Segmentation

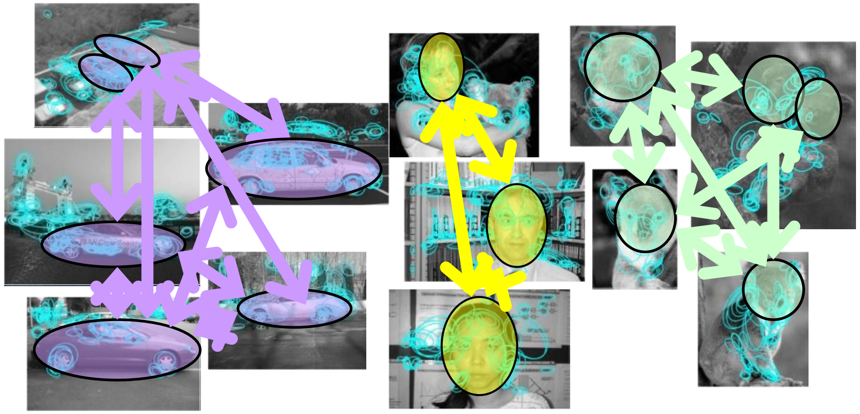
Segmentation

Introduction

Goals of Segmentation

- Gather features that belong together

Determine image regions



Object-level grouping

Slide credit: Kristen Grauman

Goals of Segmentation

- Separate image into coherent “objects”

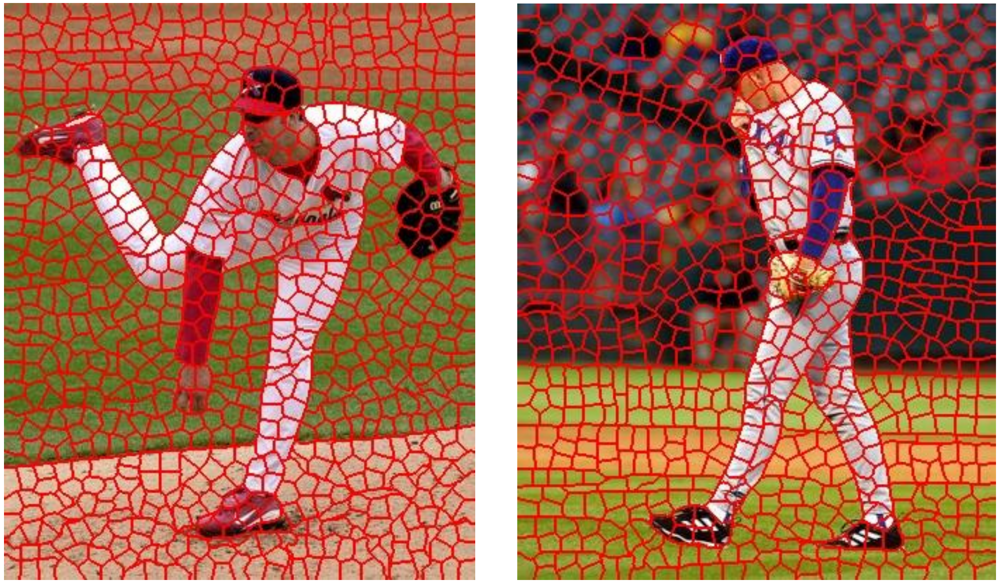
image human segmentation

Source: Lana Lazebnik

Goals of Segmentation

- Group together similar-looking pixels for efficiency of further processing

“superpixels”



X. Ren and J. Malik. **Learning a classification model for segmentation.** ICCV 2003.

Source: Lana Lazebnik

Top down vs Bottom up

- Top down
  - pixels belong together because they are from the same object

- Bottom up
  - pixels belong together because they look similar

Gestalt

- Gestalt: whole or group
  - Whole is greater than sum of its parts
  - Relationships among parts can yield new properties/features

- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

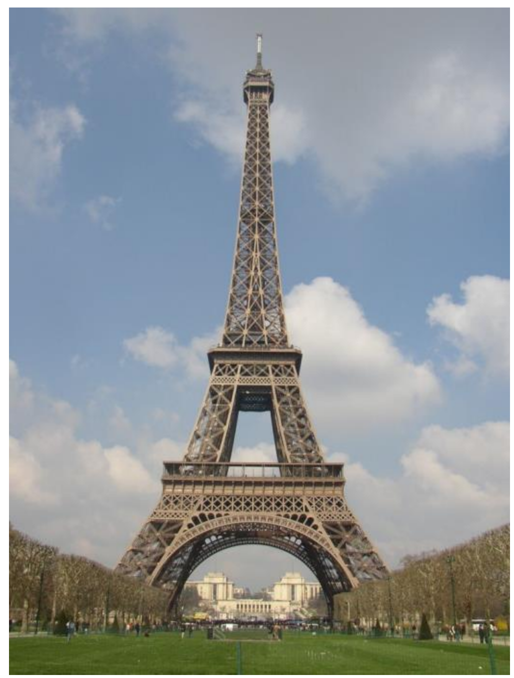
Slide credit: Kristen Grauman

Similarity

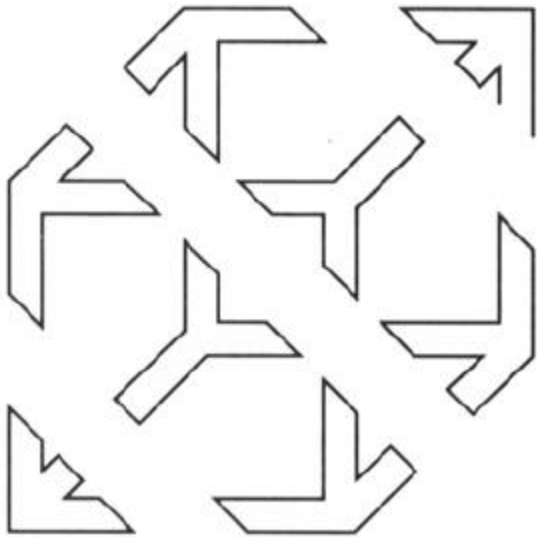
Slide credit: Kristen Grauman

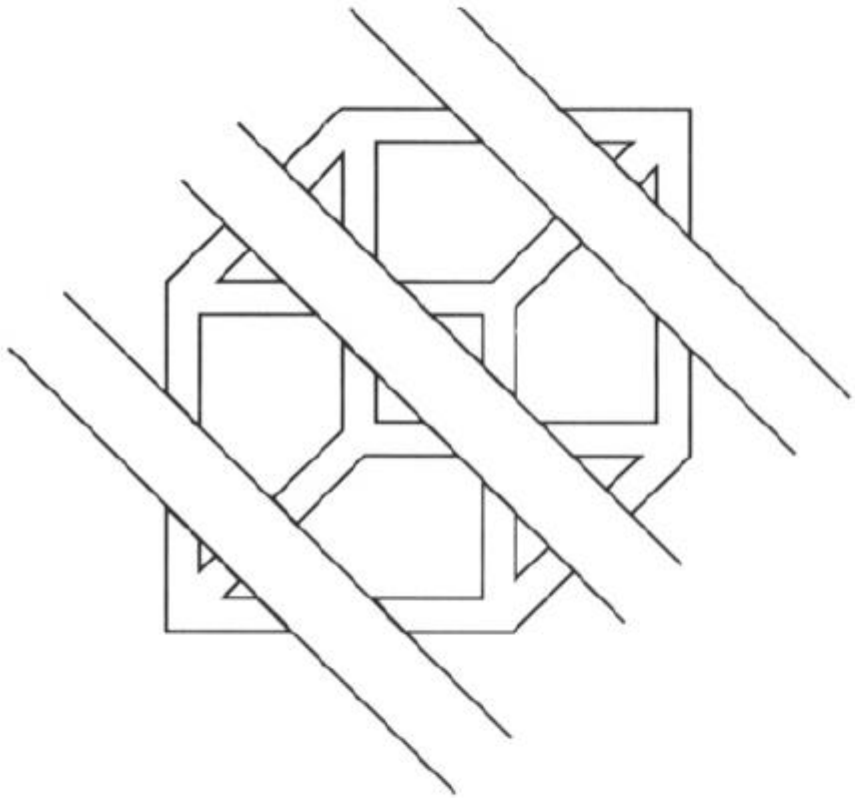
http://chicagoist.com/attachments/chicagoist_alicia/GEESE.jpg, http://[illegible]superstock.com/WI/223/1532/PreviewComp/SuperStock_1532R-0831.jpg

Symmetry

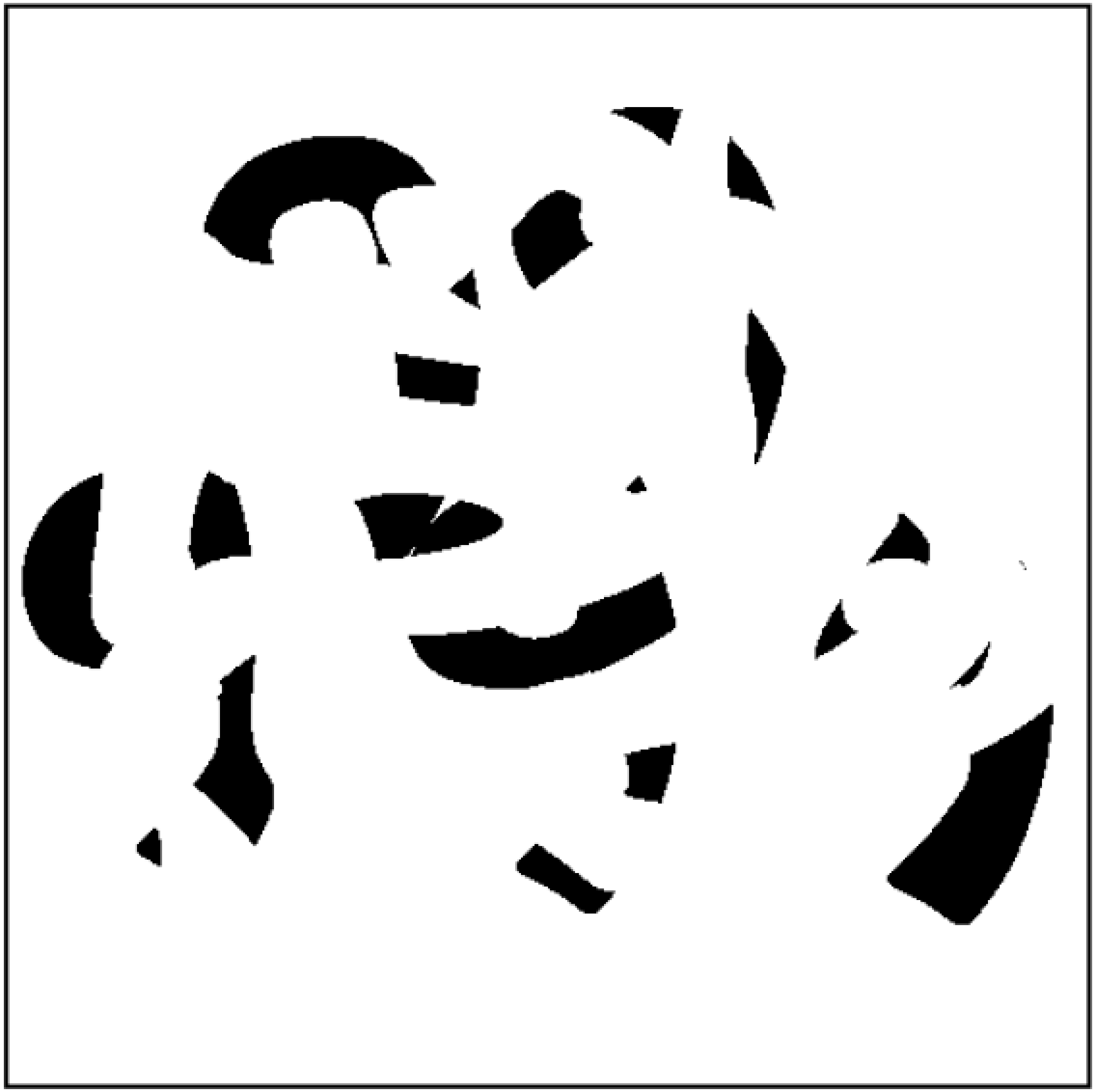


Slide credit: Kristen Grauman





Continuity, explanation by occlusion



Continuity, explanation by occlusion

Segmentation

Thresholding

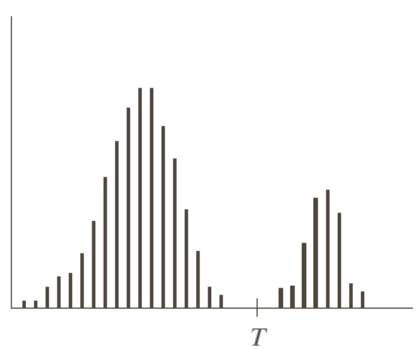
Thresholding

- What is thresholding?
- Because of its intuitive properties, simplicity of implementations, and computational speed, image thresholding enjoys a central position in application of image segmentation.

Basics of Intensity Thresholding

- Any point (x,y) in the image at which f(x,y)>T is called an object point: otherwise, the point is called a background point. (Assumption: Light object, Dark background)

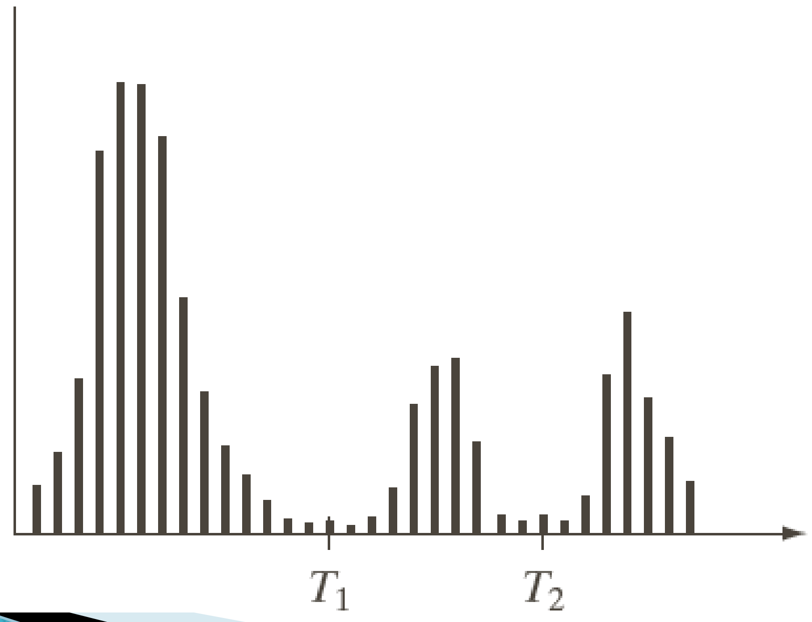
- Global thresholding vs Dynamic (Adpative) Thresholding



