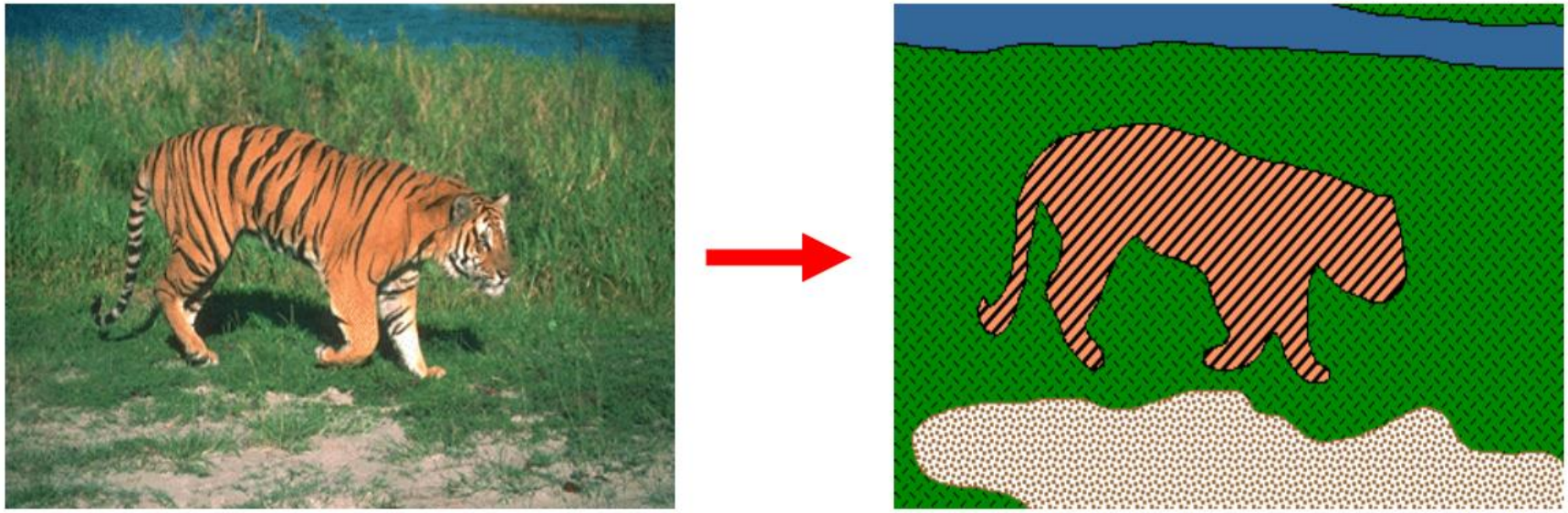
Image Segmentation

Image segmentation



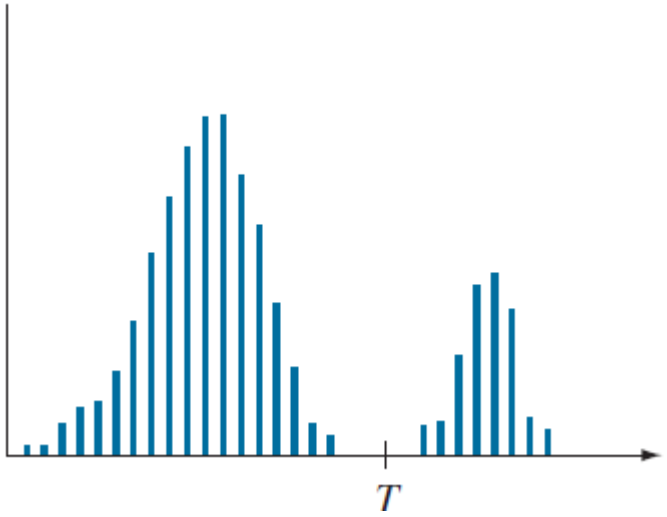
Slide credit: S. Seitz, K. Grauman

Image segmentation

- Thresholding
- Clustering
- Superpixels
- Graph cuts
- Morphological watersheds

Thresholding

The basics of intensity thresholding



Intensity histogram of an image

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

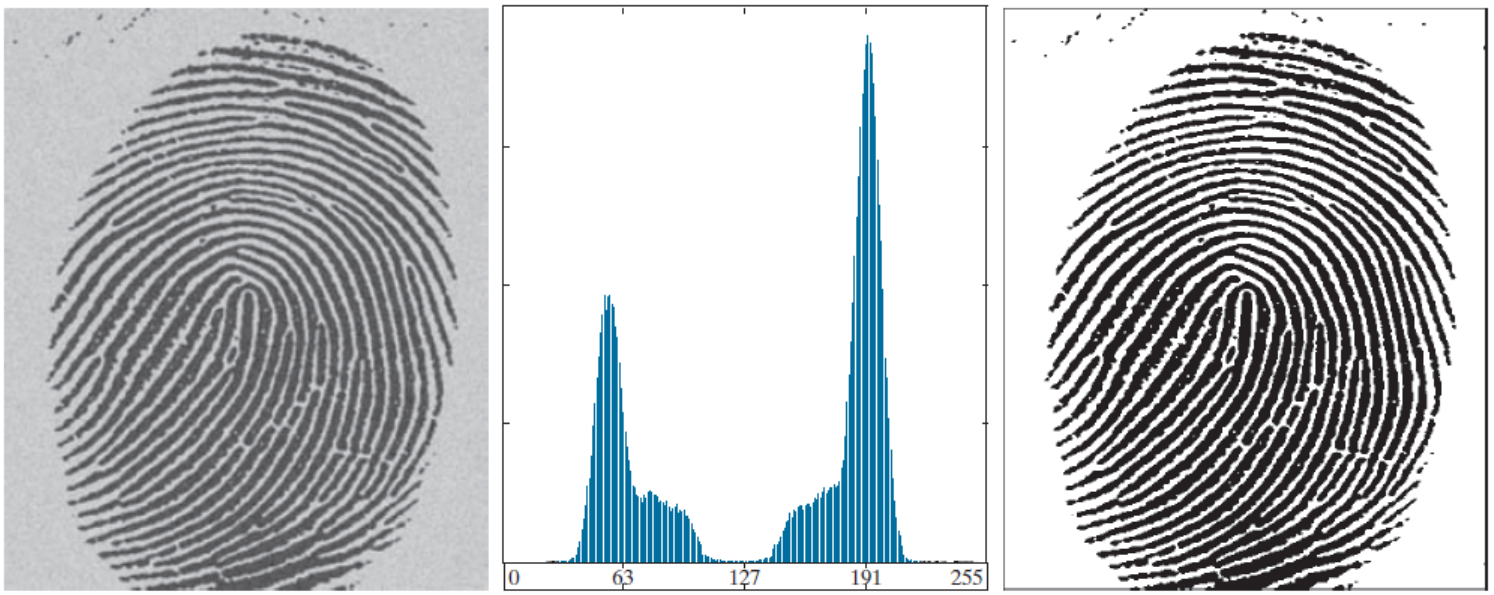
Basic global thresholding

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T in Eq. (10-46). This will produce two groups of pixels: G_1, consisting of pixels with intensity values $> T$; and G_2, consisting of pixels with values $\leq T$.
3. Compute the average (mean) intensity values m_1 and m_2 for the pixels in G_1 and G_2, respectively.
4. Compute a new threshold value midway between m_1 and m_2:

$$T = \frac{1}{2}(m_1 + m_2)$$

5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined value, ΔT.

Basic global thresholding



a b c

FIGURE 10.35 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (thin image border added for clarity). (Original image courtesy of the National Institute of Standards and Technology.).

Optimum global thresholding using Otsu's method

- Divide the pixels into two classes by choosing a threshold that maximizes between class variance

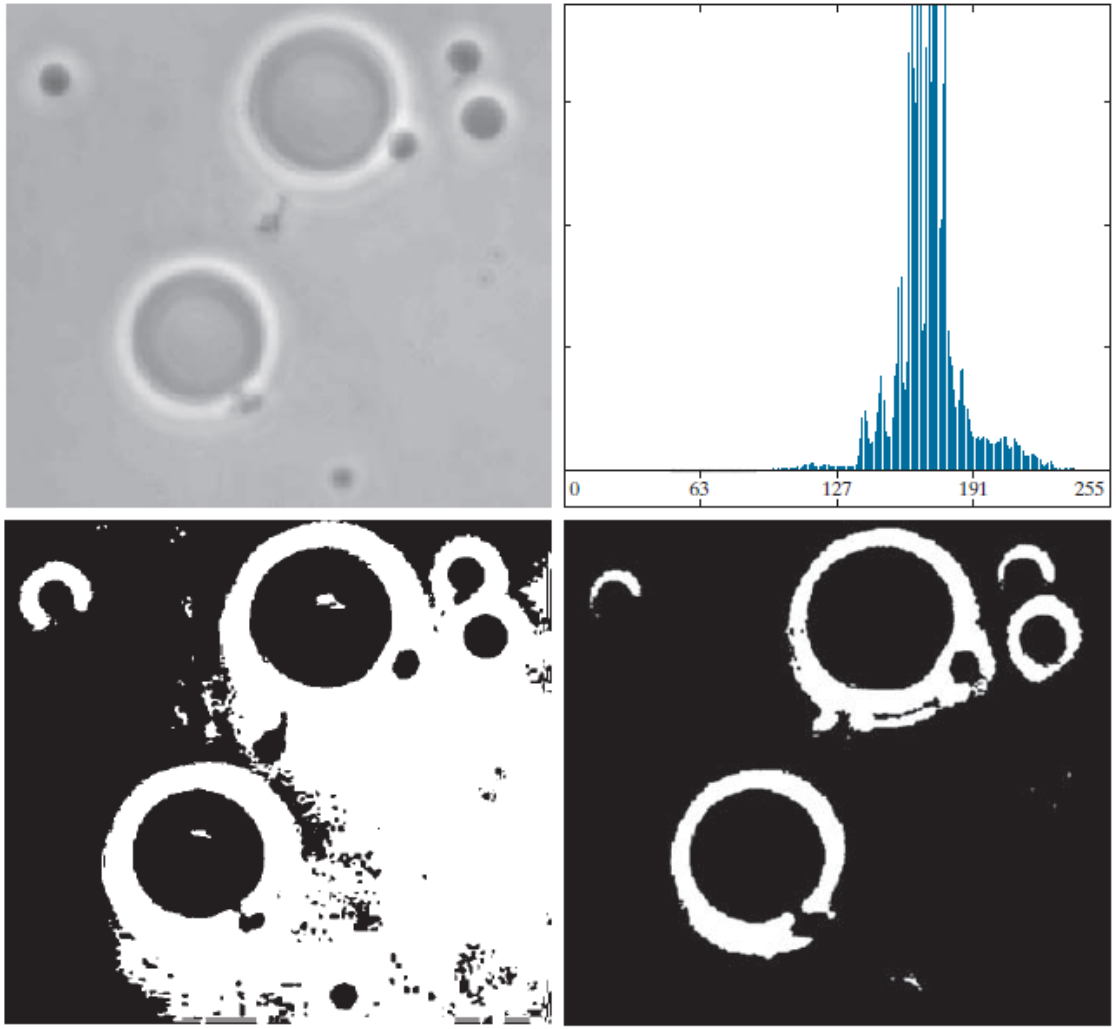
$$\sigma_B^2 = P_1 (m_1 - m_G)^2 + P_2 (m_2 - m_G)^2$$

mG: mean intensity of all pixels

P1: probability of pixels in class 1
m1: mean intensity of pixels in class 1

P2: probability of pixels in class 2
m2: mean intensity of pixels in class 2

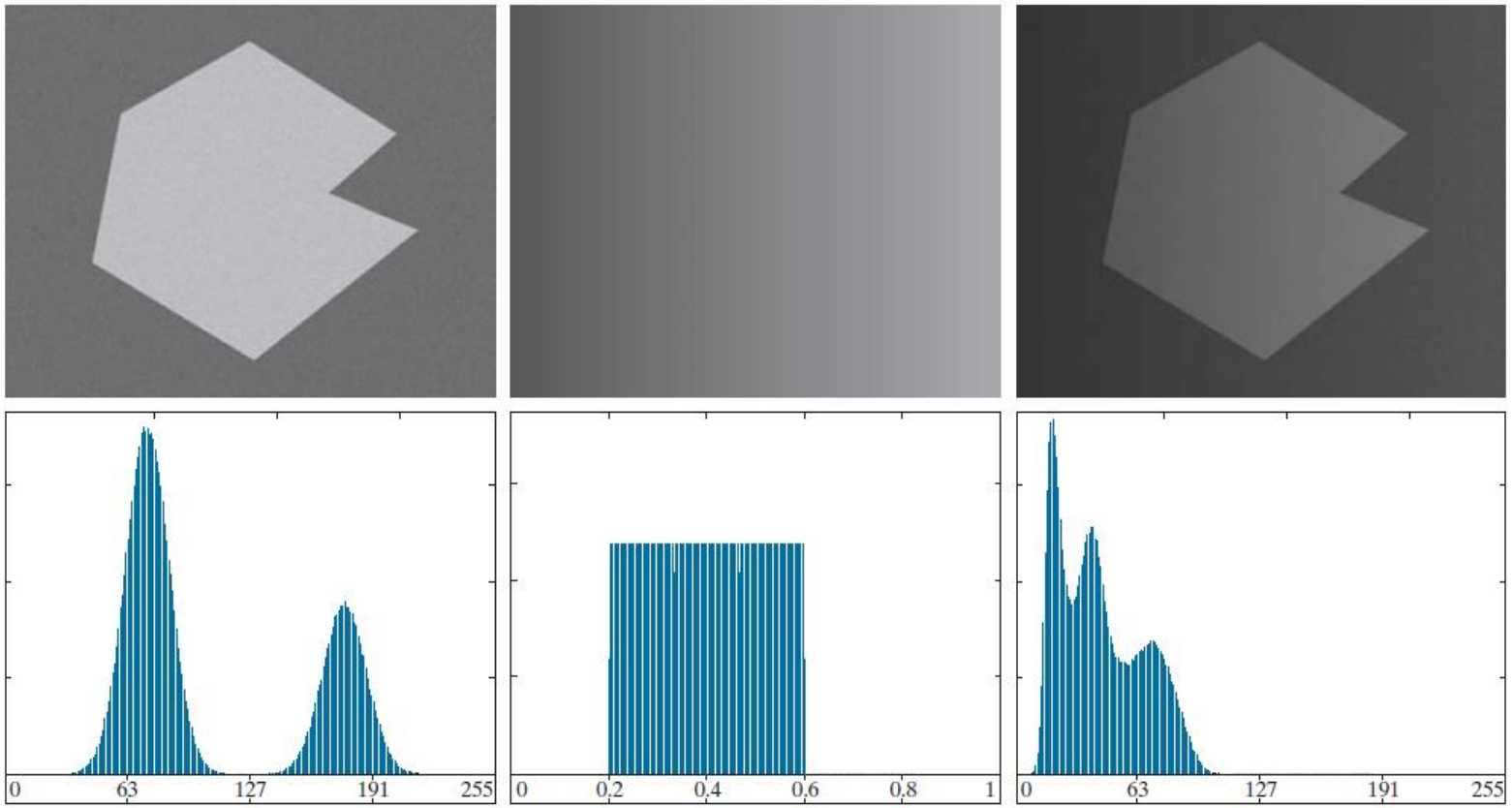
Optimum global thresholding using Otsu's method



a b
c d

FIGURE 10.36
(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.
(d) Result using Otsu's method.
(Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

