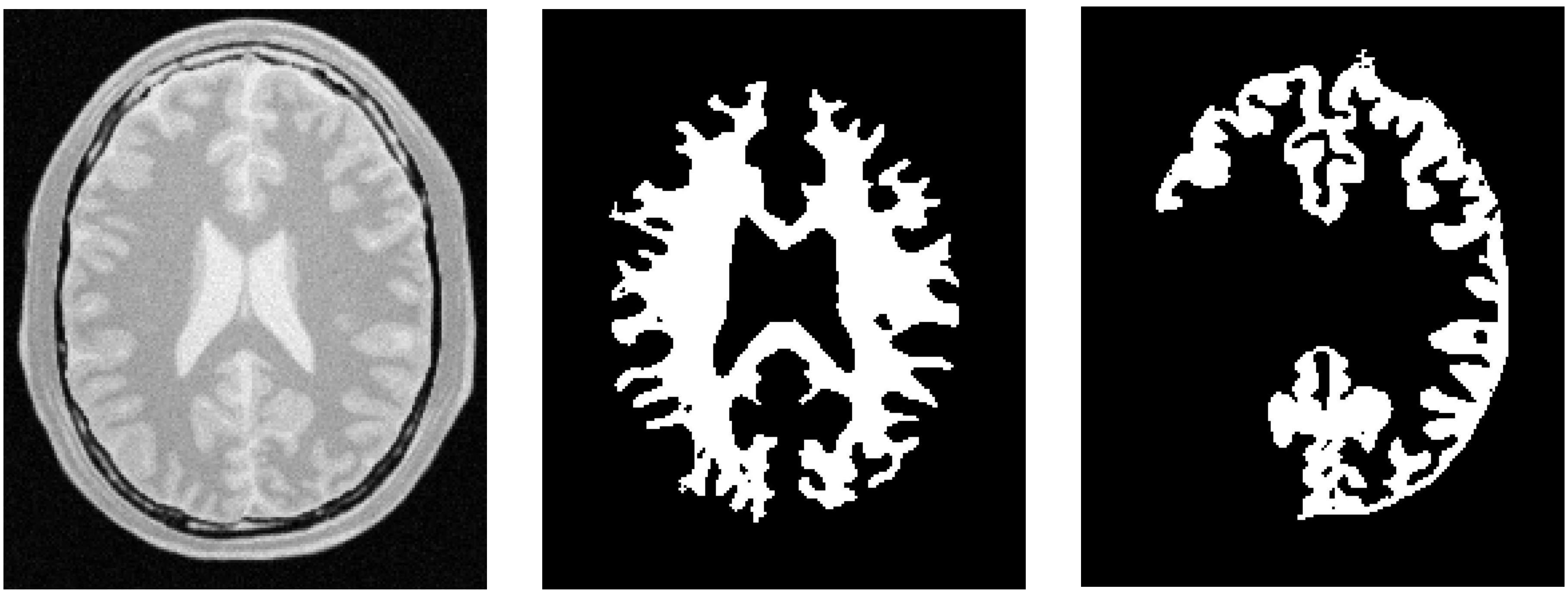
National University of Sciences and Technology (NUST)

SEECS

Digital Image Processing

Image Segmentation

Segmentation



Segmentation

- Group similar components (such as, pixels in an image, image frames in a video)

- Applications: Finding tumors, veins, etc. in medical images, finding targets in satellite/aerial images, finding people in surveillance images, summarizing video, etc.

Segmentation

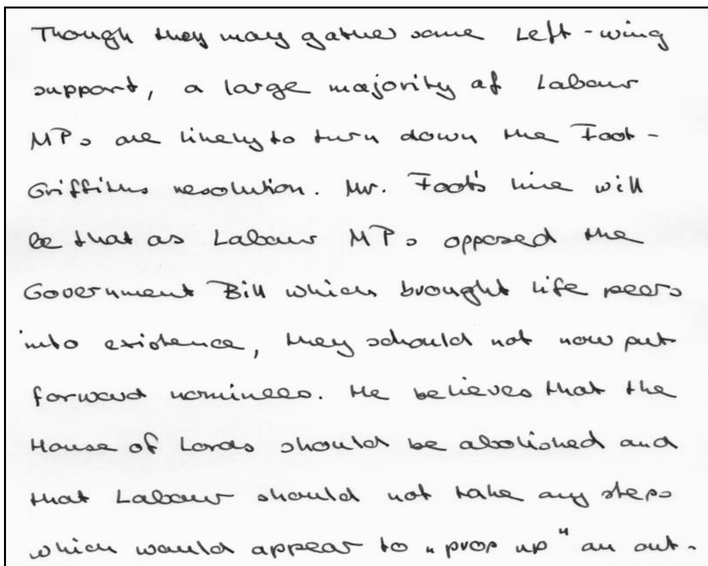
- Segmentation algorithms are based on one of two basic properties of gray-scale values:
  - Discontinuity
    - Partition an image based on abrupt changes in gray-scale levels.
    - Detection of isolated points, lines, and edges in an image.
  - Similarity
    - Thresholding, region growing, and region splitting/merging.

Thresholding

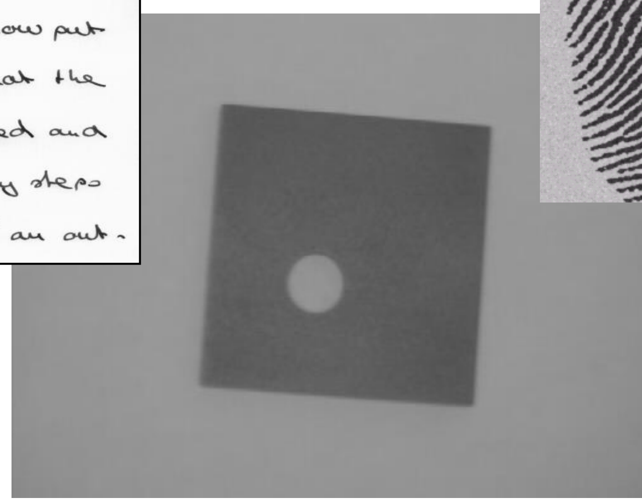
Thresholding

- Segmentation into two classes/groups
  - Foreground (Objects)
  - Background

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



Though they may gather some Left-wing support, a large majority of Labour MPs are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour MPs opposed the Government Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-



Thresholding

- GLOBAL
- LOCAL

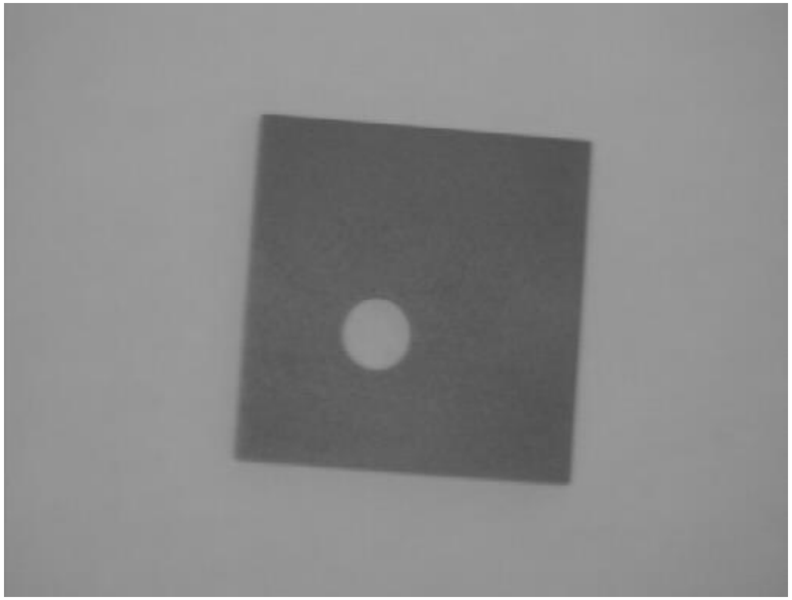
$$T = mean$$

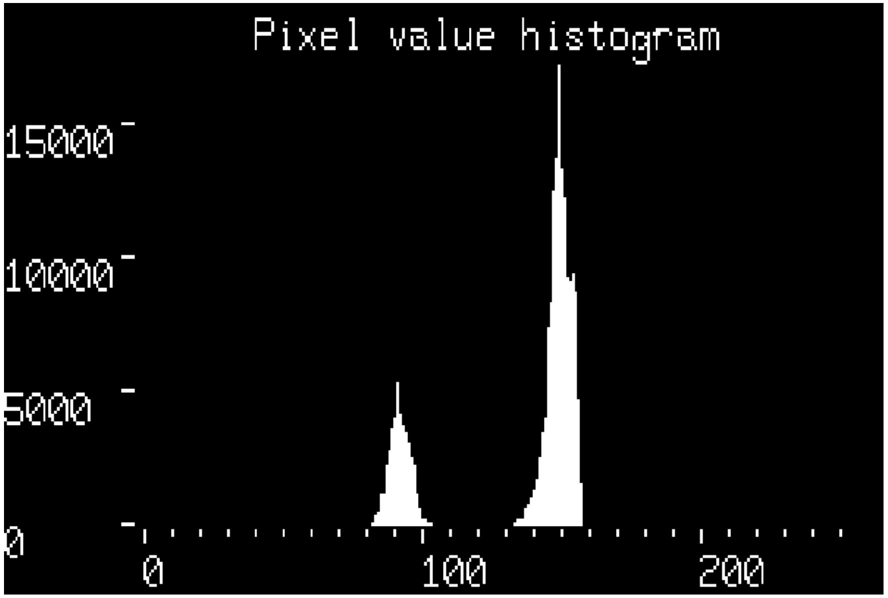
$$T = median$$

$$T = \frac{\max + \min}{2}$$

Thresholding - Global

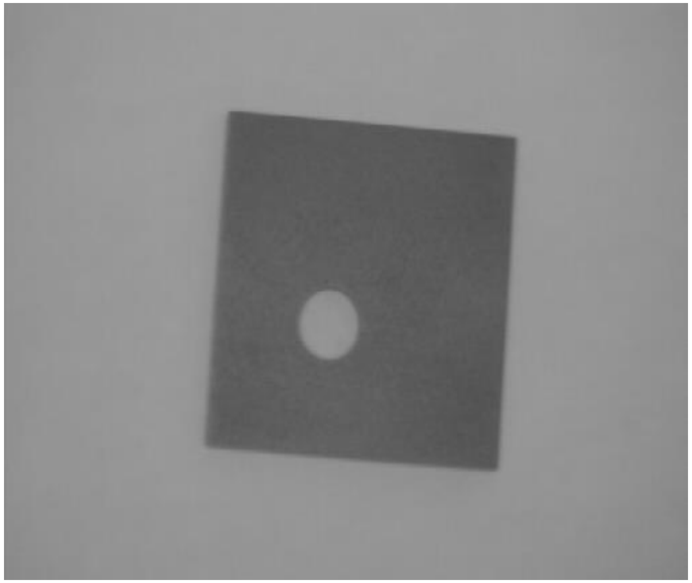
- Single threshold value for entire image
- Fixed ?
- Automatic
  - Intensity histogram

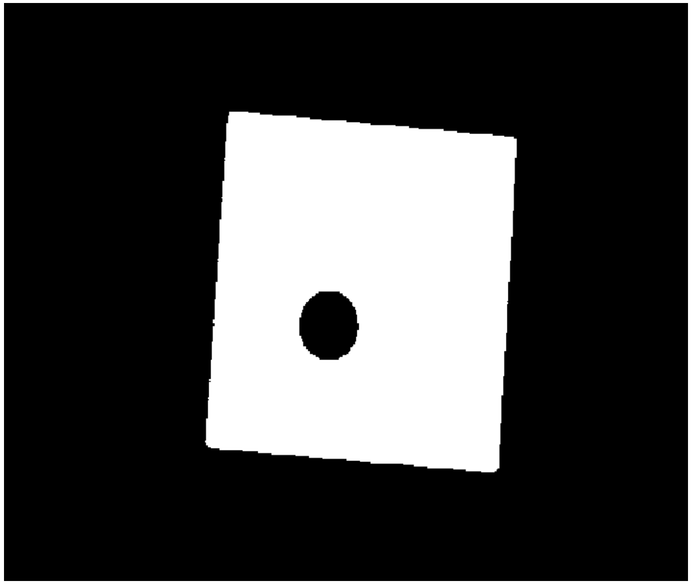




Thresholding - Global

- Single threshold value for entire image
- Fixed ?
- Automatic
  - Intensity histogram

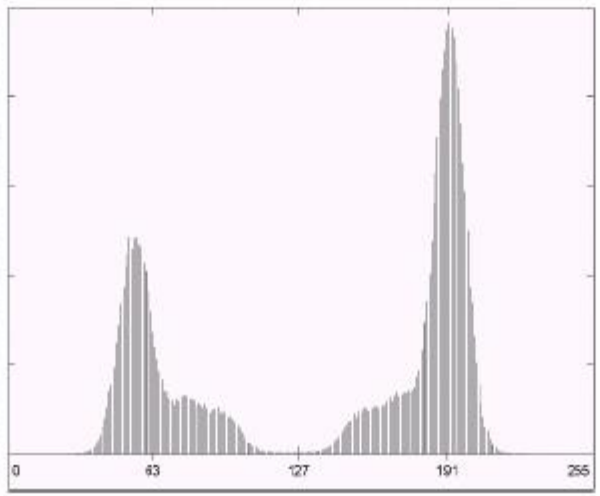


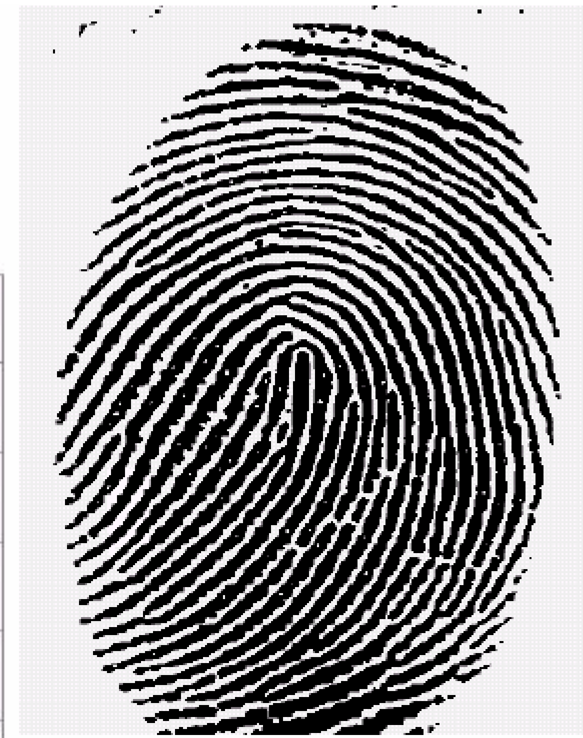


Thresholding - Global

- Estimate an initial T
- Segment Image using T: Two groups of pixels G1 and G2
- Compute average gray values m1 and m2 of two groups
- Compute new threshold value T=(m1+m2)/2
- Repeat steps 2 to 4 until: abs(T_i – T_{i-1})<epsilon

Thresholding - Global

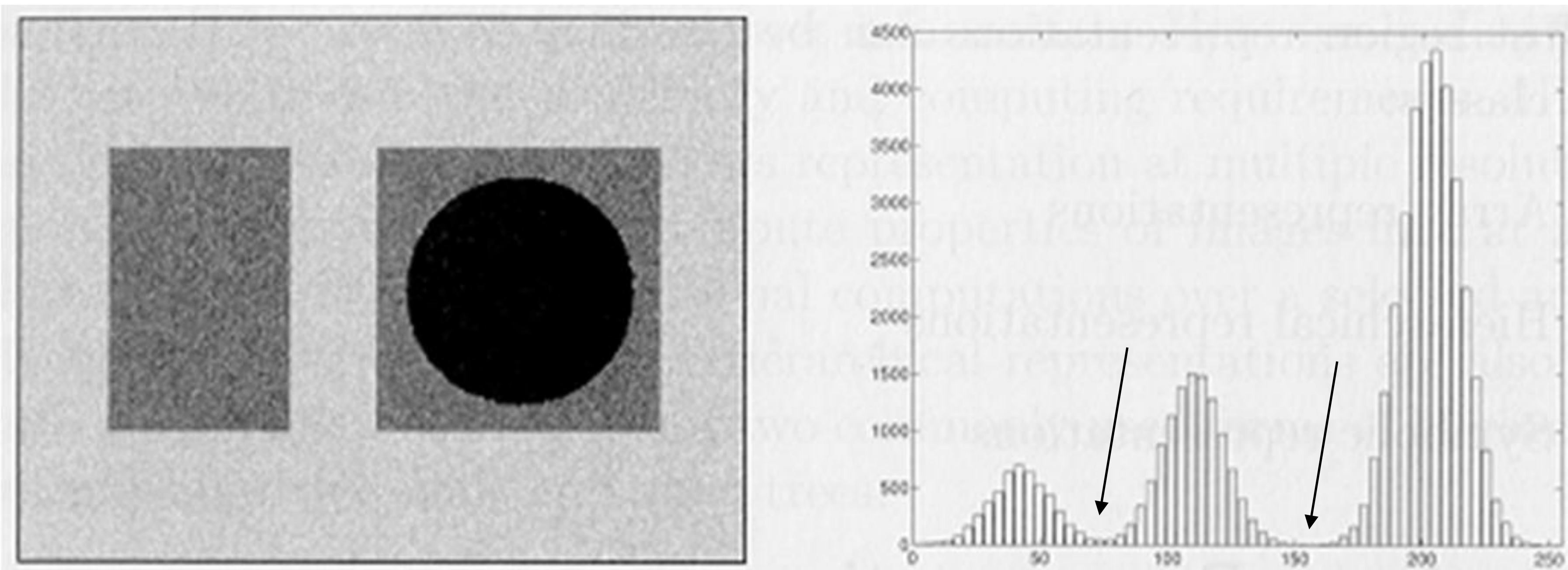




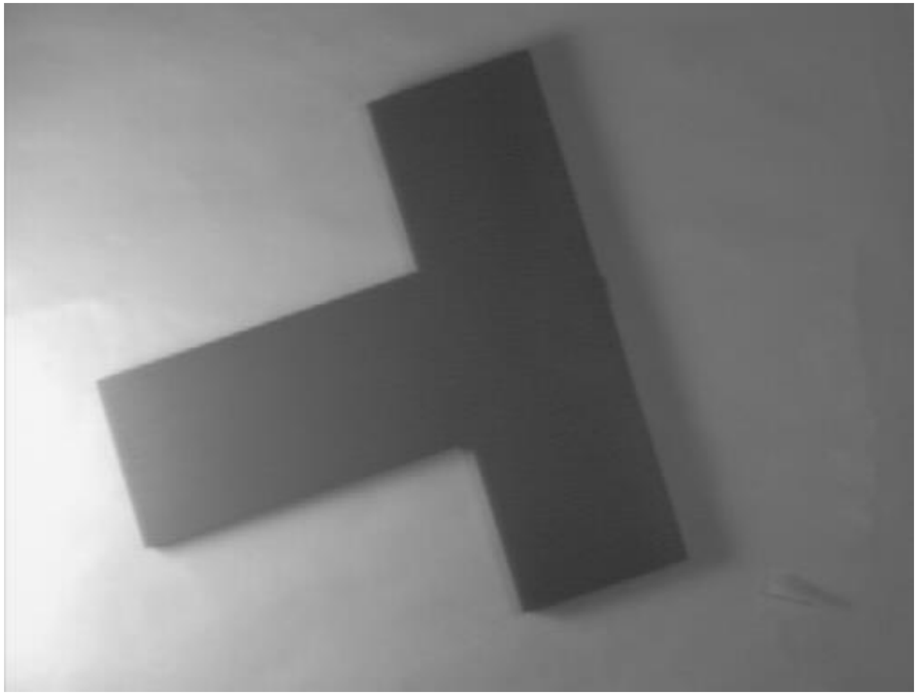
Start with average gray level and $T_0 = 0$

