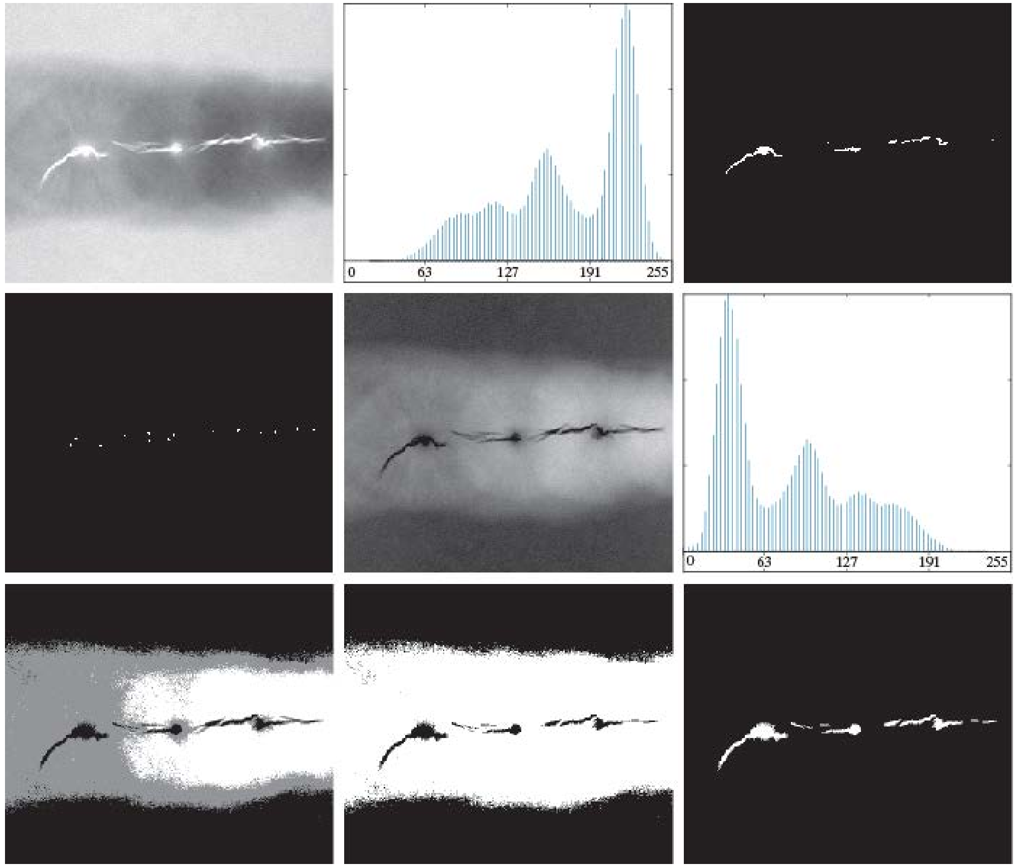
Lecture 16 – Region-based Segmentation (区域分割)

This lecture will cover:

- Region Growing (区域生长)
- Region Splitting and Merging (区域分裂与聚合)
- Region Segmentation using Clustering (聚类)
- Region Segmentation using Superpixels (超像素)
- Region Segmentation using Graph Cuts (图分割)
- Color Image Segmentation

上海科技大学
ShanghaiTech University

Region Growing (区域生长)



a b c
d e f
g h i

Figure 10.46 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between the seed value (255) and (a). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

- **Grouping pixels or subregions into larger regions based on predefined criteria for growth.**
- **Algorithm based on 8-connectivity**

Where $f(x, y)$: input image

$S(x, y)$: a seed array

Q: a predicate to be applied at each location

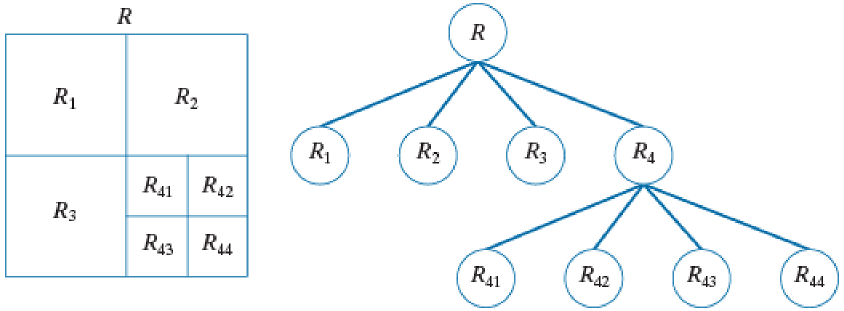
1. Find all connected components in $S(x, y)$ and erode each component to one pixel;
2. Form an image f_Q based on if satisfying Q
3. In f_Q, find all the 1-valued points which 8-connected to each seed point in S, and form an image g;
4. Label each connected component in g, and this is the segmented image obtained by region growing.

