Feature Extraction 1

李东晓

lidx@zju.edu.cn

浙江大学信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

Contents

- Advanced Image Segmentation
  - 10.5 Region Segmentation Using Clustering and Superpixels
  - 10.6 Region Segmentation Using Graph Cuts

Region Segmentation Using K-Means Clustering

- Set of vector observations $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_Q\}$
- Vector component ←→ pixel attribute (R,G,B)
- Partition Q into k disjoint cluster sets

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$$

$$C = \{C_1, C_2, \ldots, C_k\}$$

So that the following criterion of optimality is satisfied

$$\arg\min_{C} \left(\sum_{i=1}^{k} \sum_{\mathbf{z} \in C_i} \|\mathbf{z} - \mathbf{m}_i\|^2 \right)$$

NP hard!

Standard K-Means Clustering

1. Initialize randomly $\mathbf{m}_i(1),\ i = 1, 2, ..., k$
2. Assign each sample to the closest cluster

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

3. Update the cluster centers (means)

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

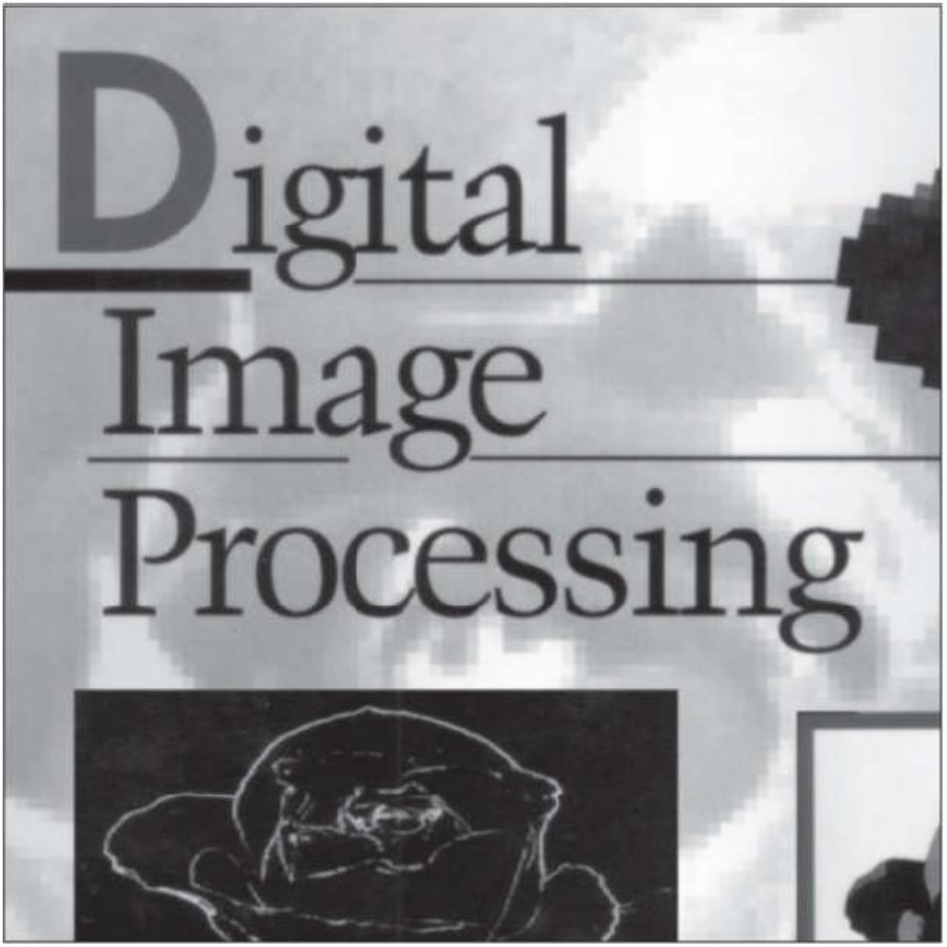
4. Test for completion

E: Euclidean norms of the differences between the mean vectors in the current and previous steps

if (E ≤ T)||(max iterations) {stop}; else {go to step 2}

K-Means Clustering Example

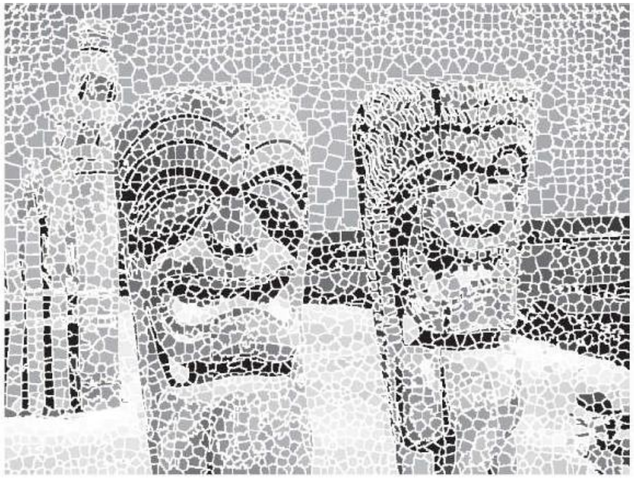
- Image size: 688 x 688
- K-means clustering of intensity: $k = 3$



Region Segmentation using Superpixels

- grouping pixels into primitive regions that are more perceptually meaningful than individual pixels
- Improve segmentation by reducing irrelevant detail
- Reduce computational load


600x800=480,000 pixels


4,000 superpixels


Superpixel image

Region Segmentation using Superpixels

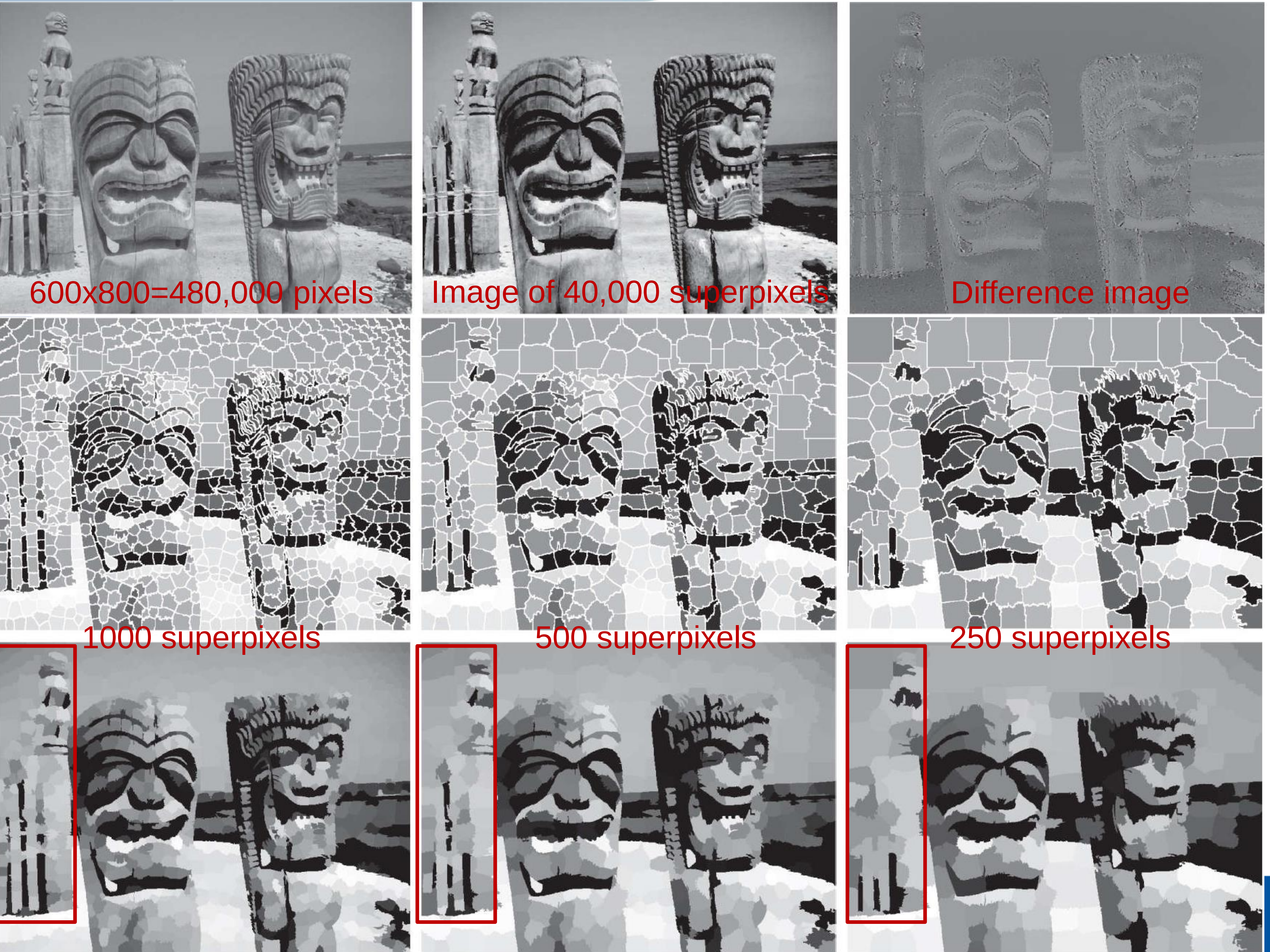
- Adherence to boundaries

Boundaries between regions of interest must be preserved in a superpixel image

600x800=480,000 pixels

Image of 40,000 superpixels

Difference image


600x800=480,000 pixels
Image of 40,000 superpixels
Difference image
1000 superpixels
500 superpixels
250 superpixels

Simple Linear Iterative Clustering (SLIC)

- Modified K-means clustering
  - Total number of pixels n_{tp}
  - Desired number of superpixels n_{sp}

$$s = [n_{tp} / n_{sp}]^{1/2}$$

1. **Initialize the algorithm:** Compute the initial superpixel cluster centers,

$$\mathbf{m}_i = [r_i \ \ g_i \ \ b_i \ \ x_i \ \ y_i]^T, i = 1, 2, \ldots, n_{sp}$$

   by sampling the image at regular grid steps, *s*. Move the cluster centers to the lowest gradient position in a 3×3 neighborhood. For each pixel location, *p*, in the image, set a label $L(p) = -1$ and a distance $d(p) = \infty$.
2. **Assign samples to cluster centers:** For each cluster center $\mathbf{m}_i$, $i = 1, 2, \ldots, n_{sp}$, compute the distance, $D_i(p)$ between $\mathbf{m}_i$ and *each* pixel *p* in a $2s \times 2s$ neighborhood about $\mathbf{m}_i$. Then, for each *p* and $i = 1, 2, \ldots, n_{sp}$, if $D_i < d(p)$, let $d(p) = D_i$ and $L(p) = i$.
3. **Update the cluster centers:** Let C_i denote the set of pixels in the image with label $L(p) = i$. Update $\mathbf{m}_i$:

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

   where $|C_i|$ is the number of pixels in set C_i, and the **z**'s are given by **Eq. (10-86)** .
4. **Test for convergence:** 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 n_{sp} norms. If $E < T$, where *T* a specified nonnegative threshold, 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, $\mathbf{m}_i$.