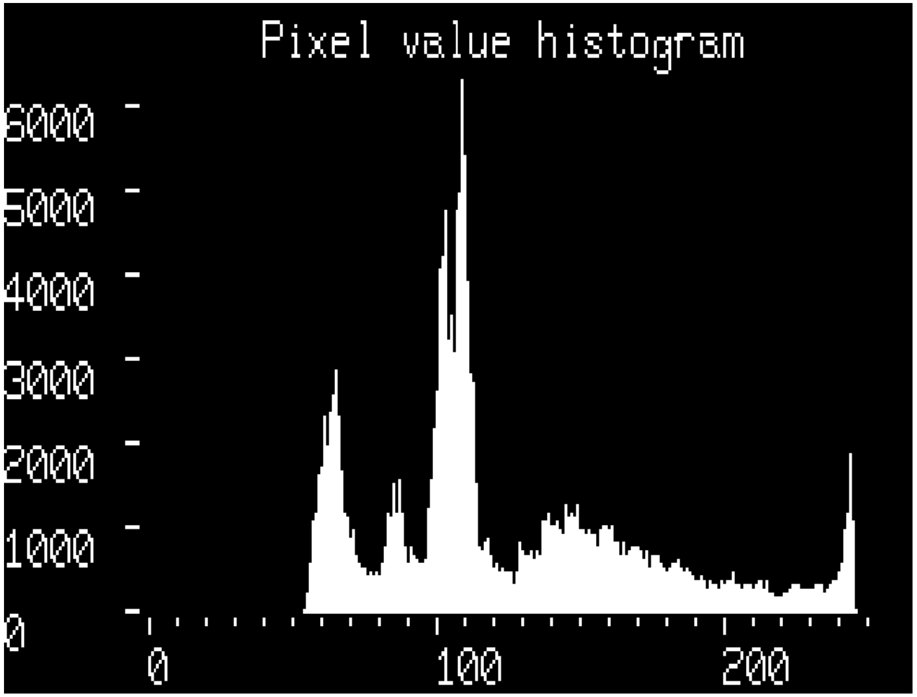
Algorithm results in $\tilde{T} = 125.4$ after 3 iterations, so let $T = 125$

Thresholding – Multi-level

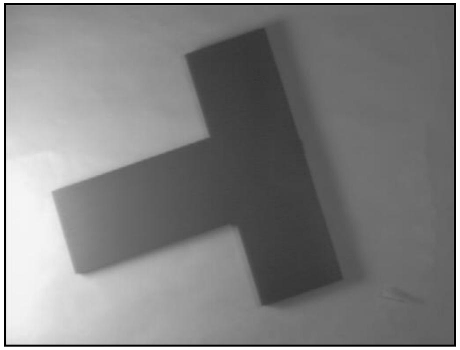


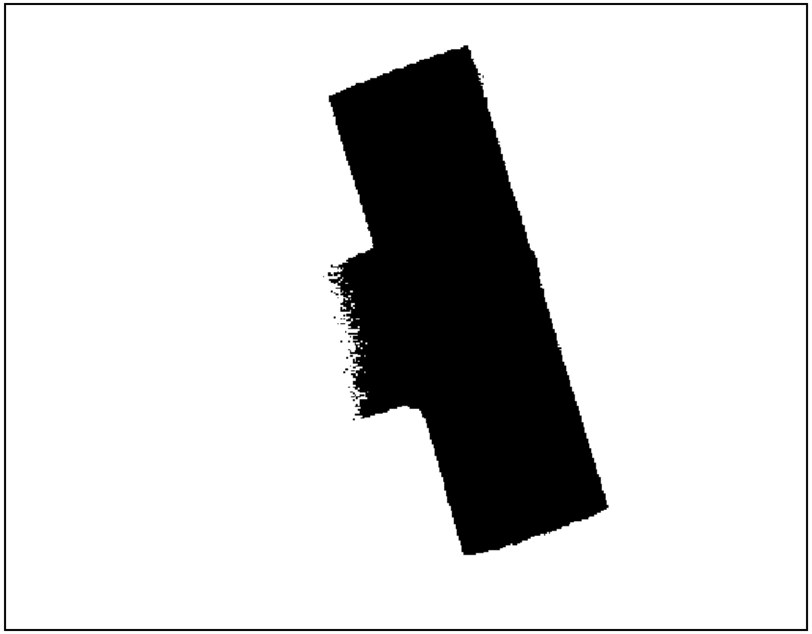
Thresholding – Non Uniform Illumination

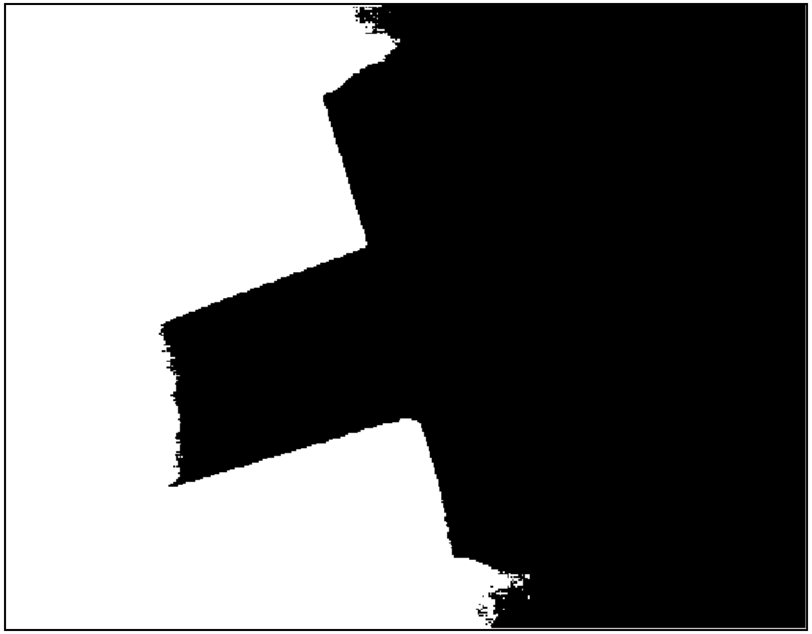




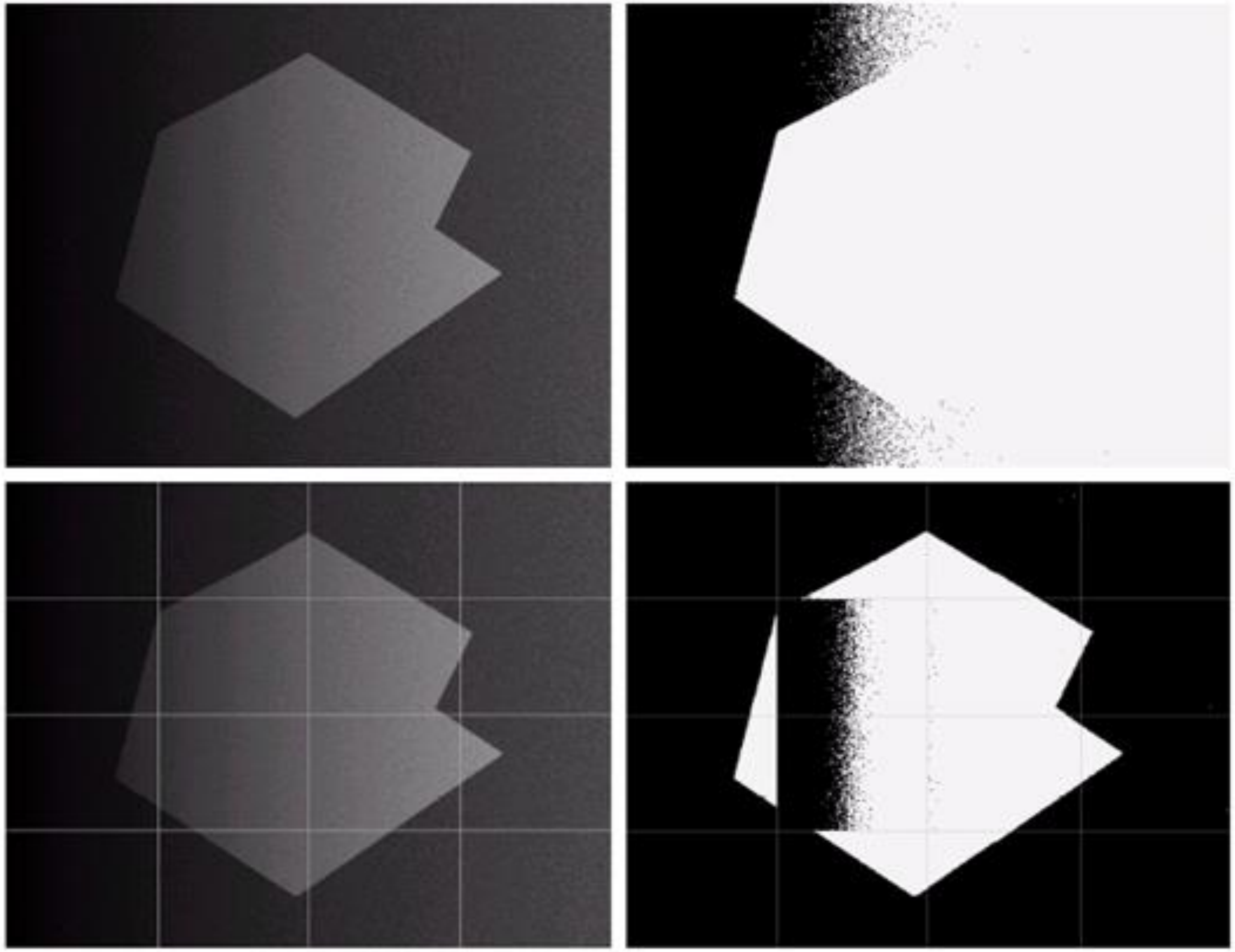
Thresholding – Non Uniform Illumination



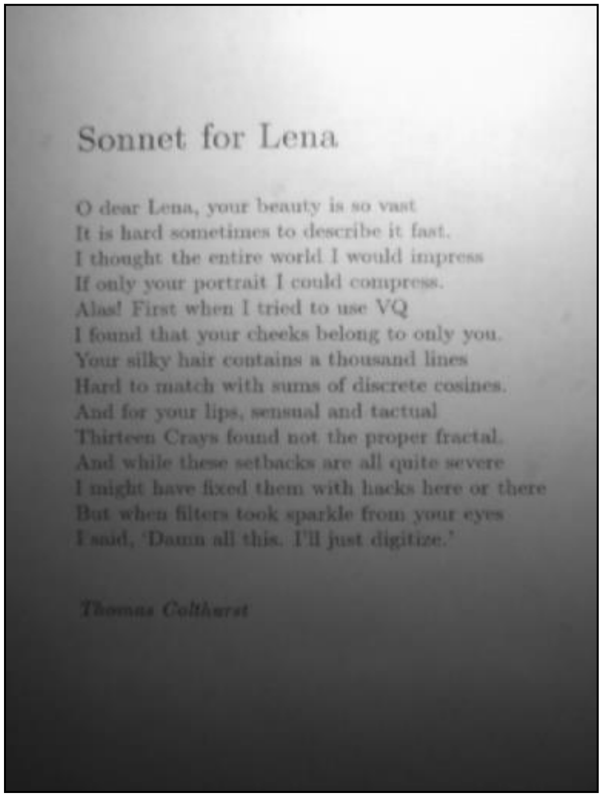


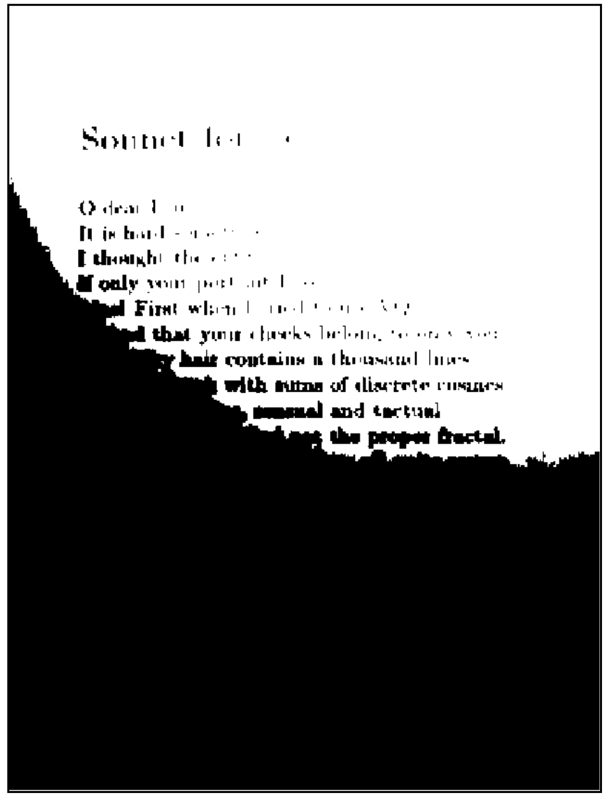


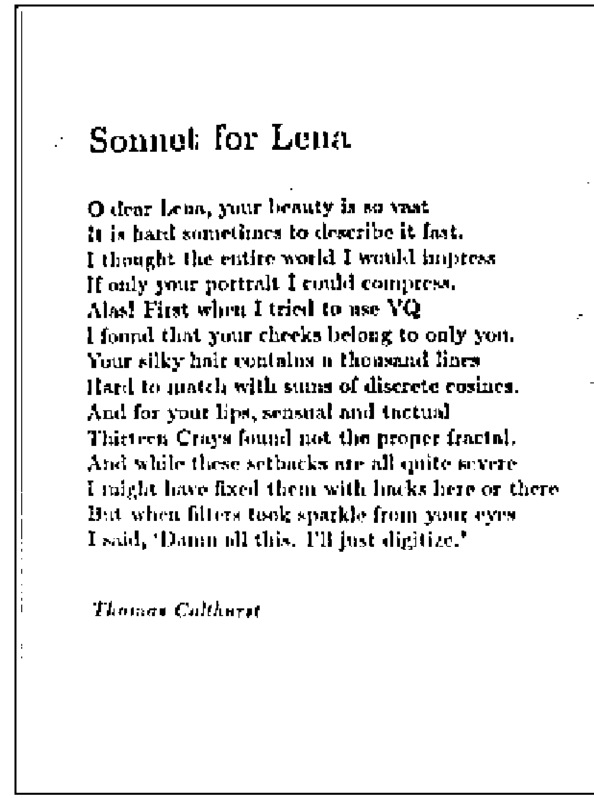
Thresholding – Non Uniform Illumination



Thresholding – Non Uniform Illumination





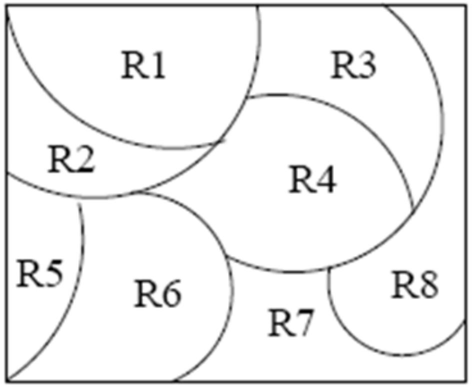


Original Image | Global Thresholding | T=mean, neighborhood=7x7

Region Growing
Region Split-Merge

Region Based Segmentation

- Divide the image into regions
  - $R_1, R_2, \ldots, R_N$
- Following properties must hold:



$$(1)\ R_1 \cup R_2 \cup \cdots \cup R_n = R$$

$$(2)\ R_i \text{ is connected}$$

$$(3)\ R_i \cap R_j = empty$$

$$(4)\ P(R_i) = \text{True}$$

Region R_i Satisfies the similarity condition

$$(5)\ P(R_i \cup R_j) = \text{False}$$

(For adjacent regions)

1. Region Growing Segmentation

- Region Growing
  - Region growing: groups pixels or subregions into larger regions.
  - *Pixel aggregation:* starts with a set of "seed" points and from these grows regions by appending to each seed points those neighboring pixels that have similar properties (such as gray level).

1. Choose the seed pixel.
2. Check the neighboring pixels and add them to the region if they are similar to the seed
3. Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added

1. Region Growing Segmentation

- Example

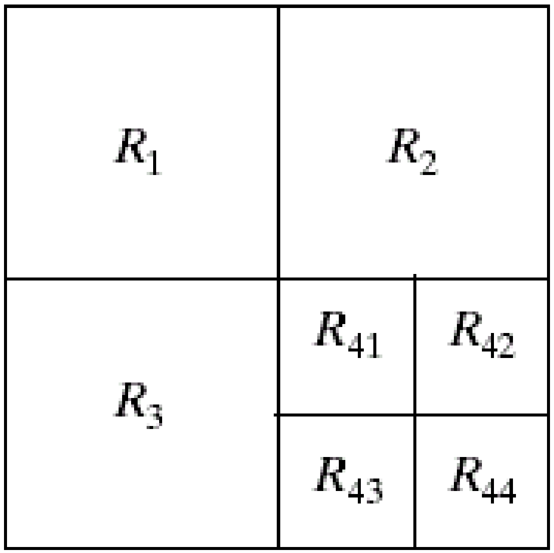
10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

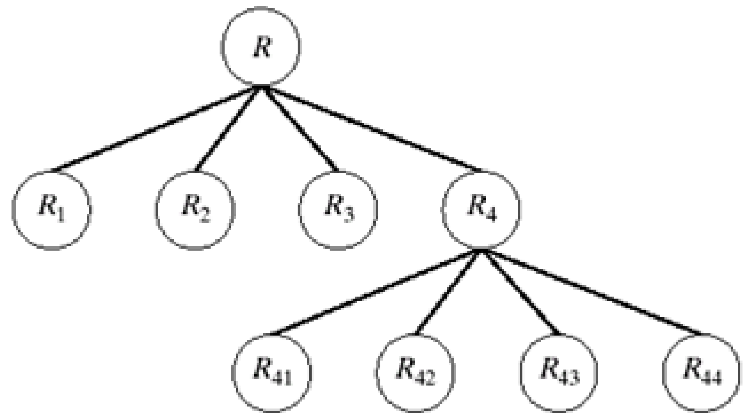
2. Region Splitting Segmentation

Region Splitting

- Initially take the image as a whole to be the area of interest.
- Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.
- If **TRUE** then the area of interest corresponds to a region in the image.
- If **FALSE** split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.
- This process continues until no further splitting occurs.

2. Region Splitting Segmentation





Problem? Adjacent regions could be same

Solution? Allow Merge

2. Region Splitting Segmentation

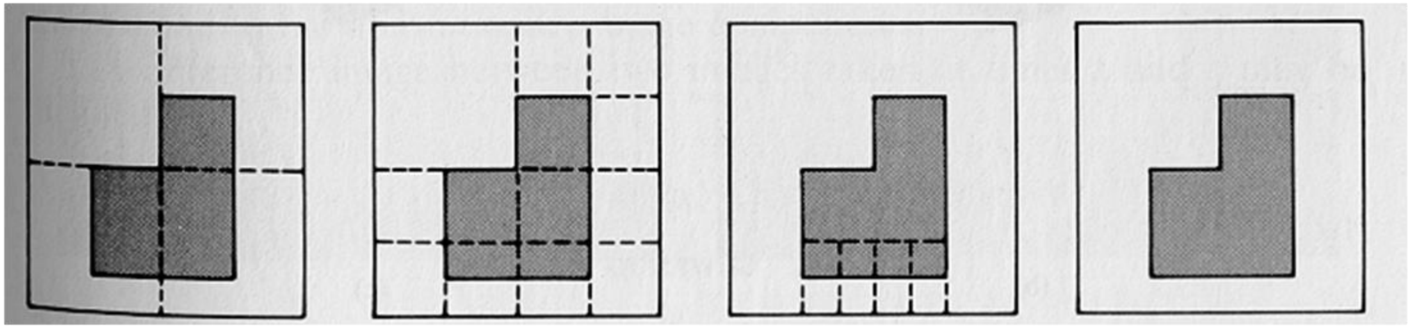
- Region Merging
  - Region merging is the opposite of region splitting.
  - Merge adjacent regions R_i and R_j for which:

$$P(R_i \cup R_j) = True$$

- Region Splitting/Merging
  - Stop when no further split or merge is possible

2. Region Splitting Segmentation

- Example



1. Split into four disjointed quadrants any region R_i where $P(R_i)$=False
2. Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k)$=True
3. Stop when no further merging or splitting is possible

