# Digital Image Processing ECE 566

Ahmad Ghasemi

Department of Electrical and Computer Engineering

University of Massachusetts Amherst

# Superpixel Segmentation

# What is a Superpixel?

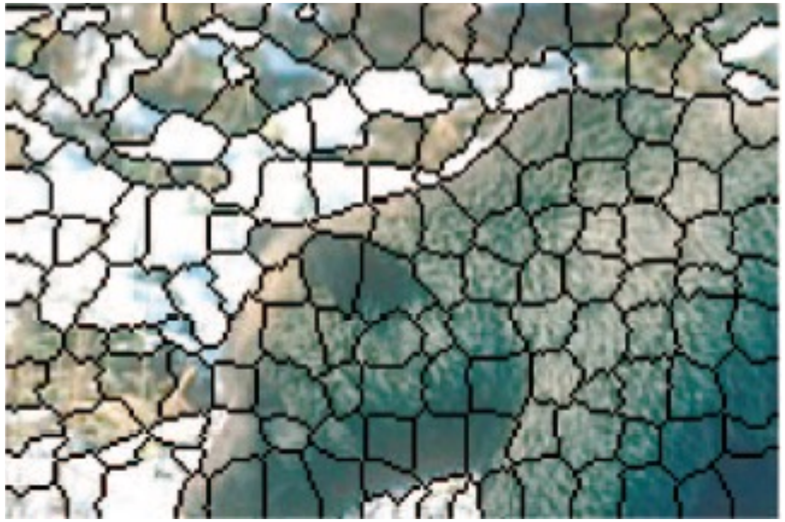
- A superpixel can be defined as a group of connected pixels with similar features; Ex: color, brightness, texture, etc.
- It can be regarded as a result of over segmentation.
- The concept was proposed in 2003 but the results of some former methods also can be called superpixels. Ex: watershed

Watershed [1]

[1] Machairas, Vaïa, Etienne Decencière, and Thomas Walter. "Spatial Repulsion Between Markers Improves Watershed Performance." *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer International Publishing, 2015. 194-202.

# Desirable Properties of Superpixels

- Good adherence to object boundaries
- Regular shape and similar size
- Compute fast and simple to use

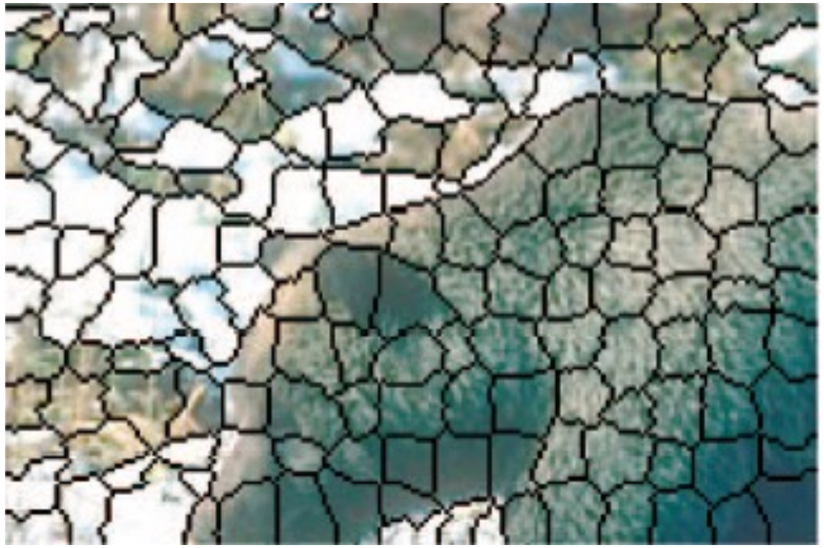


Turbopixel [2]

[2] Levinshtein, Alex, et al. "Turbopixels: Fast superpixels using geometric flows."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.12 (2009): 2290-2297.

# Advantage of Superpixels

- Reduce the computational complexity from thousands of pixels to a few hundred superpixels (High computational efficiency)

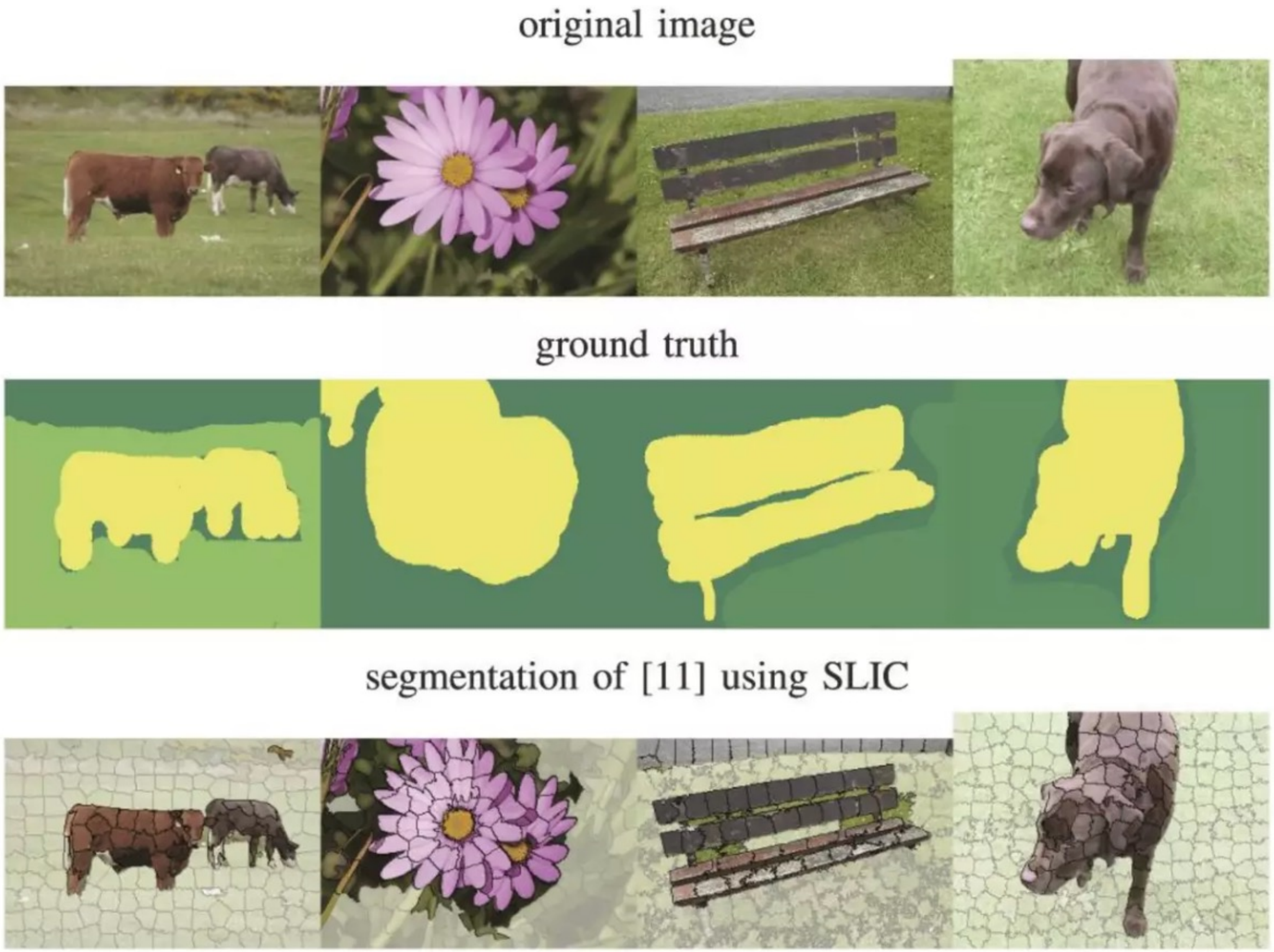


Turbopixel [2]

[2] Levinshtein, Alex, et al. "Turbopixels: Fast superpixels using geometric flows."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.12 (2009): 2290-2297.

