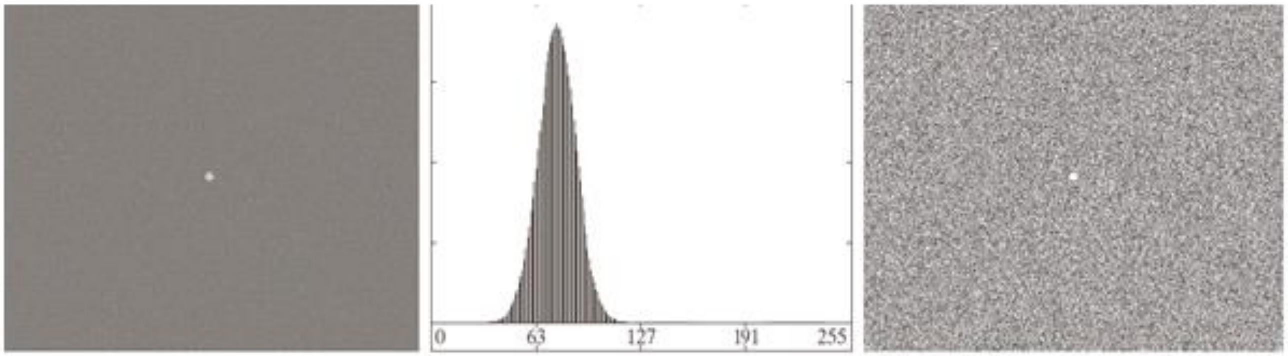
Noisy image, its histogram, and Otsu's segmentation result



Noisy image after smoothing by 5x5mask, its histogram, and Otsu's segmentation result

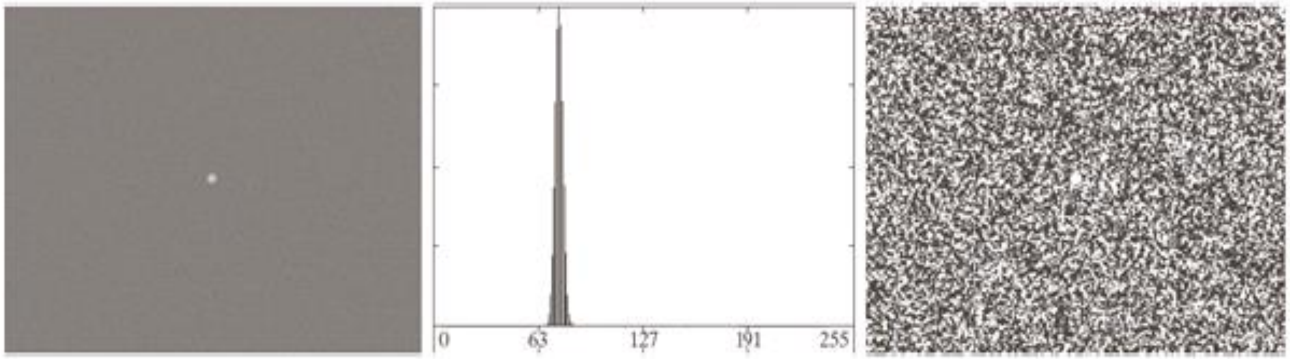
Effect of Relative Object Size on Thresholding

- As the size of the object decreases, its contribution to the image histogram decrease and it becomes harder to segment out



Segmentation Fails

Noisy image, its histogram, and Otsu's segmentation result

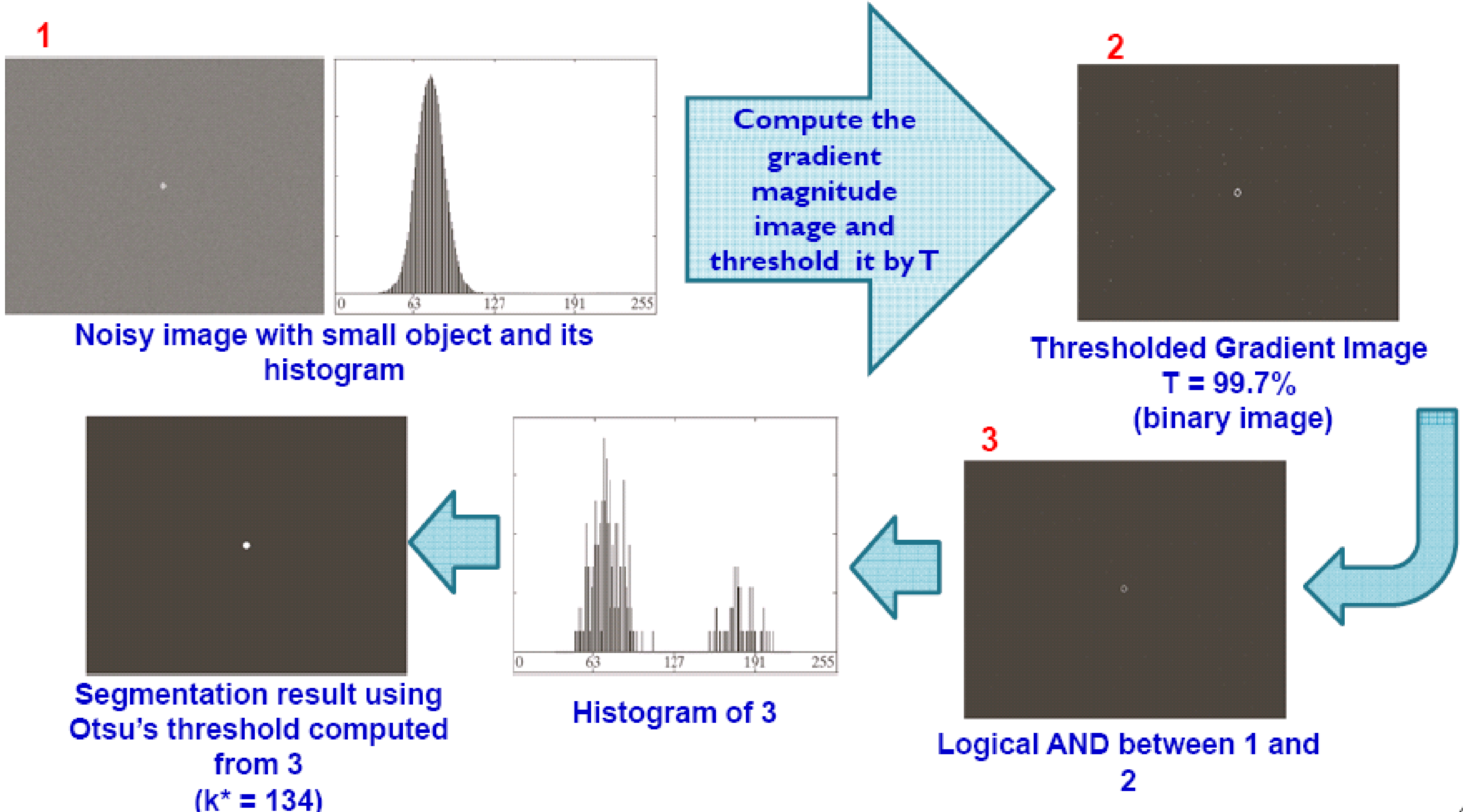


Segmentation Fails

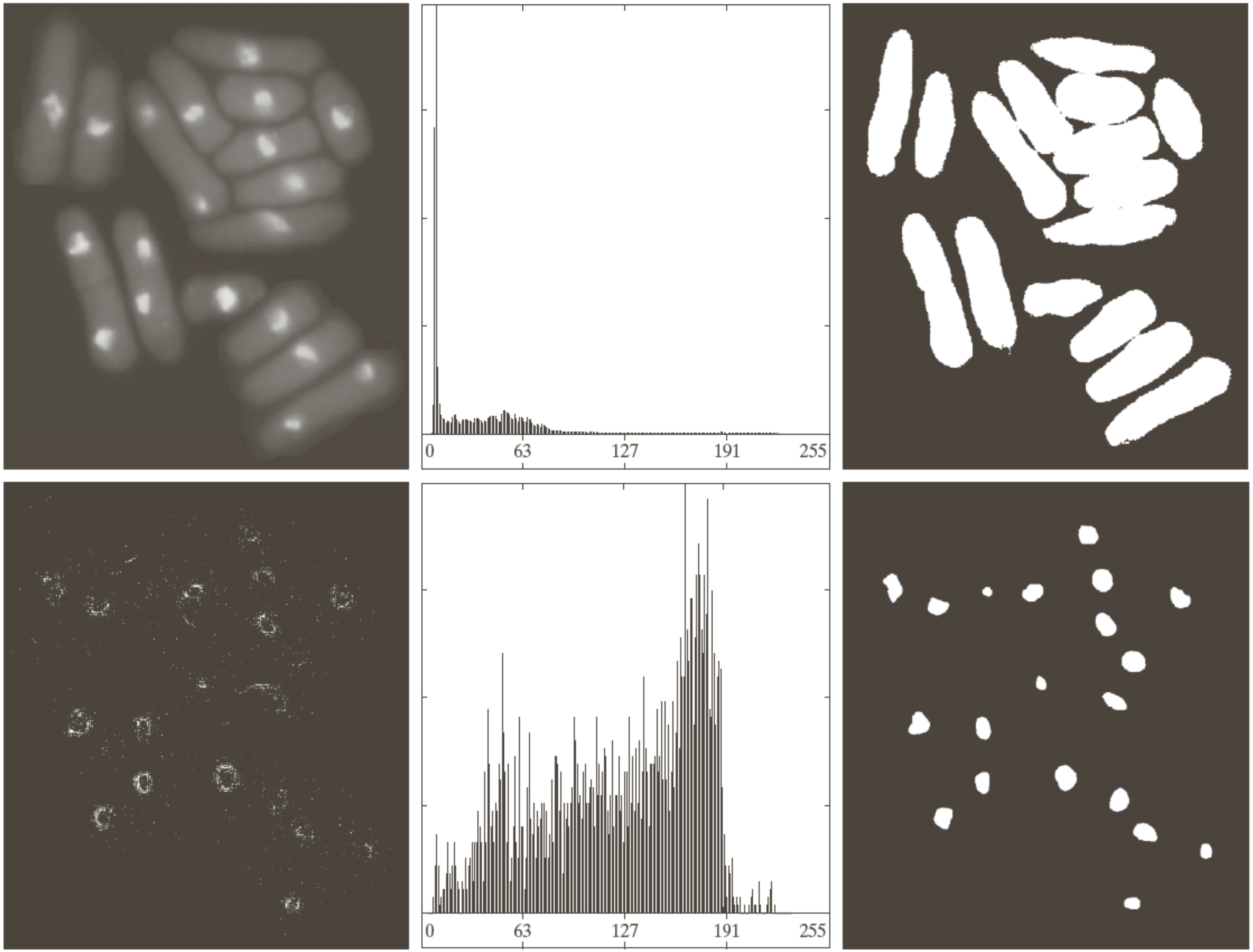
Noisy image after smoothing by 5x5mask, its histogram, and Otsu's segmentation result

Effect of Relative Object Size on Thresholding

- Global thresholding can be improved by incorporating edge information, especially when segmenting unimodal histograms

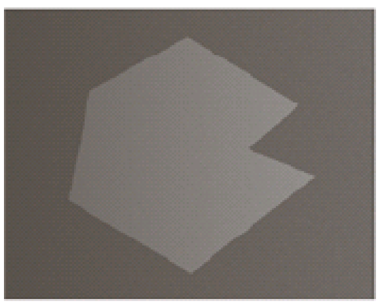


Effect of Relative Object Size on Thresholding

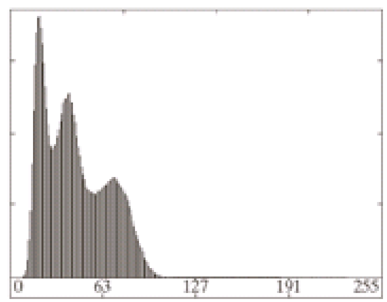


Variable Thresholding

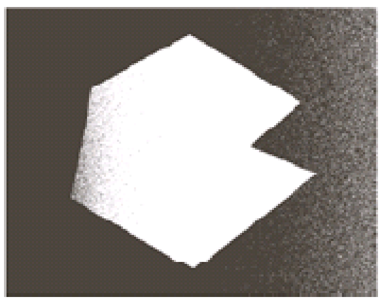
- **Variable Thresholding Based on Image Partitioning**
  - Subdivide the image into nonoverlapping rectangles to compensate in non-uniformities in illumination and/or reflectance. Threshold each subimage separately
  - The method works well when the objects and the background occupy regions of reasonably comparable size



Noisy image with nonuniform illumination



Histogram



Otsu's segmentation result

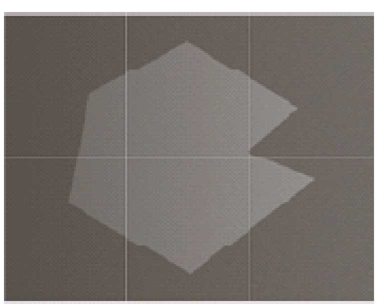
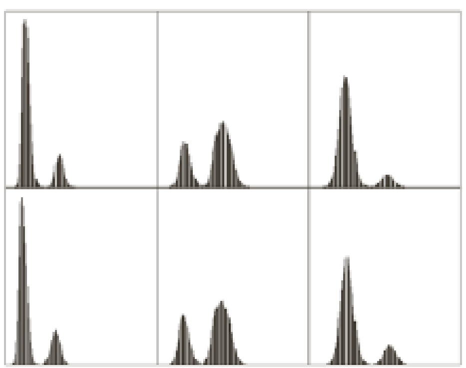


Image partitioning



Histogram of each subimage

Segmentation result after applying Otsu's method to each subimage

Variable Thresholding

- **Variable Thresholding Based on Local Image Properties**
  - Specify different threshold value at each pixel location (x,y) based on some statistical properties of the pixel's neighborhood
  - Some useful statistical measures for a neighborhood S_{xy} are the mean m_{xy} and standard deviation σ_{xy}
  - Some common forms for specifying the variable threshold are

$$T_{xy} = a\,\sigma_{xy} + b\,m_{xy}$$

OR

$$T_{xy} = a\,\sigma_{xy} + b\,m_G$$

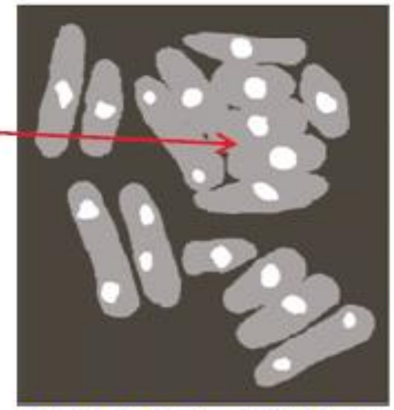
  - We can also use by using predicate functions based on the neighborhood parameters

$$g(x,y) = \begin{cases} 1\,, & f(x,y) > a\sigma_{xy} \text{ and } f(x,y) > m_{xy} \\ 0\,, & \text{otherwise} \end{cases}$$

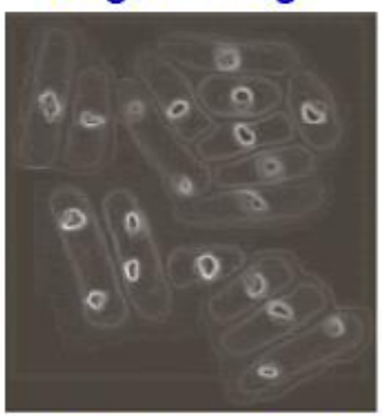
Variable Thresholding

- **Variable Thresholding Based on Local Image Properties - EXAMPLE**

Original Image



Dual Segmentation



Local Standard Deviation Image (3x3 neighborhood)

Variable Thresholding using g1(x,y)

$$g(x,y)=\begin{cases} 1\ , & f(x,y) > 30\sigma_{xy} \text{ and } f(x,y) > 1.5m_G \\ 0\ , & \text{otherwise} \end{cases}$$

Thresholding Summery

- Advantages:
  - Simple to implement
  - Fast (especially if repeating on similar images)
  - Good for some kinds of images (e.g., documents, controlled lighting)
- Disadvantages:
  - No guarantees of object coherency— may have holes, extraneous pixels, etc. (incomplete)
  - solution: post-processing with morphological operators

Other Thresholding/Binarization Techniques

- Niblack's method
- local-variance-based method by Sauvola
- Local adaptive method proposed by Bernsen
- Entropy-based method By Kapur
- learning framework for the optimization of the binarization methods by Cheriet

Segmentation by Region Growing

What is a region?