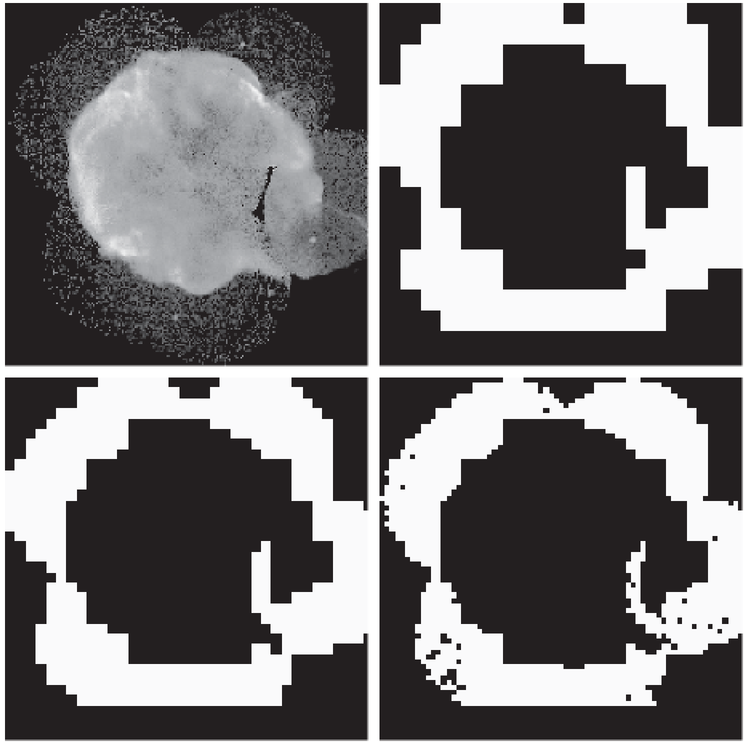
Region Splitting and Merging(区域分裂与聚合)

- **To subdivide an image initially into a set of disjoint regions and then merge and/or split the regions in an attempt to satisfy the condition of segmentation;**
- **Steps**
  1. Split into four disjoint quadrants any region R_i for which $Q(R_i) = False$ (Note: need to specify a minimum quadregion size beyond which no further splitting is carried out);
  2. Merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k) = True$;
  3. Stop when no further merging is possible.

a b

FIGURE 10.47 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.





a b
c d

FIGURE 10.48 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b) through (d) Results of limiting the smallest allowed quadregion to be of sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Segmentation using Clustering (聚类)

- **K-means Clustering:**
  - Partition the set Q into a specified number k of clusters, and each observation is assigned to the cluster with the nearest mean (smallest value).
  - **Algorithm**

Let $z = \{z_1, z_2, \cdots, z_Q\}$: set of vector observations (samples)
$C = \{C_1, C_2, \cdots, C_k\}$: disjoint cluster sets
m_i : the mean vector (centroid) of the samples in set C_i

The following criterion of optimality is satisfied

$$\underset{C}{\operatorname{argmin}} \left(\sum_{i=1}^{k} \sum_{z \in C_i} \|z - m_i\|^2 \right)$$

1. Initialize the algorithm: specify an initial set of mean $m_i(1)$
2. Assign samples to cluster:

$$z_q \rightarrow C_i \text{ if } \|z_q - m_i\|^2 < \|z_q - m_j\|^2 \quad j = 1,2,\cdots,k (j \neq i)$$

3. Update the cluster centers (means) : $m_i = \frac{1}{|C_i|} \sum_{z \in C_i} z$
4. Test for completion
   - ✓ Compute the Euclidean norms of the difference between mean vectors;
   - ✓ Compute the residual error E as the sum of the k norms, stop if $E \leq T$

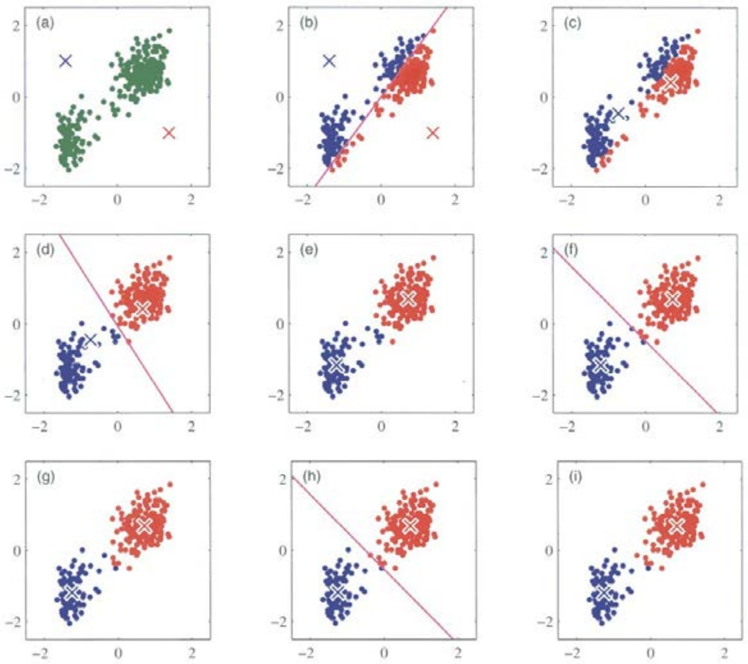


Figure Illustration of K-means algorithm from [Bishop 2006]: (a) initials (represented with crosses) are randomly selected from data set; (b), whole data is assigned into clusters depending on which cluster center is nearest; (c) each cluster center is re-computed to be the mean of the points assigned to the corresponding cluster; (d)-(h) shows iterative improvement of cluster centers; (i) cluster means are stable and are no more changing. These are chosen as representative, and clusters are selected accordingly.

