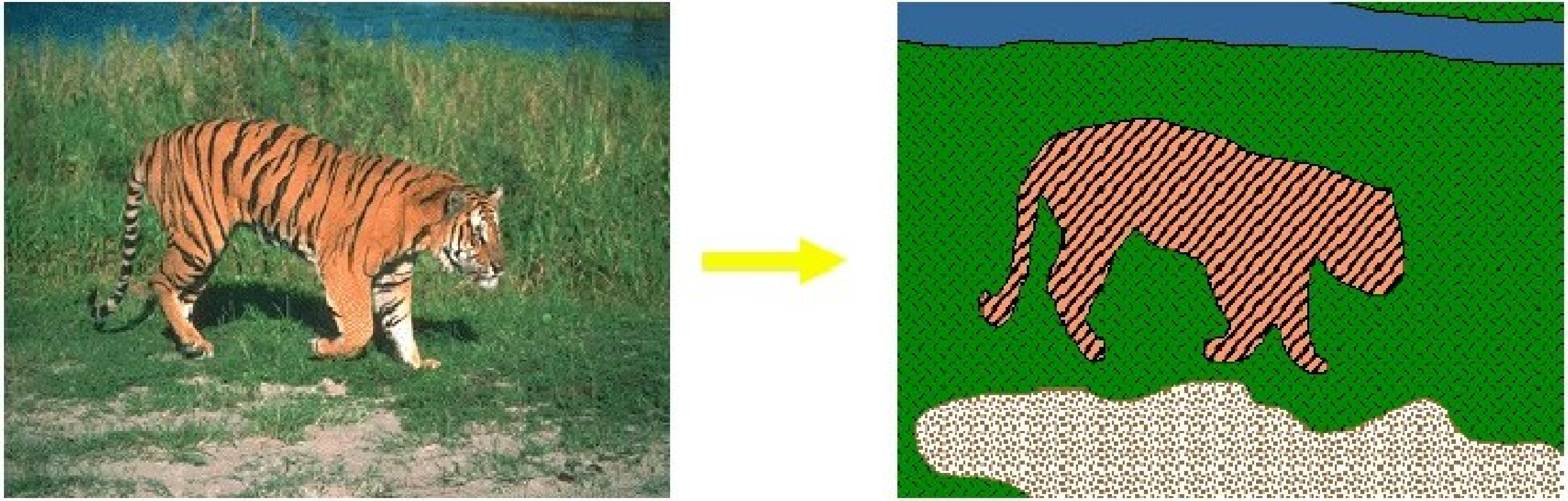
UNIVERSITÀ
CA' FOSCARI
VENEZIA

# Segmentation

Andrea Torsello
DAIS
Università Ca' Foscari
via Torino 155,
30172 Mestre (VE)

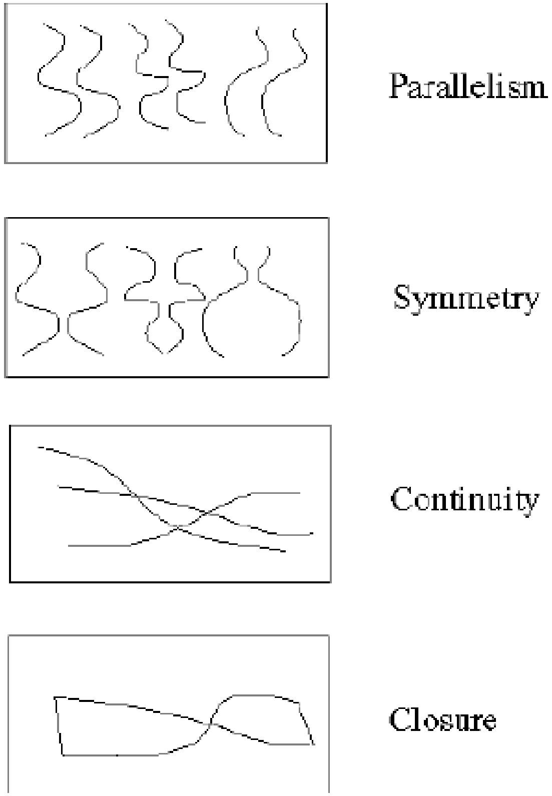
# Segmentation

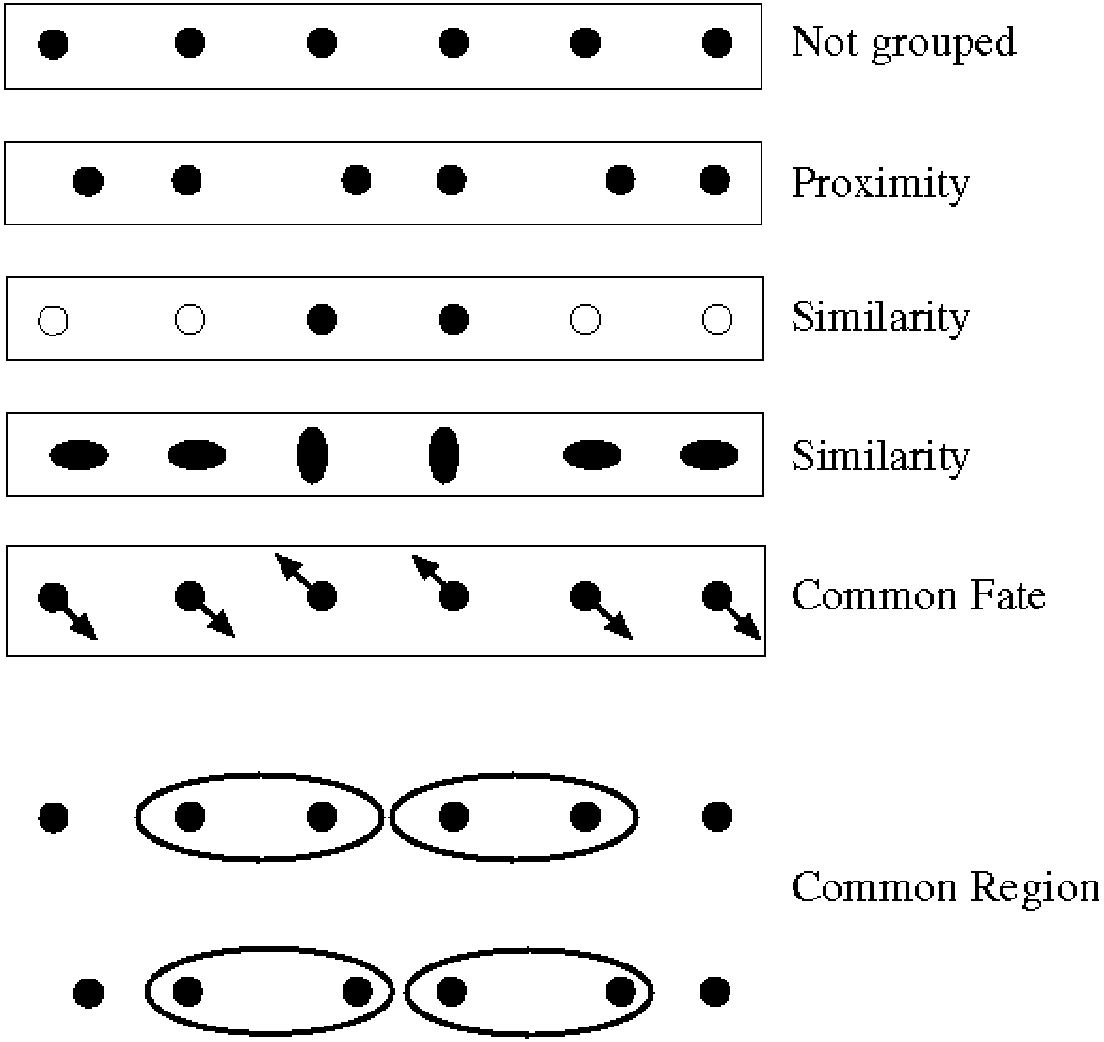
- Segmentation is the task of dividing an image into regions
- Regions are:
  - Characterized by their visual properties
  - Linked with underlying semantics of the image (objects in th scene)
  - Offer a compact representation of the elements to be



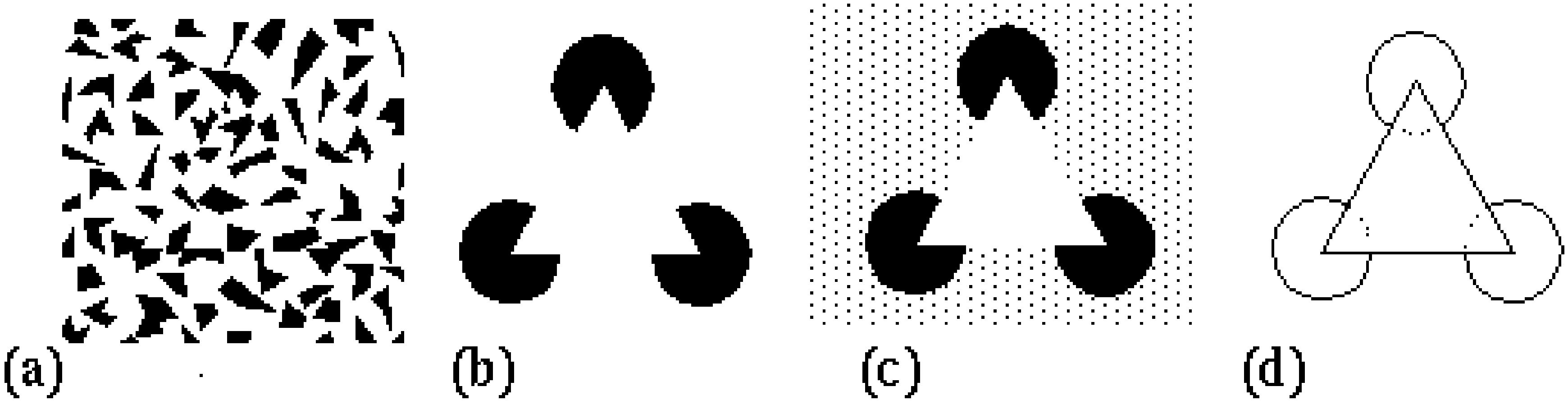
# Gestalt

- Gestalt psychology identifies several properties that result in grouping/segmentation

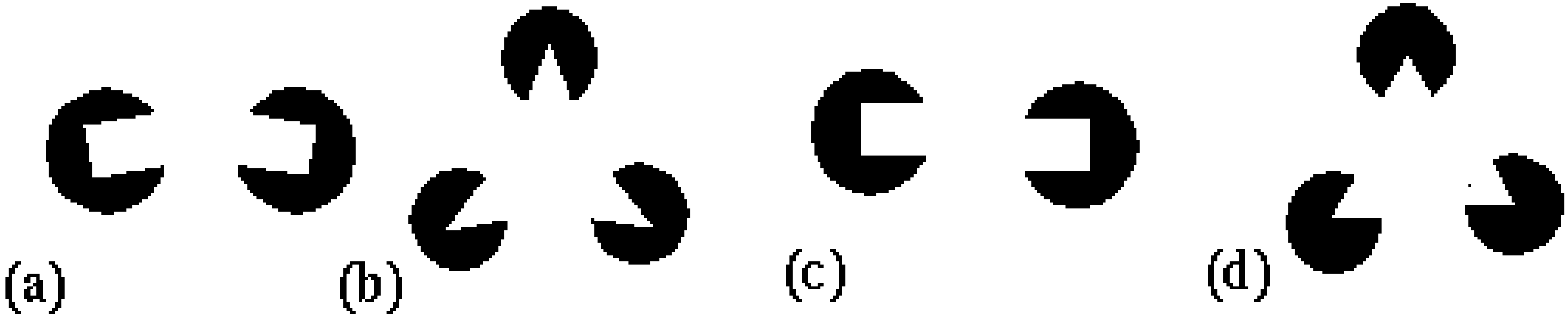




# Groupings by Invisible Completions

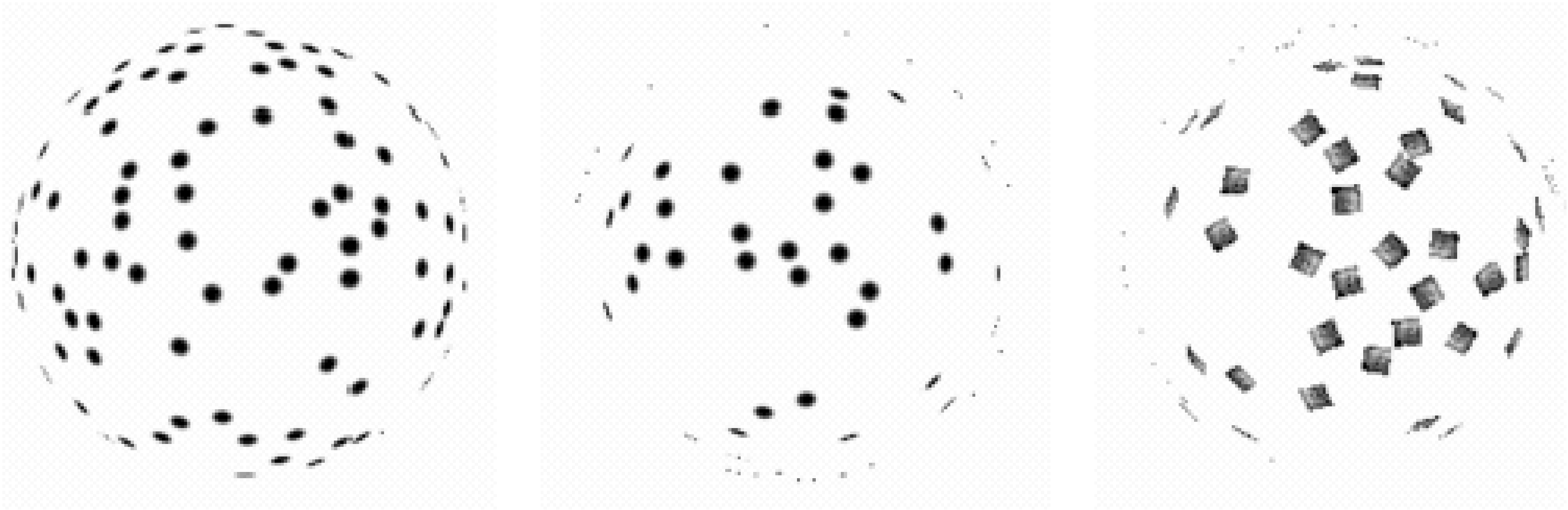


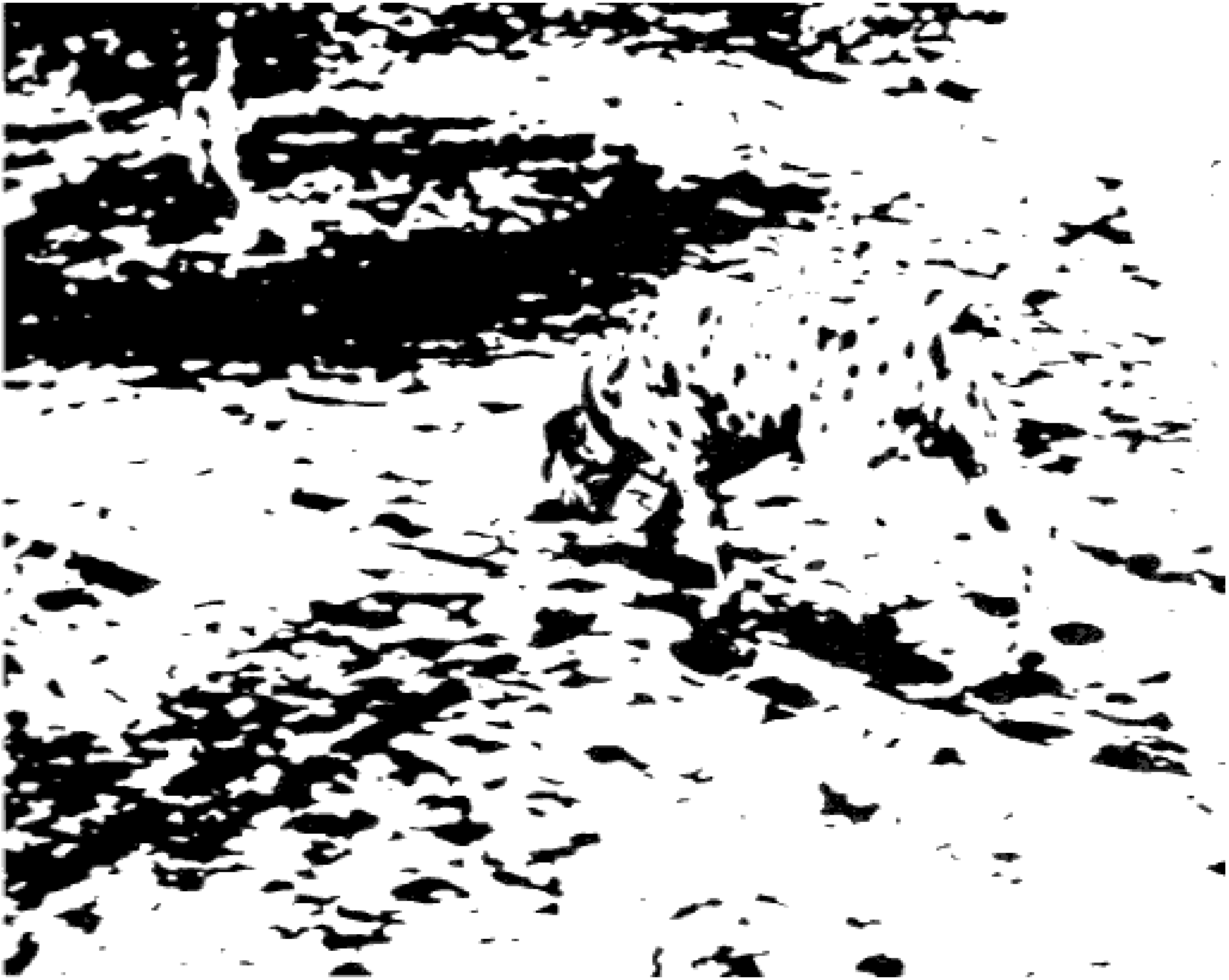
Stressing the invisible groupings:

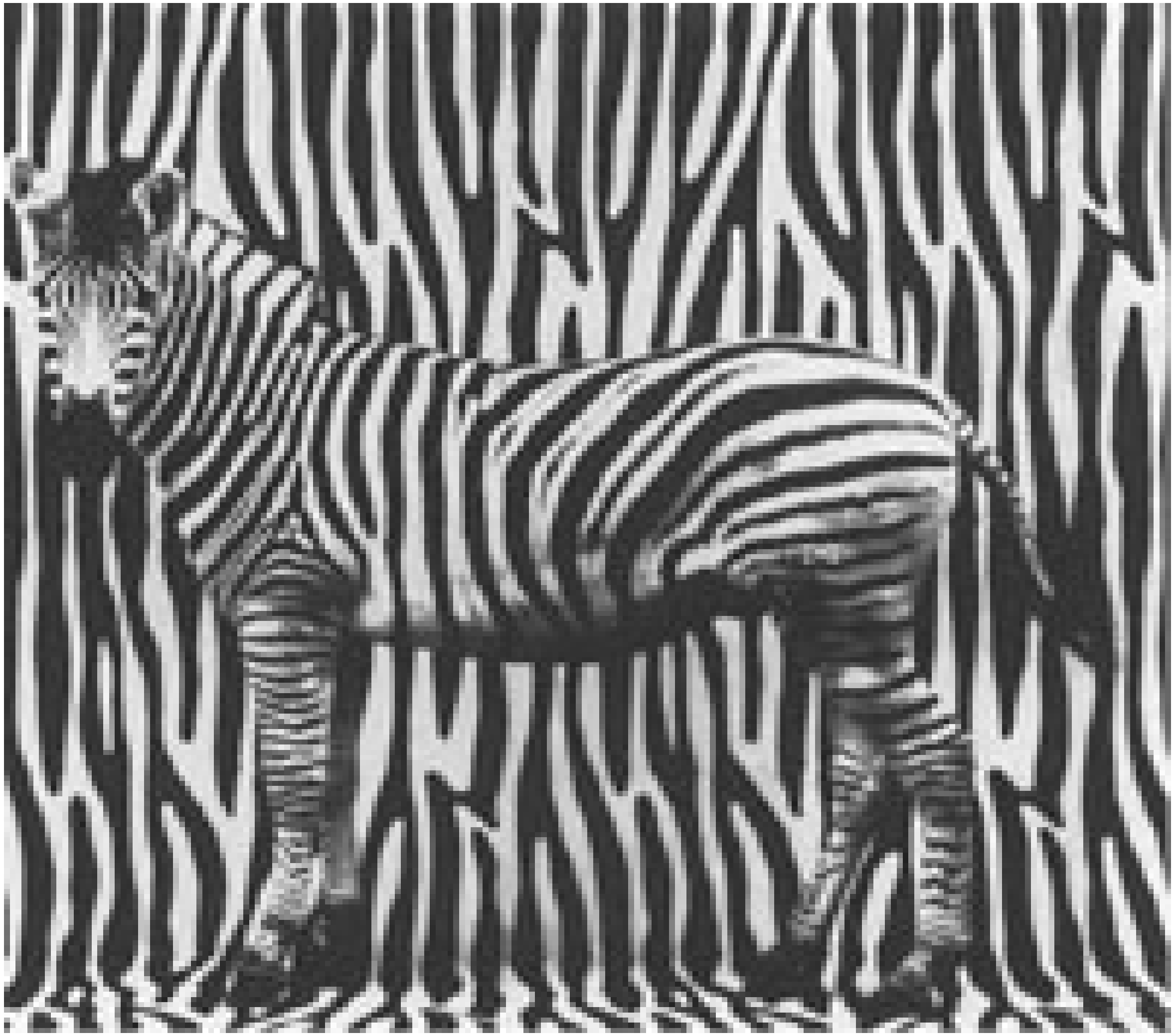


# 3D cues

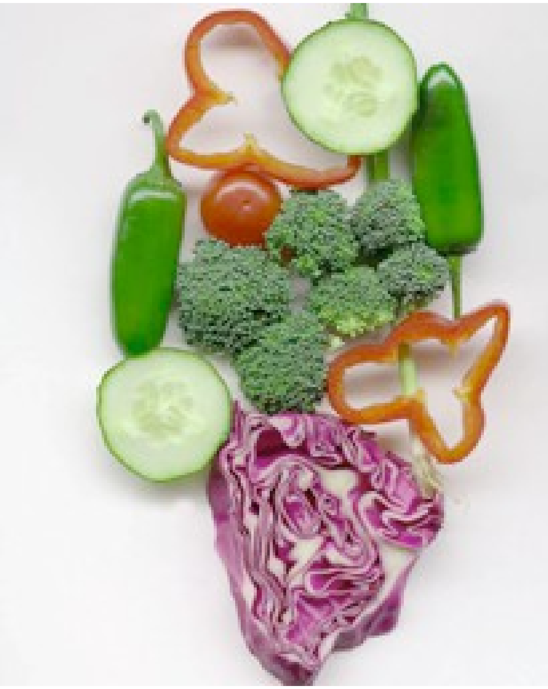
Why do these tokens belong together?



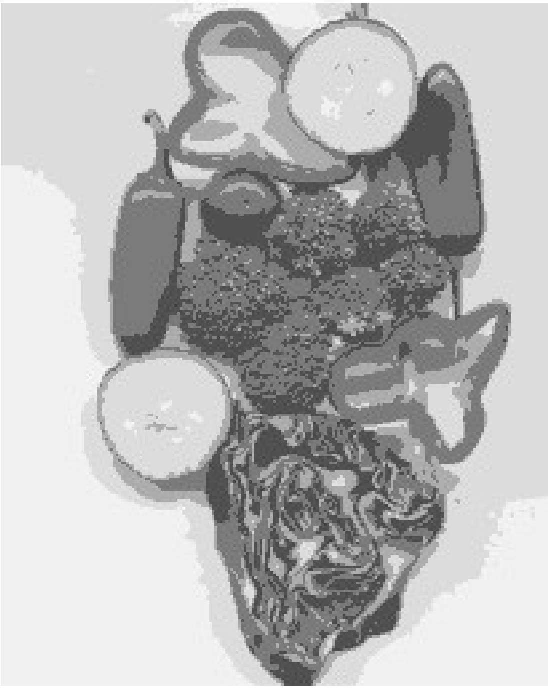




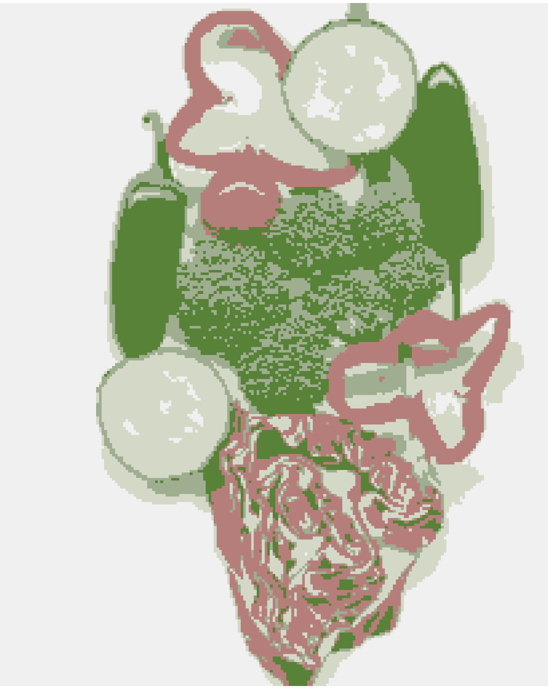
# Segmentation as clustering



Image

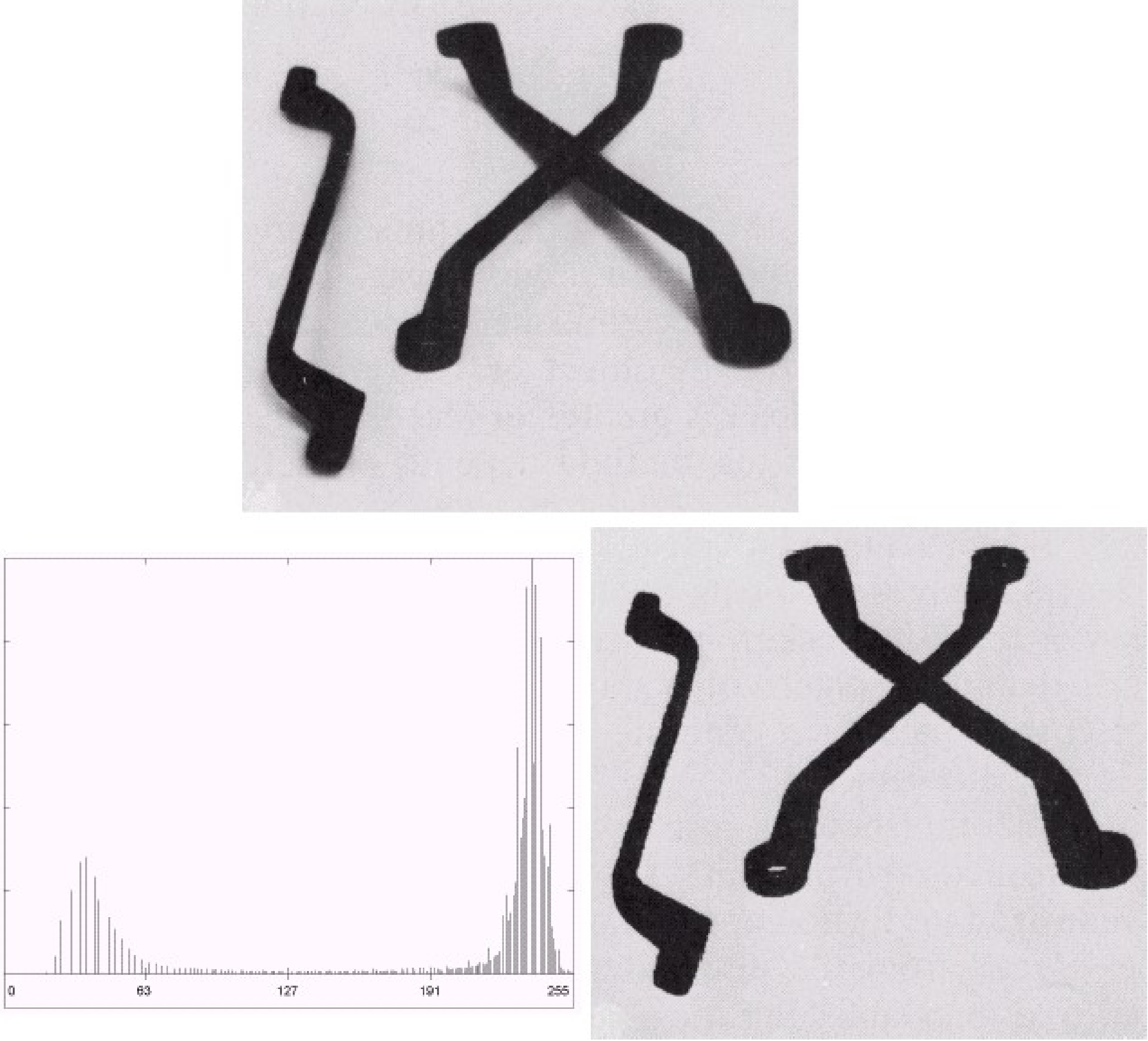


Clusters on intensity



Clusters on color

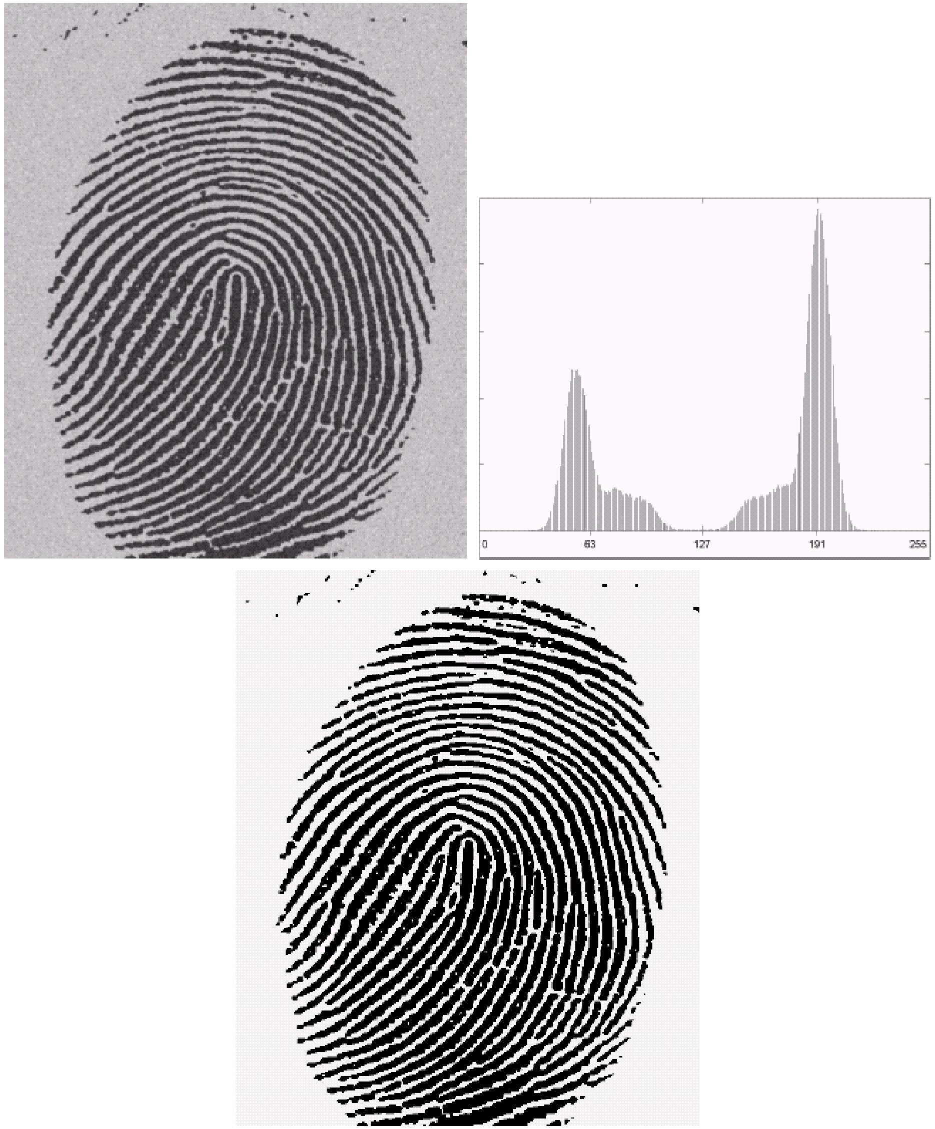
# Thresholding



a
b c

**FIGURE 10.28** (a) Original image. (b) Image histogram. (c) Result of global thresholding with *T* midway between the maximum and minimum gray levels.

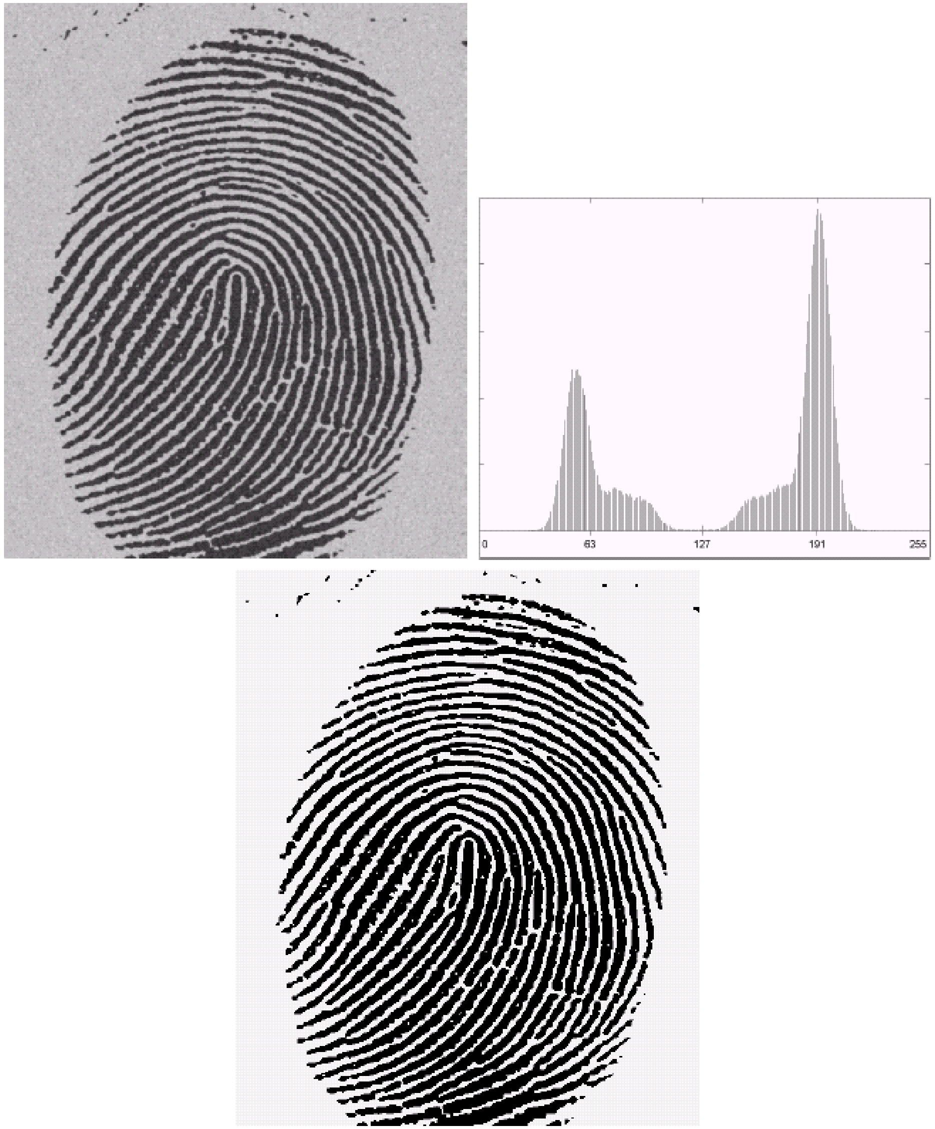
# Estimating the threshold



a b
c

**FIGURE 10.29** (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

# Estimating the threshold



a b
c

**FIGURE 10.29** (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

# Otsu’s Thresholding Method

- Based on a very simple idea: Find the threshold that minimizes the weighted within-class variance.
- This turns out to be the same as maximizing the between-class variance.
- Operates directly on the gray level histogram [e.g. 256 numbers, P(i)], so it’s fast (once the histogram is computed).
- Assumptions:
  - Histogram (and the image) are bimodal.
  - No use of spatial coherence, nor any other notion of object structure.
  - Assumes uniform illumination, so the bimodal brightness behavior arises from object appearance differences only.