Specifying the Distance Measure

- Euclidean distance of color

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

- Spatial Euclidean distance

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

3D: Supervoxel

$$d_s = [(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2]^{1/2}$$

- Composite distance

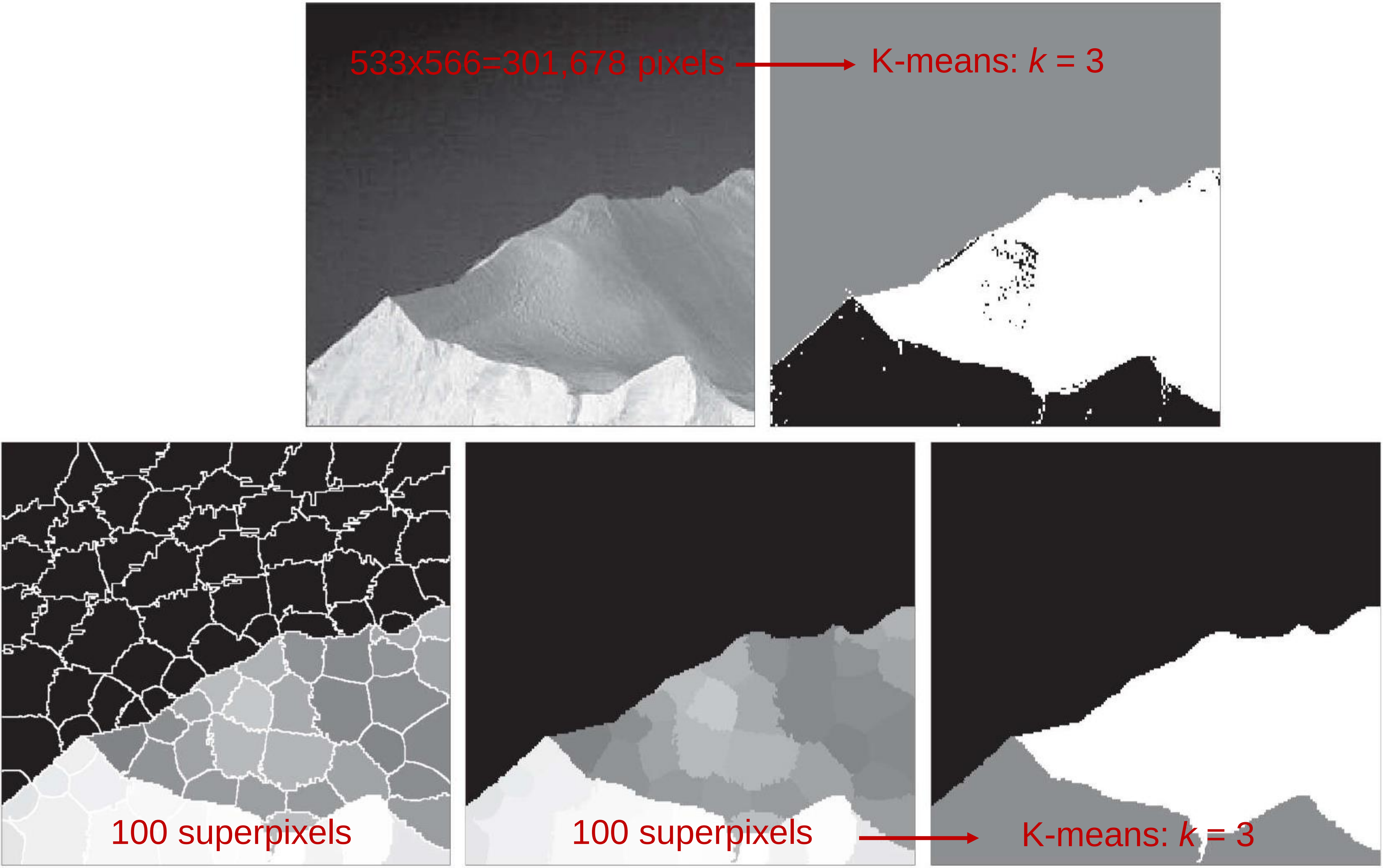
$$D = \left[\left(\frac{d_c}{d_{cm}}\right)^2 + \left(\frac{d_s}{d_{sm}}\right)^2\right]^{1/2} \Rightarrow D = \left[\left(\frac{d_c}{c}\right)^2 + \left(\frac{d_s}{s}\right)^2\right]^{1/2}$$

weight between color and spatial similarity

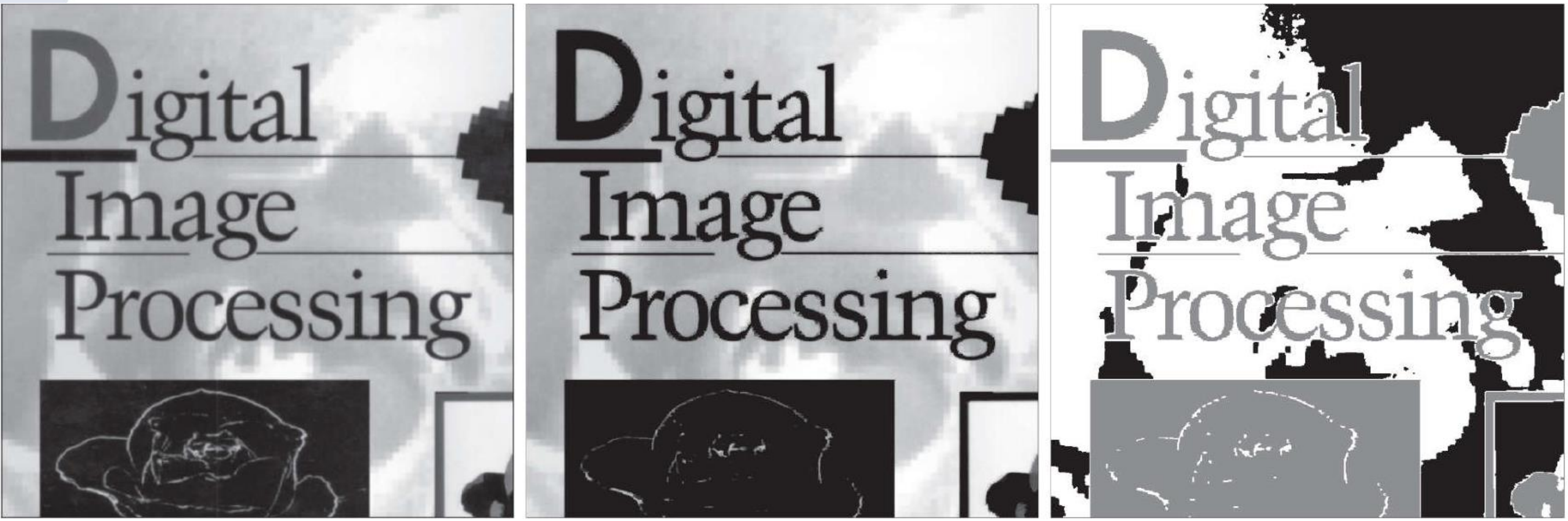
d_{cm} and d_{sm} are the maximum expected values of d_c and d_s

Segmentation Examples

Superpixel-based Segmentation Example



Superpixel-based Segmentation Example



688x688=473,344 pixels

95,000 superpixels → K-means: k = 3

Visually quite similar

Region Segmentation Using Graph Cuts

- Image → weighted, undirected graph G = (V, E)