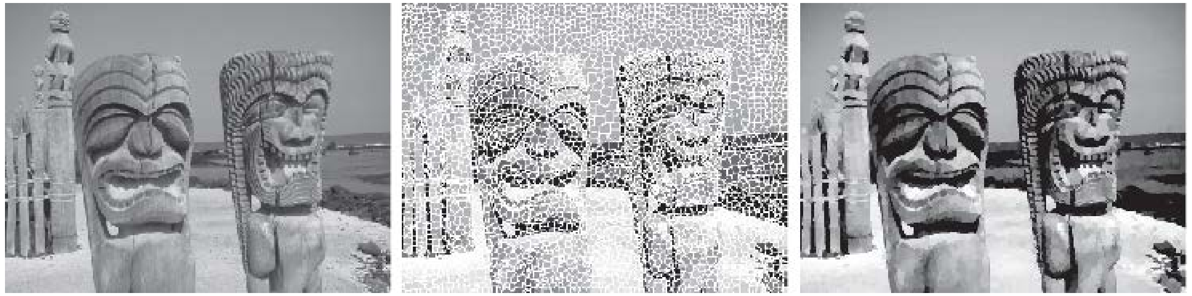
a b

FIGURE 10.49 (a) Image of size 688 × 688 pixels. (b) Image segmented using the *k*-means algorithm with *k* = 3.

Segmentation using Superpixels (超像素)

➢ **Superpixels (超像素):**

- To replace the standard pixel grid by grouping pixels into primitive regions (more perceptually meaningful).
- To lessen computational load and to improve the performance of segmentation algorithms by reducing irrelevant details.
- Requirement of any superpixel representation is adherence to boundaries.



a b c

FIGURE 10.50 (a) Image of size 600 × 480 (480,000) pixels. (b) Image composed of 4,000 superpixels (the boundaries between superpixels (in white) are superimposed on the superpixel image for reference—the boundaries are not part of the data). (c) Superpixel image. (Original image courtesy of the U.S. National Park Services.).

a b c

FIGURE 10.51 (a) Original image. (b) Image composed of 40,000 superpixels. (c) Difference between (a) and (b).

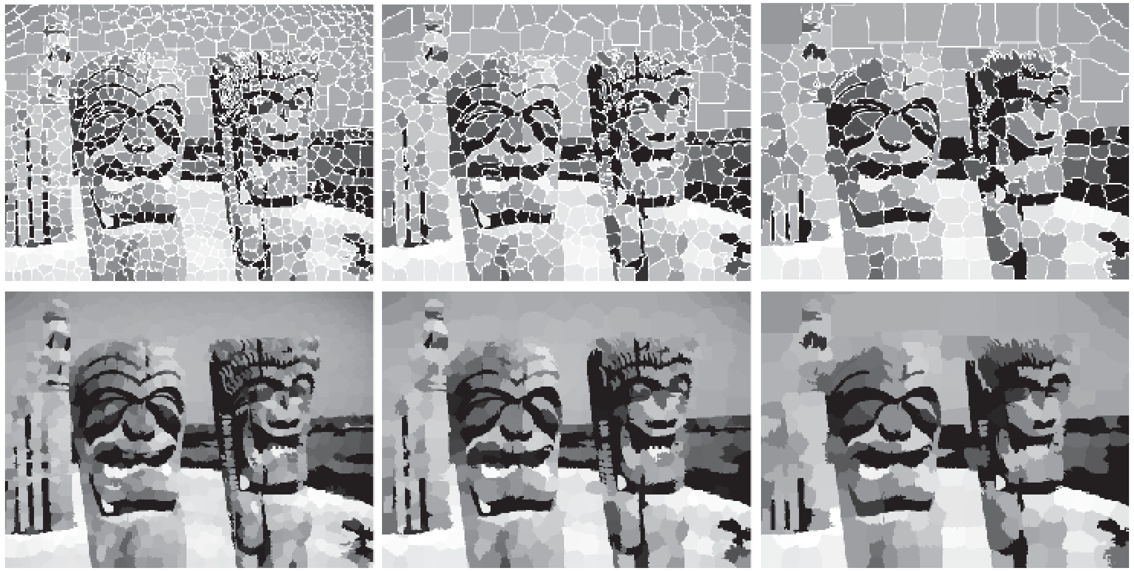


FIGURE 10.52 Top row: Results of using 1,000, 500, and 250 superpixels in the representation of Fig. 10.50(a). As before, the boundaries between superpixels are superimposed on the images for reference. Bottom row: Superpixel images.