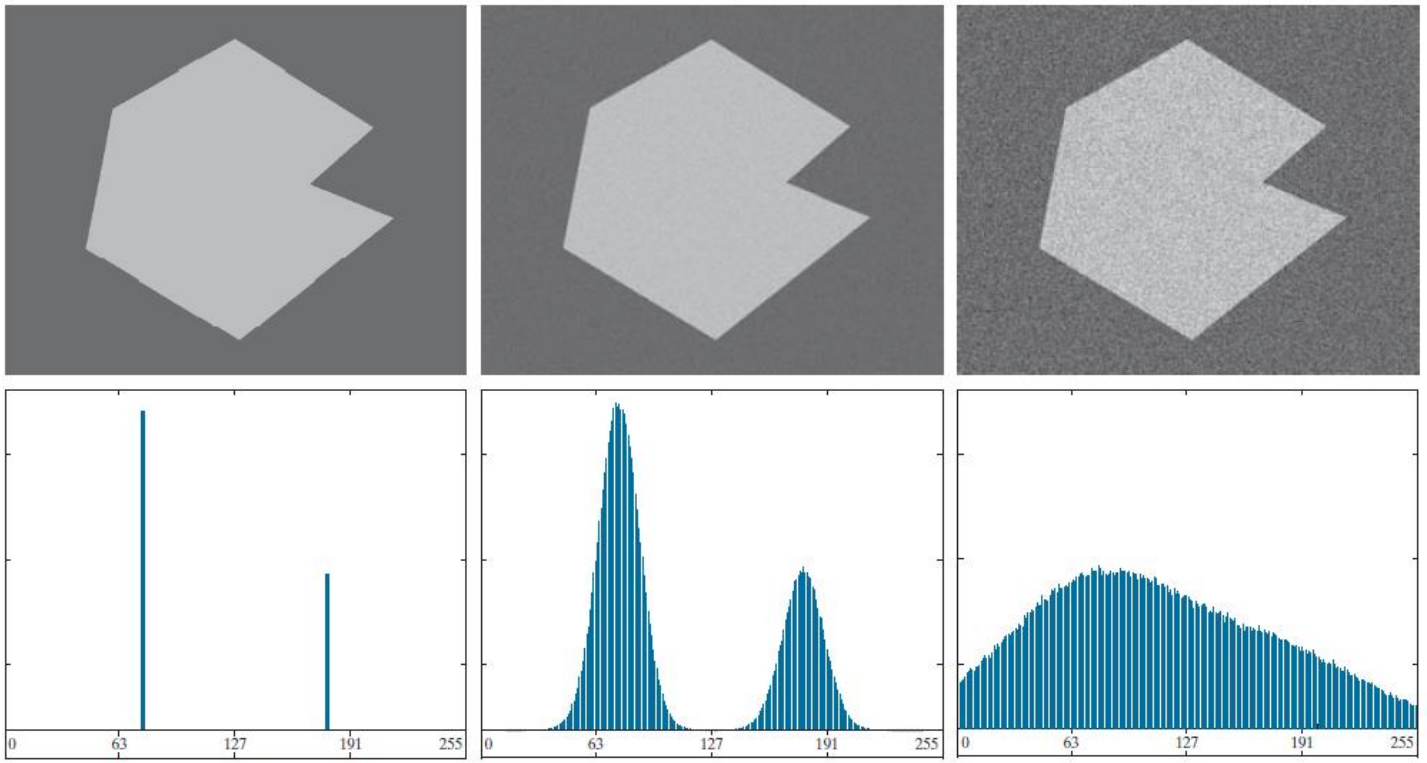
The role of illumination and reflectance in image thresholding



a b c
d e f

FIGURE 10.34 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d) through (f) Corresponding histograms.

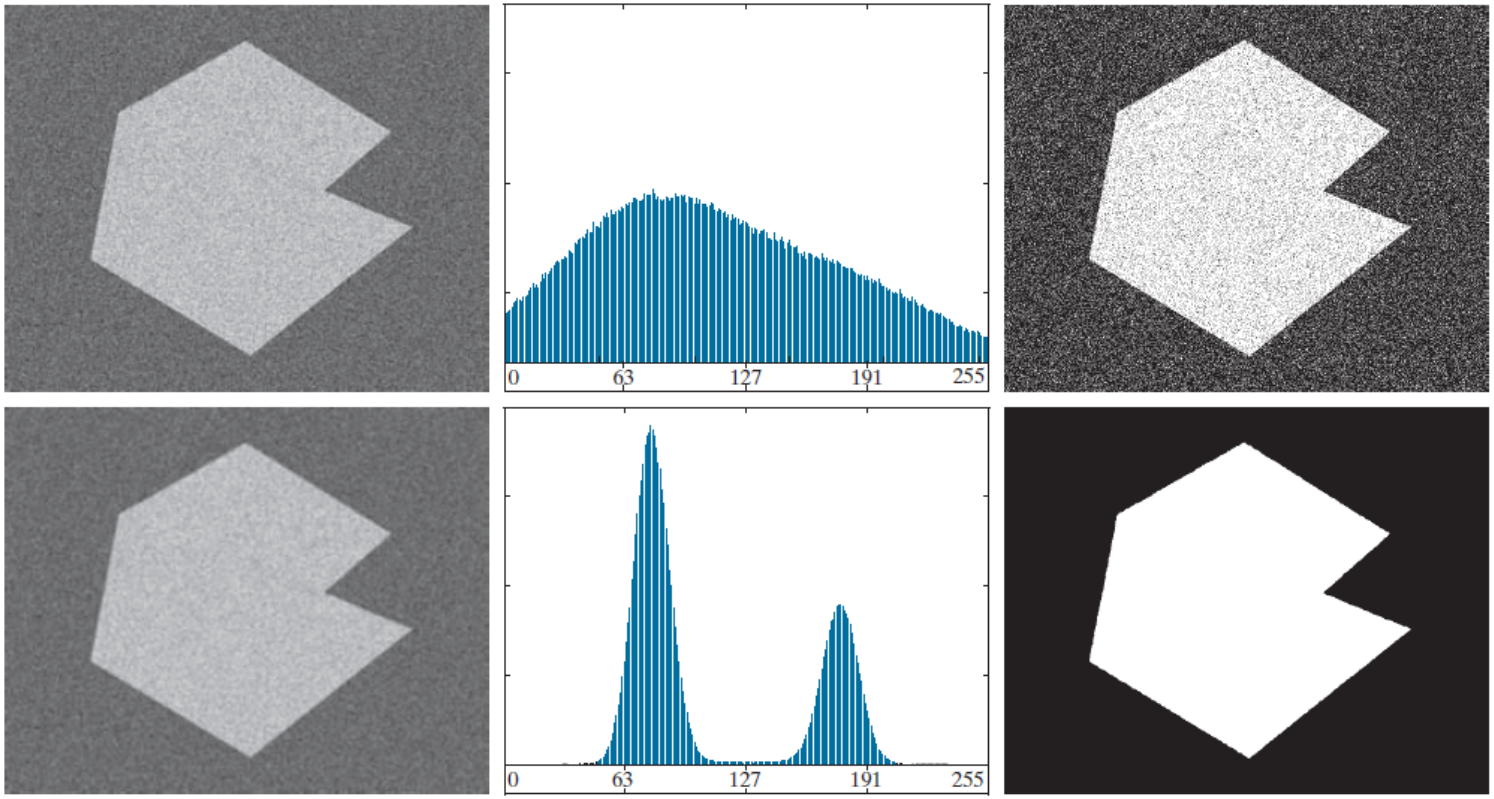
The role of noise in image thresholding



a b c
d e f

FIGURE 10.33 (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) through (f) Corresponding histograms.

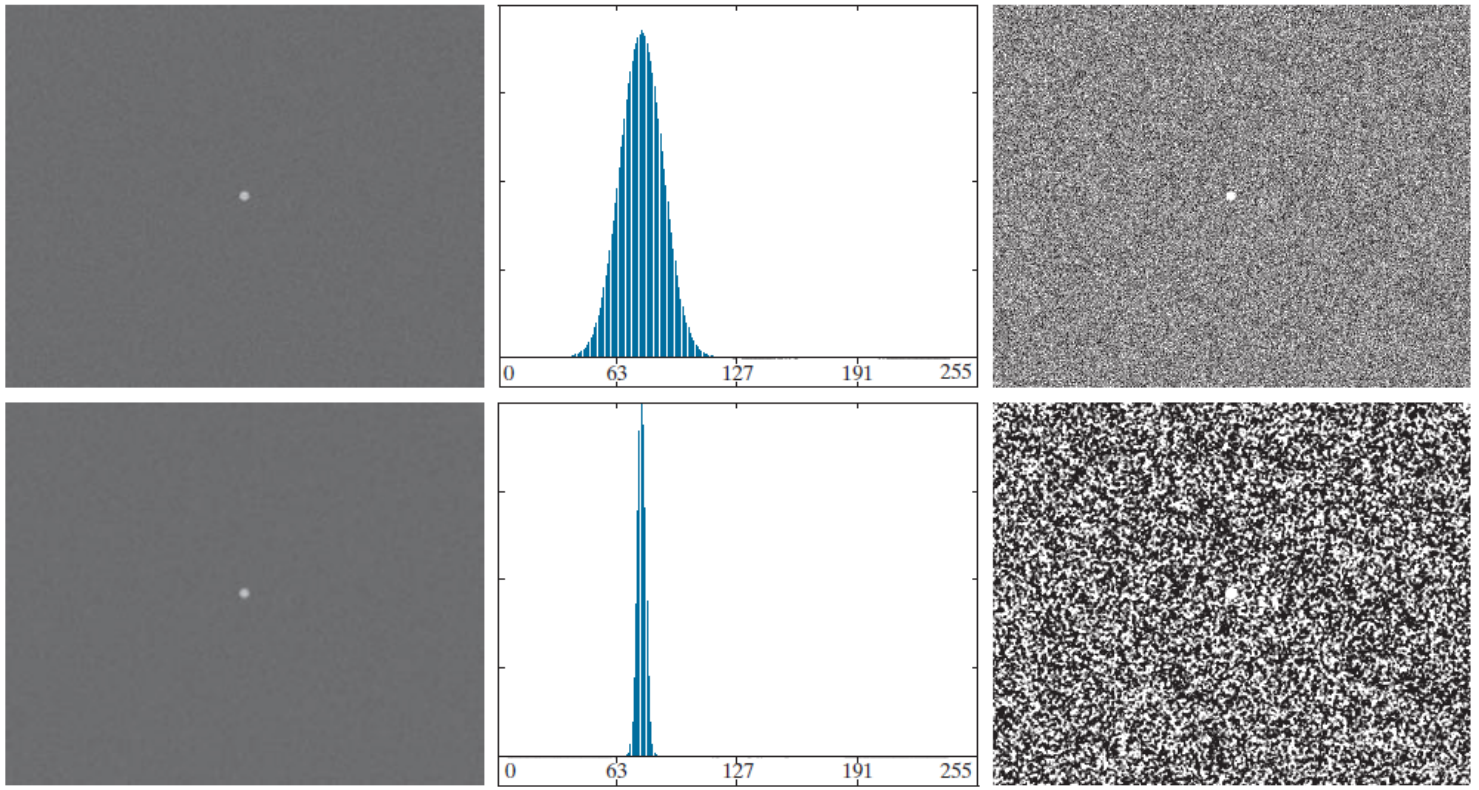
Using image smoothing to improve global thresholding



a b c
d e f

FIGURE 10.37 (a) Noisy image from Fig. 10.33(c) and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method.

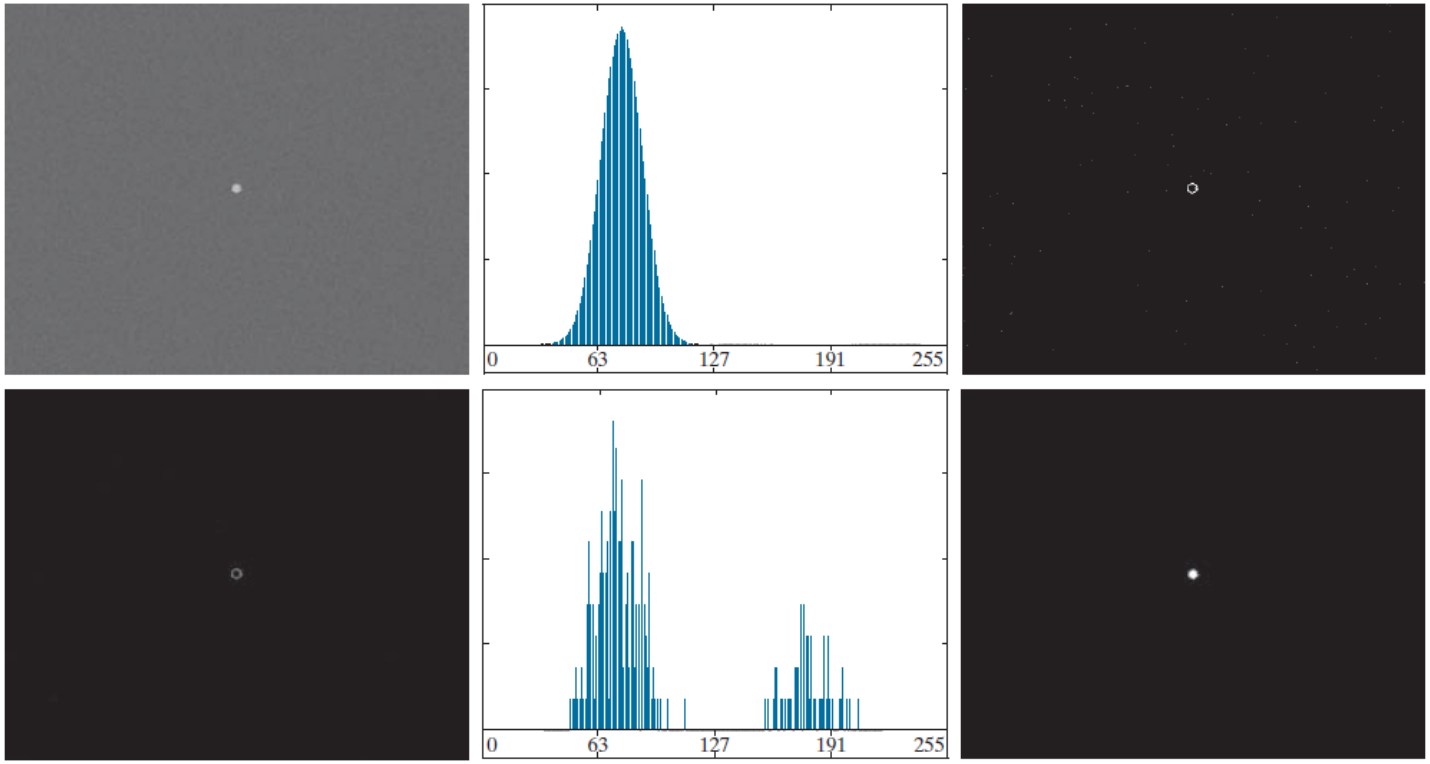
Using image smoothing to improve global thresholding



a b c
d e f

FIGURE 10.38 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases to extract the object of interest. (See Fig. 10.39 for a better solution.)