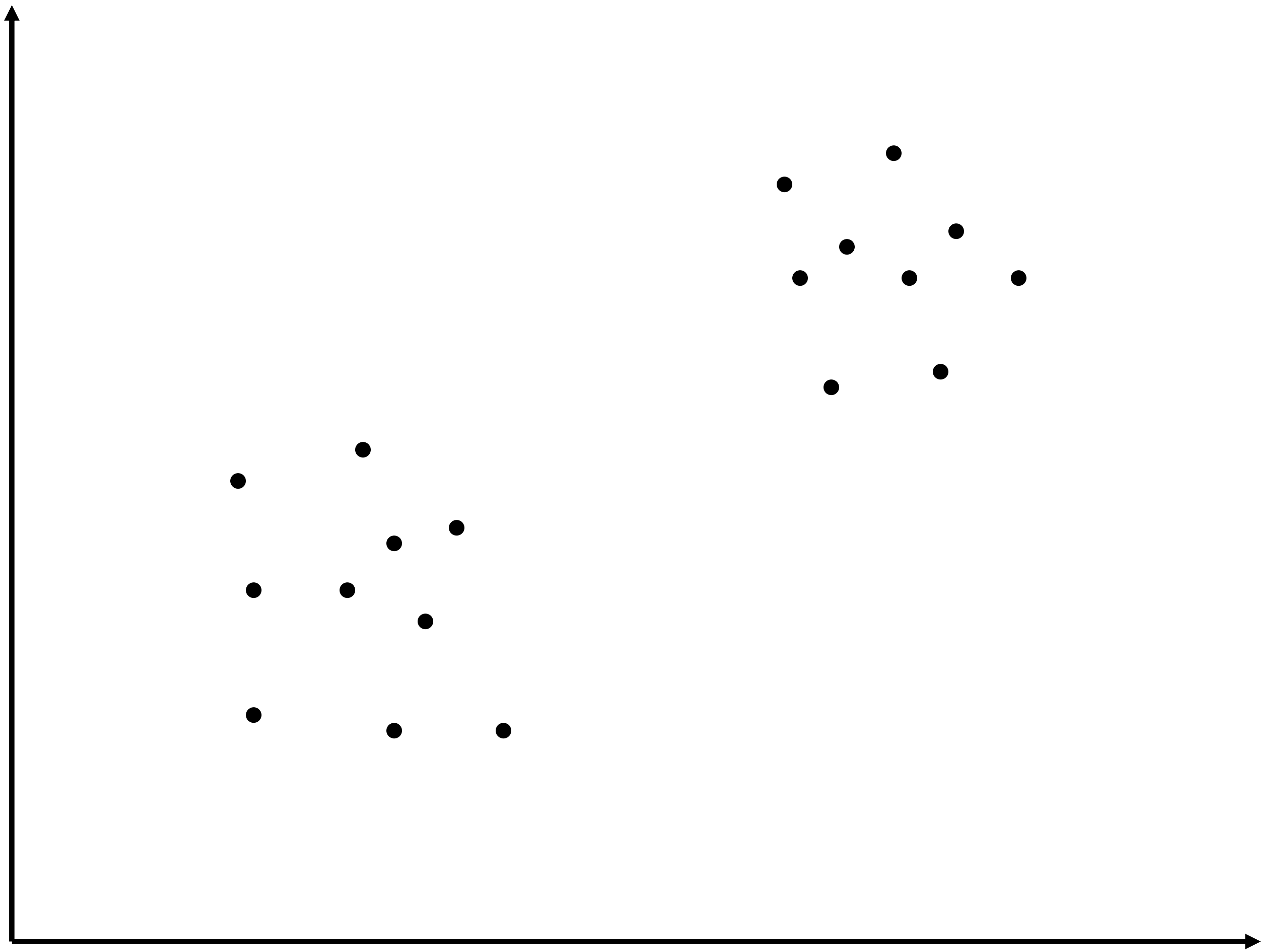
K-Mean

C-Mean

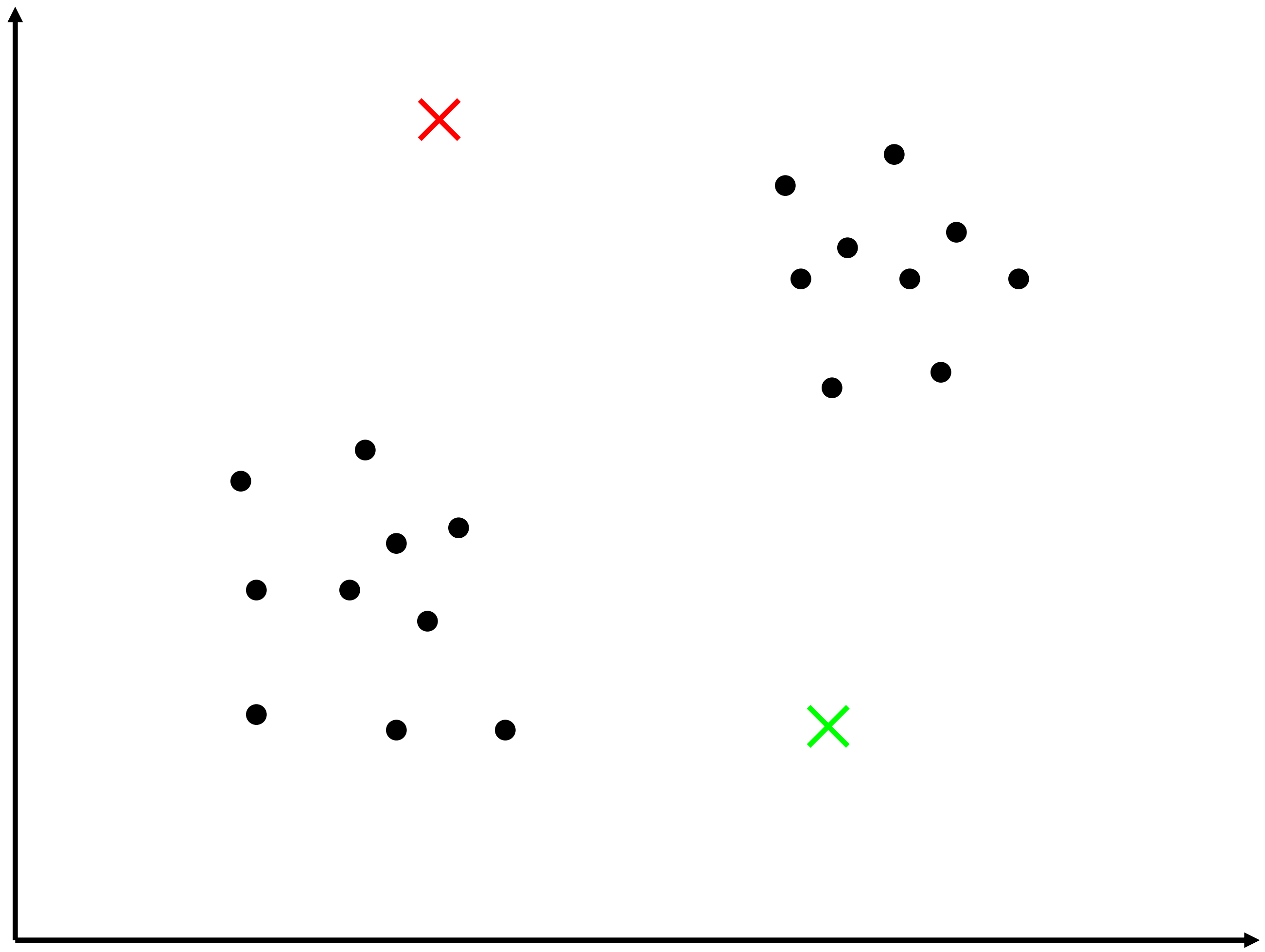
3. K-Mean Segmentation

1. Chose the number (K) of clusters and randomly select the centroids of each cluster.
2. For each data point:
   - Calculate the distance from the data point to each cluster.
   - Assign the data point to the closest cluster.
3. Re-compute the centroid of each cluster.
4. Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