上海科技大学 ShanghaiTech University

Segmentation using Superpixels (超像素)

➢ Simple Linear Iterative Clustering (SLIC) superpixel Algorithm

- A modification of K-means algorithm
- Algorithm: $z = \{r, g, b, x, y\}$: 5-dimensional vector associated with pixel, n_{sp}: the desired number of superpixels, n_{tp}: the total number of pixels in the images

1. Initialize the algorithm:
   - ✓ $s = [n_{\text{tp}}/n_{\text{sp}}]^{1/2}$ is the regular grid;
   - ✓ Move the cluster centers to the lowest gradient position of 3*3 neighborhood
   - ✓ For each pixel location p, set a label $L(p) = -1$ and a distance $d(p) = \infty$
2. Assign samples to cluster:
   - ✓ For each cluster center m_i, compute the distance $D_i(p)$ between m_i and each pixel p in a $2s \times 2s$ neighborhood of m_i
   - ✓ For each p and $i = 1,2, \cdots n_{\text{sp}}$, if $D_i < d(p)$, let $d(p) = D_i$ and $L(p) = i$
3. Update the cluster centers (means) :
   - ✓ Let C_i denote the set of pixels in the image with label $L(p) = i$, update $m_i = \frac{1}{|C_i|} \sum_{z \in C_i} z$
4. Test for convergence:
   - ✓ If yes, go to Step 5;
   - ✓ Else, go back to Step 2
5. Post-process the superpixel regions:
   - ✓ replace all the superpixels in each region C_i by their average value m_i

Segmentation using Superpixels (超像素)

➢ **Specifying the Distance Measure**

- For a space whose coordinates are colors and spatial variables;
- By normalizing the distance of the various components, then combining them into a single measure:

Let d_c : the color Euclidean distance between two points in a cluster

$$d_c = \left[(r_j - r_i)^2 + (g_j - g_i)^2 + (b_j - b_i)^2\right]^{1/2}$$

d_s : the spatial Euclidean distance between two points in a cluster

$$d_s = \left[(x_j - x_i)^2 + (y_j - y_i)^2\right]^{1/2}$$

d_{cm} : the maximum expected value of d_c, usually set as a constant $\boldsymbol{c}$

d_{sm} : the maximum expected value of d_s, $d_{sm} = s = \left[n_{\text{tp}}/n_{\text{sp}}\right]^{1/2}$

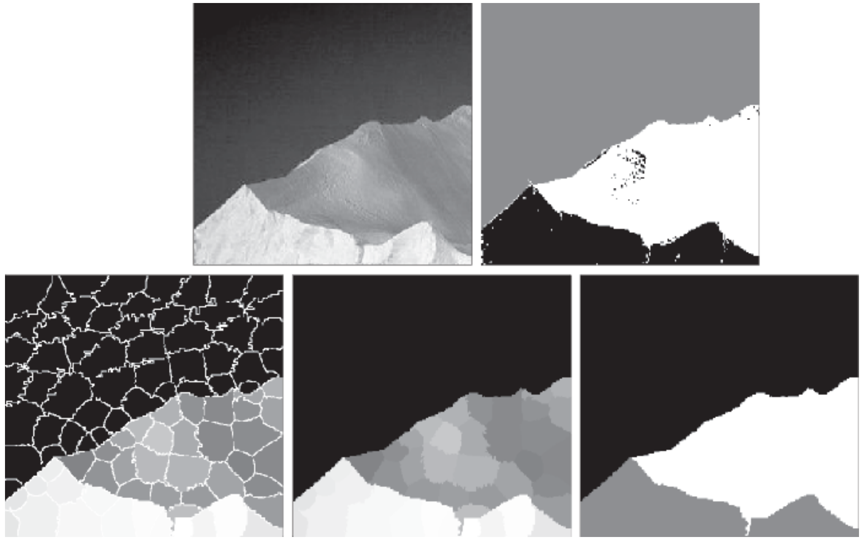
Then

$$D = \left[\left(\frac{d_c}{d_{cm}}\right)^2 + \left(\frac{d_s}{d_{sm}}\right)^2\right]^{1/2} = \left[\left(\frac{d_c}{c}\right)^2 + \left(\frac{d_s}{s}\right)^2\right]^{1/2} \Rightarrow D = \left[{d_c}^2 + \left(\frac{d_s}{s}\right)^2 c^2\right]^{1/2}$$

Where

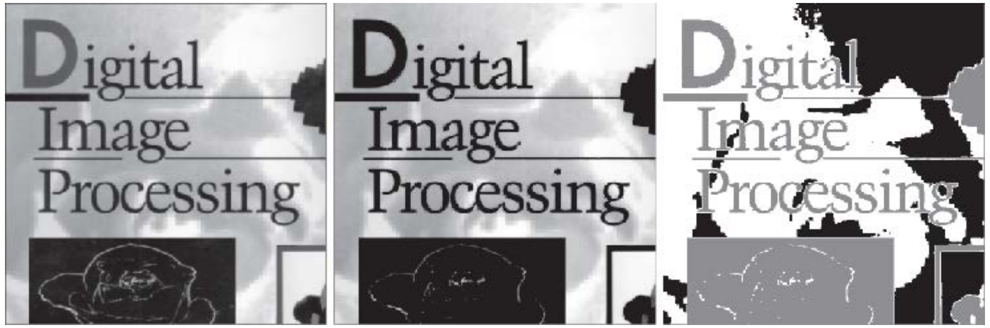
for grayscale image: $d_c = \left[(l_j - l_i)^2\right]^{1/2}$

for 3D image (supervoxel): $d_s = \left[(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2\right]^{1/2}$



a b
c d e

FIGURE 10.53 (a) Image of size 533 × 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.)



a b c

FIGURE 10.54 (a) Image of size 688 × 688 (473,344) pixels. (b) 95000-element superpixel image. (c) Segmentation of the superpixel image using the k-mean algorithm with k = 3.