# Otsu’s Thresholding Method

- Let P(i) be the frequency (histogram) of intensity le
- Estimate the class probabilities after a sepataration
intensity t as:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

- Theclass means are:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

- The individual class variances are:

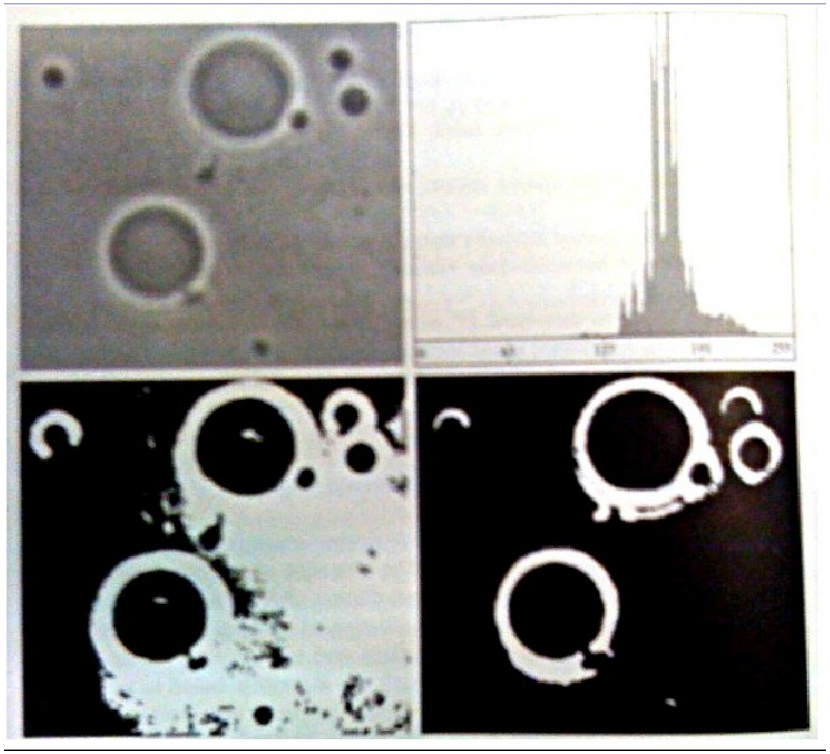
$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \qquad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

# Otsu's Thresholding Method

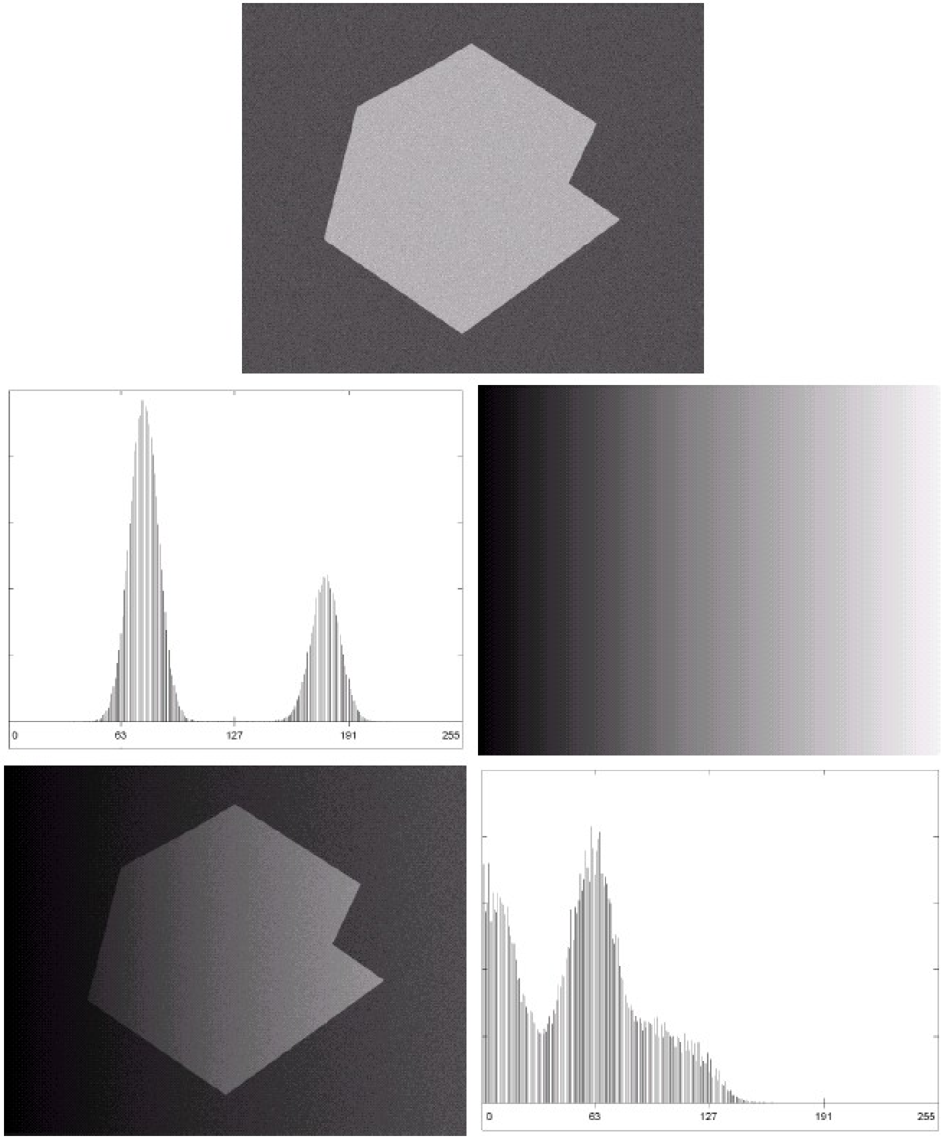
- Finally, the *weighted within-class variance* is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

- The optimal threshold  minimizes this value



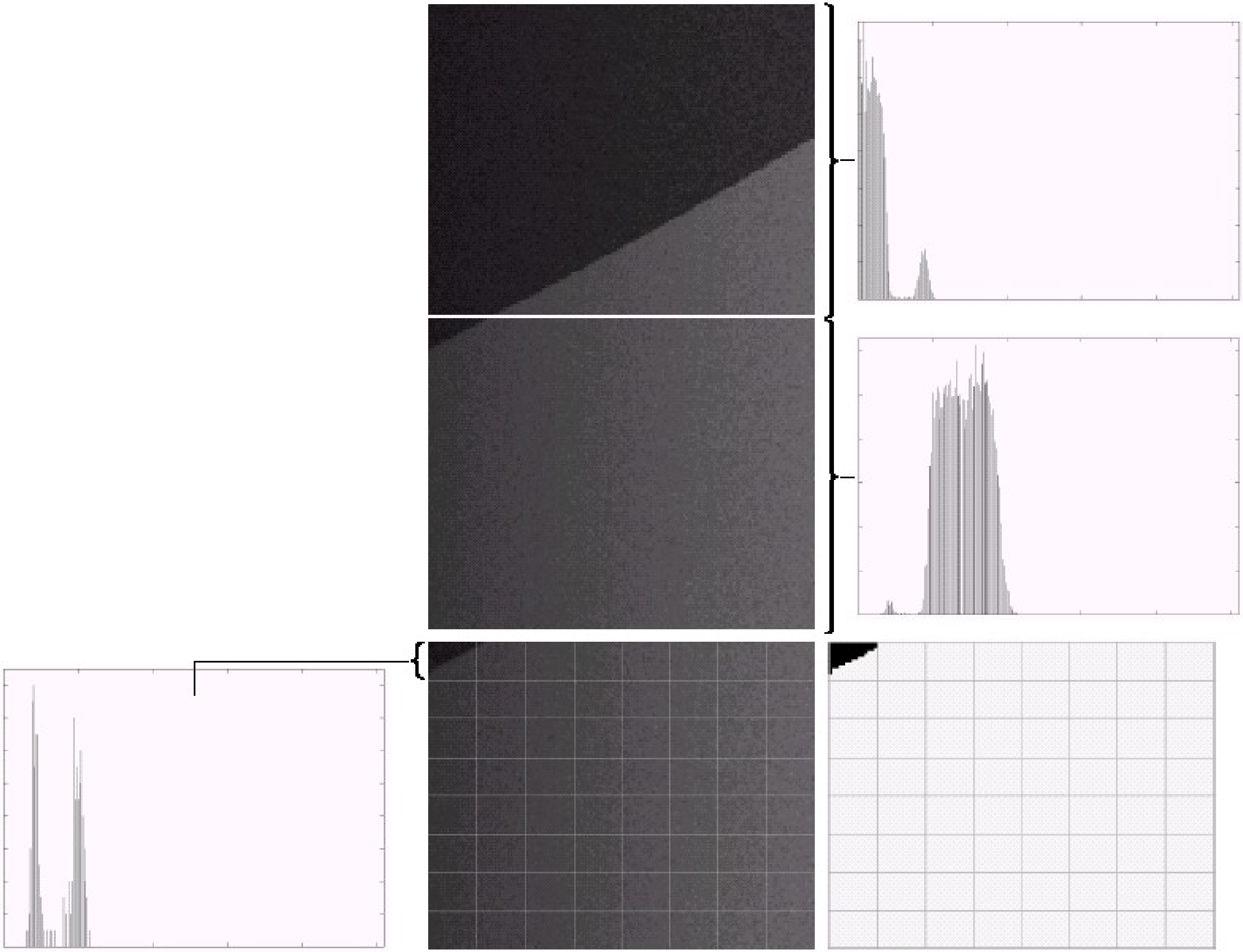
# Effects of illumination gradients



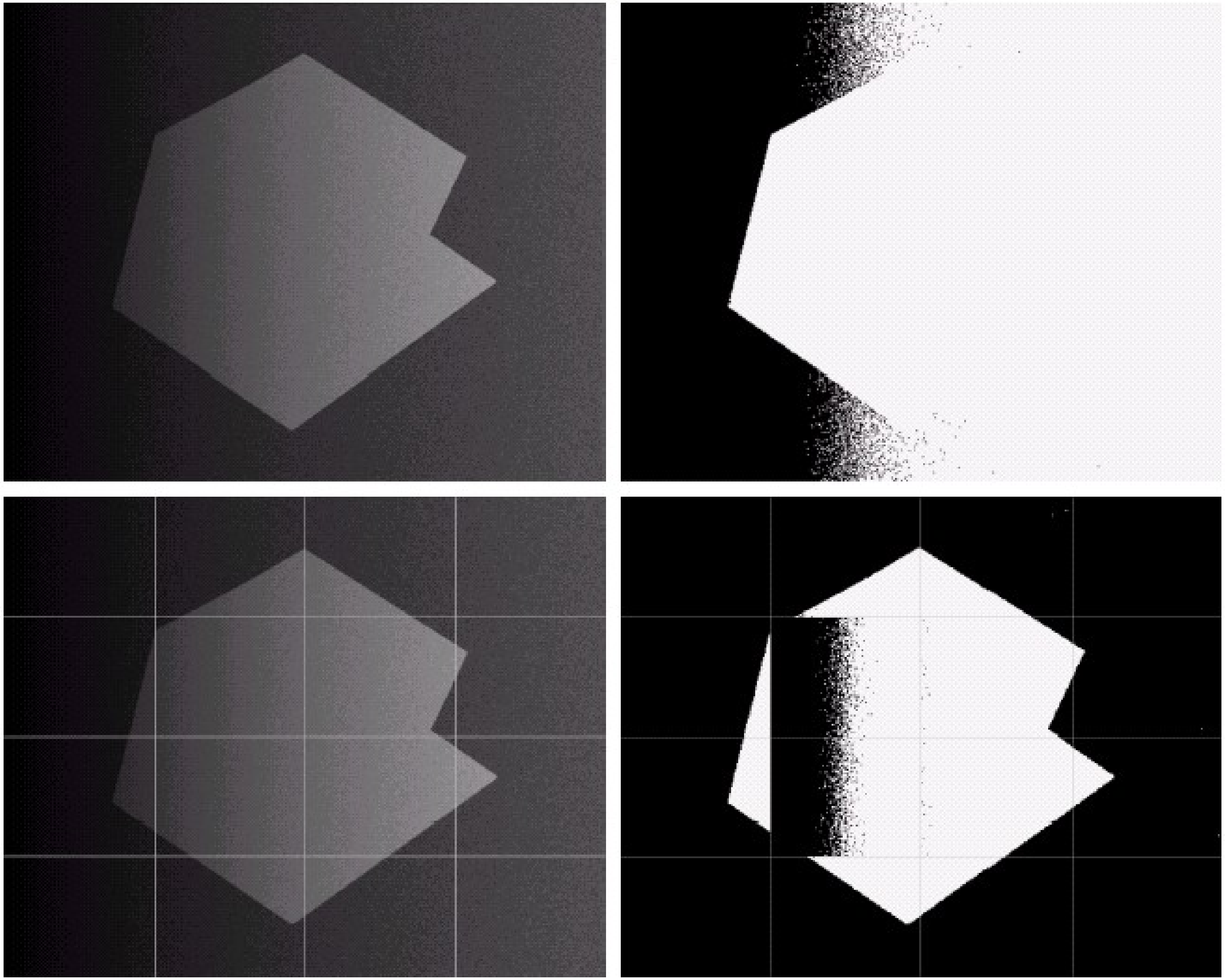
a
b c
d e

**FIGURE 10.27** (a) Computer generated reflectance function. (b) Histogram of reflectance function. (c) Computer generated illumination function. (d) Product of (a) and (c). (e) Histogram of product image.

# Adaptive Threshold



# Adaptive Threshold



a b
c d

**FIGURE 10.30** (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

# k-means

- We are given an unlabeled training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- We want to group the data into a few cohesive clusters.
  - Assume for the moment that
    - the number K of clusters is given
    - The clusters form a partition of the data: data-points are in one and only one cluster

- How do we define cohesiveness?
- Intuitively, we might require that intra-cluster distances are compared with the inter-cluster distances.
- We can formalize this notion by introducing a set of vectors

  $\mu_k$ , where k = 1, ..., K
- $\mu_k$ is a prototype associated with the k th cluster, representing the centers of the clusters.

- Our goal is then to find
  - an assignment of data points to clusters
  - vectors $\{\mu_k\}$,
- such that the sum of the squares of the distances of the data point to the cluster center $\mu_k$ , is a minimum.

# k-means

- Let us introduce a binary indicator variable $r_{nk} \in \{0,1\}$ describing which of the K clusters data point $x_n$ is assigned to.
- We can then define a *distortion measure* as

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

**K-means algorithm (Lloyd, 1982)**

- We optimize J s through an iterative procedure involving two successive steps corresponding to
  - optimization with respect to the $r_n$
  - optimization with respect to the $\mu_k$ .
- 
- First we choose some initial values for the $\mu_k$ .
  - In the first step Then minimize J with respect to the $r_n$ , keeping the $\mu_k$ fixed.
  - In the second step we minimize J with respect to the $\mu_k$ , keeping $r_n$ fixed.
- This two-stage optimization is repeated until convergence.

# k-means

***Optimization of*** r$_{nk}$

- Since J is linear in r$_{nk}$, we can give minimum in a closed form solution by setting r$_{nk}$=1 for whichever value of k gives the m$\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

***Optimization of*** $\mu_k$

- Function J is quadratic in $\mu_k$, and it can be minimized by setting its derivative with respect to $\mu_k$ to zero giving

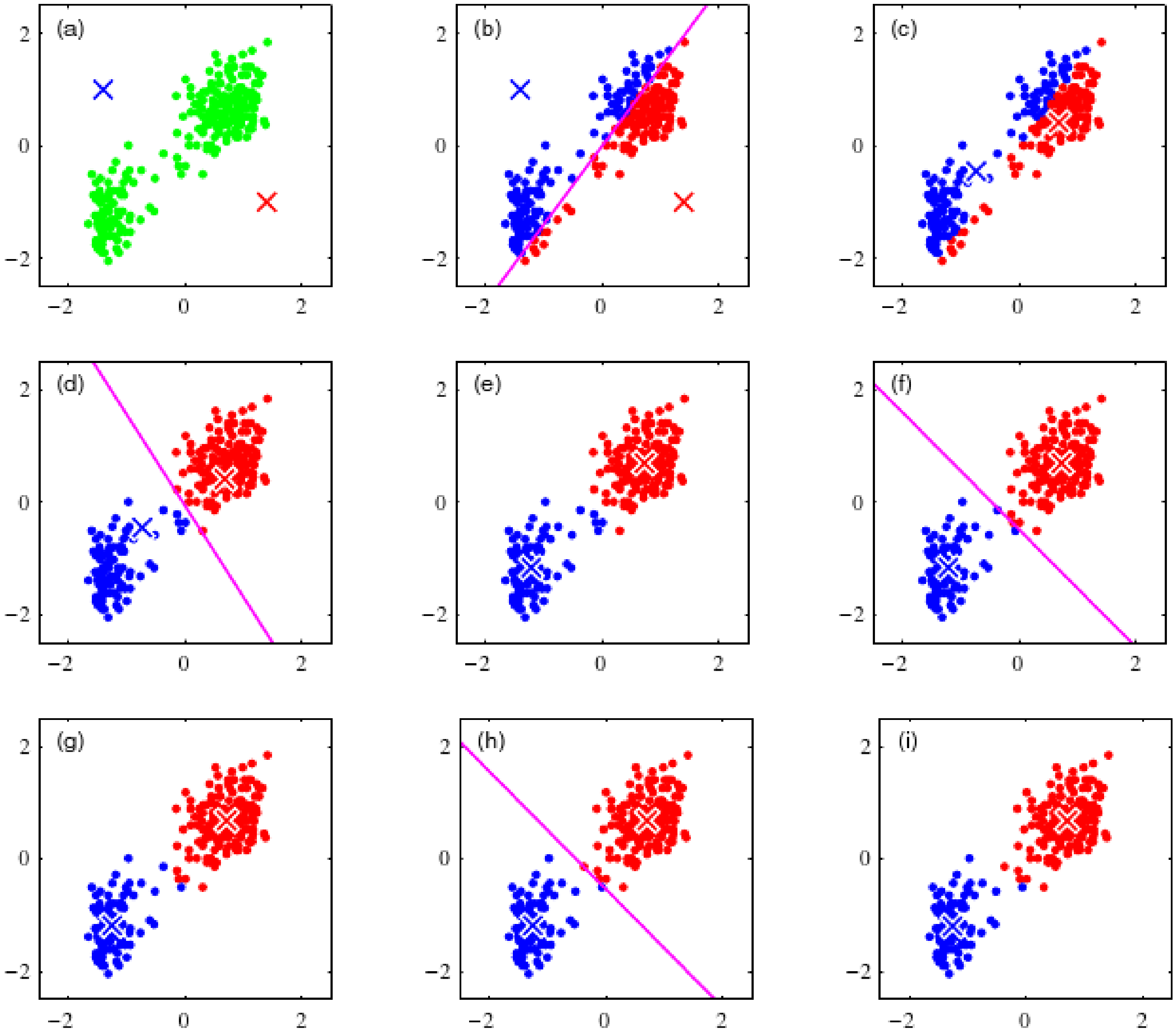
$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \qquad\qquad \boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}.$$

- This sets $\mu_k$ equal to the mean of all of the data points $\mathbf{x}_n$ assigned to cluster $k$, hence the name K-means algorithm.