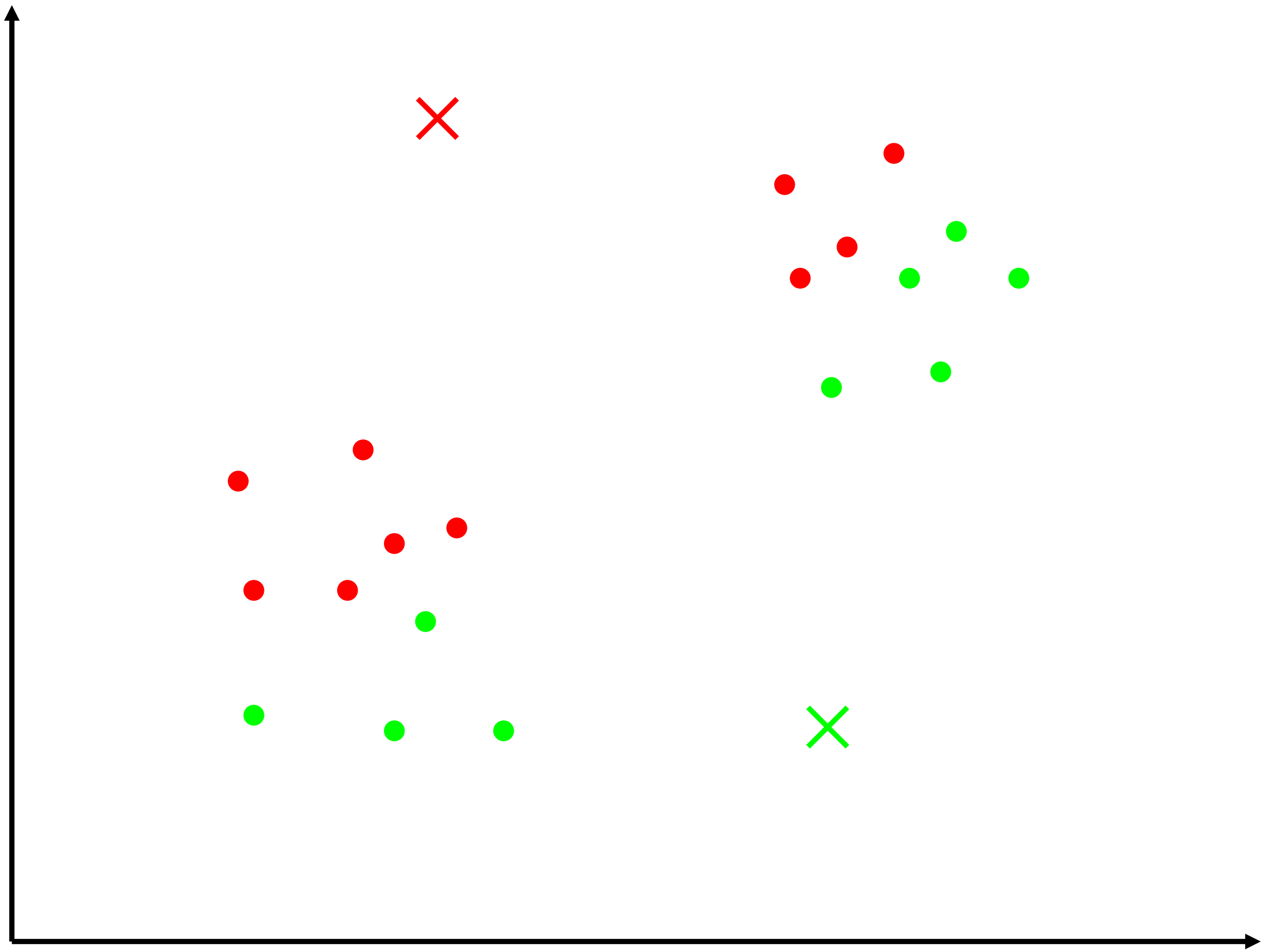
3. K-Mean Segmentation



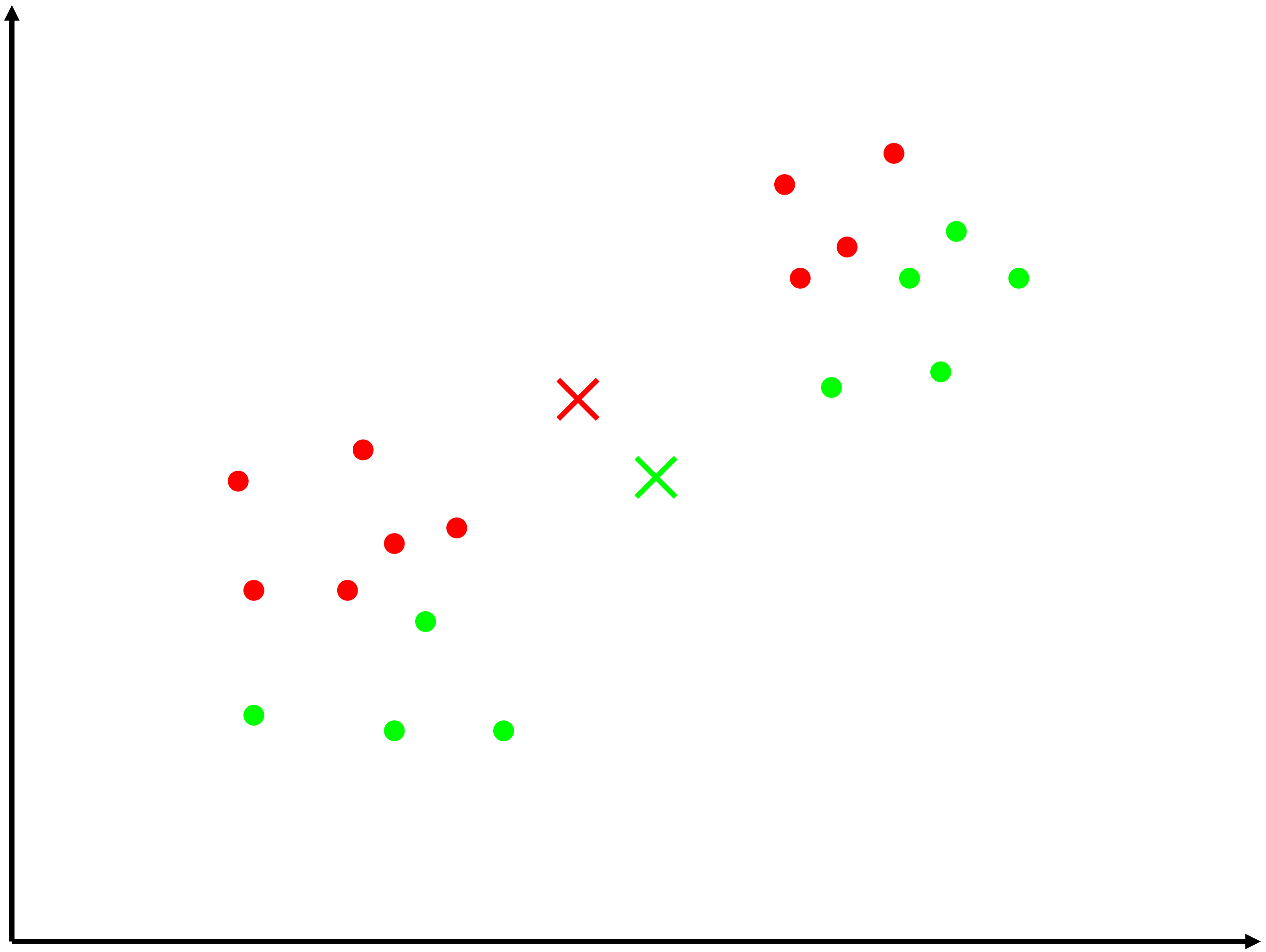
3. K-Mean Segmentation



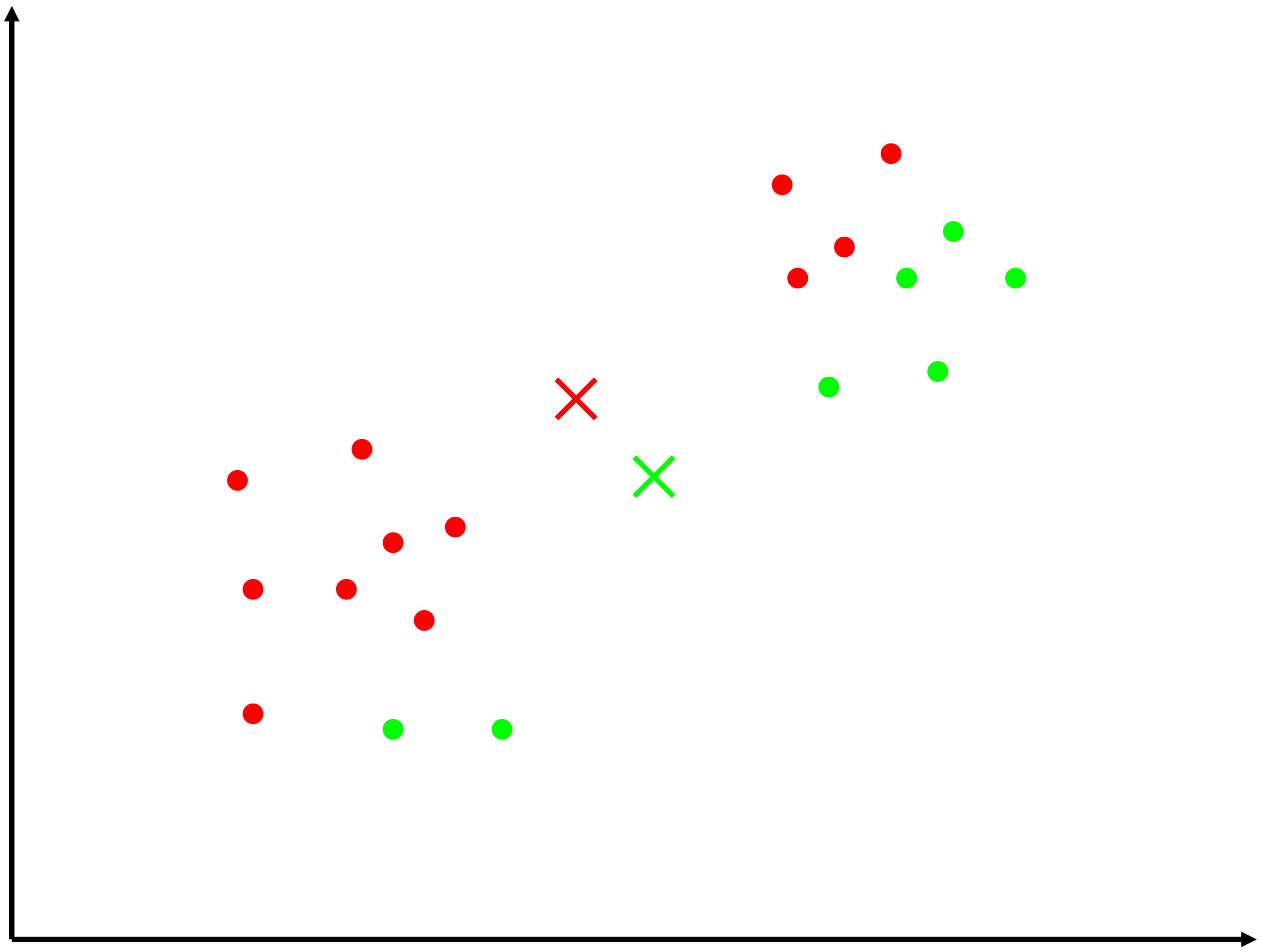
3. K-Mean Segmentation



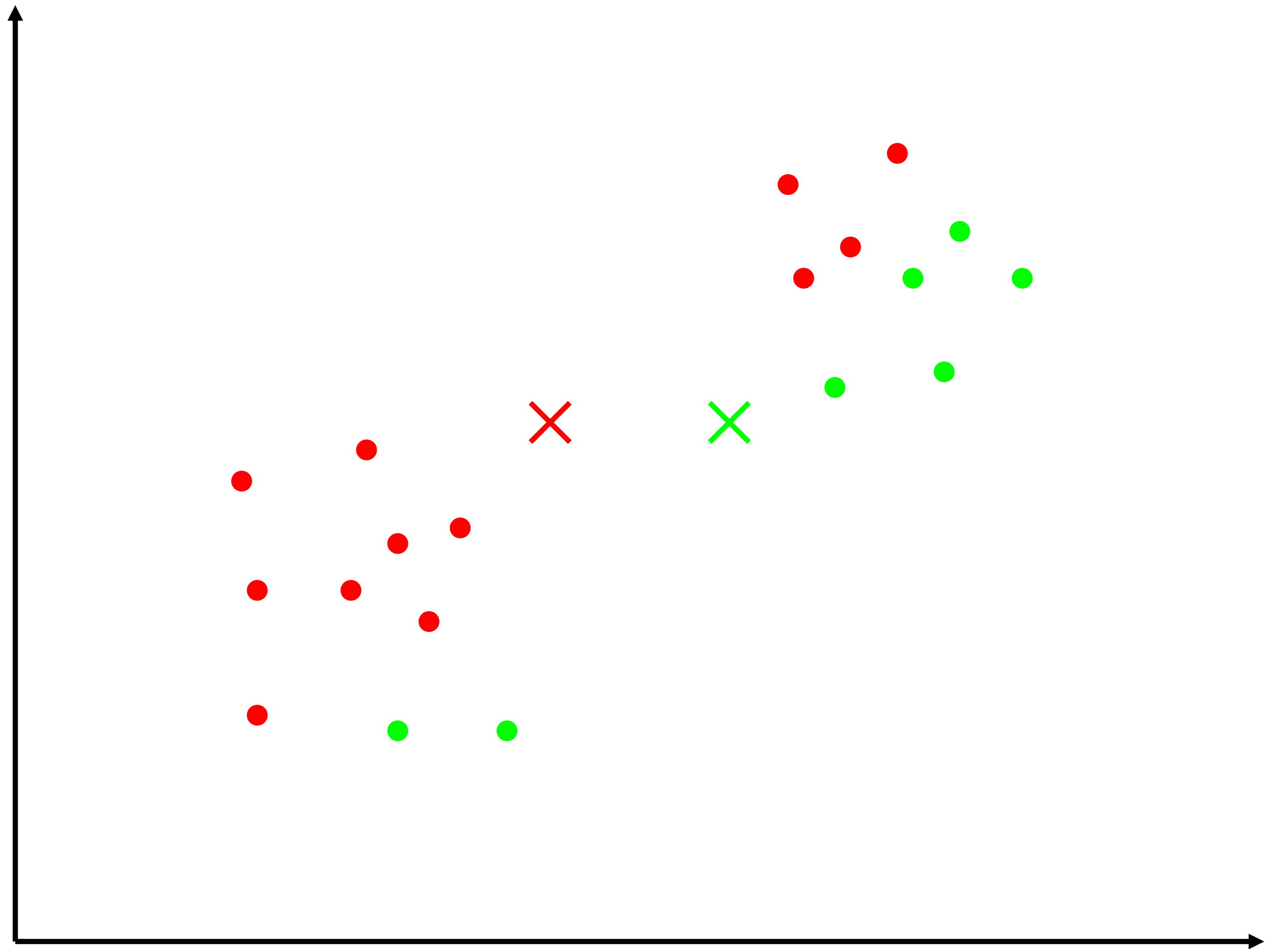
3. K-Mean Segmentation



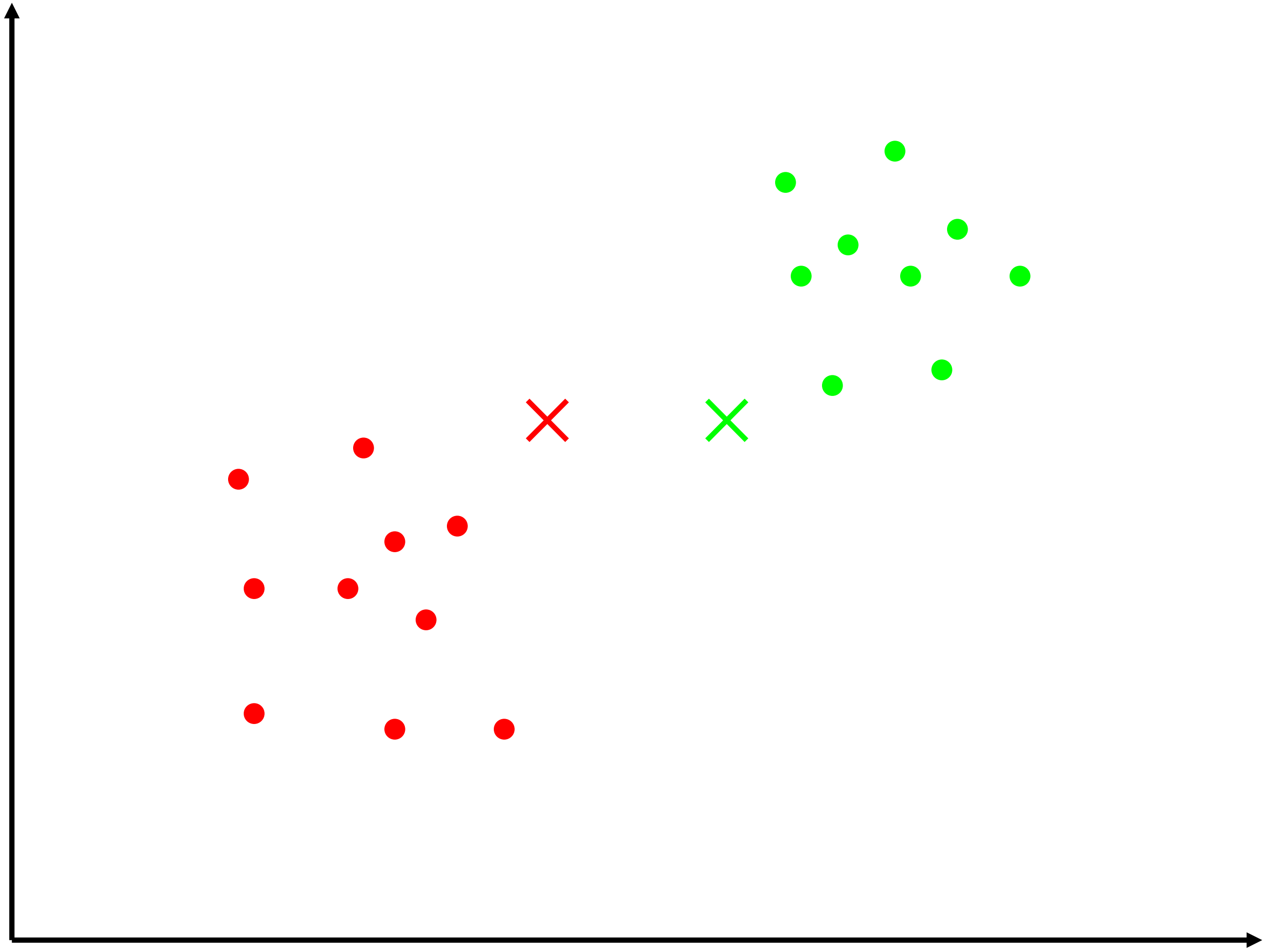
3. K-Mean Segmentation



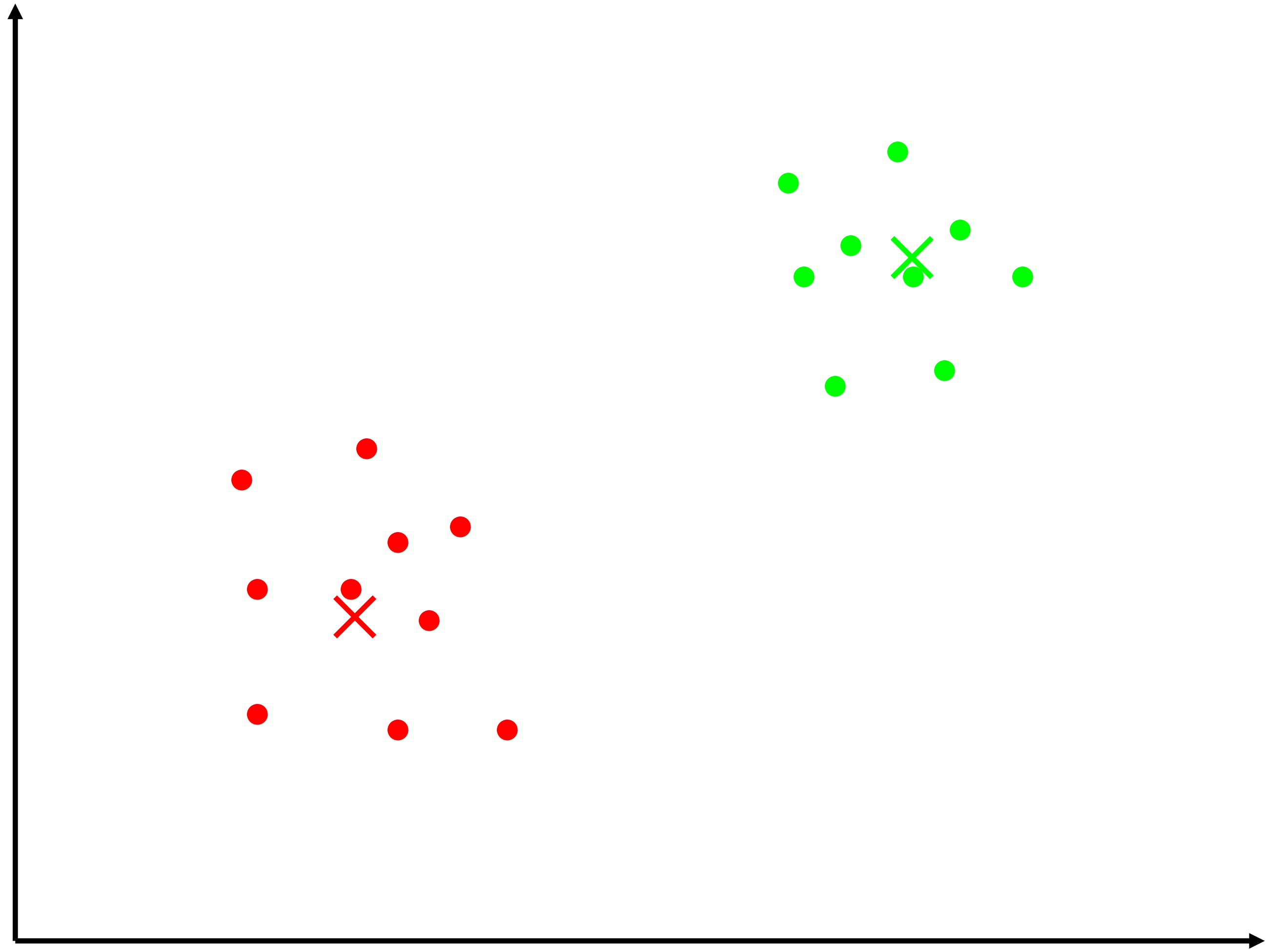
3. K-Mean Segmentation



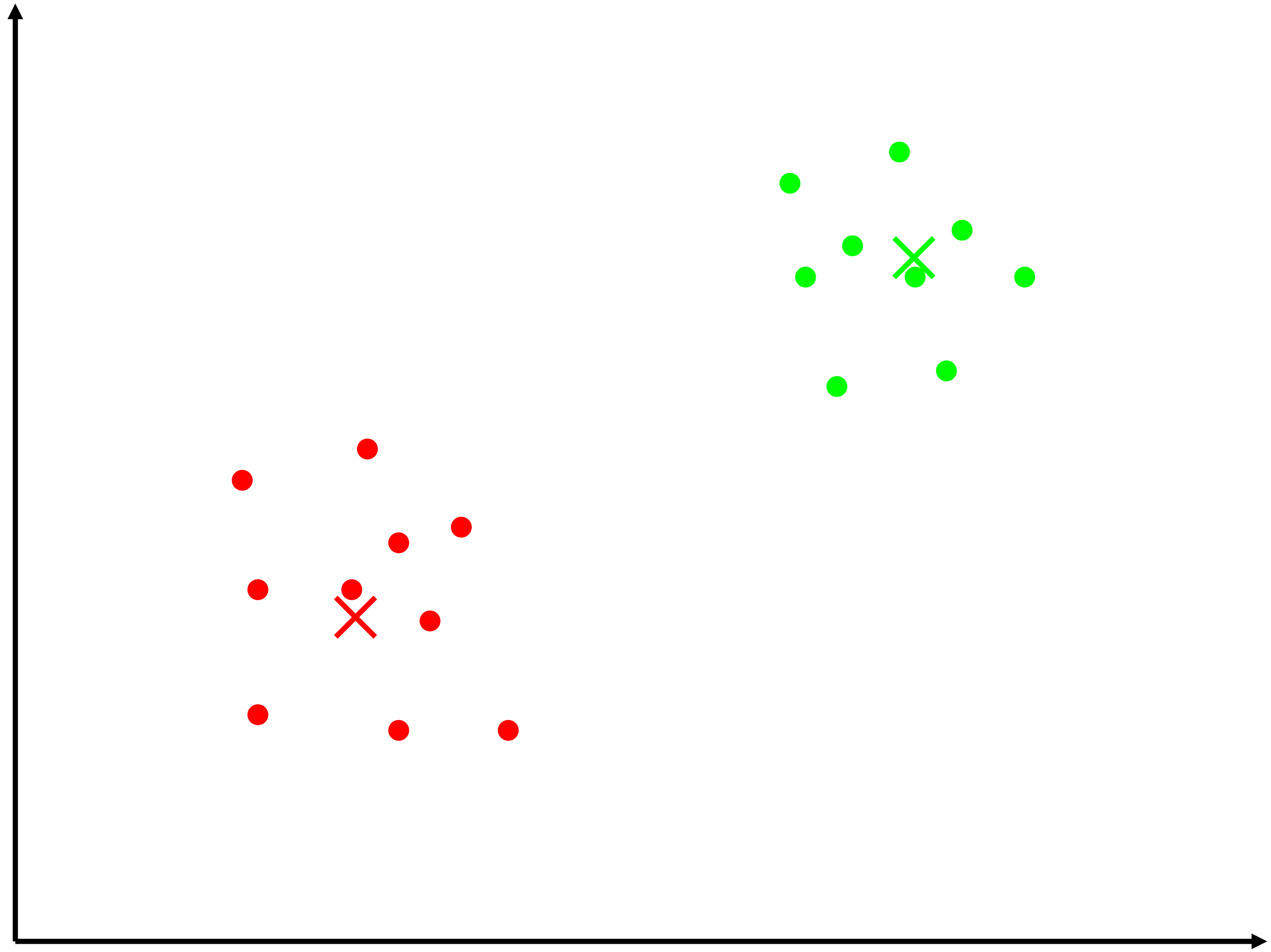
3. K-Mean Segmentation



3. K-Mean Segmentation



3. K-Mean Segmentation



4. C-Mean Segmentation

1. Chose the number of clusters and randomly select the centroids of each cluster.
2. For each data point:
   - Calculate the distance from the data point to each cluster.
   - Instead of assigning the pixel completely to one cluster, use the weights depending on the distance of that pixel from each cluster.
   - The closer the cluster, the higher the weigh, and vice versa.
   - Re-compute the centers of the clusters using these weighted distances.

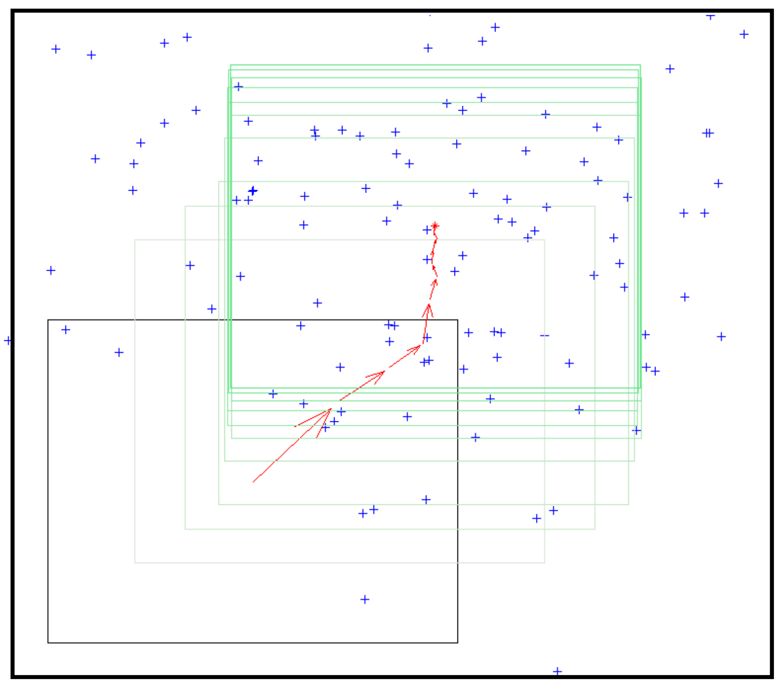
Mean Shift Segmentation

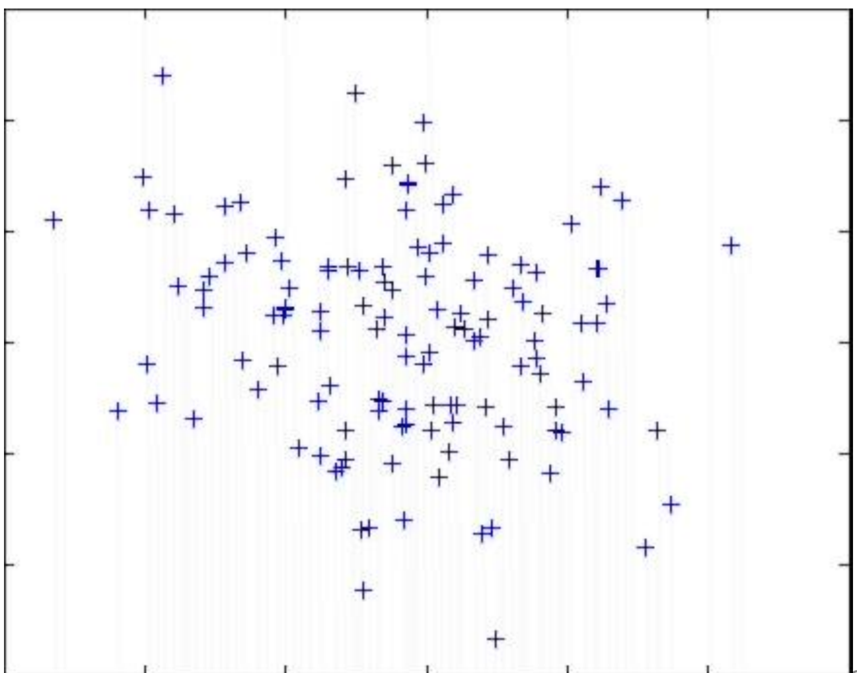
5. Mean Shift Segmentation

Mean Shift Algorithm

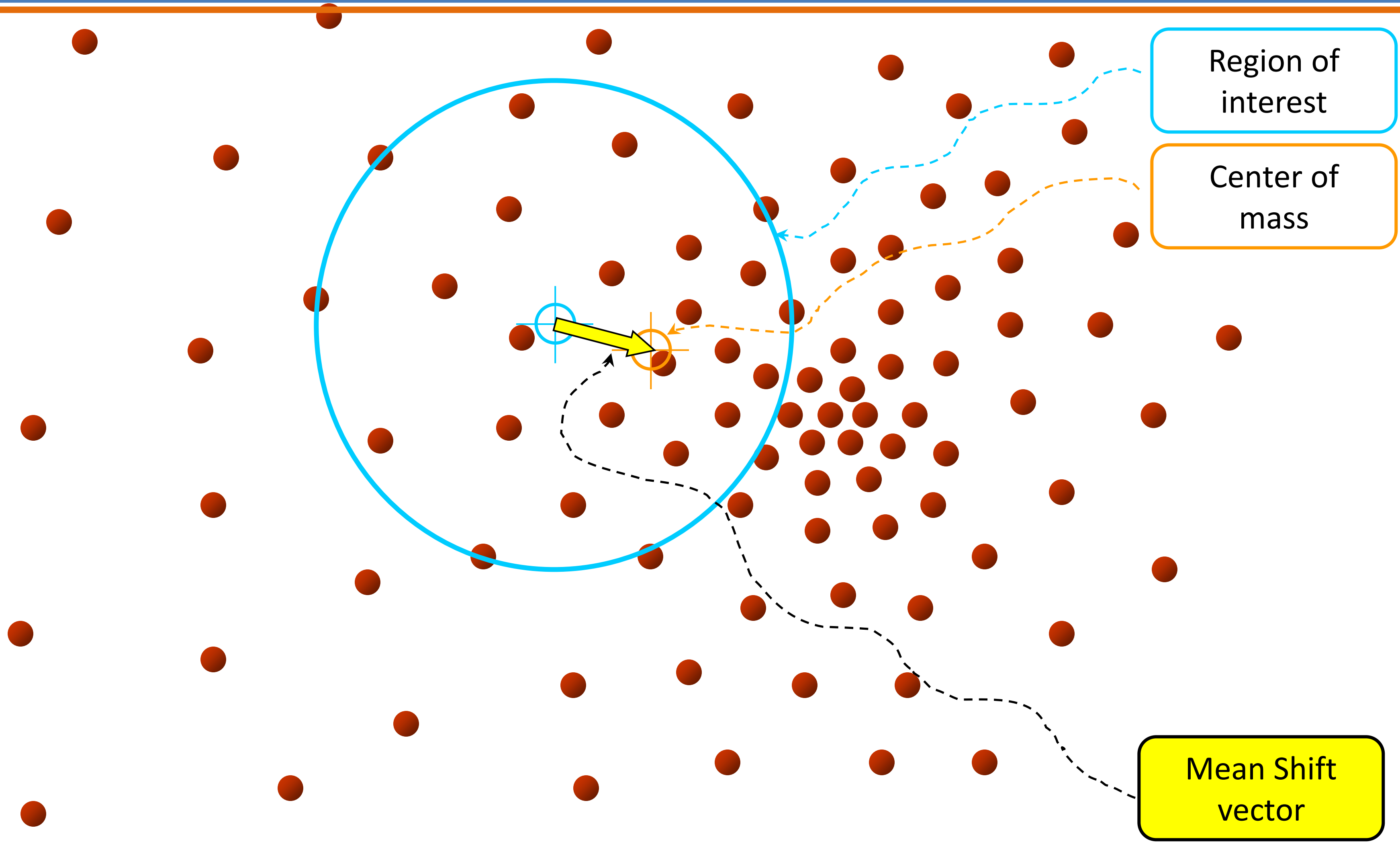
1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