上海科技大学
ShanghaiTech University

Segmentation using Graph Cuts (图分割)

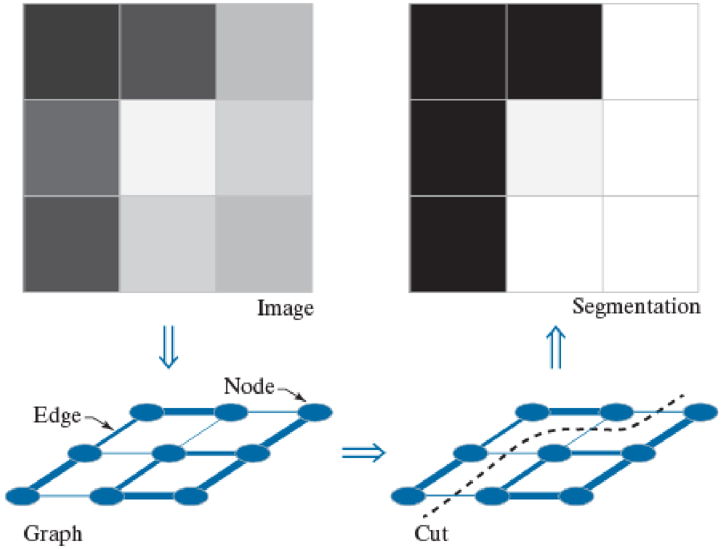
- **Images as graphs**
  - A **graph** G is a mathematical structure and $G = (V, E)$, where

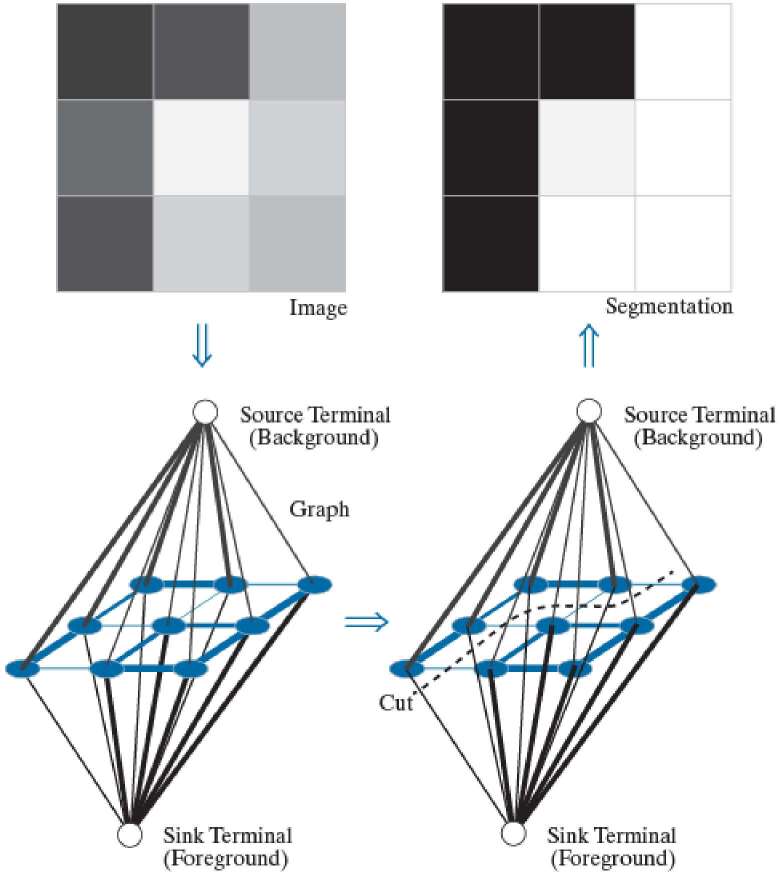
    V: a set of **nodes** or vortex

    E: a set of **edges** (ordered pairs of elements from V)
  - G is undirected if $(u, v) \in E$ implies that $(v, u) \in E$ and vice versa, or G is directed : otherwise
  - W is a symmetric matrix which characterizes the edges in G, where the element $w(i, j)$ is a weight associated with the edge that connects nodes i and j, and $w(i, j) = w(j, i)$ due to undirected.

a b
c d

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





a b
c d

FIGURE 10.56
(a) Same image as in Fig. 10.55(a).
(c) Corresponding graph and terminal nodes. (d) Graph cut. (b) Segmented image.