Basics of Intensity Thresholding

- Multiple thresholding

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



Role of Noise in Image Thresholding

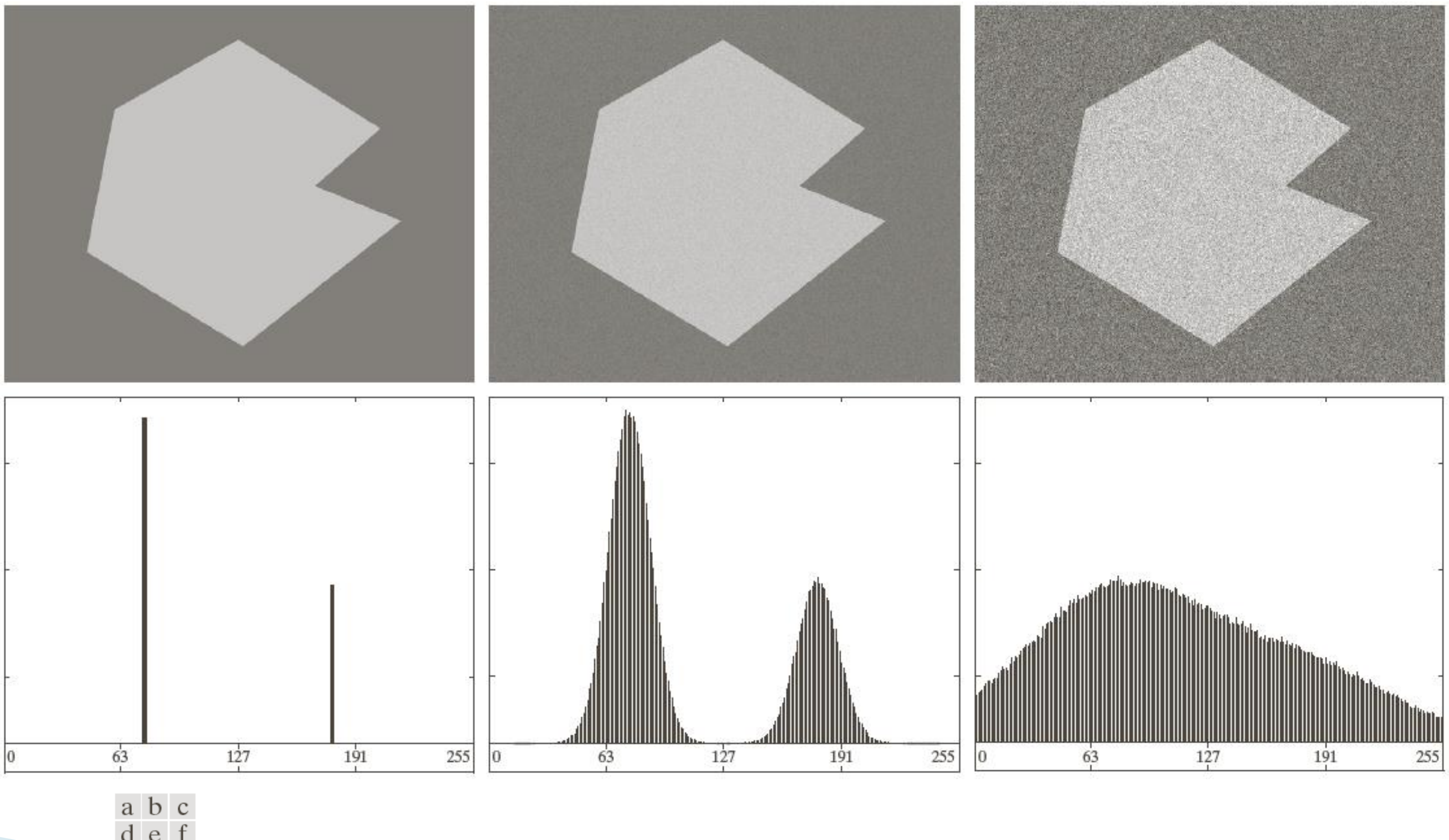
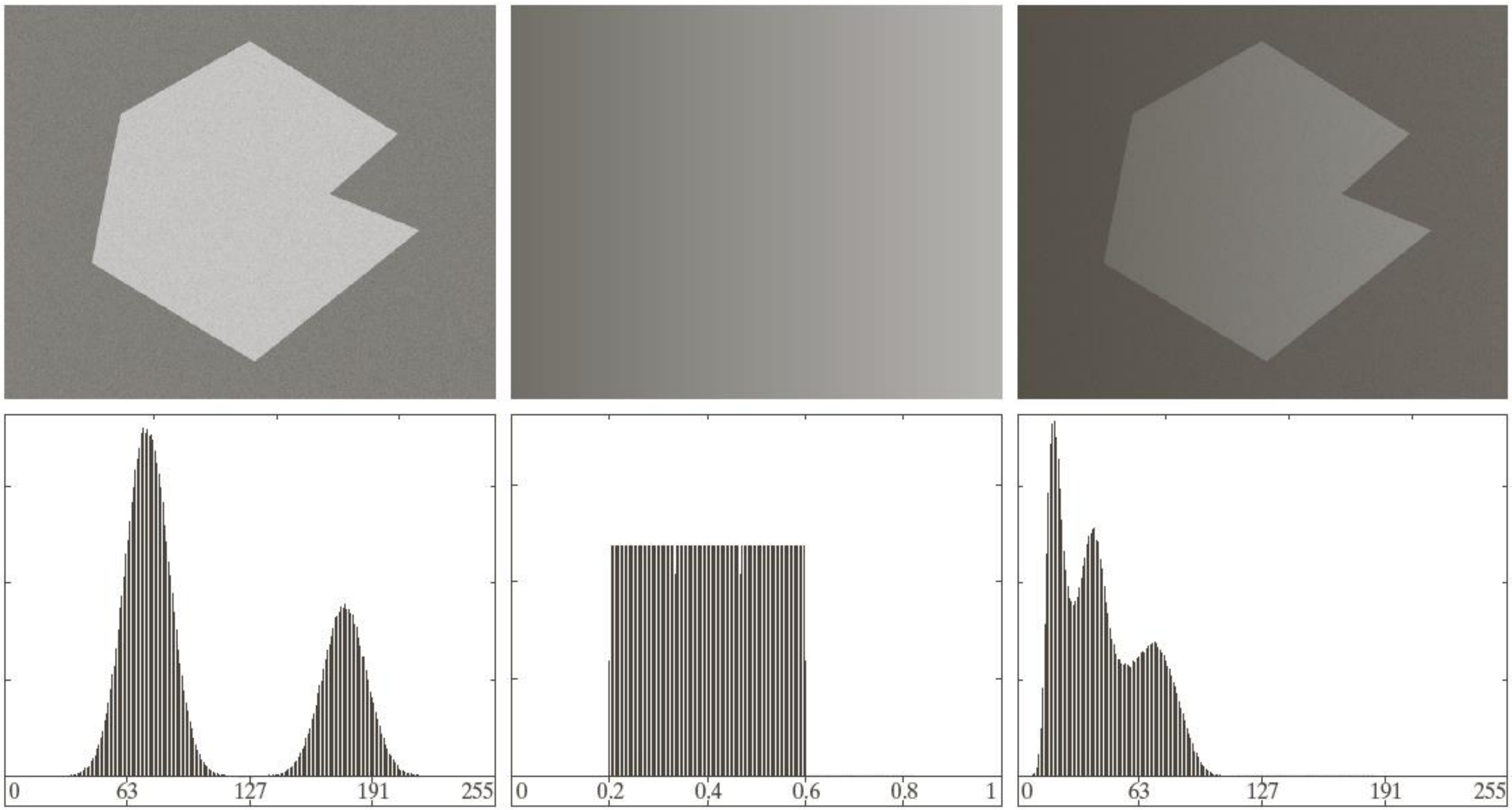


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Role of Illumination and Reflectance



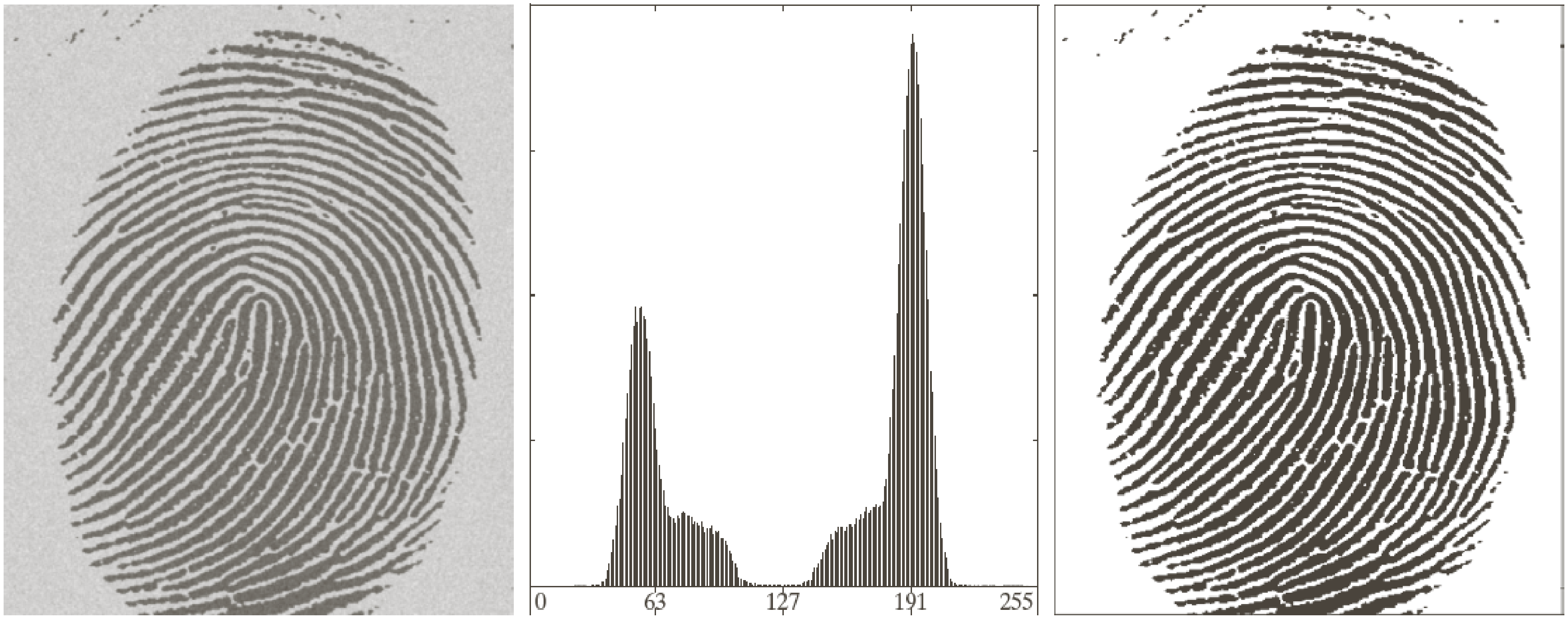
a	b	c
d	e	f

FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Basic Global Thresholding

- Select an initial estimate for T.
- Segment the image using T to generate 2 regions, G_1 & G_2.
- Compute the average gray level values m_1 and m_2 for regions G_1 and G_2, respectively.
- Compute the new T as $(m_1+m_2)/2$.
- Repeat the steps 2 ~ 4 until the difference between values of T in successive iterations is smaller than a predefined parameter.

Example



a b c

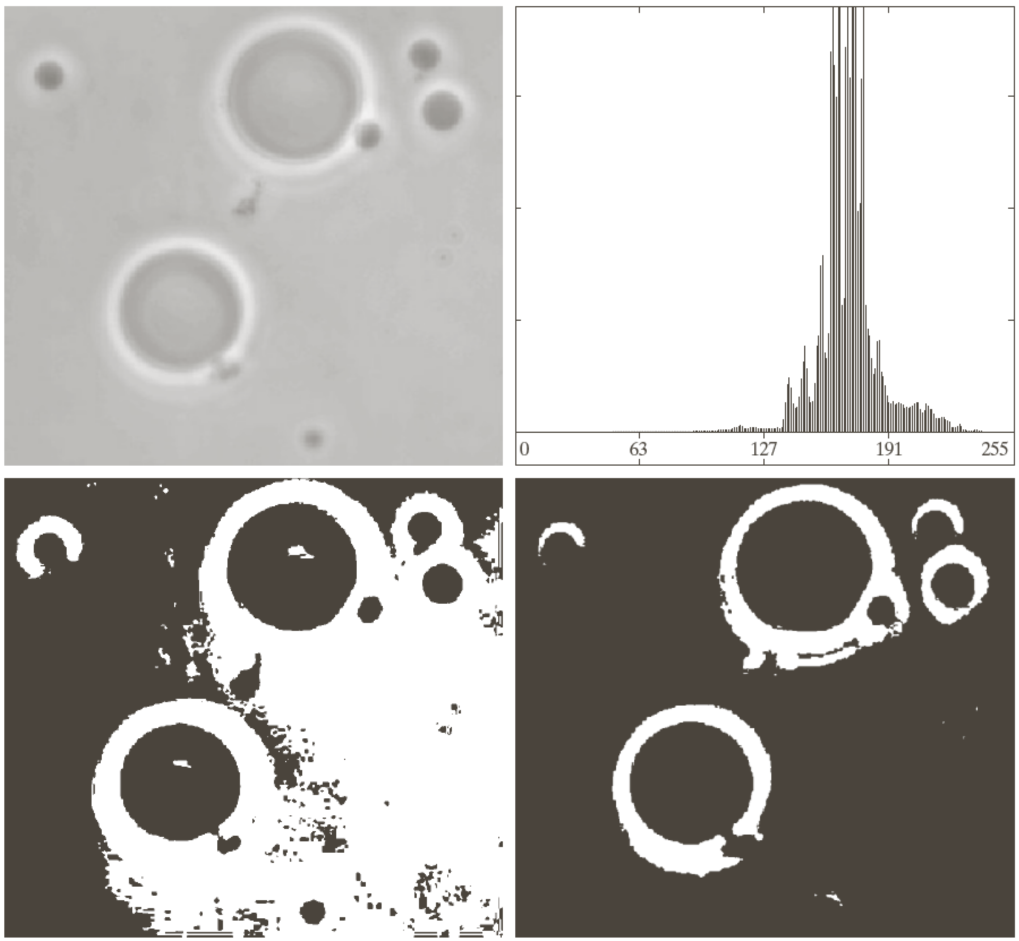
FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Otsu's Method

- Optimal in the sense that it maximizes the between-class variance.