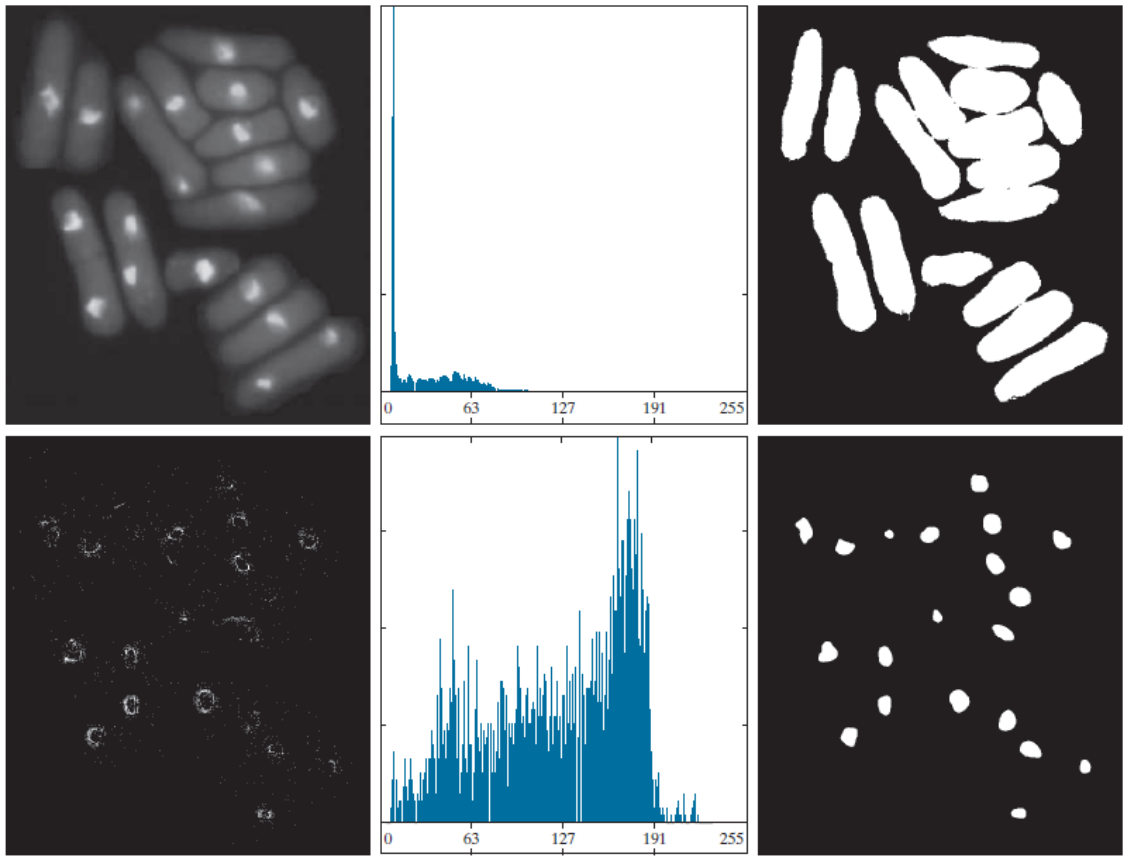
Using edges to improve global thresholding



a b c
d e f

FIGURE 10.39 (a) Noisy image from Fig. 10.38(a) and (b) its histogram. (c) Mask image formed as the 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.

Using edges to improve global thresholding



a b c
d e f

FIGURE 10.40 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Mask image formed by thresholding the absolute Laplacian image. (e) Histogram of the non-zero 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.)

Clustering

Region segmentation using k-means clustering

Given a set $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_Q\}$ of vector observation and a specified value of k, the algorithm is as follows:

1. ***Initialize the algorithm:*** Specify an initial set of means, $\mathbf{m}_i(1)$, $i = 1, 2, \ldots, k$.
2. ***Assign samples to clusters:*** Assign each sample to the cluster set whose mean is the closest (ties are resolved arbitrarily, but samples are assigned to only *one* cluster):

$$\mathbf{z}_q \rightarrow C_i \quad \text{if} \quad \|\mathbf{z}_q - \mathbf{m}_i\|^2 < \|\mathbf{z}_q - \mathbf{m}_j\|^2 \quad j = 1, 2, \ldots, k \; (j \neq i); \quad q = 1, 2, \ldots, Q$$

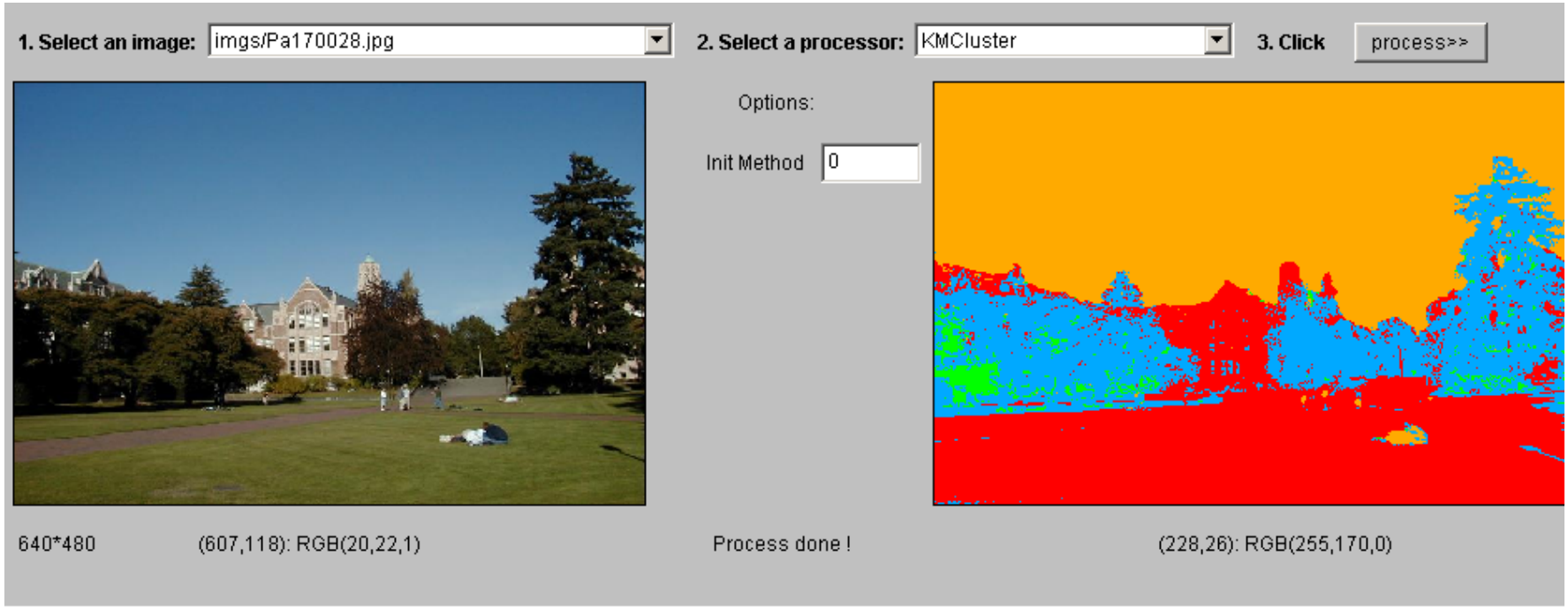
3. ***Update the cluster centers (means):***

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \qquad i = 1, 2, \ldots, k$$

where $|C_i|$ is the number of samples in cluster set C_i.

4. ***Test for completion:*** Compute the Euclidean norms of the differences between the mean vectors in the current and previous steps. Compute the residual error, E, as the sum of the k norms. Stop if $E \leq T$, where T a specified, nonnegative threshold. Else, go back to Step 2.

Region segmentation using k-means clustering

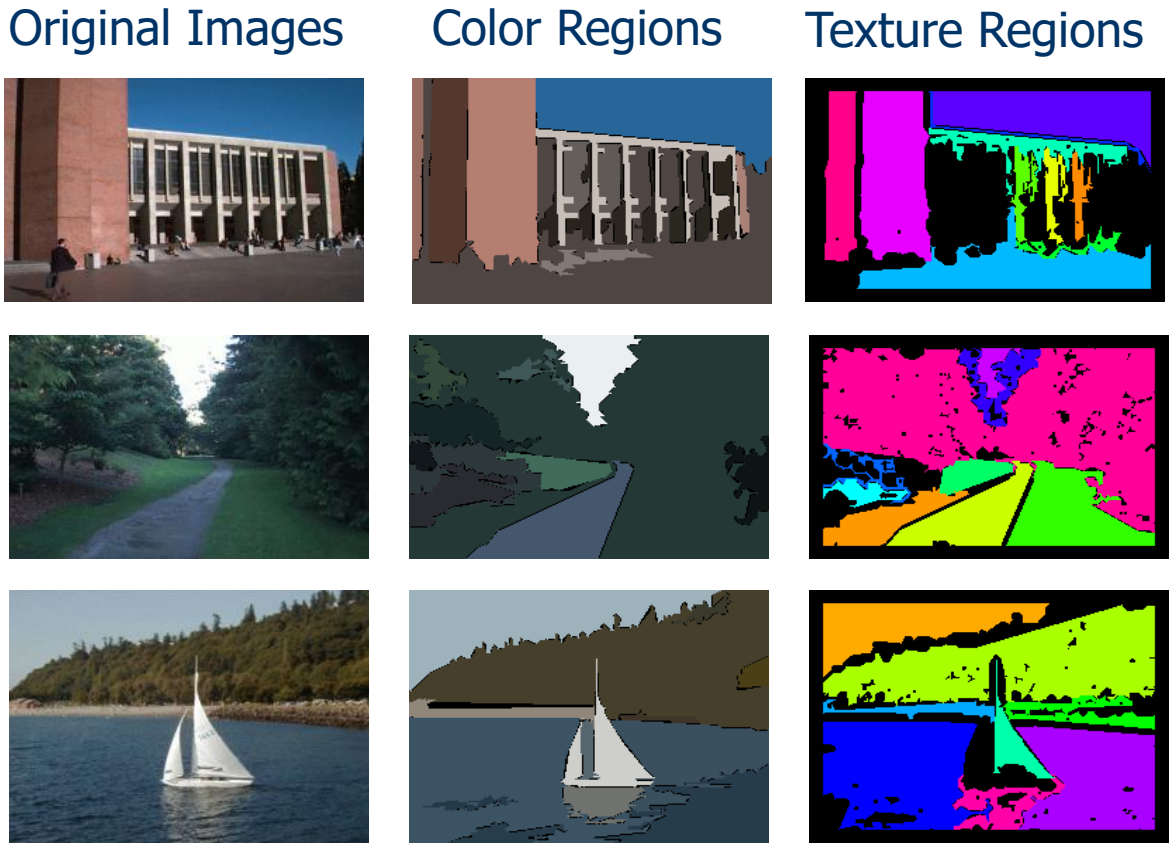


K-means clustering of color

Slide credit: S. Aksoy

Region segmentation using k-means clustering

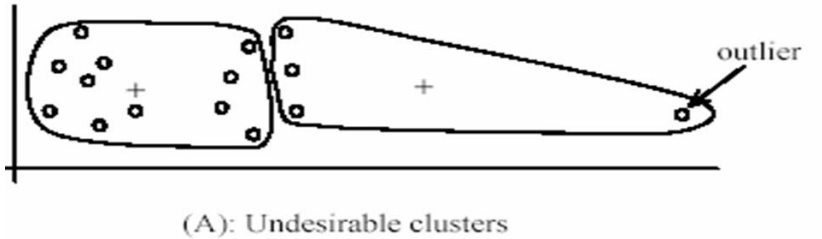
- Clustering can also be used with other features (e.g., texture) in addition to color.



Slide credit: S. Aksoy

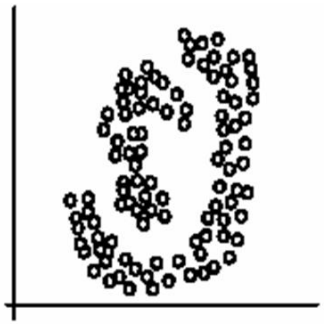
Region segmentation using k-means clustering

- Pros of k-means:
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error
- Cons of k-means:
  - Setting k
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters

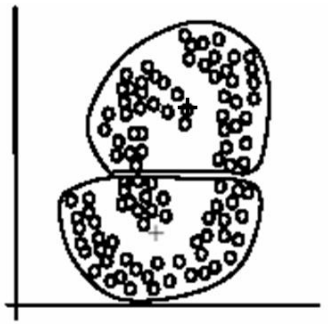


(A): Undesirable clusters

(B): Ideal clusters



(A): Two natural clusters



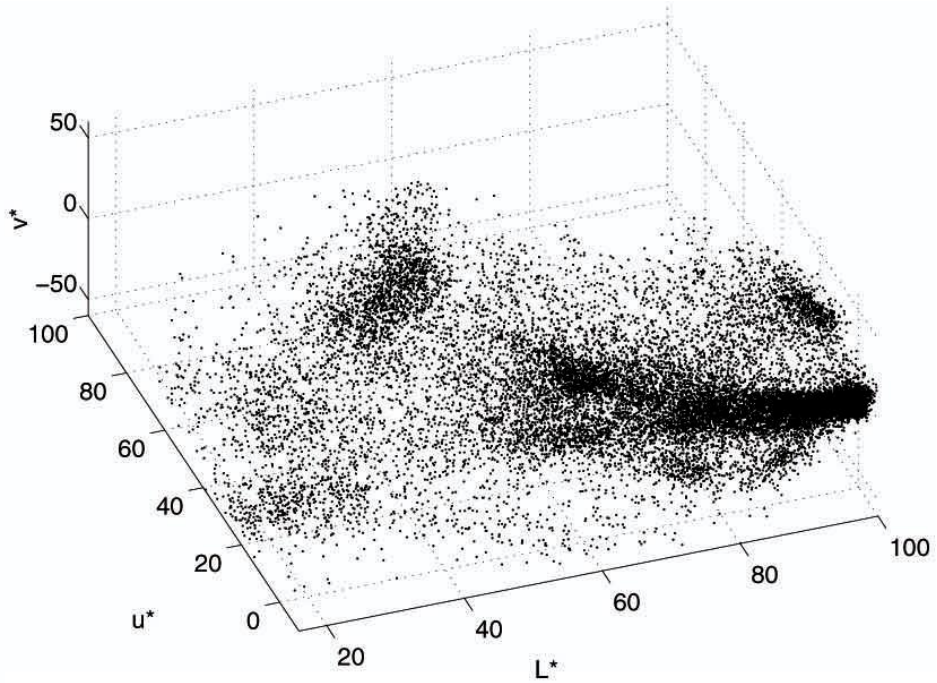
(B): *k*-means clusters

Slide credit: K. Grauman

Mean shift algorithm

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

Image

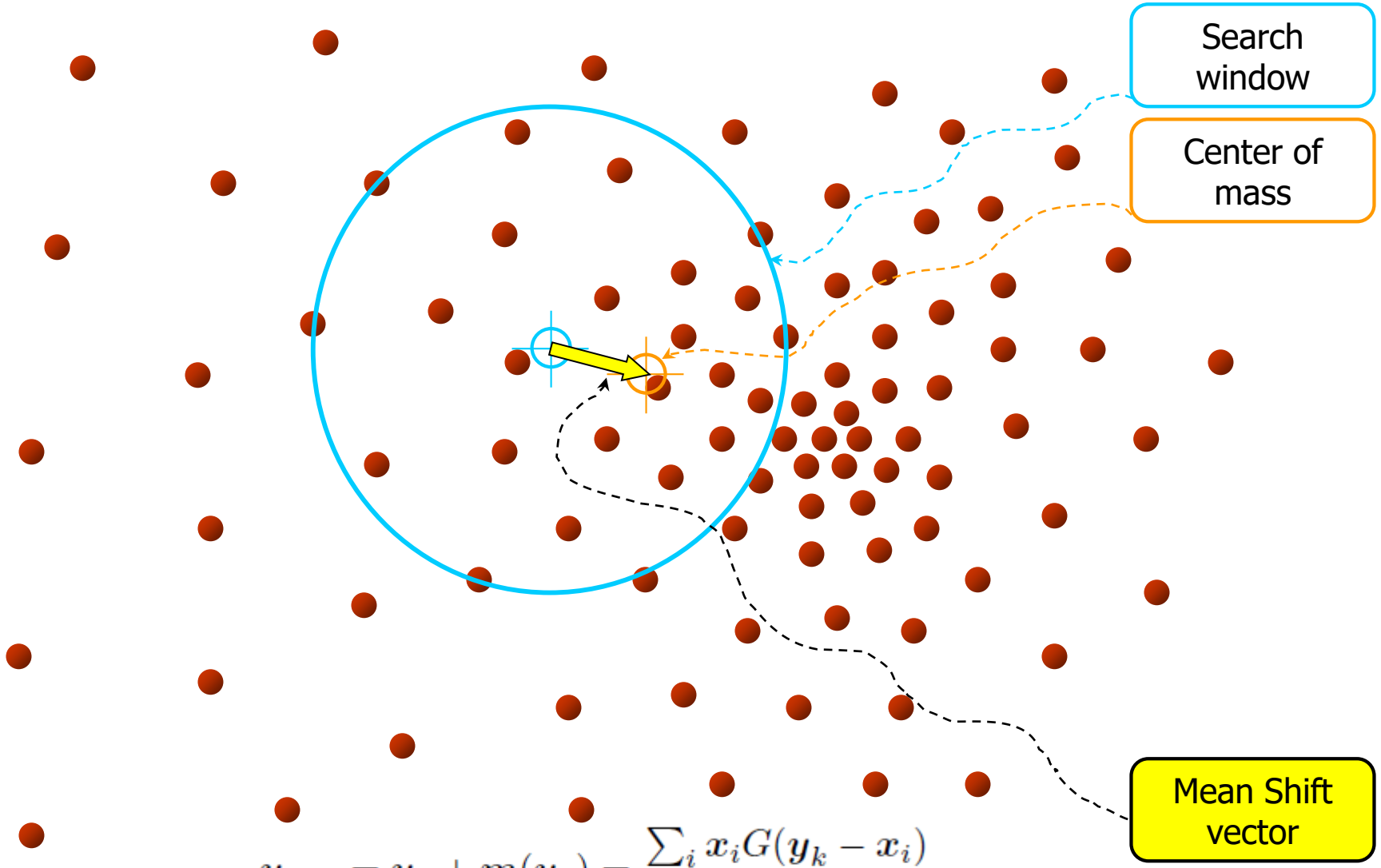


Feature space
(L*u*v* color values)