  V: graph nodes ← image pixels / superpixels

  $E \subseteq V \times V$: graph edges ← 4 / 8 – adjacency

  $w(i, j)$: edge weight ← similarity between nodes

  inverse of difference, correlation

- Segmentation → find Graph Cuts

$$A \cup B = V \ \text{ and } \ A \cap B = \emptyset$$

  – Background / foreground segmentation

  – Similarity within a subset is high

  – Similarity across subsets is low

浙江大学 信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

Region Segmentation Using Graph Cuts

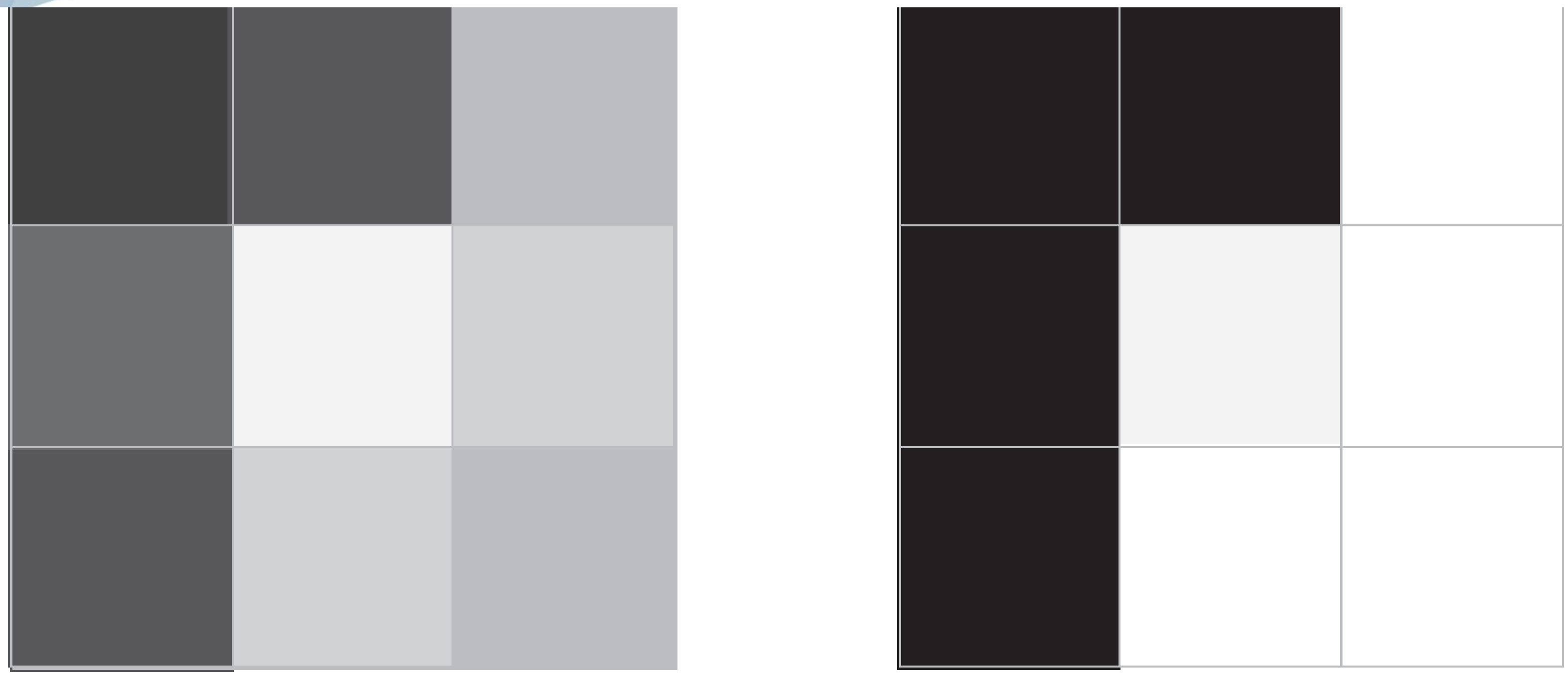
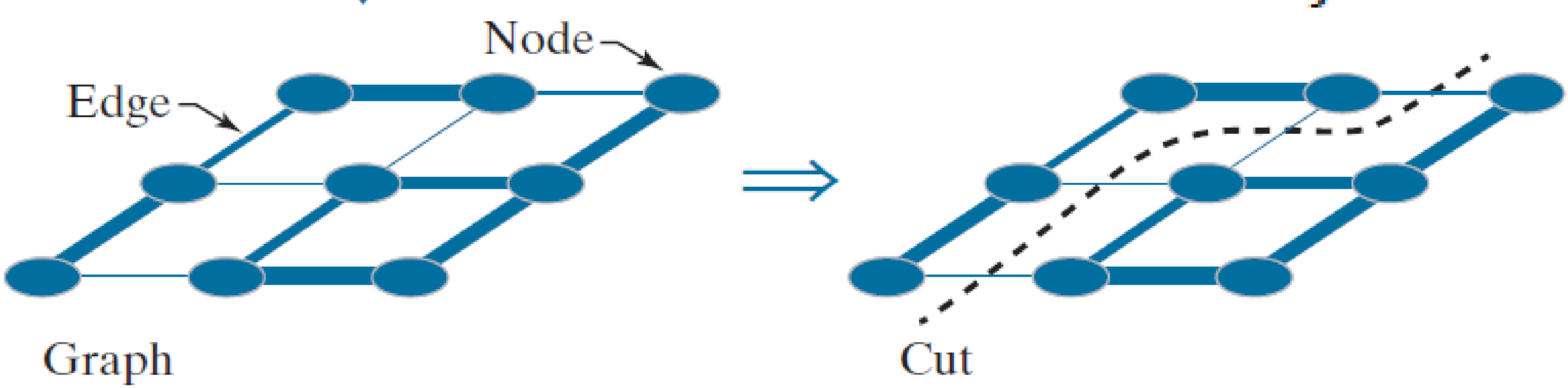


Image Segmentation

$$w(i,j) = 1/(|I(n_i) - I(n_j)| + c)$$



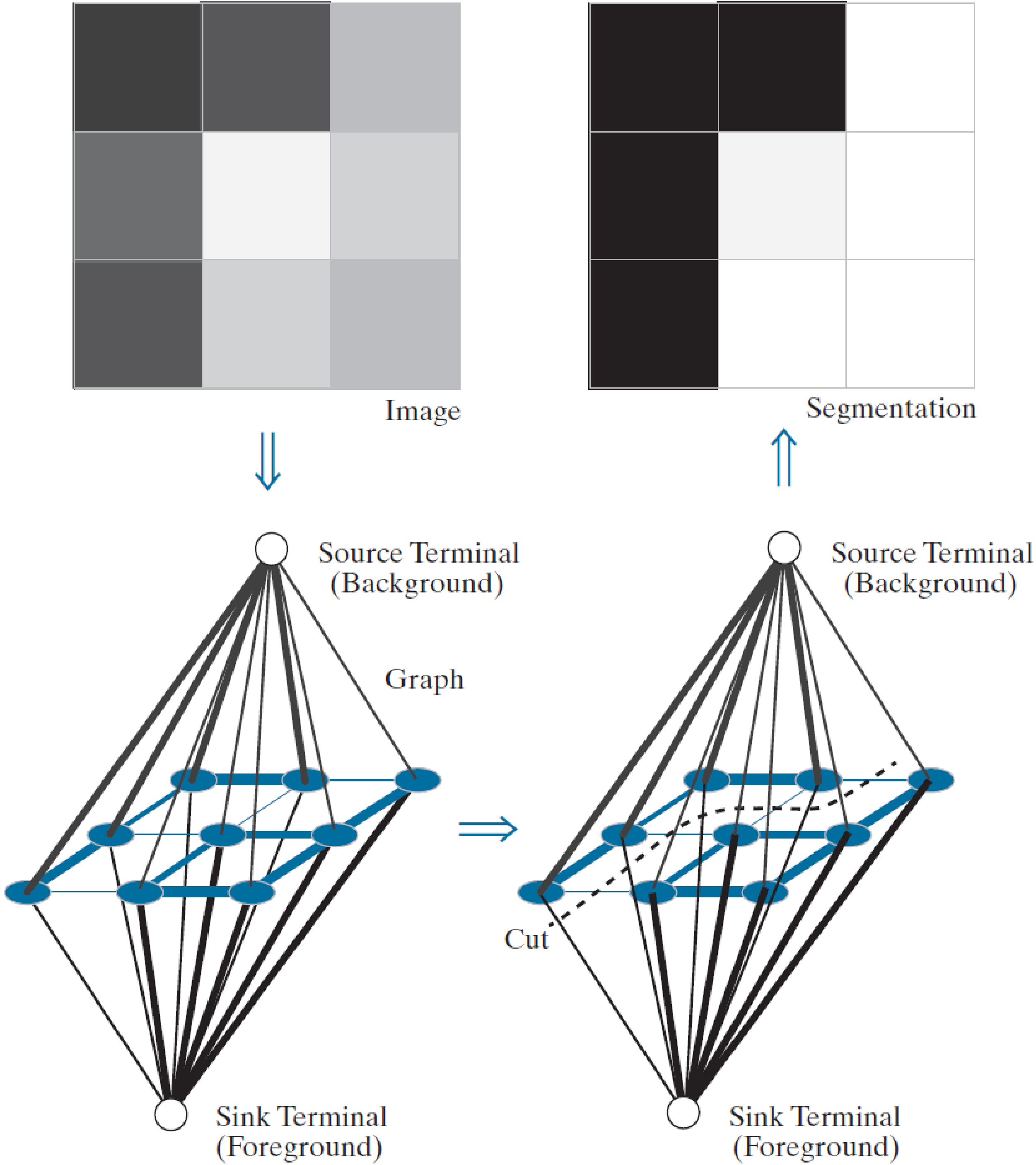


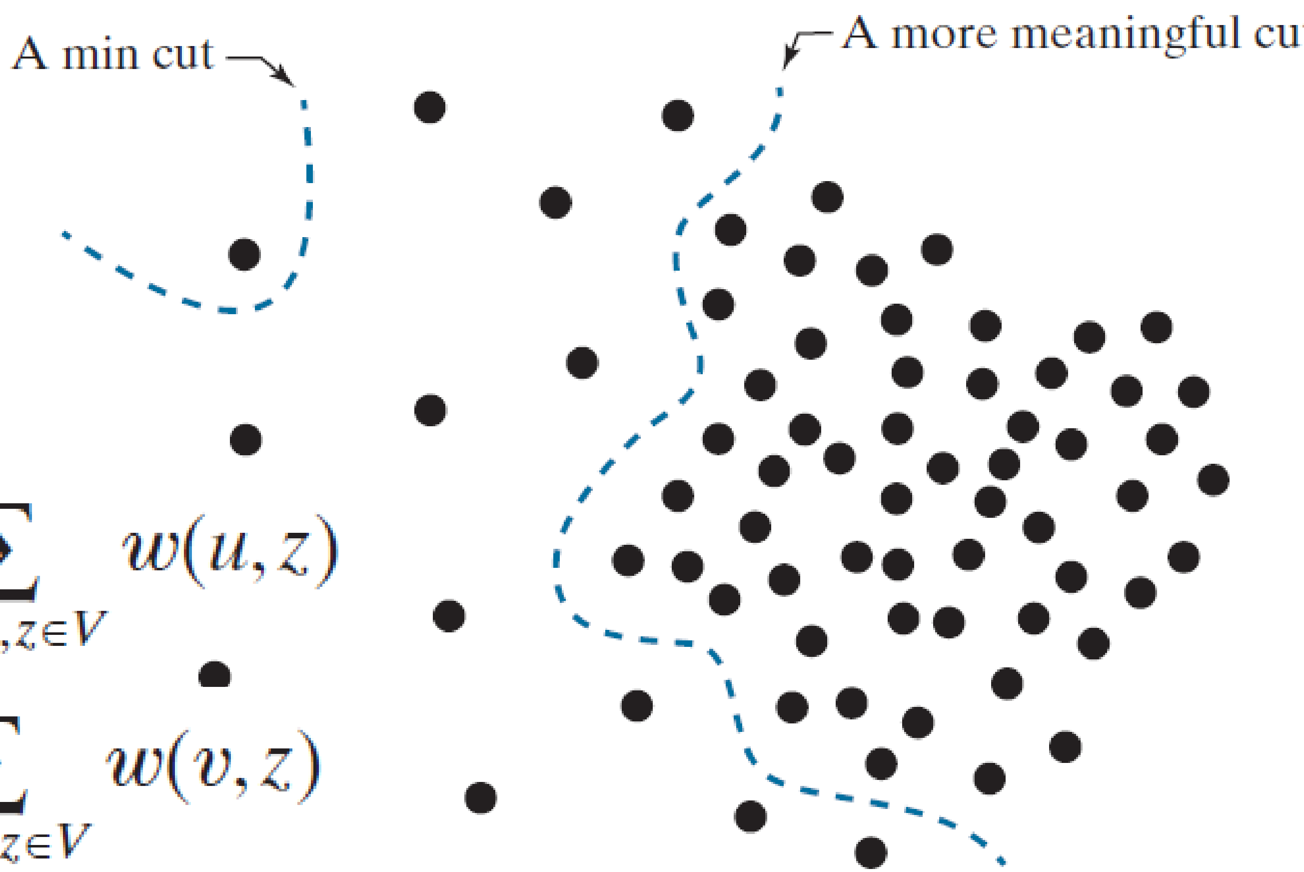
Image
Segmentation
Source Terminal
(Background)
Graph
Sink Terminal
(Foreground)
Source Terminal
(Background)
Cut
Sink Terminal
(Foreground)