Definition:

- A group of connected pixels with similar properties.

Image interpretation:

- ´Region` is an important concept in interpreting an image because regions *may* correspond to objects in a scene.
- Consequently, for a correct interpretation of an image, we need to partition an image into regions that correspond to objects or parts of an object.
- Partitioning into regions done often by using gray values of the image pixels.

Region-based approach

Idea:

- Those pixels that correspond to an object are grouped together and marked.

Principles:

- Similarity:
  - Gray value differences
  - Gray value variance
- Spatial proximity:
  - Euclidean distance
  - Compactness of a region

Assumption:

- Points on same object map to nearby pixels on the image with similar gray values.
- The assumption does not hold true in all cases. Consequently, the solution is to group pixels using principles above and use domain-dependent knowledge.

Region growing segmentation

Principle/Idea:

- Region growing is the simplest region-based segmentation that groups pixels or sub-regions into larger regions based on pre-defined criteria.
- The pixel aggregation starts with a set of “seed” points in a way that the corresponding regions grow by appending to each seed points those neighboring pixels that have ***similar*** properties (such as gray level, texture, color, shape).

Comparison:

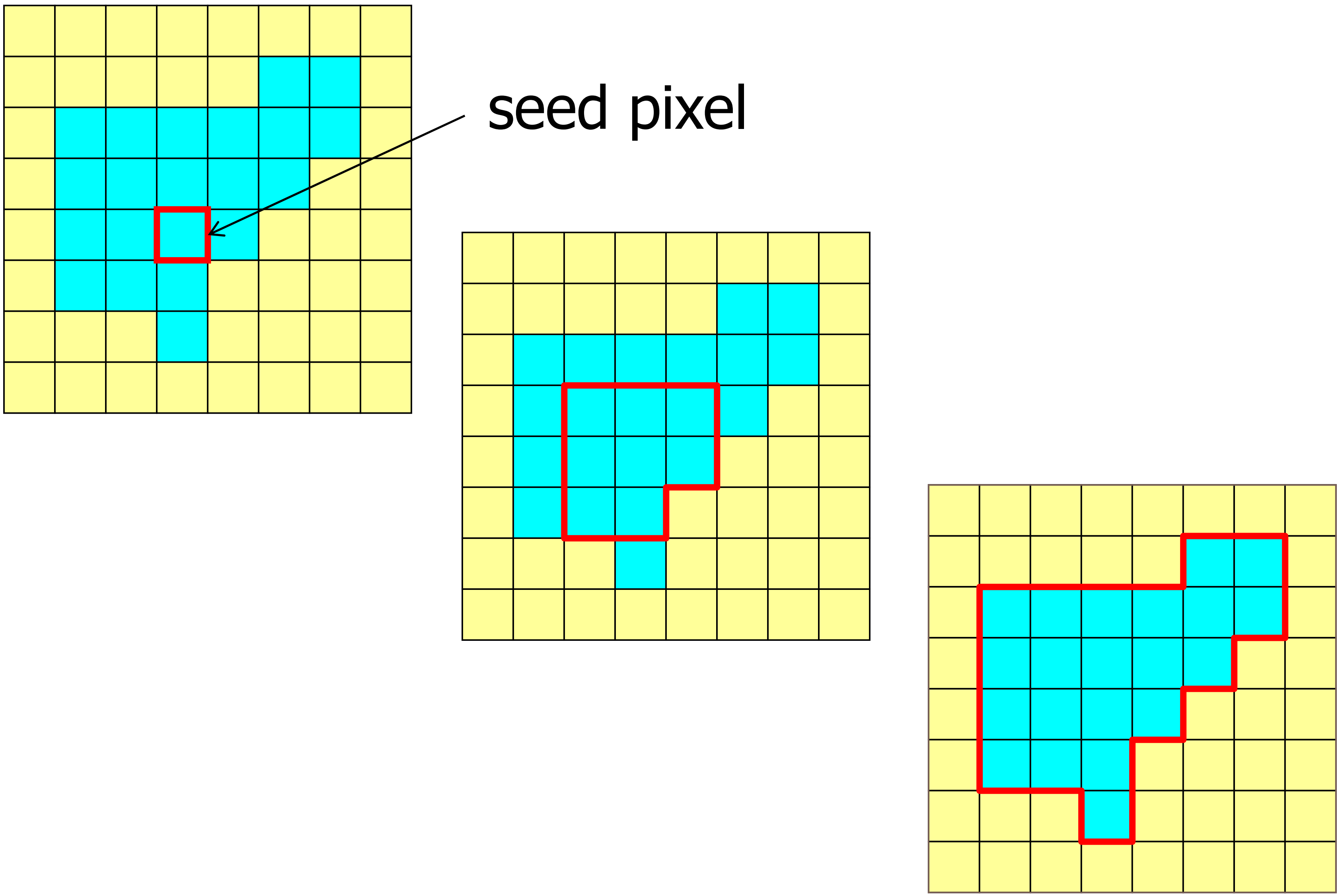
- Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

seed *growing* *final region*

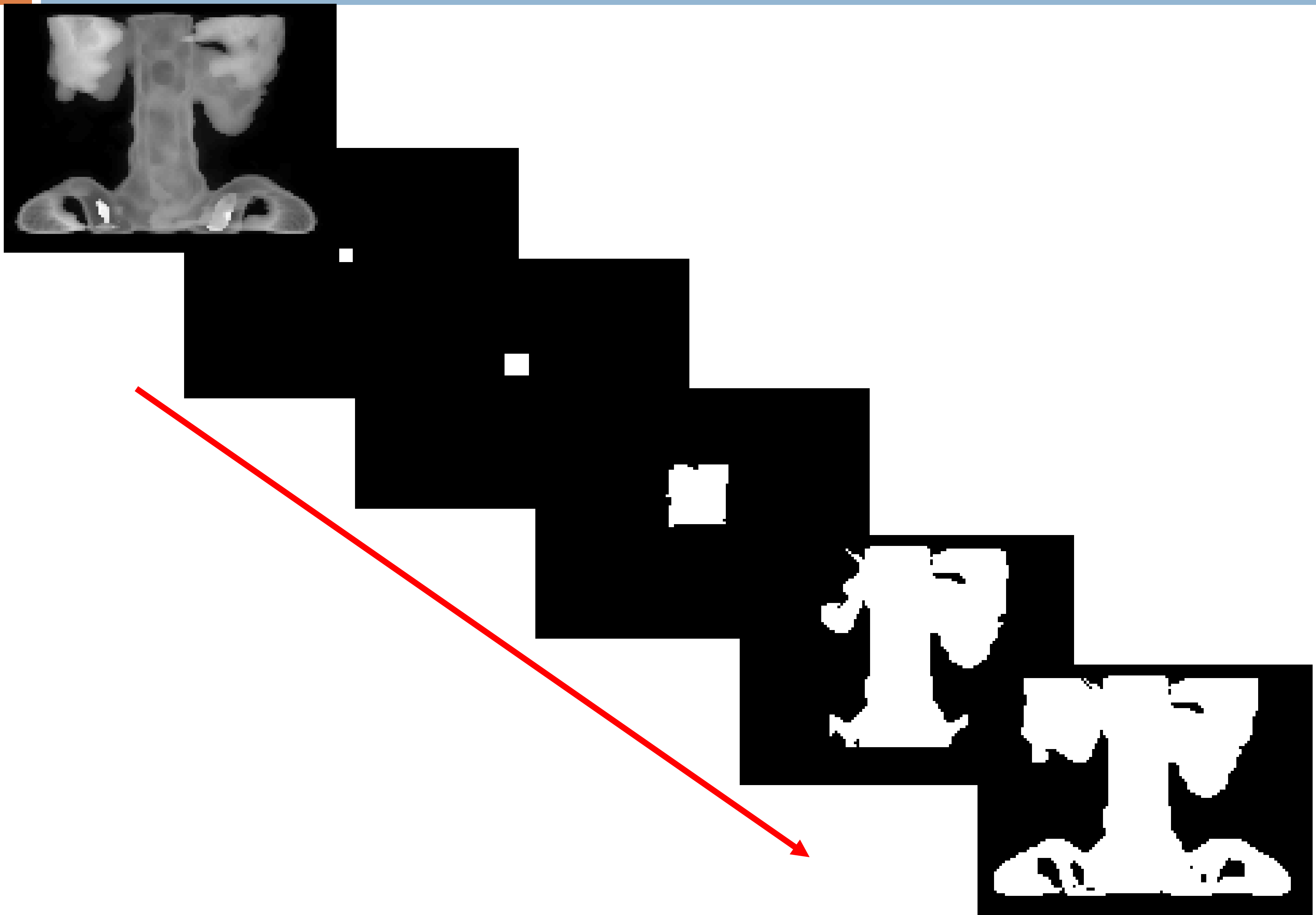
Segmentation by Region Growing

- **A General Region Growing Algorithm**
  1. Define the similarity criteria (difference intensity, variance, …)
  2. Define the stopping criteria (properties of the region, size, shape, …)
  3. Select a set of seed pixels; S
  4. Define an empty array O(x,y) and initialize its elements to 0's except at seed locations. Seed locations are assigned different intensity values
  5. For each seed in S, check its neighbors (4-,8-, or d-neighbors) against the similarity criteria
  6. If any of the neighboring pixels satisfy the criteria, then set the corresponding location in O to the same intensity level as its seed
  7. Check the if the stopping criteria is not met. If not, repeat steps 4 through 7 to the newly added pixels
  8. Repeat steps 4 through 7 until all seed pixels are processed

Segmentation by Region Growing



Segmentation by Region Growing

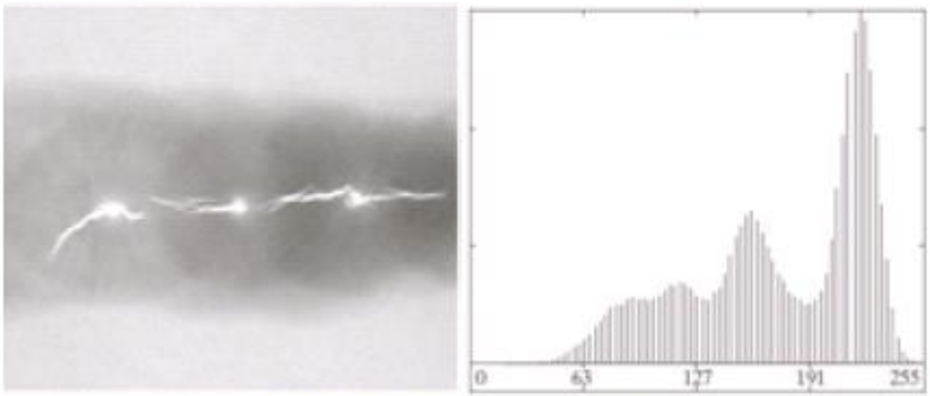


Segmentation by Region Growing

- How to select the seeds ?
  - Nature of problem
  - Random
  - Interactively
- How to select the similarity properties?
  - Nature of problem and image type (color ,monochrome ..)
  - Use statistical measures of local neighborhoods
  - May incorporate pixel location

Segmentation by Region Growing

- **Example: Segment the cracks in the weld**



Original Image and its histogram

Thresholding result to identify regions of high intensity

Seeds specified by random selection from those in the thresholded image

Result of region growing

$$g(x,y) = \begin{cases} 1, & |f(x,y)-N(x,y)| > 126 \\ 0, & \text{otherwise} \end{cases}$$

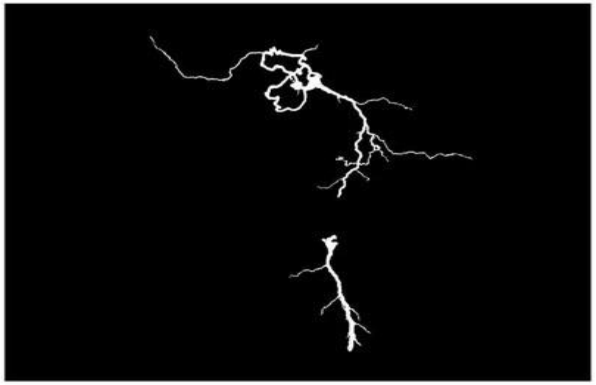
Seed-based region growing segmentation: *example*

Problem: To isolate the strongest lightning region of the image on the right hand side without splitting it apart.
Solution: To choose the points having the highest gray-scale value which is 255 as the seed points shown in the image immediately below.

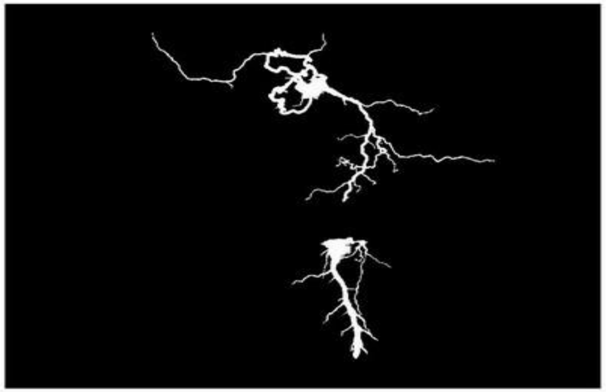
original image

threshold = 255 returns multiple seeds

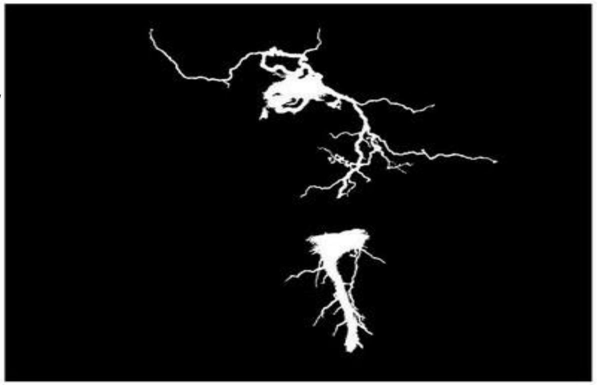
threshold: 225~255



threshold: 190~225



threshold: 155~255



Region growing segmentation: advantages & disadvantages

Advantages:

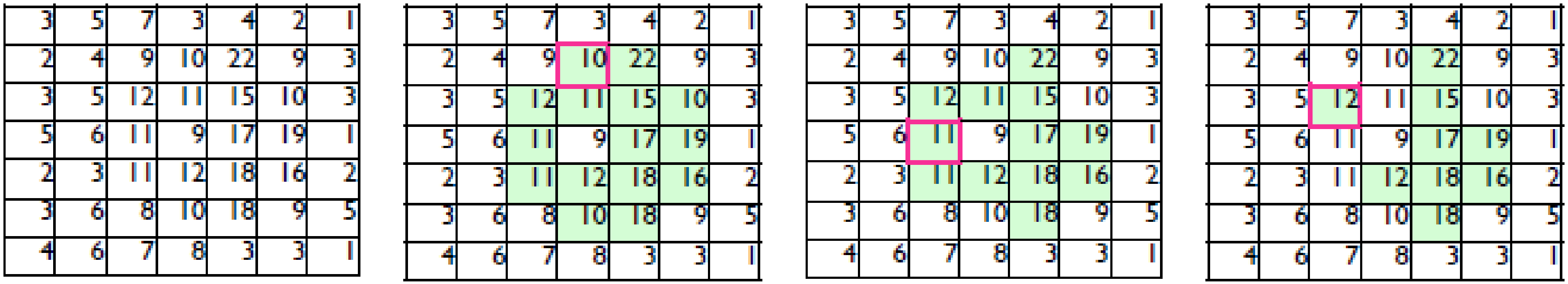
- It is a fast method.
- It is conceptually simple.