- The two phases are repeated in turn until there is no further change in the assignments (or until some maximum number of iterations is exceeded).

# k-means

UNIVERSITÀ
CA' FOSCARI
VENEZIA

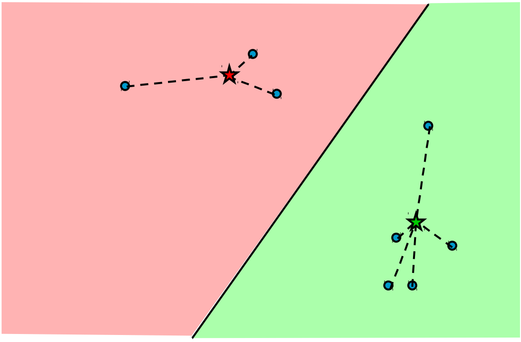


# k-means

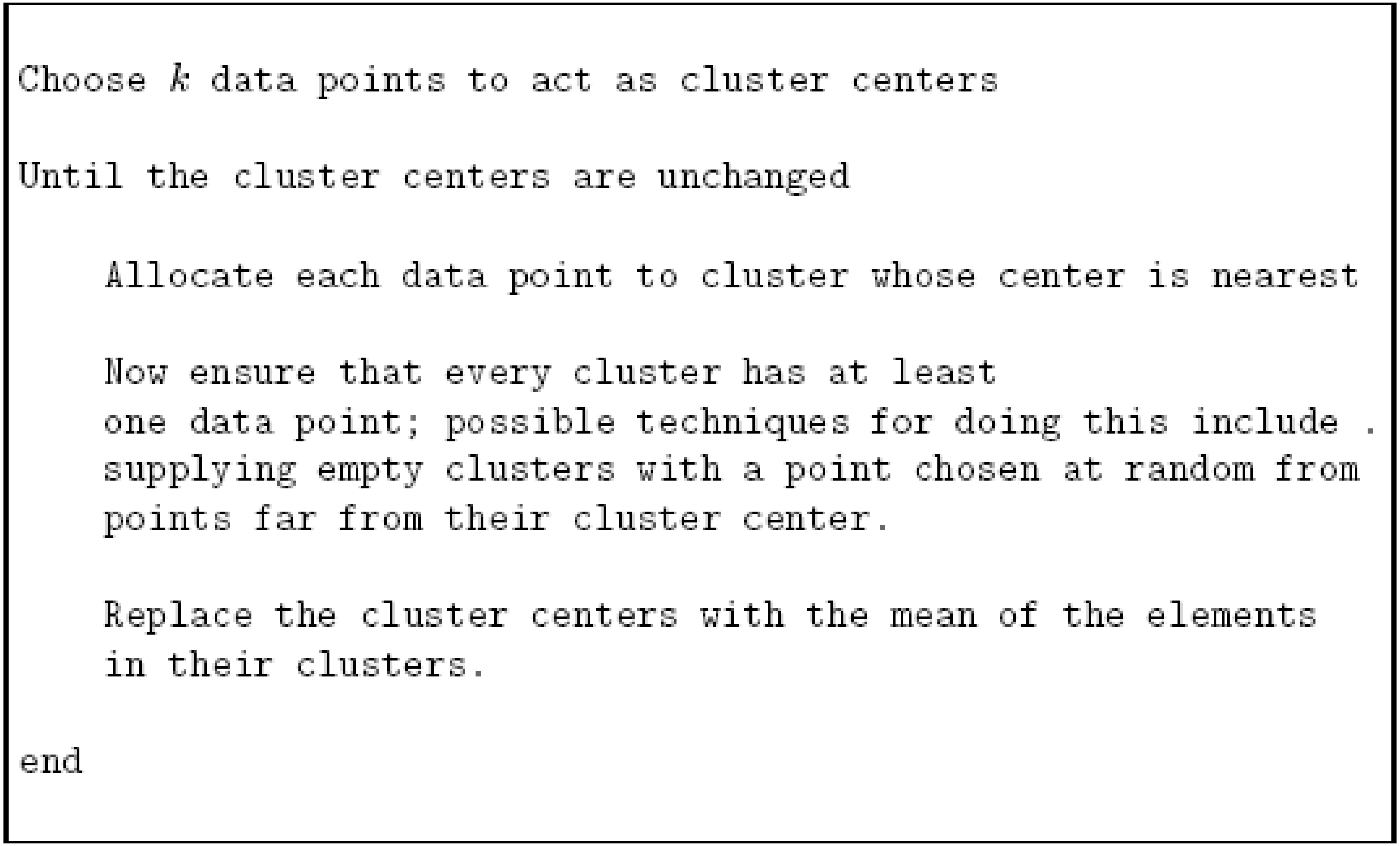
- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error

$$\sum_{k \in \text{clusters}} \sum_{i \in \text{element of k-th cluster}} ||x_i - \mu_j||^2$$

- Iterative algorithm to find local optimum
  - Assign tokens to closest cluster center
  - Update cluster center to mean associated tokens



# k-means



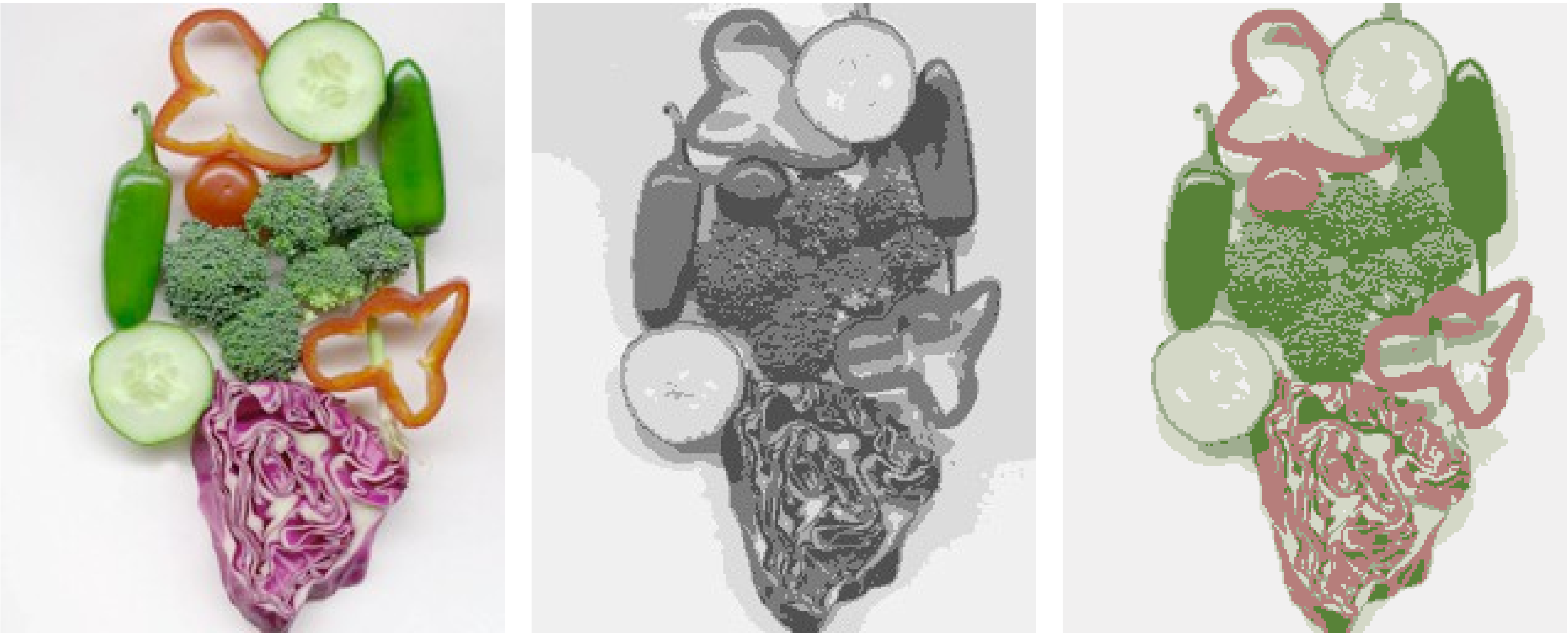
```
Choose $k$ data points to act as cluster centers

Until the cluster centers are unchanged

    Allocate each data point to cluster whose center is nearest

    Now ensure that every cluster has at least
    one data point; possible techniques for doing this include .
    supplying empty clusters with a point chosen at random from
    points far from their cluster center.

    Replace the cluster centers with the mean of the elements
    in their clusters.

end
```

**Algorithm 16.5:** *Clustering by K-Means*

# K-means for segmentation

- Select a value of K
- Select a feature vector for every pixel (color, texture, position, or combination of these etc.)
- Define a similarity measure between feature vectors (Usually Euclidean Distance).
- Apply K-Means Algorithm.
- Apply Connected Components Algorithm.
- Merge any components of size less than some threshold to an adjacent component that is most similar to it.

# k-means



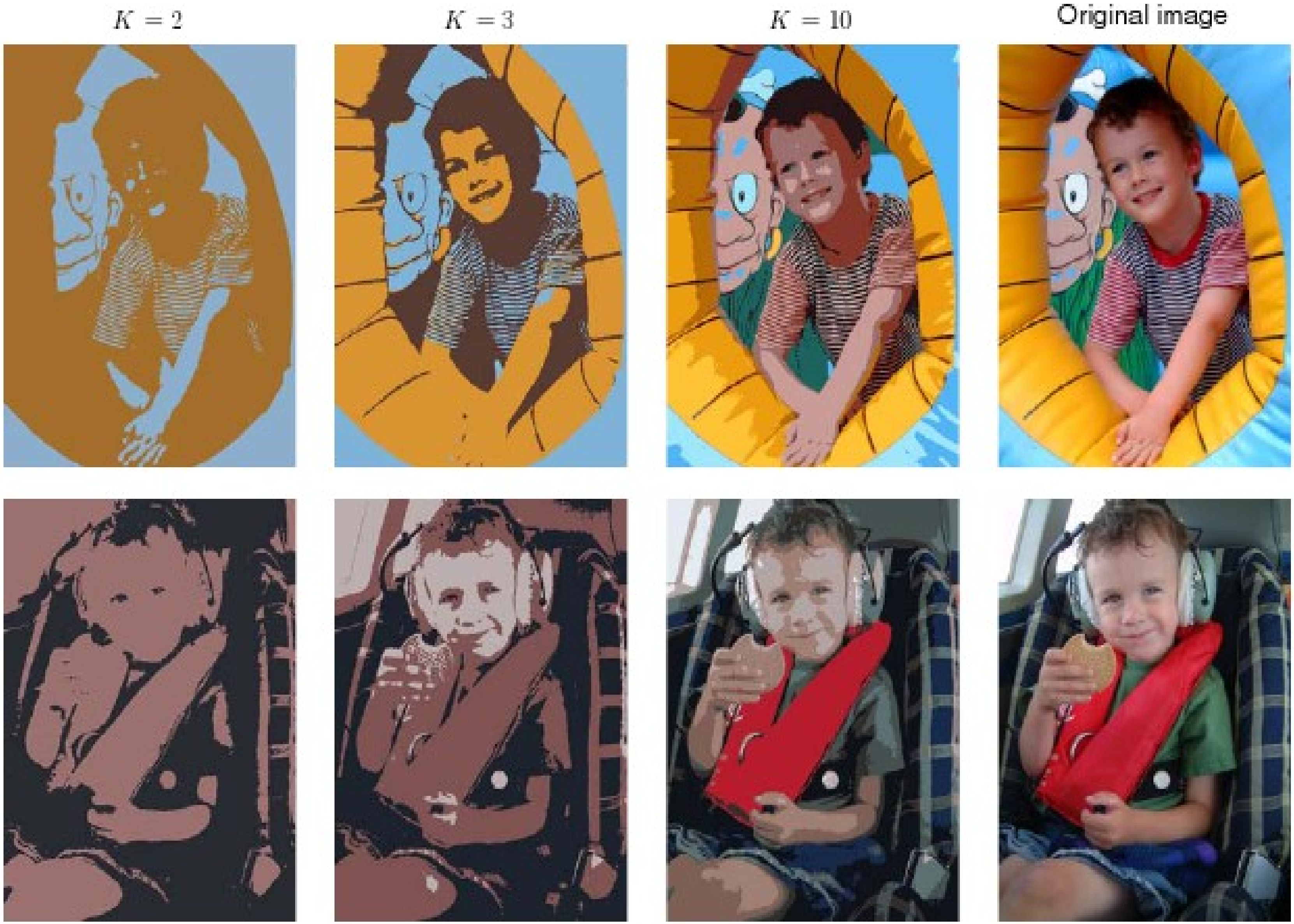
Image

Clusters on intensity

Clusters on color

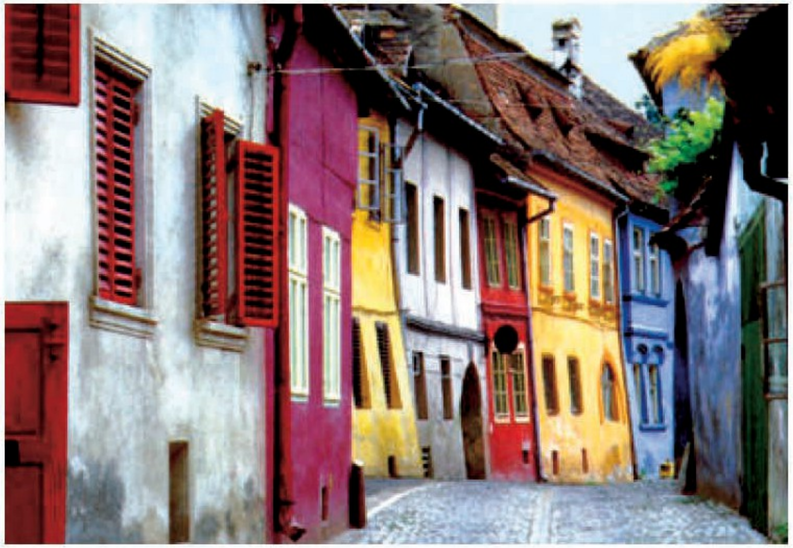
K-means clustering using intensity alone and color alone

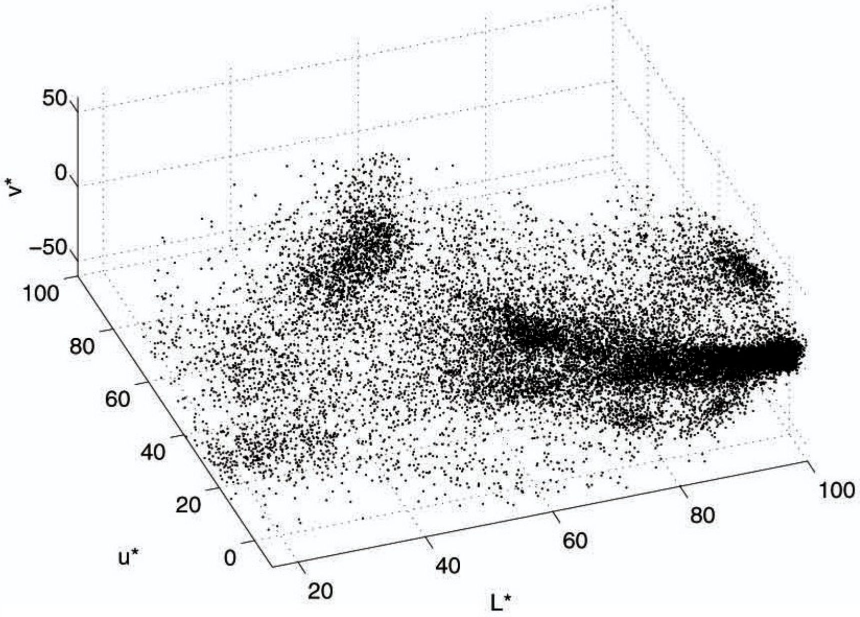
# k-means

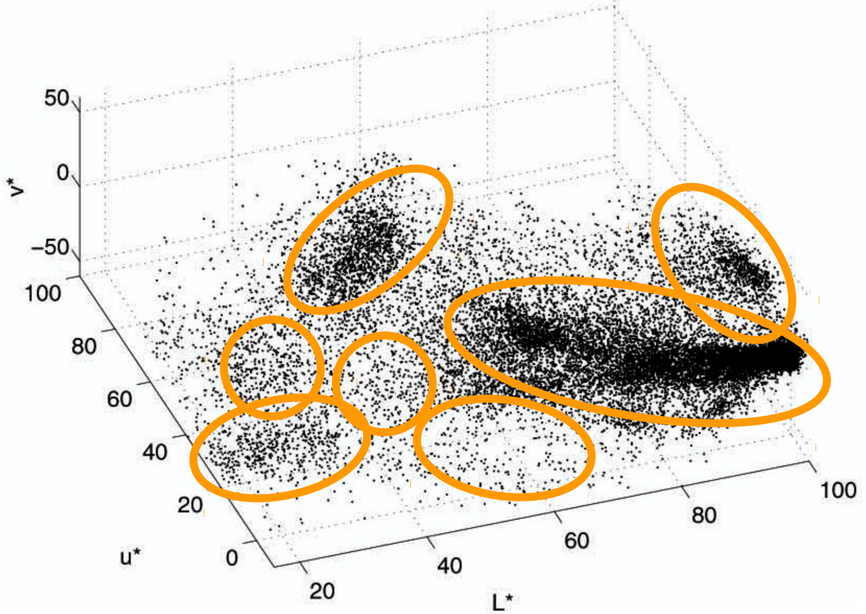


# Mean Shift

- A non-parametric technique for analyzing complex multimodal feature spaces and estimating the stationary points (modes) of th underlying probability density function ***without explicitly estimating it***