上海科技大学 ShanghaiTech University

Segmentation using Graph Cuts (图分割)

- **Graph Cuts**

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

Where $A \cup B = V$ and $A \cap B = \emptyset$

The normalized cut ($Ncut$)

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

Where

$$assoc(A,V) = \sum_{u \in A, z \in V} w(u,z) \qquad assoc(B,V) = \sum_{v \in B, z \in V} w(v,z)$$

- **Computing Graph Cuts**

$$Ncut(A,B) = \frac{\sum_{x_i>0, x_j<0} -w(i,j)x_i x_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0, x_j>0} -w(i,j)x_i x_j}{\sum_{x_i<0} d_i}$$

Where

$$d_i = \sum_j w(i,j)$$

$x_i = 1$ if node n_i is in A, and $x_i = -1$ if node n_i is in B

- **Generalized eigen-system**

$$(\boldsymbol{D} - \boldsymbol{W})\mathbf{x} = \lambda \boldsymbol{D}\mathbf{x}$$

Where $\boldsymbol{D}$: a $K \times K$ diagonal matrix with main diagonal elements d_i

$\boldsymbol{W}$: a $K \times K$ weight matrix with elements $w(i,j)$

$\mathbf{x}$: K dimensional indicator vector

K: the number of nodes in V

The objective is to find a vector $\mathbf{x}$ that minimizes $\boldsymbol{Ncut(A,B)}$ (Minimum Graph Cut)

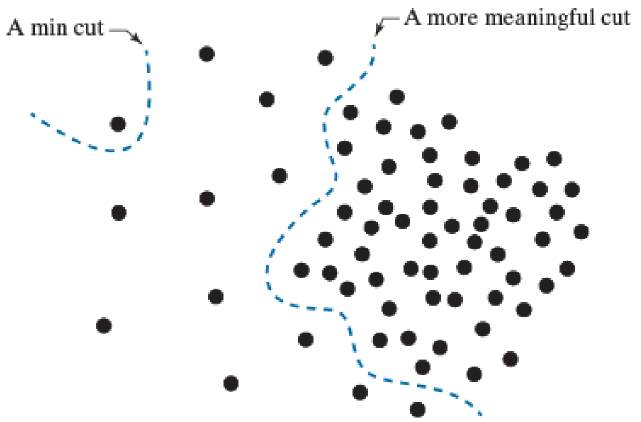


FIGURE 10.57 An example showing how a min cut can lead to a meaningless segmentation. In this example, the similarity between pixels is defined as their spatial proximity, which results in two distinct regions.

Segmentation using Graph Cuts (图分割)

➢ Graph Cut segmentation algorithm

1. Specify a weighted graph, $G = (V, E)$ in which V contains the points in the feature space, and E contains the edges of the graph. Compute the edge weights and use them to construct matrices $\boldsymbol{W}$ and $\boldsymbol{D}$. Let N denote the desired number of partitions of the graph.
2. Solve the eigenvalue system $(\boldsymbol{D} - \boldsymbol{W})\mathbf{x} = \lambda \boldsymbol{D}\mathbf{x}$ to find the eigenvector with the second smallest eigenvalue.
3. Use the eigenvector from Step 2 to bipartition the graph by finding the splitting point such that $Ncut(A, B)$ is minimized.
4. If the number of cuts has not reached N, decide if the current partition should be subdivided by checking the stability of the cut.
5. Recursively repartition the segmented parts if necessary.

➢ Specifying weight for graph cut segmentation

$$w(i,j) = \begin{cases} e^{-\frac{[I(n_i)-I(n_j)]^2}{\sigma_I^2}} e^{-\frac{dist(n_i,n_j)}{\sigma_d^2}} & \text{if } dist(n_i,n_j) < r \\ 0, & \text{otherwise} \end{cases}$$

Where $I(n_i)$: intensity of node n_i

σ_I^2 and σ_d^2 : the constants determining the spread of the two Gaussian-like functions

$dist(n_i, n_j)$: the distance (Euclidean distance) between the two nodes

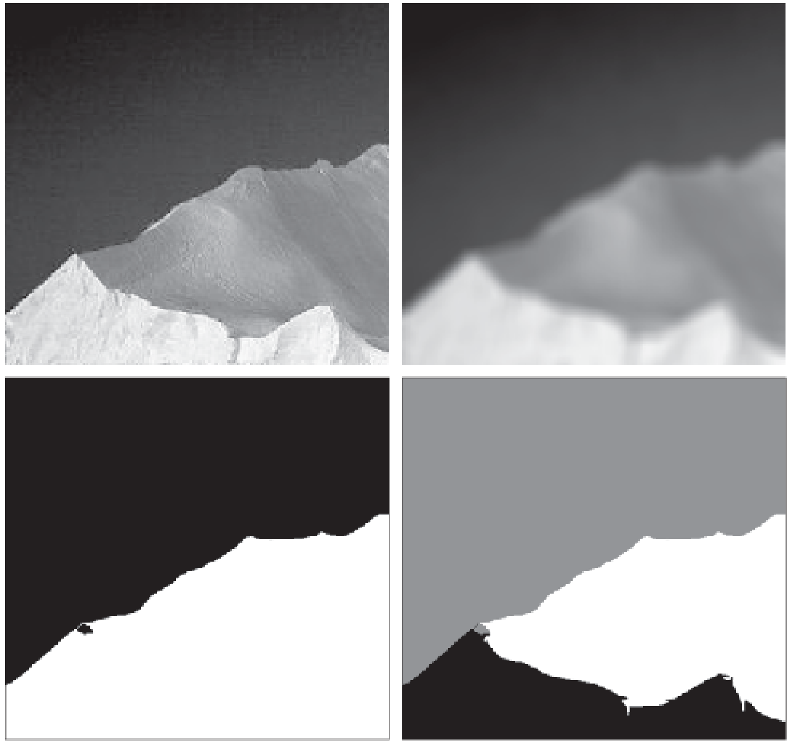
r : a radial constant that establishes how far away the similarity is considered.