Disadvantages:

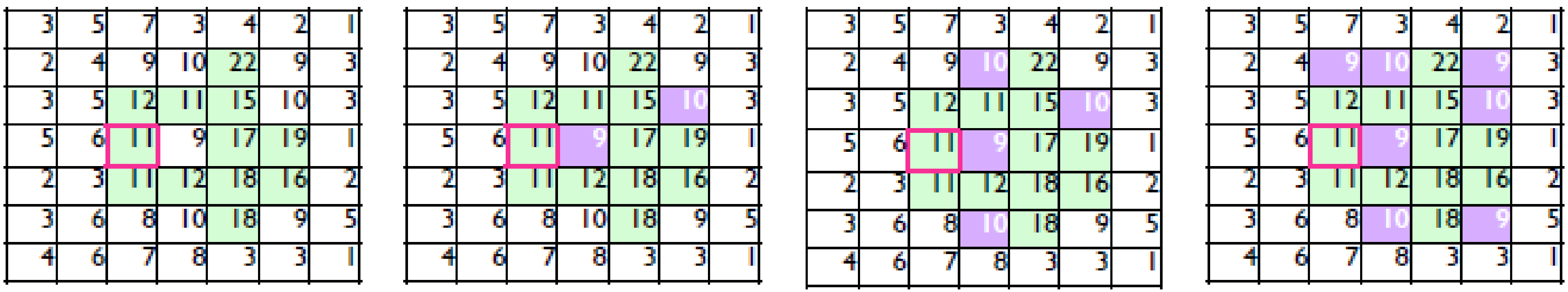
- Local method: no global view of the problem.
- Gradient problem: in practice, there is almost always a continuous path of points related to color close that connects two points of an image. Thus, unless we use a pre-defined variance (threshold), this will lead to the gradient problem:

- Algorithm very sensitive to noise.

Comparison of histogram and region growing

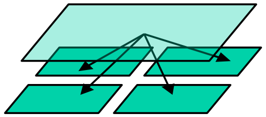


threshold $T \geq 10$ threshold $T \geq 11$ threshold $T \geq 12$



region growing with variance of 2 in respect to value 11 with reference to threshold $T \geq 11$

Region splitting and merging segmentation



Region splitting:

- Unlike region growing which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

Region merging:

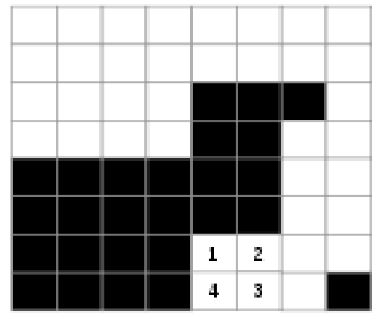
- Region merging is the opposite of region splitting, and works as a way of avoiding over-segmentation.
- Start with small regions (e.g. 2x2 or 4x4 regions) and merge the regions that have similar characteristics (such as gray level, variance).

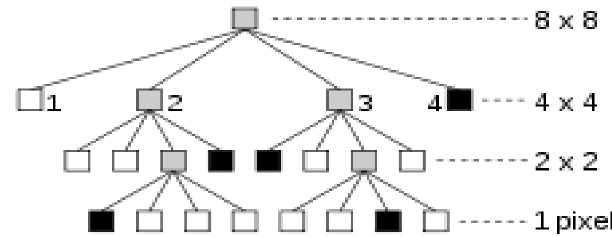
original image *splitting & merging* *thresholding seg.*

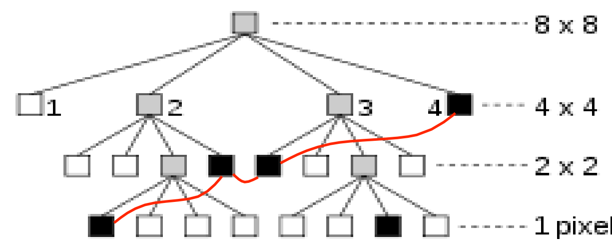
Splitting & merging: data structures

Two data structures:

- ***Quadtree*** for splitting.
  - Splitting is a top-down procedure that creates regions that may be adjacent and homogeneous, but not merged.
- ***RAG*** (region adjacency graph) for splitting and merging.
  - Splitting and merging work together iteratively, i.e., at each iteration of quadtree partitioning.
  - RAG has an embedded quadtree for splitting that represents 4 containment relations.
  - RAG also represents 4 adjacency relations (one per square side).





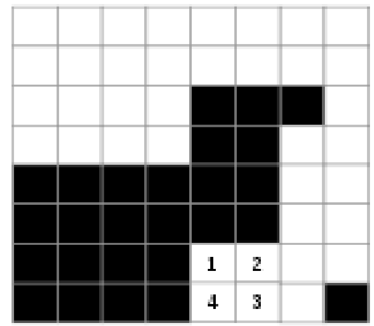


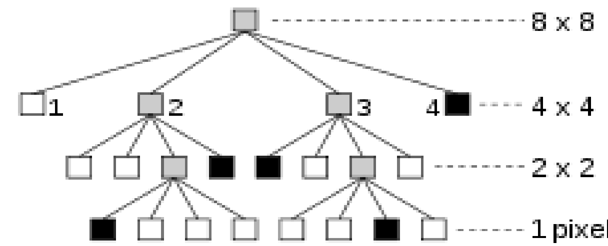
RAG with adjacency relations (in red) for big black region.

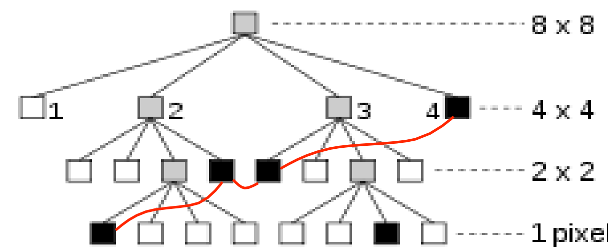
Splitting & merging segmentation algorithm

Algorithm:

- If a region R is inhomogeneous (P(R)=FALSE), then R is split into four sub-regions.
- If two adjacent regions R_i, R_j are homogeneous ($P(R_i \cup R_j)$=TRUE), they are then merged.
- The algorithm stops when no further splitting or merging is possible.

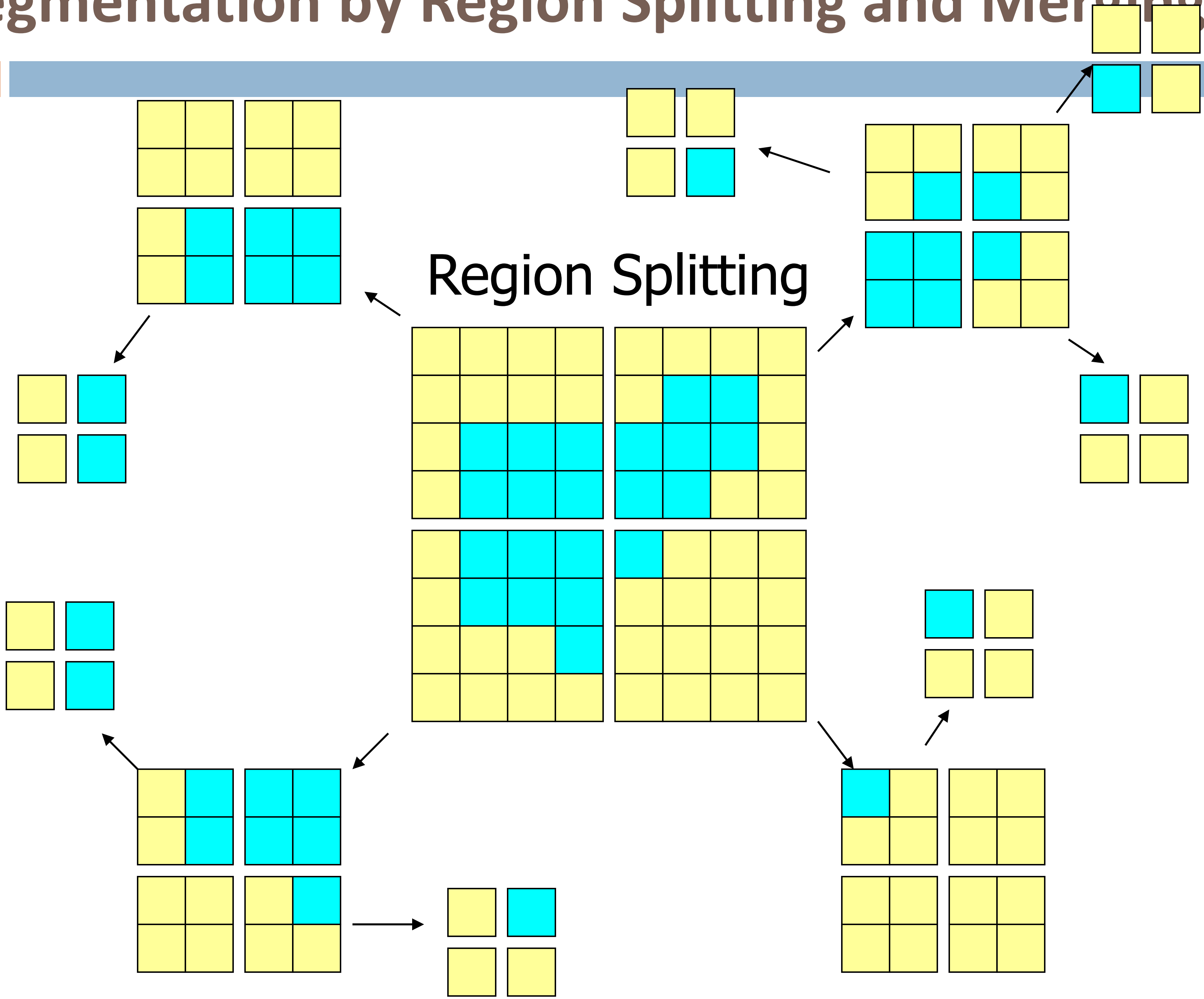




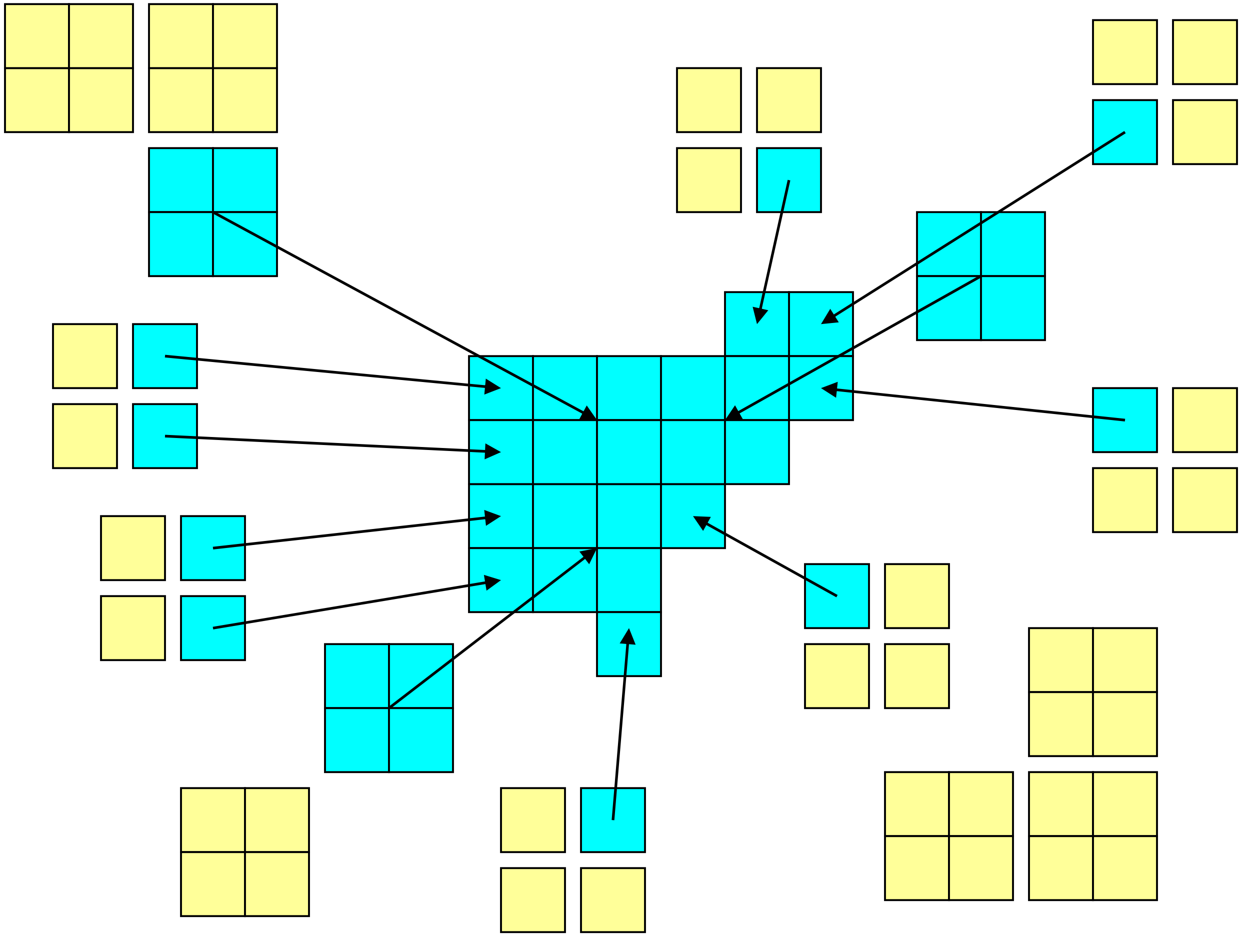


RAG with adjacency relations (in red) for big black region.