Adapted from Kristen Grauman

Slide credit: S. Aksoy

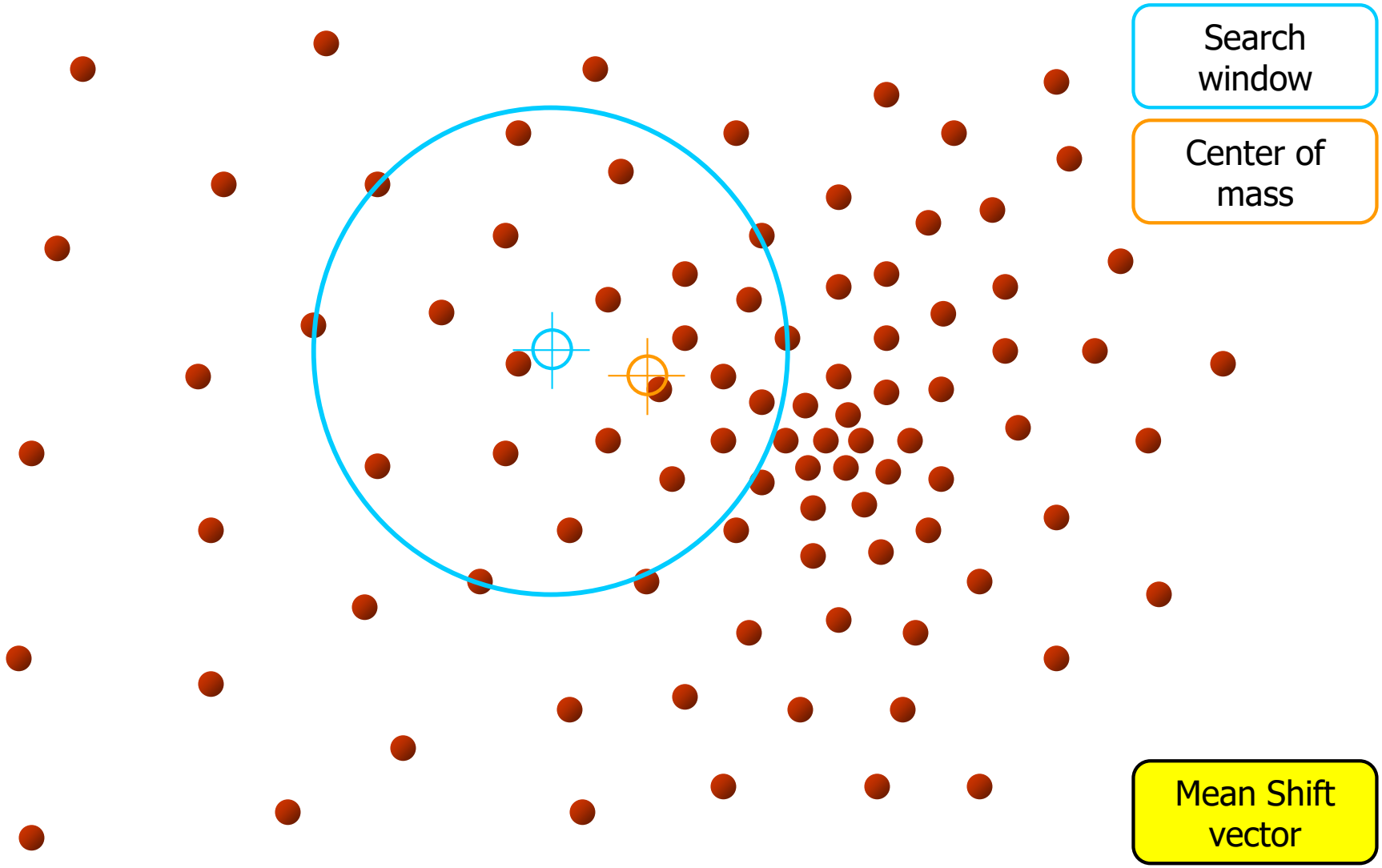
Mean shift algorithm



$$y_{k+1} = y_k + m(y_k) = \frac{\sum_i x_i G(y_k - x_i)}{\sum_i G(y_k - x_i)}$$

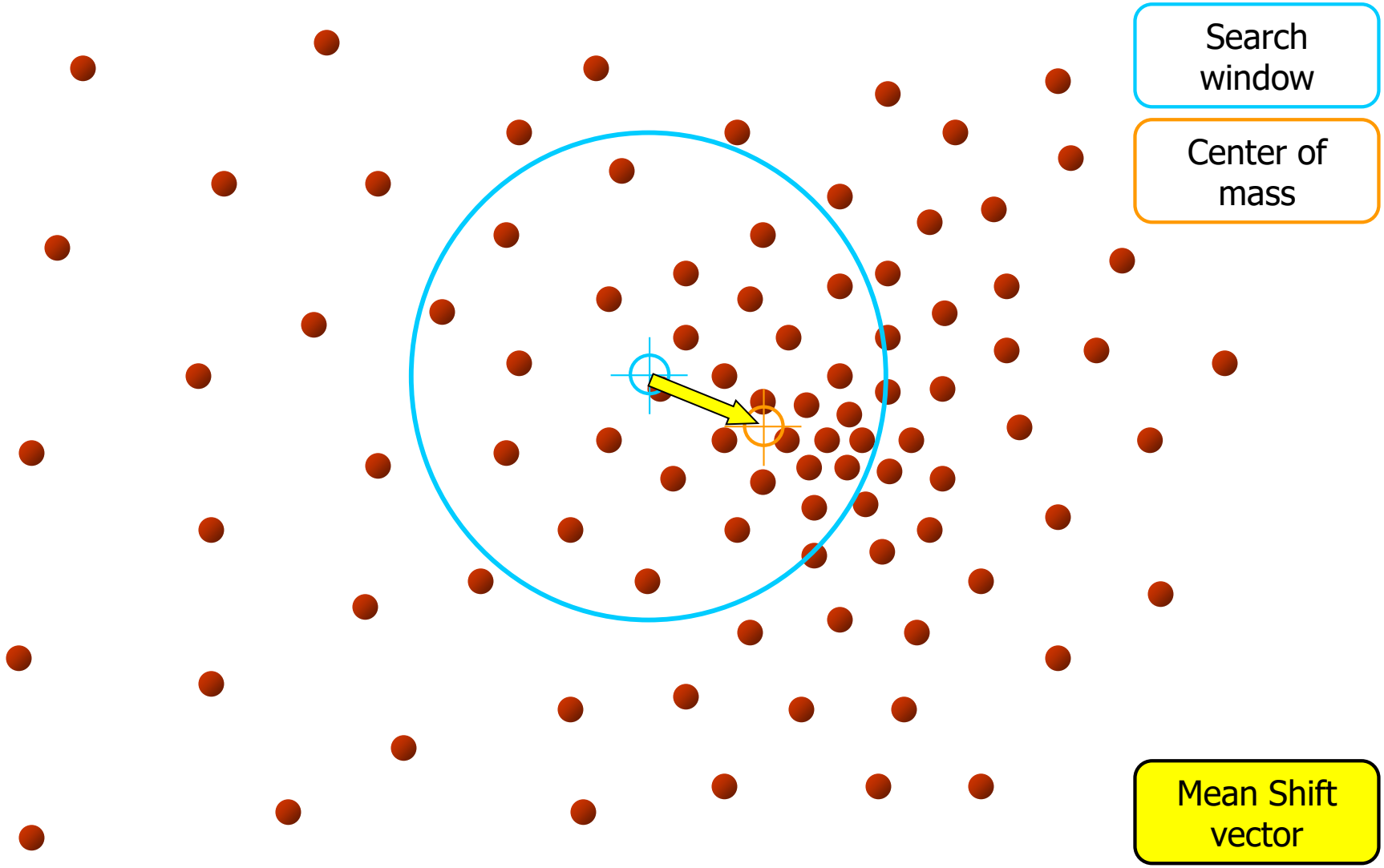
Slide credit: S. Aksoy

Mean shift algorithm



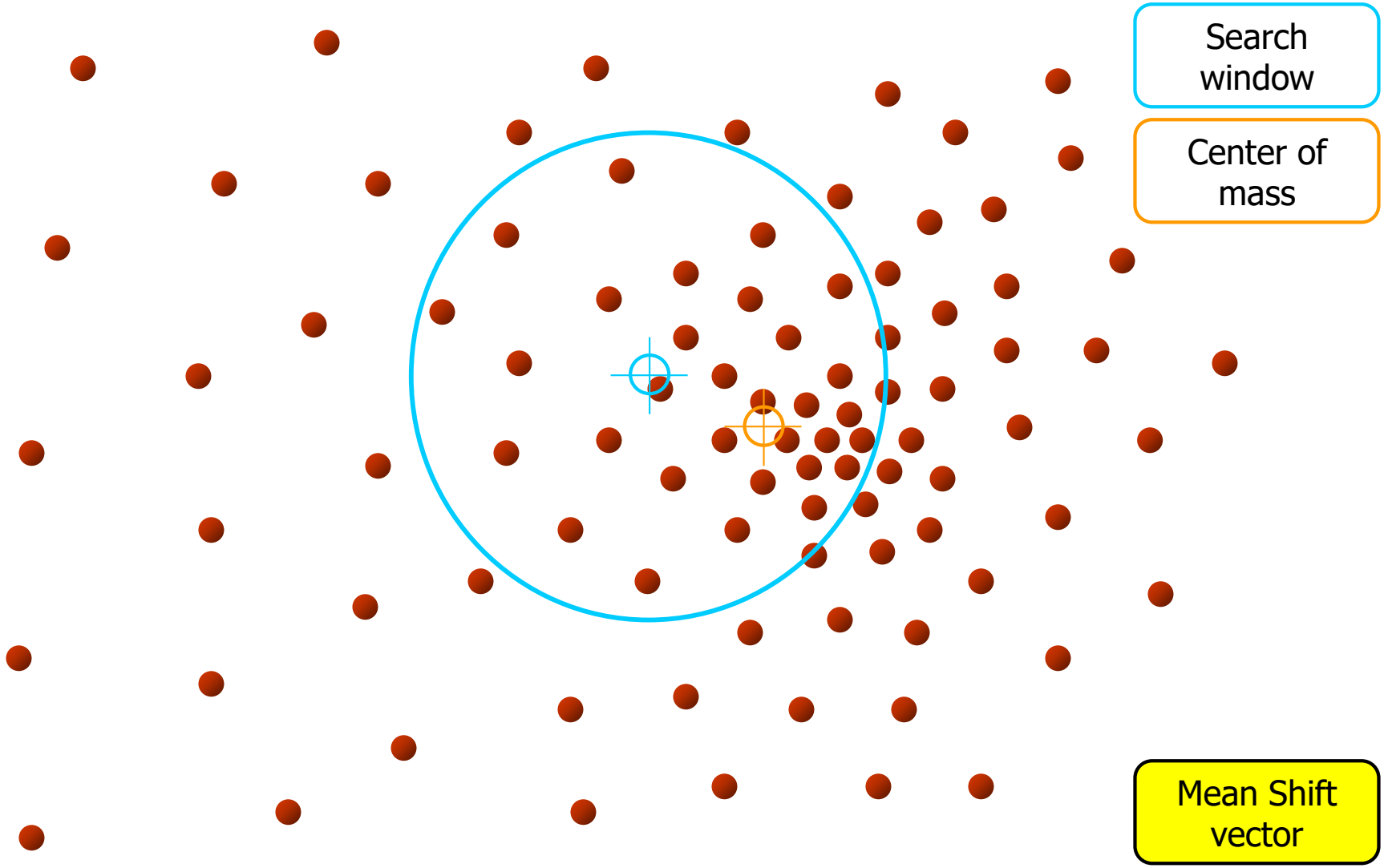
Slide credit: S. Aksoy

Mean shift algorithm



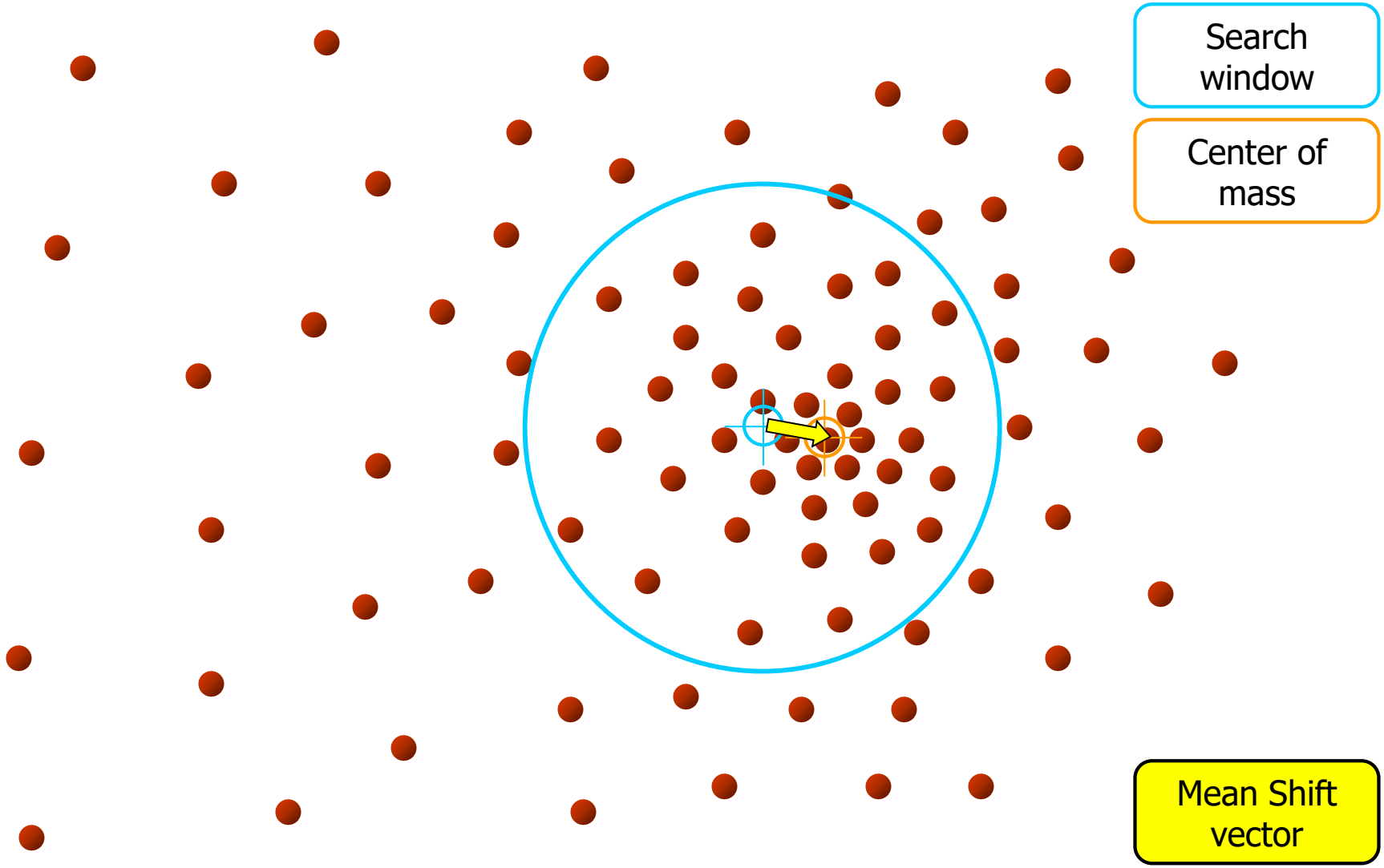
Slide credit: S. Aksoy

Mean shift algorithm



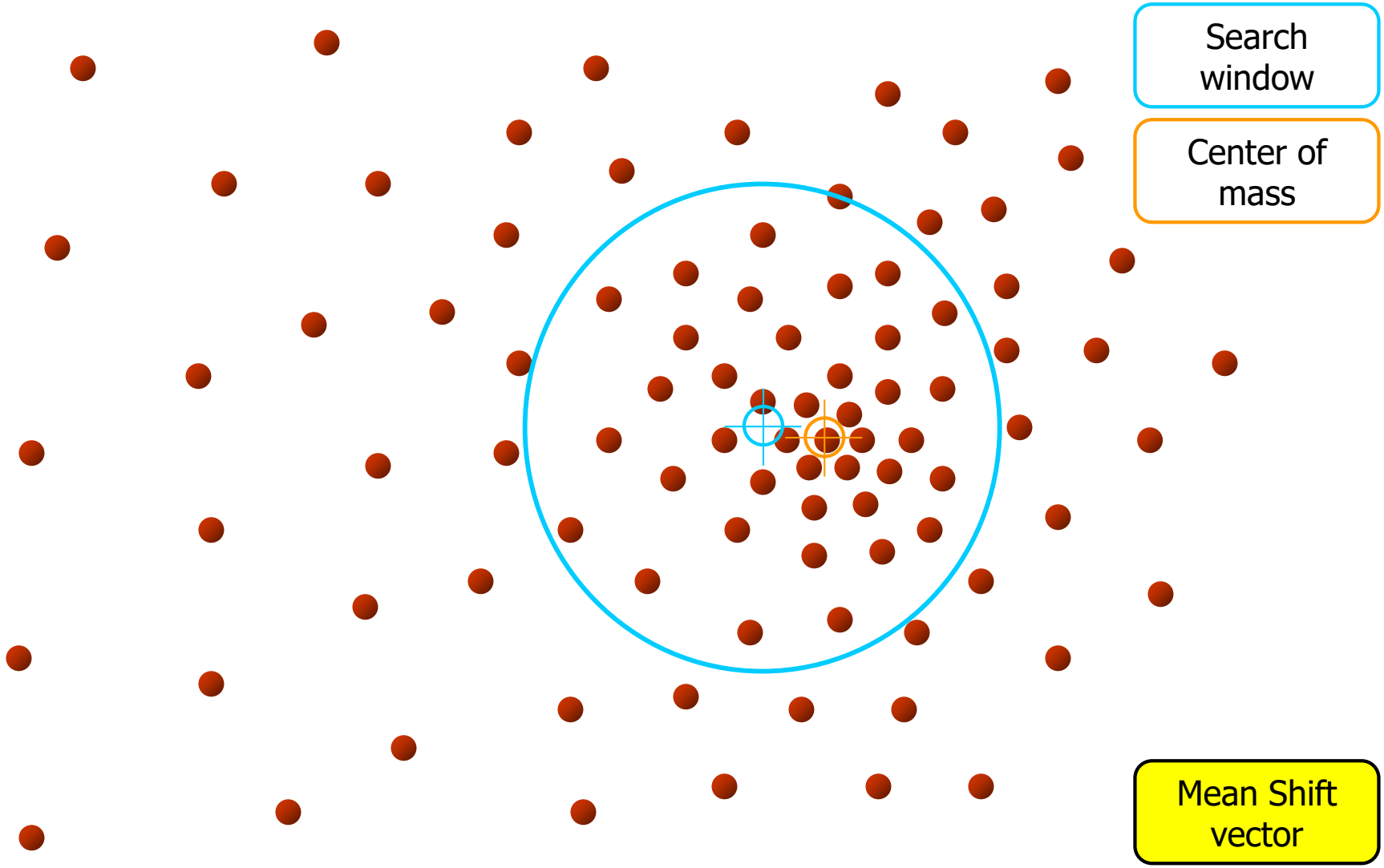
Slide credit: S. Aksoy

Mean shift algorithm



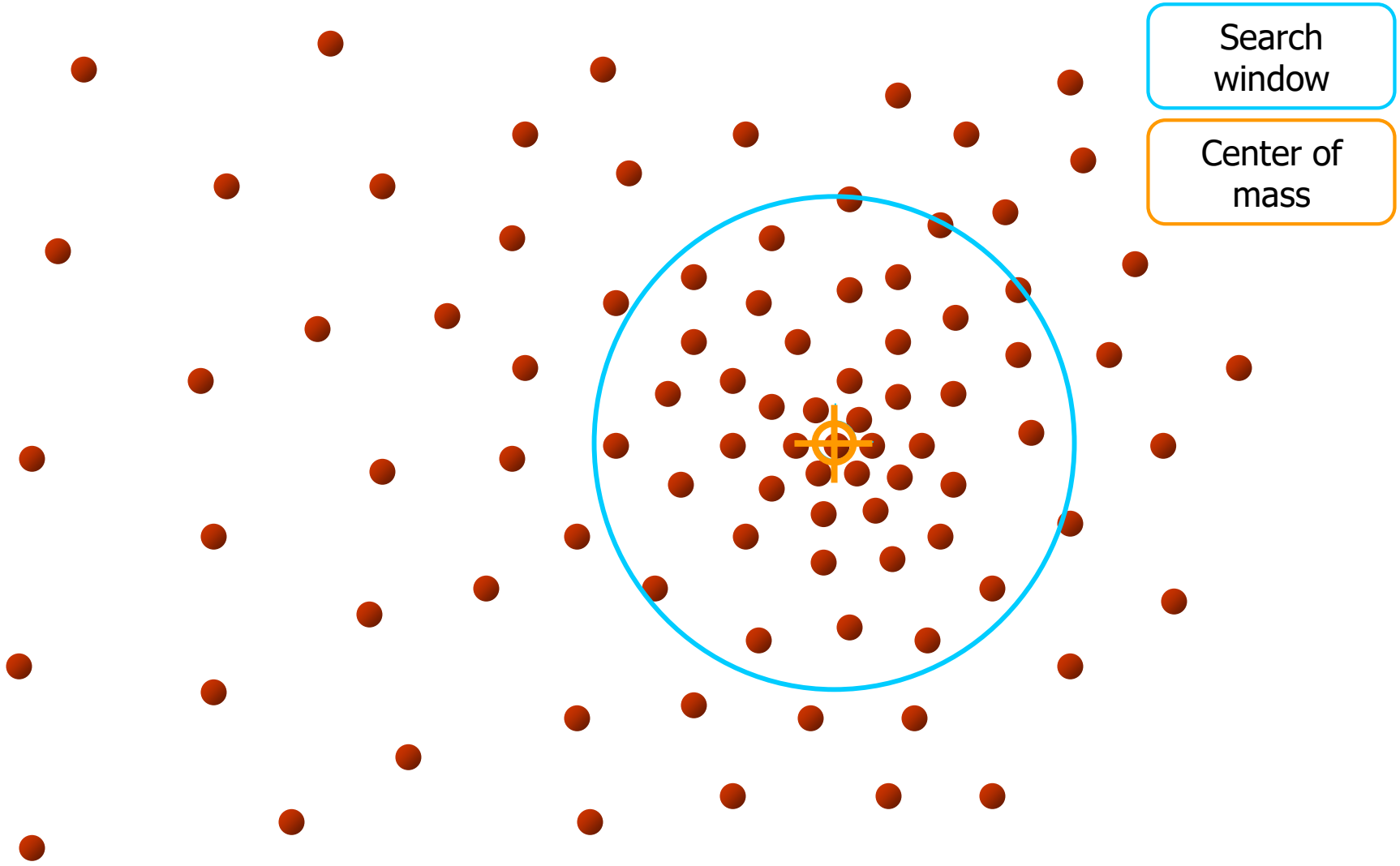
Slide credit: S. Aksoy

Mean shift algorithm



Slide credit: S. Aksoy

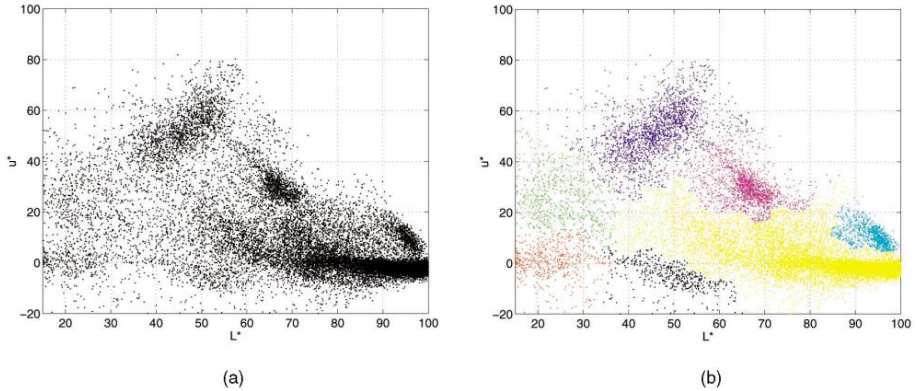
Mean shift algorithm

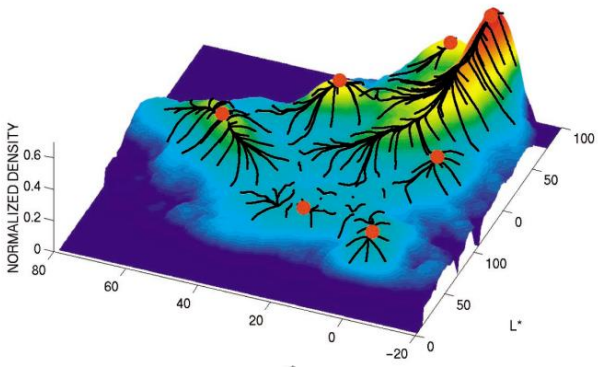


Slide credit: S. Aksoy

Mean shift clustering/segmentation

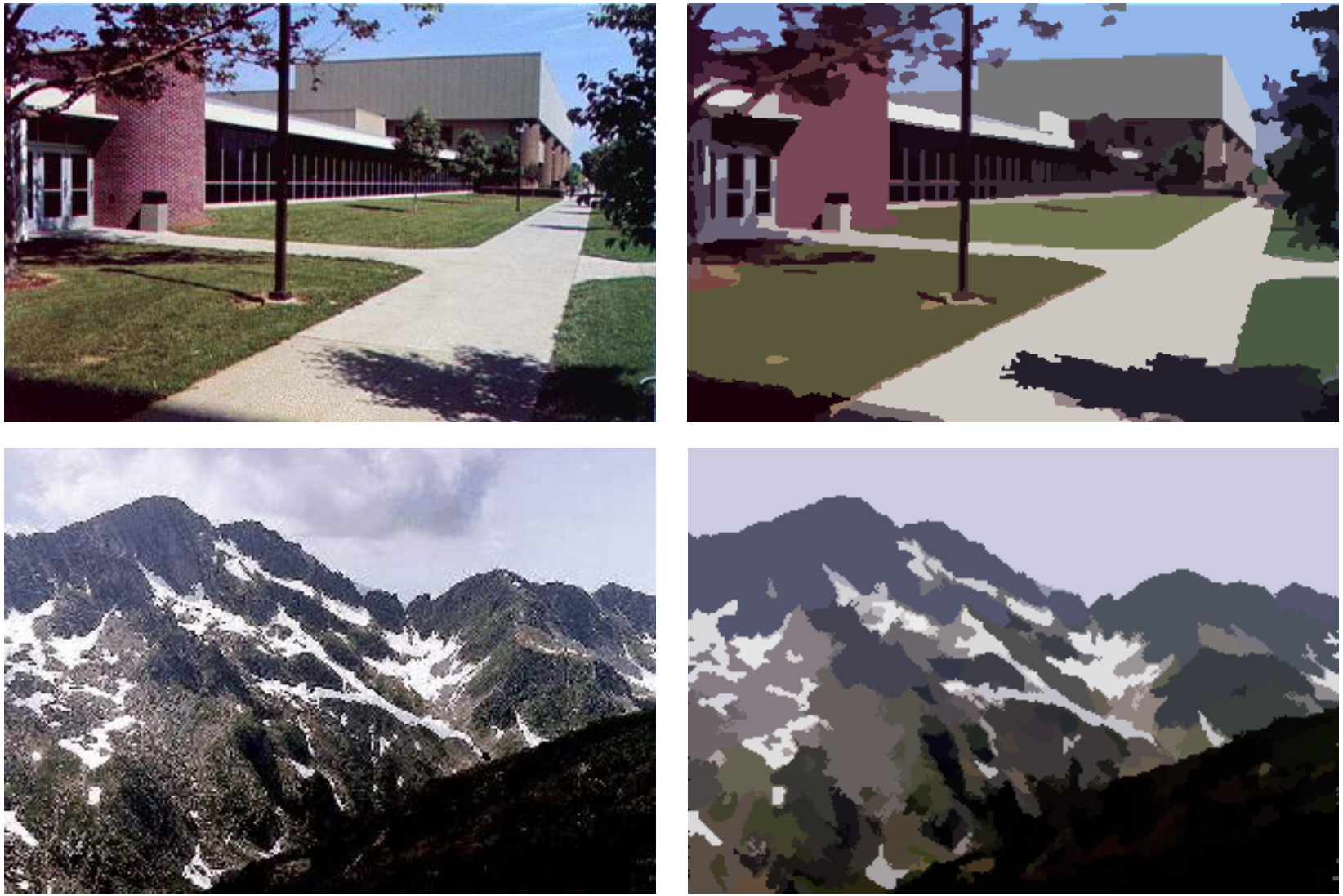
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same "peak" or mode





Slide credit: S. Aksoy

Mean shift segmentation



Slide credit: S. Aksoy

Mean shift segmentation

- Pros:
  - Does not assume shape on clusters
  - One parameter choice (window size)
  - Generic technique
  - Find multiple modes
- Cons:
  - Selection of window size
  - Does not scale well with dimension of feature space

Adapted from Kristen Grauman

Slide credit: S. Aksoy

Superpixels

Region segmentation using superpixels

- The idea behind superpixels is to replace the standard pixel grid by grouping pixels into primitive regions that are more perceptually meaningful than individual pixels.
- The objectives are
  - to lessen computational load
  - to improve the performance of segmentation algorithms by reducing irrelevant detail
- Desirable properties:
  - Good adherence to object boundaries
  - Regular shape and similar size
  - Fast to compute and simple to use