# Application of Superpixels

- Segmentation



# Superpixel Segmentation

- Gradient-based superpixel methods
  - SLIC
  - Turbopixel

- Graph-based superpixel methods
  - Ncut
  - ERS

# Gradient-Based Segmentation

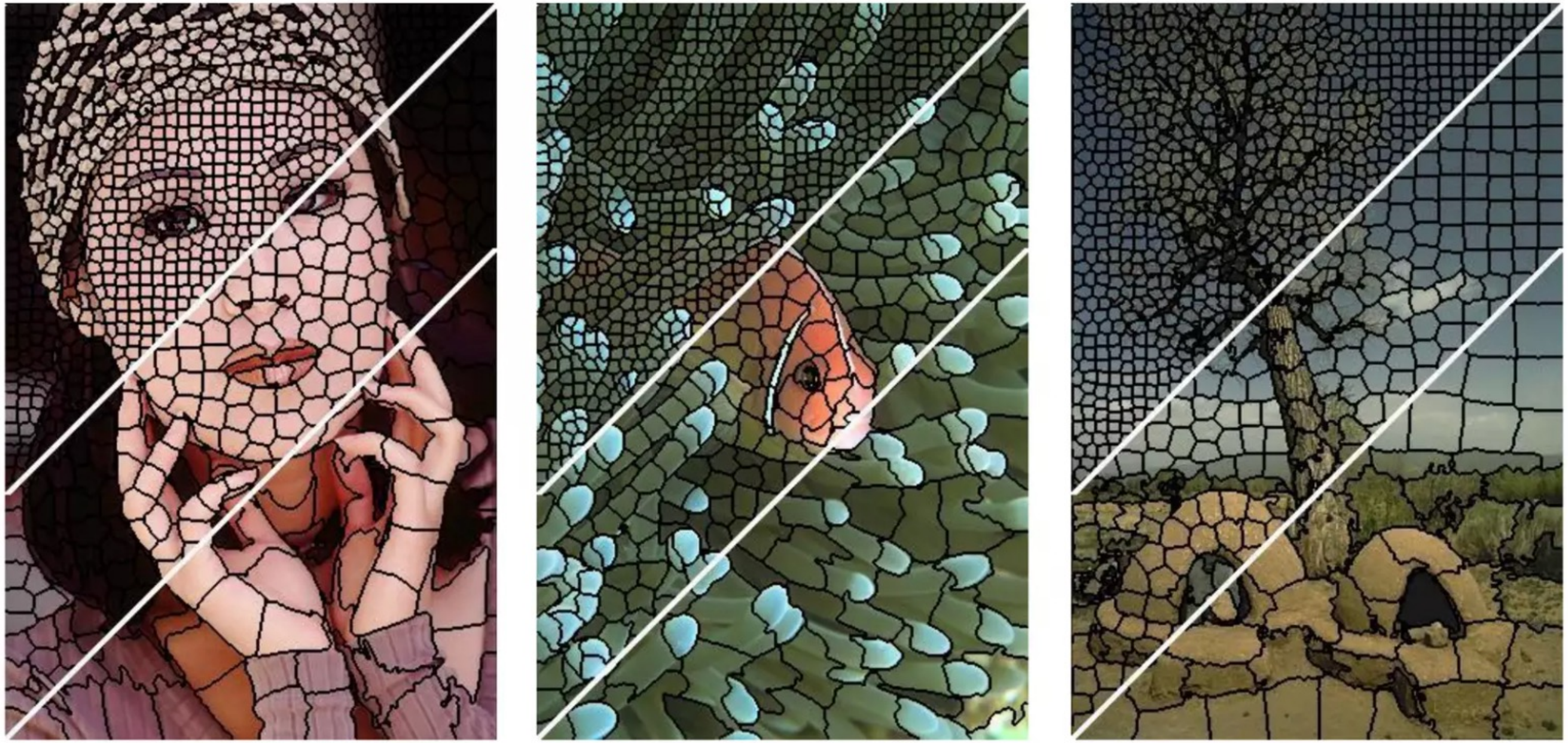
- Starting from rough initial clusters and then iteratively refine the clusters by gradient until some convergence criterion is met.

- Simple linear iterative clustering (SLIC) [3]

[3] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012): 2274-2282.

# Simple linear iterative clustering (SLIC)

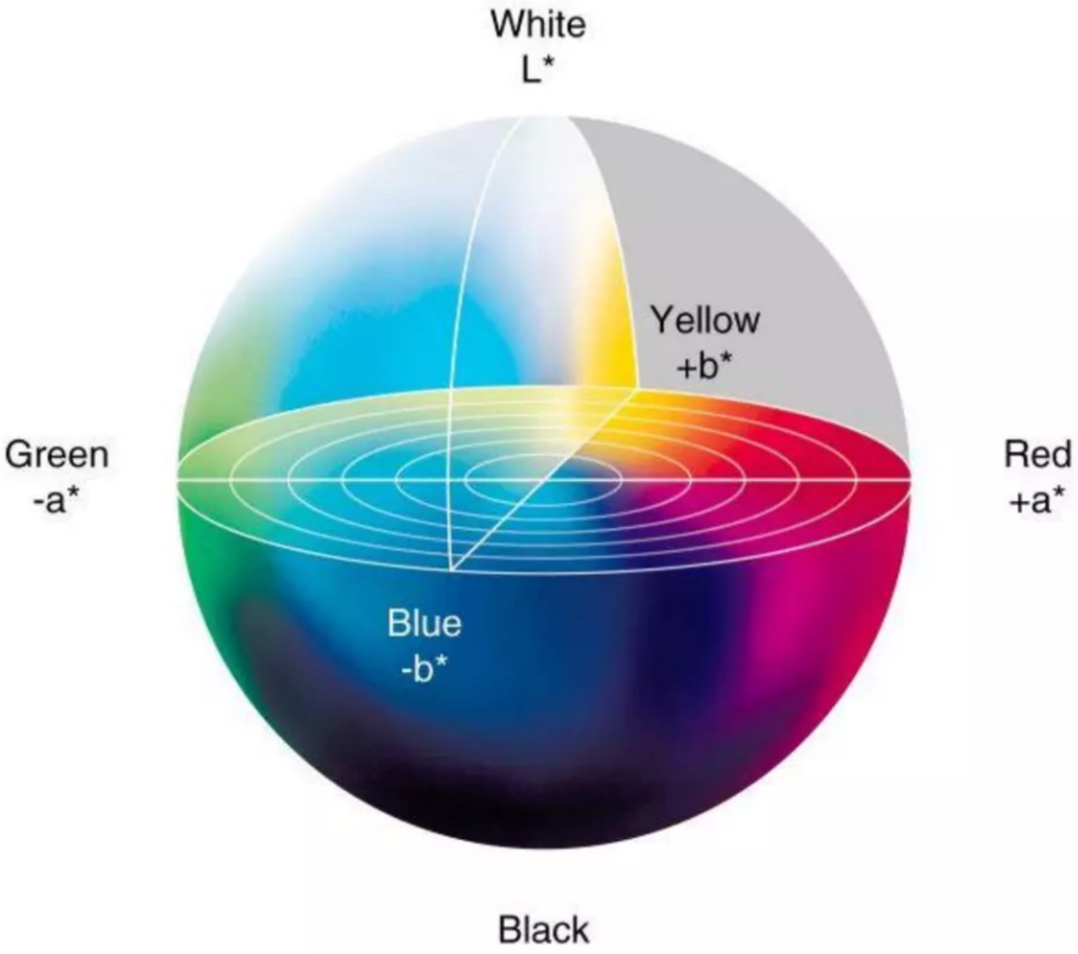
**Sample segmentation output**



[3] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012): 2274-2282.

# Simple linear iterative clustering (SLIC)

- For color images, SLIC works in the LAB color space.




[3] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012): 2274-2282.

# Simple linear iterative clustering (SLIC)

- Converting RGB to the LAB color space.

$$L = 116.\, h\left(\frac{Y}{Y_w}\right) - 16$$

$$A = 500\left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right)\right]$$

$$B = 200\left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right)\right]$$

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + \dfrac{16}{116} & q \leq 0.008856 \end{cases}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X_w = 0.950450 \qquad Y_w = 1.000000 \qquad Z_w = 0.088754$$

[3] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012): 2274-2282.

# Converting RGB to the LAB color: Example

# Simple linear iterative clustering (SLIC): Example

- Image size : 300 × 300
- Number of pixels: 90000
- Number of superpixels: 45
- Each pixel is assigned to a 5-valued vector (L, A, B, x, y) for color and position.

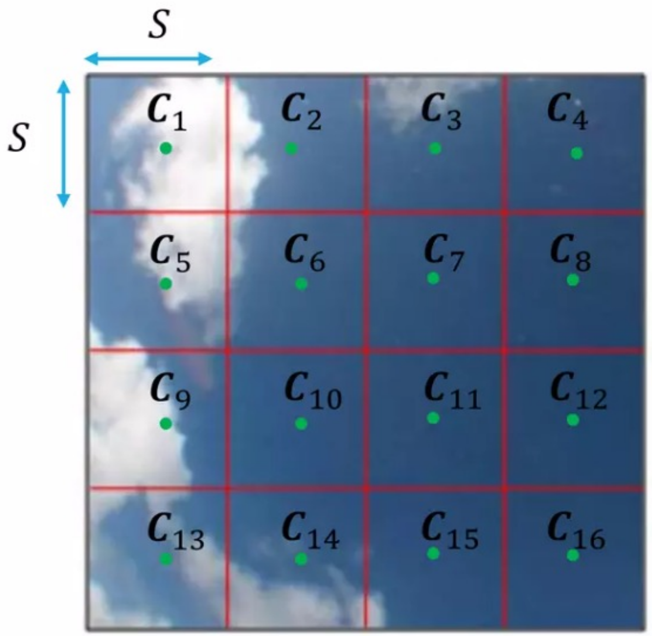
# Steps of SLIC

- Image size : 40 $\times$ 40
- Number of pixels: 1600
- Number of superpixels: 16