Segmentation by Region Splitting and Merging



Merging

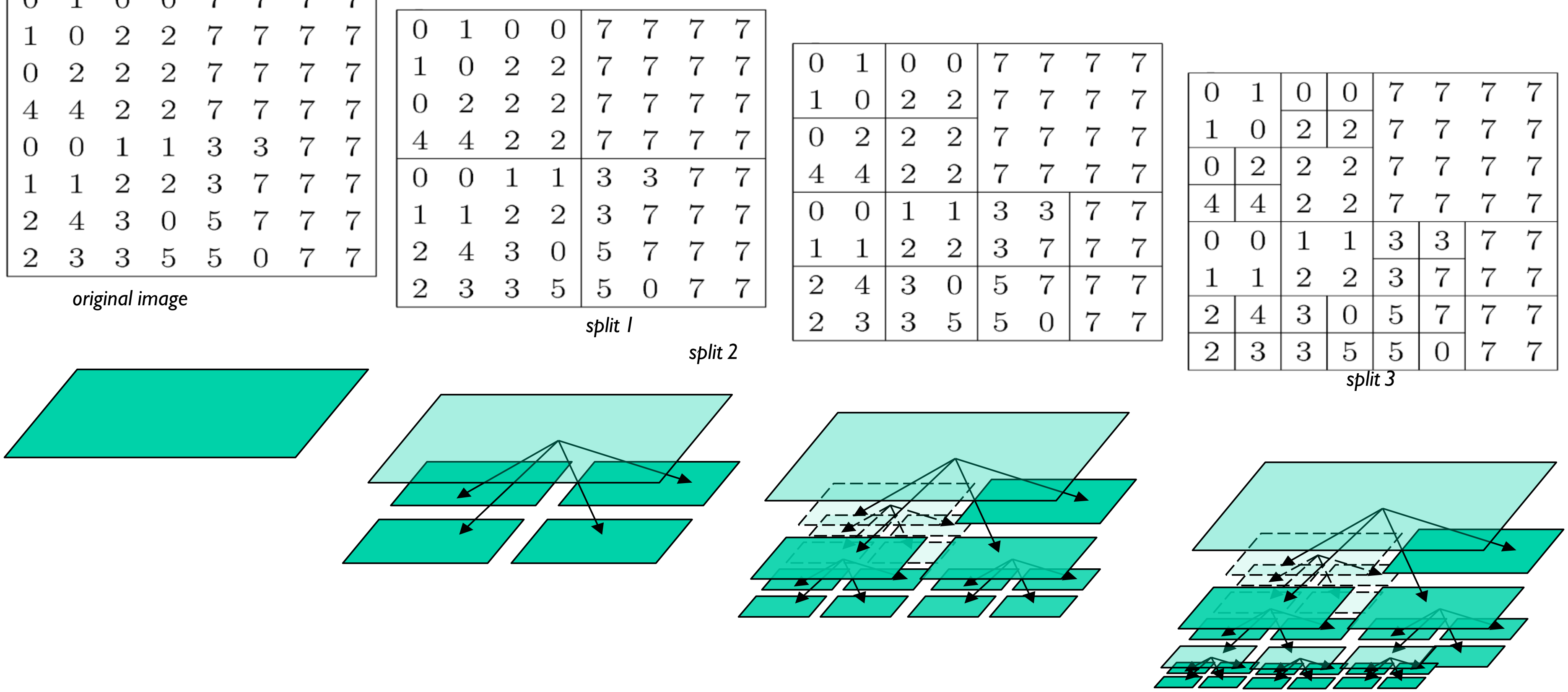


Region splitting: *example*

In this example, the **criterion of homogeneity** is the variance of 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

original image



split 1

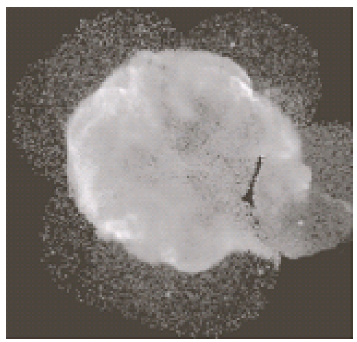
split 2

split 3

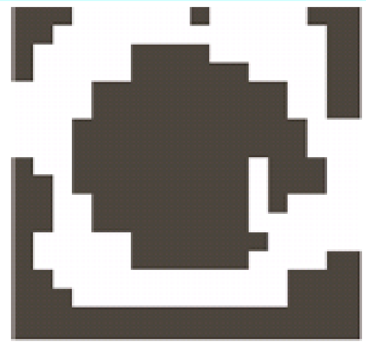
Segmentation by Region Splitting and Merging

- **Example**
  - Segment the less dense matter which is of noisy nature (high standard deviation when compared to background and the dense region) and moderate intensity
  - Use the predicate function

$$g(x,y)=\begin{cases} 1\,, & \sigma > 10 \text{ and } \; 0<m<126 \\ 0\,, & \text{otherwise} \end{cases}$$



Original

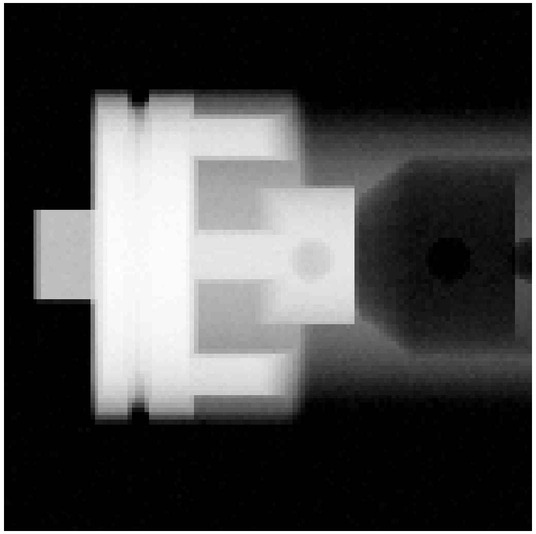


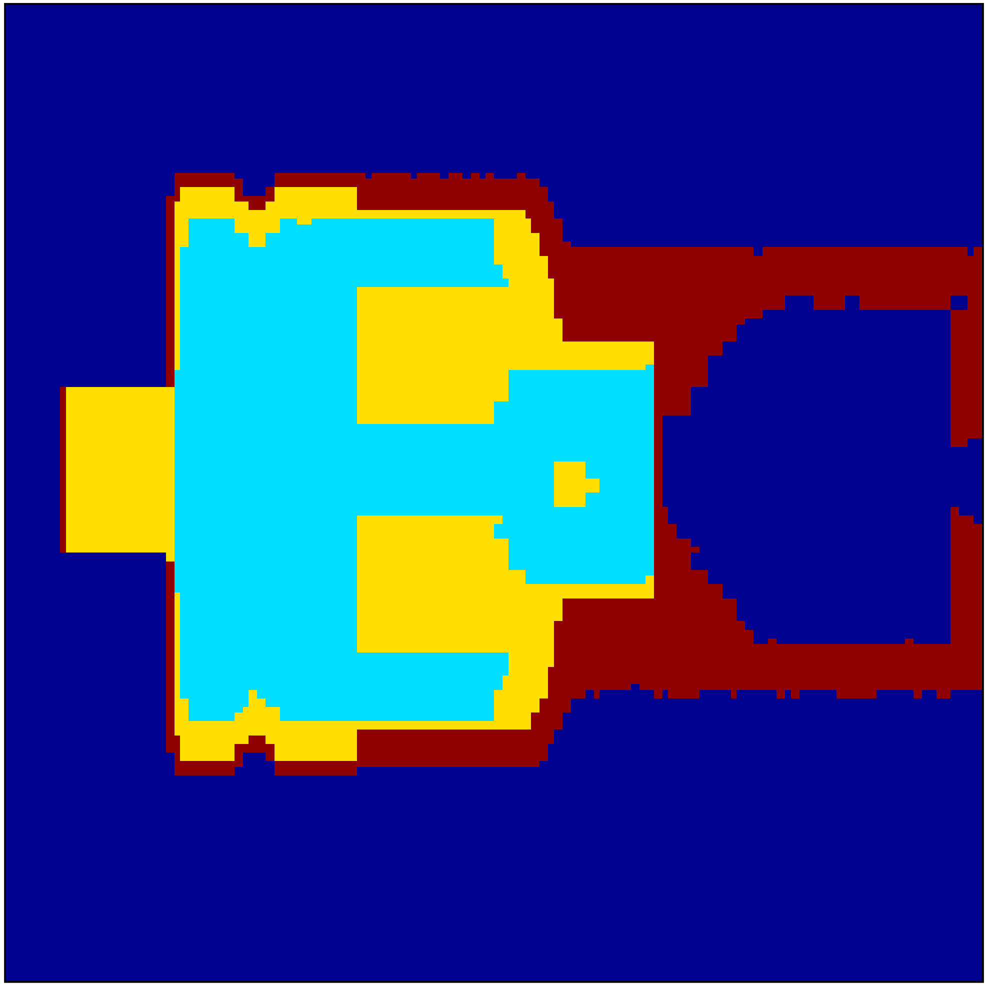
Minimum quadrant size 32x32

Minimum quadrant size 16x16

Minimum quadrant size 8x8

Segmentation by Region Splitting and Merging





Clustering Based Segmentation Methods

- What is Clustering?
  - Organizing data into classes such that:
    - High intra-class similarity
    - Low inter-class similarity

- What is similarity ?
  - **Cluster by features**
    - **Color**
    - **Intensity**
    - **Location**
    - **Texture**
    - **....**

K-means

- Given a K, find a partition of K clusters to optimise the chosen partitioning criterion (cost function)
  - global optimum: exhaustively search all partitions
- The K-means algorithm: a heuristic method
  - K-means algorithm (MacQueen'67): each cluster is represented by the centre of the cluster and the algorithm converges to stable centriods of clusters.
  - K-means algorithm is the simplest partitioning method for clustering analysis and widely used in data mining applications.

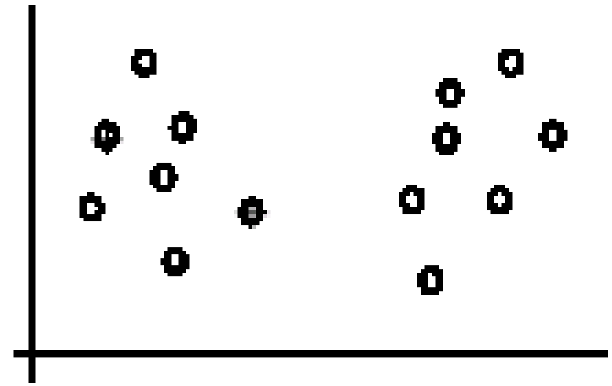
K-Means Clustering Algorithm

- Given the cluster number *K*, the *K-means* algorithm is carried out in three steps after initialization:

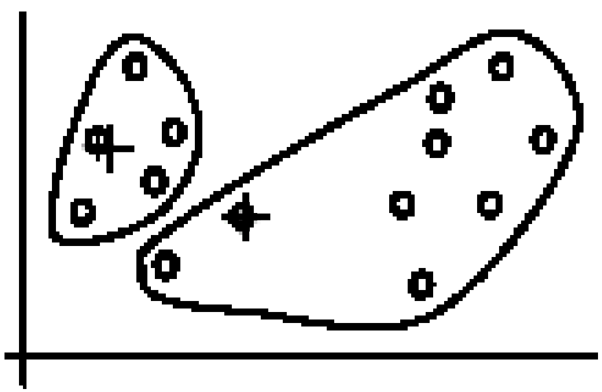
  Initialisation: set seed points (randomly)

  1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
  2) Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
  3) Go back to Step 1), stop when no more new assignment (i.e., membership in each cluster no longer changes)

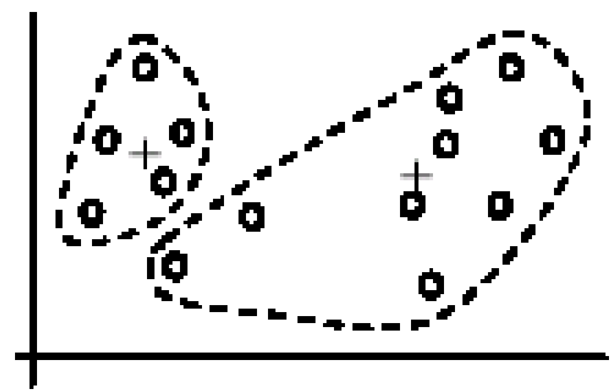
An example



(A). Random selection of *k* centers

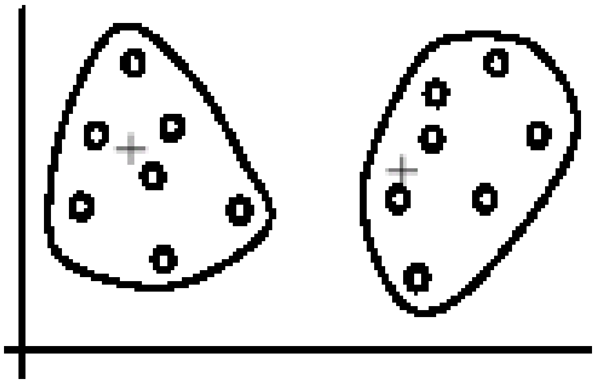


Iteration 1: (B). Cluster assignment

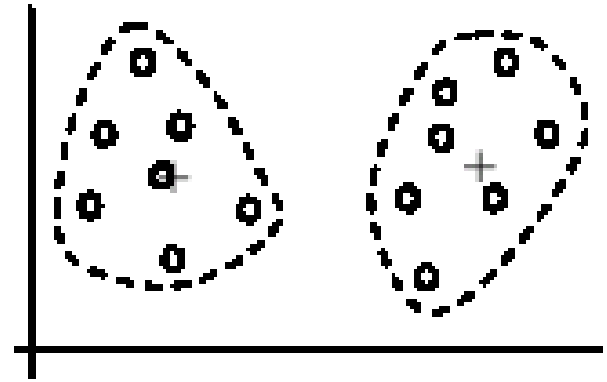


(C). Re-compute centroids

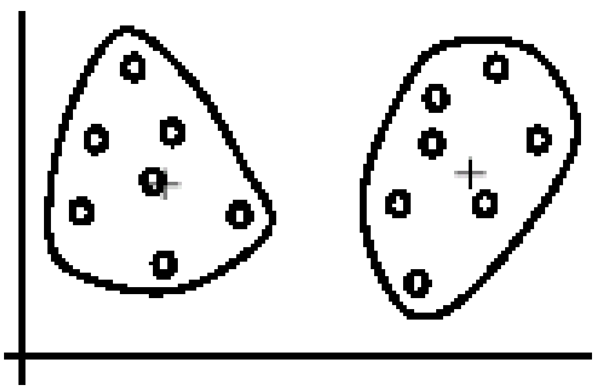
An example (cont …)



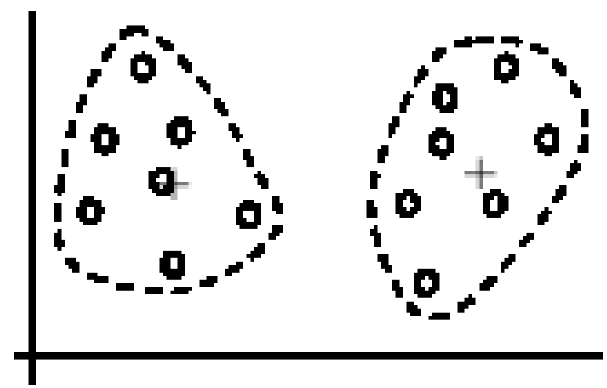
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

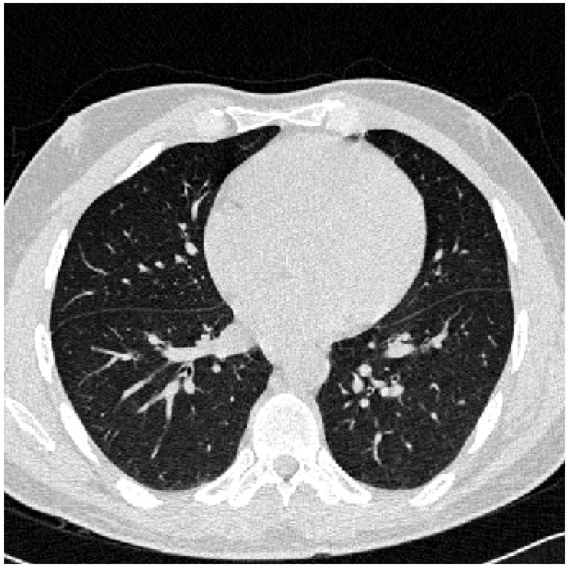
Stopping/convergence criterion

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

- C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

K-means Image Segmentation - Example



An image (*I*) Three-cluster image (*J*) on gray values of *I*

Note that *K*-means result is "noisy"

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- **k-modes**
- **K-Medoids**
- **Fuzzy C-Means Clustering**