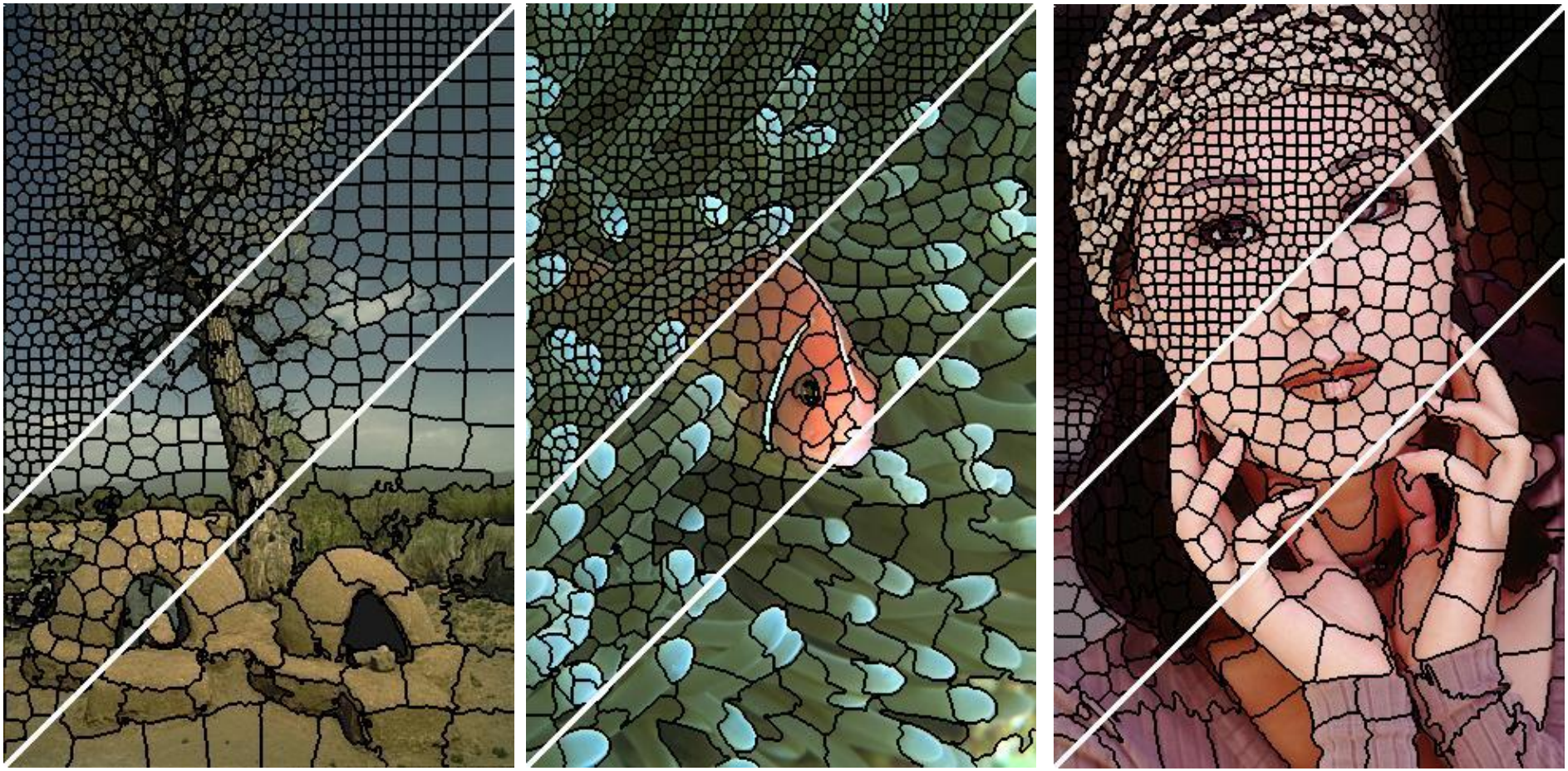
Superpixels

- Start from rough initial clusters and iteratively refine them until some convergence criterion is met.
- Simple linear iterative clustering (SLIC):
  1. Initialize cluster centers on pixel grid in steps S.
     - Features: Lab color, x-y position
  2. Move centers to position in 3x3 window with smallest gradient.
  3. Compare each pixel to cluster center within 2S pixel distance and assign to nearest.
  4. Recompute cluster centers as mean color/position of pixels belonging to each cluster.
  5. Stop when residual error is small.

Adapted from Derek Hoiem

Slide credit: S. Aksoy

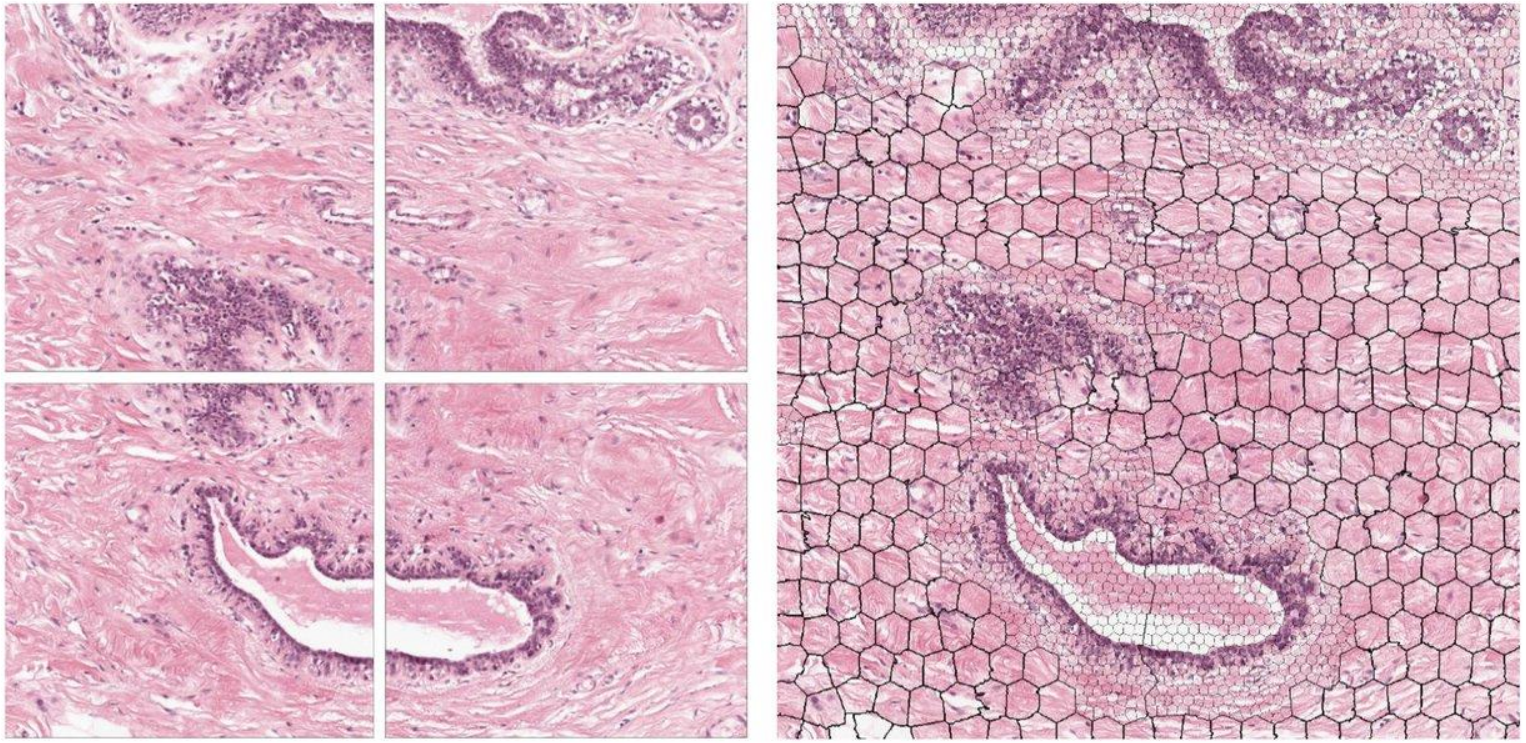
Superpixels



http://ivrl.epfl.ch/research/superpixels

Slide credit: S. Aksoy

Superpixels



Multi-scale superpixel segmentation of a breast biopsy image.

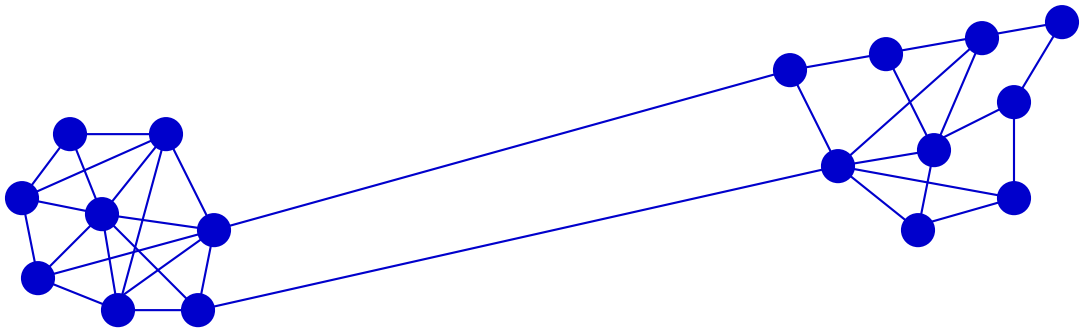
B. E. Bejnordi, G. Litjens, M. Hermsen, N. Karssemeijer, and J. A. van der Laak, "A multi-scale superpixel classification approach to the detection of regions of interest in whole slide histopathology images," SPIE Medical Imaging Symposium, 2015.

Slide credit: S. Aksoy

Graph cuts

Graph-based segmentation

- An image is represented by a graph whose nodes are pixels or small groups of pixels.
- The goal is to partition the nodes into disjoint sets so that the similarity within each set is high and across different sets is low.



Slide credit: S. Aksoy

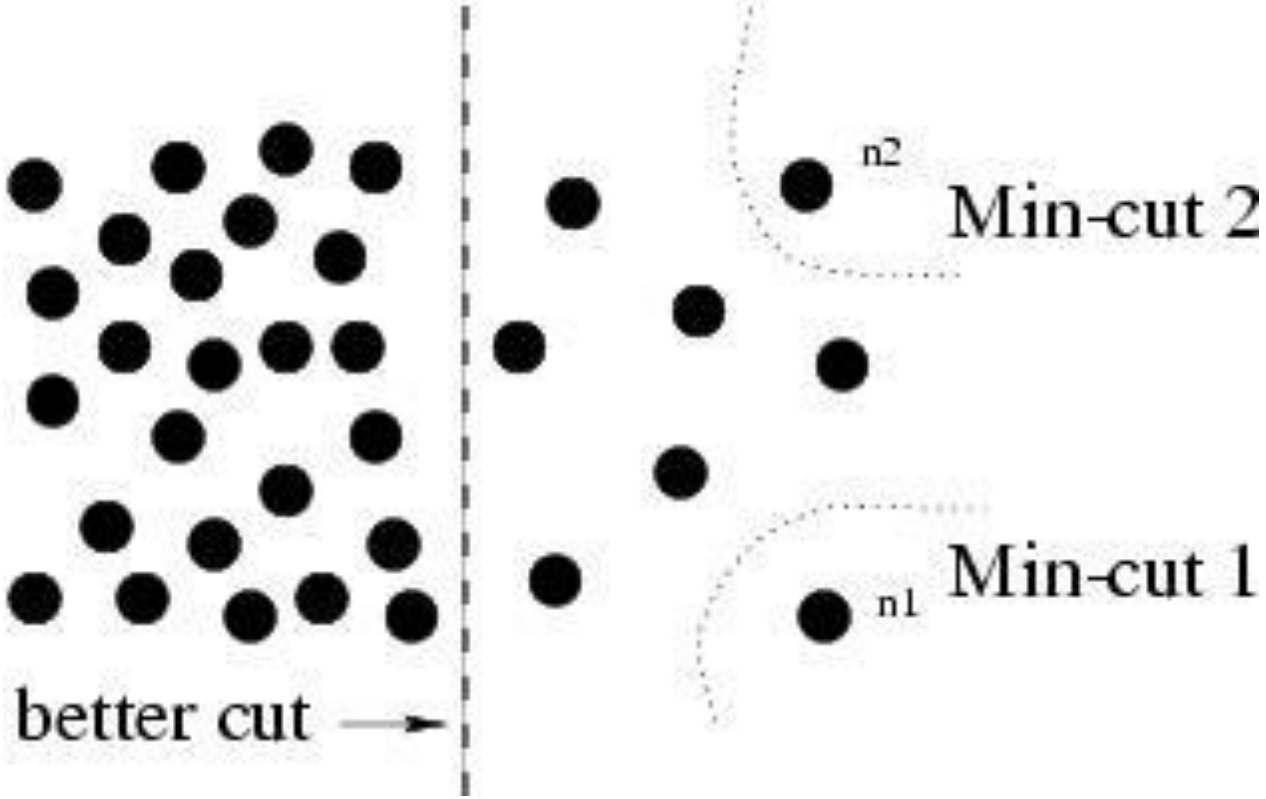
Graph-based segmentation

- Let G = (V,E) be a graph. Each edge (u,v) has a weight w(u,v) that represents the similarity between u and v.
- Graph G can be broken into 2 disjoint graphs with node sets A and B by removing edges that connect these sets.
- Let $\text{cut}(A,B) = \sum_{u \in A,\, v \in B} w(u,v)$.
- One way to segment G is to find the minimum cut.

Slide credit: S. Aksoy

Graph-based segmentation

- Problem with minimum cut: weight of cut proportional to number of edges in the cut; tends to produce small, isolated components.



Slide credit: S. Aksoy

Graph-based segmentation

- Shi and Malik proposed the normalized cut.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

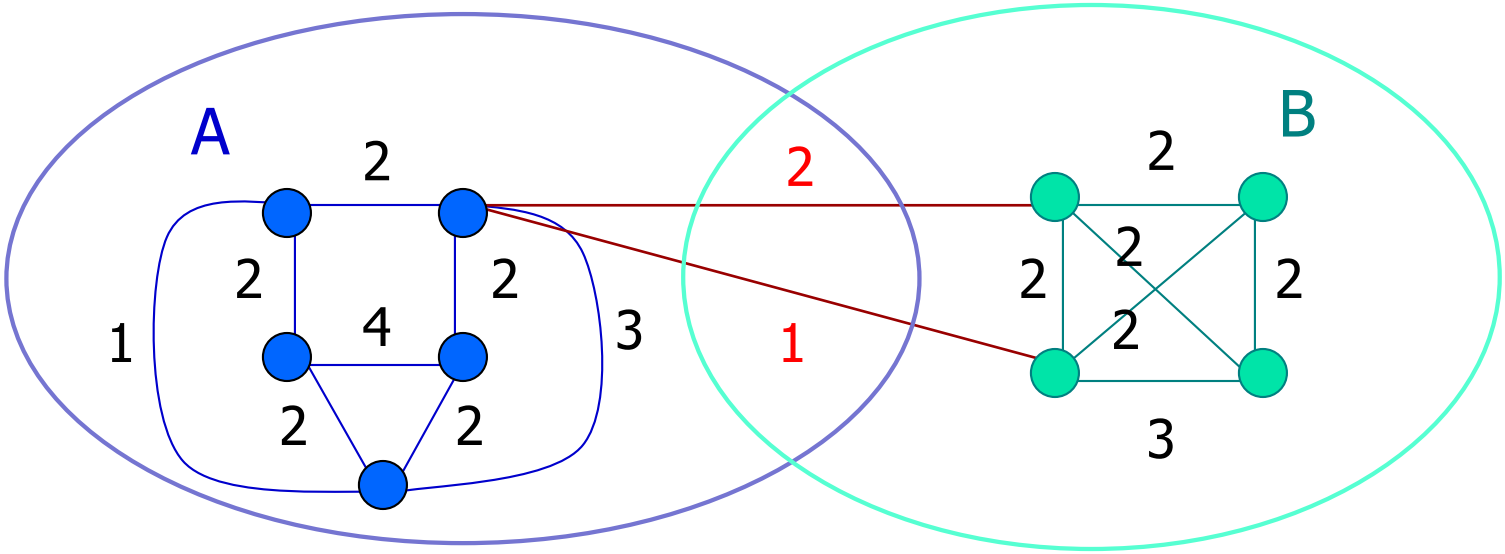
Normalized cut

$$assoc(A,V) = \sum_{u \in A,\, t \in V} w(u,t)$$

How much is A connected to the graph as a whole

Slide credit: S. Aksoy

Graph-based segmentation



$$\text{Ncut}(A,B) = \frac{3}{21} + \frac{3}{16}$$

Slide credit: S. Aksoy

Graph-based segmentation

- Shi and Malik turned graph cuts into an eigenvector/eigenvalue problem.
- Set up a weighted graph G=(V,E).
  - V is the set of (N) pixels.
  - E is a set of weighted edges (weight w_{ij} gives the similarity between nodes i and j).
  - Length N vector d: d_i is the sum of the weights from node i to all other nodes.
  - N x N matrix D: D is a diagonal matrix with d on its diagonal.
  - N x N symmetric matrix W: $W_{ij} = w_{ij}$.