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





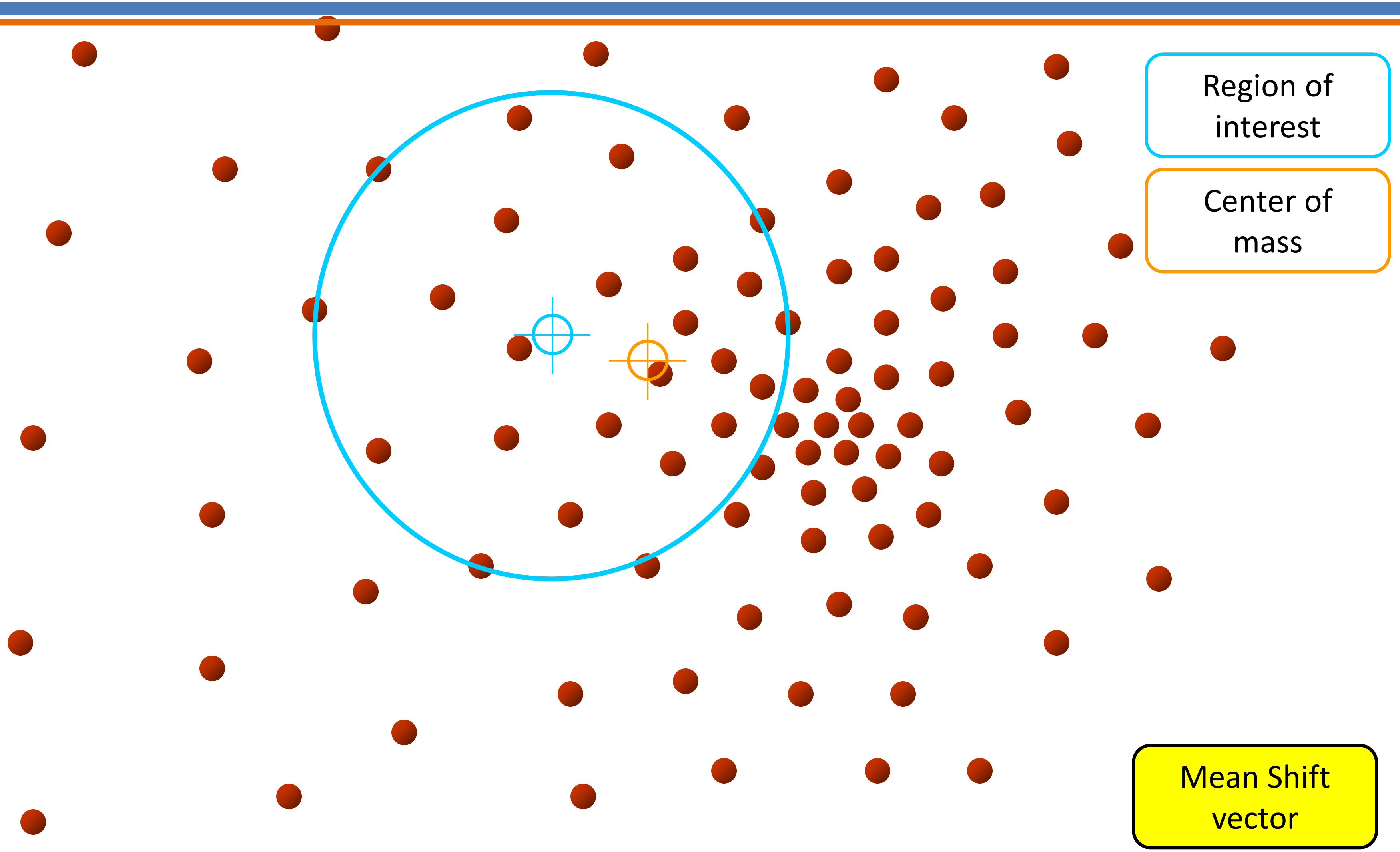
5. Mean Shift Segmentation



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

Distribution of identical billiard balls

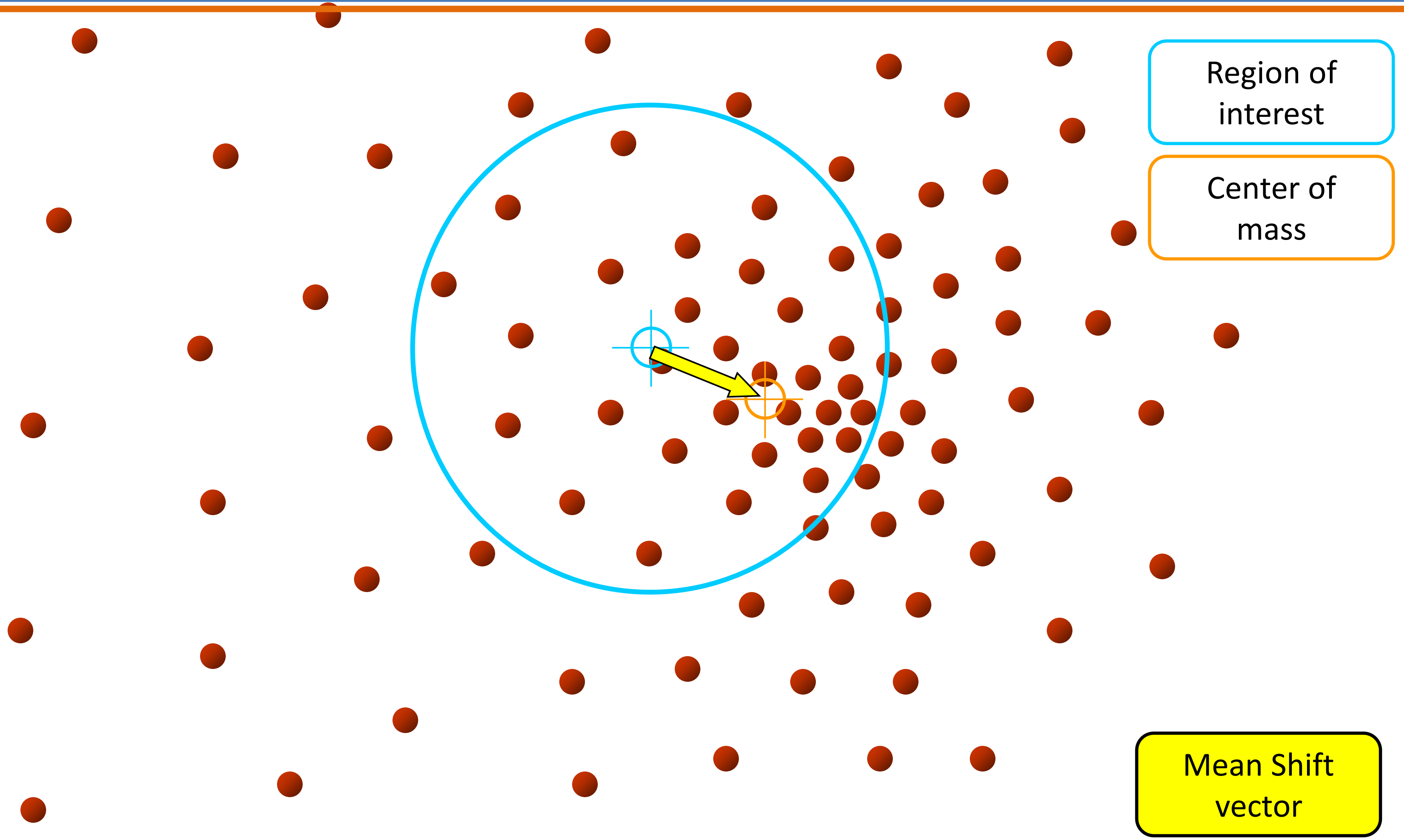
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

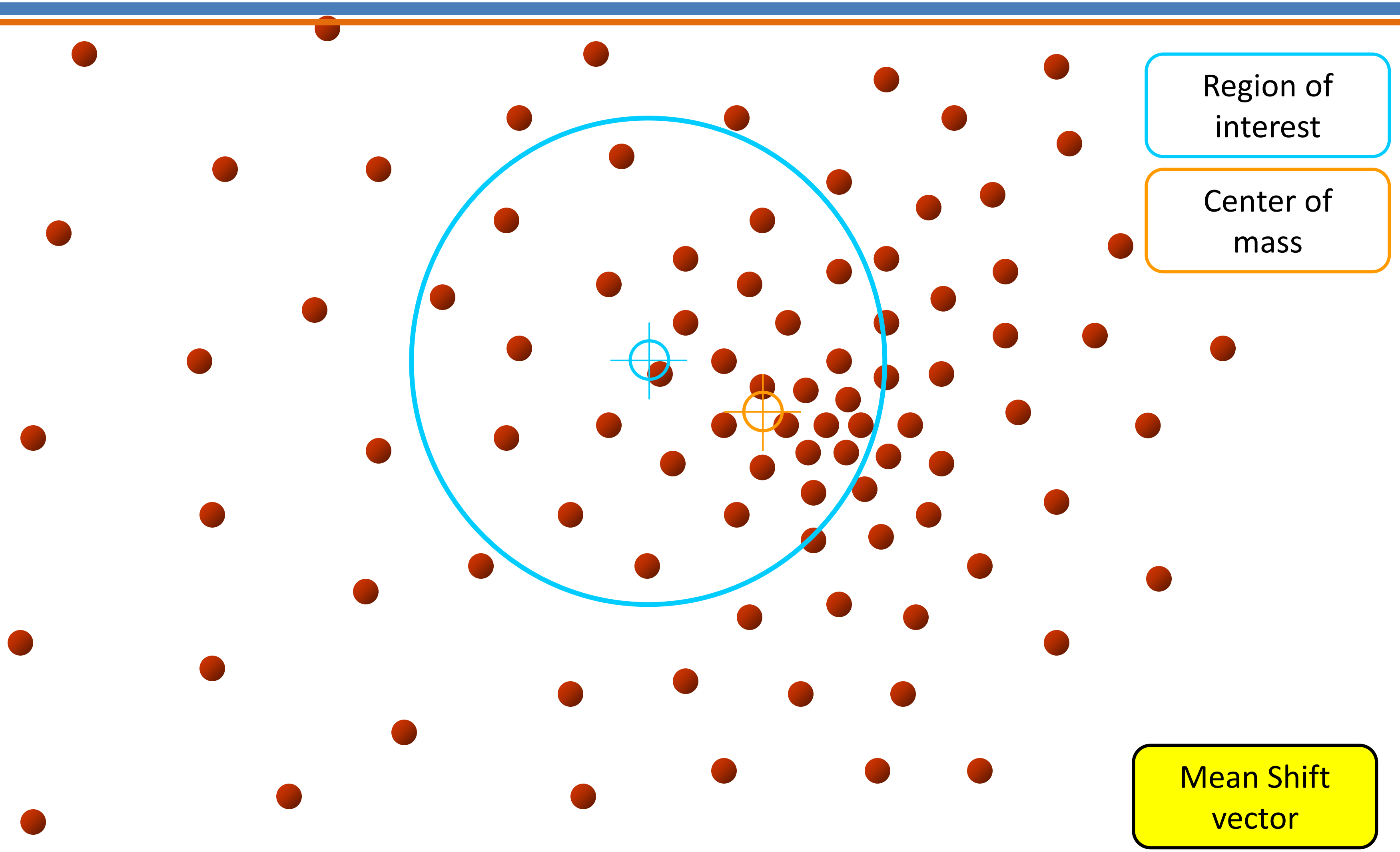
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

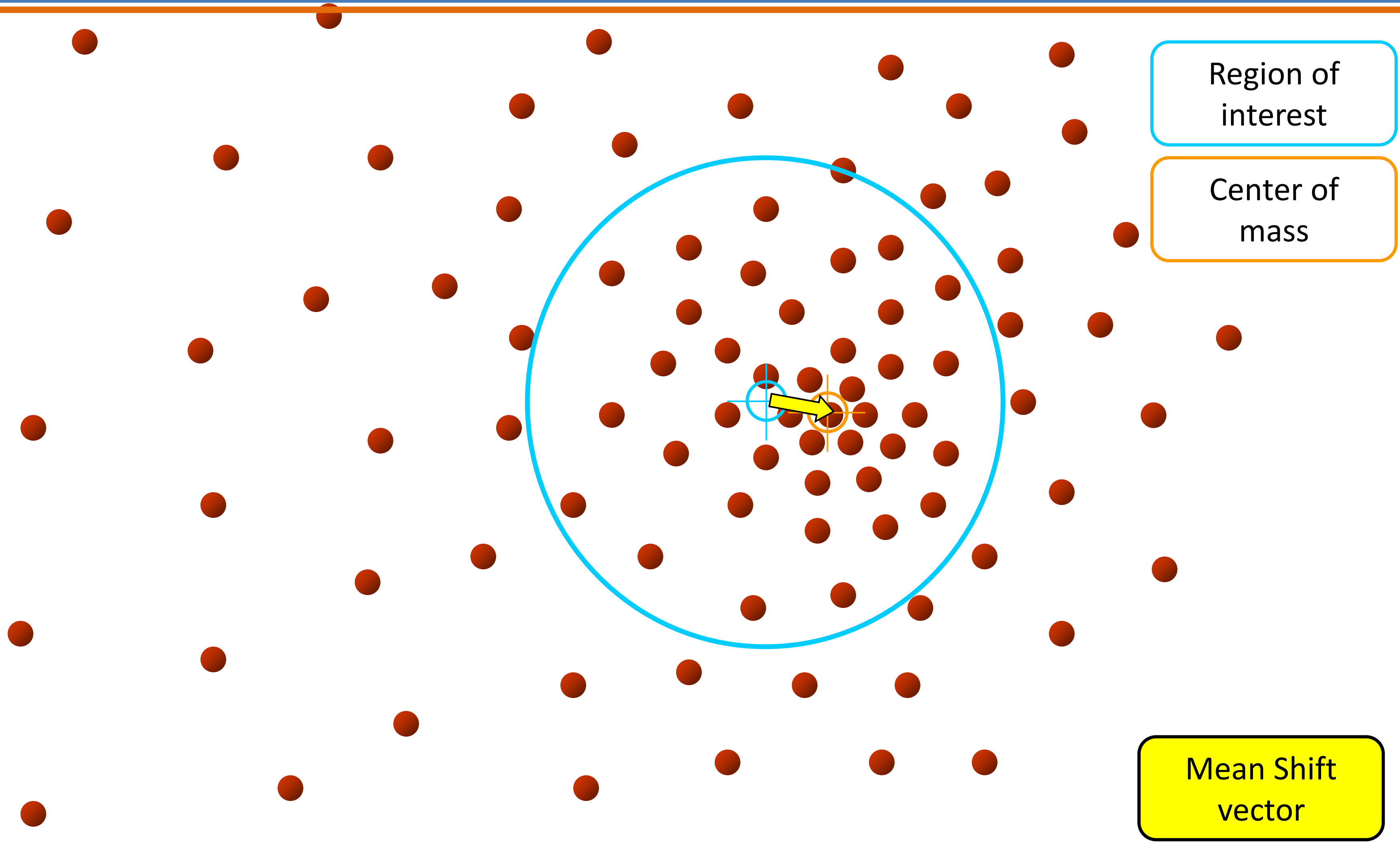
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

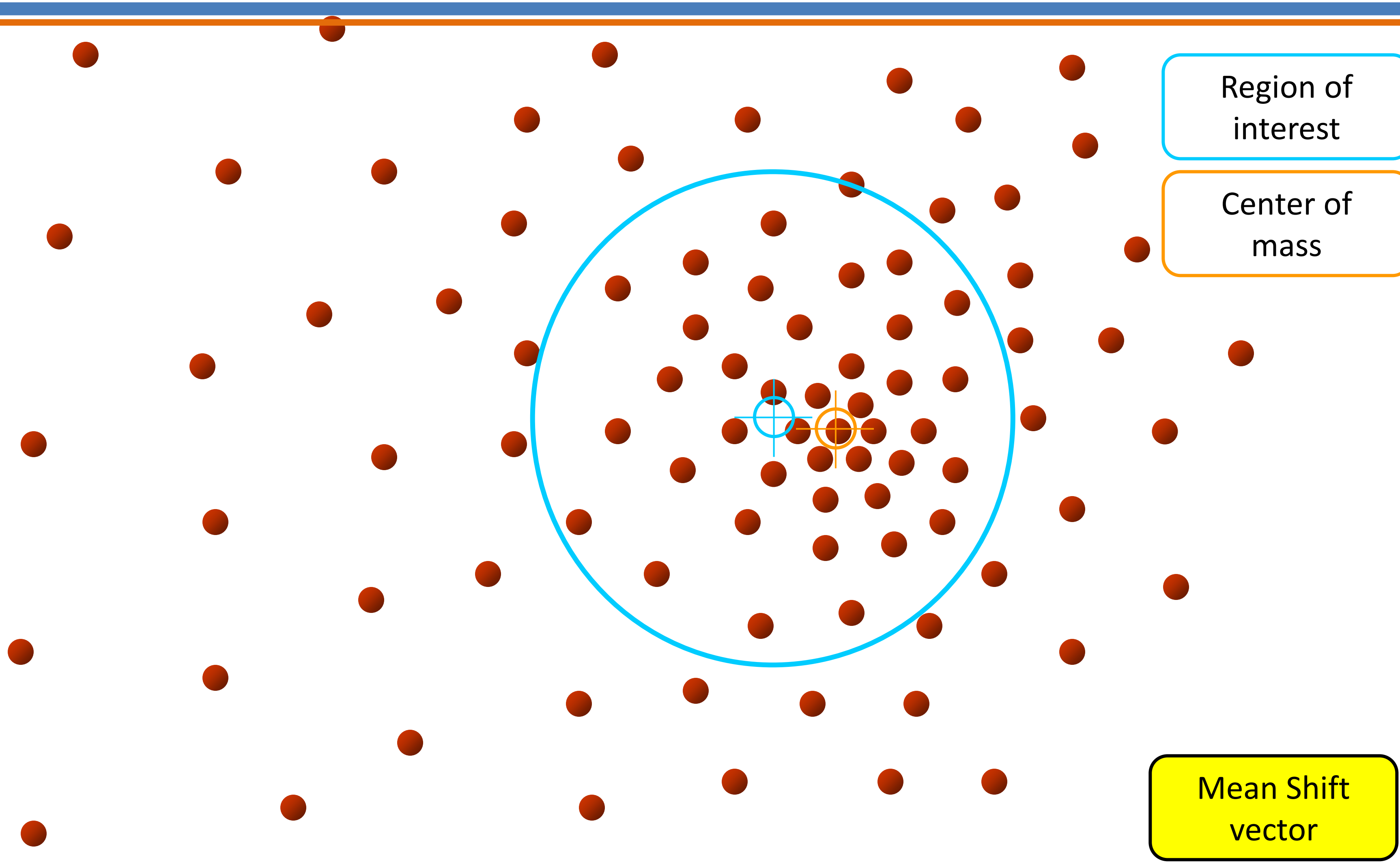
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

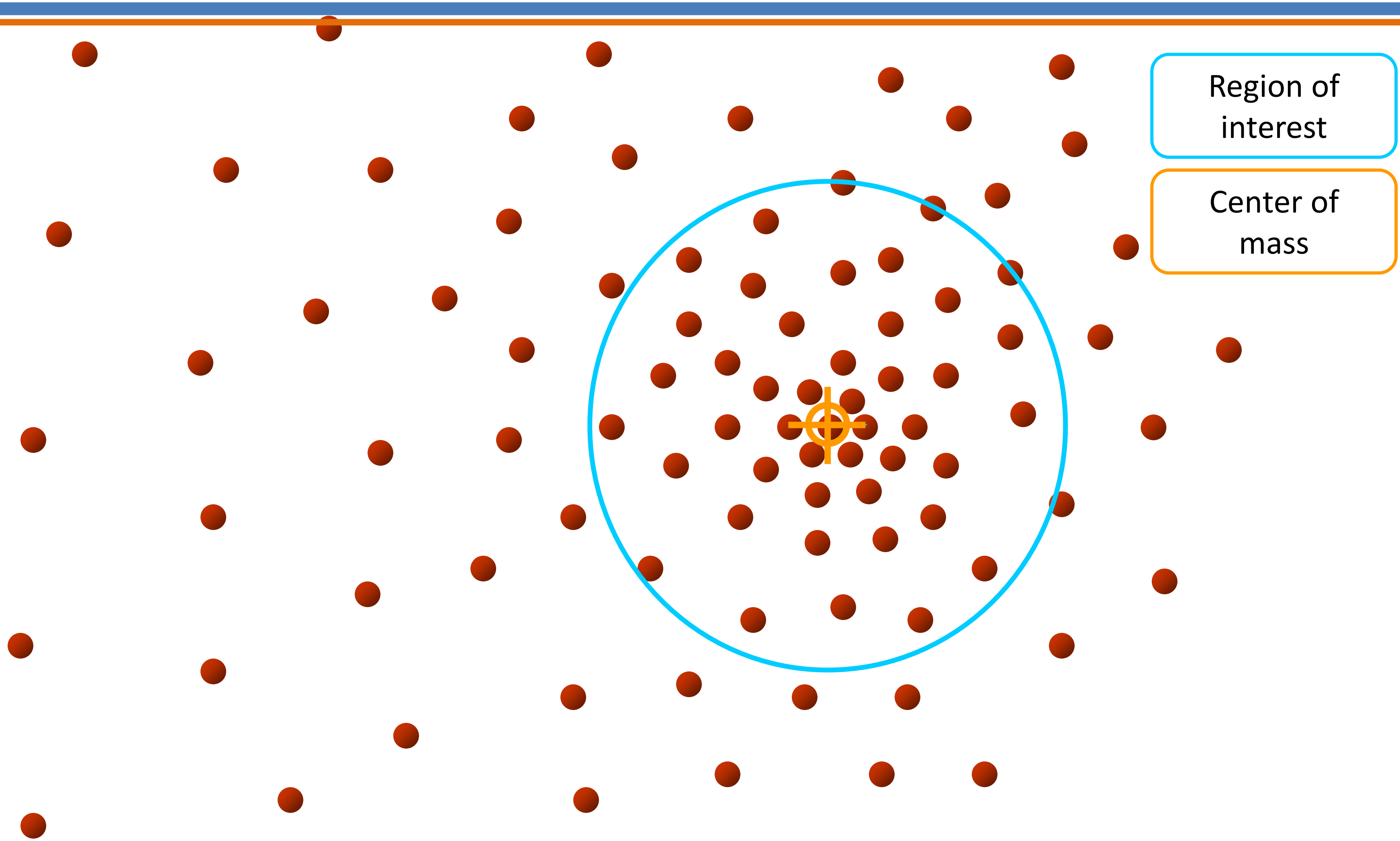
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