Mean-Shift Clustering/Segmentation

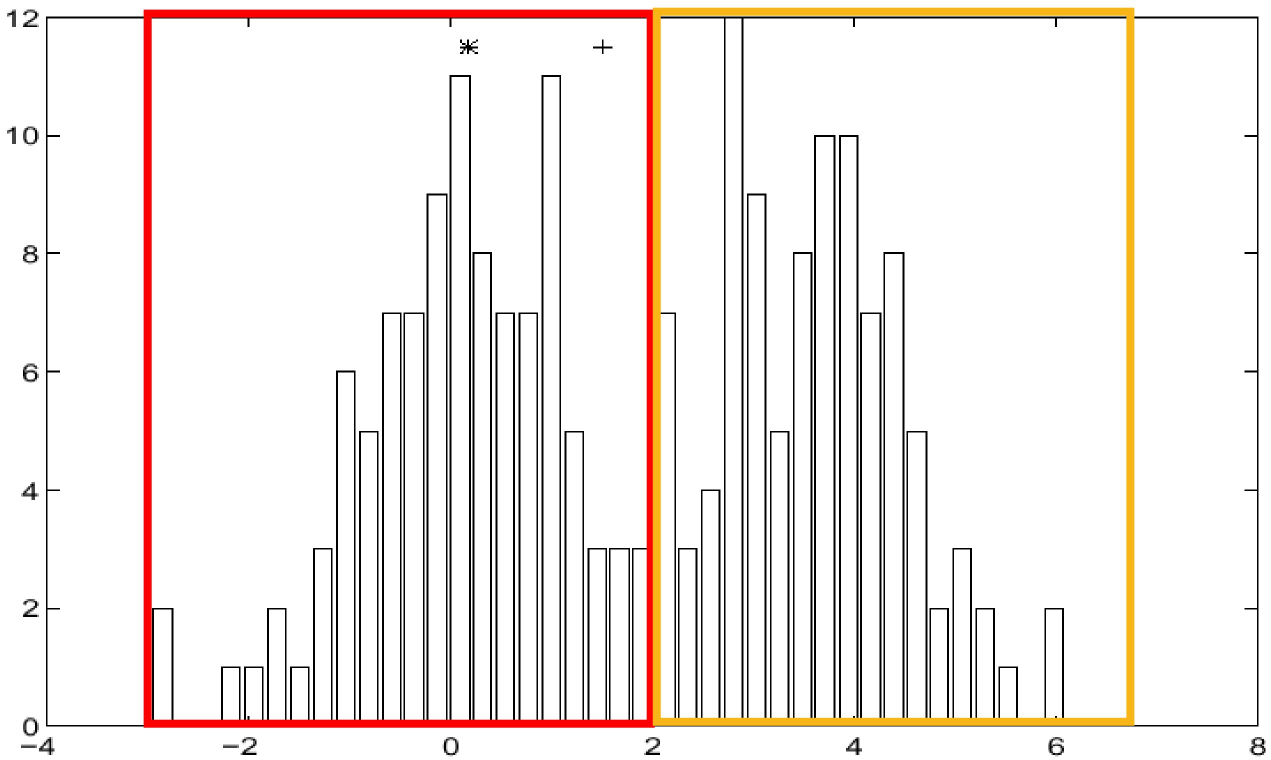
- Simple, like K-means
- But you don't have to select K
- Statistical method
- Guaranteed to converge to a fixed number of clusters.

Mean-Shift Segmentation

- **Mean-shift** is a variant of an iterative steepest-ascent method to seek stationary points (i.e., peaks) in a density function, which is applicable in many areas of multi-dimensional data analysis.
- Attempts to find all possible cluster centers in feature space (unlike k-means, where there is a requirement to know the number of different clusters).
- The mean shift algorithm seeks *modes* or local maxima of density in the feature space

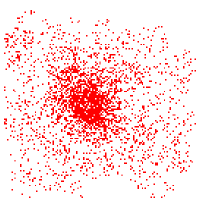
Mean-Shift Motivation

- Finding Modes in a Histogram
  - How Many Modes Are There?
    - Easy to see, hard to compute



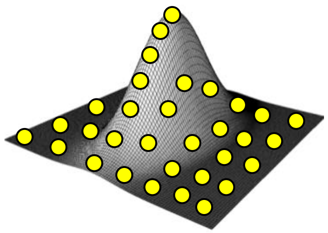
What is Mean Shift ?

A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in $\mathbf{R}^N$



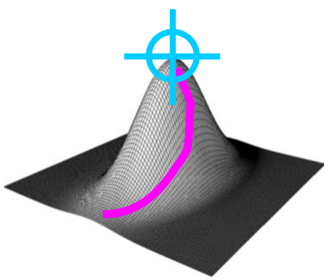
Data

Non-parametric
Density Estimation



Discrete PDF Representation

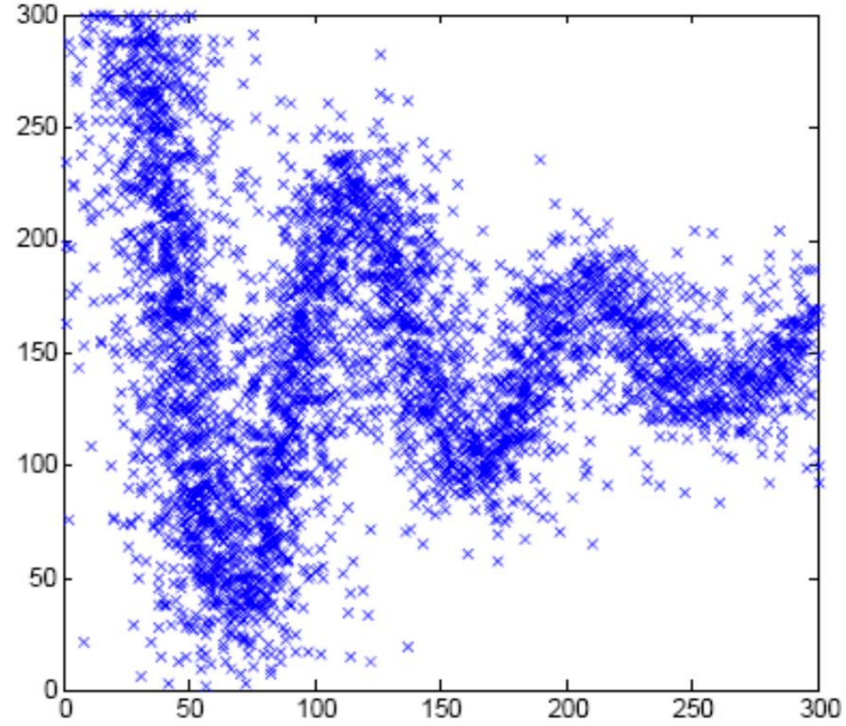
Non-parametric
Density **GRADIENT** Estimation
(Mean Shift)



PDF Analysis

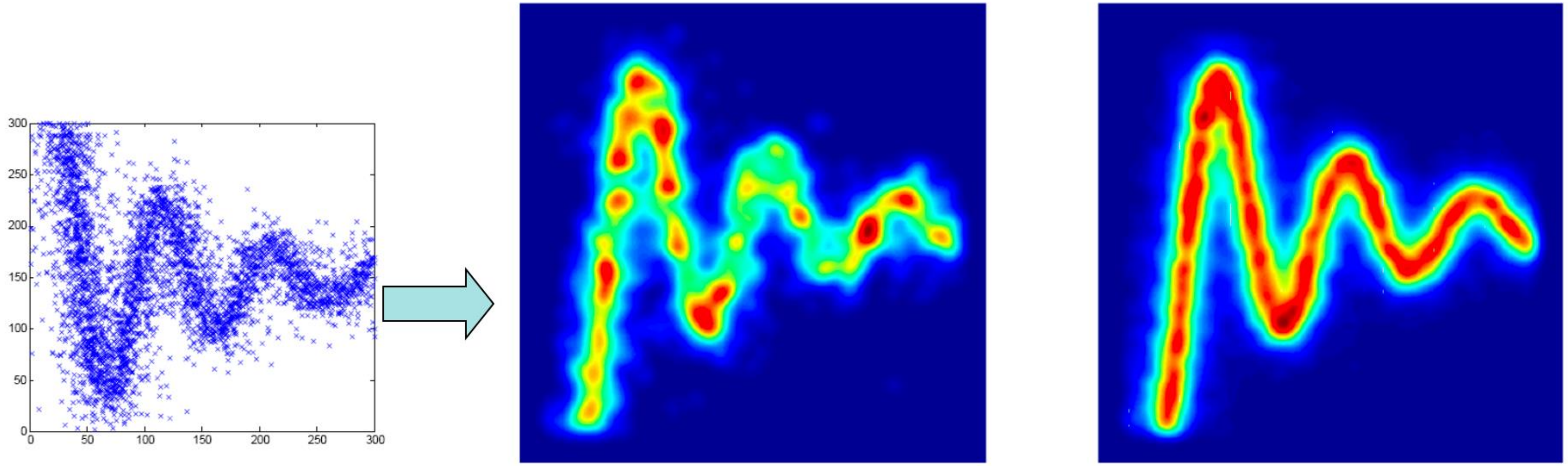
Density Estimation

- What is the distribution that generated these points?
- **Parametric model:**
  - Can express distr. With a few parameters (mean And variance)
  - Limited in flexibility!



Non-parametric density estimation

- Focus on kernel density estimates, using the data to define the distribution
- Build distribution by putting a little mass of probability around each data-point



(a) 2000 Samples (b) 20000 Samples

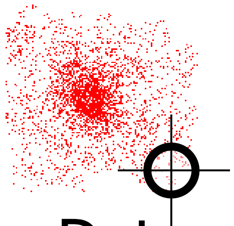
Mean-Shift Basics

- Mean-Shift is a procedure for locating maxima of a density function given discrete data samples from that function.
- This is an iterative method, and we start with an initial estimation of *x.*
- Let a kernel function *K(xi-x)* be given, determining the weight of nearby points for re-estimation of the mean.
- The weighted mean of the density in the window determined by K

Kernel Density Estimation

$$P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x}-\mathbf{x}_i)$$

A function of some finite number of data points $x_1 \ldots x_n$

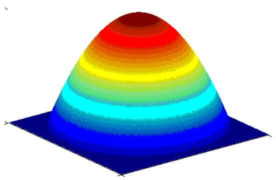


Data

Examples:

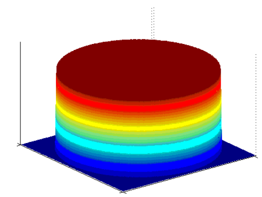
- Epanechnikov Kernel

$$K_E(\mathbf{x}) = \begin{cases} c\left(1-\|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



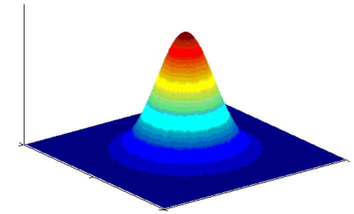
- Uniform Kernel

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



- Normal Kernel

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$



Kernel Density Estimation

Gradient

$$\nabla P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\mathbf{x}-\mathbf{x}_i)$$

Give up estimating the PDF !
Estimate **<u>ONLY</u>** the gradient

Using the Kernel form:

$$K(\mathbf{x}-\mathbf{x}_i) = ck\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

Size of window

We get :

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \square \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

$$g(\mathbf{x}) = -k'(\mathbf{x})$$

Computing The Mean Shift

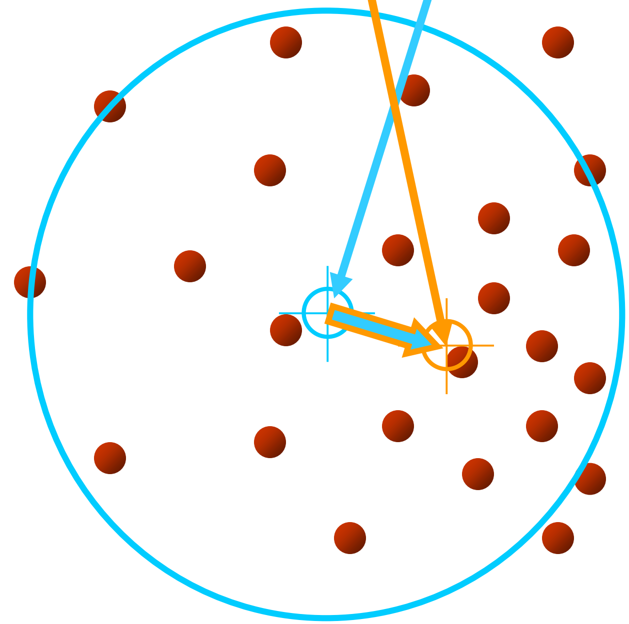
$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \cdot \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

Yet another Kernel density estimation !

Simple Mean Shift procedure:

- Compute mean shift vector

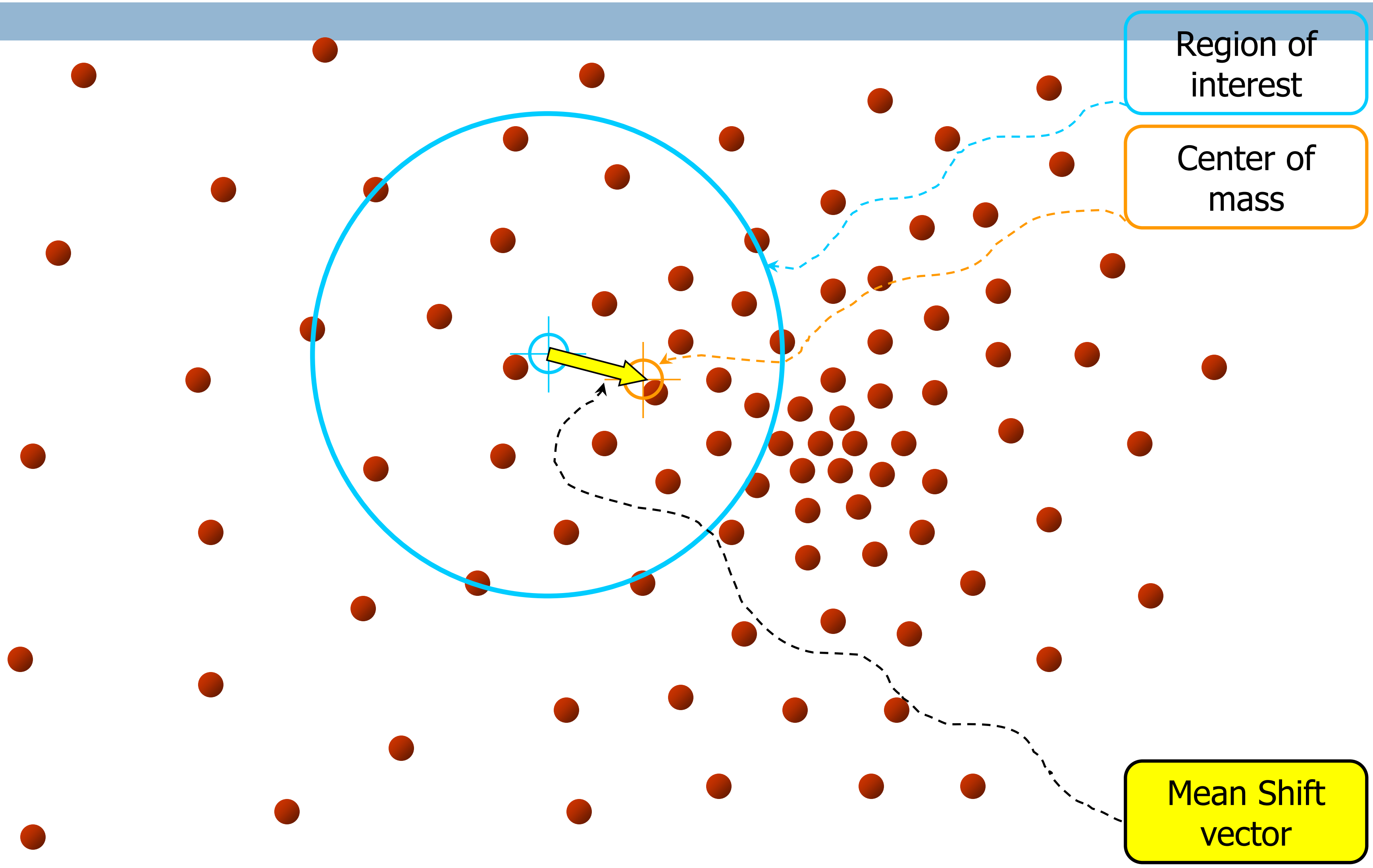
$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)} - \mathbf{x}\right]$$