Slide credit: S. Aksoy

Graph-based segmentation

- Let x be a characteristic vector of a set A of nodes.
  - x_i = 1 if node i is in a set A
  - x_i = -1 otherwise
- Let y be a continuous approximation to x

$$y = (1 + x) - \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i}(1 - x).$$

- Solve the system of equations

  $(D - W)\, y = \lambda\, D\, y$

  for the eigenvectors y and eigenvalues λ.
- Use the eigenvector y with second smallest eigenvalue to bipartition the graph (y → x → A).
- If further subdivision is merited, repeat recursively.

Slide credit: S. Aksoy

Graph-based segmentation

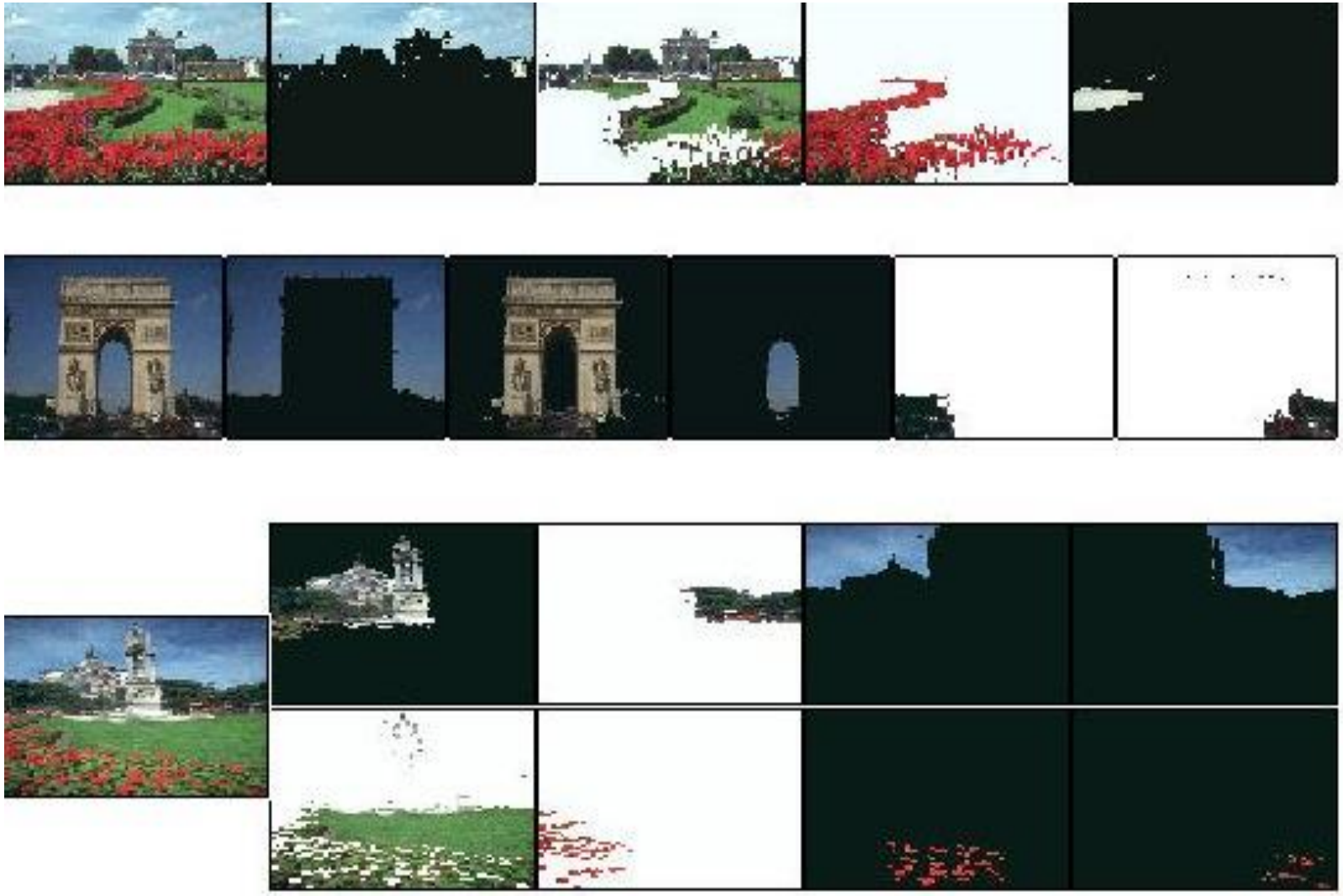
- Edge weights w(i,j) can be defined by

$$w(i,j) = e^{-||F(i)-F(j)||^2 / \sigma_I^2} * \begin{cases} e^{-||X(i)-X(j)||^2 / \sigma_X^2} & \text{if } ||X(i)-X(j)||^2 < r \\ 0 & \text{otherwise} \end{cases}$$

  where
  - X(i) is the spatial location of node I
  - F(i) is the feature vector for node I which can be intensity, color, texture, motion…
- The formula is set up so that w(i,j) is 0 for nodes that are too far apart.

Slide credit: S. Aksoy

Graph-based segmentation



Slide credit: S. Aksoy

Graph-based segmentation

- Pros:
  - Generic framework, flexible to choice of function that computes weights ("affinities") between nodes
  - Does not require model of the data distribution

- Cons:
  - Time complexity can be high
  - Dense, highly connected graphs $\rightarrow$ many affinity computations
  - Solving eigenvalue problem

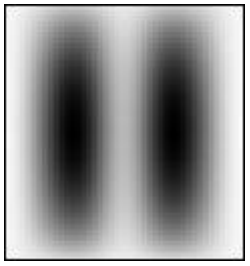
Adapted from Kristen Grauman

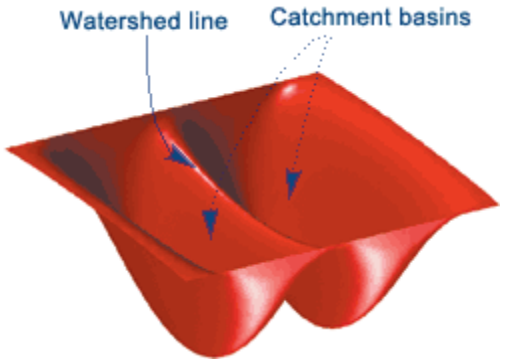
Slide credit: S. Aksoy

Morphological watersheds

Watershed segmentation

- The image can be interpreted as a topographic surface, with both valleys and mountains.
- Three types of points can be considered:
  - Points belonging to a regional minimum.
  - Points at which a drop of water, if placed at the location of any of those points, would fall to a single minimum.
    → catchment basins
  - Points at which water would be equally likely to fall to more than one such minimum.
    → watershed lines





Slide credit: S. Aksoy

Watershed segmentation

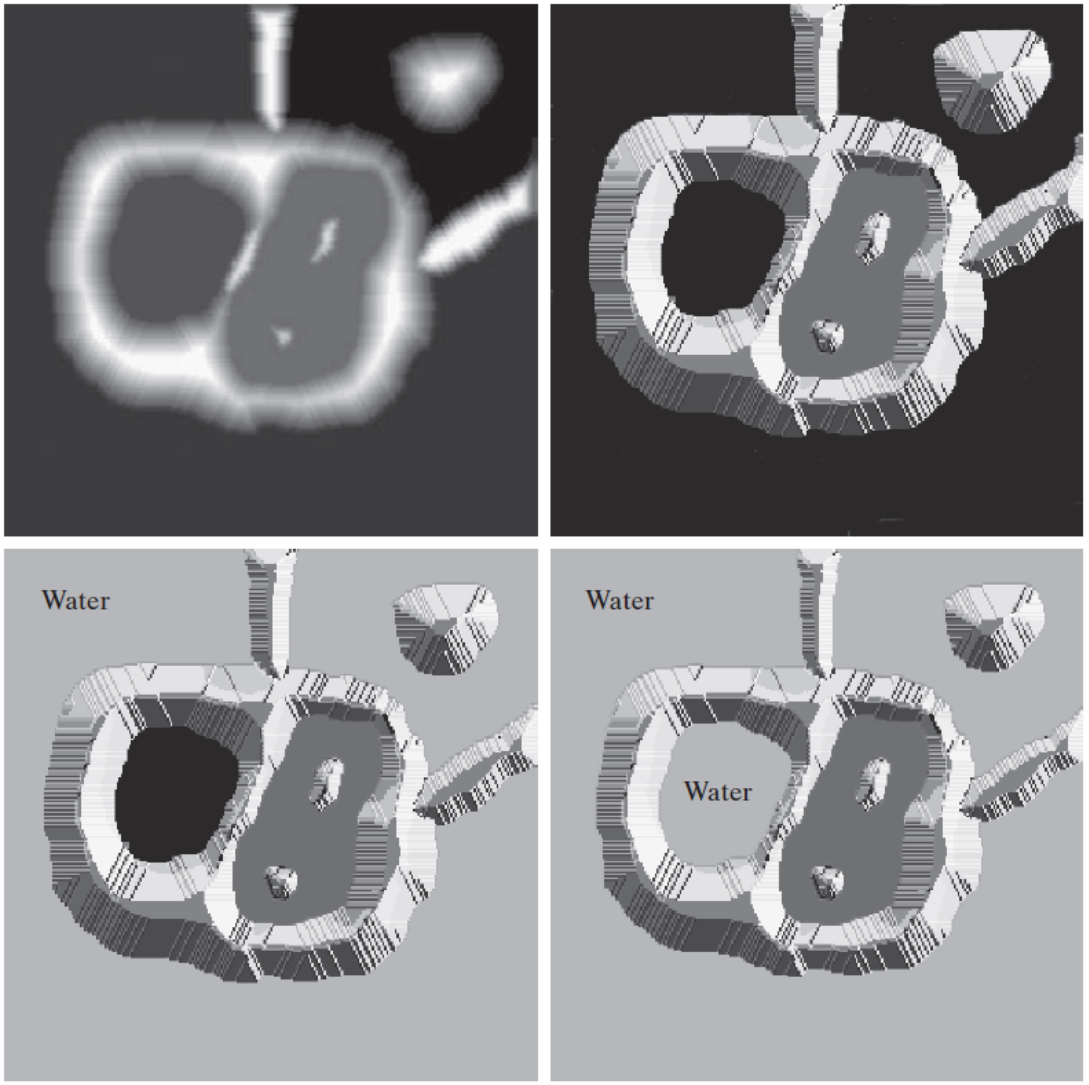
- Assume that there is a hole in each minima and the surface is immersed into a lake.
- The water will enter through the holes at the minima and flood the surface.
- To avoid the water coming from two different minima to meet, a dam is build whenever there would be a merge of the water.
- Finally, the only thing visible of the surface would be the dams. These dam walls are called the watershed lines.

Slide credit: S. Aksoy

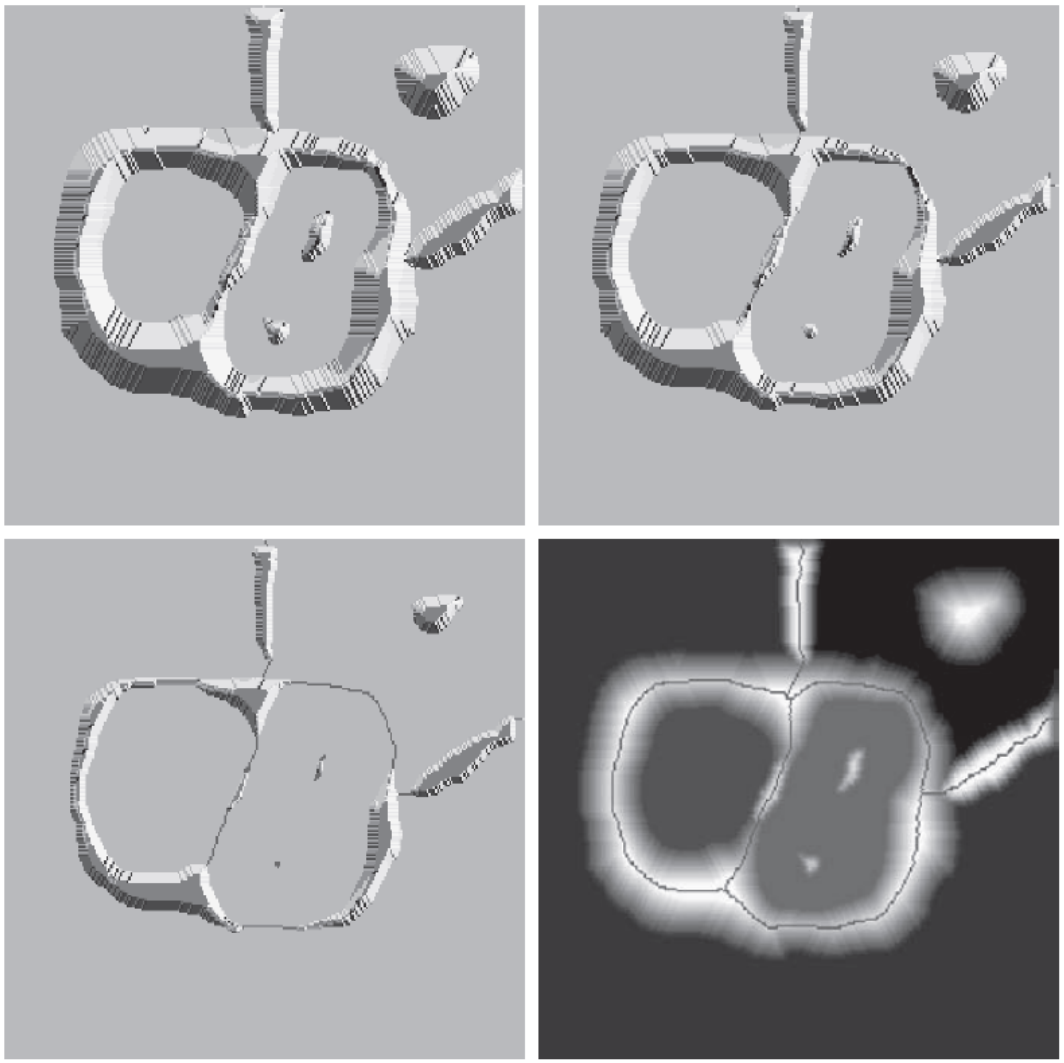
Watershed segmentation

a c
b d

FIGURE 10.57
(a) Original image.
(b) Topographic view. Only the background is *black*. The basin on the left is slightly lighter than black.
(c) and (d) Two stages of flooding. All constant dark values of gray are intensities in the original image. Only constant *light gray* represents "water."
(Courtesy of Dr. S. Beucher, CMM/ Ecole des Mines de Paris.)
(*Continued on next page*.)