# Steps of SLIC

1. Initialize $K$ cluster centers $\boldsymbol{C_k} = [\boldsymbol{l_k}, \boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{x_k}, \boldsymbol{y_k}]^{\boldsymbol{T}}$ on a regular grid spaced $S = \sqrt{\frac{N}{K}}$ pixels apart.
   - each superpixel has approximately $\frac{N}{K}$ pixels

For each pixel location, $p$, set a label $L(p) = -1$ and a distance $d(p) = \infty$



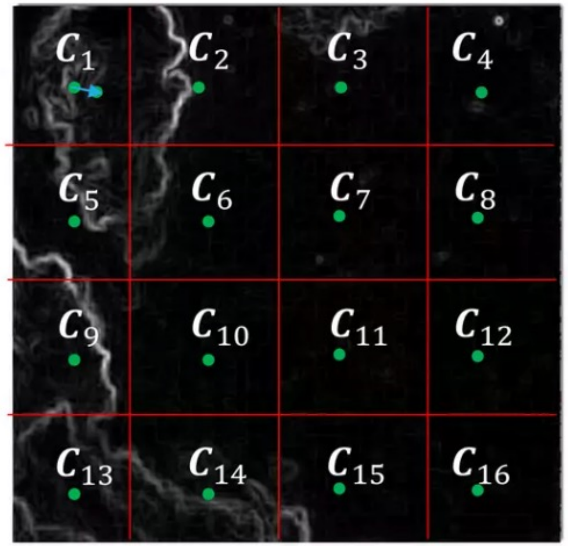

$$\frac{N}{K} = \frac{1600}{16} = 100$$

$$S = \sqrt{\frac{N}{K}} = \sqrt{100} = 10$$

# Steps of SLIC

2. move these cluster centers to the positions with the lowest gradients in a 3 × 3 neighborhood;

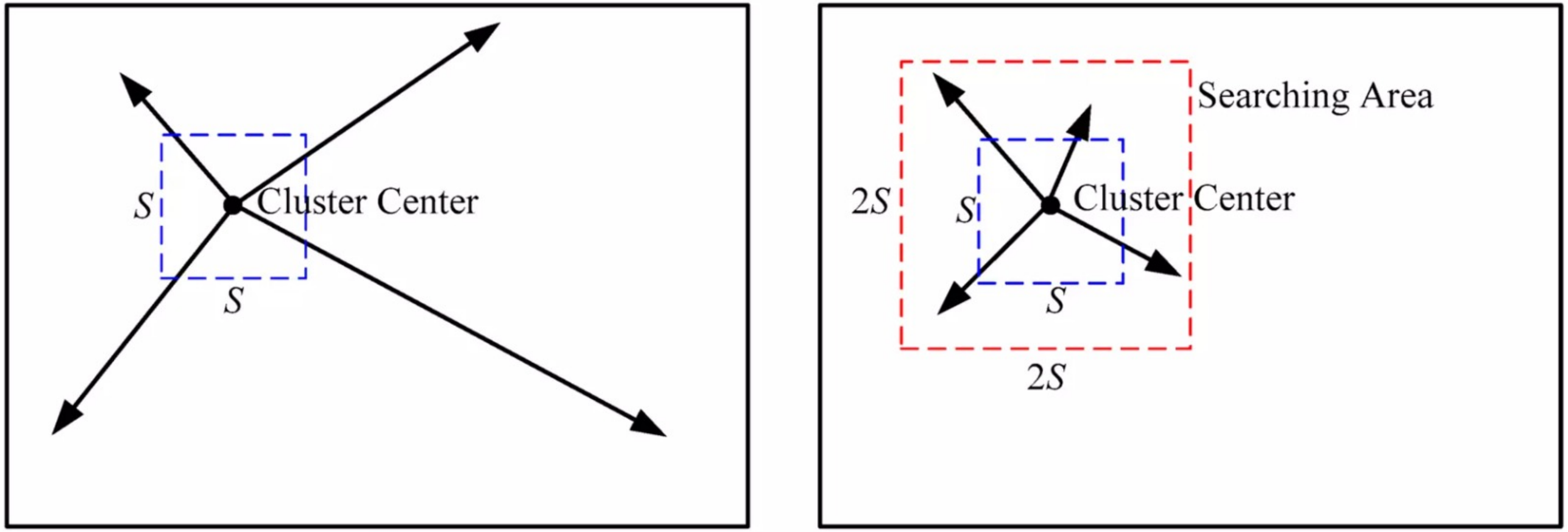
This is done to avoid centering a superpixel on an edge, and to reduce to chance of seeding superpixel with a noisy pixel.




$$G(x,y) = \|I(x,y-1) - I(x,y+1)\|^2 + \|I(x-1,y) - I(x+1,y)\|^2$$

# Steps of SLIC

3. Assign pixels. Designate each pixel to a closest cluster center in a local search space;




(a) K-means algorithm

(b) SLIC algorithm

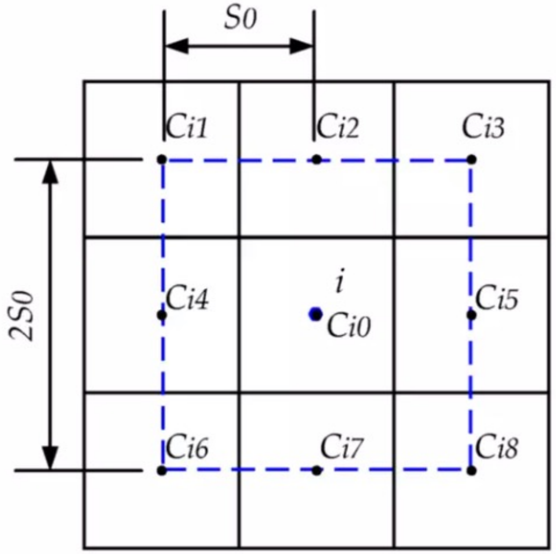
# Steps of SLIC

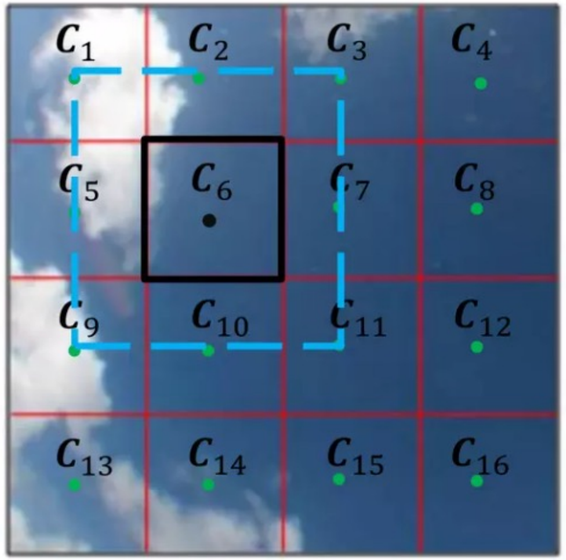
3. Assign pixels. Designate each pixel to a closest cluster center in a local search space;

for each cluster center $C_k$

Assign the best matching pixels from a $\mathbf{2S} \times \mathbf{2S}$ around the cluster center according to the distance measure.

end for




: Superpixel　: $2S0 \times 2S0$ search region　• : Superpixel center　• : Image pixel

# Steps of SLIC

3. Assign pixels. Designate each pixel to a closest cluster center in a local search space;

## Distance measure

Color distance: $$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

position distance: $$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$\boxed{D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}}$$