- Translate the Kernel window by **m(x)**

$$g(\mathbf{x}) = -k'(\mathbf{x})$$

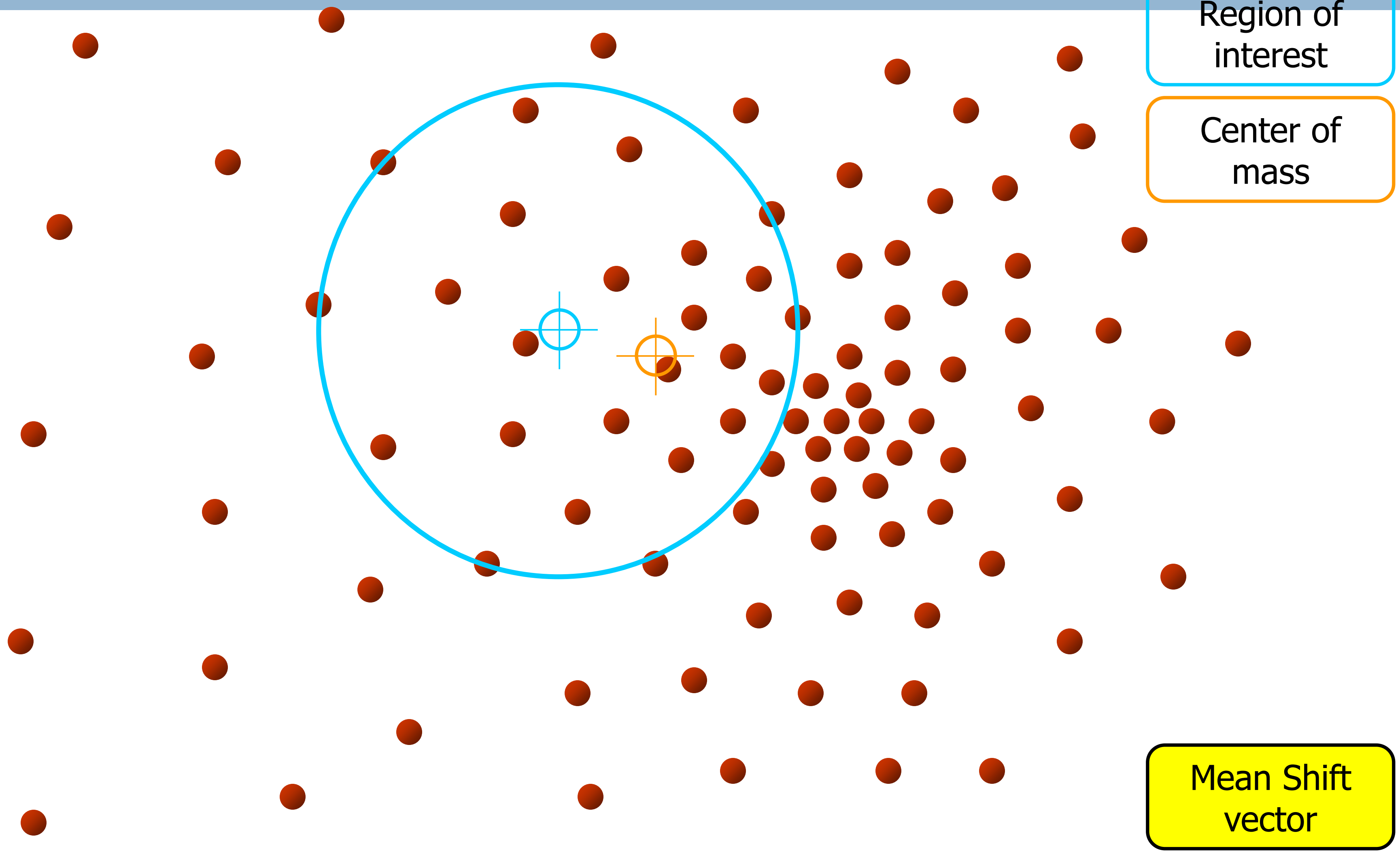
Intuitive Description



<u>**Objective**</u> **: Find the densest region**

Distribution of identical billiard balls

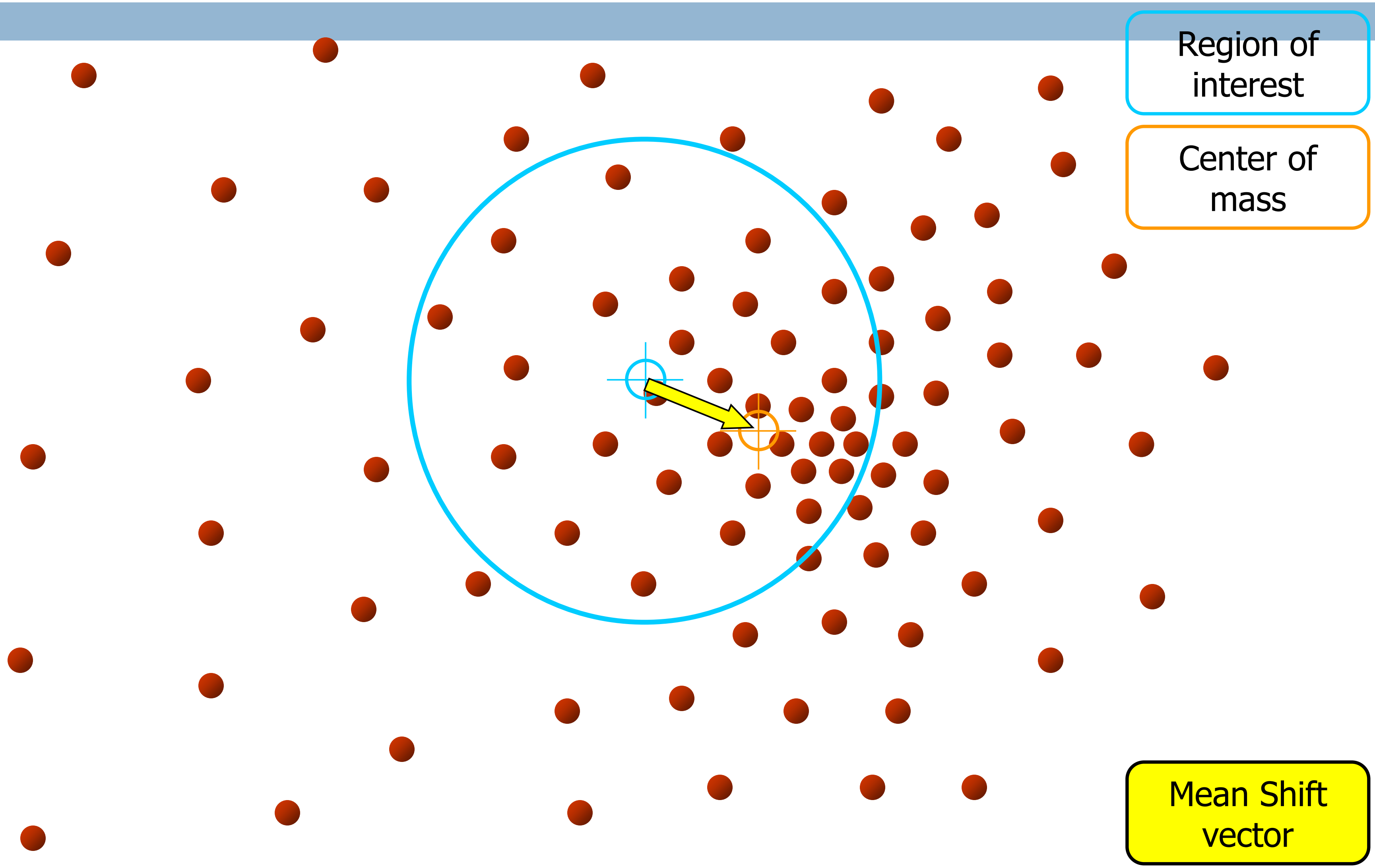
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

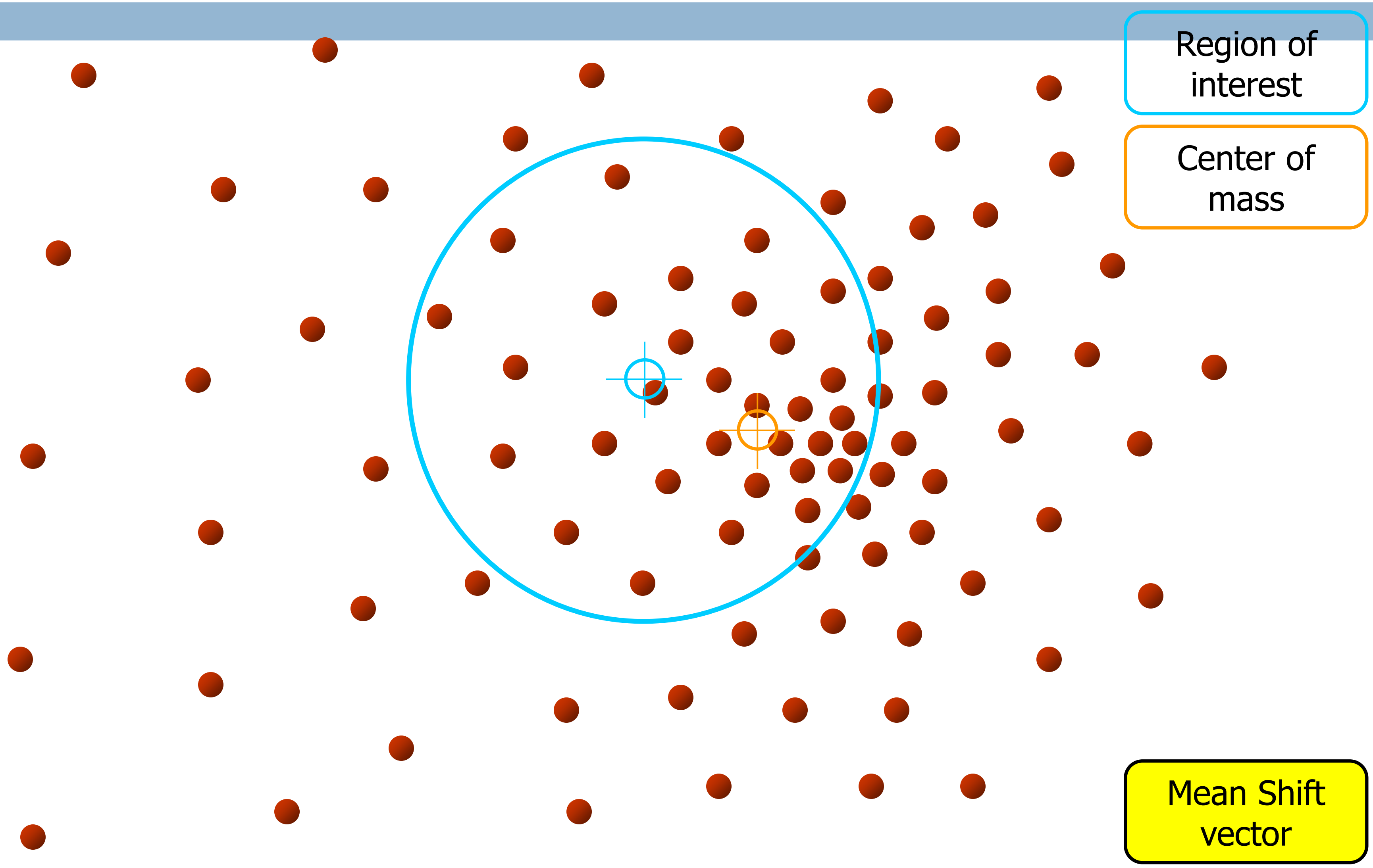
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