Max-Flow, Min-Cut Theorem

Normalized cut

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

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



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

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

NP hard problem

min Normalized cut

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

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

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

max Normalized association

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

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

Computing Minimal Graph Cuts

- Indicator vector **x**:

$$x_i = \begin{cases} 1 & n_i \in A \\ -1 & n_i \in B \end{cases}$$

Find **x**, minimize *Ncut*

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

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

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

Generalized Eigenvalue Problem

- Allow indicator vector **x** to be real, continuous numbers $(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda \mathbf{D}\mathbf{y}$
- Solution of **x** = eigenvector of 2nd smallest eigenvalue
- **D**: *K* x *K* diagonal matrix of $d_i,\ i = 1, 2, \ldots, K$
- **W**: *K* x *K* wieight matrix of $w(i, j)$
- Standard eigenvalue problem

$$\mathbf{A}\mathbf{z} = \lambda \mathbf{z}$$

$$\mathbf{A} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$$

$$\mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y} \xrightarrow[\text{Split point}]{\text{Binarize}} \mathbf{X}$$

edge weight ←→ similarity

- Example

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

Segmentation Using Graph Cuts

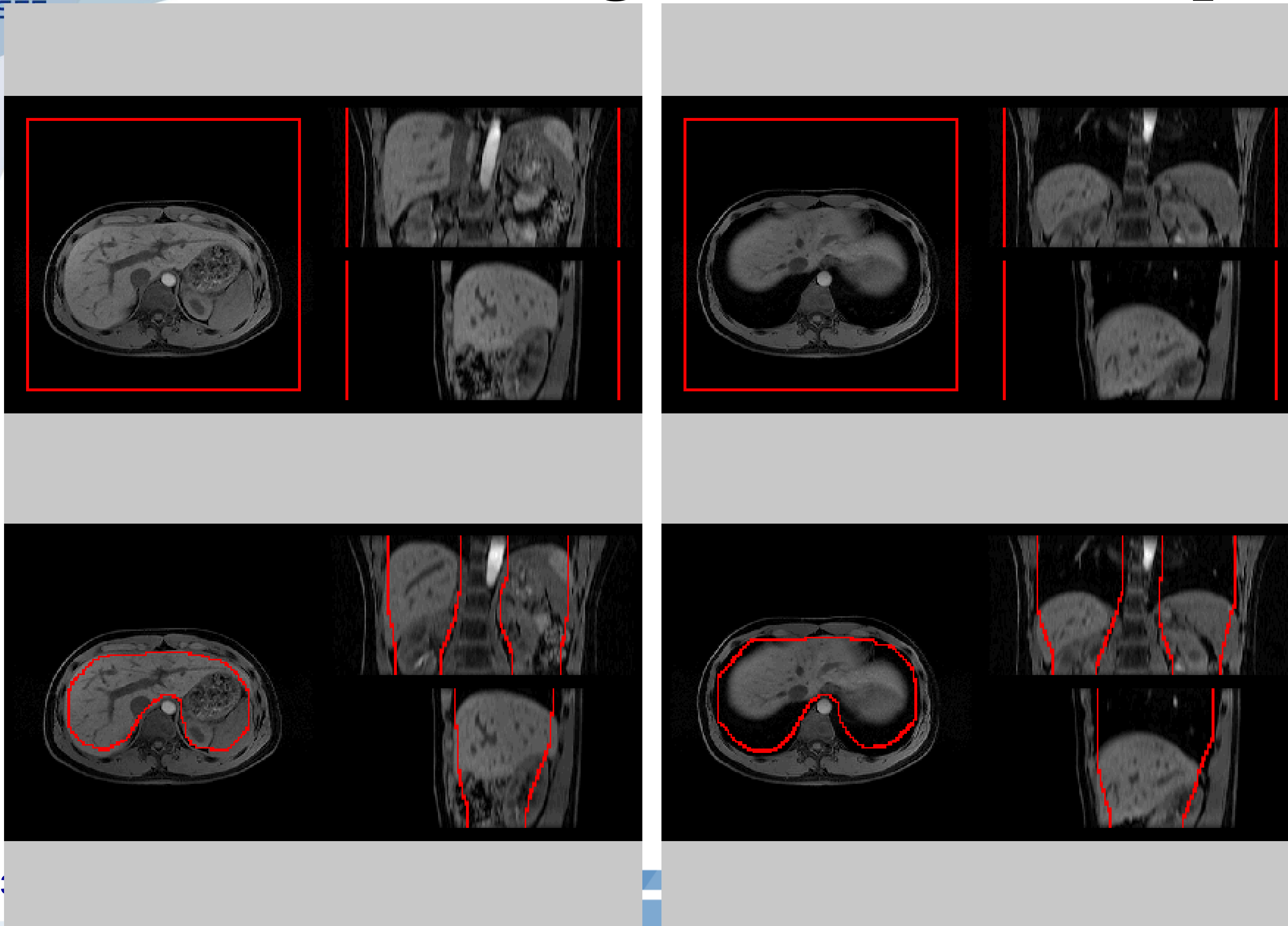
- Advanced Segmentation

Smoothed with 25x25 box kernel

600x600 pixels

Graph cut segmentation
$K = 2$

Segmentation Examples

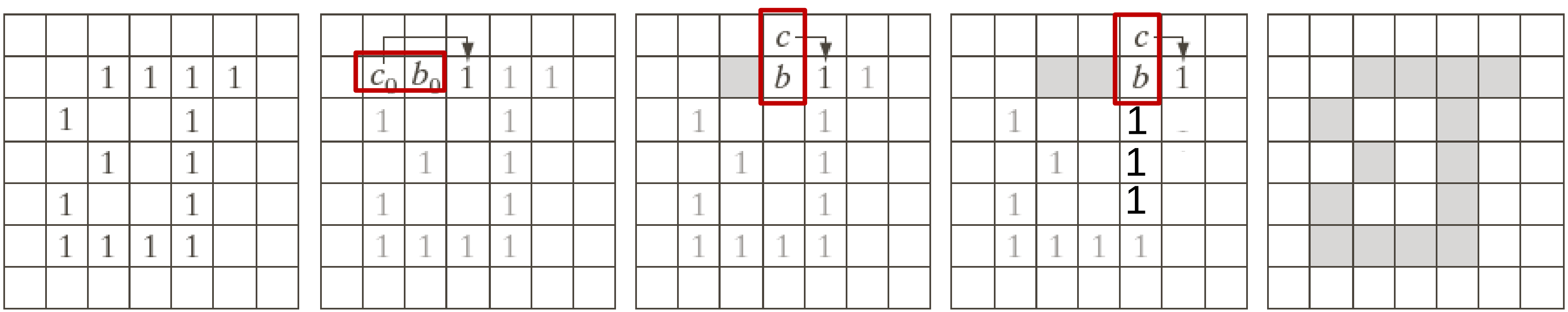


11 Feature Extraction

- Boundary Preprocessing
- Boundary Feature Descriptors
- Regional Feature Descriptors
- Principle Components as Feature Descriptors
- Whole-Image Features
- Scale-Invariant Feature Transform (SIFT)

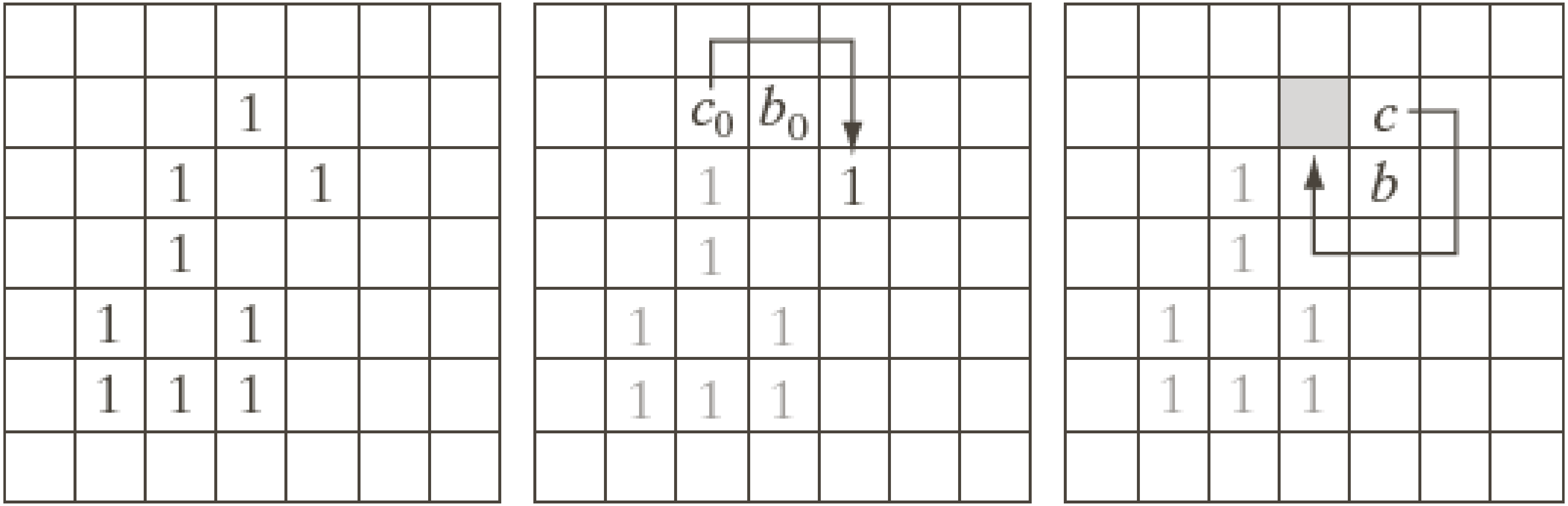
Moore Boundary Following (Tracing)