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

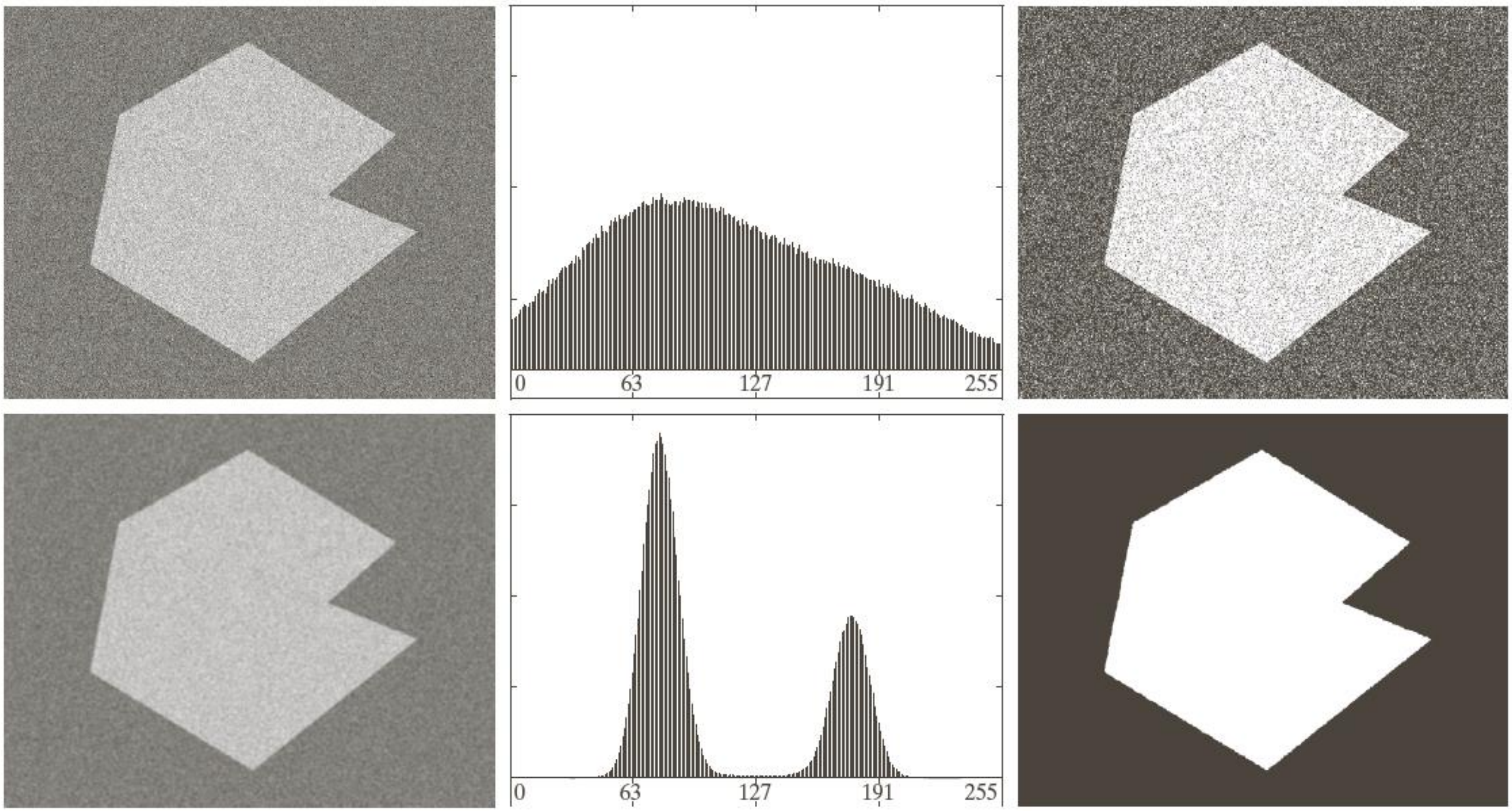
  - P_1: Probability of class 1 for threshold k
  - P_2: Probability of class 2 for threshold k
  - m_1: Mean of class 1 for threshold k
  - m_2: Mean of class 2 for threshold k
  - m_G: global mean



a b
c d

FIGURE 10.39
(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.2.
(d) Result obtained using Otsu's method.
(Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Using Image Smoothing to Improve Global Thresholding



a	b	c
d	e	f

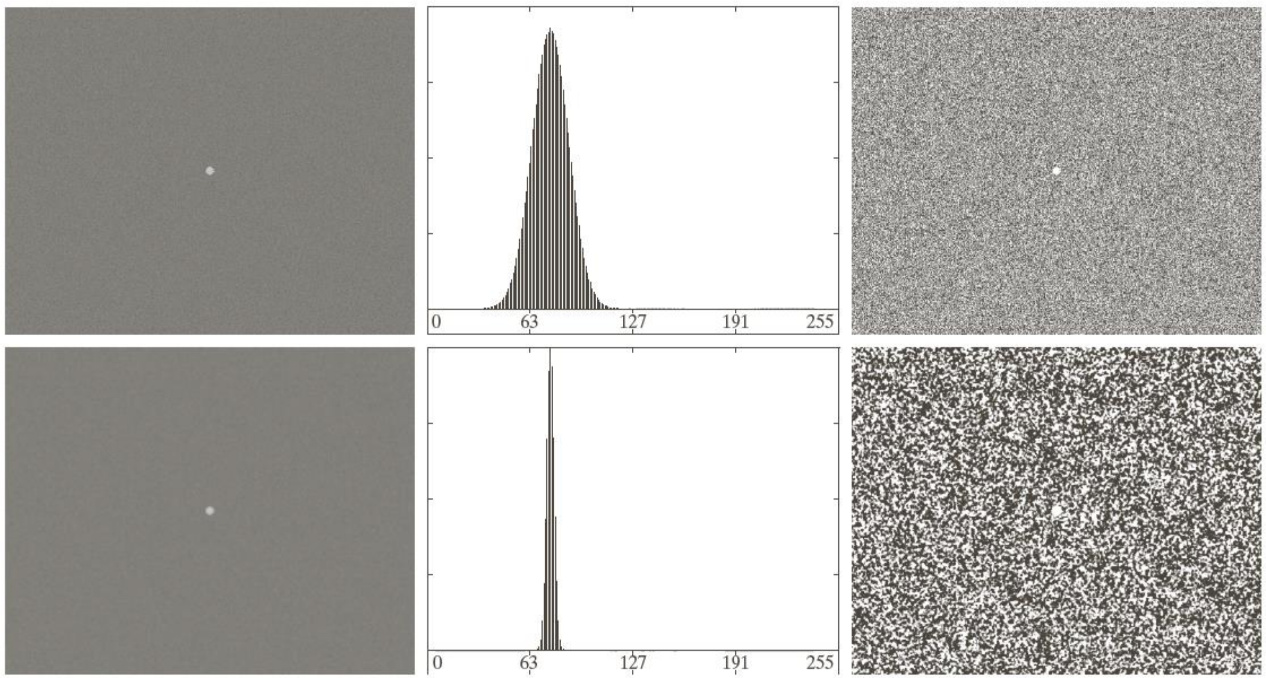
FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Using Edges to Improve Global Thresholding

- Histogram of an image composed of a small object on a large background area would be dominated by a large peak.

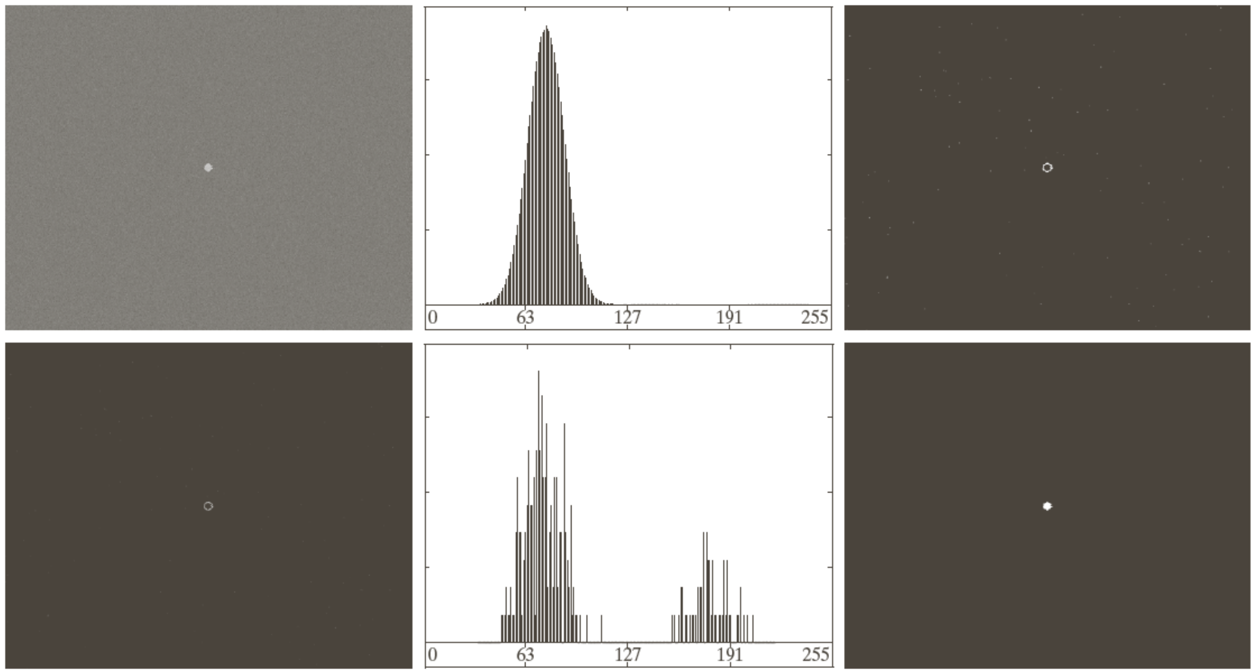
- One approach for improving the shape of histograms is to consider only those pixels that lie on or near the edges between objects and the backgorund

Example



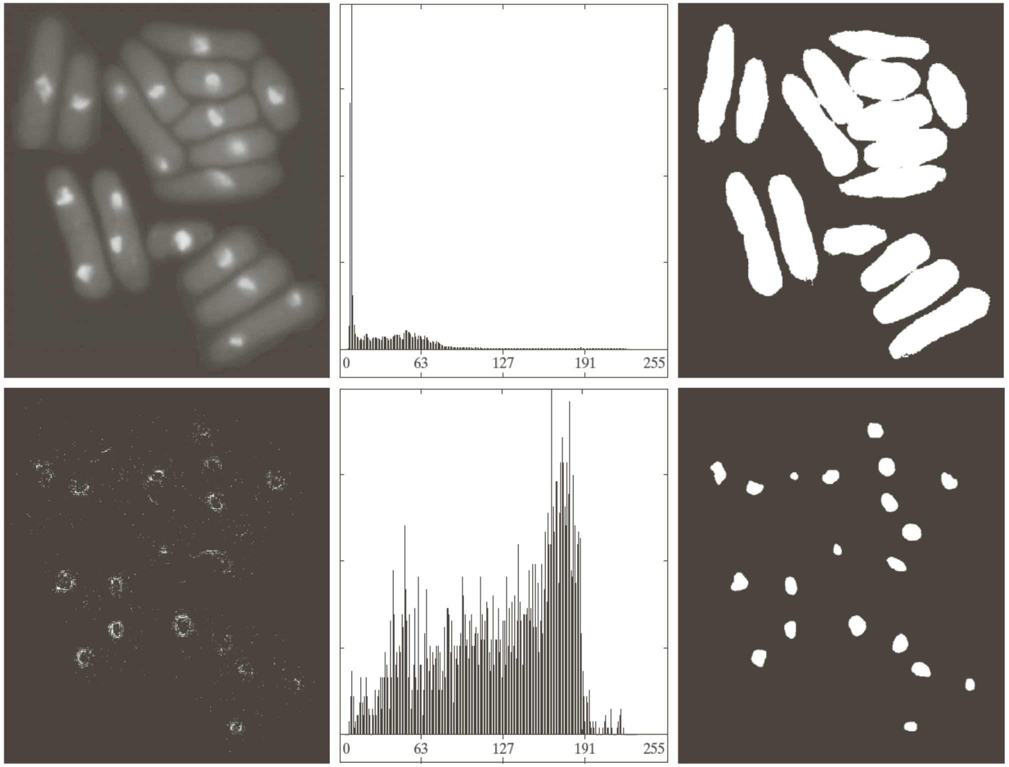
a b c
d e f

FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.



a	b	c
d	e	f

FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.



a b c
d e f

FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

FIGURE 10.44 Image in Fig. 10.43(a) segmented using the same procedure as explained in Figs. 10.43(d)–(f), but using a lower value to threshold the absolute Laplacian image.

Multiple Thresholds

$$\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2$$

P_k: Probability of class k for the given threshold
m_k: Mean of class k for the given threshold
m_G: global mean

Segmentation

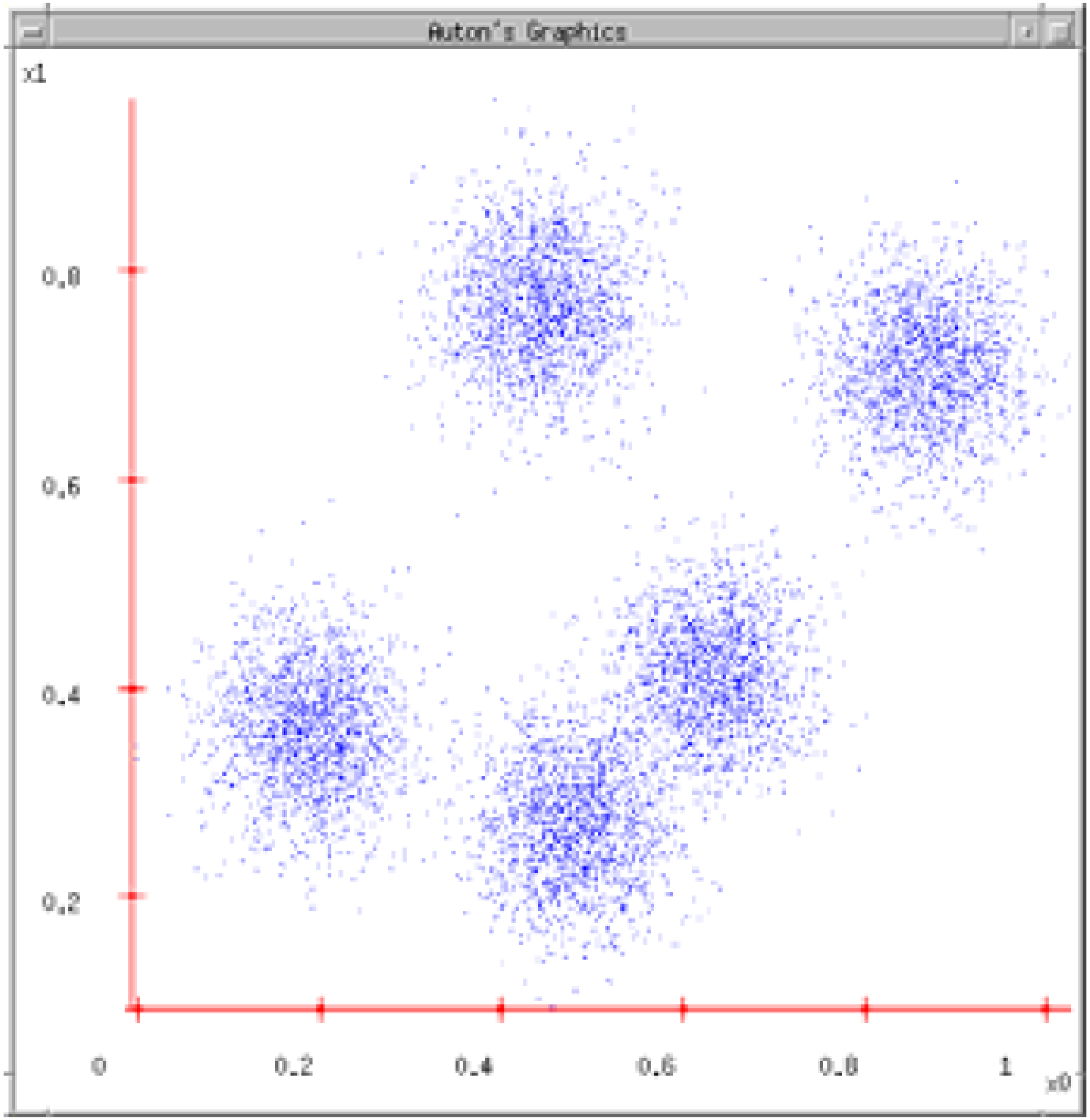
K-Means Clustering

K-Means clustering

- Step 1:
  Randomly initialize the cluster centers, c_1, ..., c_K
- Step 2:
  For each point p, find the closest c_i. Put p into cluster *i*.
- Step 3:
  Set c_i to be the mean of points in cluster I
- If c_i have changed, repeat Step 2