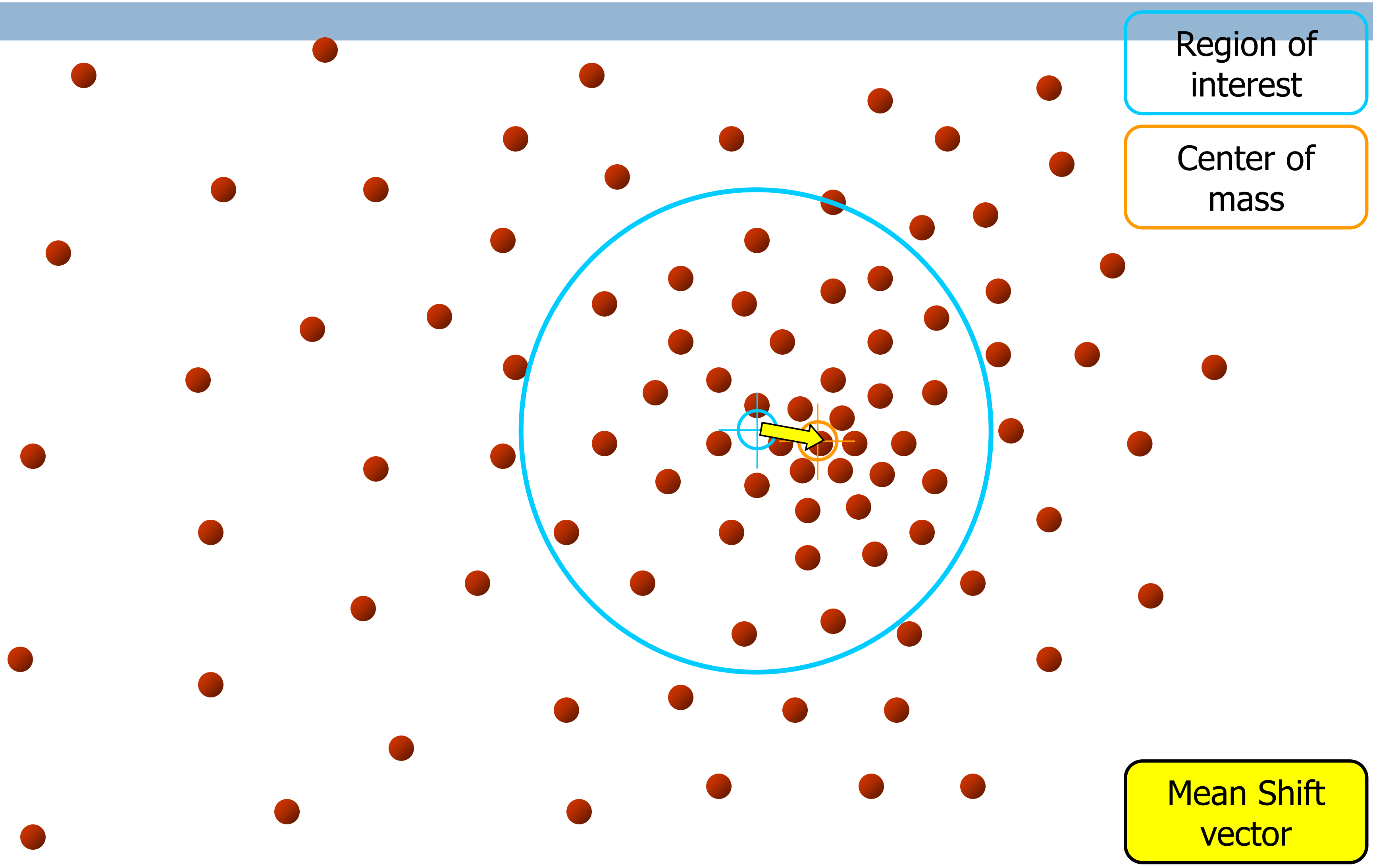
Intuitive Description



<u>Objective</u> : Find the densest region

Distribution of identical billiard balls

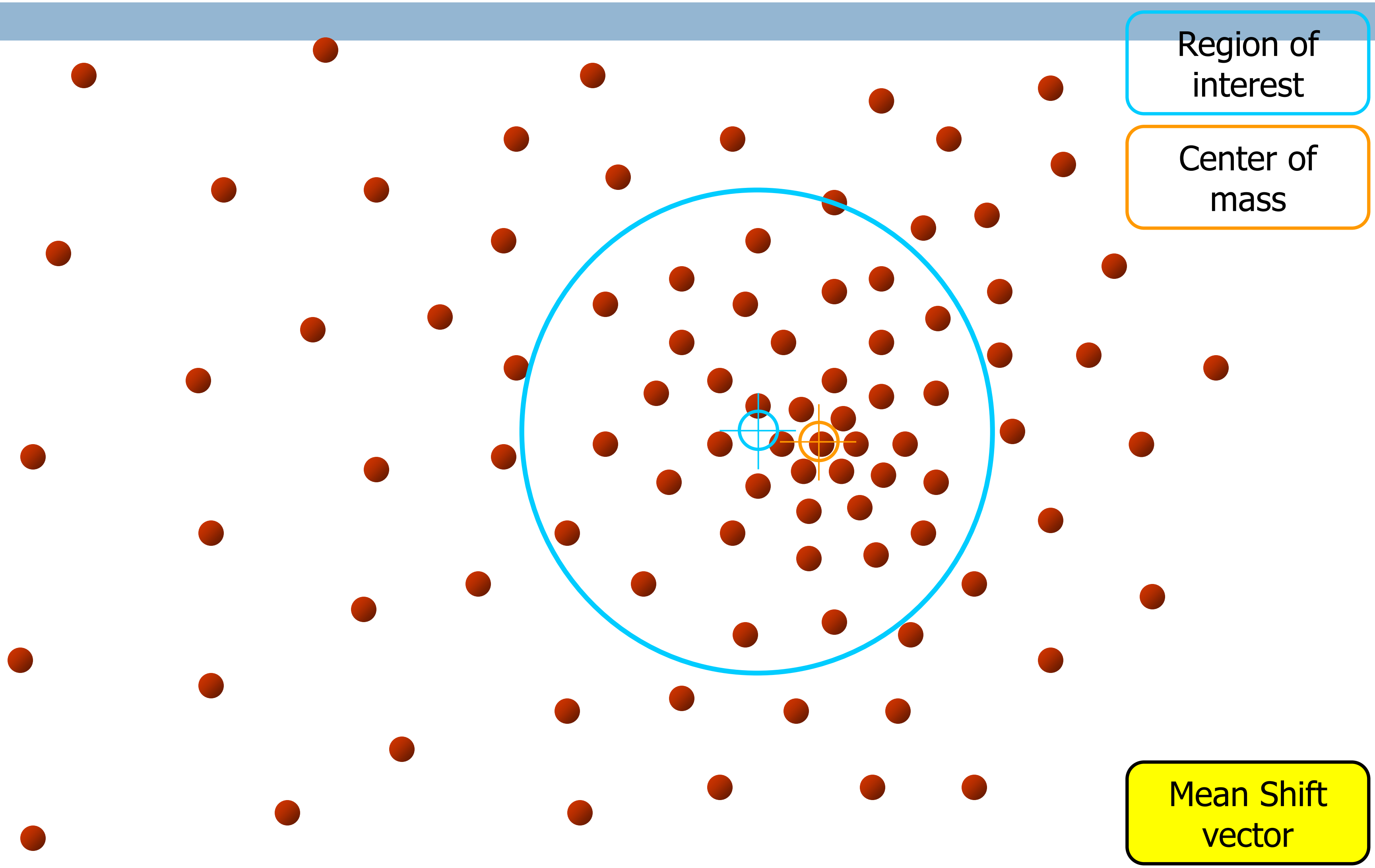
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

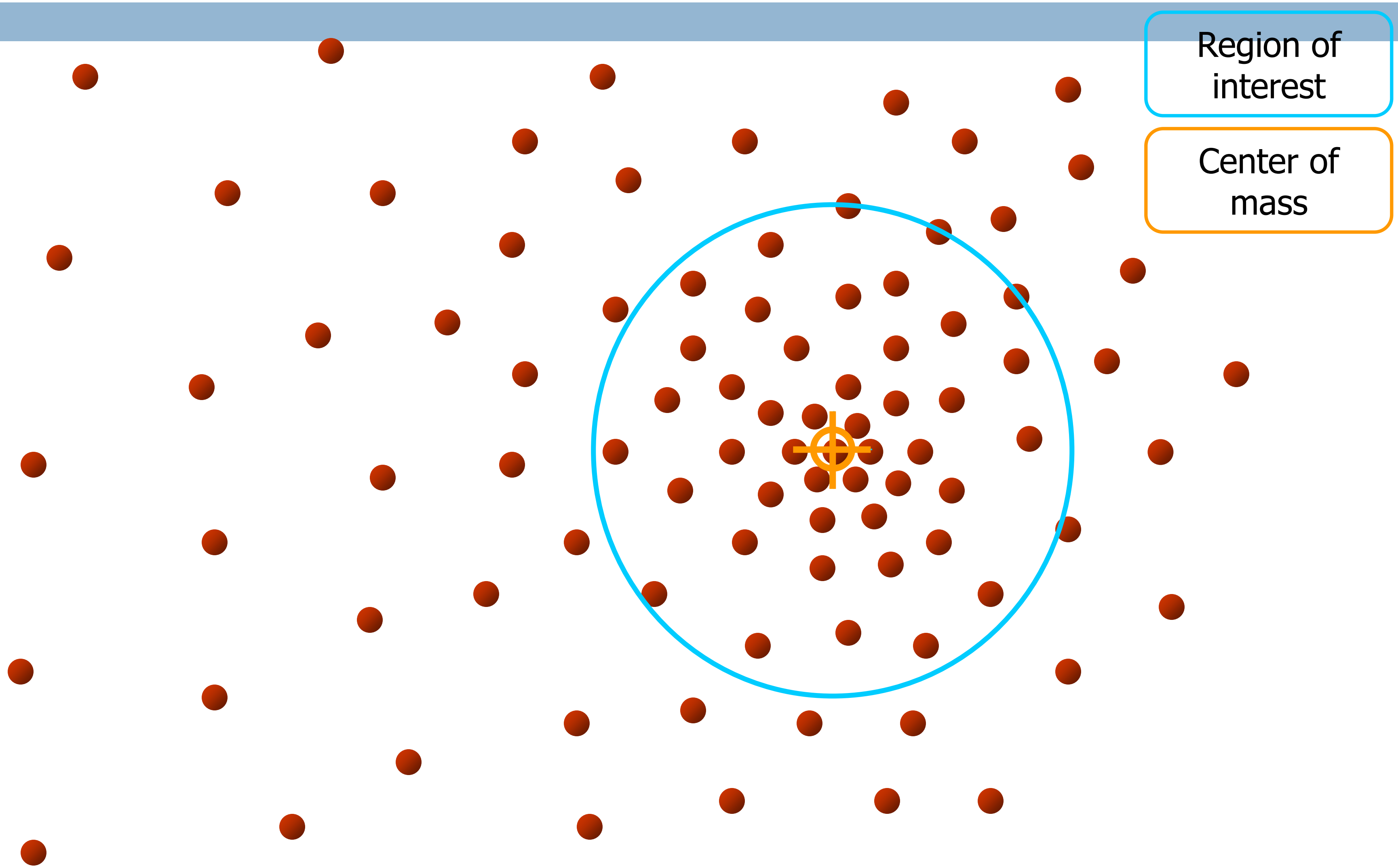
Intuitive Description



<u>Objective</u> : Find the densest region

Distribution of identical billiard balls

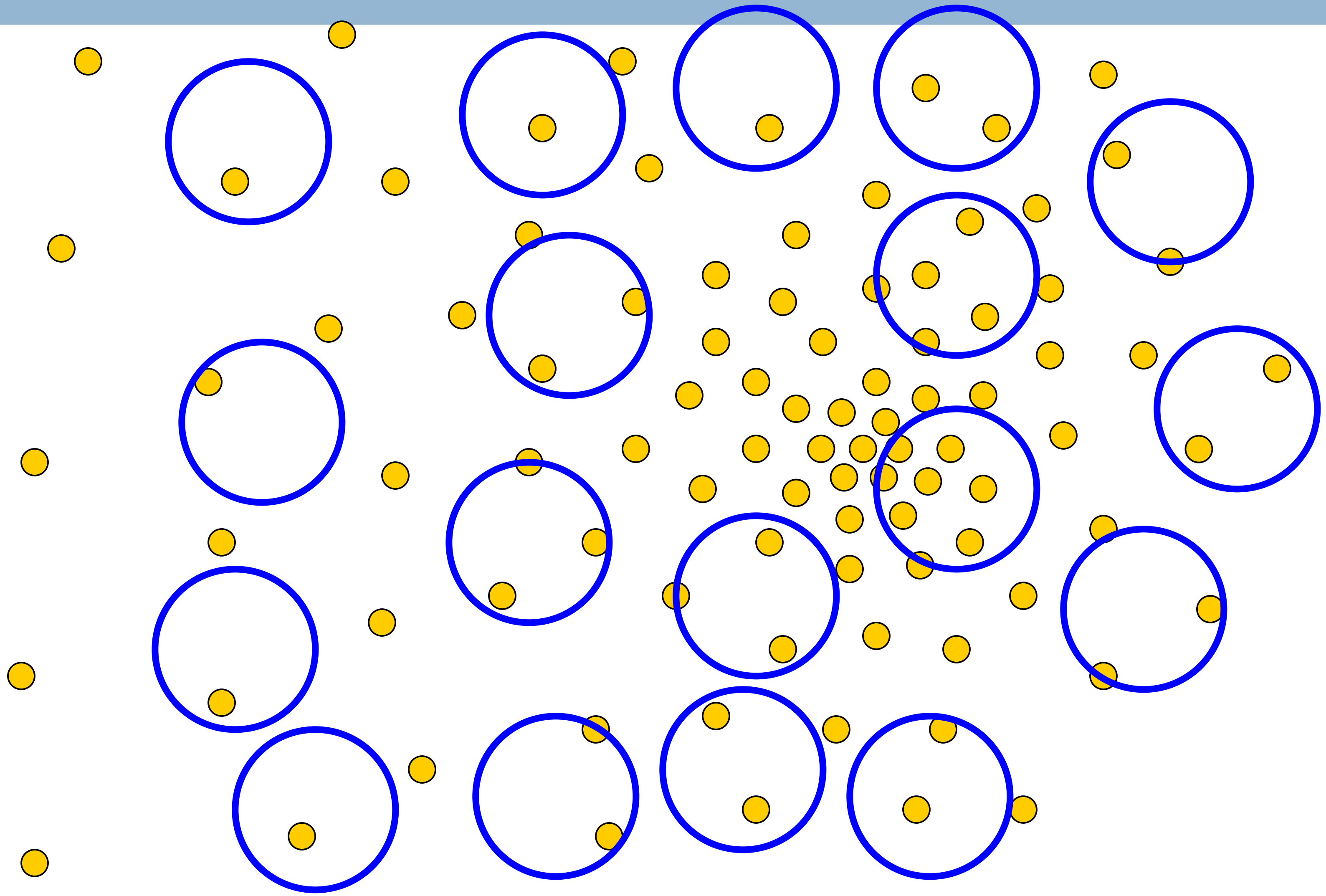
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