50
0
−50
v*
100
80
60
40
20
0
u*
20
40
60
80
100
L*



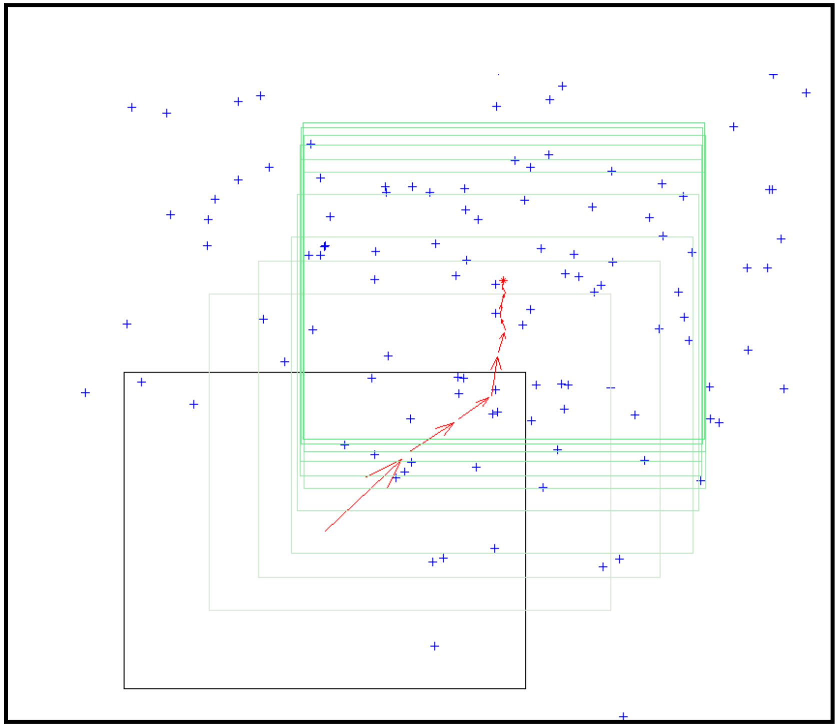
## Parametric Density Estimation?

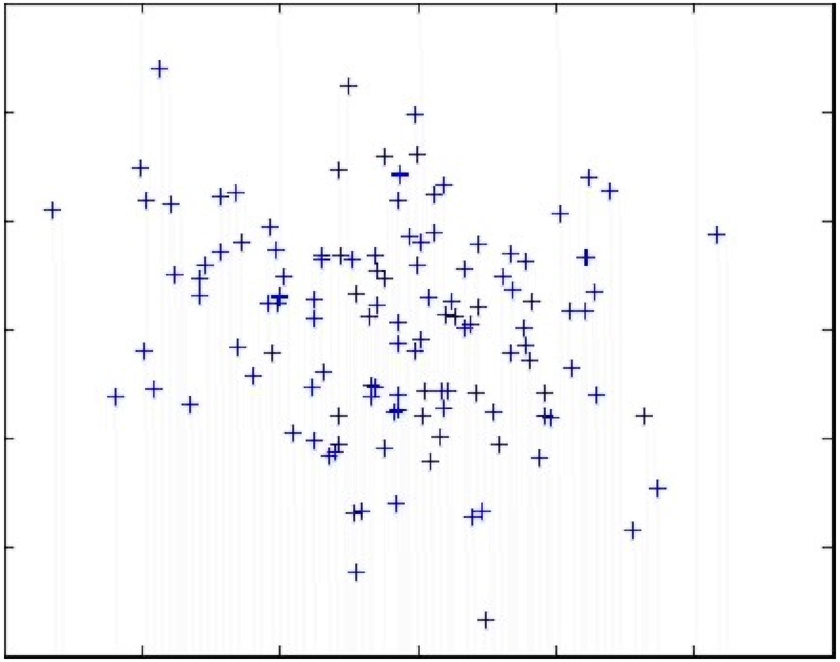
# Mean Shift Algorithm

**Mean Shift Algorithm**

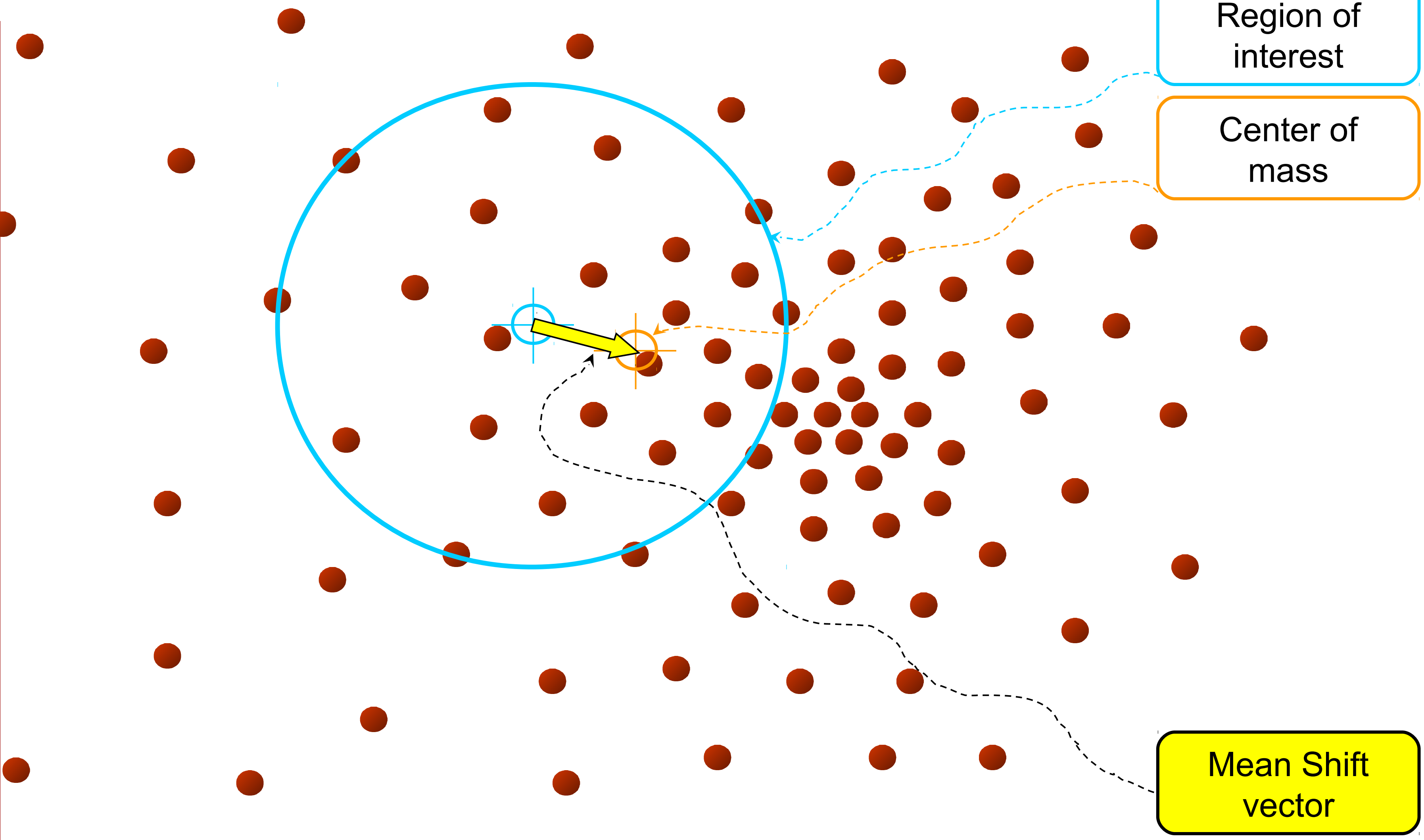
1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

The mean shift algorithm seeks the “mode” or point of highest density of a data distribution:



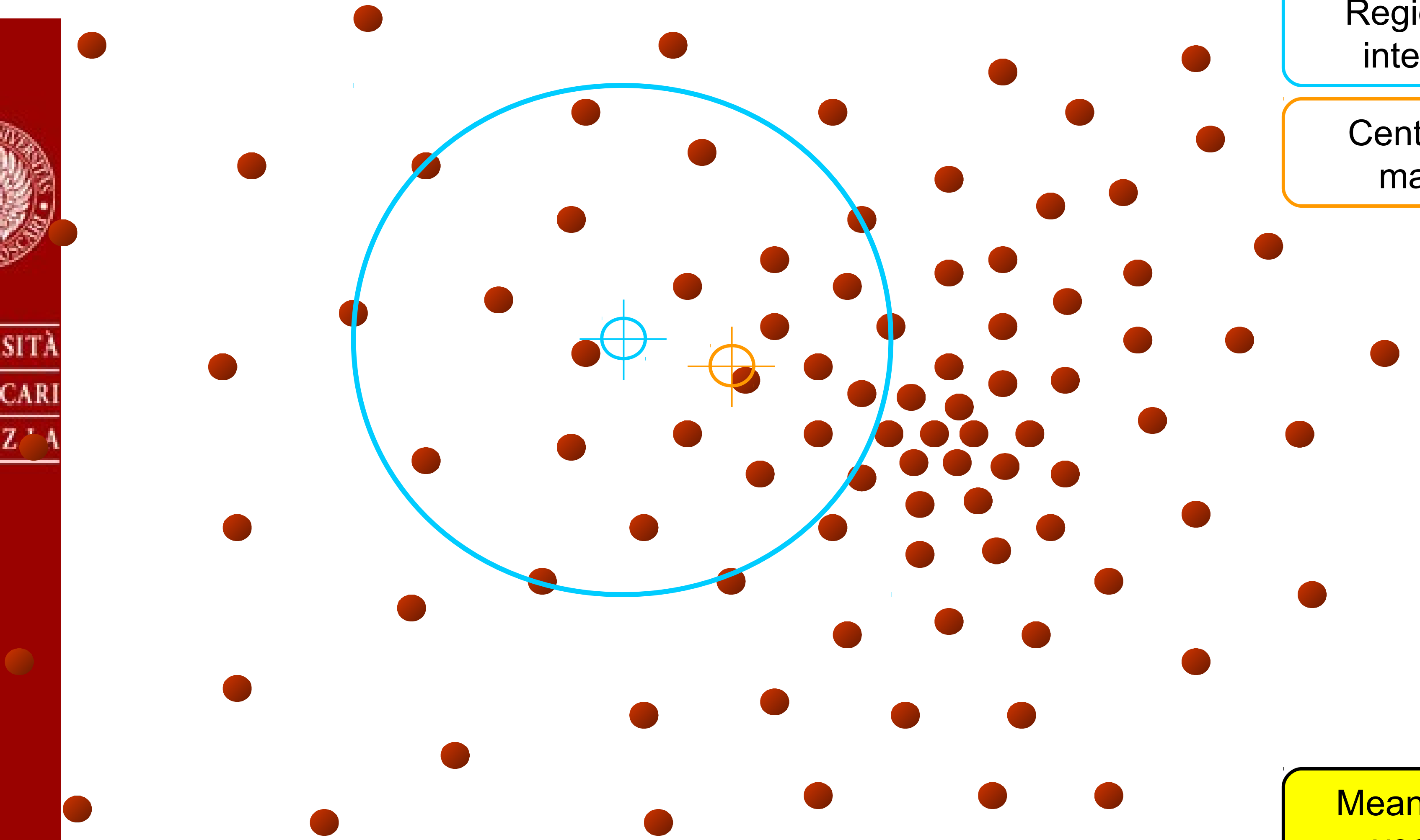


# Intuitive Description



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

# Intuitive Description



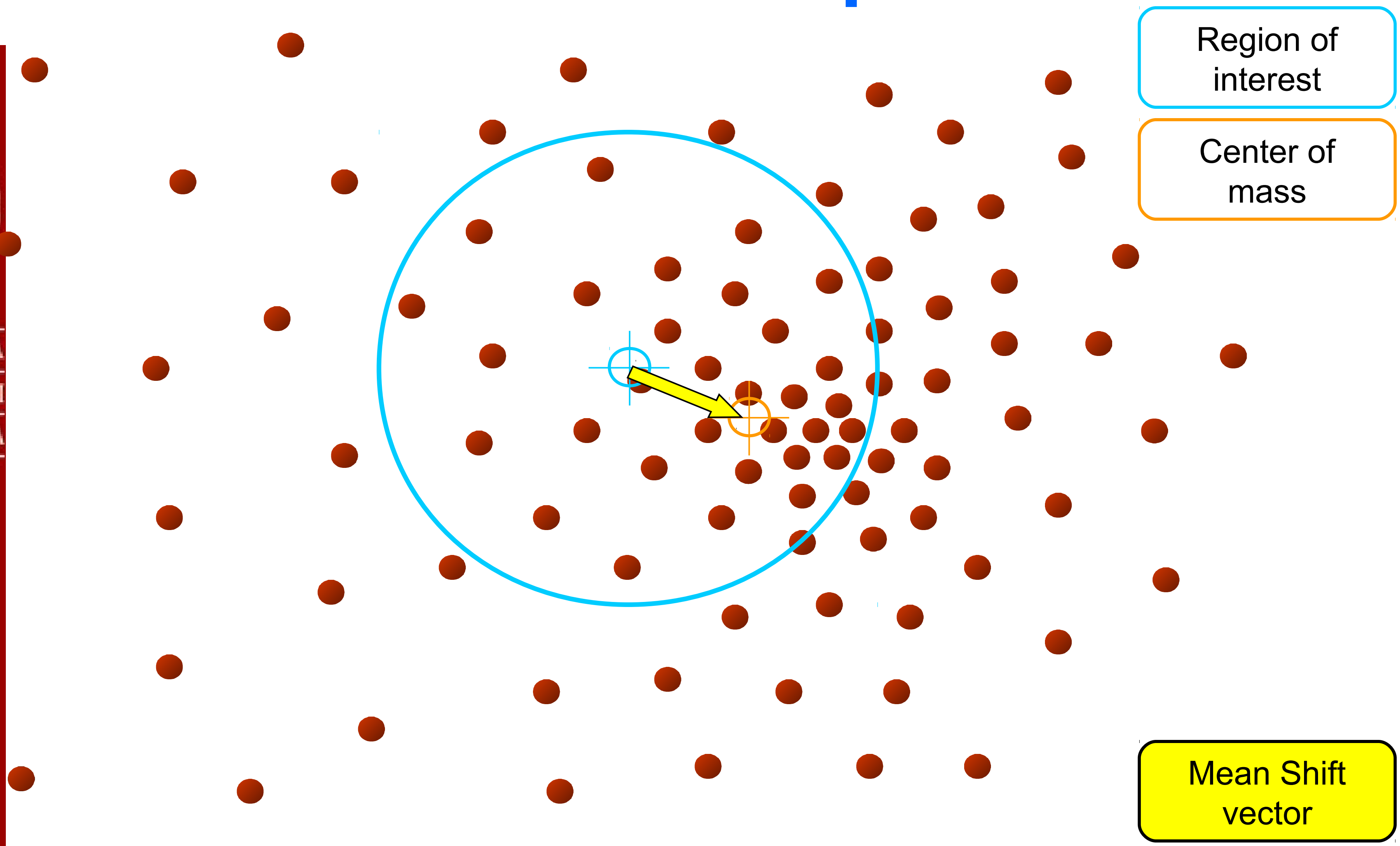
Region of interest

Center of mass

Mean Shift vector

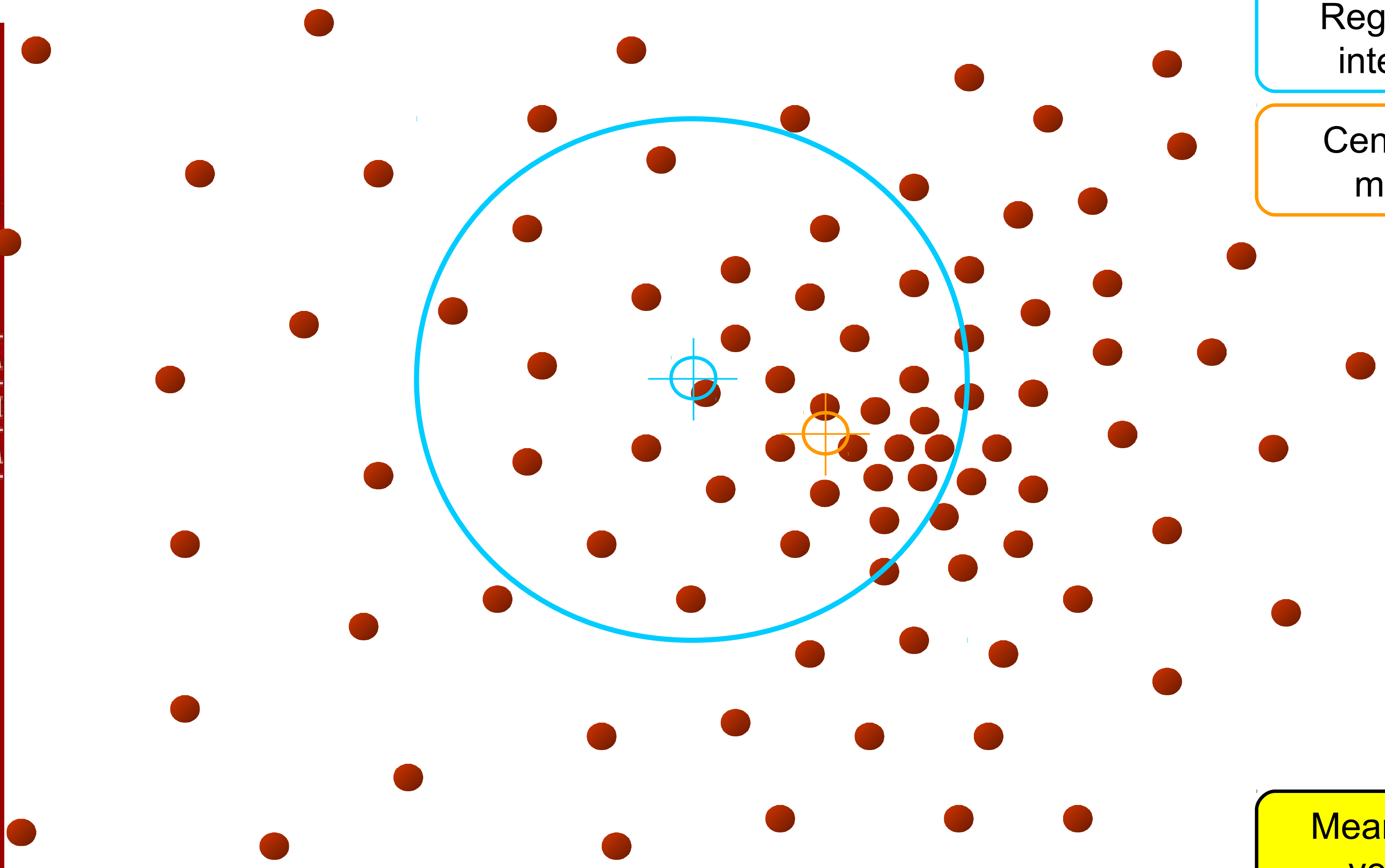
Objective : Find the densest region

# Intuitive Description



**Objective** : Find the densest region

# Intuitive Description



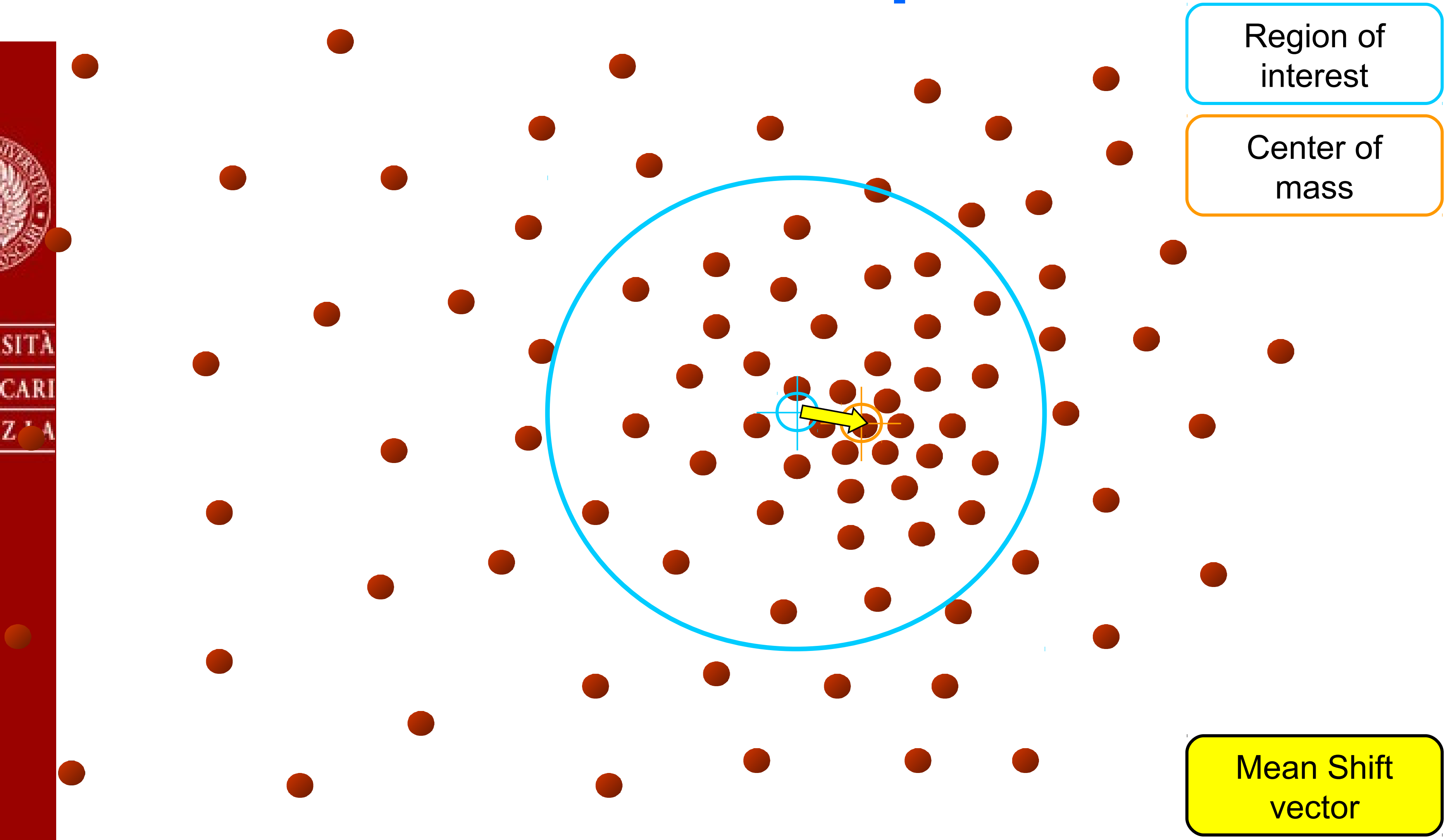
Region of interest

Center of mass

Mean Shift vector

**Objective : Find the densest region**

# Intuitive Description



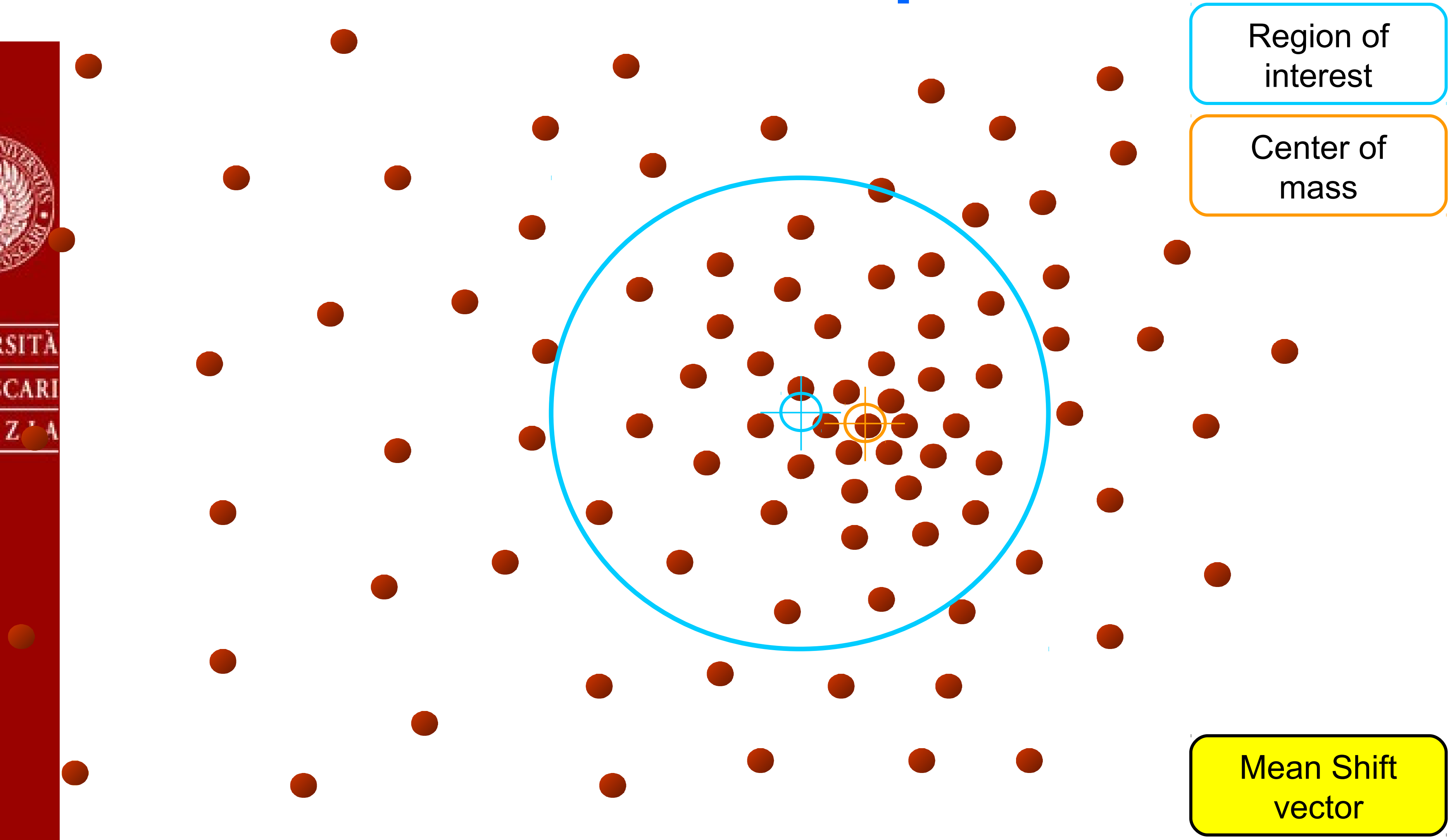
Region of interest

Center of mass

Mean Shift vector

Objective : Find the densest region

# Intuitive Description



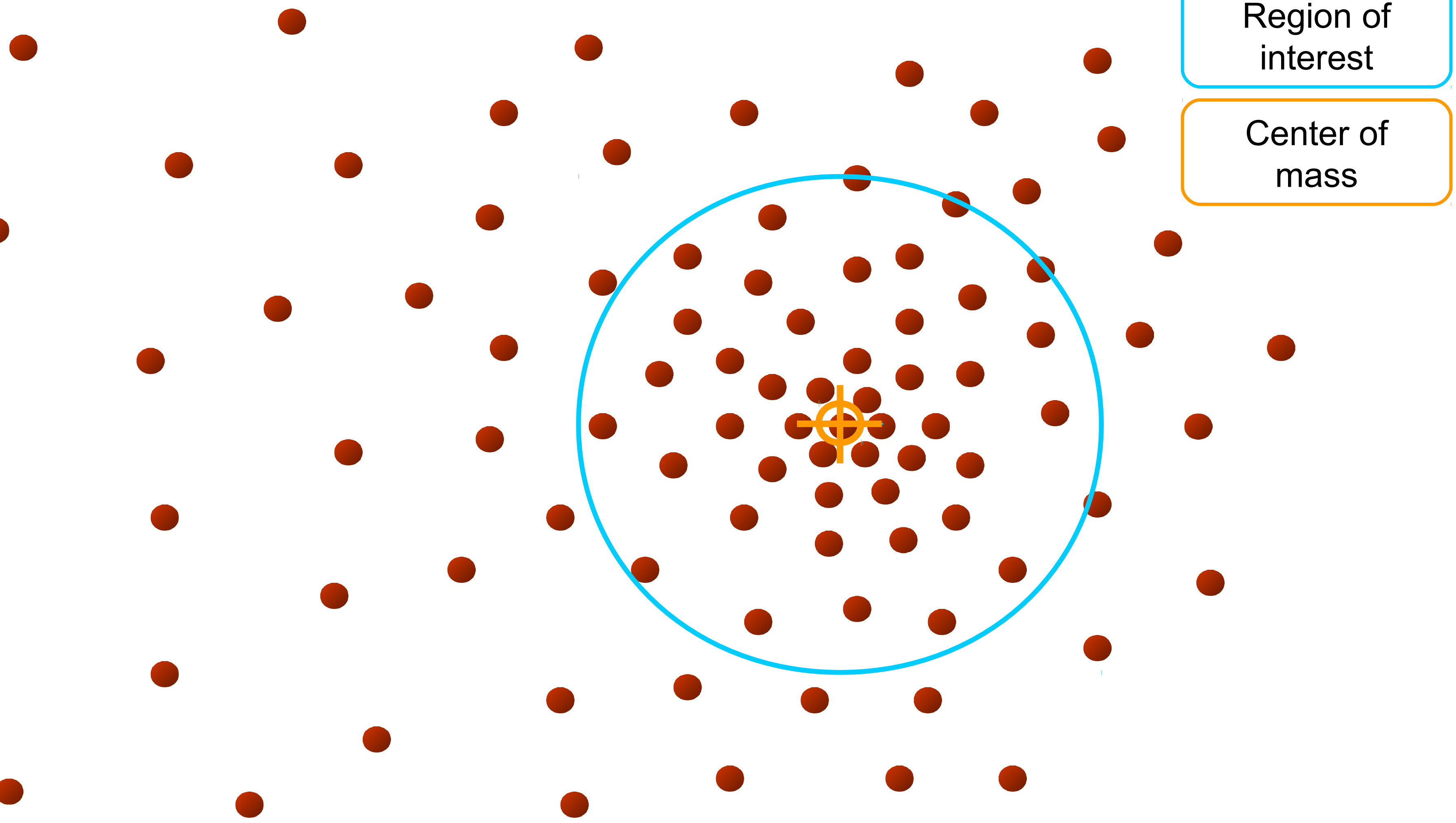
Region of interest

Center of mass

Mean Shift vector

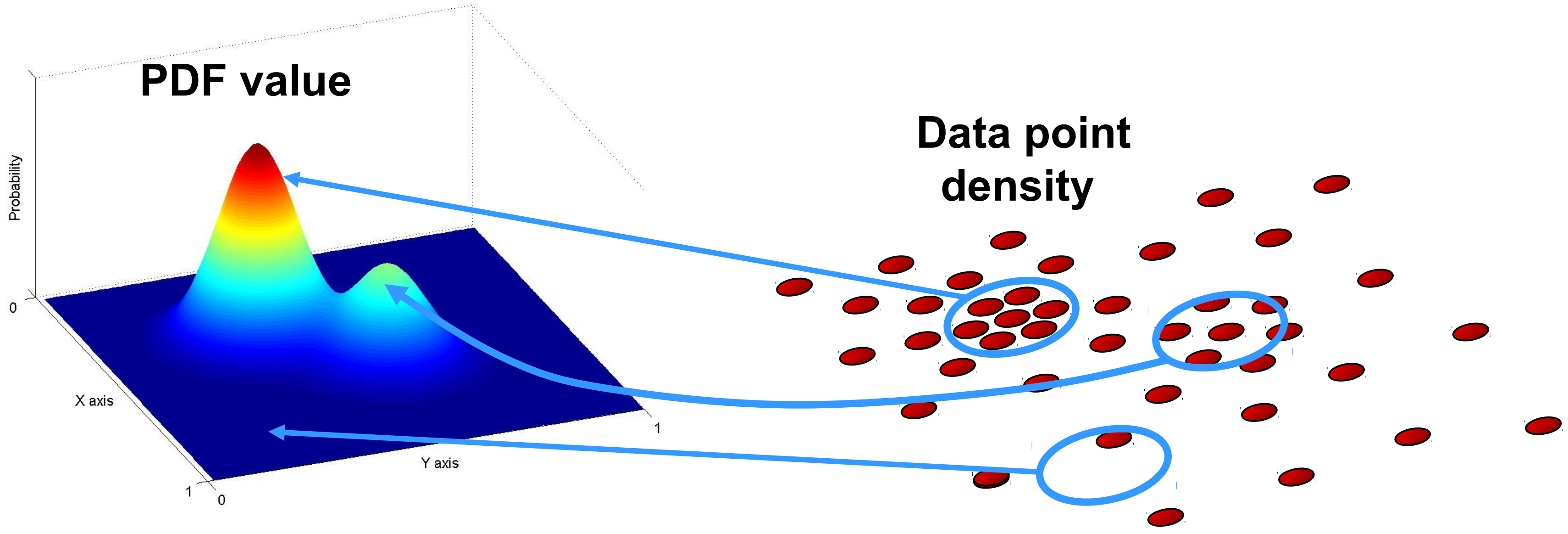
**Objective : Find the densest region**

# Intuitive Description



**Objective** : Find the densest region

# Non-parametric Density Estimation



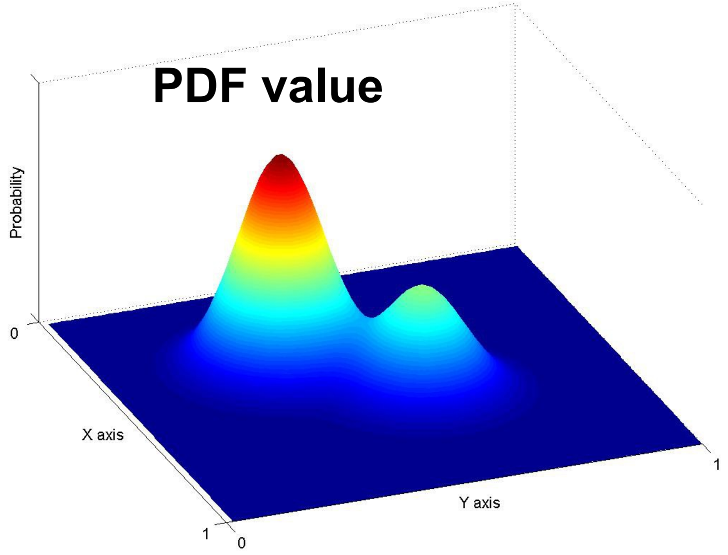
PDF value

Data point density

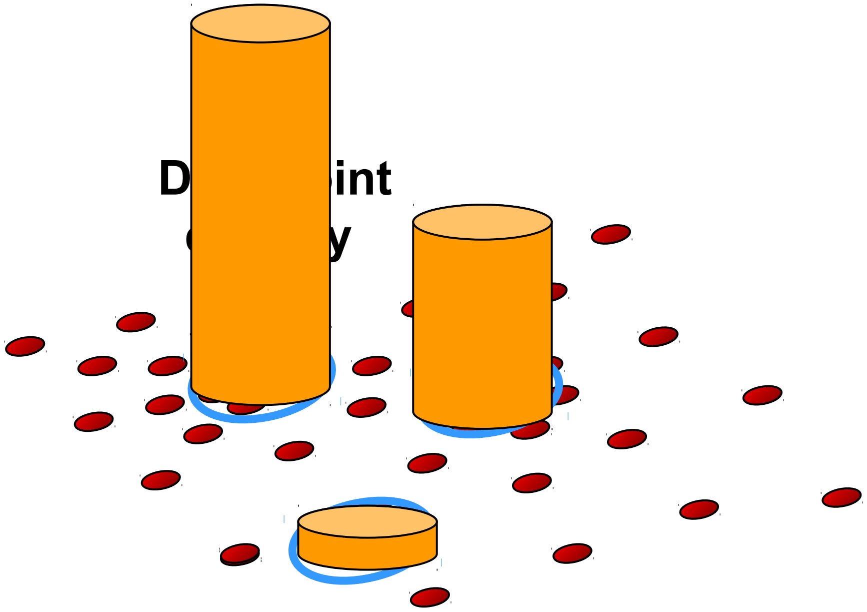
Assumed Underlying PDF

Data Samples

# Non-parametric Density Estimation

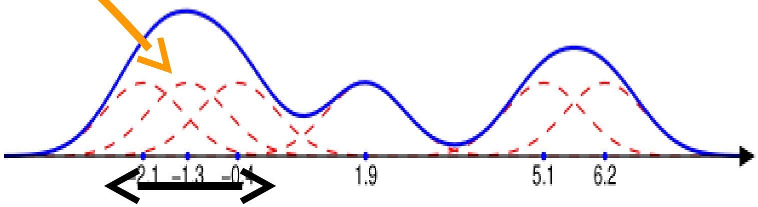


Assumed Underlying PDF



Data Samples

# Parzen Windows



$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(|\mathbf{H}|^{-1/2}\mathbf{x})$$