K-means

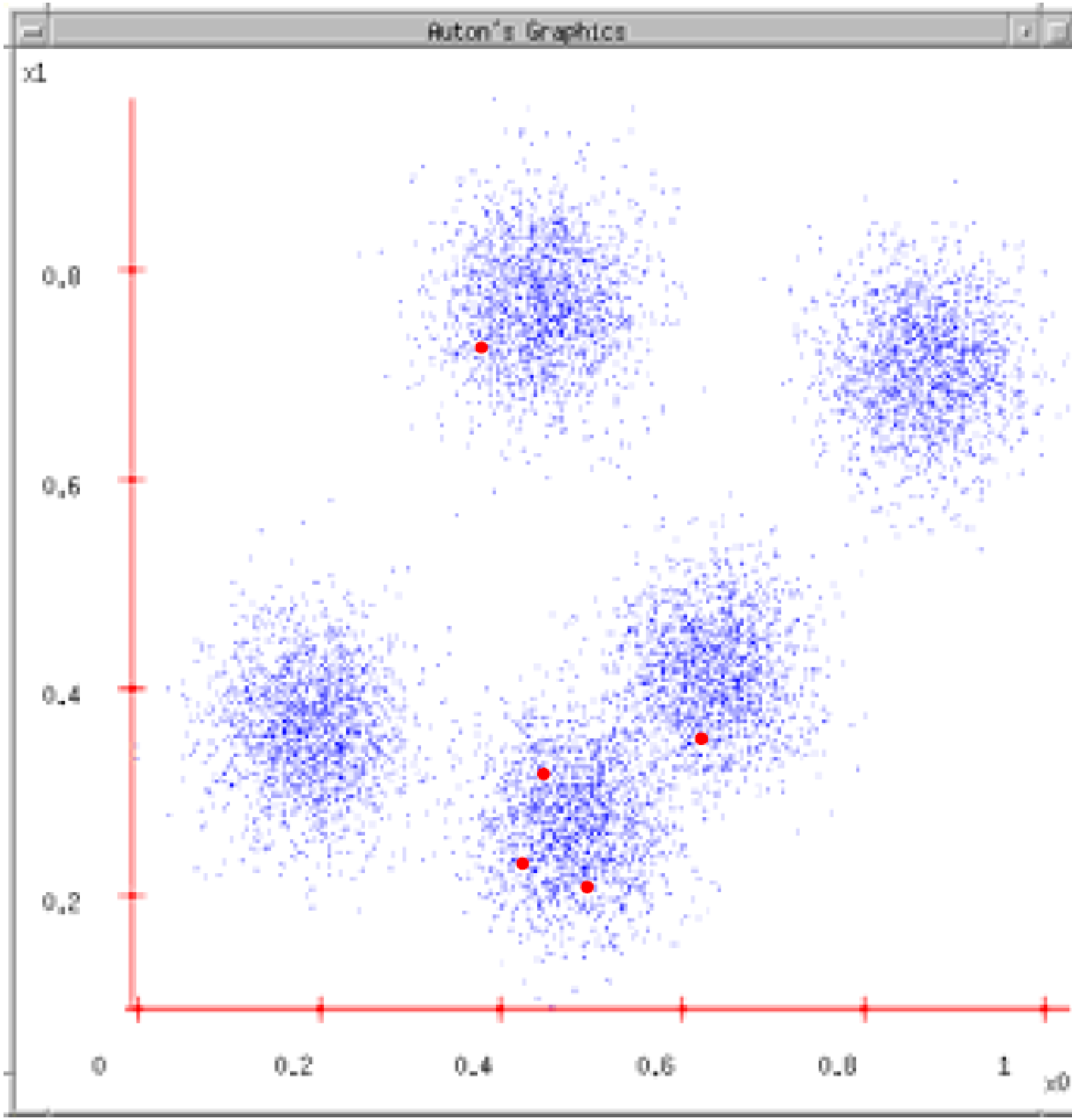
1. Ask user how many clusters they'd like. *(e.g. k=5)*



Slide credit Andrew Moore

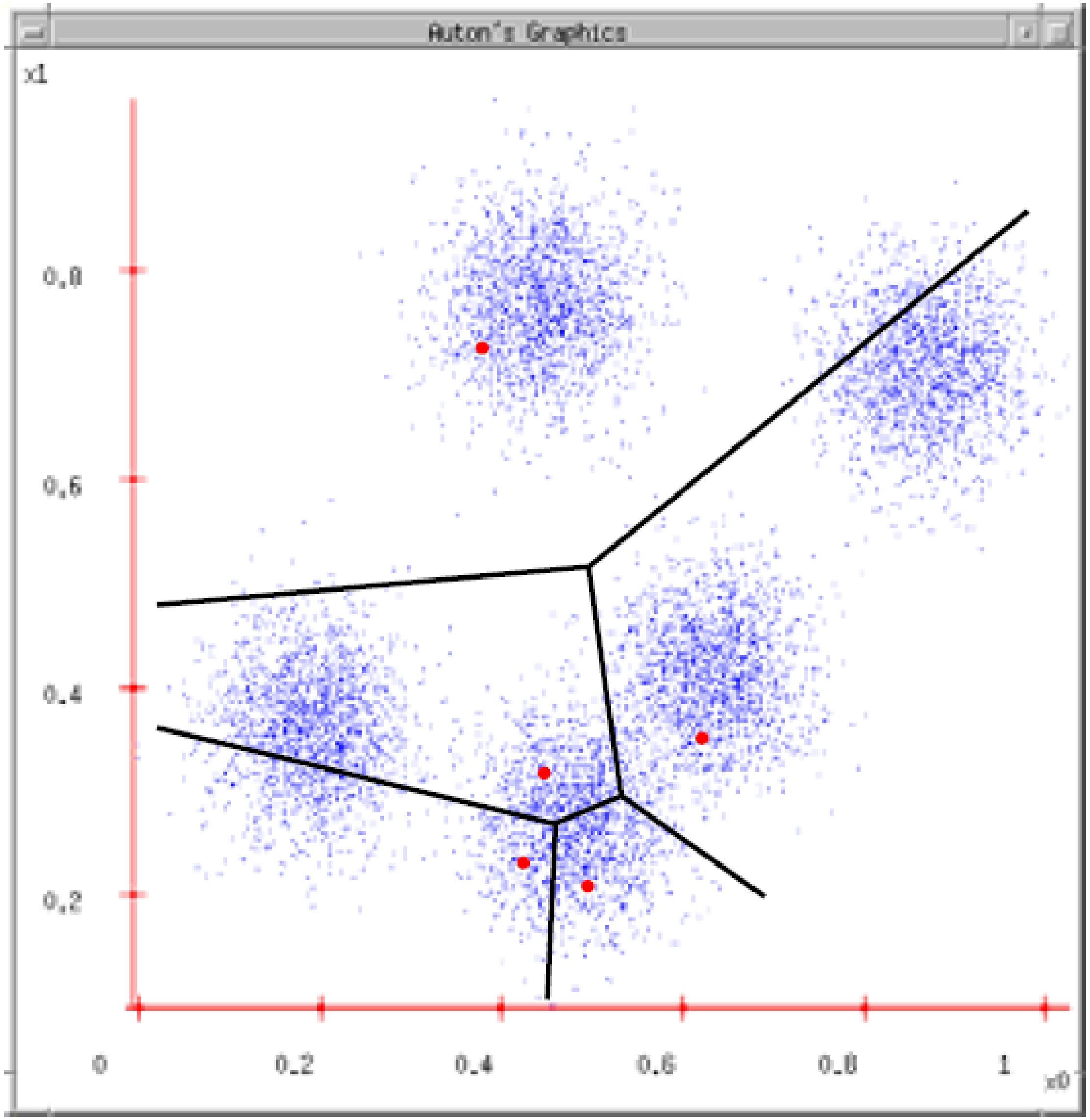
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



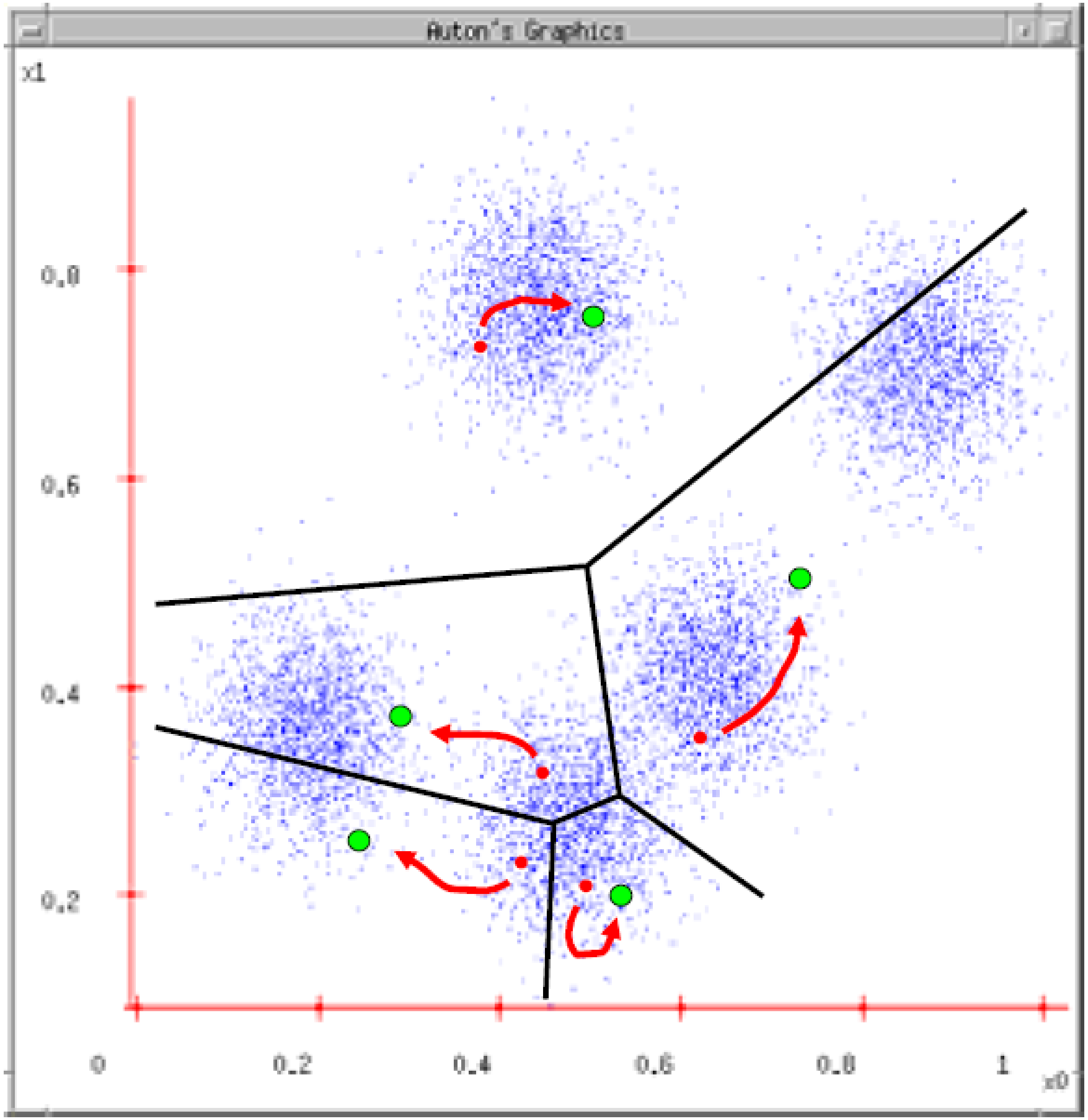
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



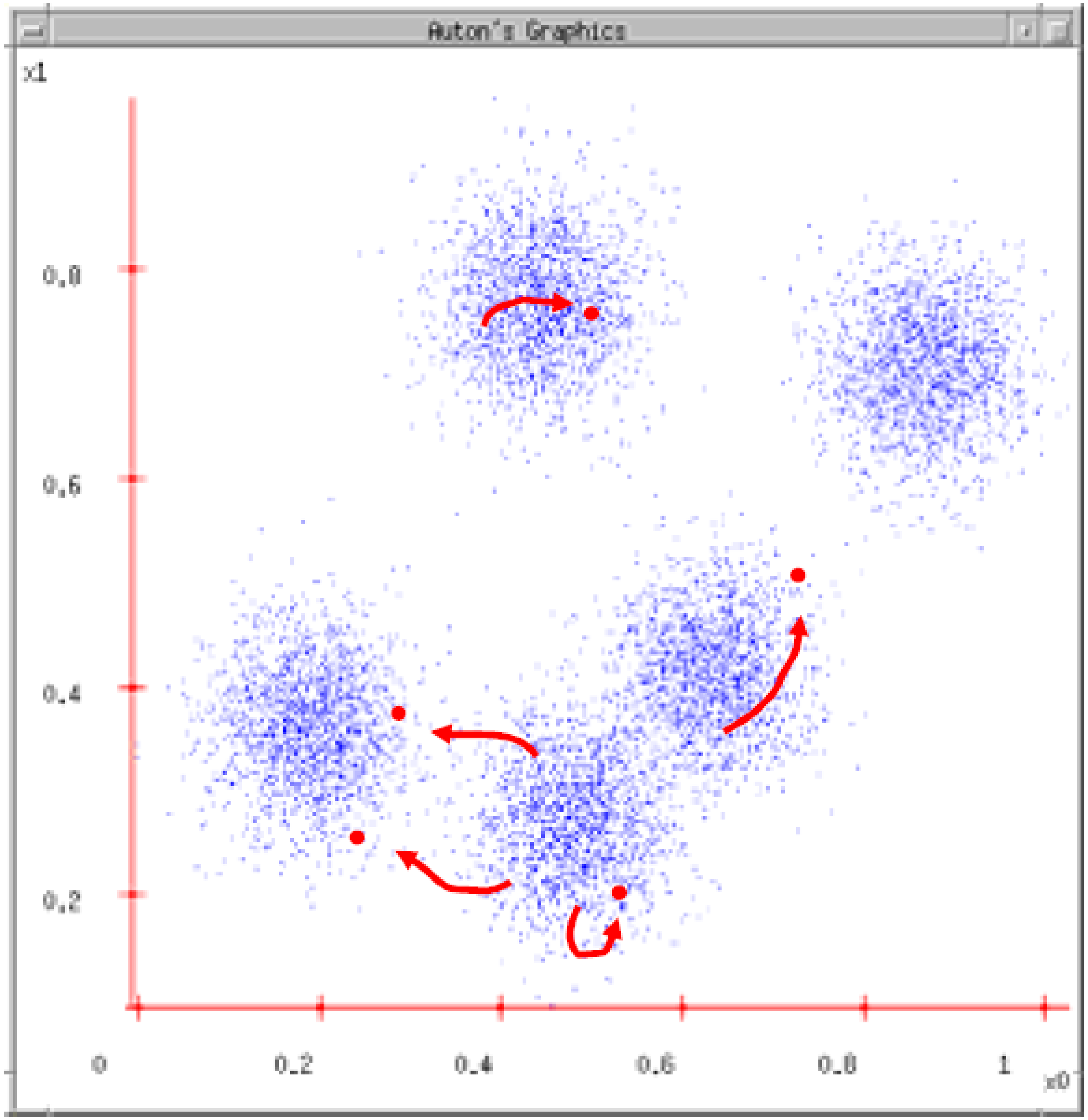
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