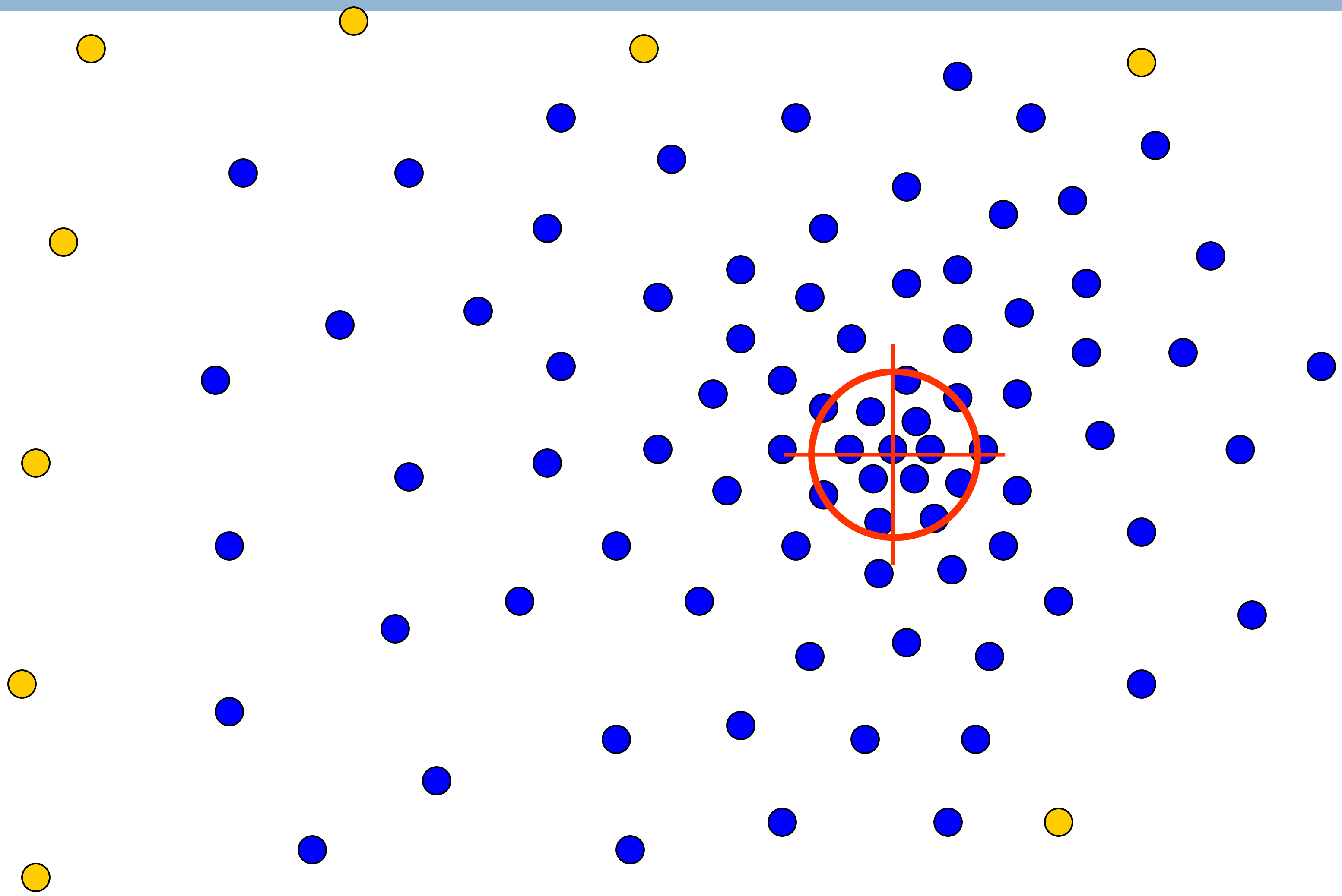
Real Modality Analysis



Tessellate the space with windows

Run the procedure in parallel

Real Modality Analysis



The blue data points were traversed by the windows towards the mode

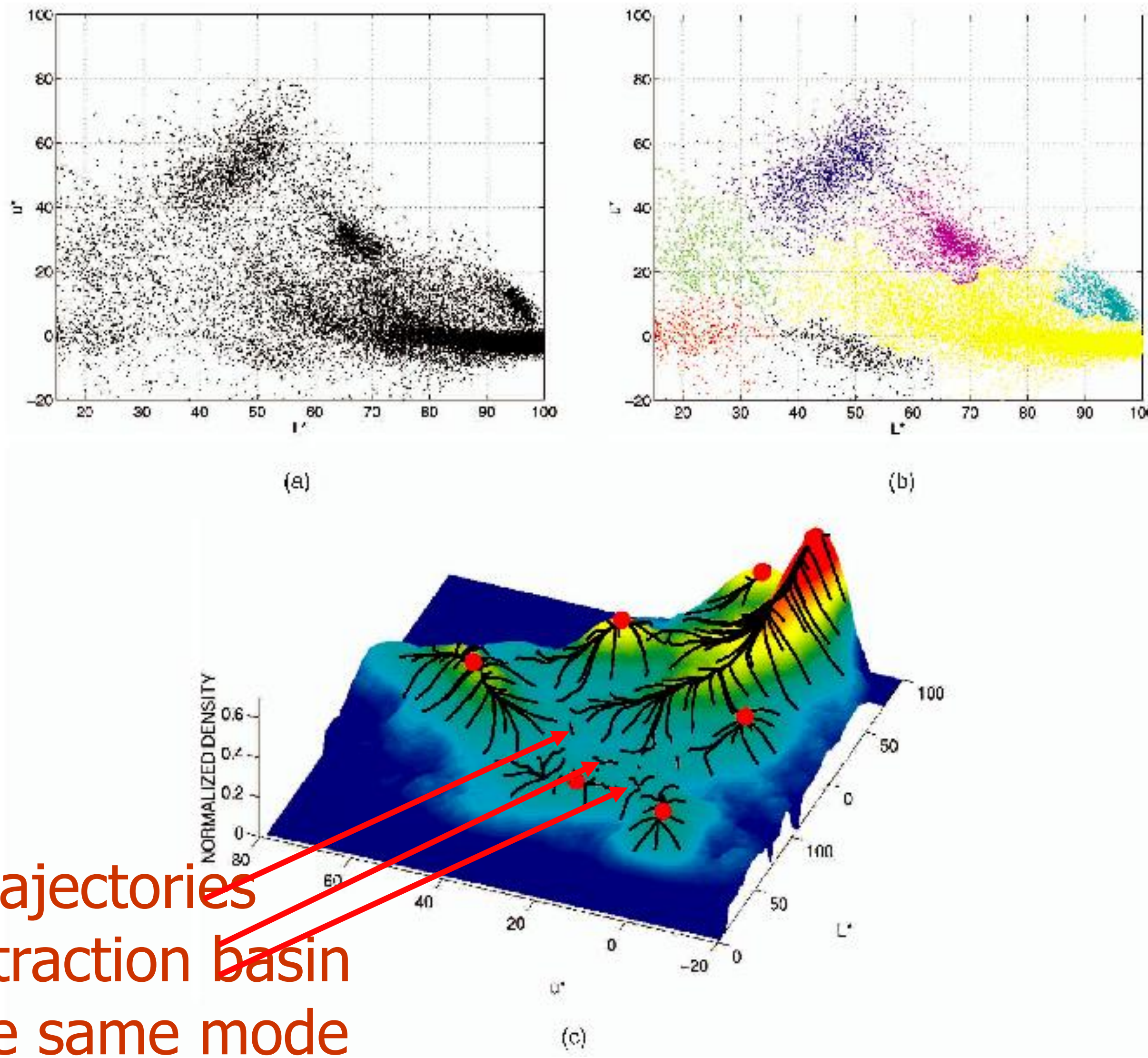
Mean-Shift Clustering

- Find features (color, gradients, texture, etc)
- The mean shift algorithm seeks *modes* of the given set of points
  1. Choose kernel and bandwidth
  2. For each point:
     a) Center a window on that point
     b) Compute the mean of the data in the search window
     c) Center the search window at the new mean location
     d) Repeat (b,c) until convergence
  3. Assign points that lead to nearby modes to the same cluster

Clustering - Example

2D space representation

Final clusters



Not all trajectories in the attraction basin reach the same mode

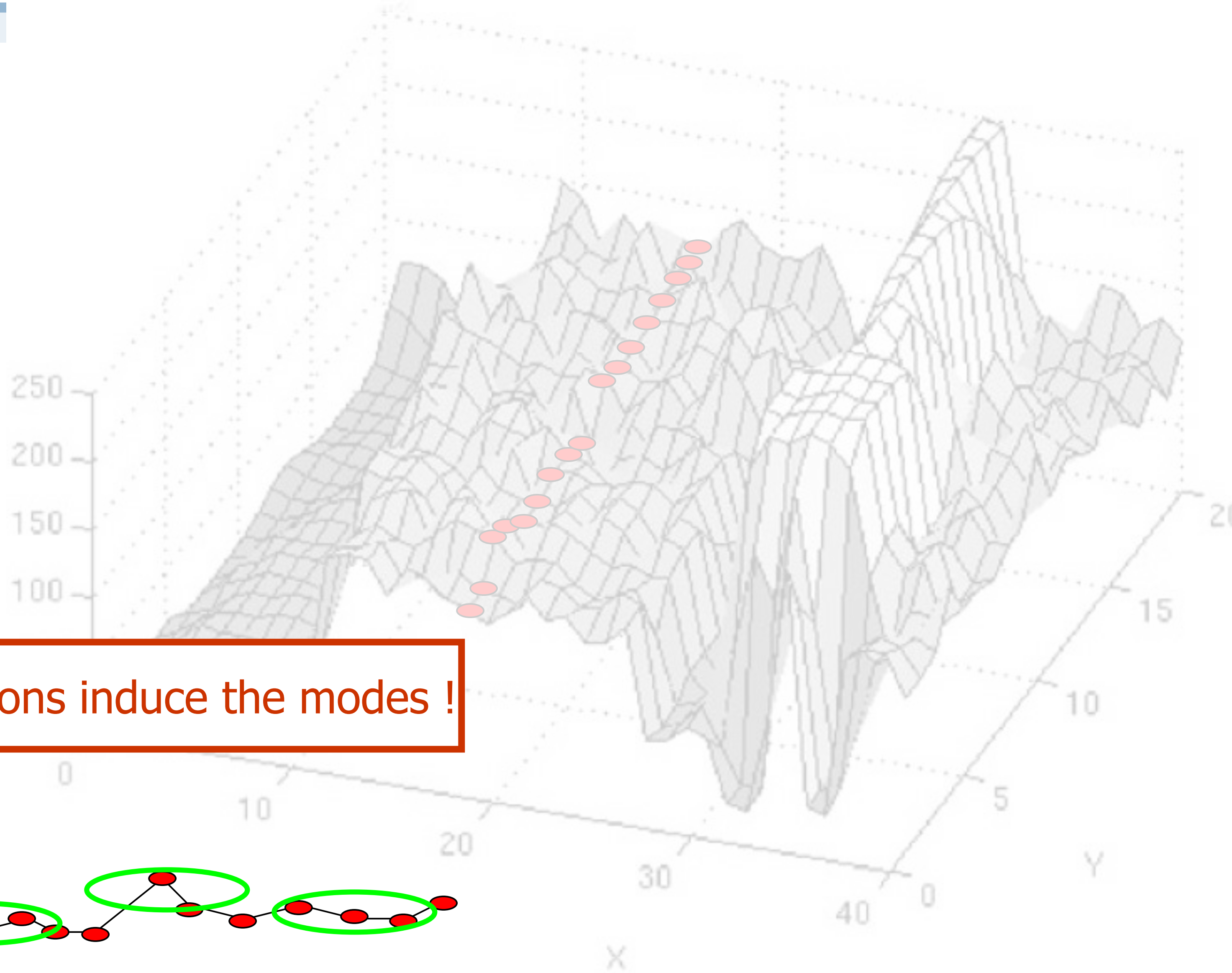
Discontinuity Preserving Smoothing

Feature space : Joint domain = spatial coordinates + color space

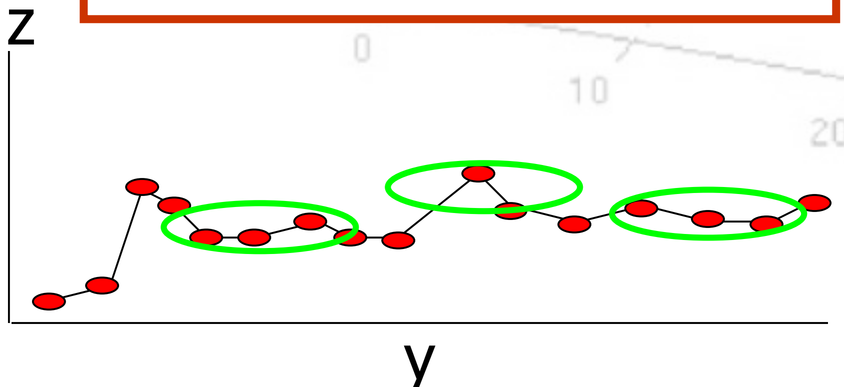
$$K(\mathbf{x}) = C \cdot k_s\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|\right) \cdot k_r\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|\right)$$

Meaning : treat the image as data points in the spatial and gray level domain

Discontinuity Preserving Smoothing



Flat regions induce the modes !



Discontinuity Preserving Smoothing - Example

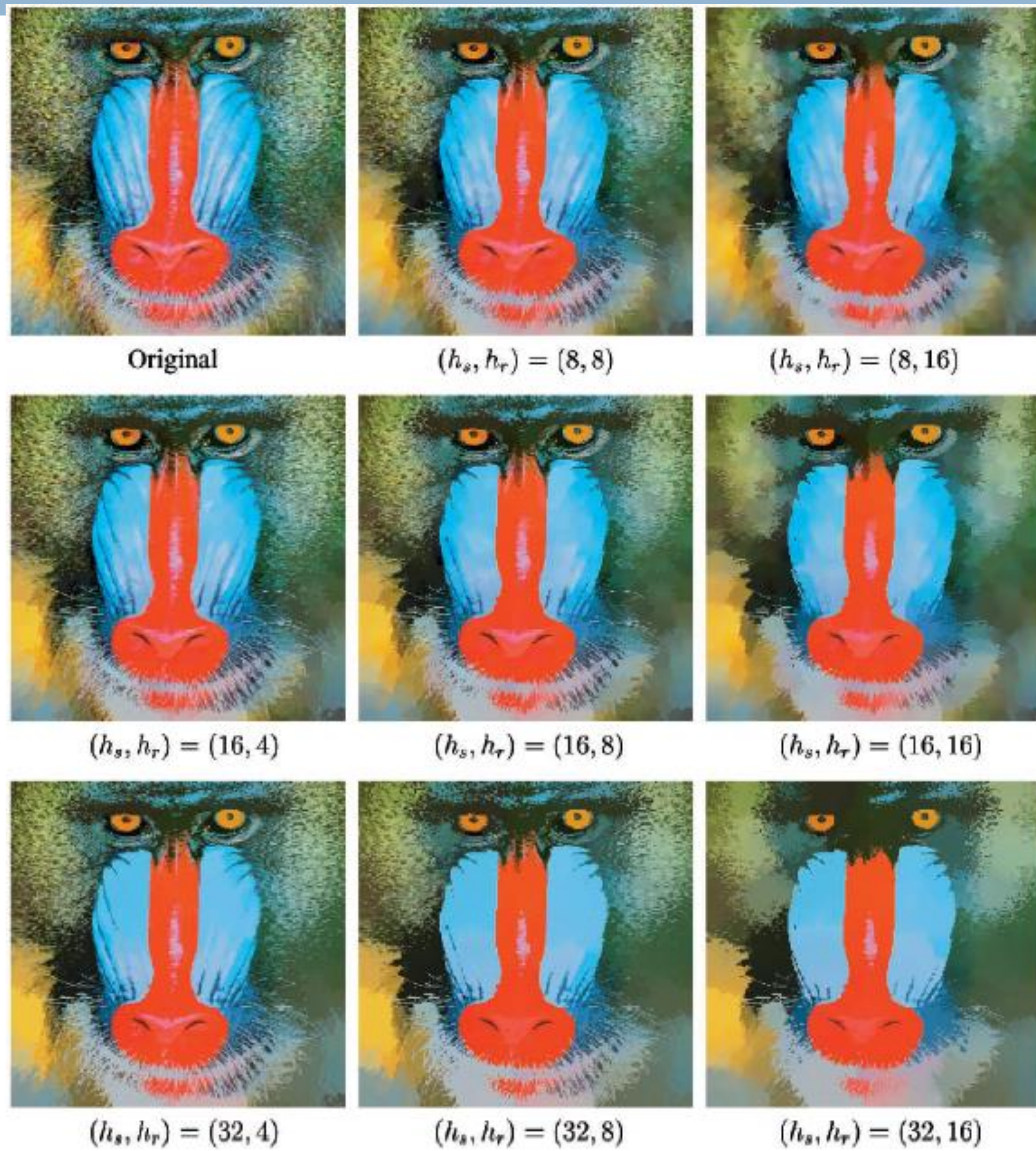



Parameters of the Mean-Shift Clustering

- h_s: spatial resolution parameter
  - Affects the smoothing, connectivity of segments
- h_r: range resolution parameter
  - Affects the number of segments
- M: size of smallest segment
  - Should be choosen based on size of noisy patches

Discontinuity Preserving Smoothing

The effect of window size in spatial and range spaces



Mean Shift Segmentation

Segment = Cluster, or Cluster of Clusters

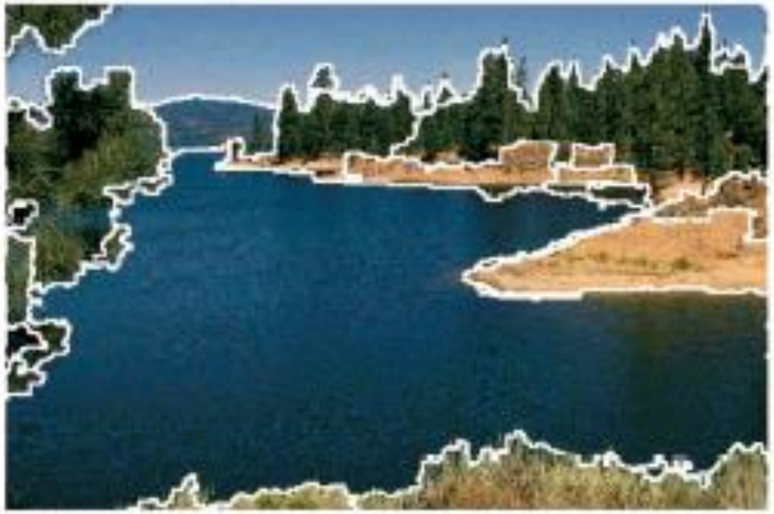
Algorithm:

- Run Filtering (*discontinuity preserving smoothing*)
- Cluster the clusters which are closer than window size

Mean Shift Segmentation - Example

when feature space is only gray levels…

Mean Shift Segmentation - Example

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
  - Output depends on window size
    - Inappropriate window size can cause modes to be merged, or generate additional “shallow” modes
    - Use adaptive window size: CAMshift (Continuously Adaptive Meanshift)
  - Computationally expensive
  - Does not scale well with dimension of feature space
- When to use it
  - Oversegmentatoin
  - Multiple segmentations
  - Tracking, clustering, filtering applications

Other Segmentation Techniques

- Morphological Watersheds
- Graph-based methods (graph---cut, random walk)
- Shape-based methods (level set, active contours)
- Energy minimization methods (MRF,..)
- Machine Learning based methods