m= Compactness of superpixel $\quad S = \sqrt{N/K}$

University of Massachusetts Amherst

# Steps of SLIC

3. Assign pixels. Designate each pixel to a closest cluster center in a local search space;

## Distance measure

Color distance: $d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$

position distance: $d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$

$$\boxed{D = \sqrt{d_c^{\,2} + \left(\frac{d_s}{S}\right)^2 m^2}}$$

$$\boxed{D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}}$$

m= Compactness of superpixel $\quad S = \sqrt{N / K}$

# Steps of SLIC

3. Assign pixels. Designate each pixel to a closest cluster center in a local search space;

## Distance measure

Color distance: $d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$

position distance: $d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$

$$\boxed{D = \sqrt{d_c^{\,2} + \left(\frac{d_s}{S}\right)^2 m^2}}$$

$$\boxed{D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}}$$

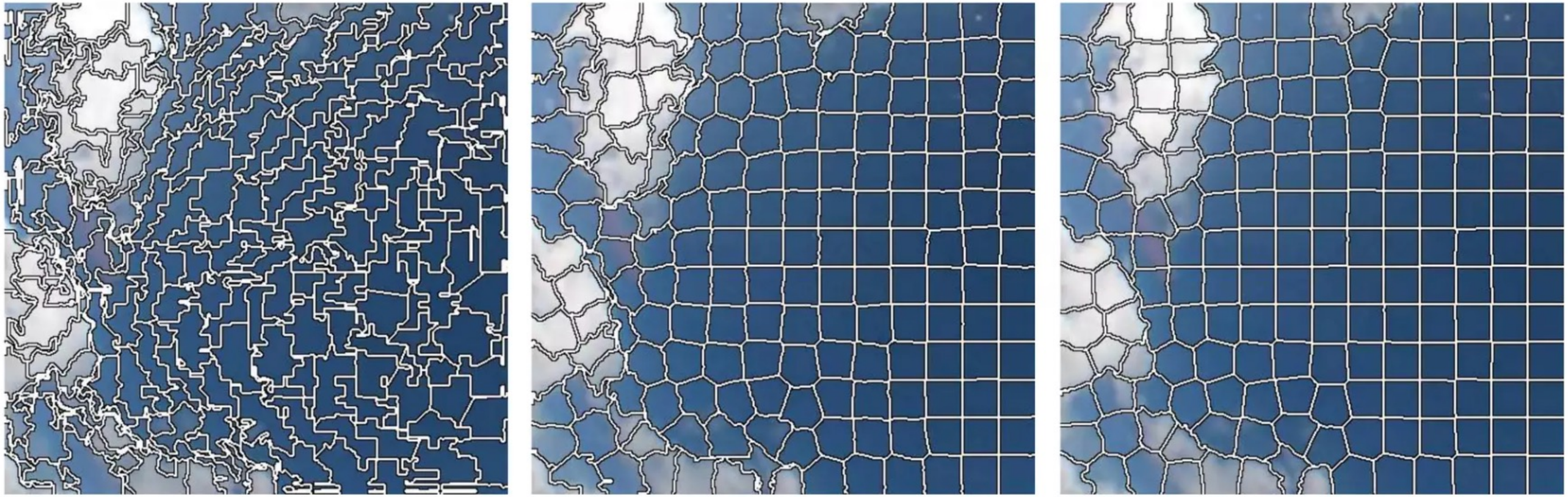
m= Compactness of superpixel $\quad S = \sqrt{N / K}$

if $D_i < d(p)$

➔ $d(p) = D_i$ and $L(p) = i$

# Steps of SLIC

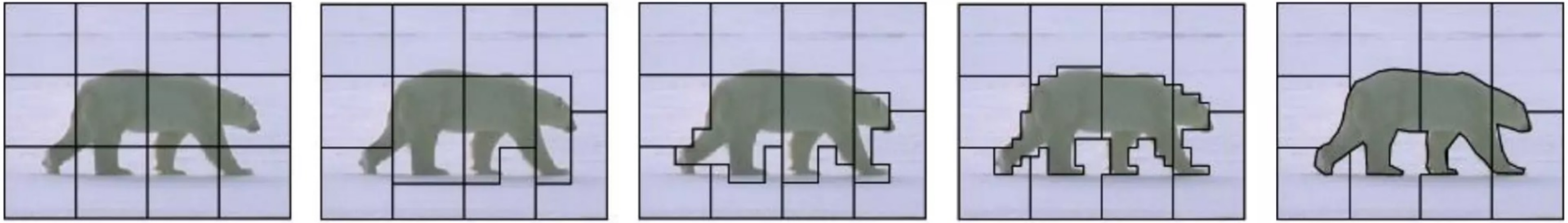
Compactness of Superpixels (m)



m = 1 m = 20 m = 40

# Steps of SLIC

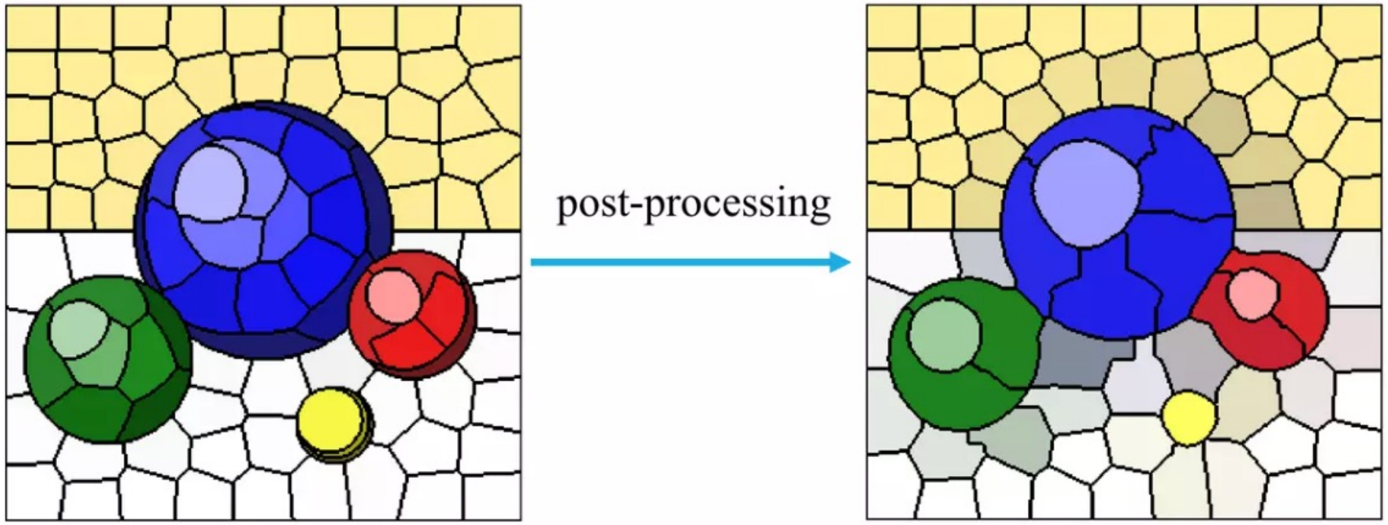
3. Assign pixels. Designate each pixel to a closest cluster center in a local search space
4. Update cluster centers. Set each cluster center as the mean of all pixels in the corresponding cluster;
5. Repeat steps (3)–(4) until the clusters do not change or error (difference between previous cluster and new cluster) is converge.



initialization iteration 1 iteration 2 iteration 3 iteration 4

# Steps of SLIC

6. Post-processing. The connected components algorithm is used to reassign isolated regions to nearby superpixels if the size of the isolated regions is smaller than a minimum size $S_{min}$.