Watershed segmentation

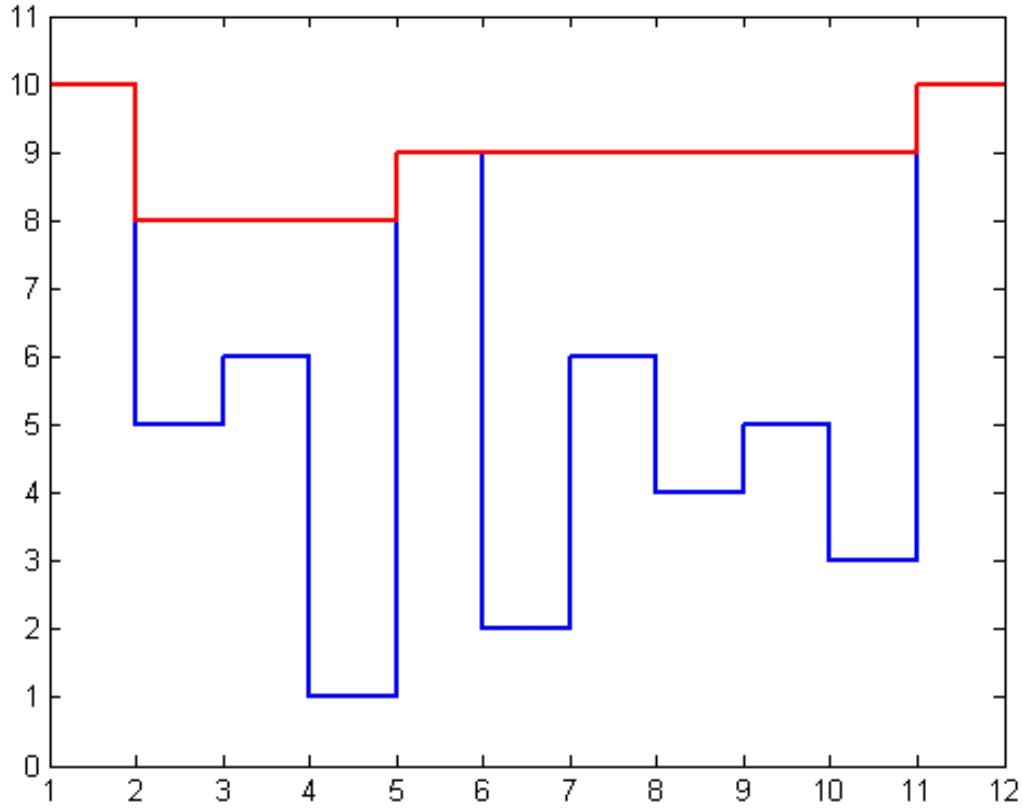


e f
g h

FIGURE 10.57
(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 superimposed on the original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed segmentation

- A multi-scale segmentation can be obtained by iteratively smoothing the topographic surface.



Slide credit: S. Aksoy

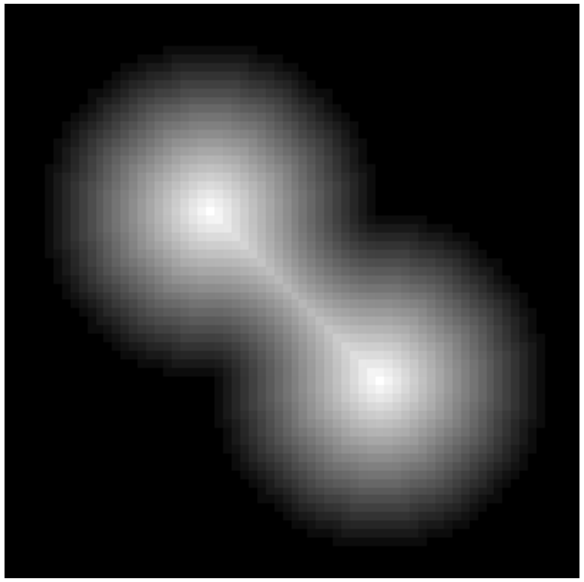
Watershed segmentation

- The key behind using the watershed transform for segmentation is this: change your image into another image whose catchment basins are the objects you want to identify.
- Examples:
  - Distance transform can be used with binary images where the catchment basins correspond to the foreground components of interest.
  - Gradient can be used with grayscale images where the catchment basins should theoretically correspond to the homogeneous grey level regions of the image.

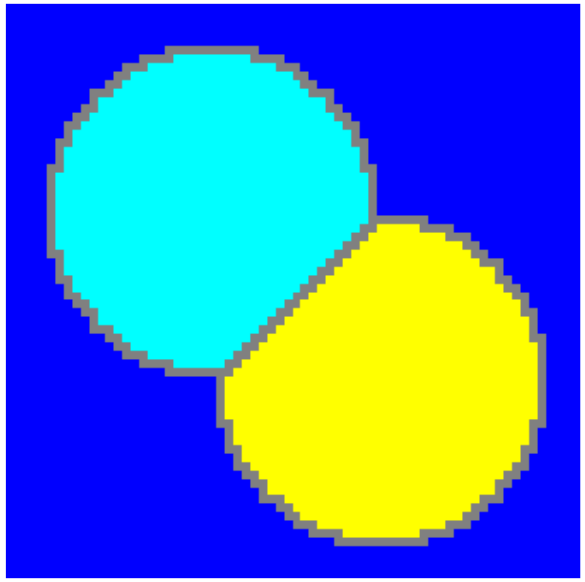
Slide credit: S. Aksoy

Watershed segmentation

Binary image.



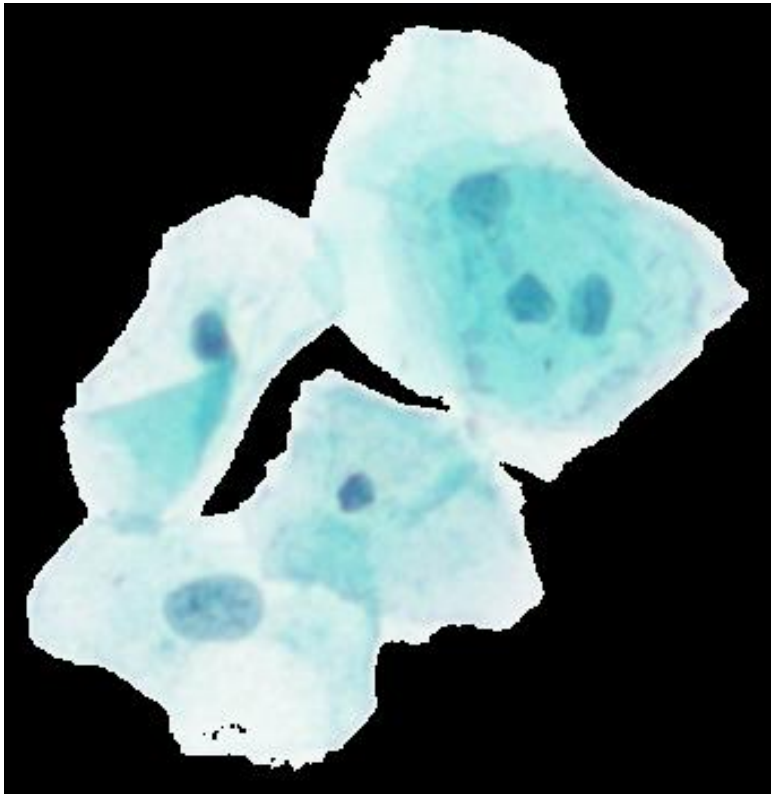
Distance transform of the complement of the binary image.



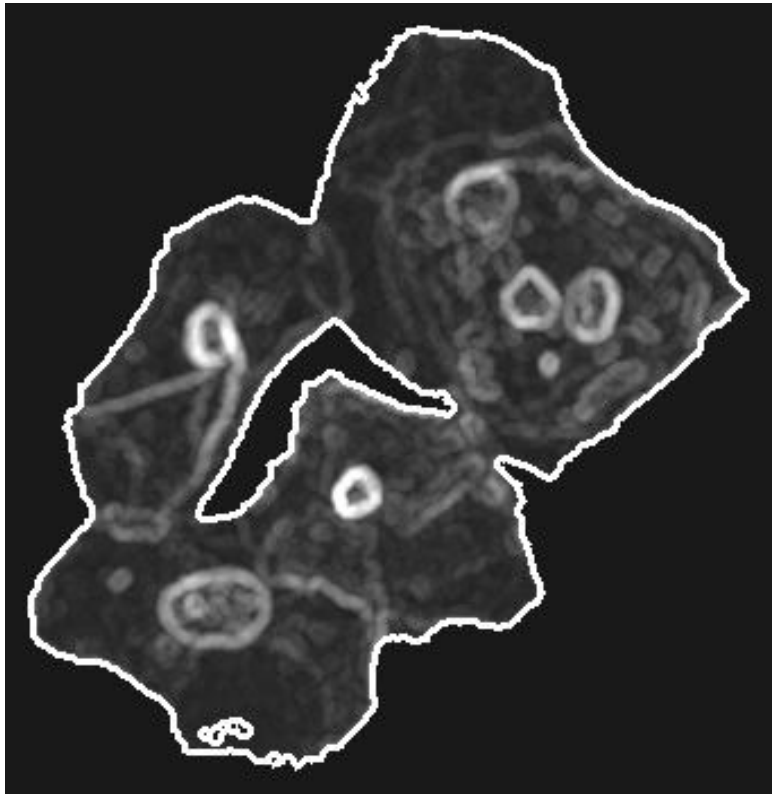
Watershed transform after complementing the distance transform, and forcing pixels that do not belong to the objects to be at –Inf.

Slide credit: S. Aksoy

Watershed segmentation



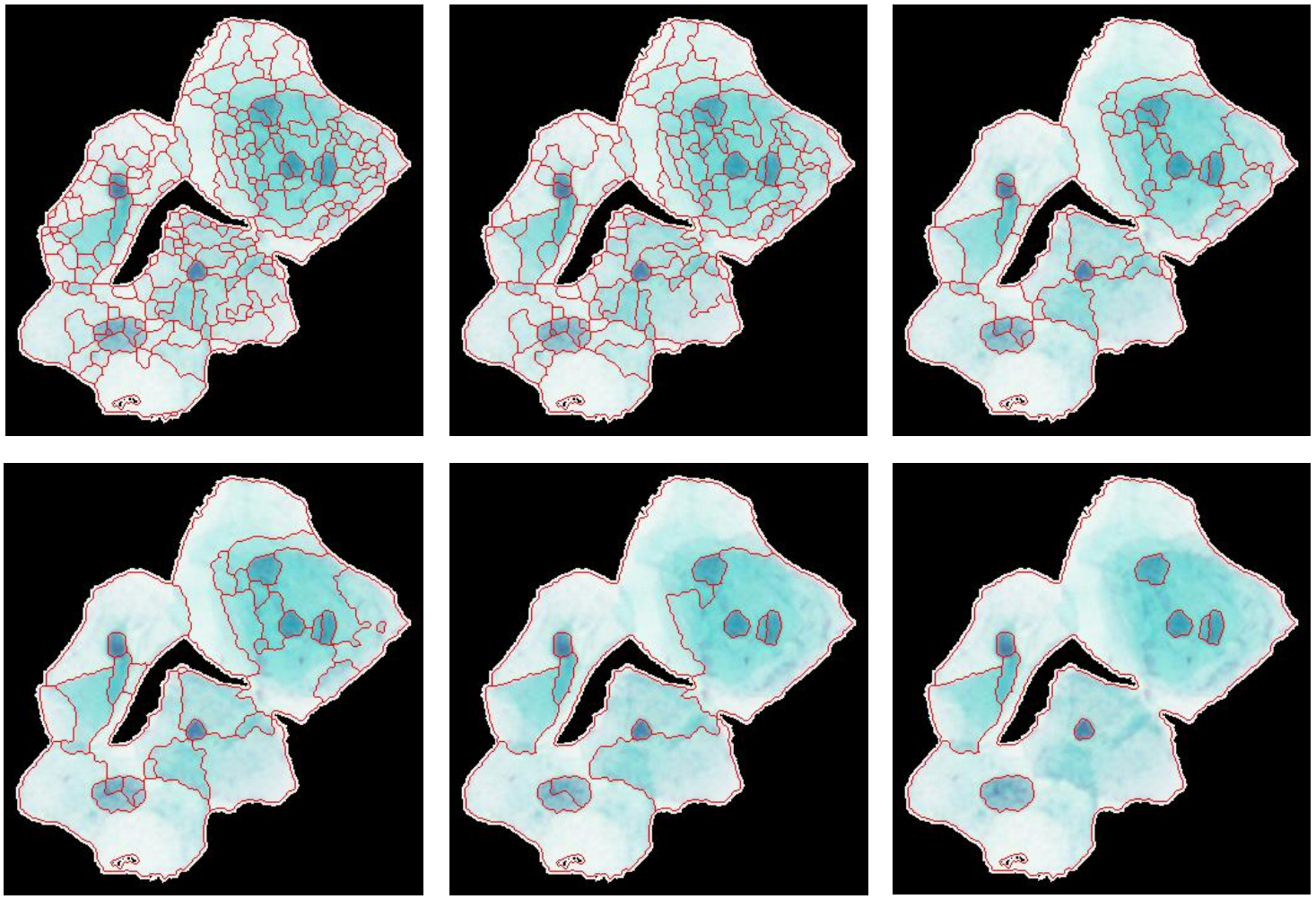
A cell image.



Gradient of the cell image.

Slide credit: S. Aksoy

Watershed segmentation



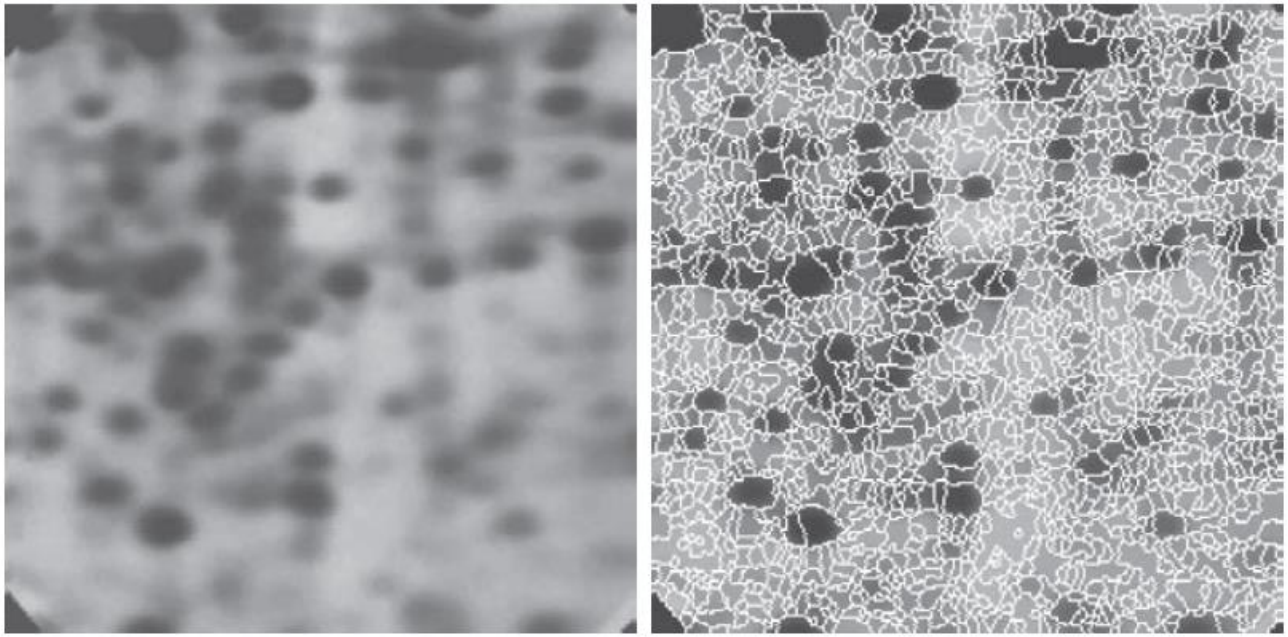
Multi-scale watershed segmentation of the cell image.

Slide credit: S. Aksoy

Over-segmentation

a b

FIGURE 10.60
(a) Electrophoresis image.
(b) Result of applying the watershed segmentation algorithm to the gradient image.
Over-segmentation is evident.
(Courtesy of Dr. S. Beucher, CMM/ Ecole des Mines de Paris.)



The use of markers

a b

FIGURE 10.61
(a) Image showing internal markers (light gray regions) and external markers (watershed lines).
(b) Result of segmentation. Note the improvement over Fig. 10.60(b). (Courtesy of Dr. S. Beucher, CMM/ Ecole des Mines de Paris.)