Segmentation using Graph Cuts (图分割)

- Graph Cuts are ideally suited for obtaining a rough segmentation of principal regions in an image, especially for tasks such as providing broad cues for autonomous navigation, for searching image databases, and for low-level image analysis.

a b c

FIGURE 10.58 (a) Image of size 600 × 600 pixels. (b) Image smoothed with a 25 × 25 box kernel. (c) Graph cut segmentation obtained by specifying two regions.

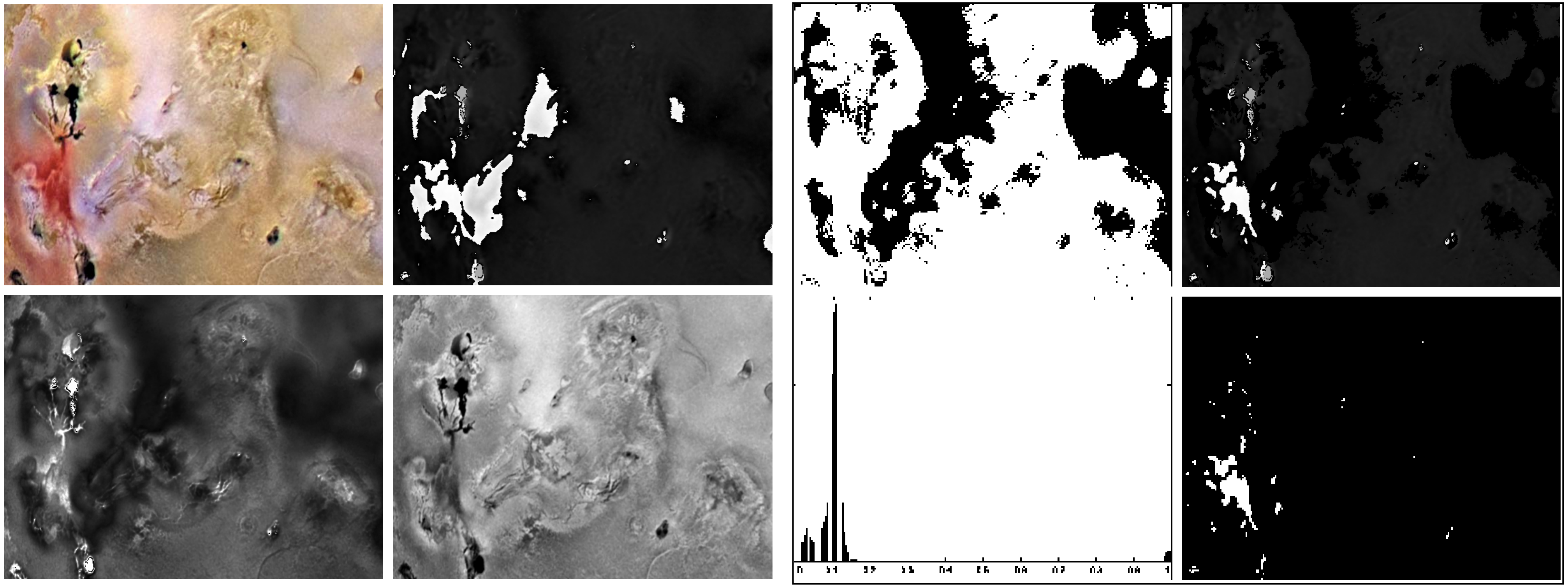


a b
c d

FIGURE 10.59 (a) 678 × 720 image of an iceberg. (b) Image smoothed using a 25 × 25 box kernel. (c) Result of graph cut segmentation with two regions specified. (d) Result of segmentation with three regions specified. [Compare (d) and Fig. 10.42(c).] (Original image courtesy of NOAA.)

Color Image Segmentation

- **In HSI color space**
  - Saturation is used as masking image to isolate further ROI in the hue image;



a b e f
c d g h

FIGURE 7.40 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components from (a).