# K-Means vs SLIC

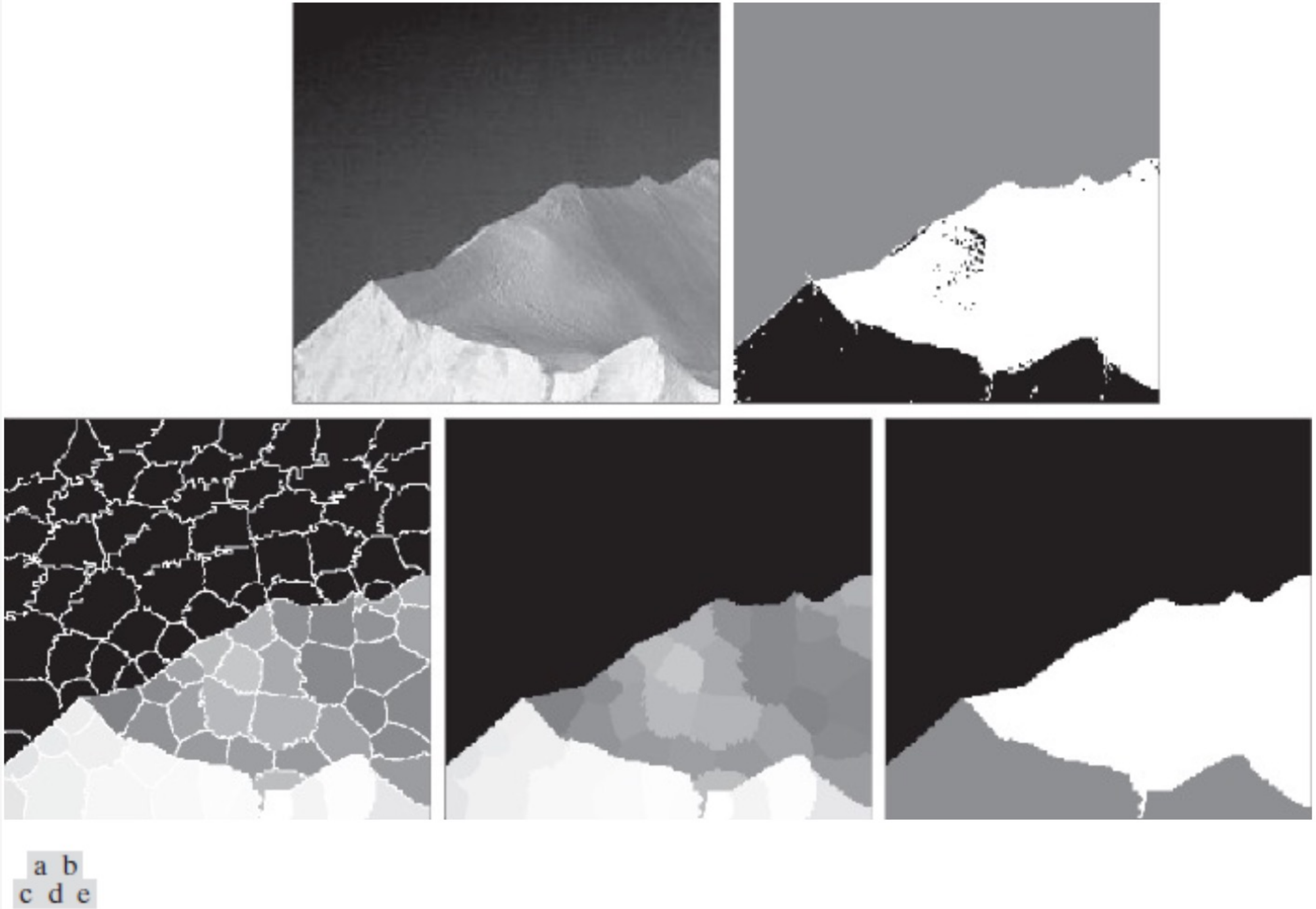



FIGURE 10.53 (a) Image of size $533 \times 566$ (301,678) pixels. (b) Image segmented using the $k$-means algorithm. (c) 100-element superpixel image showing boundaries for reference. (d) Same image without boundaries. (e) Superpixel image (d) segmented using the $k$-means algorithm. (Original image courtesy of NOAA.)

# Comparison

[GS04] Graph-based segmentation

[NC05] Normalized cuts

[TP09] Turbopixels

[QS09] QuickShift

SLIC

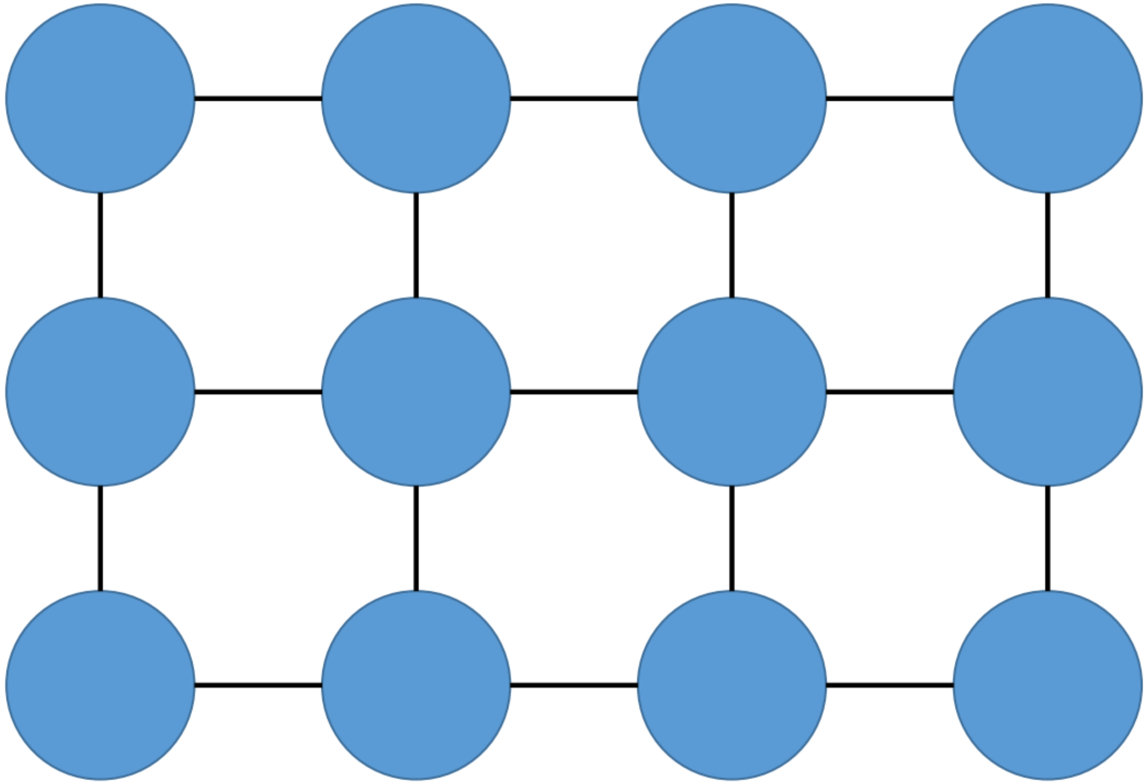
University of Massachusetts Amherst

# Graph-Based Segmentation

- Images as graphs
- Set of points of the feature space represented as a weighted, undirected graph, G = {Nodes, Edges} = {V, E}.
- The points of the feature space are the nodes of the graph.
- Edge between every pair of nodes.

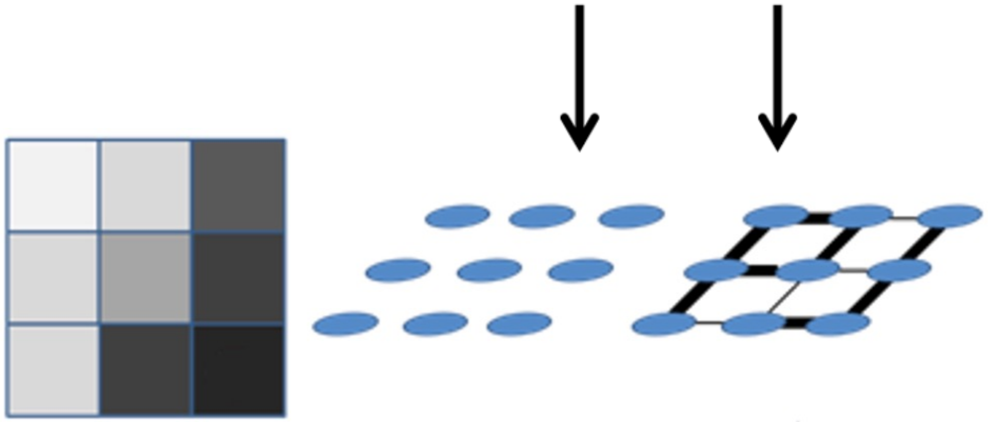
# Graph-Based Segmentation

Images can be viewed as graphs



Nodes: pixels

Edges: Constraints between neighboring pixels



# Graph-Based Segmentation

- Weight on each edge, $w(i,j)$ or $w_{ij}$, is a function of the similarity between the nodes $i$ and $j$.

- Partition the set of vertices into disjoint sets where similarity within the sets is high and across the sets is low.

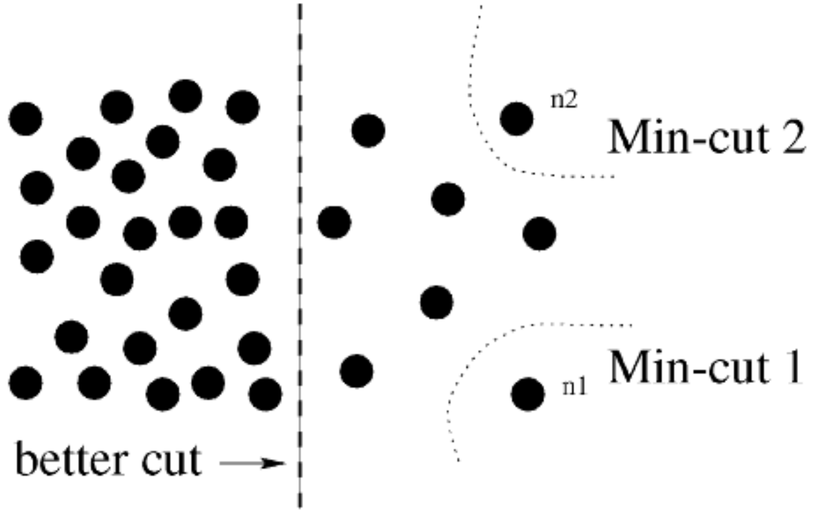
- Therefore, Segmentation is equivalent to Graph partition.

# Graph-Based Segmentation

- Cut: a set of edges whose removal disconnects the graph.
- The cost of the cut is the sum of the weights on cut edges.

$$A \cup B = V,\ \ A \cap B = \varnothing,\ cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

- **Min Cut** is a method of minimizing the cost of the cut, but it favors cutting small sets of isolated nodes in the graph.



# Segmentation with Min Cut

- By Min Cut method, the graph is partitioned into clusters.
- Each cluster is considered as an image segment.
- Min Cut method uses the HCS (Highly Connected Subgraphs) Algorithm to find the clusters.