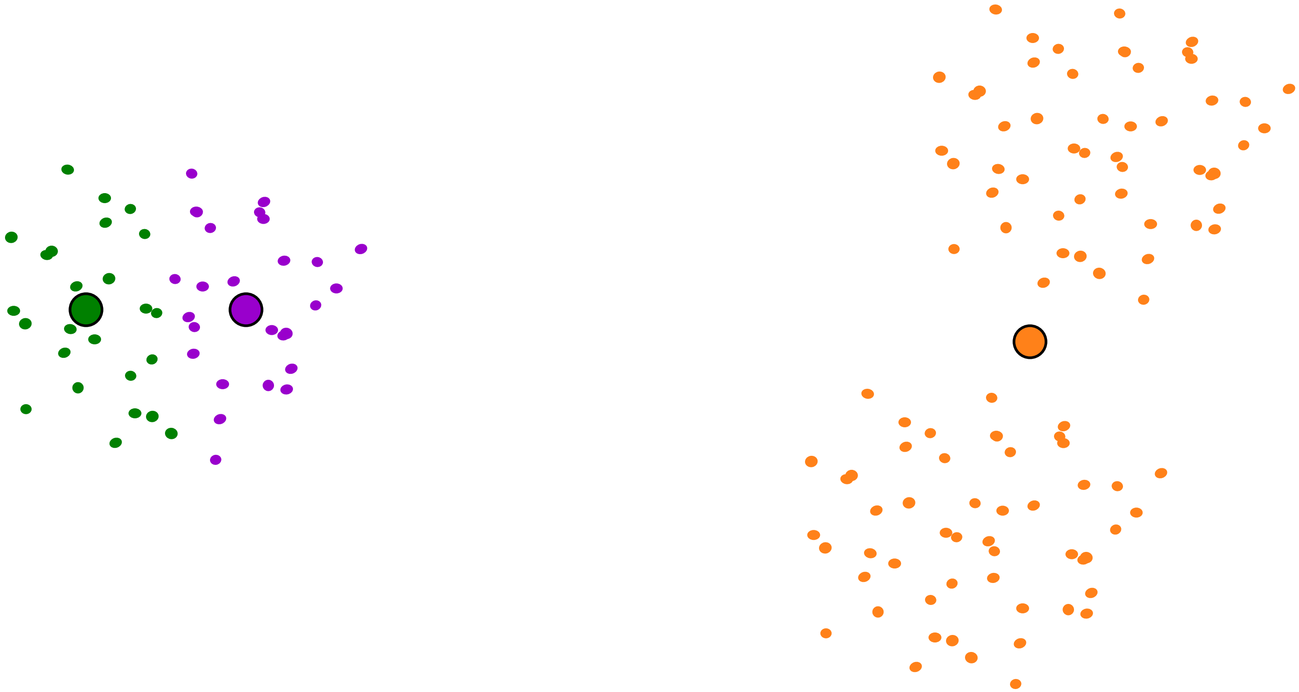
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



Limitation of K-Means

- A local optimum:

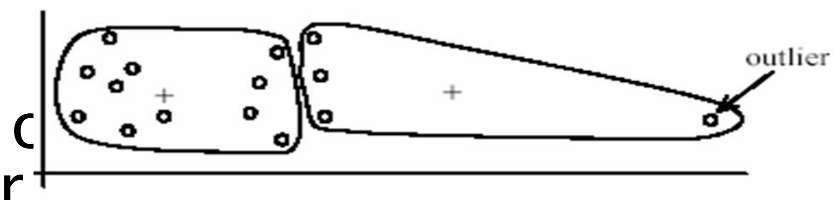


Slide credit: Dan Klein

K-means: pros and cons

Pros

- Simple, fast to compute
- Converges to local minimum c f within-cluster squared error

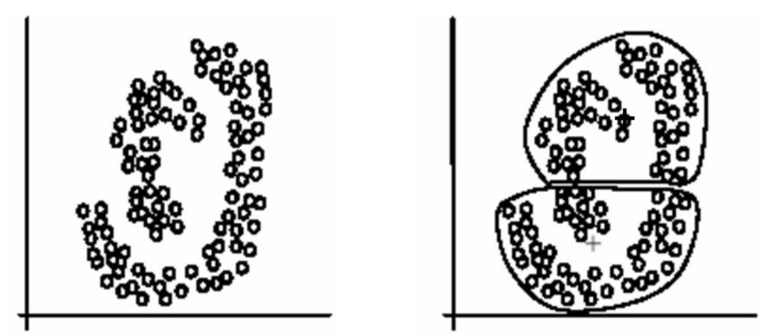


(A): Undesirable clusters

(B): Ideal clusters

Cons/issues

- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be comp uted



(A): Two natural clusters (B): *k*-means clusters

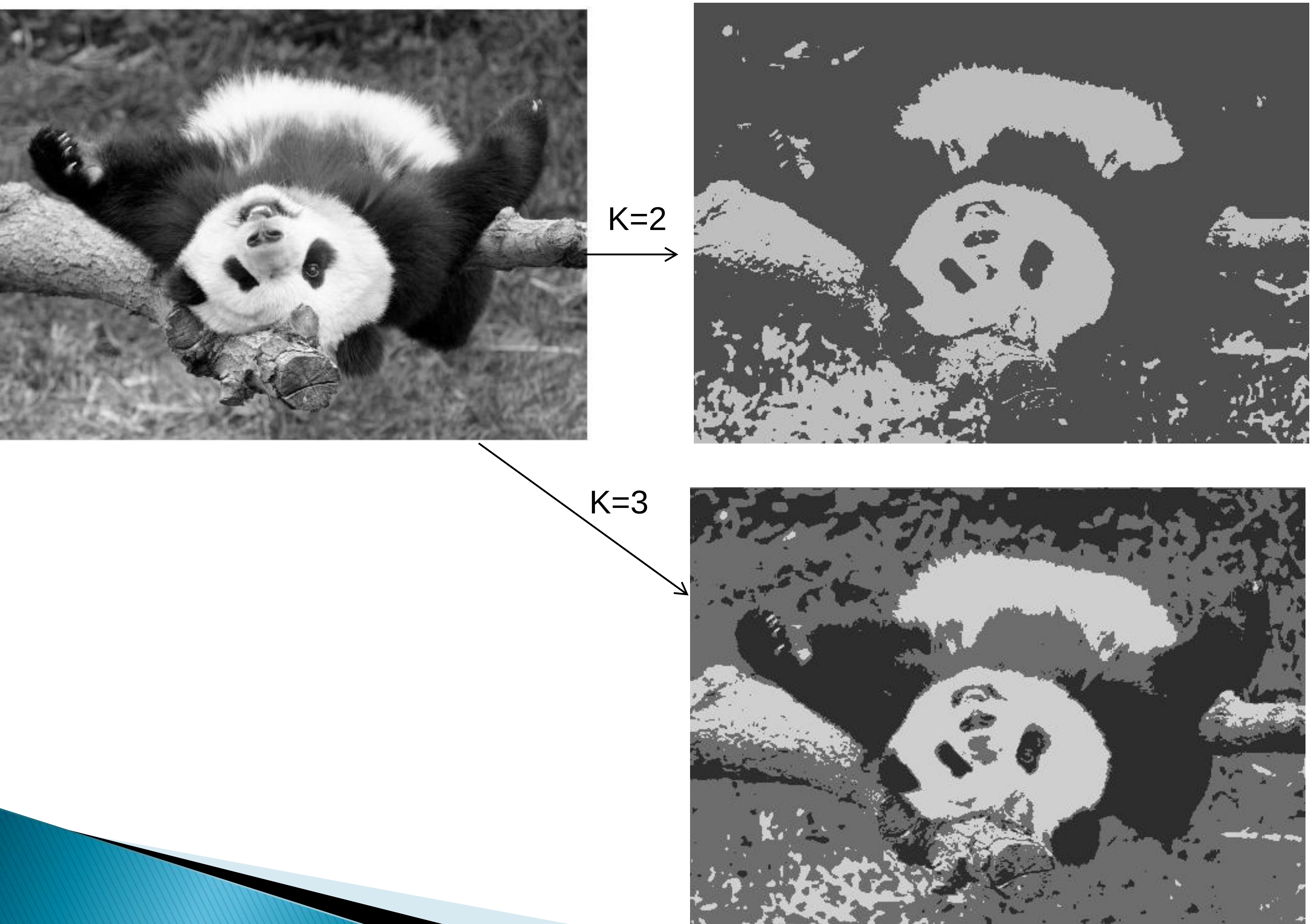
Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

Feature space: intensity value (1-d)

Slide credit: Kristen Grauman

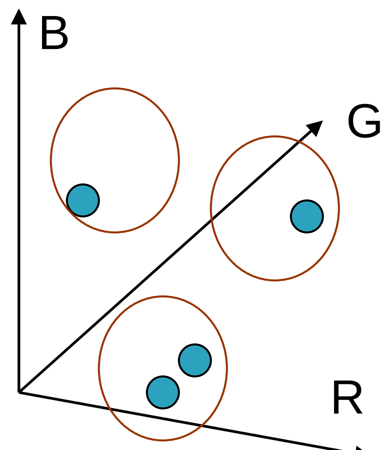


Slide credit: Kristen Grauman

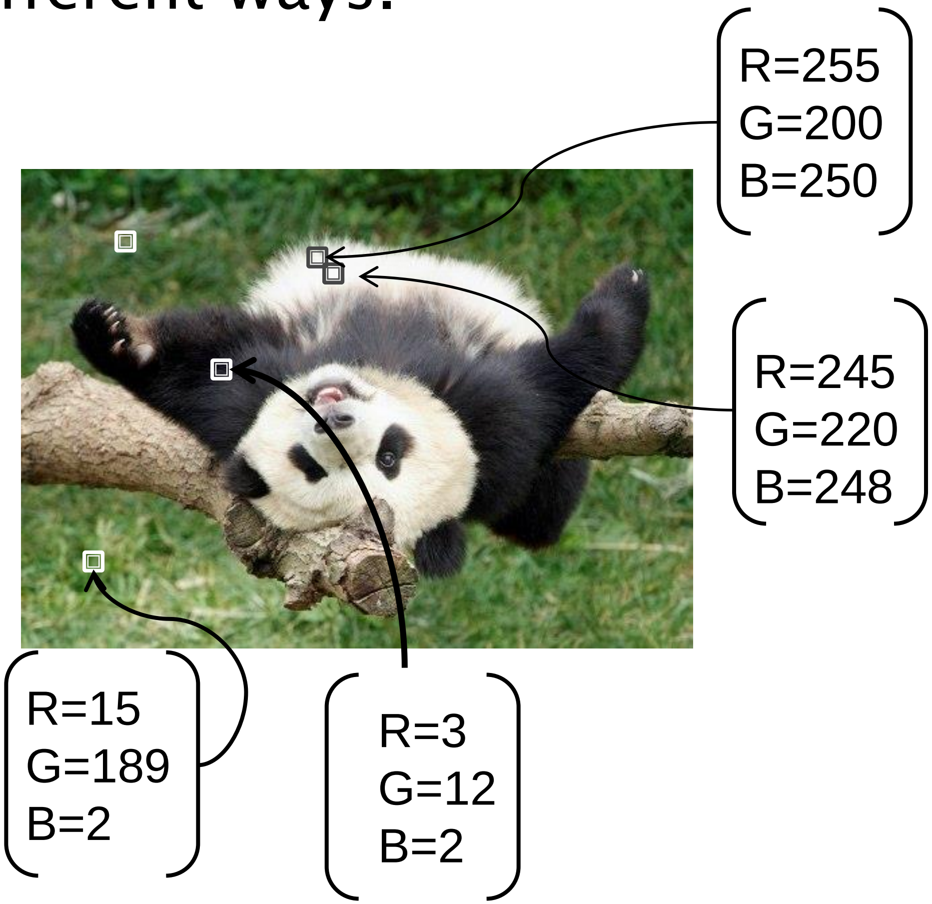
Segmentation as clustering

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **color** similarity



Feature space: color value (3-d)



Slide credit: Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based
on **intensity** similarity

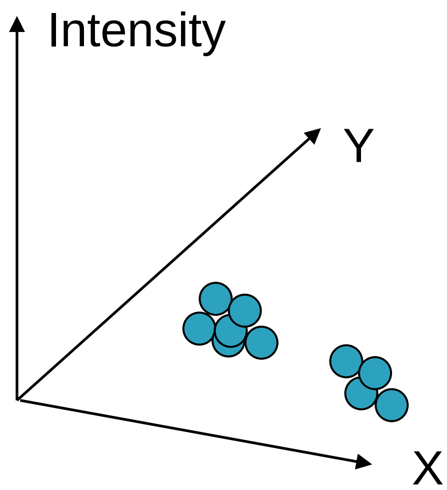
Clusters based on intensity similar
ity don’t have to be spatially coher
ent.

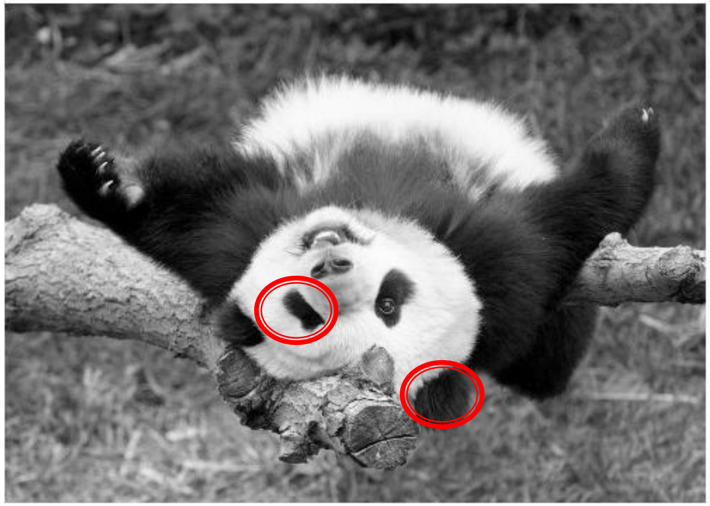
Slide credit: Kristen Grauman

Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on **int ensity+position** similarity





Both regions are black, but if we al so include **position (x,y),** then we could group the two into distinct se gments; way to encode both simila rity & proximity.

Slide credit: Kristen Grauman

Segmentation as clustering

- Color, brightness, position alone are not enough to distinguish all regions…

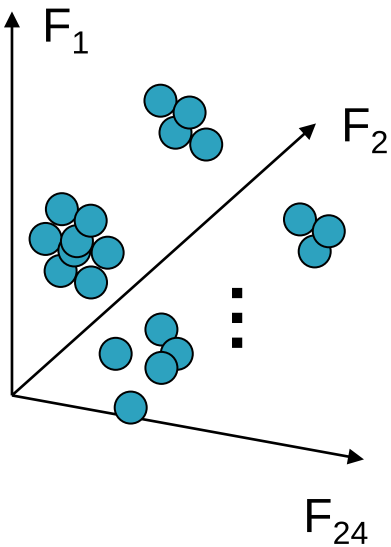


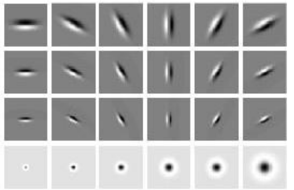
Slide credit: Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **texture** similarity