1. Starting point $b=b_0$: uppermost, leftmost
   $c=c_0$: west neighbor of b_0
2. Let the 8-neighbor of b, starting at c and proceeding in clockwise, be denoted by $n_1, n_2, \ldots n_8$. Find the 1st n_k labeled 1.
3. Let $b=n_k$, $c=n_{k-1}$
4. Repeat step 2 and 3 until return to b_0 & b_1



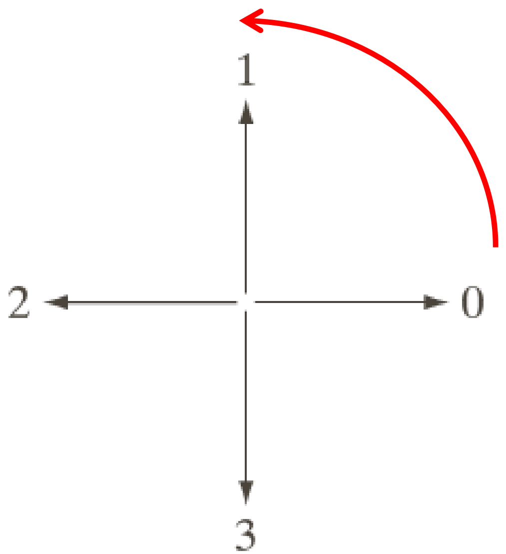
Attention: b_0 & b_1

- Error when the stopping rule is returning to b_0 only

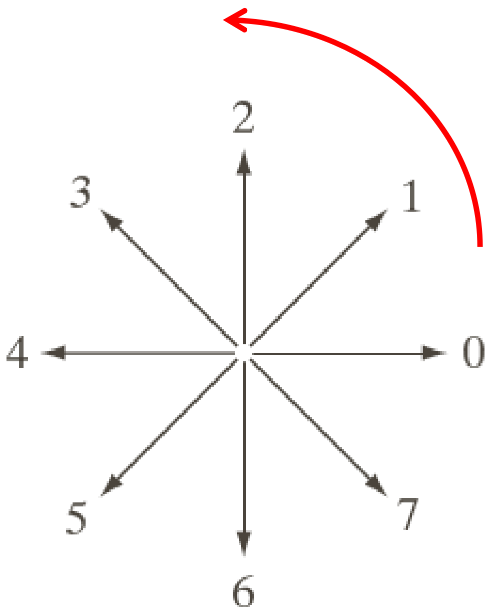


Chain Codes (链码)

- A connected sequence of straight-line segments of specified length and direction
- Freeman chain code: a sequence of directional numbers



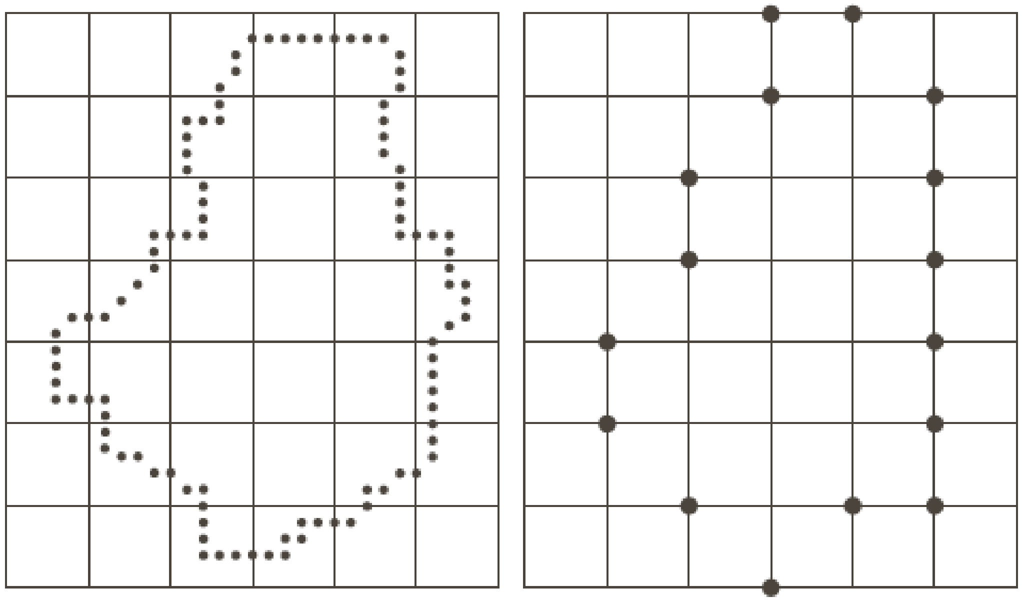
4-directional



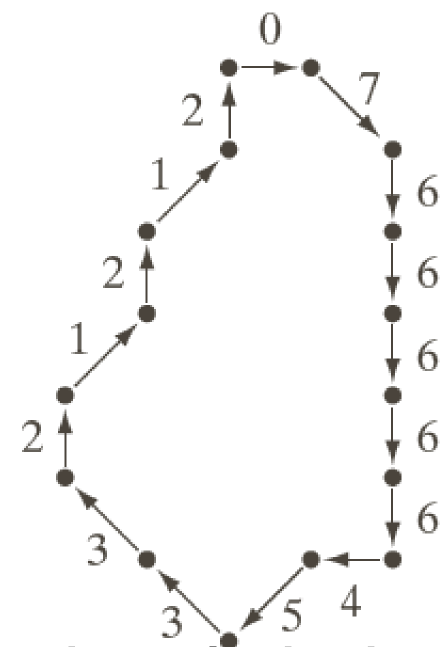
8-directional

Chain Codes

- Resample using a larger grid spacing
  - Short chain
  - Coarse: Robust to noise / imperfect segmentation
- Example chain code: 07666666453321212



resampling



8-directional chain code

Normalization of Chain Codes

- Starting point: circular → minimum integer
- Rotation: 1st difference of the chain code