Kernel Properties

1. Bounded
2. Compact support
3. Normalized
4. Symmetric
5. Exponential decay
6. Uncorrelated

$$\int_{R^d} K(\mathbf{x}) d\mathbf{x} = 1$$

# Kernels and Bandwidths

- Kernel Types

$K^P(\mathbf{x}) = \prod_{i=1}^{d} K_1(x_i)$

(product of univariate kernels)

$K^S(\mathbf{x}) = a_{k,d} K_1(||\mathbf{x}||)$

(radially symmetric kernel)

$$K(\mathbf{x}) = c_{k,d}\, k(||\mathbf{x}||^2)$$

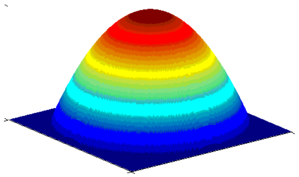
- Bandwidth Parameter $\mathbf{H} = h^2\mathbf{I}$

$$\hat{f}(\mathbf{x}) \quad = \quad \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

# Various Kernels

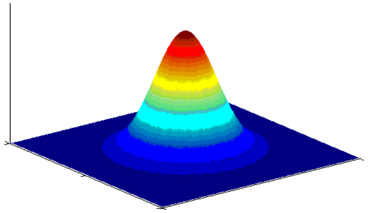
### Epanechnikov

$$\begin{aligned} k_E(x) &= \begin{cases} 1 - x & 0 \geq x \geq 1 \\ 0 & x > 1 \end{cases} \\ \Rightarrow K_E(\mathbf{x}) &= \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1 - \|\mathbf{x}\|^2) & \|\mathbf{x}\| \geq 1 \\ 0 & \|\mathbf{x}\| > 1 \end{cases} \end{aligned}$$



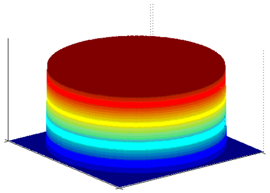
### Normal

$$\begin{aligned} k_N(x) &= e^{-\frac{1}{2}x} \\ \Rightarrow K_N(\mathbf{x}) &= (2\pi)^{-d/2} e^{-\frac{1}{2}\|\mathbf{x}\|^2} \end{aligned}$$



### Uniform

$$\begin{aligned} k_U(x) &= \begin{cases} 1 & 0 \geq x \geq 1 \\ 0 & x > 1 \end{cases} \\ \Rightarrow K_U(\mathbf{x}) &= \begin{cases} c_d & \|\mathbf{x}\| \leq 1 \\ 0 & \|\mathbf{x}\| > 1 \end{cases} \end{aligned}$$



# Density Gradient Estimation

Modes of the probability density

Epanechnikov → Uniform

Normal → Normal

# Mean Shift

$$\nabla \hat{f}_{h,K}(\mathbf{x})$$

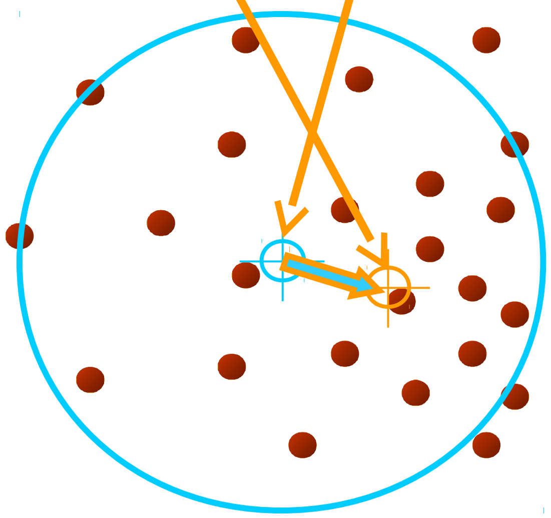
$$= \frac{2c_{k,d}}{nh^{d+2}} \underbrace{\left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right]}_{\text{KDE}} \underbrace{\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right]}_{\text{Mean Shift}}$$

## Mean Shift Algorithm

- compute mean shift vector

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

- translate kernel (window) by mean shift vector



# Mean Shift

$$\nabla \hat{f}_{h,K}(\mathbf{x}) = \hat{f}_{h,G}(\mathbf{x}) \frac{2c_{k,d}}{h^2 c_{g,d}} \mathbf{m}_{h,G}(\mathbf{x})$$

$$\Rightarrow \mathbf{m}_{h,G}(\mathbf{x}) = \frac{1}{2} h^2 c \frac{\nabla \hat{f}_{h,K}(\mathbf{x})}{\hat{f}_{h,G}(\mathbf{x})}$$

- Mean Shift is proportional to the *normalized* density gradient estimate obtained with kernel $K$

- The normalization is by the density estimate computed with kernel $G$

# Properties of Mean Shift

- Guaranteed convergence
  - Gradient Ascent algorithms are guaranteed to converge only for infinitesimal steps.
  - The normalization of the mean shift vector ensures that it converges.
  - Large magnitude in low-density regions, refined steps near local maxima → Adaptive Gradient Ascent.

- Mode Detection
  - Let $\{\mathbf{y}_j\}_{j=1,2,\ldots}$ denote the sequence of kernel locations.
  - At convergence $\mathbf{m}_{h,G}(\mathbf{y}_c) = \mathbf{y}_c - \mathbf{y}_c = 0 \Rightarrow \nabla \hat{f}_{h,K}(\mathbf{y}_c) = 0$
  - Once $\mathbf{y}_j$ gets sufficiently close to a mode of $\hat{f}_{h,K}$ it will converge to the mode.
  - The set of all locations that converge to the same mode define the *basin of attraction* of that mode.

# Properties of Mean Shift

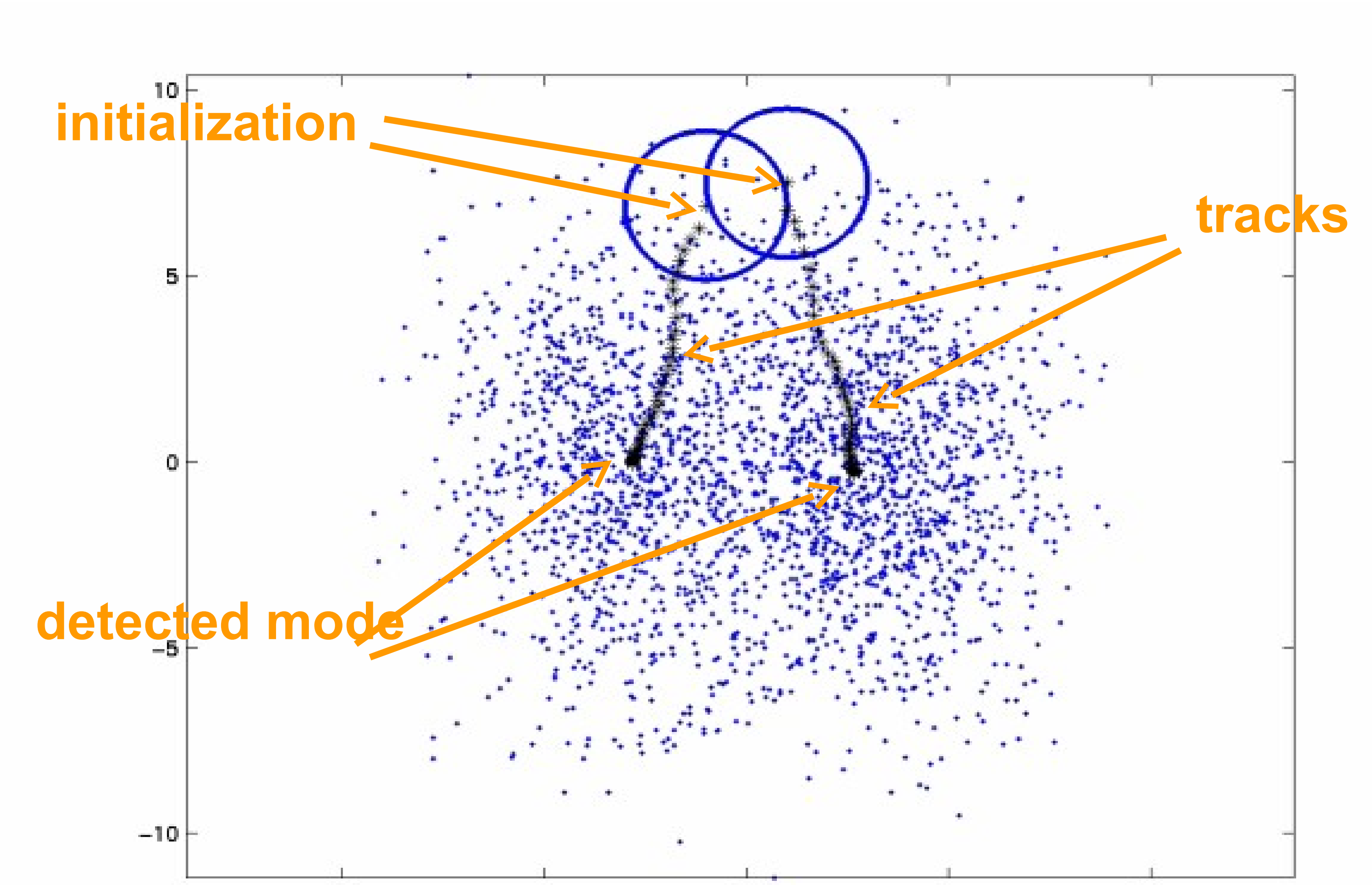
- Smooth Trajectory
  - The angle between two consecutive mean shift vectors computed using the normal kernel is always less that 90°
  - In practice the convergence of mean shift using the normal kernel is very slow and typically the uniform kernel is used

# Mode detection using Mean Shift

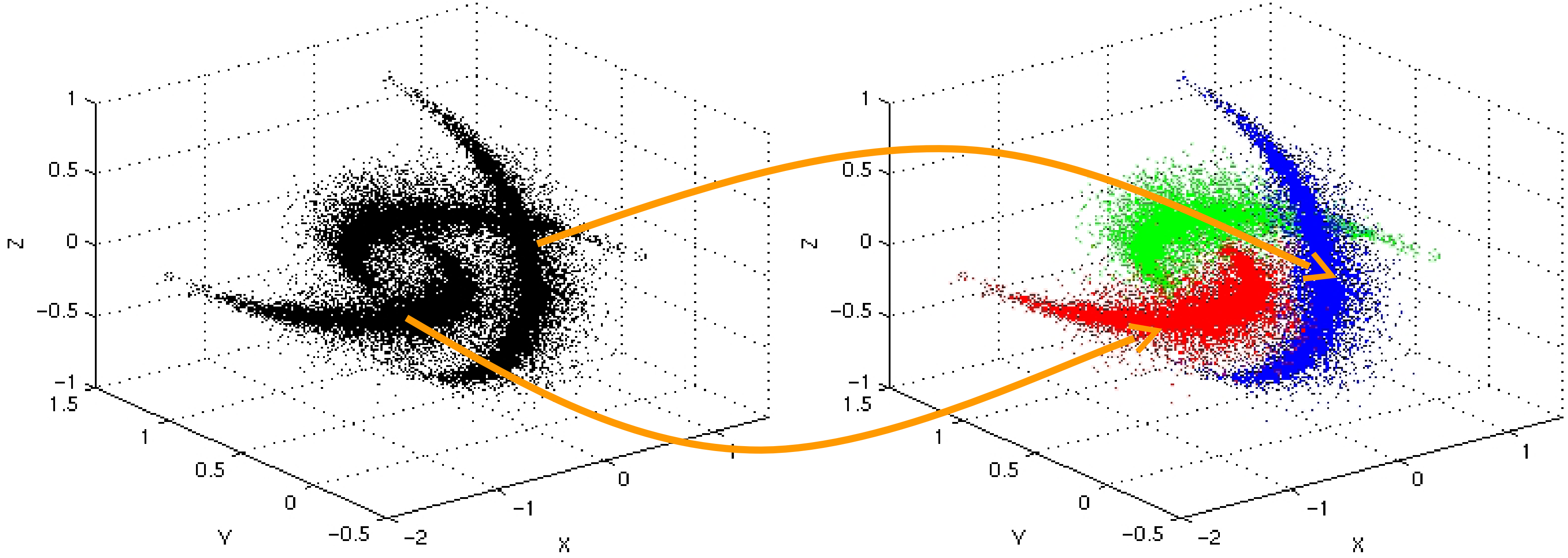
UNIVERSITÀ CA' FOSCARI VENEZIA

- Run Mean Shift to find the stationary points
  - To detect multiple modes, run in parallel starting with initializations covering the entire feature space.
- Prune the stationary points by retaining local maxi
  - Merge modes at a distance of less than the bandwidth.
- Clustering from the modes
  - The basin of attraction of each mode delineates a cluster o arbitrary shape.