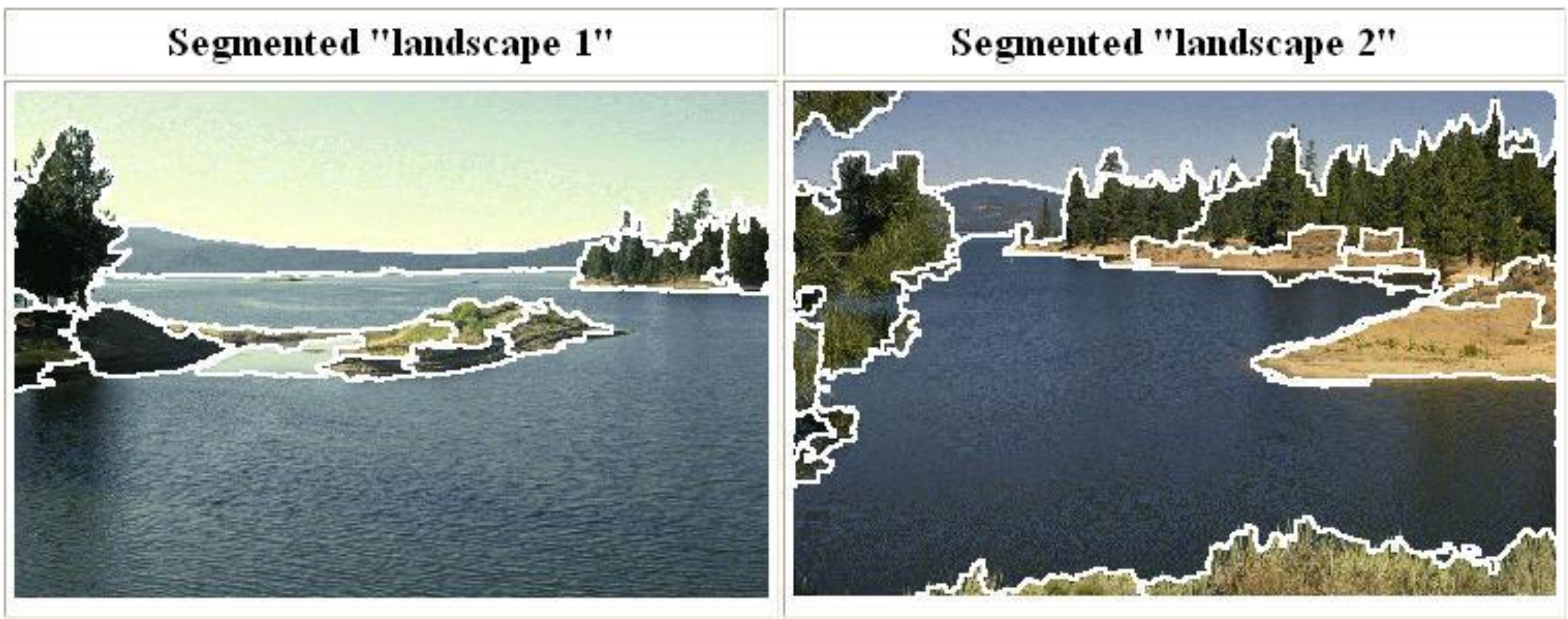


Filter bank of 24 filters

Feature space: filter bank responses (e.g., 24-d)

Mean shift clustering and segmentation

- An advanced and versatile technique for clustering-based segmentation



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

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

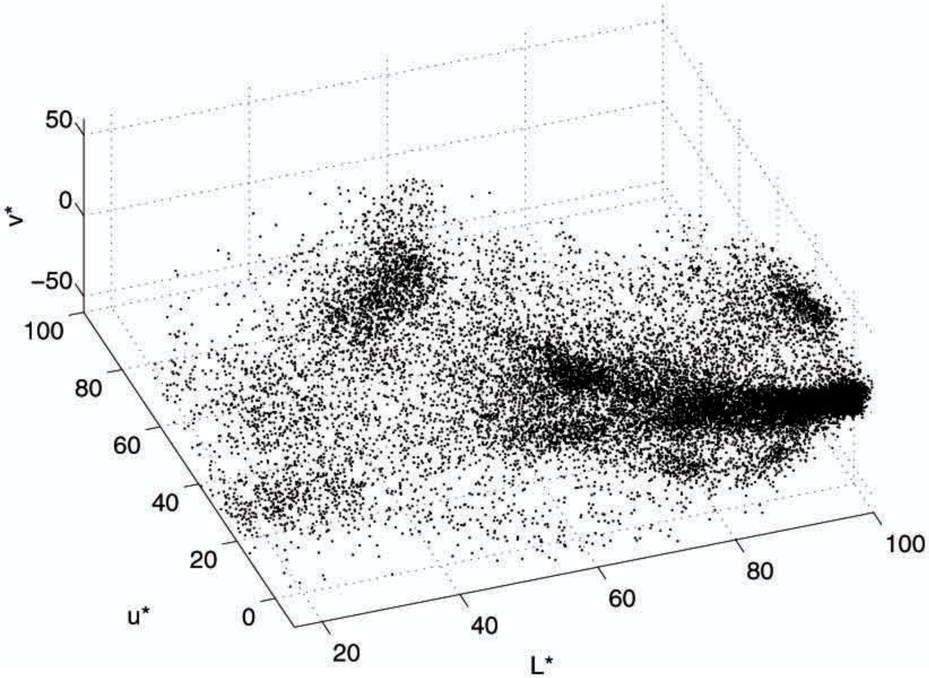
Slide credit: Svetlana Lazebnik

Mean shift algorithm

- The mean shift algorithm seeks *modes* or local maxima of density in the feature space

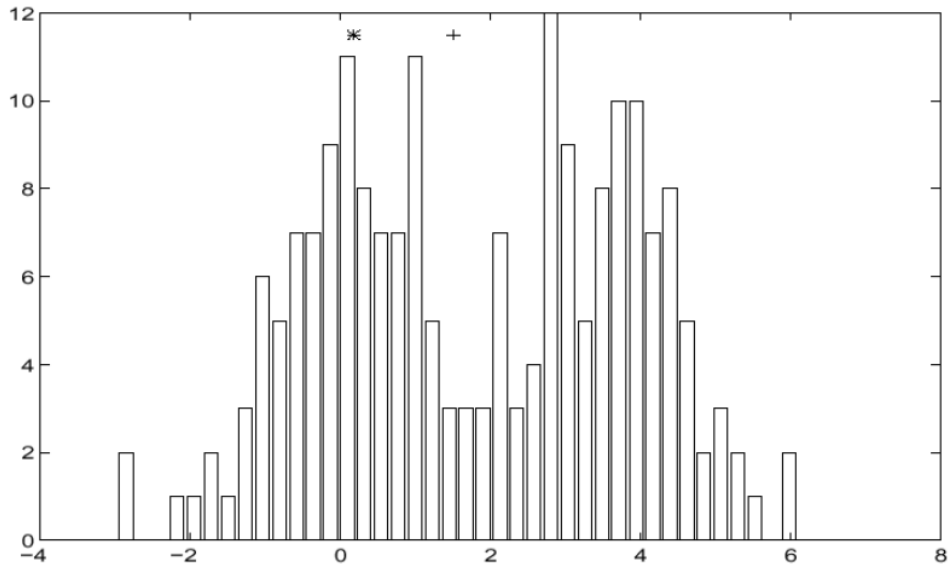
image

Feature space (L*u*v* color values)



Slide credit: Kristen Grauman

Mean-Shift Algorithm



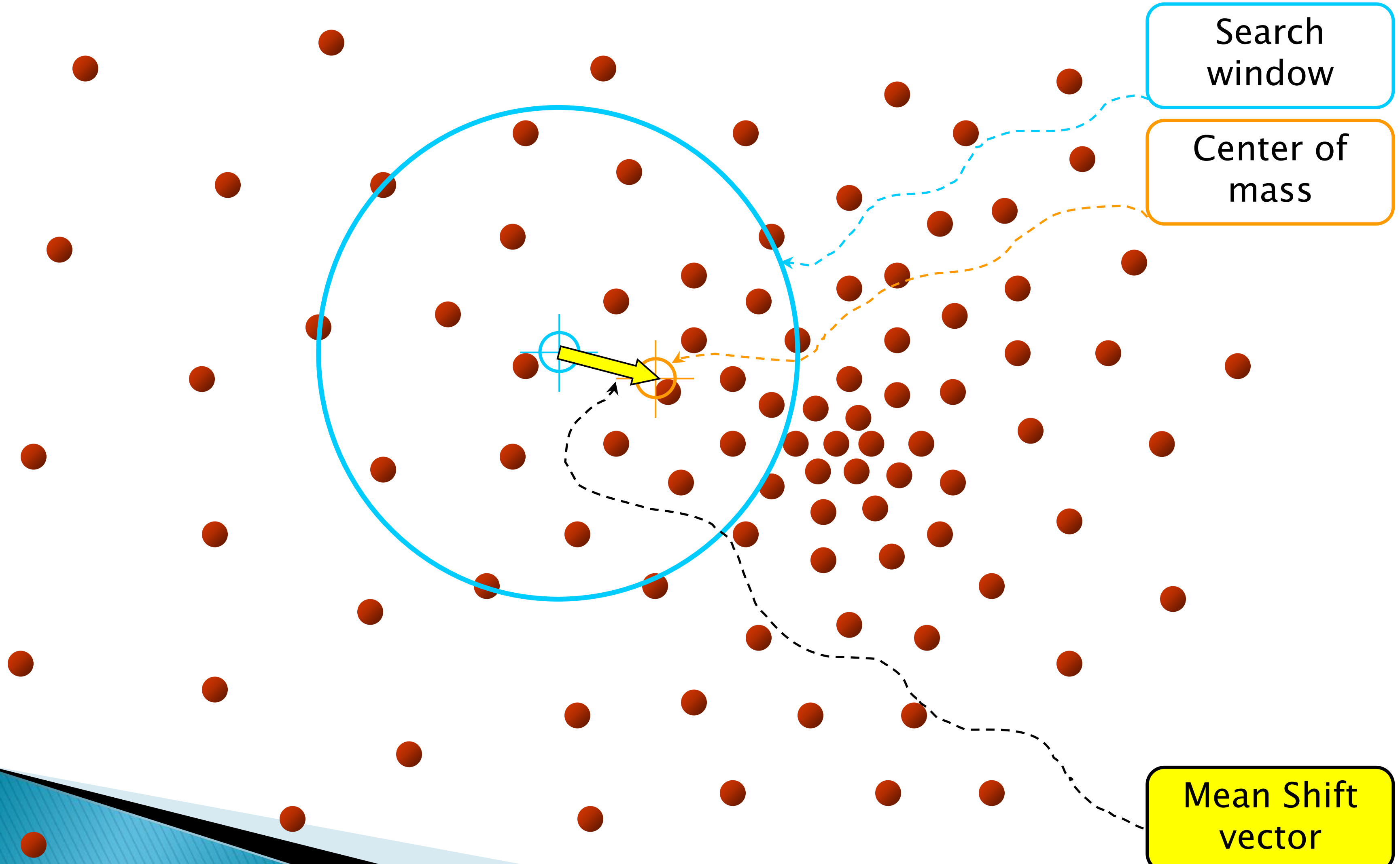
Iterative Mode Search

1. Initialize random seed, and window W
2. Calculate center of gravity (the "mean") of W:
3. Shift the search window to the mean
4. Repeat Step 2 until convergence

$$\sum_{x \in W} x H(x)$$

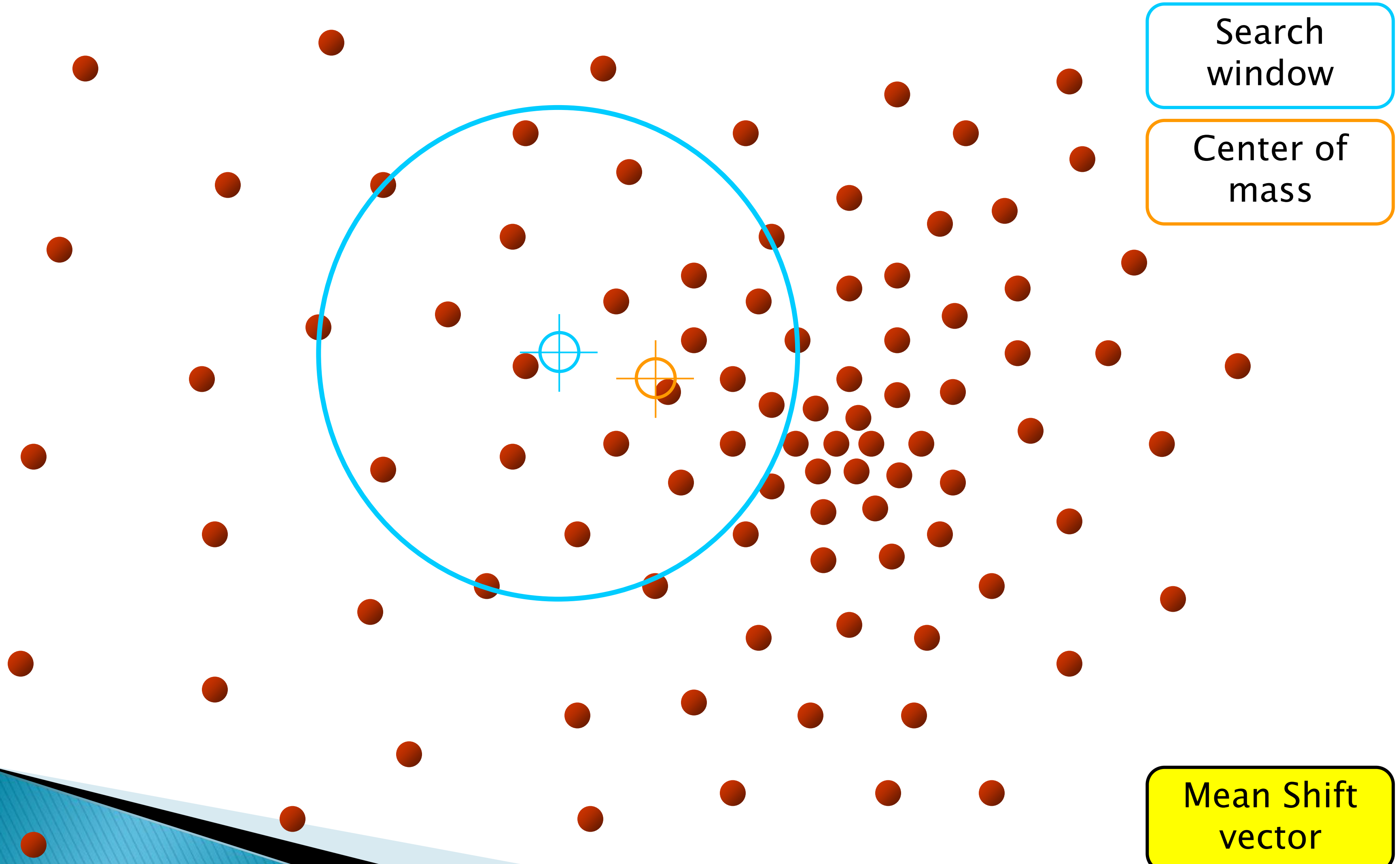
Slide credit: Svetlana Lazebnik

Mean Shift



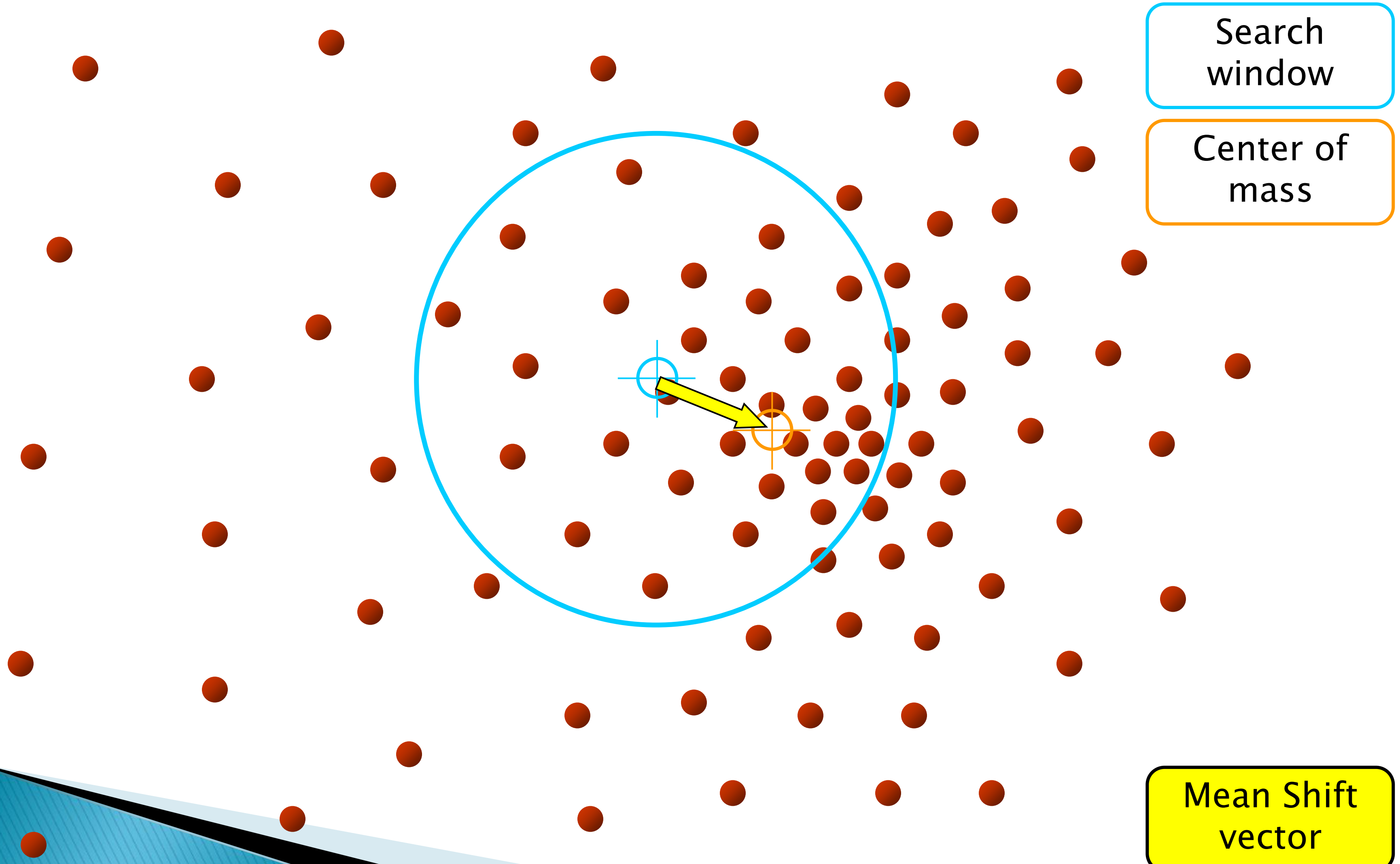
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