e.g. 4-direction chain code 10103322

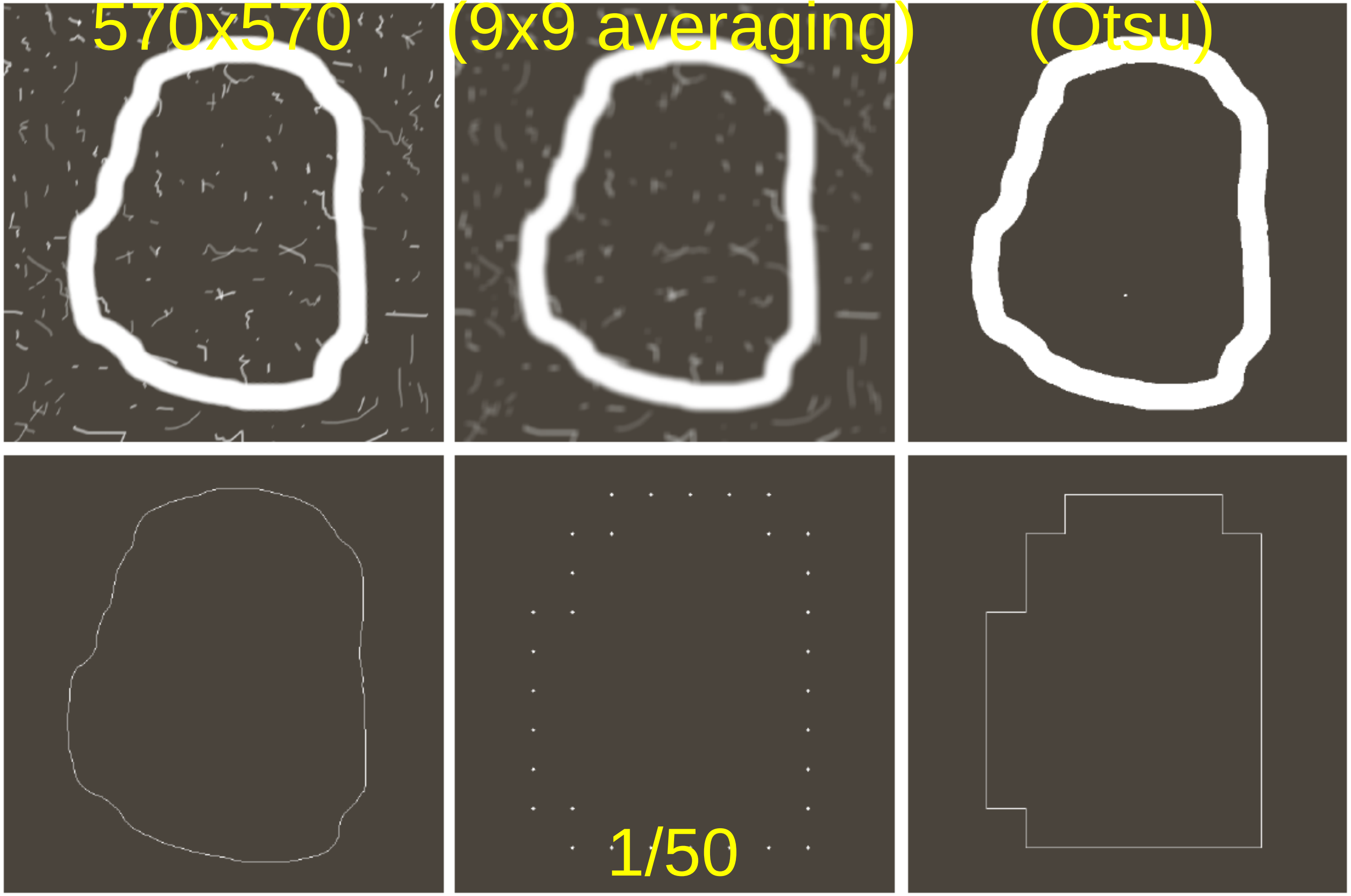
→1st difference: 33133030

Works for integer multiples of 45/90 degree

Orient the resampling grid along the principal / eigen axes

- Size: alter the size of resampling grid

Noisy image Smoothed Segmented



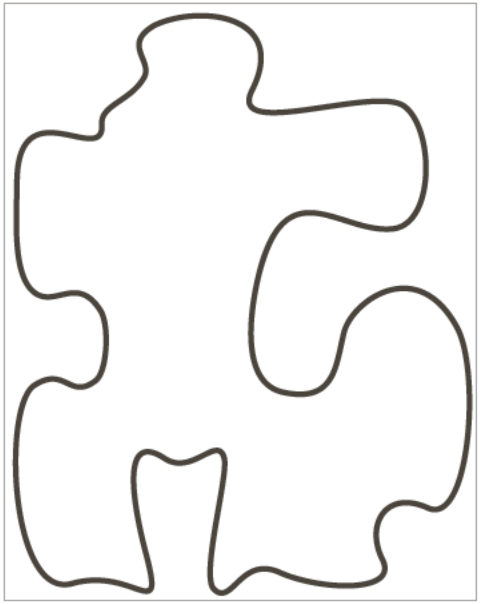
Outer boundary Subsampled Connected

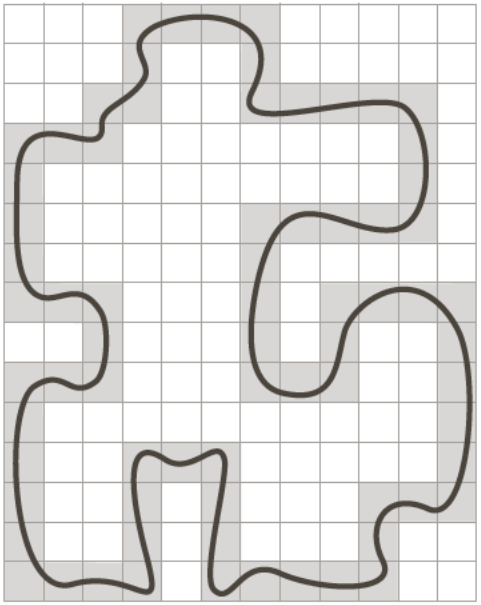
8-dir chain code: 0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2

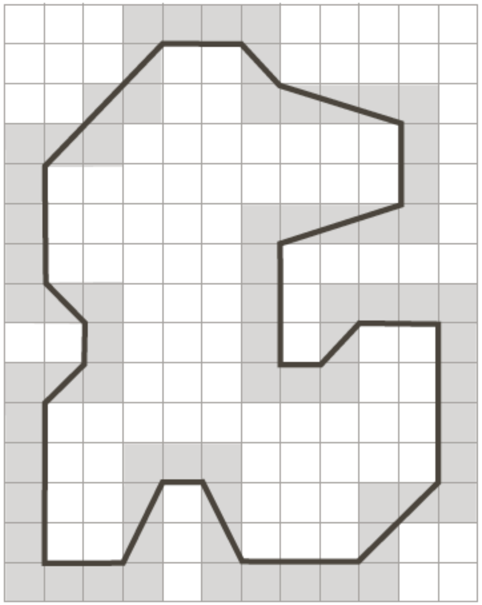
1st difference code: 0 0 0 6 2 6 0 0 0 0 0 0 0 6 0 0 0 0 0 6 2 6 0 0 0 0 6 2 0 6 2 6

Polygonal Approximation using Minimum-Perimeter Polygons (MPP)

- Inner / Outer wall: cell size → accuracy
- Shrink the boundary (rubber band)

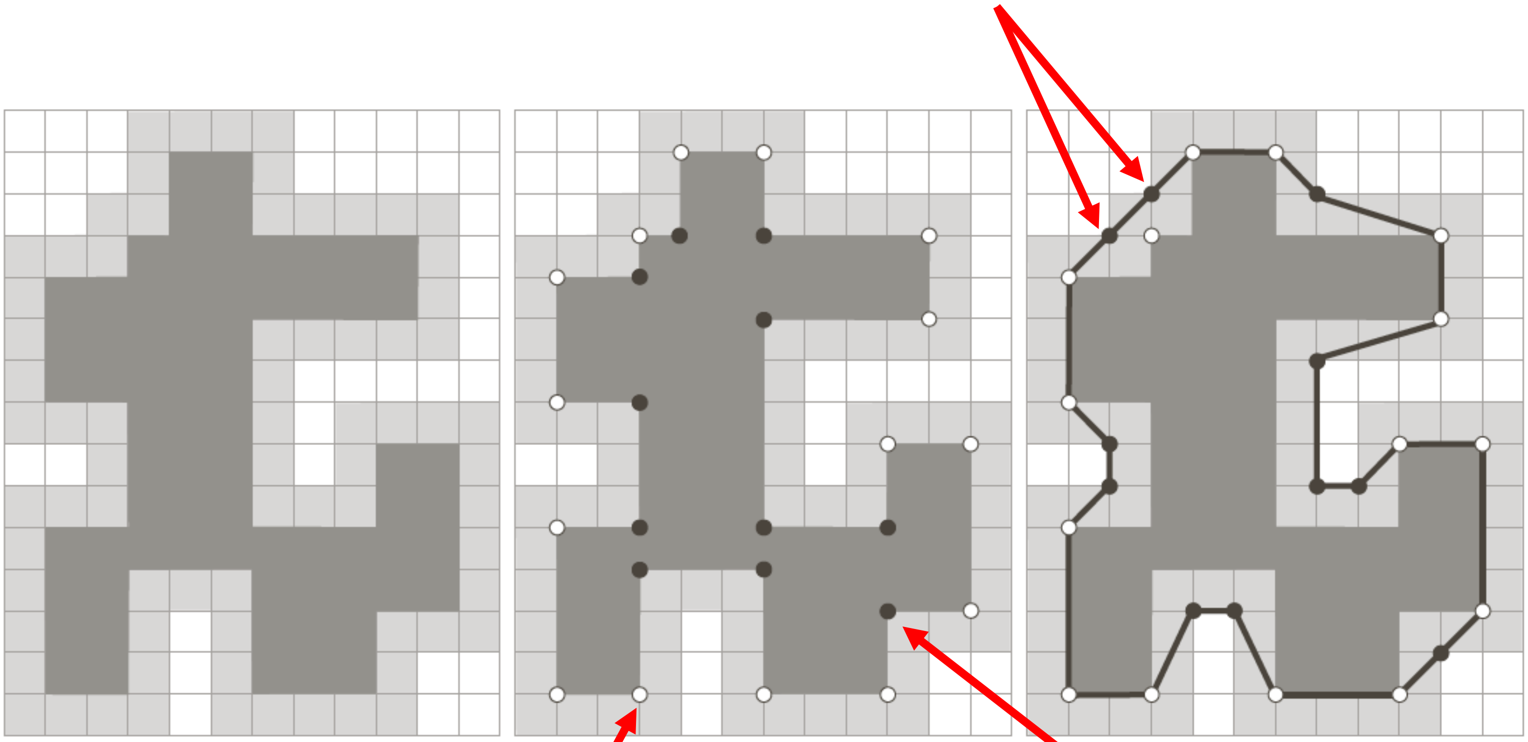

Object boundary


Enclosed by cells


MPP

MPP Approximation

- Concave vertices displaced to their diagonal mirror locations in the outer wall



Region (dark) Convex vertex Concave vertex

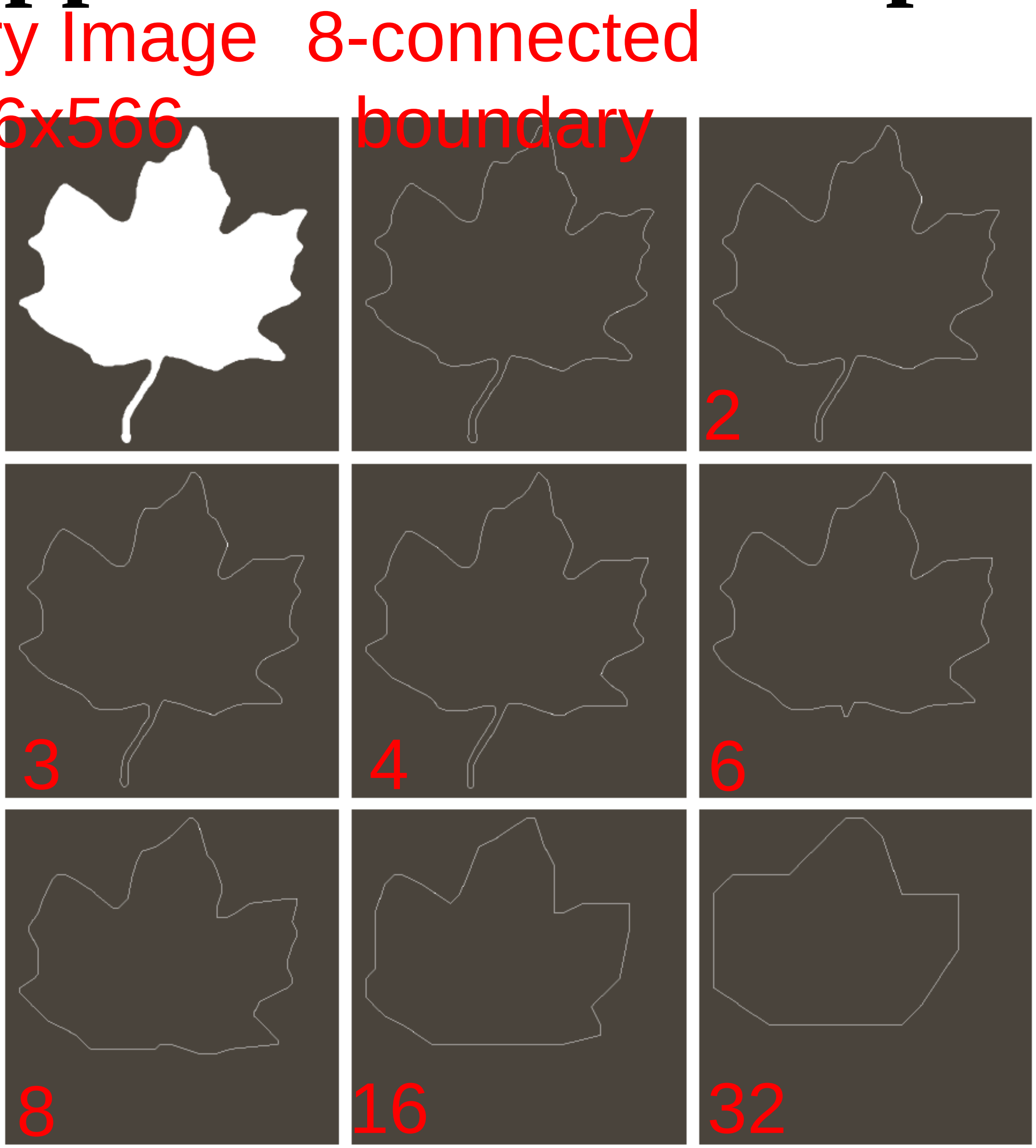
MPP algorithm (P519-521)

- Cellular complex: the set of cells enclosing the boundary
- Assumption & Observations:

1. The MPP bounded by a simply connected cellular complex is not self-intersecting.
2. Every *convex* vertex of the MPP is a *W* vertex, but not every *W* vertex of a boundary is a vertex of the MPP.
3. Every *mirrored concave* vertex of the MPP is a *B* vertex, but not every *B* vertex of a boundary is a vertex of the MPP.
4. All *B* vertices are on or outside the MPP, and all *W* vertices are on or inside the MPP.
5. The uppermost, leftmost vertex in a sequence of vertices contained in a cellular complex is always a *W* vertex of the MPP.