Color Image Segmentation

➢ In RGB color space

- The Euclidean distance between z and a (estimate of average color)

$$D(z,a) = \|z - a\| = [(z-a)^T(z-a)]^{1/2}$$
$$= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{1/2}$$

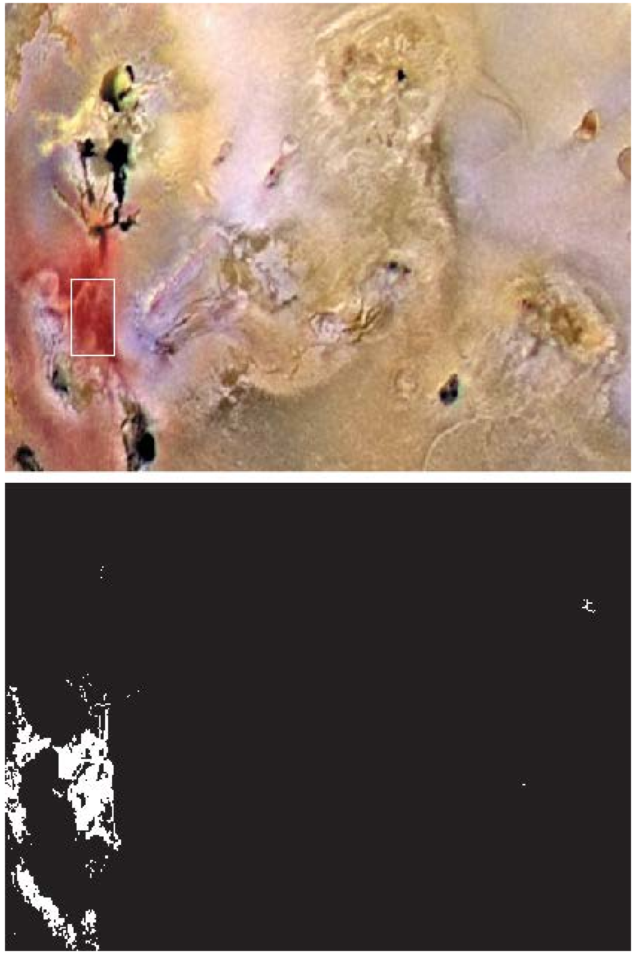
- z is similar to a if $D(z,a) \le D_0$

- A useful generalization of distance measure

$$D(z,a) = [(z-a)^T C^{-1} (z-a)]^{1/2}$$

where C is the covariance matrix of the samples chosen to be representative of the color range to be segmented.

- Bounding box (边界盒)
  a) center at a
  b) edge along color axes and proportional to the standard deviation of the samples.

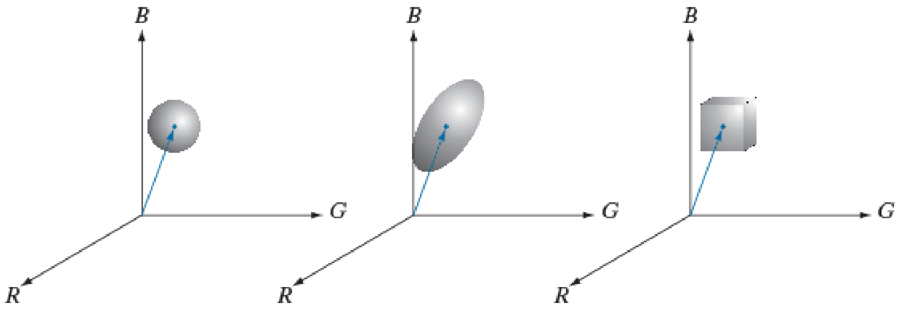


a
b

FIGURE 7.42 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 7.40(h).

a b c

FIGURE 7.41 Three approaches for enclosing data regions for RGB vector segmentation.



Color Image Segmentation

a b
c d

FIGURE 7.44 (a) RGB image. (b) Gradient computed in RGB color vector space. (c) Gradient image formed by the elementwise sum of three individual gradient images, each computed using the Sobel operators. (d) Difference between (b) and (c).

a b c

FIGURE 7.45 Component gradient images of the color image in Fig. 7.44. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 7.44(c).

Color edge detection

- The maximum rate of change of $f(x,y)$ at location (x,y) is given by

$$\theta(x,y) = \frac{1}{2}\tan^{-1}\left[\frac{2g_{xy}}{g_{xx}-g_{yy}}\right]$$

$$F_\theta(x,y) = \left\{\frac{1}{2}\left[\left(g_{xx}+g_{yy}\right)+\left(g_{xx}-g_{yy}\right)\cos 2\theta(x,y)+2g_{xy}\sin 2\theta(x,y)\right]\right\}^{1/2}$$

where

$$\boldsymbol{u} = \frac{\partial R}{\partial x}\boldsymbol{r} + \frac{\partial G}{\partial x}\boldsymbol{g} + \frac{\partial B}{\partial x}\boldsymbol{b} \qquad \boldsymbol{v} = \frac{\partial R}{\partial y}\boldsymbol{r} + \frac{\partial G}{\partial y}\boldsymbol{g} + \frac{\partial B}{\partial y}\boldsymbol{b}$$

$$g_{xx} = \boldsymbol{u}\cdot\boldsymbol{u} = \boldsymbol{u}^T\boldsymbol{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2$$

$$g_{yy} = \boldsymbol{v}\cdot\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2$$

$$g_{xx} = \boldsymbol{u}\cdot\boldsymbol{v} = \boldsymbol{u}^T\boldsymbol{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$