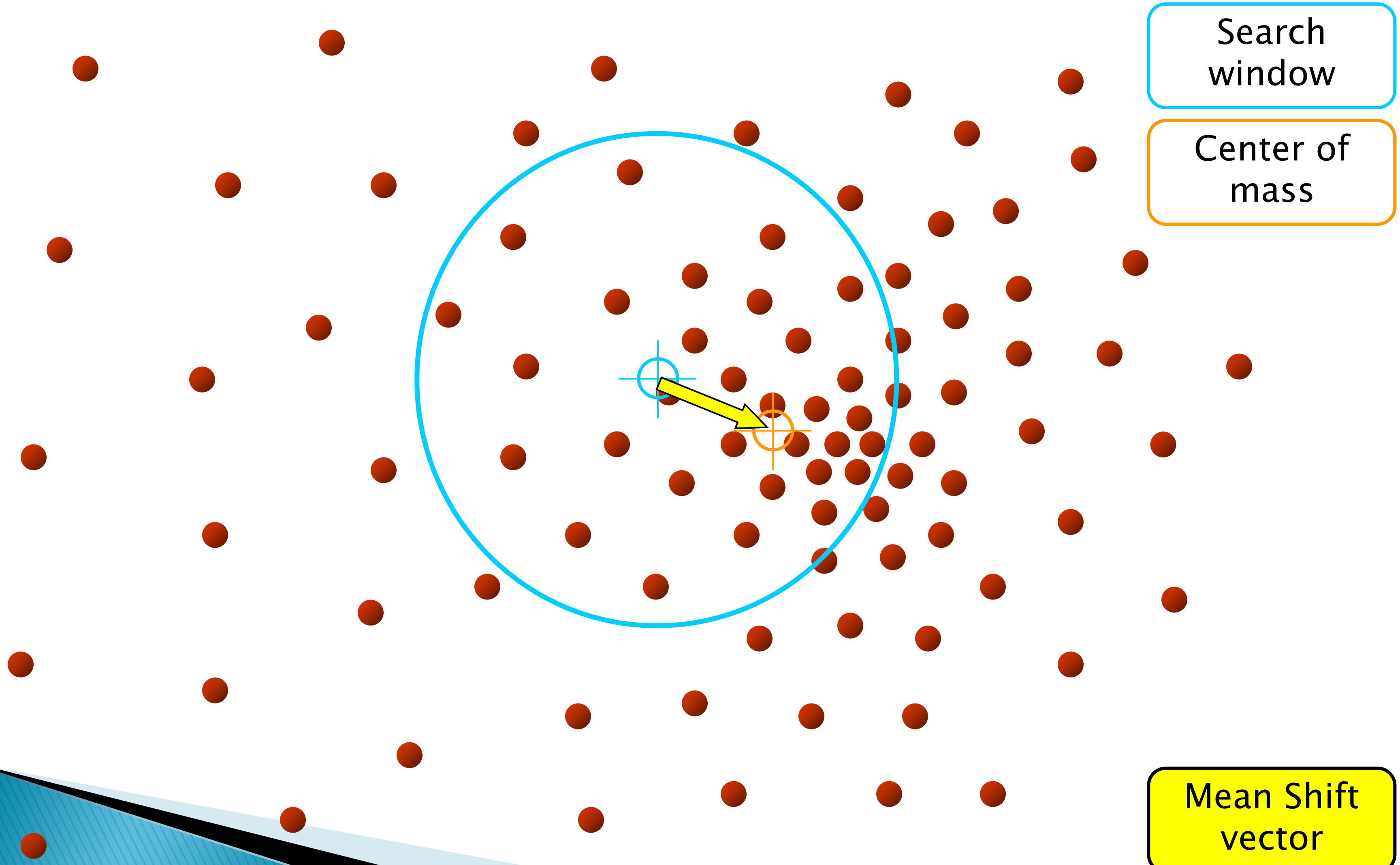
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



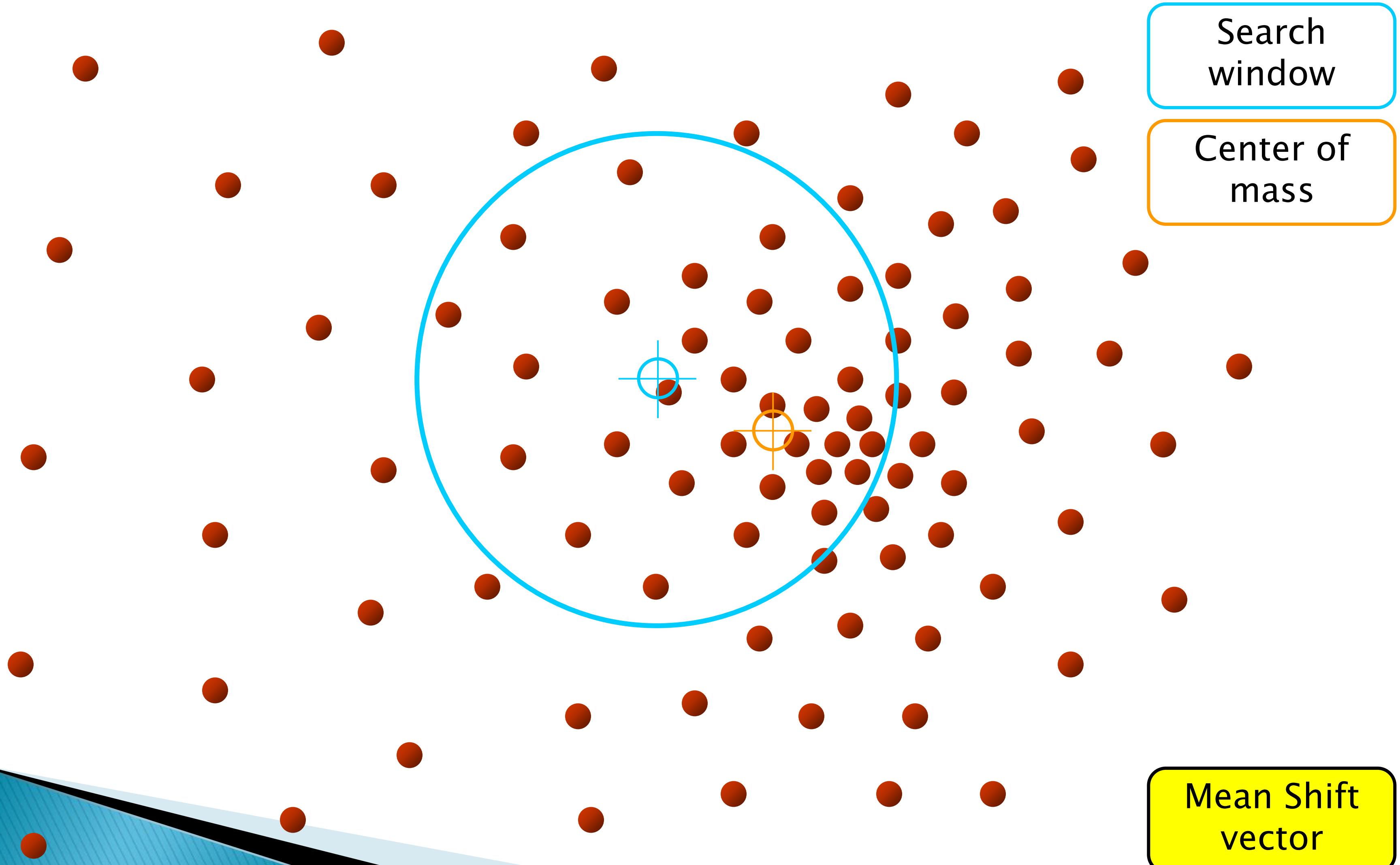
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



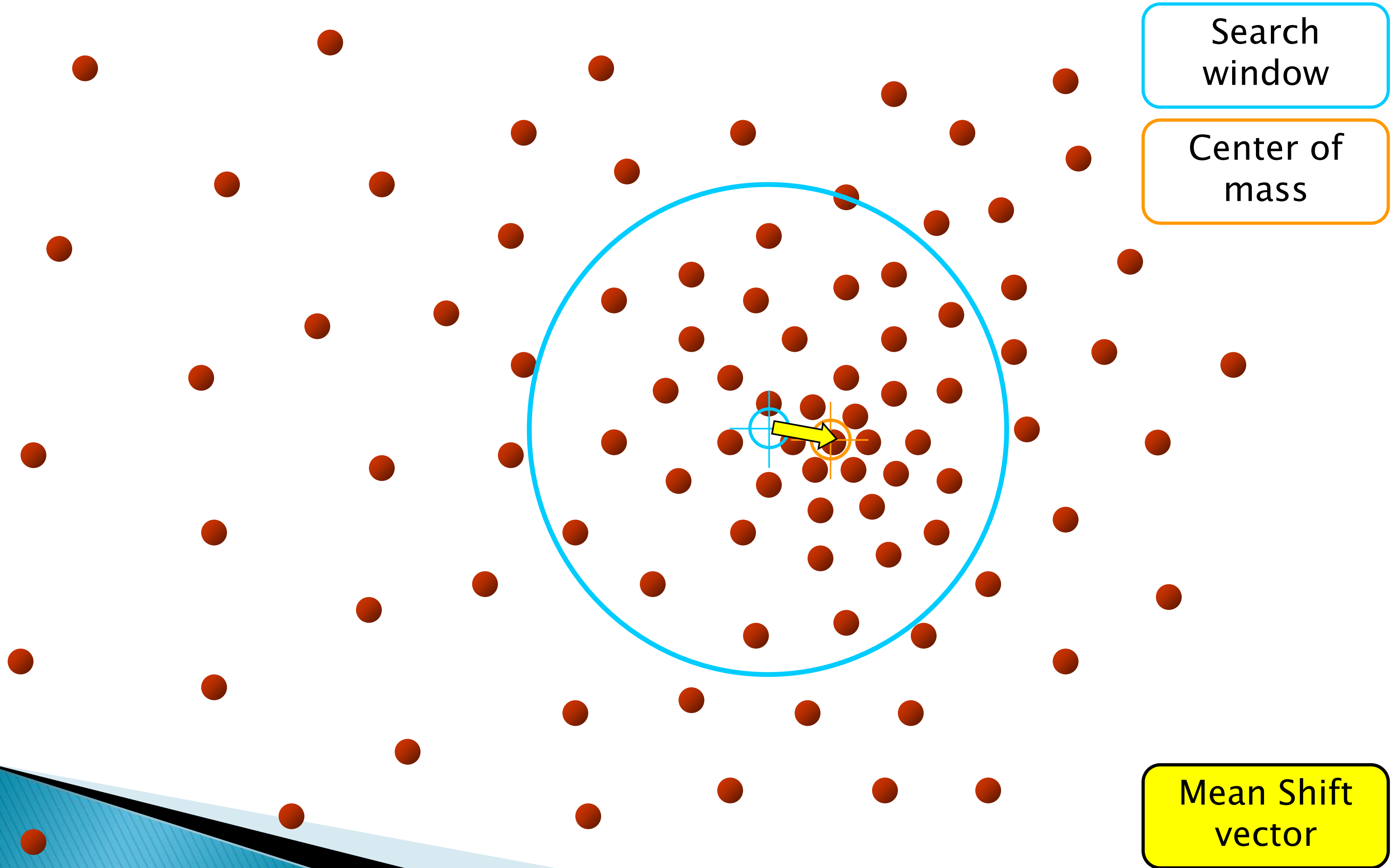
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



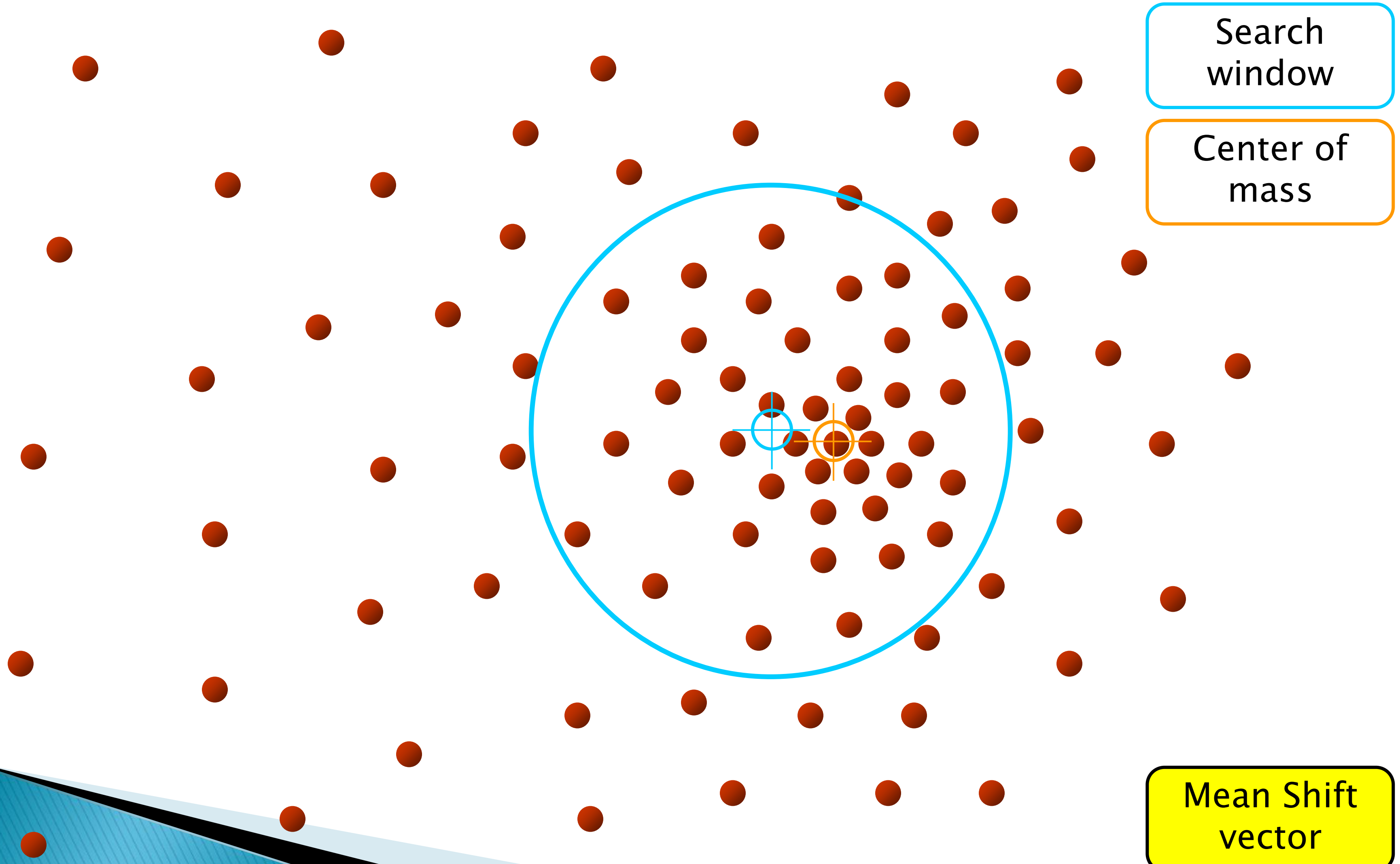
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