MPP Approximation Example

- Size of cells:
  2,
  3, 4, 6,
  8, 16, 32
- Number of points
  1900, 206,
  160, 127, 92,
  66, 32, 13

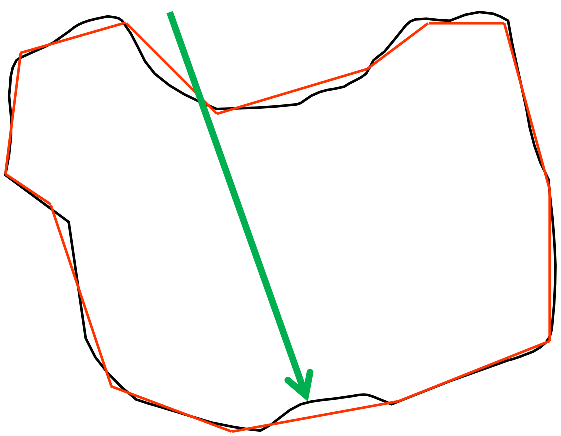


Other Polygonal Approximation

- Merging techniques

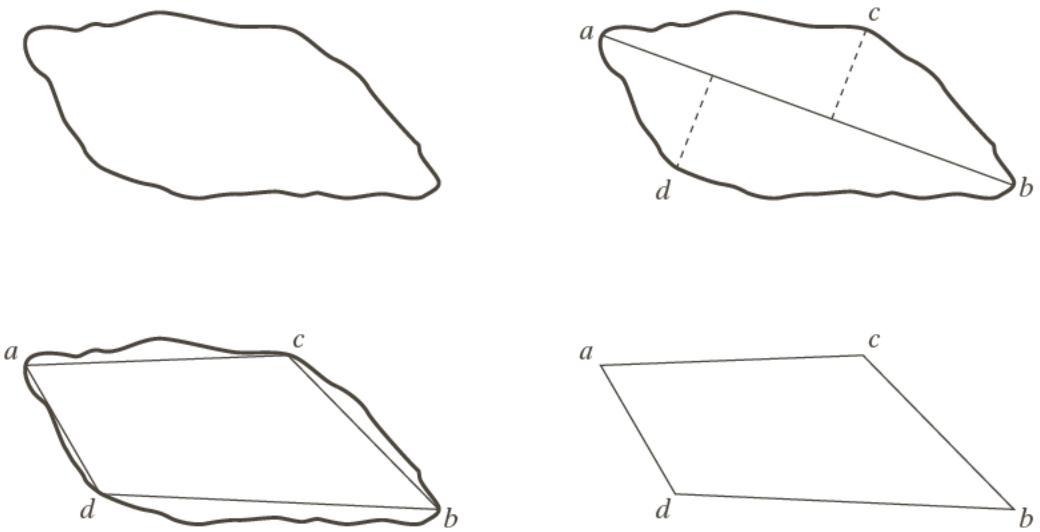
1. Merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold
2. Reset the error and continue merging

- Problem: corner

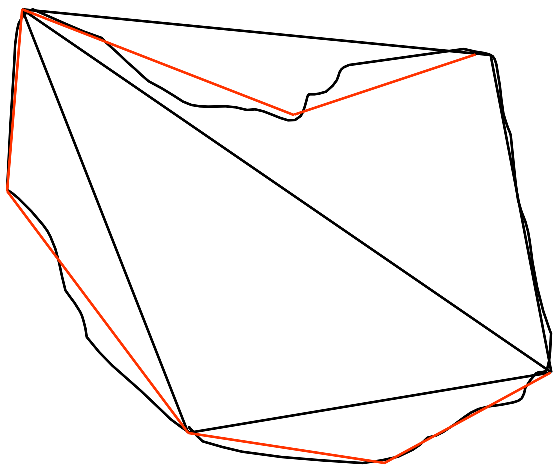


Other Polygonal Approximation

- Splitting techniques
- Splitting – Merging combined techniques



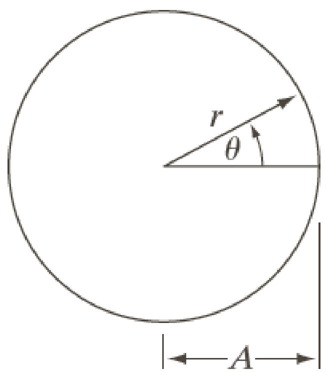
Example 1

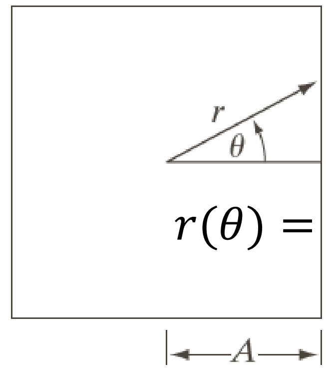


Example 2

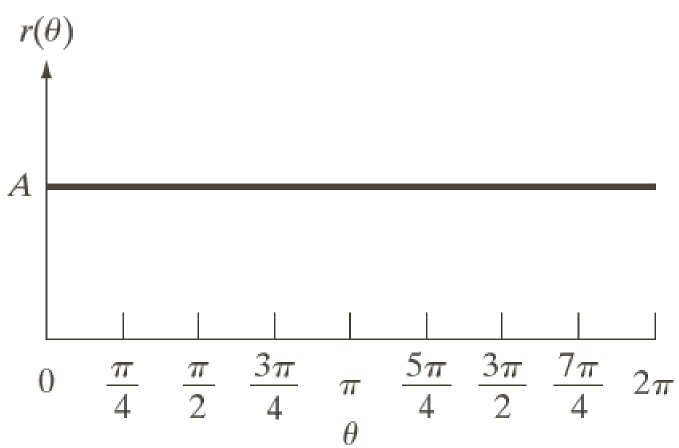
Signatures

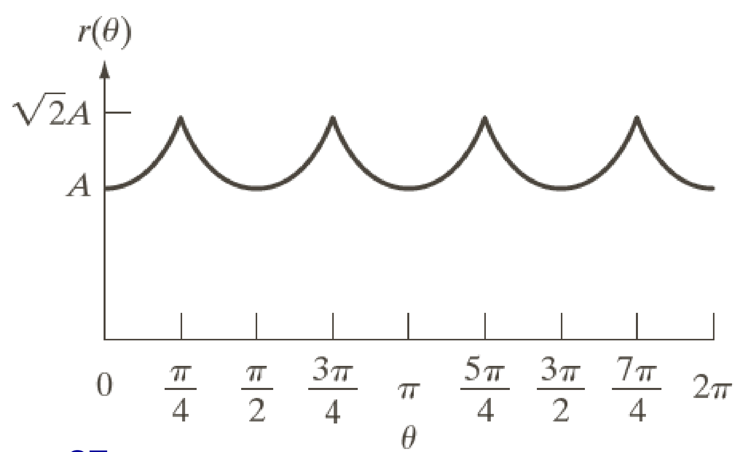
- 1D functional representation of a boundary
- Distance-versus-angle, w.r.t the centroid
- Normalization: starting point, scaling





$$r(\theta)=\begin{cases}\dfrac{A}{\cos\theta}=A\sec\theta & \theta\in[0,\dfrac{\pi}{4}]\\ \dfrac{A}{\sin\theta}=A\csc\theta & \theta\in[\dfrac{\pi}{4},\dfrac{\pi}{2}]\end{cases}$$





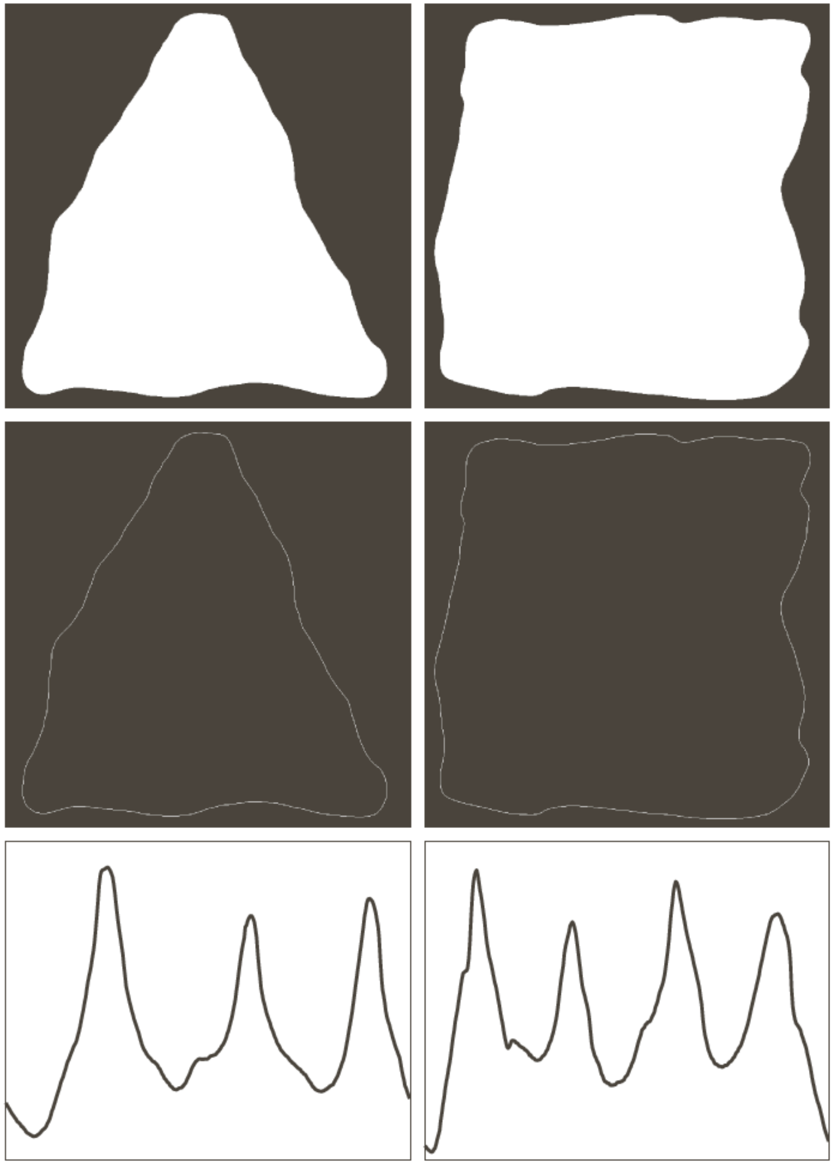
Signatures (cont…)