# Segmentation with Min Cut

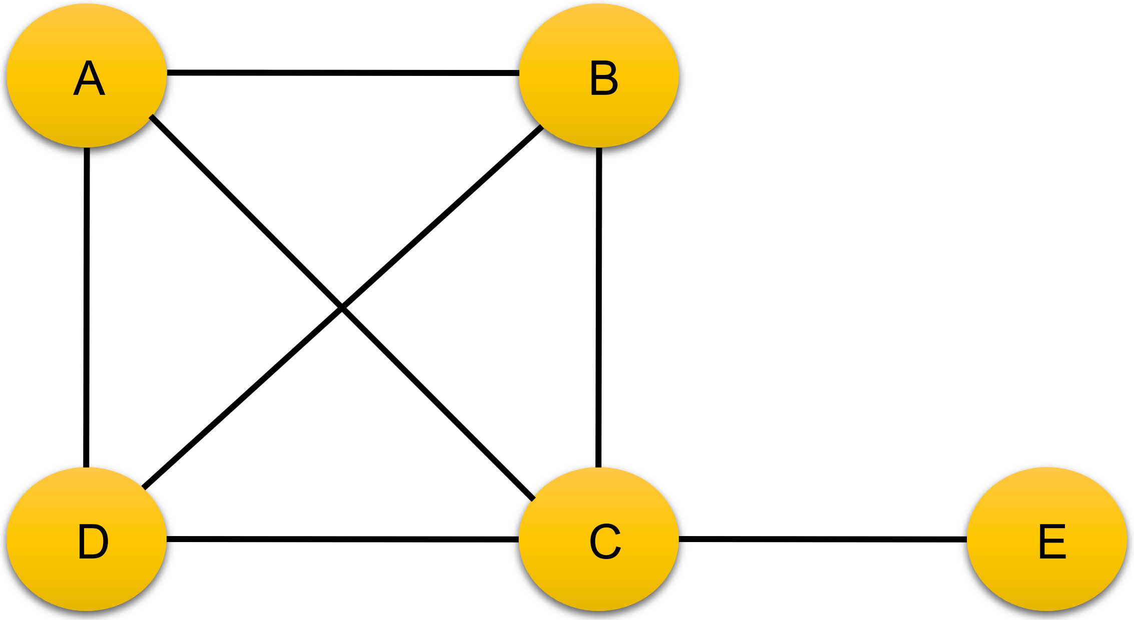
- Edge Connectivity $k(G)$ is the minimum number of edges whose removal results in a disconnected graph.
- For a graph with vertices $n > 1$ to be highly connected if its edge-connectivity $k(G) \ > \frac{n}{2}$.
- A highly connected subgraph (HCS) is an induced subgraph $H$ in $G$ such that $H$ is highly connected.

# Segmentation with Min Cut: Example

No. of nodes = 5

Edge Connectivity; $k(G)$ = 1

Not HCS!

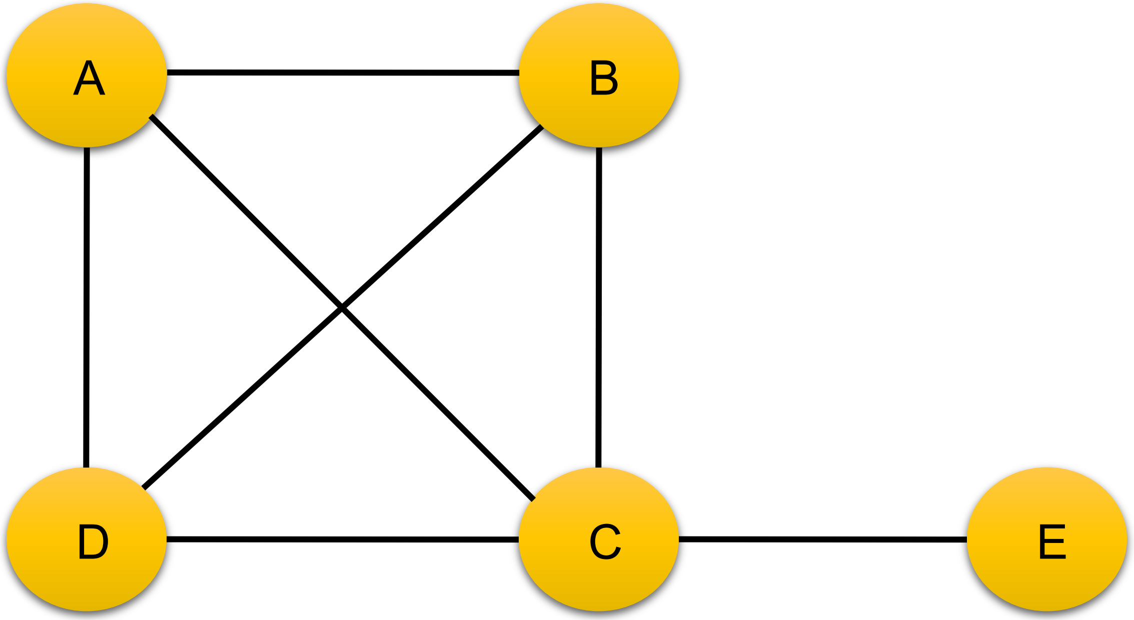


# Segmentation with Min Cut: Example

No. of nodes = 5

Edge Connectivity; $k(G)$ = 1

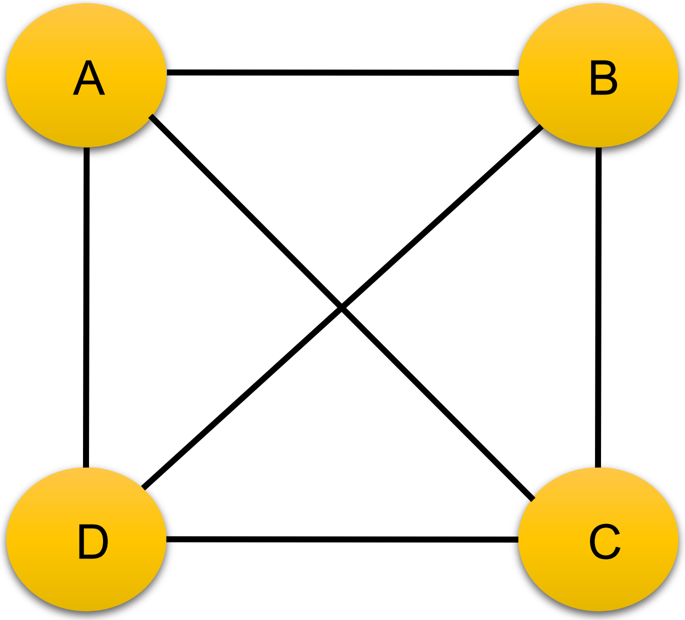
Not HCS!




No. of nodes = 4

Edge Connectivity; $k(G)$ = 3

HCS!



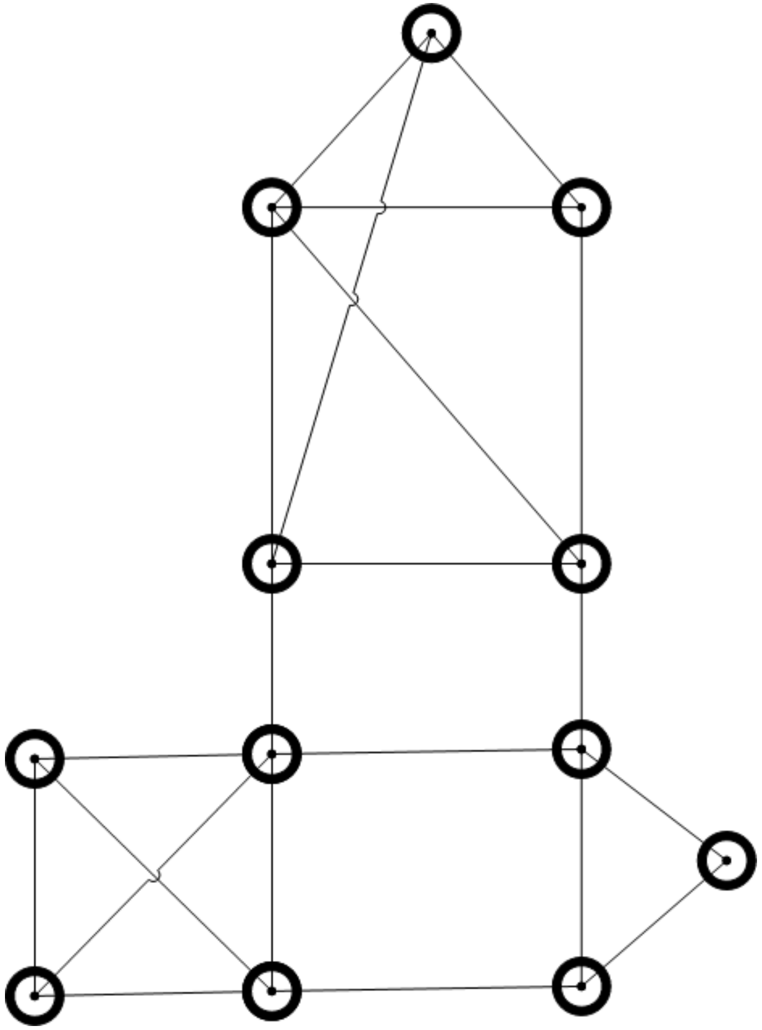
# HCS algorithm - Min Cut Method

```
HCS(G(V,E))
begin
  (H, H',C) ← MINCUT(G)
  if G is highly connected
      then return (G)
  else
      HCS(H)
      HCS(H')
  end if
end
```