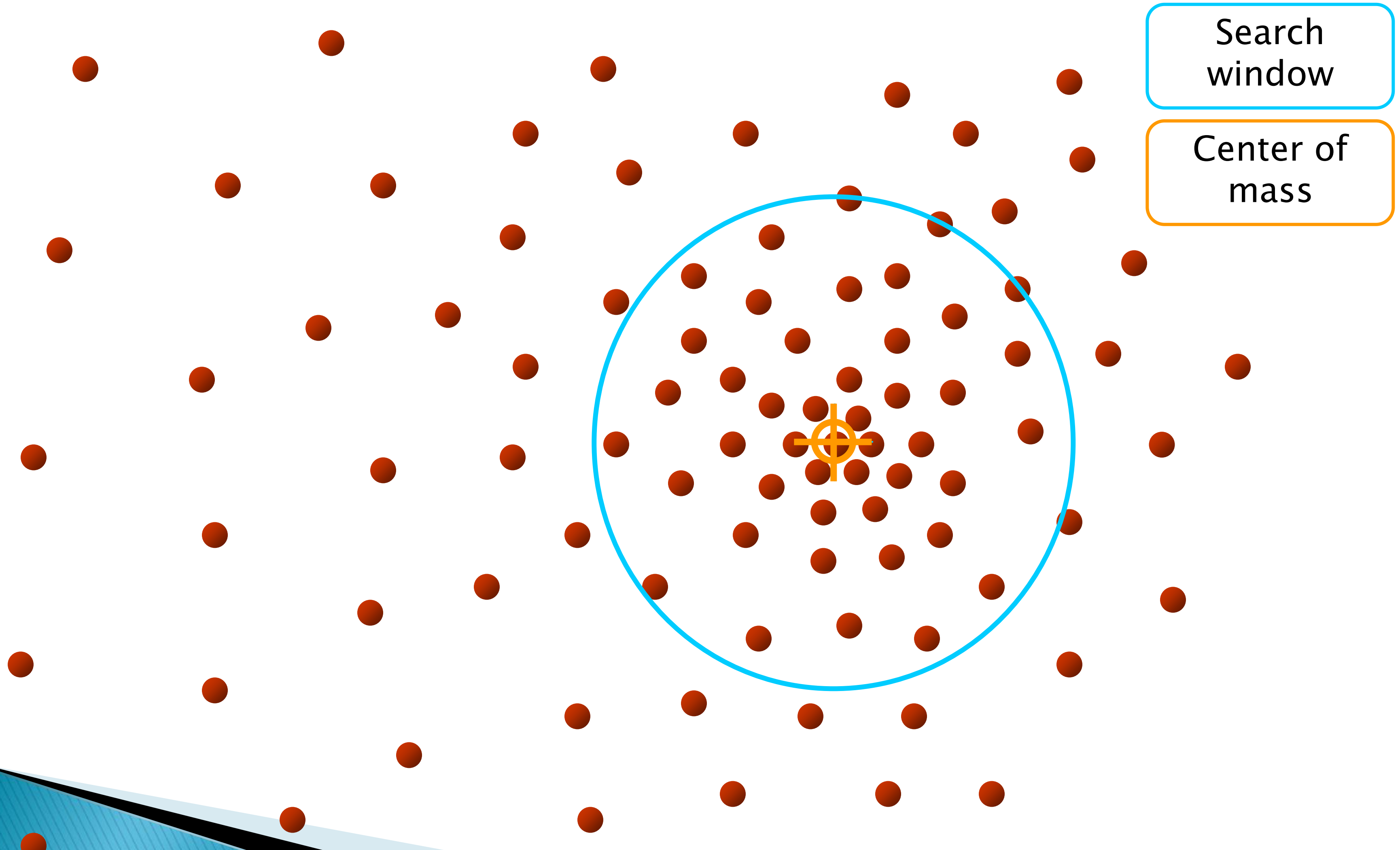
Slide by Y. Ukrainitz & B. Sarel

Mean Shift



Slide by Y. Ukrainitz & B. Sarel

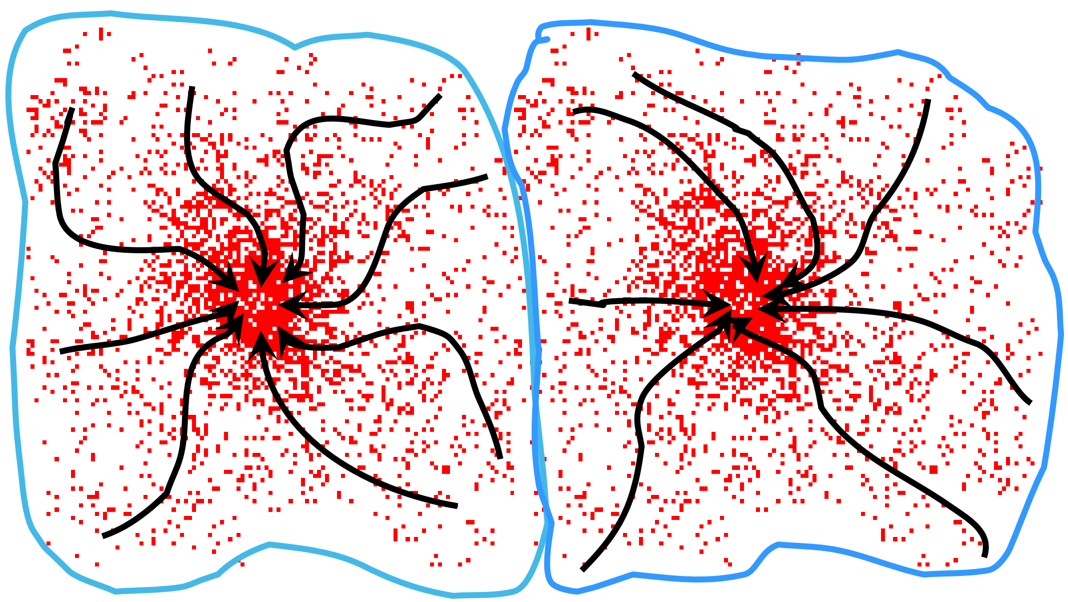
Mean Shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering

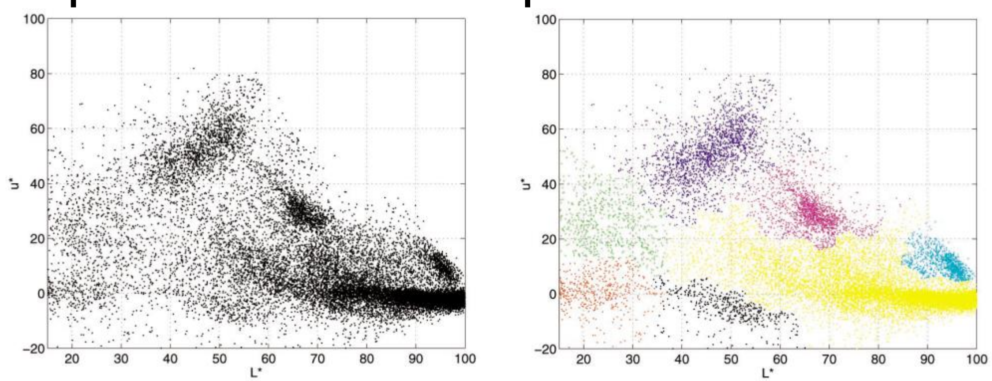
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



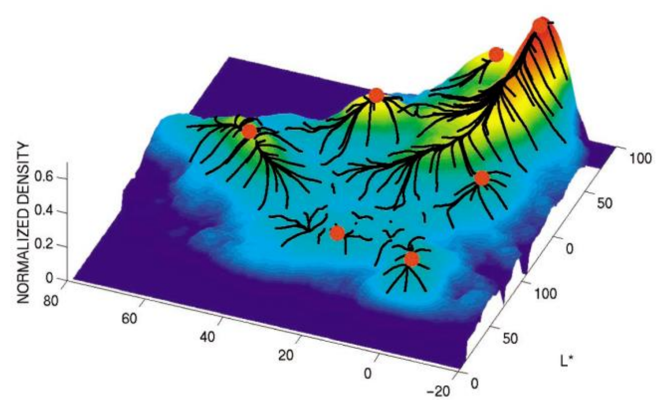
Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering/segmentation

- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode



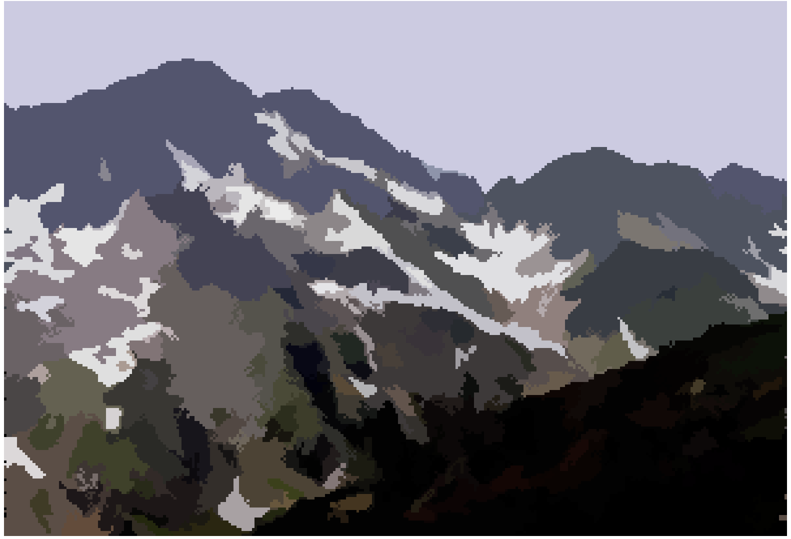
(a) (b)



Slide credit: Svetlana Lazebnik

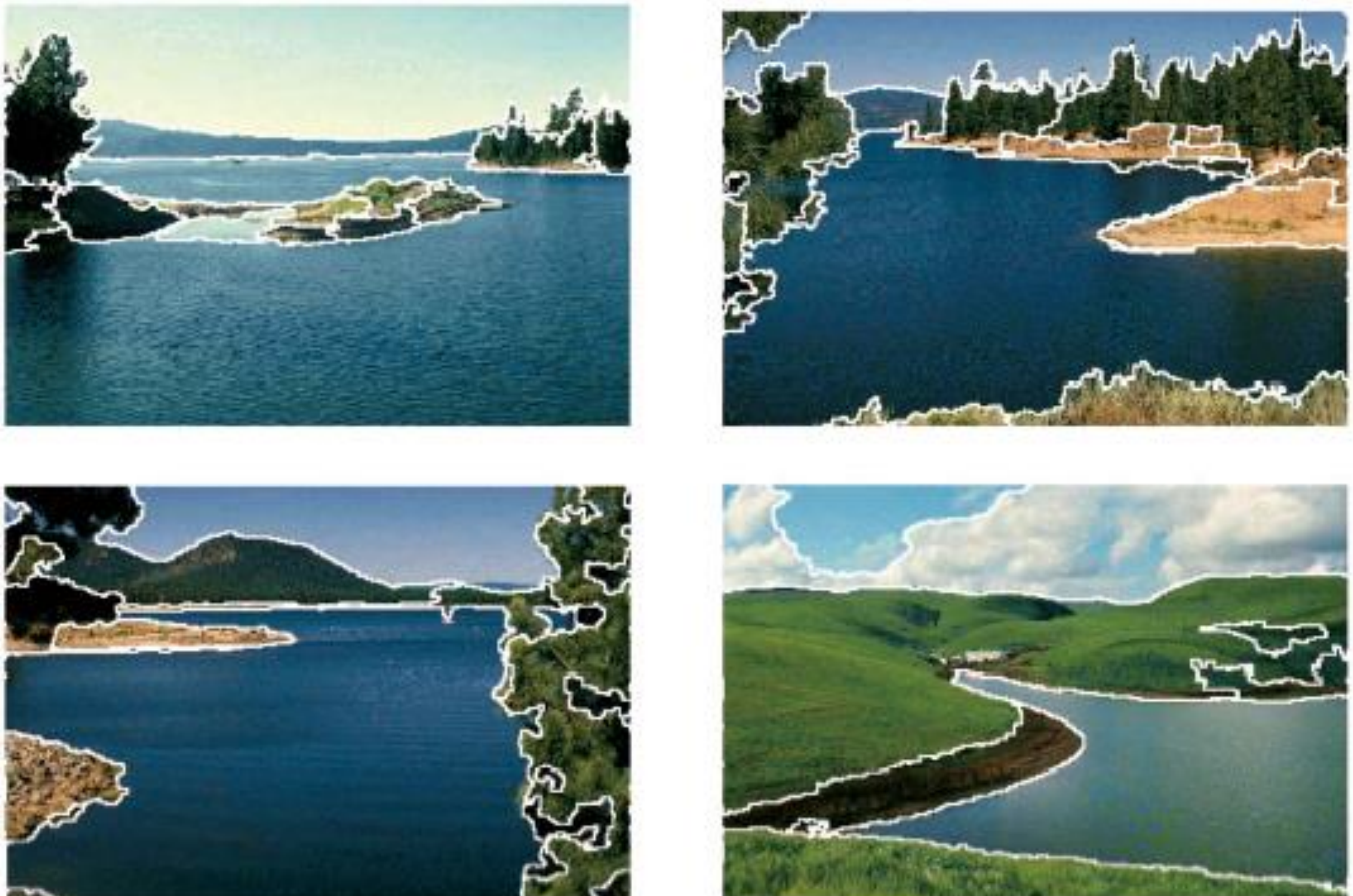
Mean shift segmentation results



Slide credit: Svetlana Lazebnik

Mean shift segmentation results



Slide credit: Svetlana Lazebnik

Mean shift segmentation results

Slide credit: Svetlana Lazebnik

Summary Mean-Shift

- Pros
  - General, application-independent tool
  - Model-free, does not assume any prior shape (spherical, elliptical, etc.) on data clusters
  - Just a single parameter (window size h)
    - h has a physical meaning (unlike k-means)
  - Finds variable number of modes
  - Robust to outliers

- Cons
  - Output depends on window size
  - Window size (bandwidth) selection is not trivial