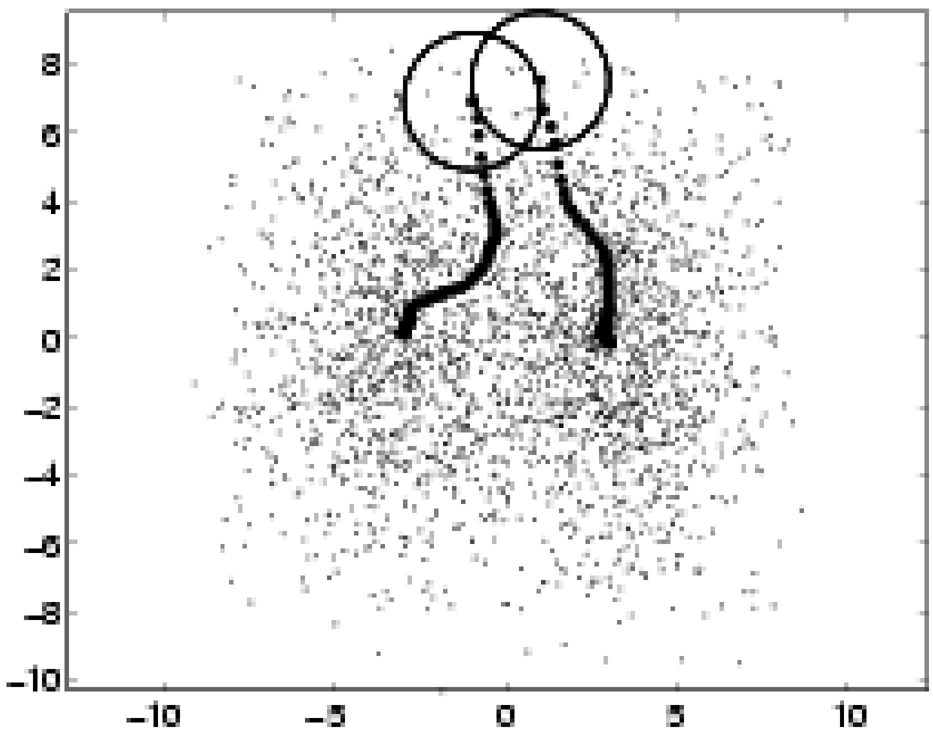
5. Mean Shift Segmentation



Objective : Find the densest region

Distribution of identical billiard balls

5. Mean Shift Segmentation - Example

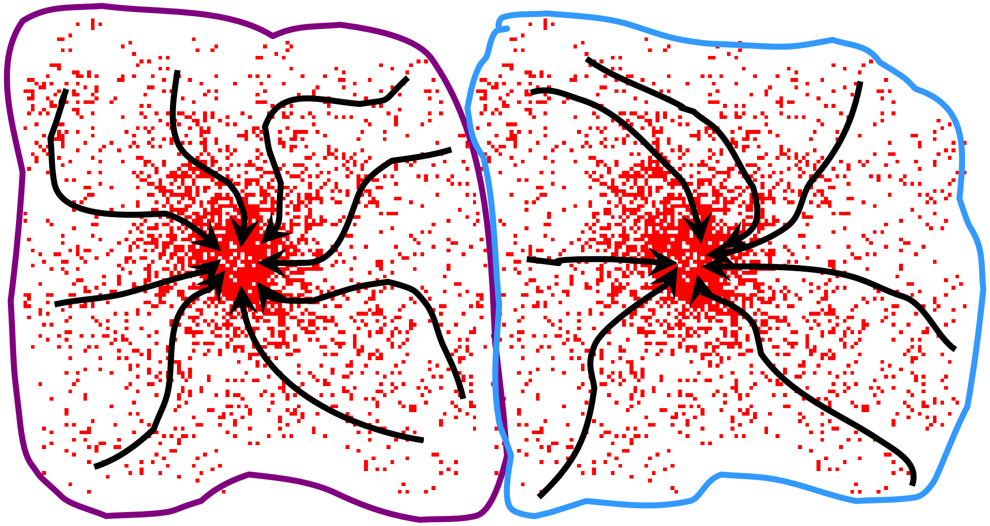


Window tracks signify the steepest ascent directions

5. Mean Shift Segmentation - Example

<u>Cluster</u> : All data points in the ***attraction basin*** of a mode

<u>Attraction basin</u> : the region for which all trajectories lead to the same mode



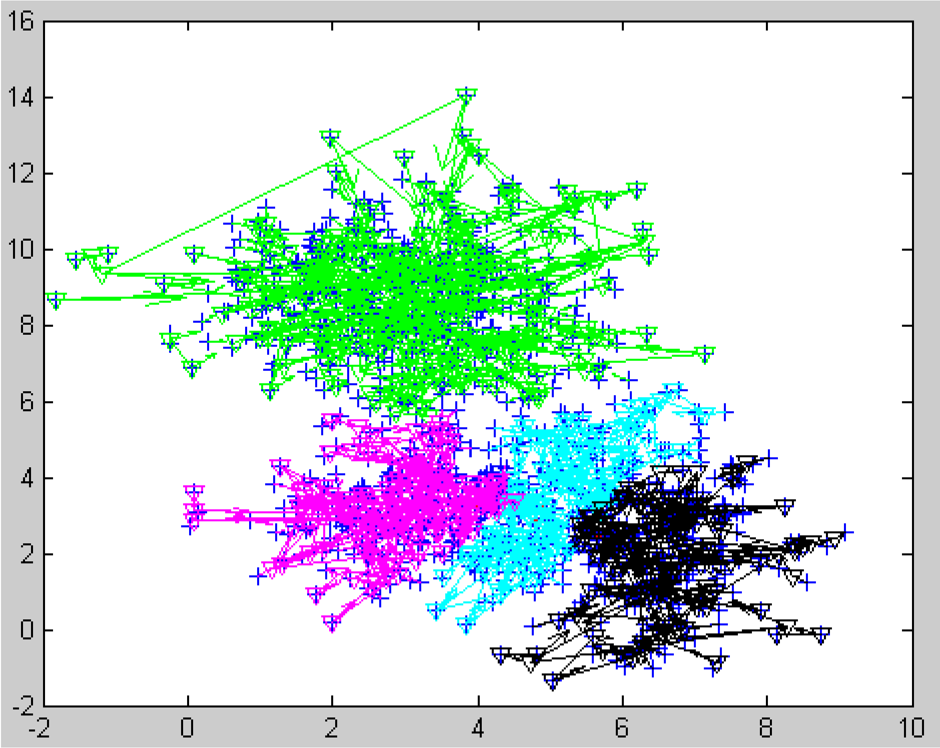
Mean Shift Segmentation Extension

MSS Is scale (search window size) sensitive. Solution, use all scales:

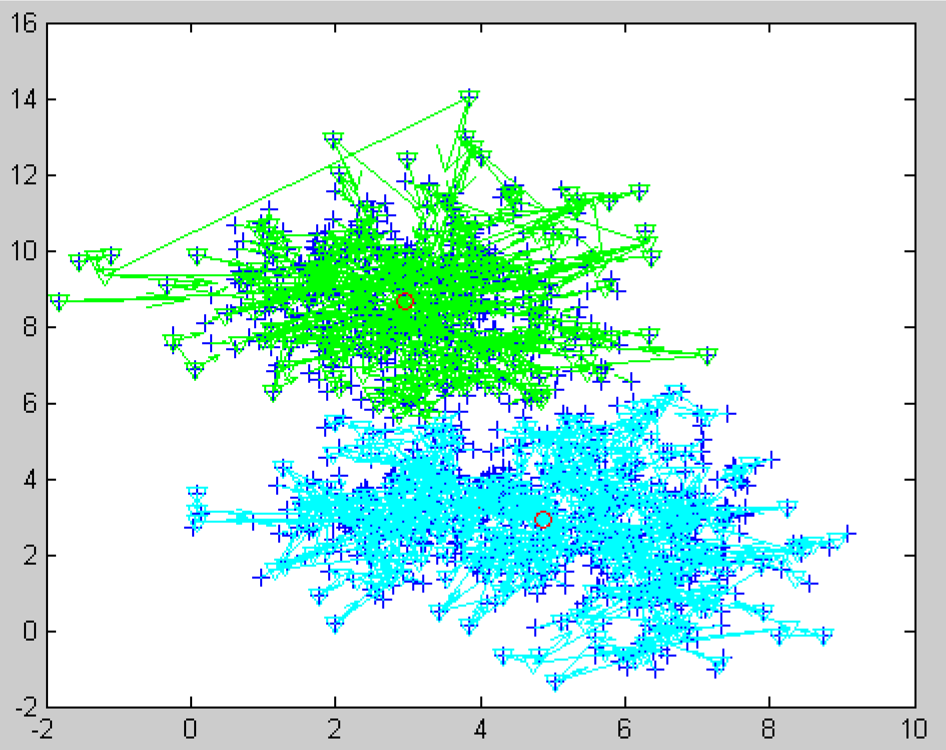
Place a tiny mean shift window over each data point

1. Grow the window and mean shift it
2. Track windows that merge along with the data they traverse
3. Until everything is merged into one cluster

Best 4 clusters:



Best 2 clusters:



K - Nearest Neighbors

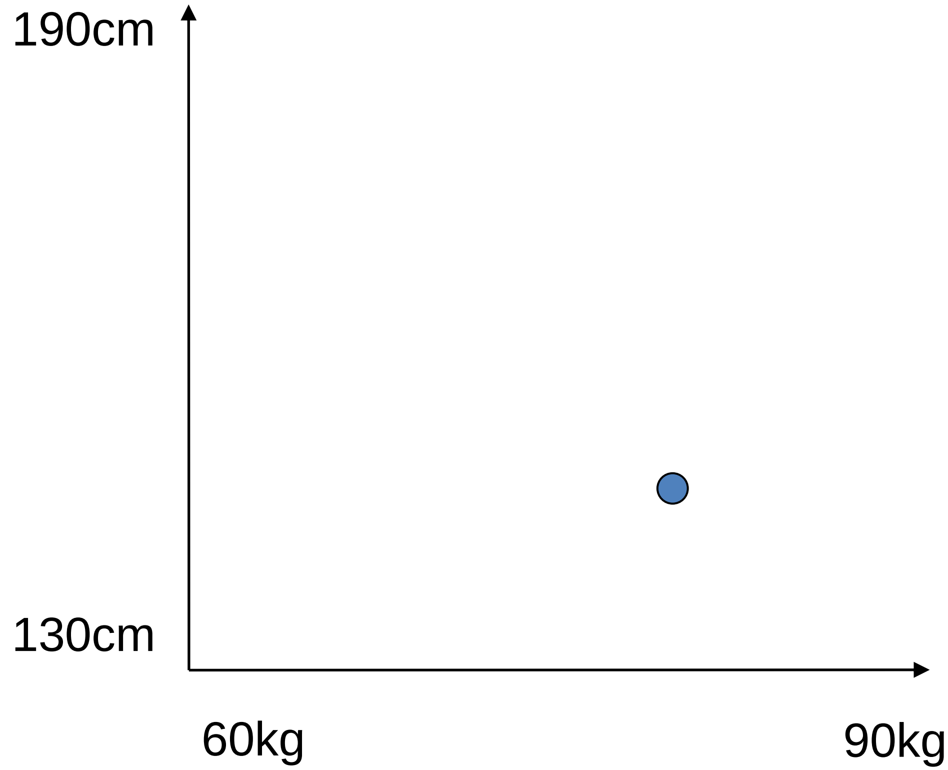
Problem Statement

Can we LEARN to recognise a rugby player?

What are the “features” of a rugby player?

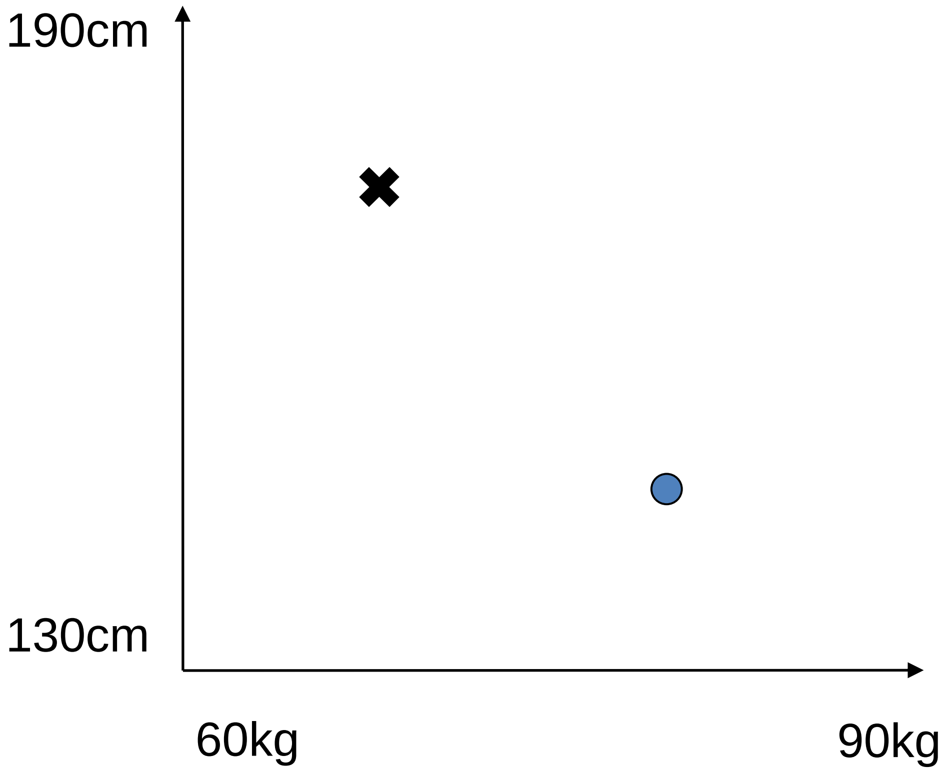
Features

Rugby players = short + heavy?



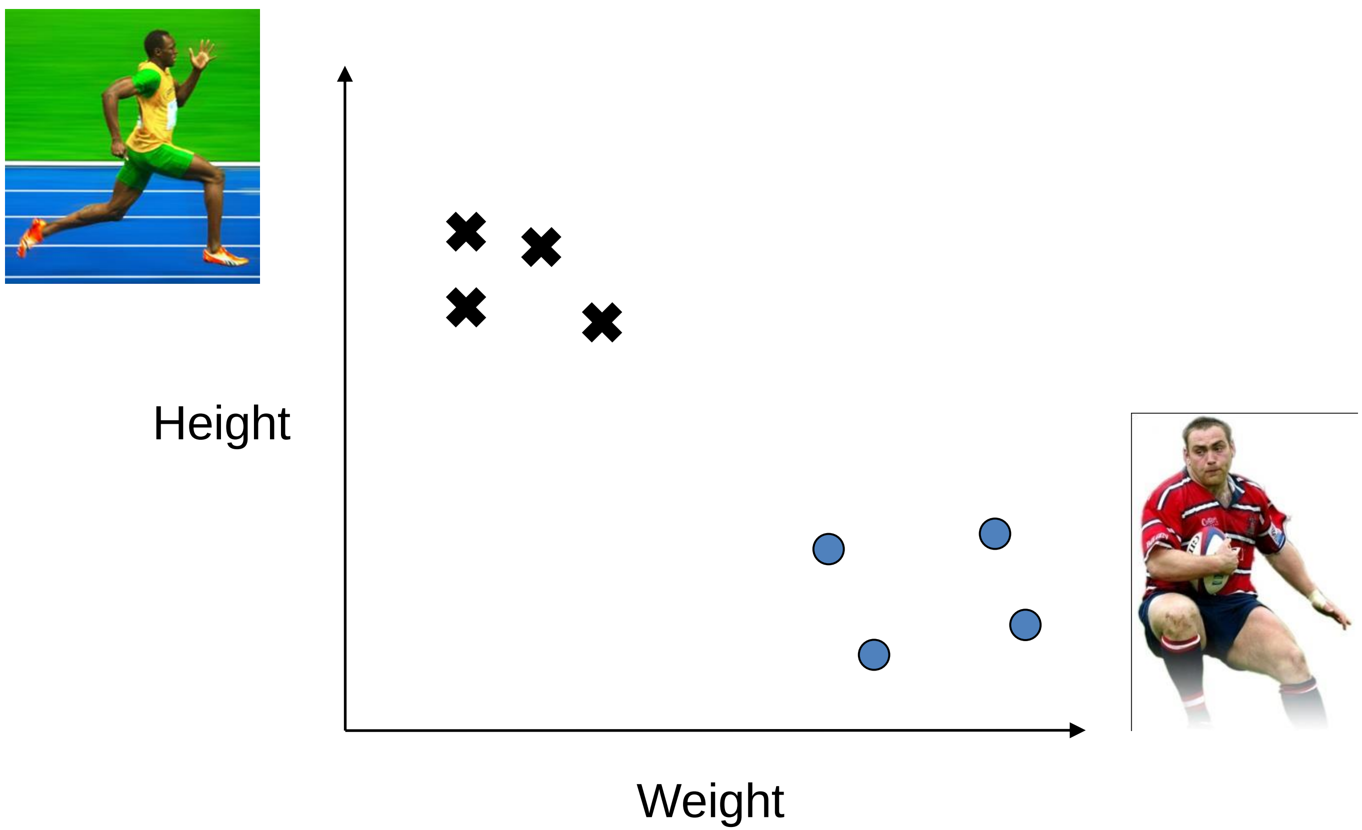
Features

Sprinters = tall + light?

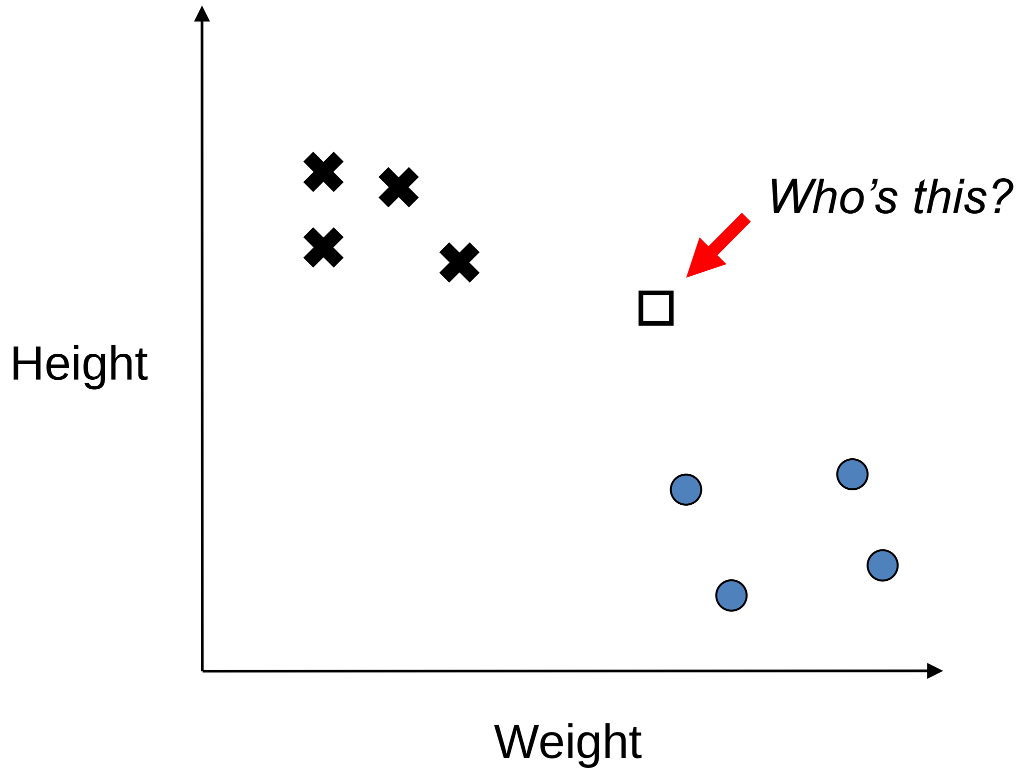


Feature Space

Rugby players “cluster” separately in the space.