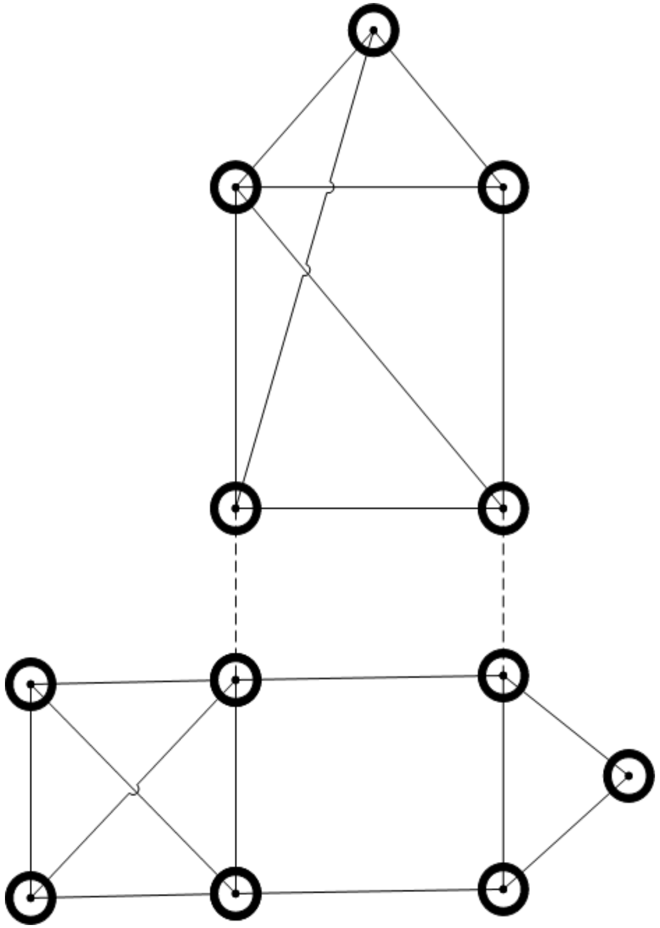
# HCS algorithm - Min Cut Method

- The procedure MINCUT(G) returns H, H’ and C where C is the minimum cut which separates G into the subgraphs H and H’.
- Procedure HCS returns a graph in case it identifies it as a cluster.
- Single vertices are not considered clusters and are grouped into singletons set S.

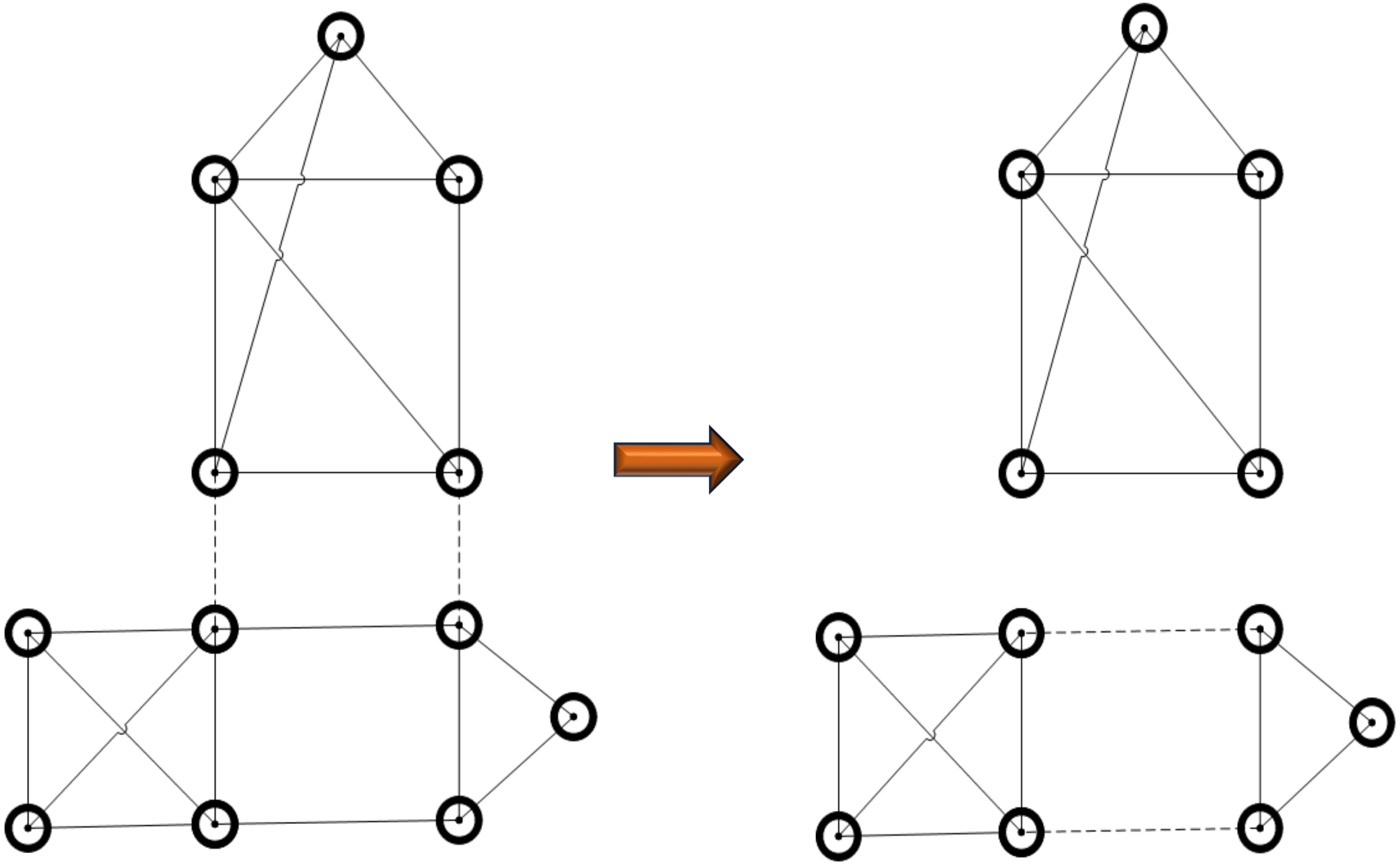
# HCS algorithm - Min Cut Method: Example



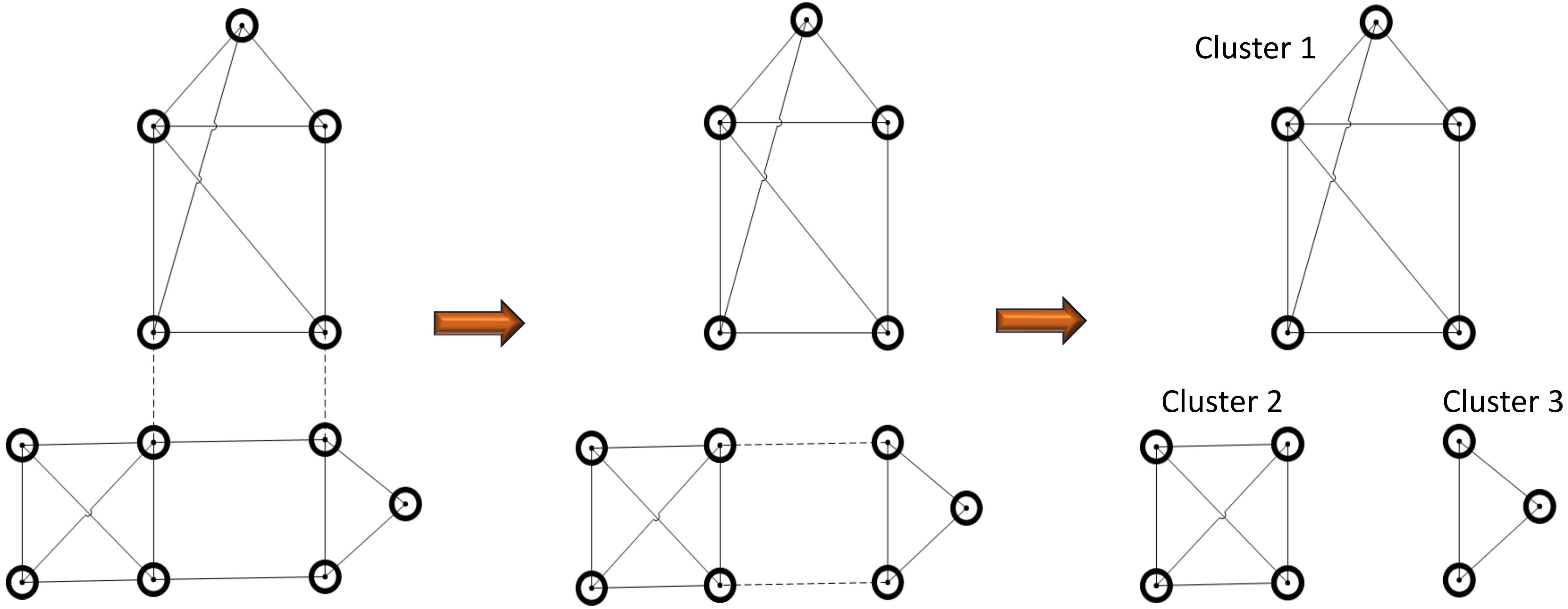
# HCS algorithm - Min Cut Method: Example