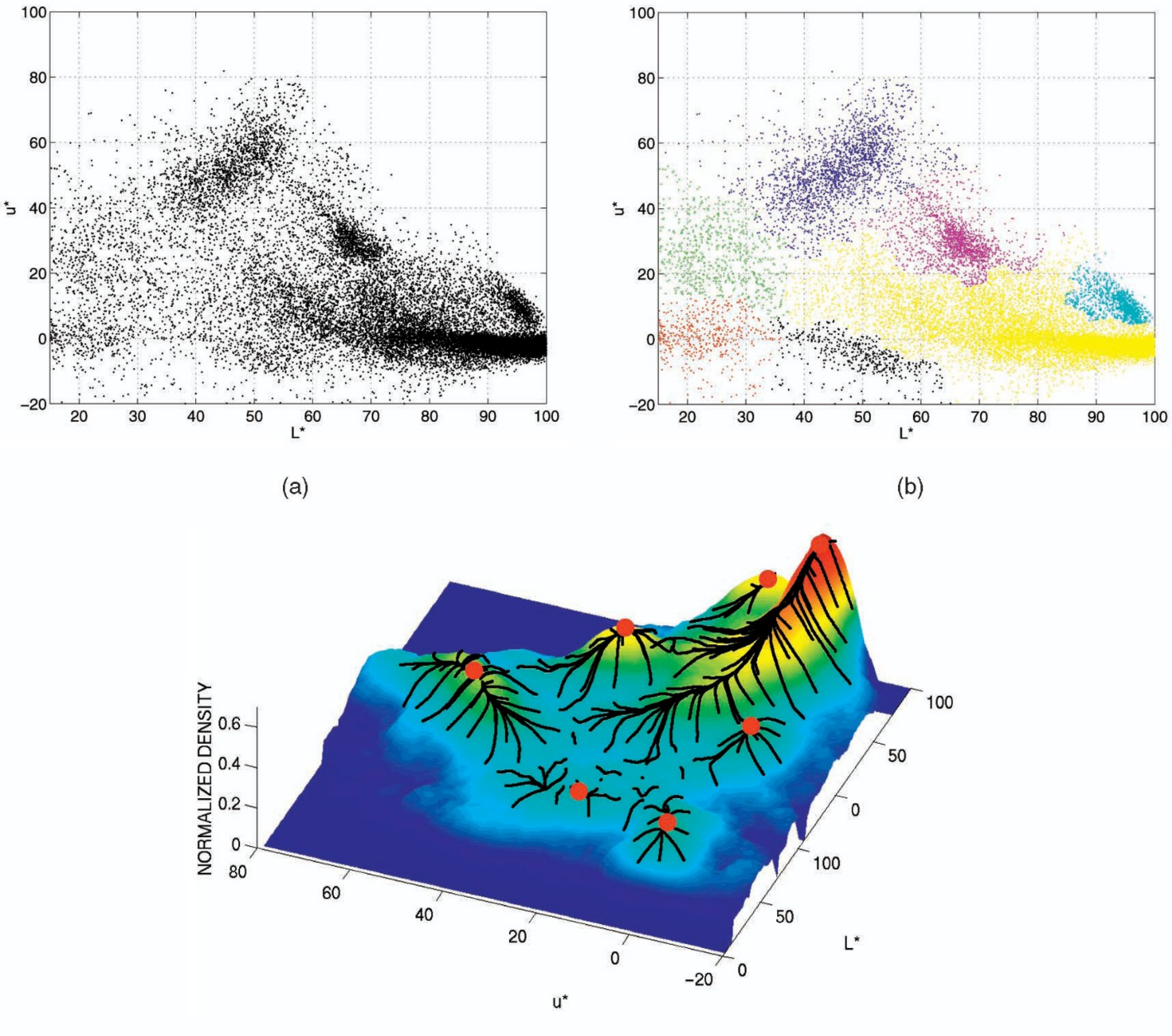
# Mode Finding on Real Data



# Mean Shift Clustering



# Clustering on Real Data

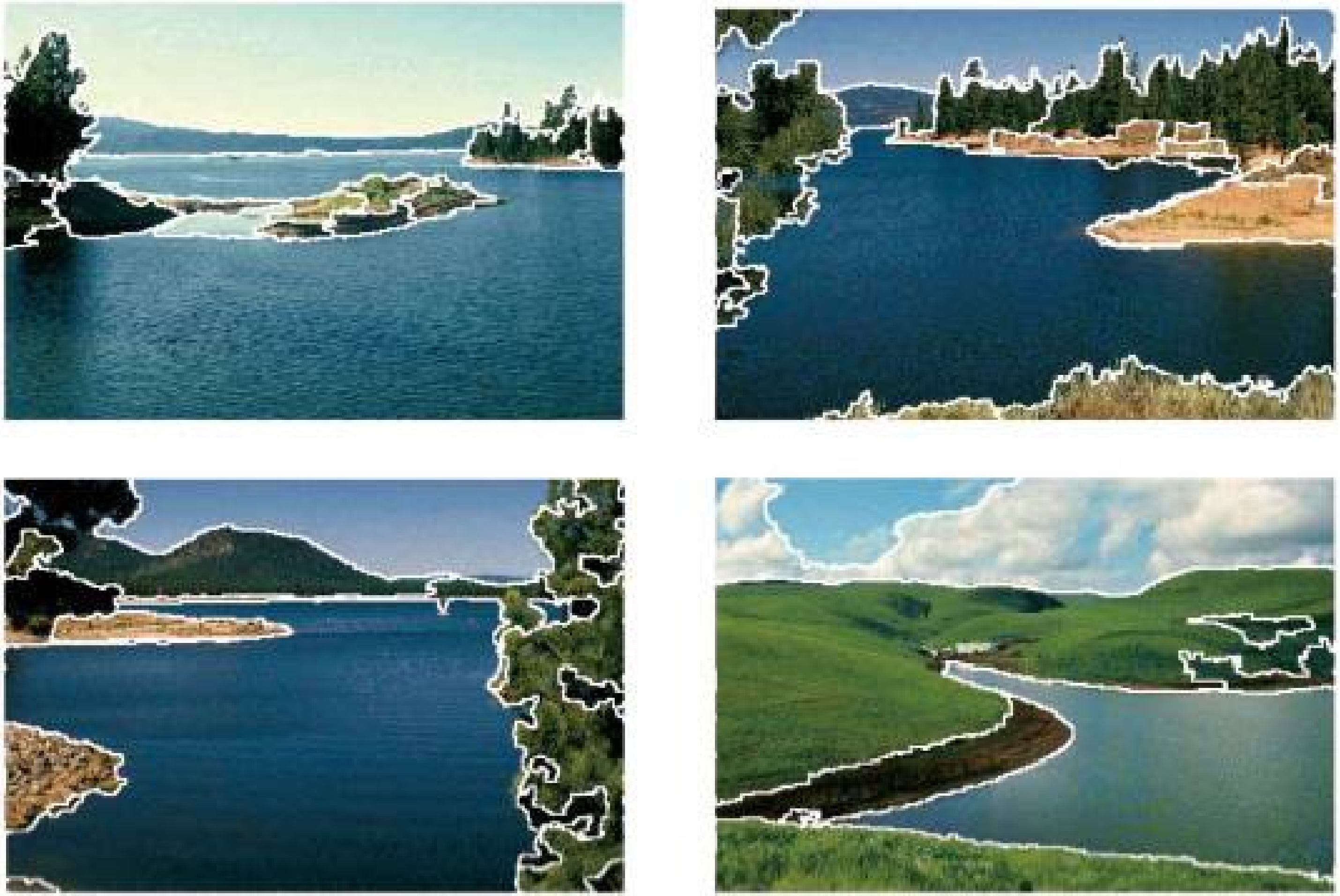


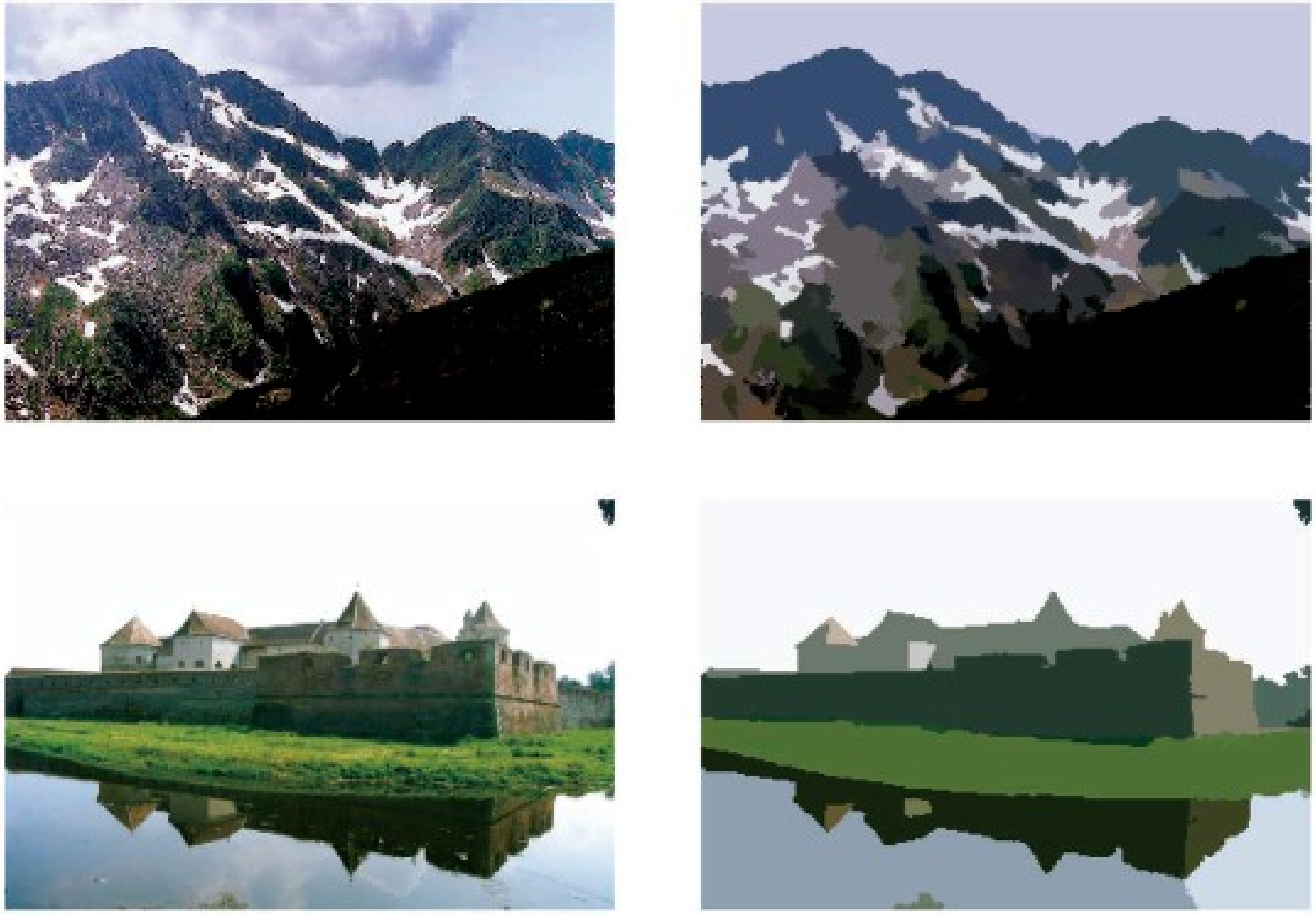
(a)

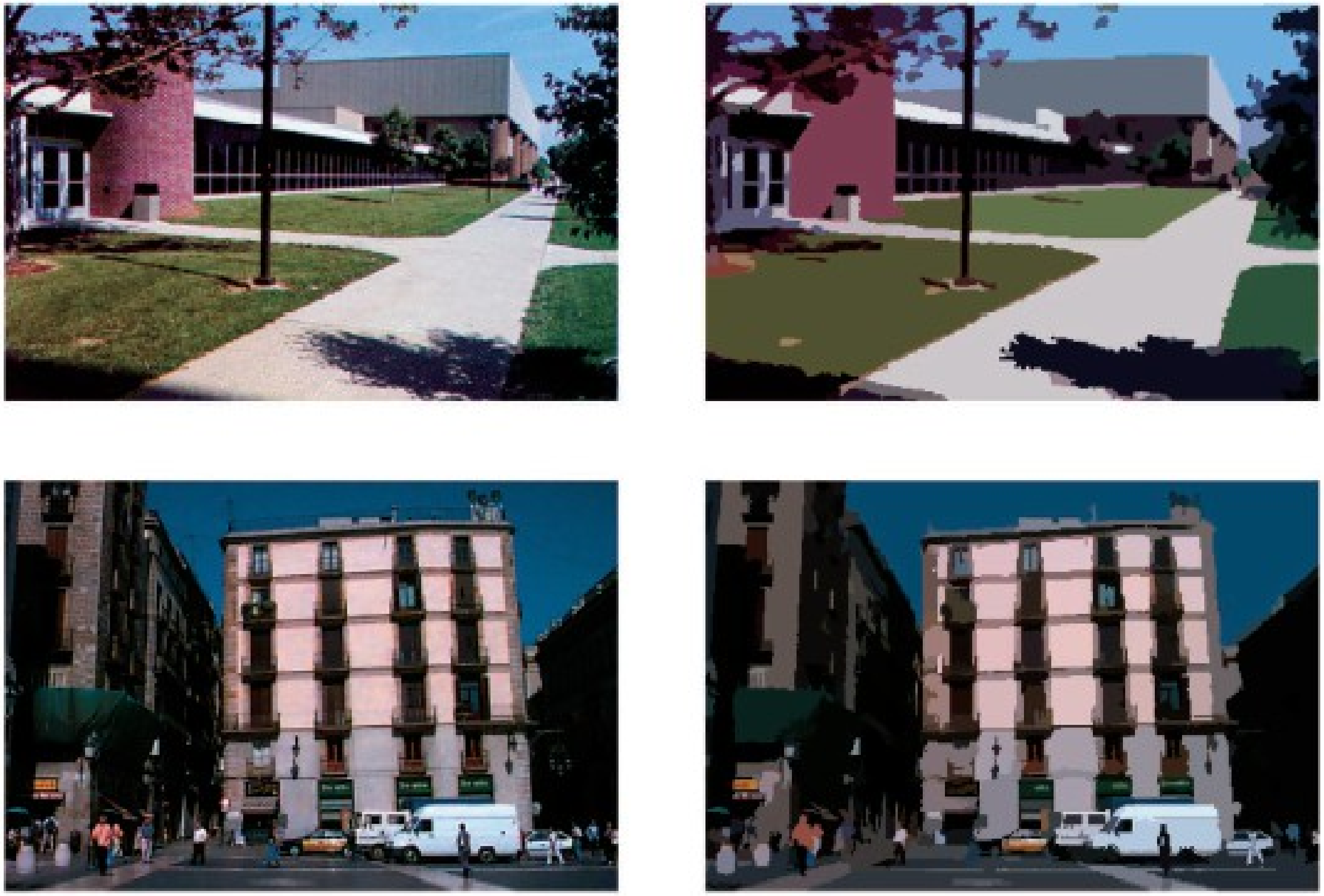
(b)

(c)

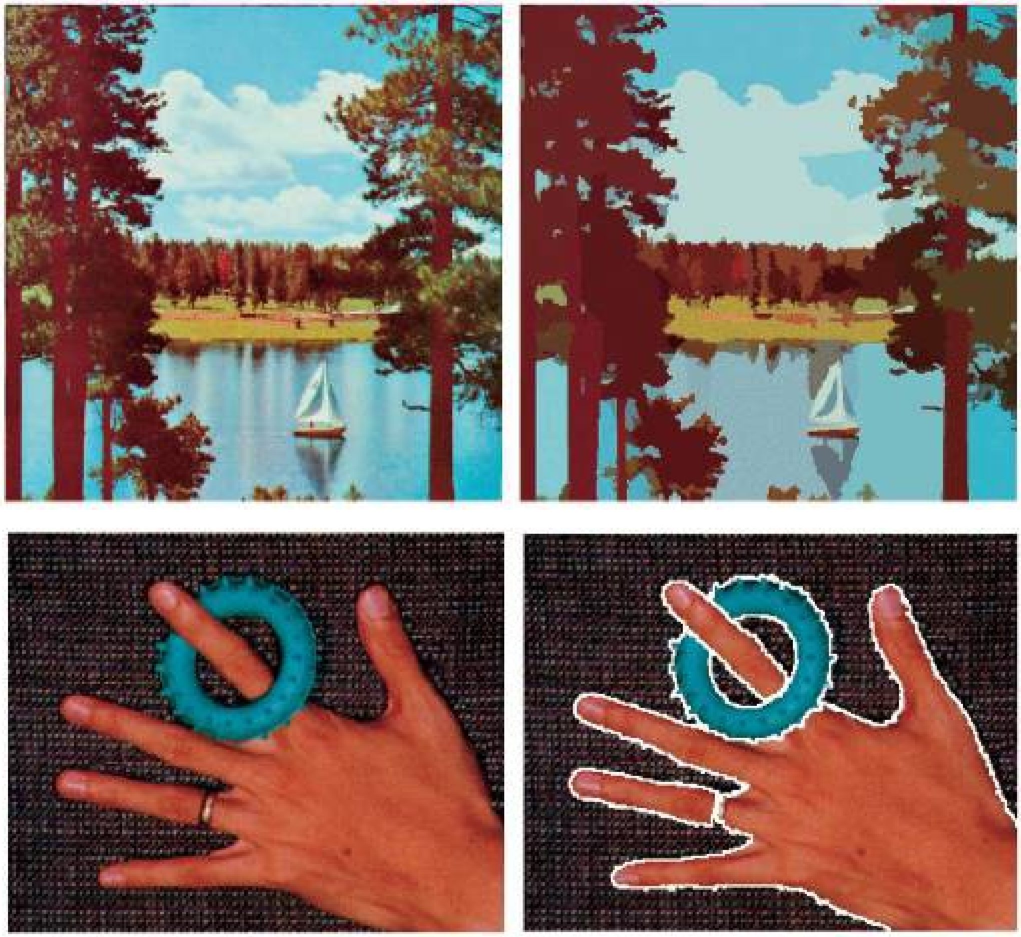
# Mean Shift Segmentation

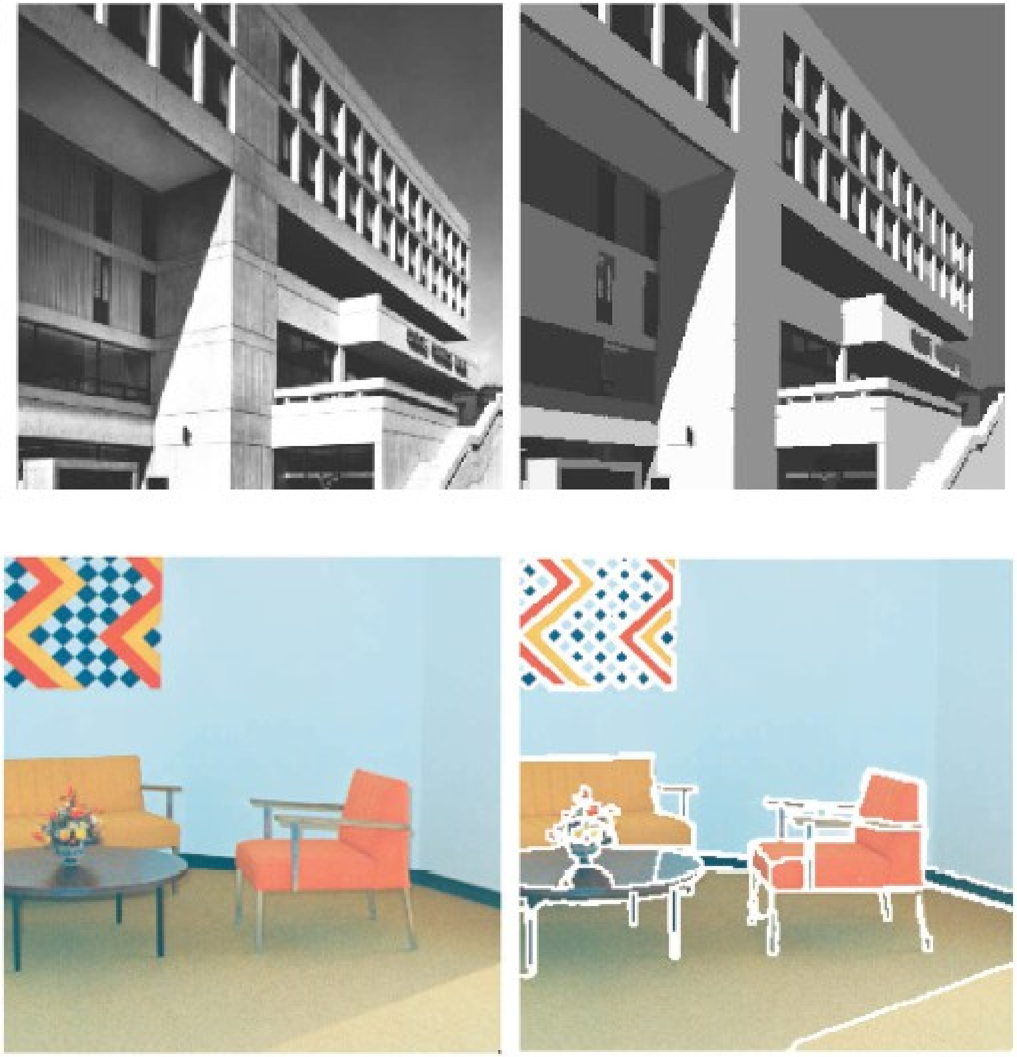






UNIVERSITÀ
CA' FOSCARI
VENEZIA





# Notes on implementation

- Tracing the tracks for each point can be too slow for image segmentation.
- There are two common heuristics used to speedup the algorithm:

1) **Basin of attraction:** Upon finding a peak, associate each data point that is at a distance r from the peak with the cluster defined by that peak.

2) Points that are within a distance of r/c of the search path are associated with the converged peak, where c is some constant value. c = 4 is a common value of image segmetnation.