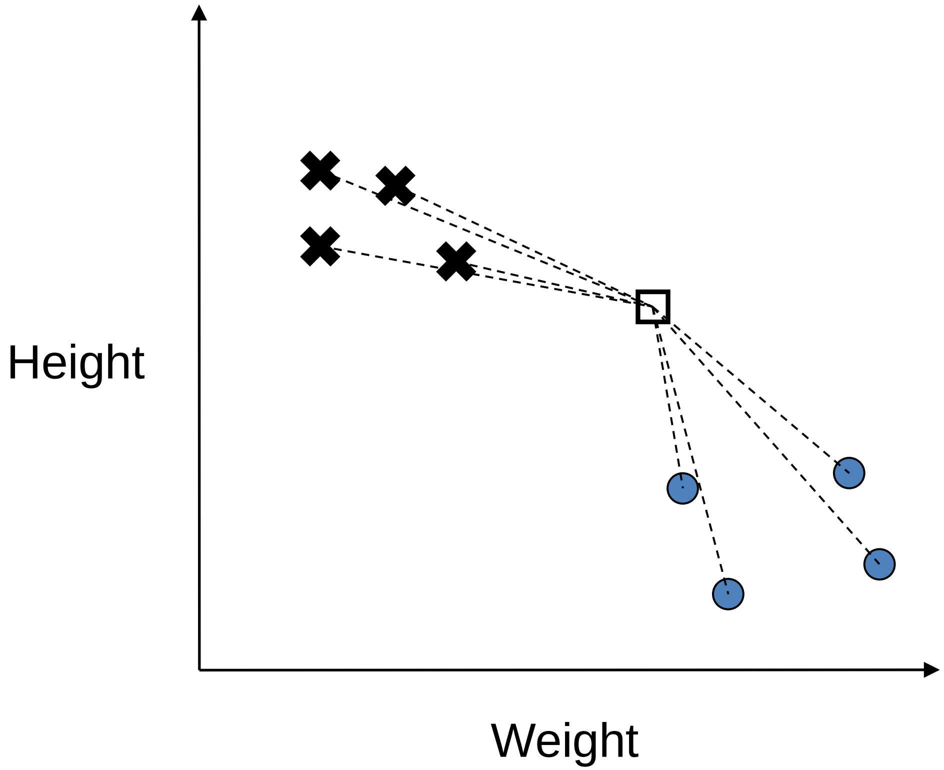


K-NN Algorithm



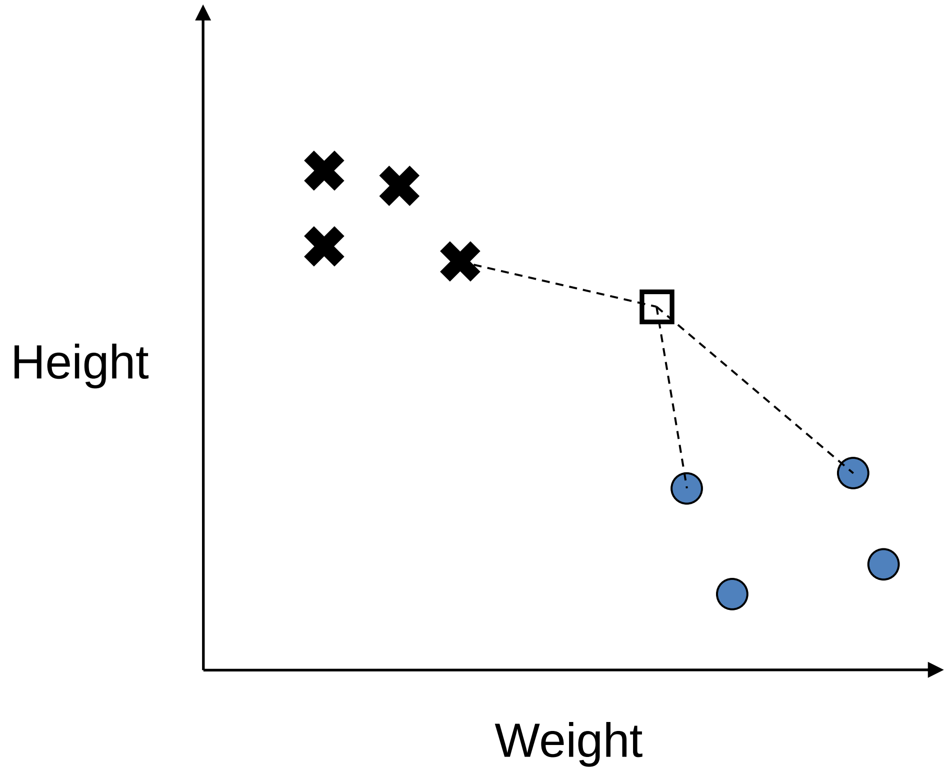
K-NN Algorithm

1. *Measure distance to all points*



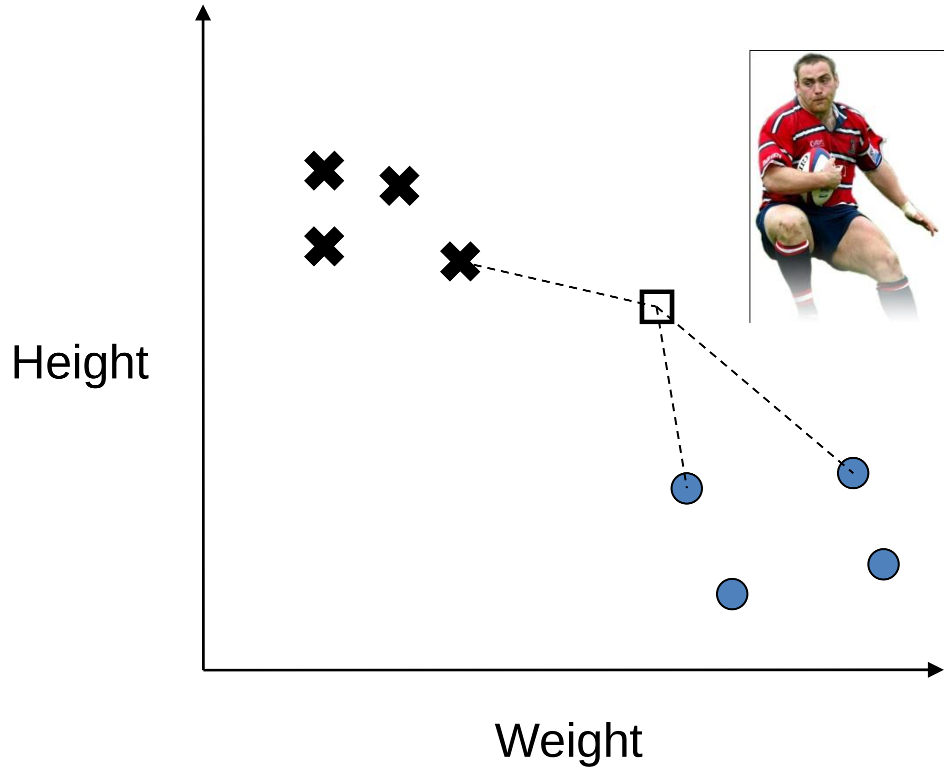
K-NN Algorithm

1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it could be more)**



K-NN Algorithm

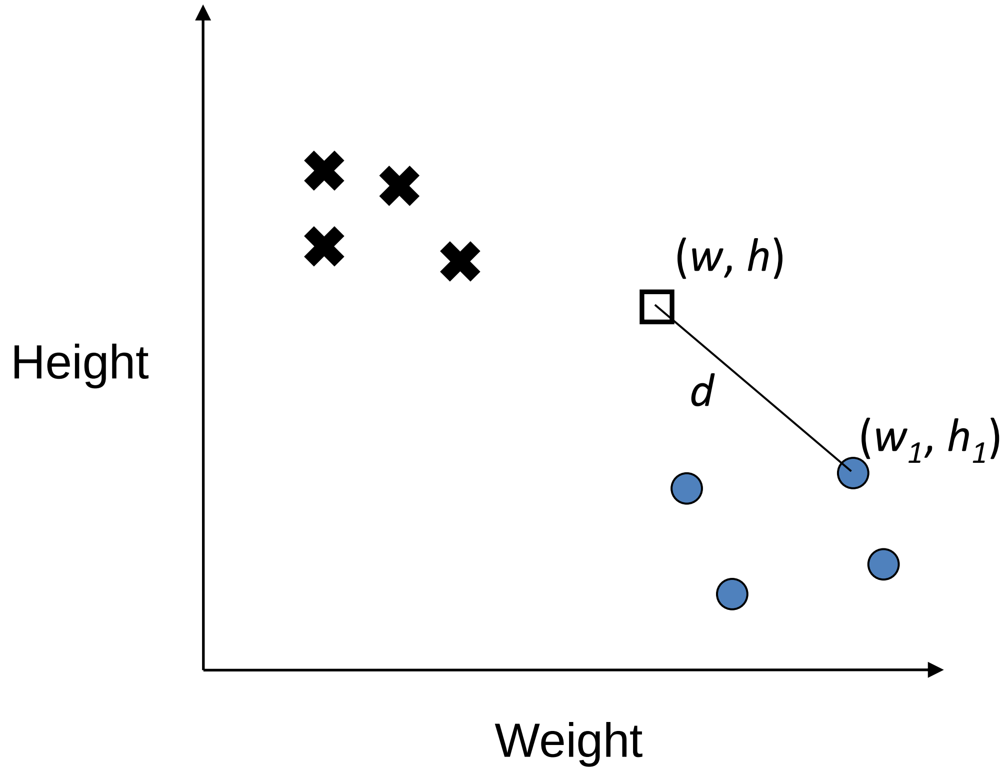
1. *Measure distance to all points*
2. *Find closest "k" points* ← **(here k=3, but it can be more)**
3. *Assign majority class*



K-NN Algorithm

"Euclidean distance"

$$d = \sqrt{(w - w_1)^2 + (h - h_1)^2}$$



K-NN Algorithm

for each testing point

    measure distance to every training point

    find the k closest points

    identify the most common class among those k

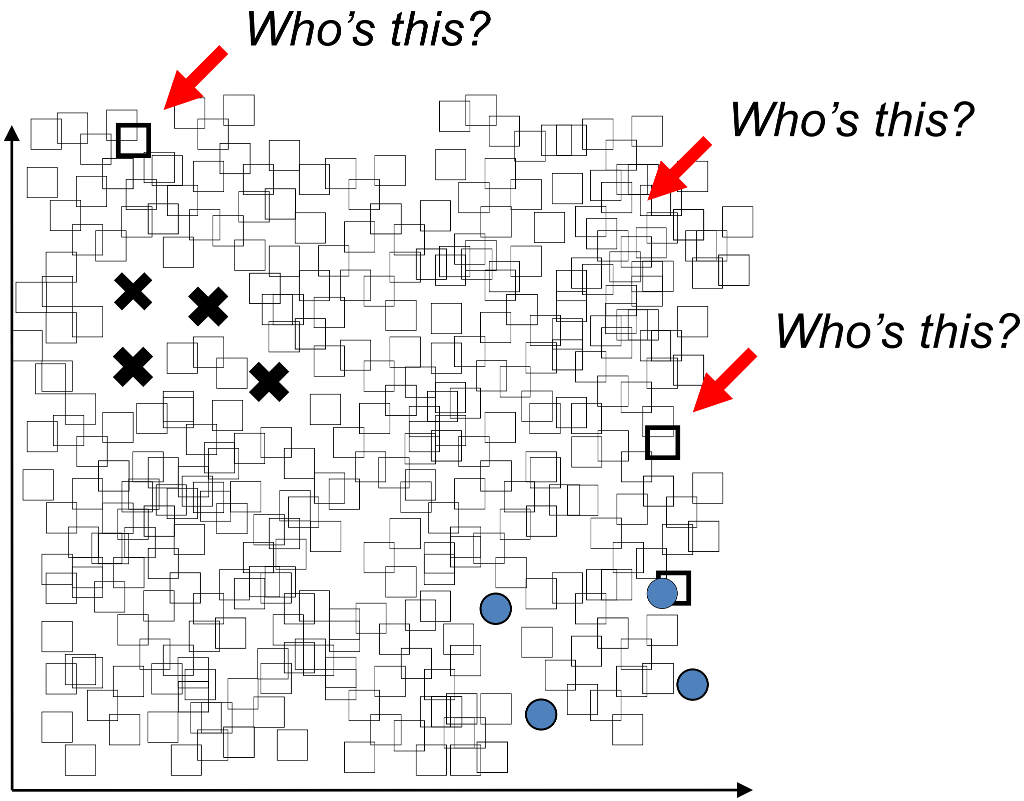
    predict that class

end

- Advantage: Surprisingly good classifier!
- Disadvantage: Have to store the entire training set in memory

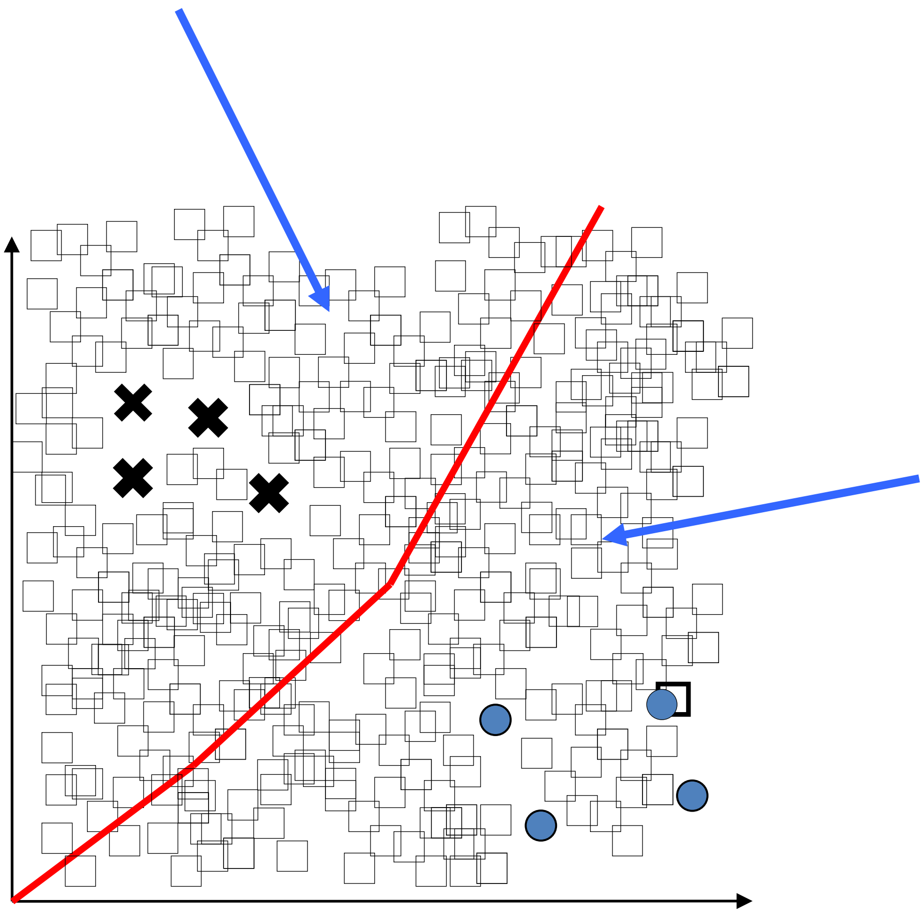
Decision Boundary

Imagine testing loads of points….



Decision Boundary

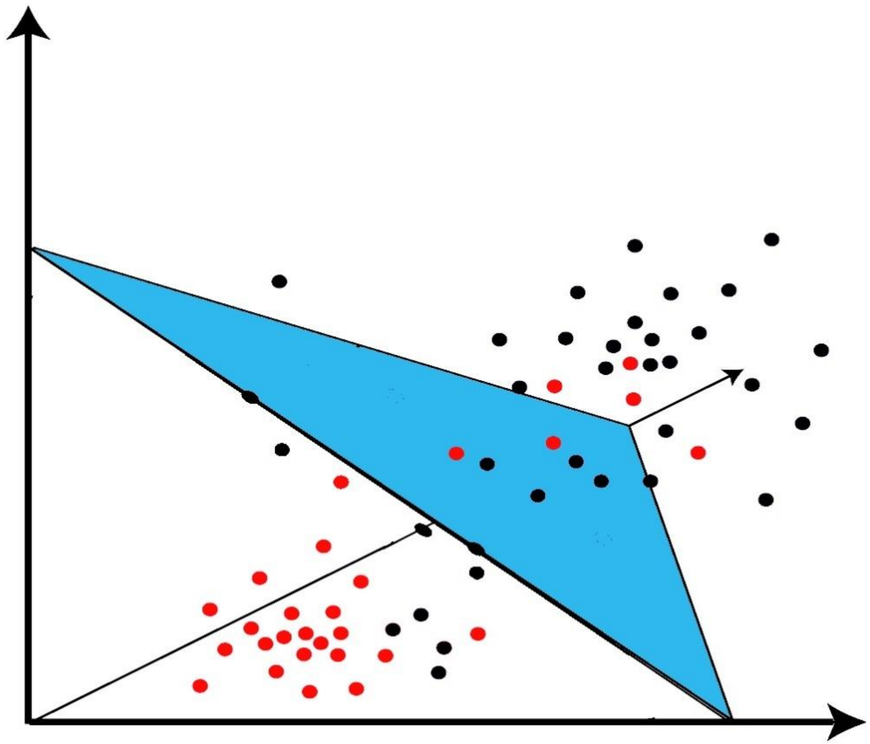
These “hypothetical” points
are closest to the sprinters



These “hypothetical” points
are closest to the rugby players

Decision Boundary

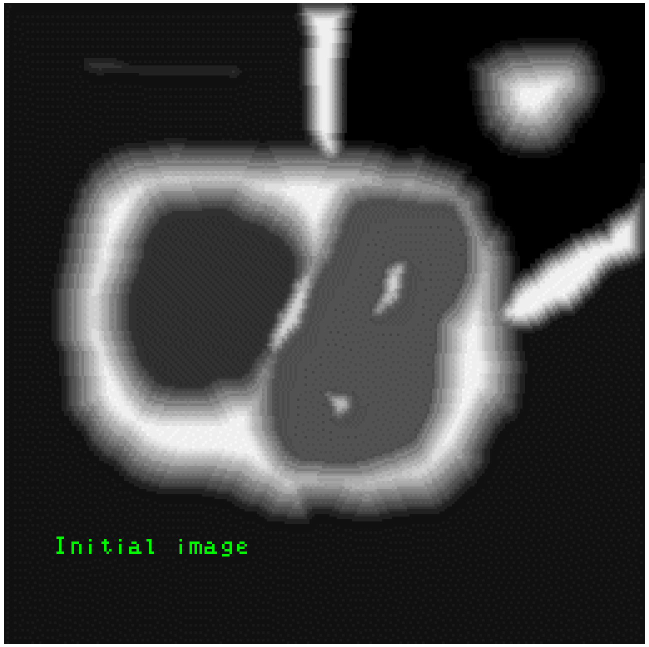
Called a decision **surface** in 3-d and upward…

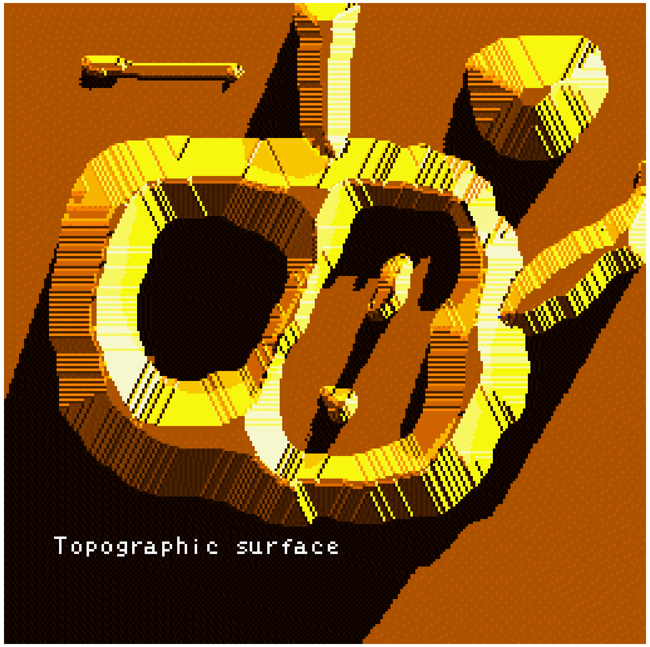


Watershed Segmentation

6. Watershed Segmentation

- Image is visualized in 3-dimensions
  - 2 spatial dimensions
  - grey levels
- Any grey tone image can be considered as a topological surface