# HCS algorithm - Min Cut Method: Example



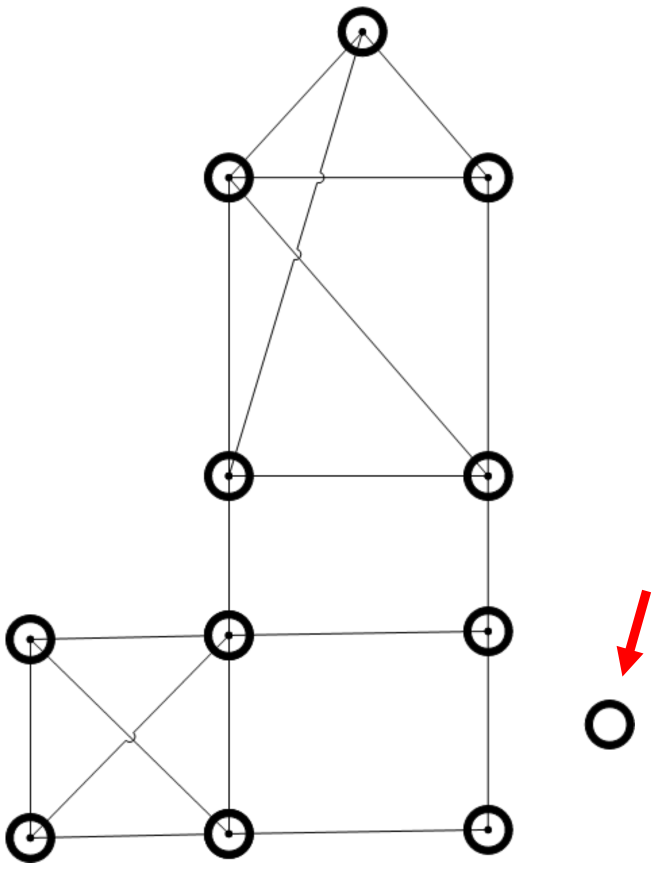
# HCS algorithm - Min Cut Method: Example



# HCS algorithm - Min Cut Method

- The problem with HCS Algorithm – Min Cut method is that there is a chance that many singletons may be separated.

- To overcome this issue, Normalized-cut (Ncut) for image segmentation has been proposed.



# Segmentation as a graph cut

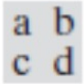
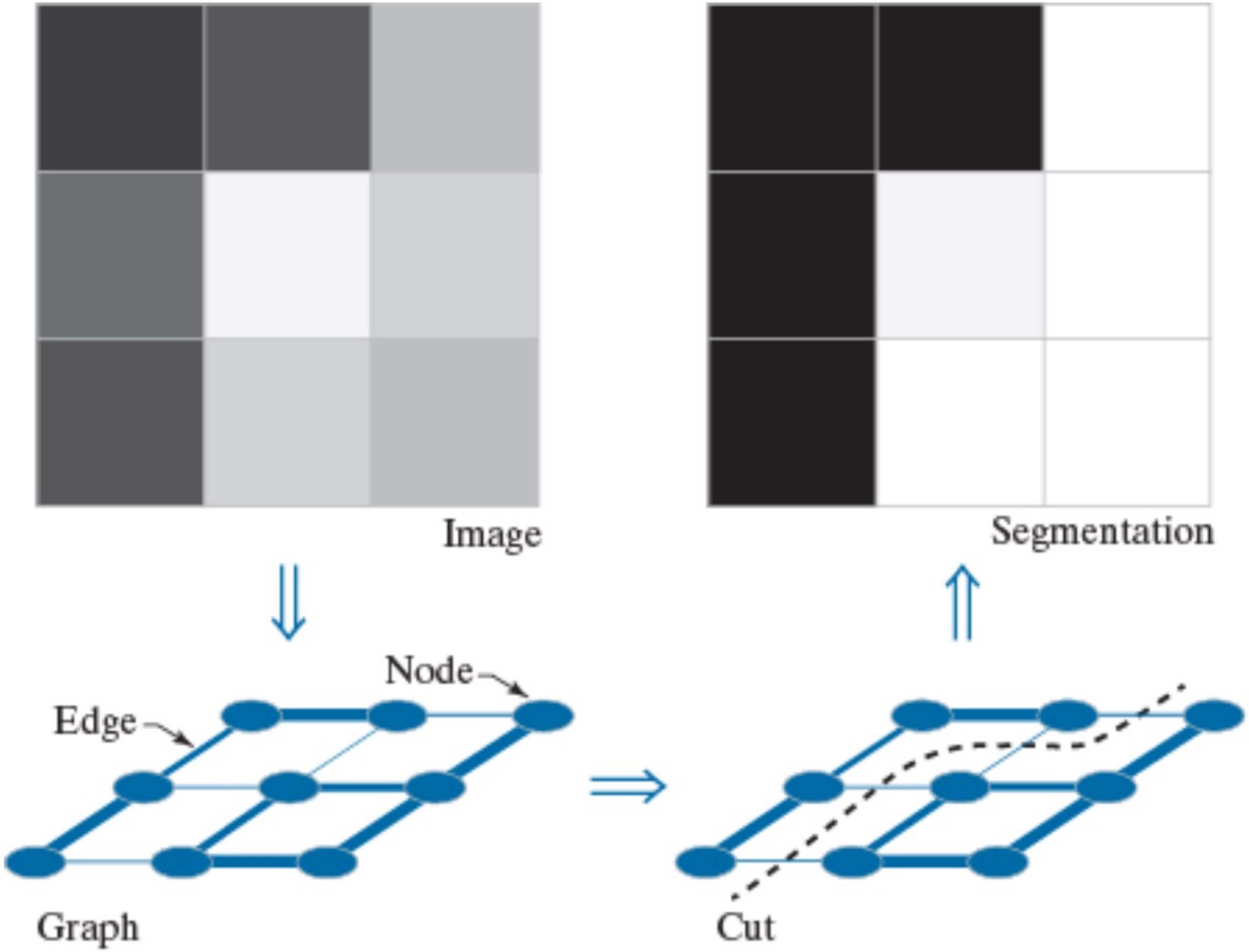



FIGURE 10.55
(a) A 3×3 image.
(c) A corresponding graph.
(d) Graph cut.
(c) Segmented image.

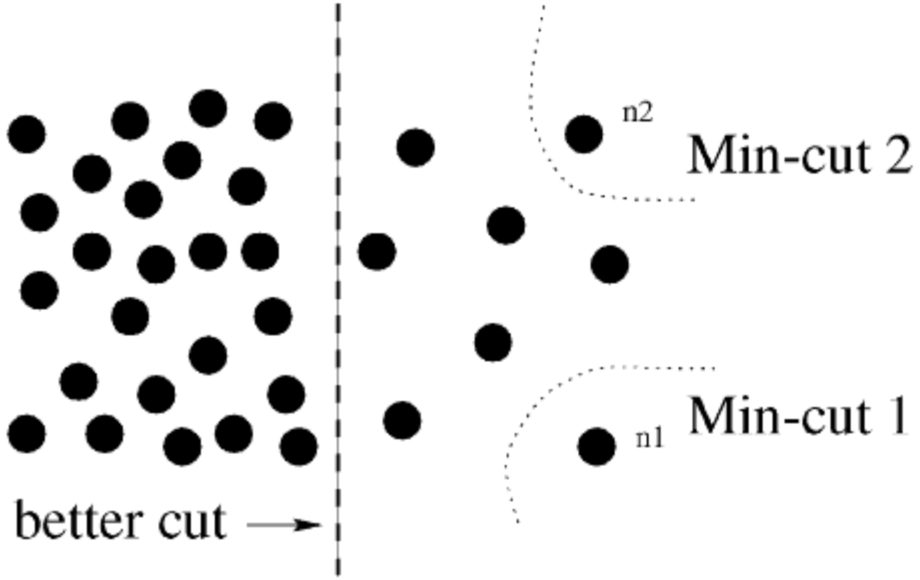


# Normalized cut (Ncut)

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

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

Total connection from nodes in A to all nodes in the graph




- Minimum cut criteria favors cutting small sets of isolated nodes in the graph.
- Minimize NCut to segment images.
- The cut is made such that the normalized cut value between the two sets A & B is minimum.

[5] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."

# Ncut: Example

- Regular and compact shape
- High computational cost especially for images with large size

The average superpixel size in the upper left of each image is 100 pixels and is 300 in the lower right.

[5] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."