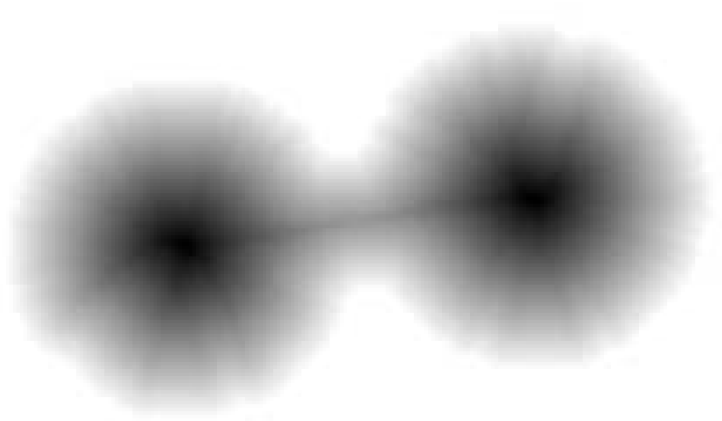


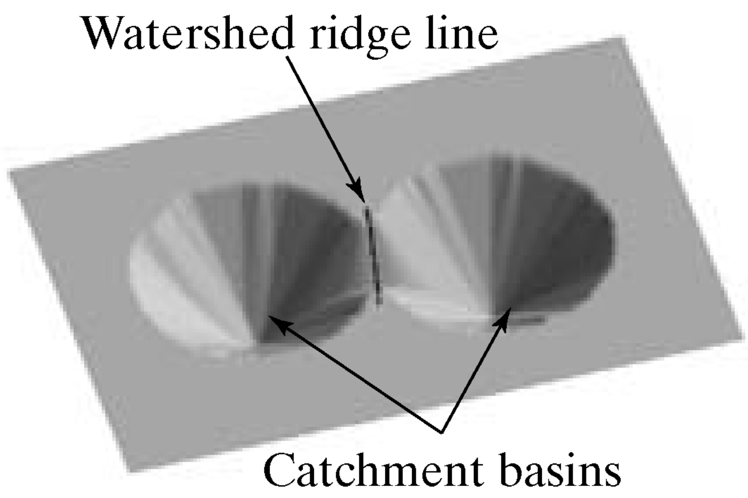
6. Watershed Segmentation

Three types of points

- Points belonging to a regional minimum
- Catchment basin / watershed of a regional minimum
  - Points at which a drop of water will certainly fall to a single minimum
- Divide lines / Watershed lines
  - Points at which a drop of water will be equally likely to fall to more than one minimum
  - Crest lines on the topographic surface

This technique is to identify all the third type of points for segmentation





6. Watershed Segmentation - Visualization

NUST

- Punch the regional minimum and flood the entire topography at uniform rate from below
- A dam is built to prevent the rising water from distinct catchment basins from merging
- Eventually only the tops of the dams are visible above the water line
- These dam boundaries correspond to the divide lines of the watersheds

6. Watershed Segmentation

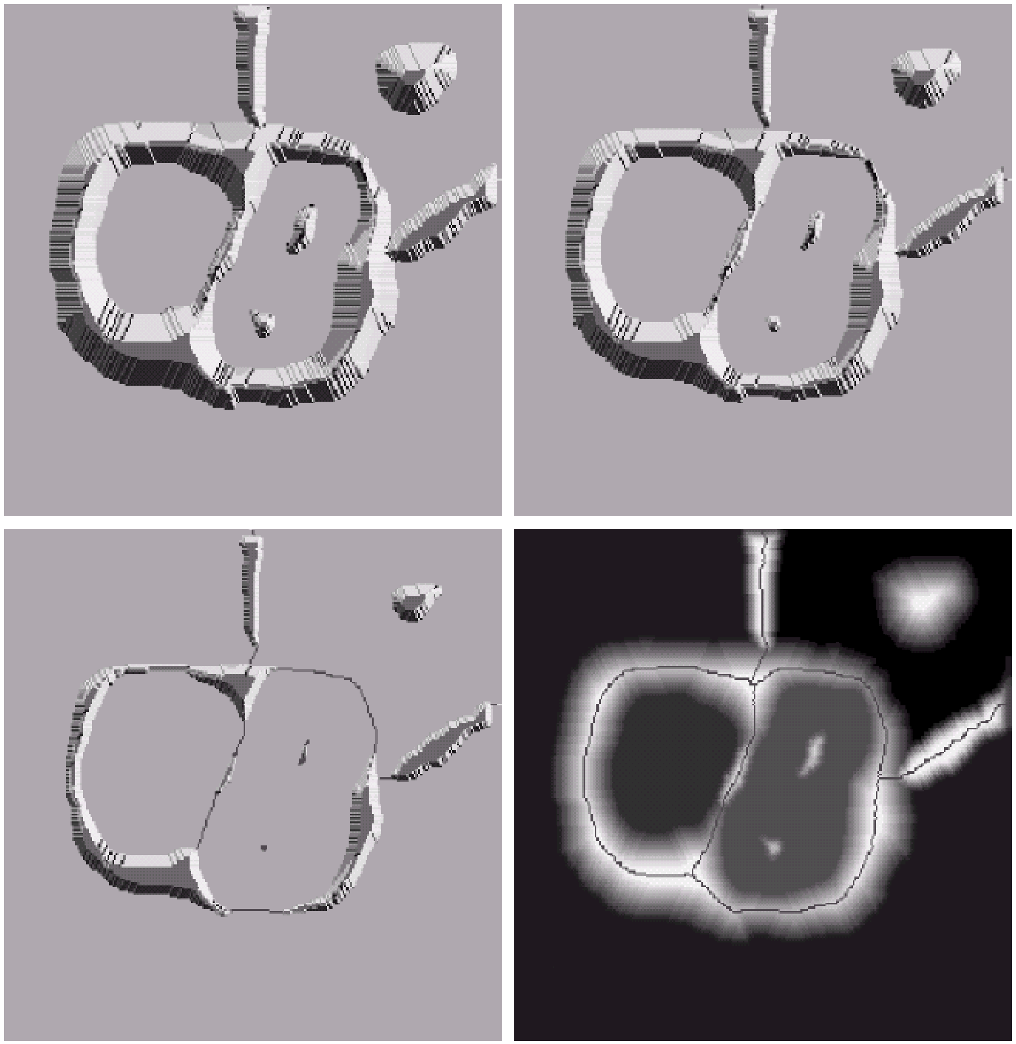
Basic Steps

1. Piercing holes in each regional minimum of I

2. The 3D topography is flooded from below gradually

3. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging

6. Watershed Segmentation

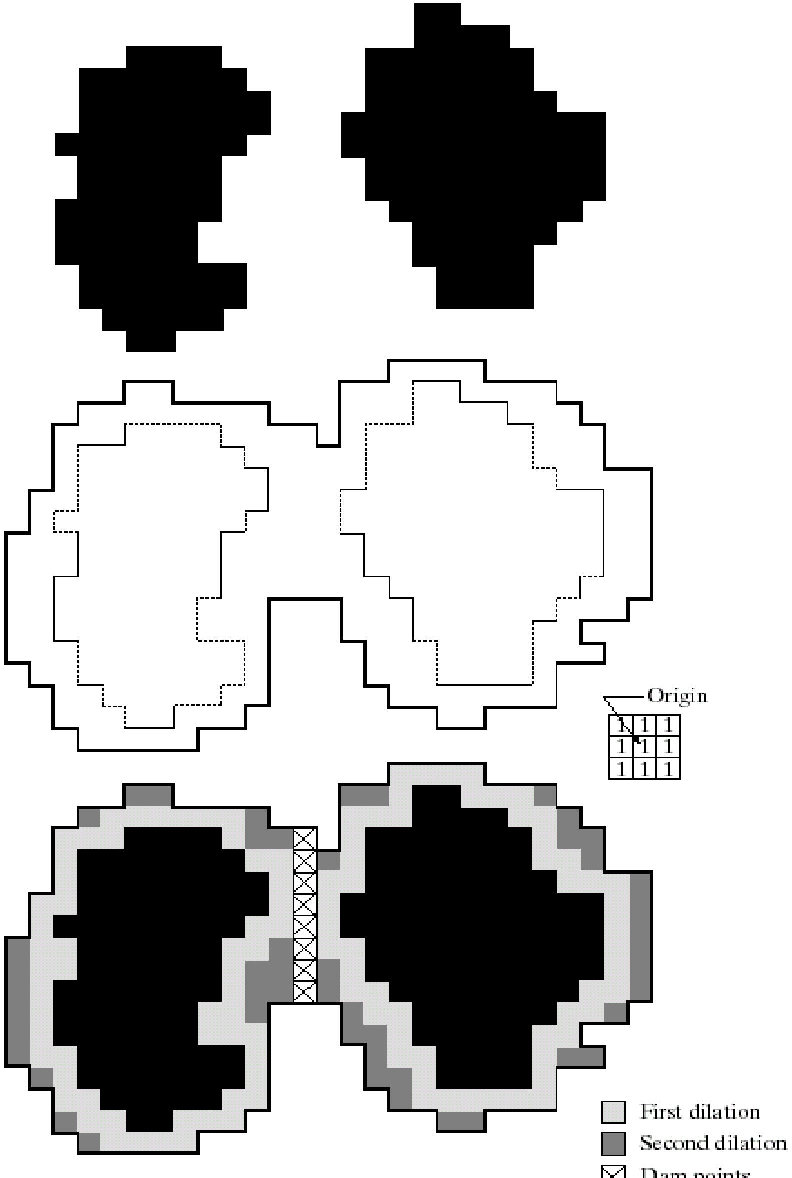


e f
g h

FIGURE 10.44 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

4. The dam boundaries correspond to the watershed lines to be extracted by a watershed segmentation algorithm
   - Eventually only constructed dams can be seen from above

6. Watershed Segmentation



- M1, M2:
  - Sets of coordinates of points in the two regional minima

- $C_{n-1}(M_1)$, $C_{n-1}(M_2)$
  - Sets of coordinates of points in the catchment basins associated with M1 M2 at stage n-1 of flooding (catchment basins up to the flooding level)

- C[n-1]
  - Union of $C_{n-1}(M_1)$, $C_{n-1}(M_2)$

6. Watershed Segmentation

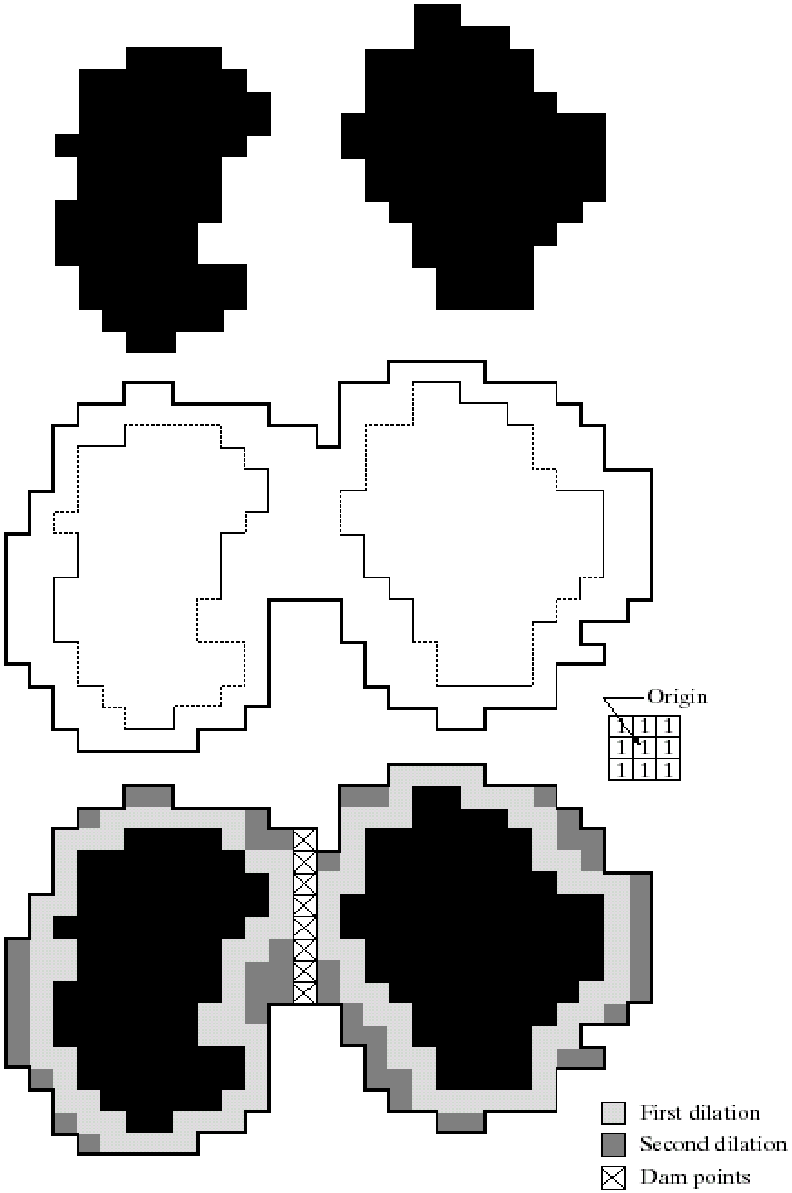
- Based on the morphological dilation

- At each step of the algorithm, the binary image in obtained in the following manner

    1. Initially, the set of pixels with minimum gray level are 1, others 0.

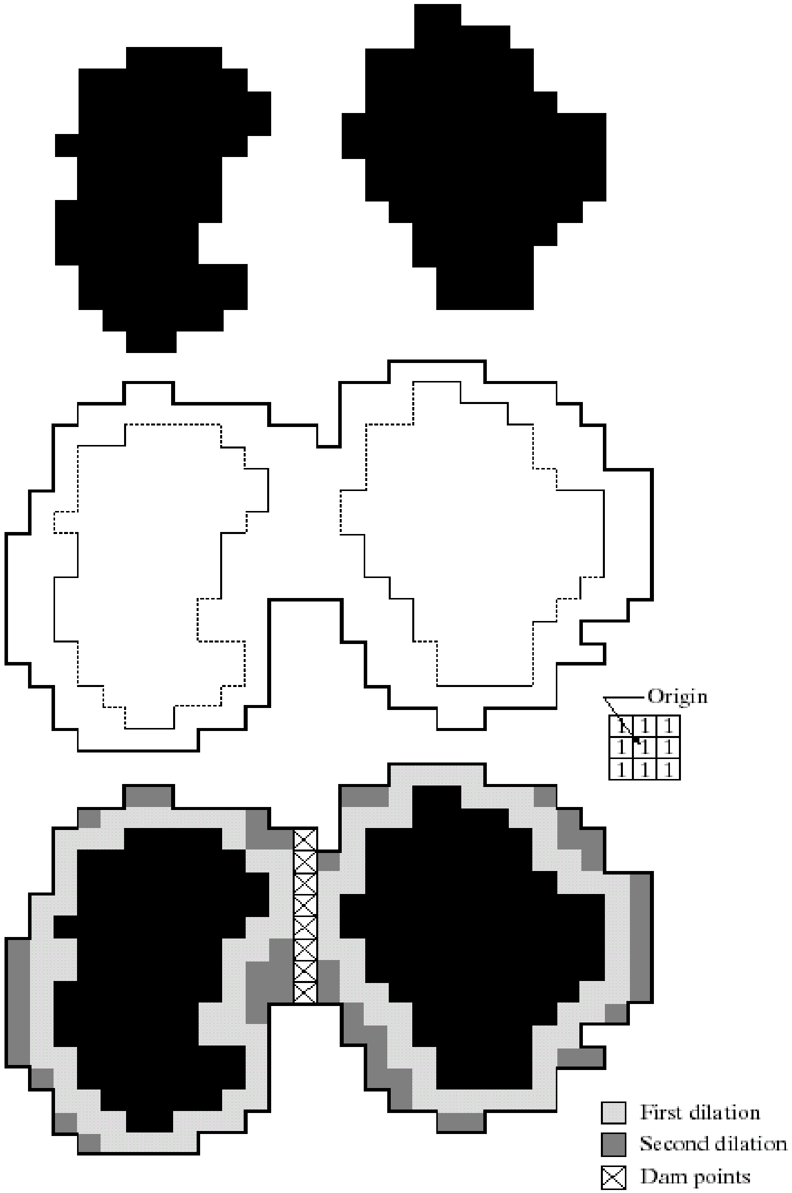
    2. In each subsequent step, we flood the 3D topography from below and the pixels covered by the rising water are 1s and others 0s.

6. Watershed Segmentation



- At flooding step n-1, there are two connected components. At flooding step n, there is only one connected component
  - This indicates that the water between the two catchment basins has merged at flooding step n
  - Use “q” to denote the single connected component
- Steps
  - Repeatedly dilate $C_{n-1}(M_1)$, $C_{n-1}(M_2)$ by the 3×3 structuring element shown, subject to the following condition
    - Constrained to q (center of the structuring element can not go beyond q during dilation

6. Watershed Segmentation



- The dam is constructed by the points on which the dilation would cause the sets being dilated to merge.
  - Resulting one-pixel thick connected path

- Setting the gray level at each point in the resultant path to a value greater than the maximum gray value of the image. Usually max+1

6. Watershed Segmentation

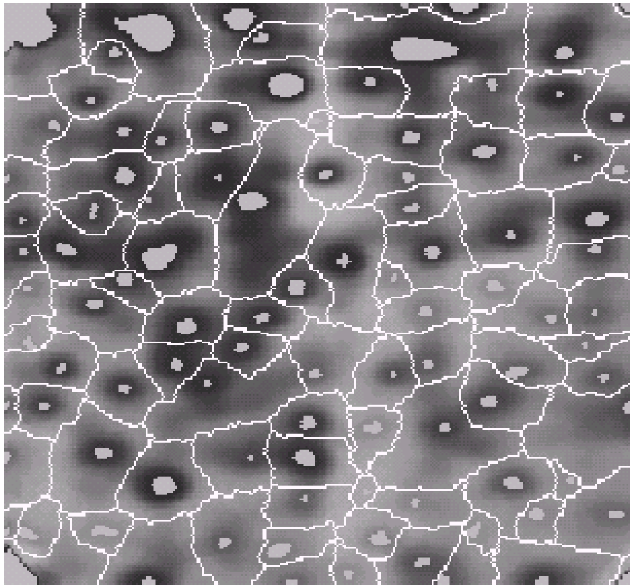
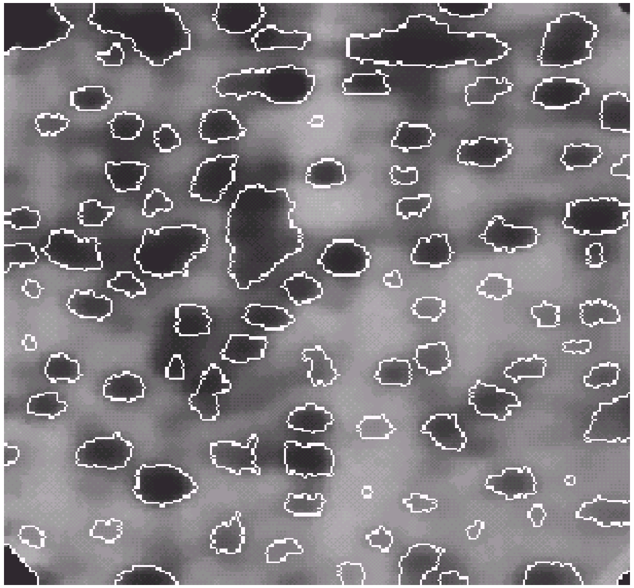
- Markers are used to limit the number of regions by specifying the objects of interest

  - Like seeds in region growing method
  - Can be assigned manually or automatically
  - Regions without markers are allowed to be merged (no dam is to be built)

6. Watershed Segmentation

1. Use markers to obtain watershed lines of the gradient of the image to be segmented.
2. Use the obtained watershed lines as external markers
3. Each region defined by the external markers contains a single internal marker and part of the background
4. The problem is reduced to partitioning each region into two parts: object (containing internal markers) and a single background (containing external markers)

   - local thresholding, region growing, region splitting and merging

End

Image Segmentation