- Example
  - Three peaks
  - Four peaks

- Other functions

  e.g. edge direction

  - Reference point / line
  - Slope density function

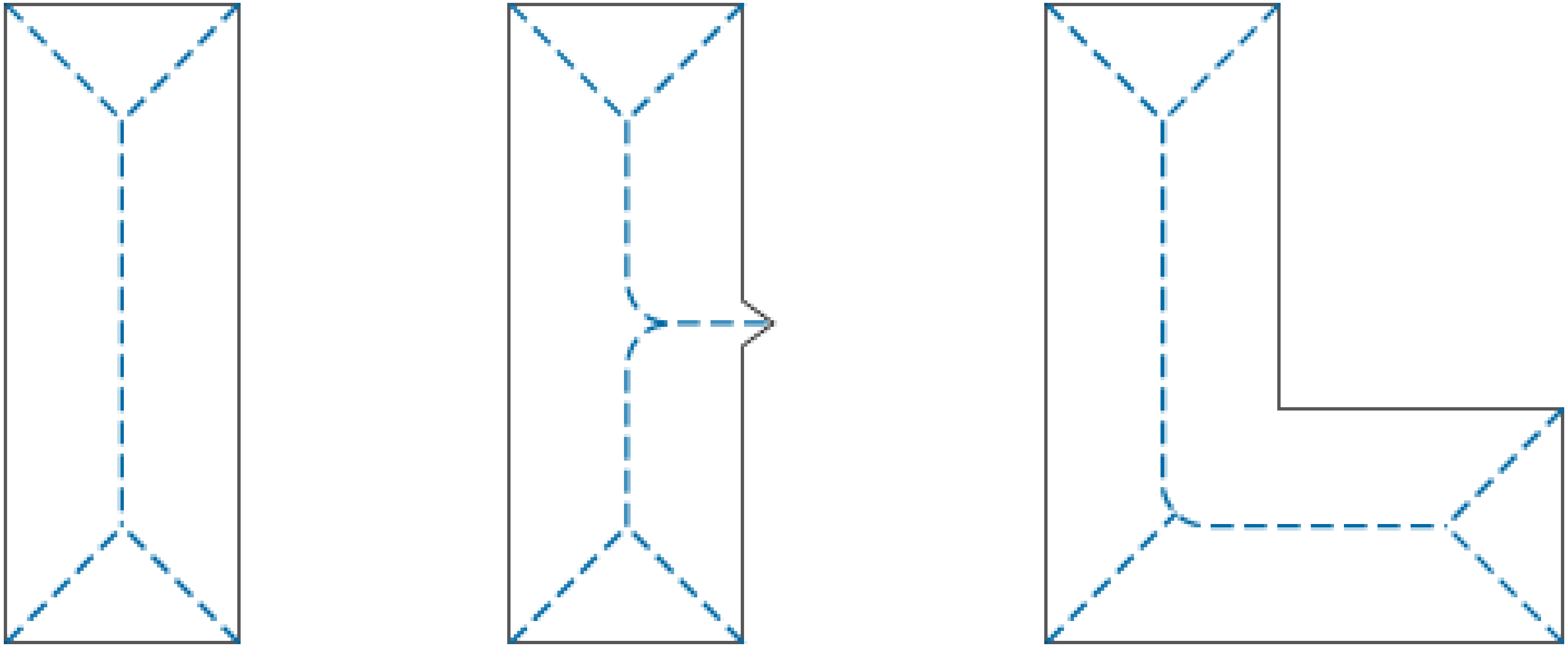


Skeletons

- 中轴变换（**medial axis transform**）可用火烧草地来比拟：

在草地边界上同时点火，火相遇（熄灭）的点就是中轴

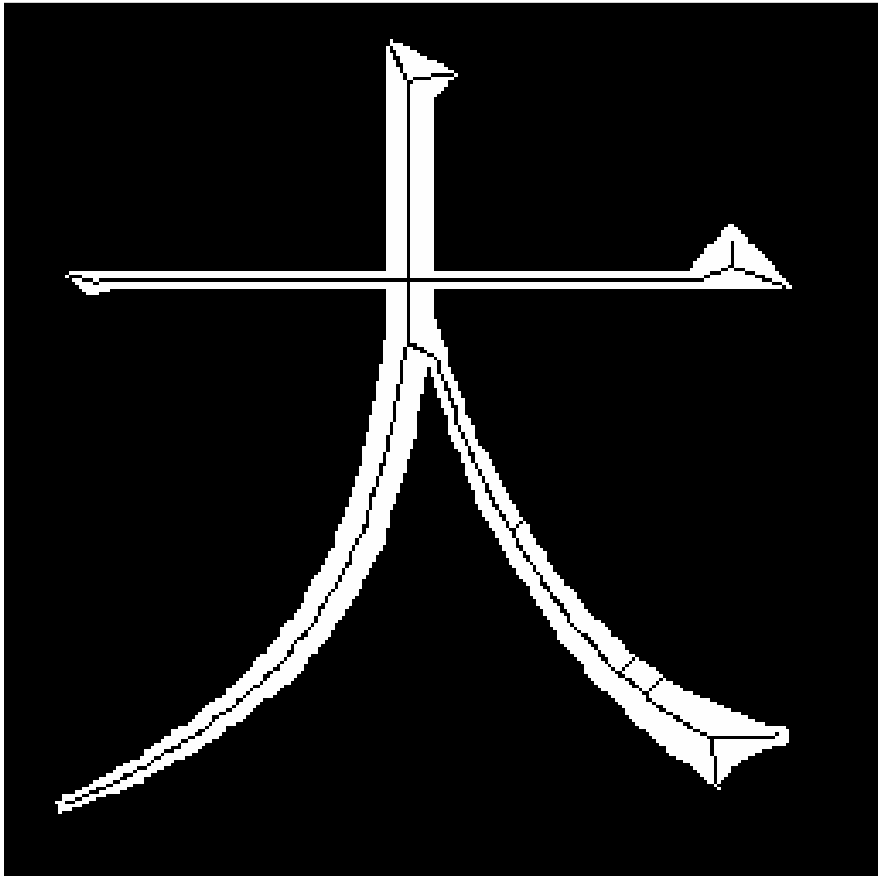
- **Each point in MAT has more than one closest neighbor on the boundary**

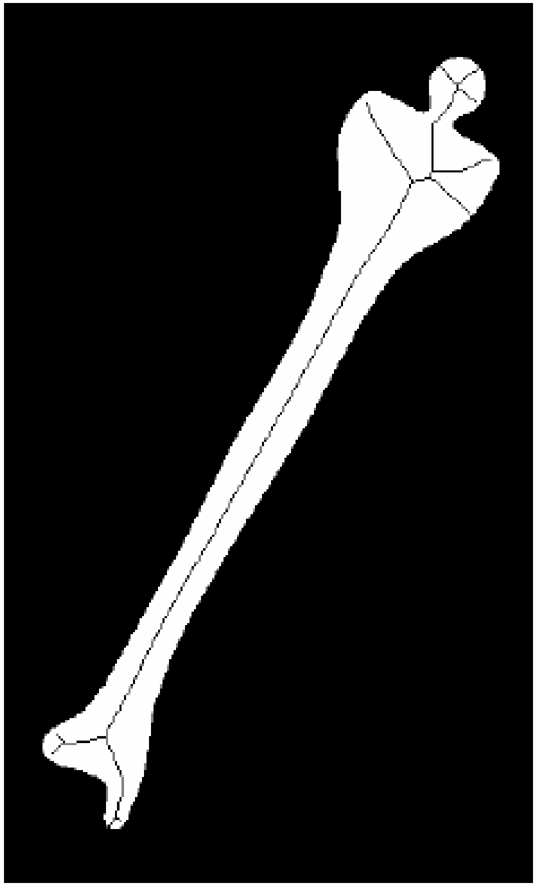


Medial axes

- 中轴:
  - 对于一个区域R，若边界为B，对区域中每一点P，我们在B上搜索与它最近的点，若能找到多于一个同样距离的最近点，则P属于R的中轴或称骨架
  - 在R内作与边界有两个以上切点的内切圆，则所有这些圆的圆心的集合就是中轴
- 对于中轴与边界相连的区域，可以通过以每个骨架点为中心的圆来重建区域，即这些圆的集合的并组成了整个区域
- 用于中轴变换的距离，除欧氏距离外，也可用其它距离
- 相对于细化，中轴变换的定义更严格，实用中可不区分

Skeletonizing Example





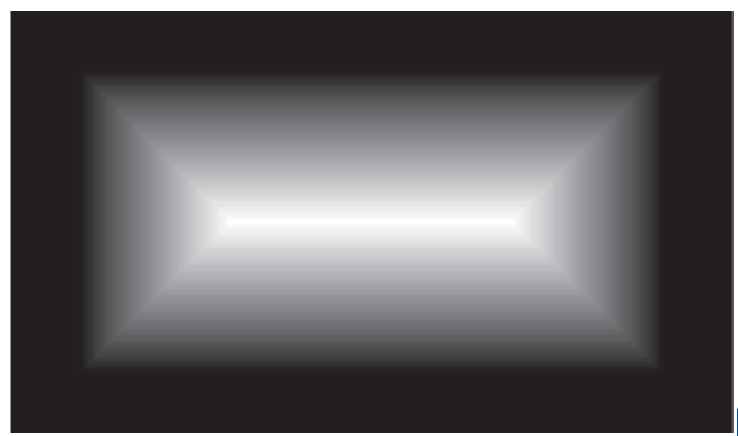
Distance Transform

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

1.41	1	1	1	1.41
1	0	0	0	1
1	0	0	0	1
1.41	1	1	1	1.41

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	1	0
0	1	2	3	3	3	2	1	0
0	1	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0



a	b
c	d

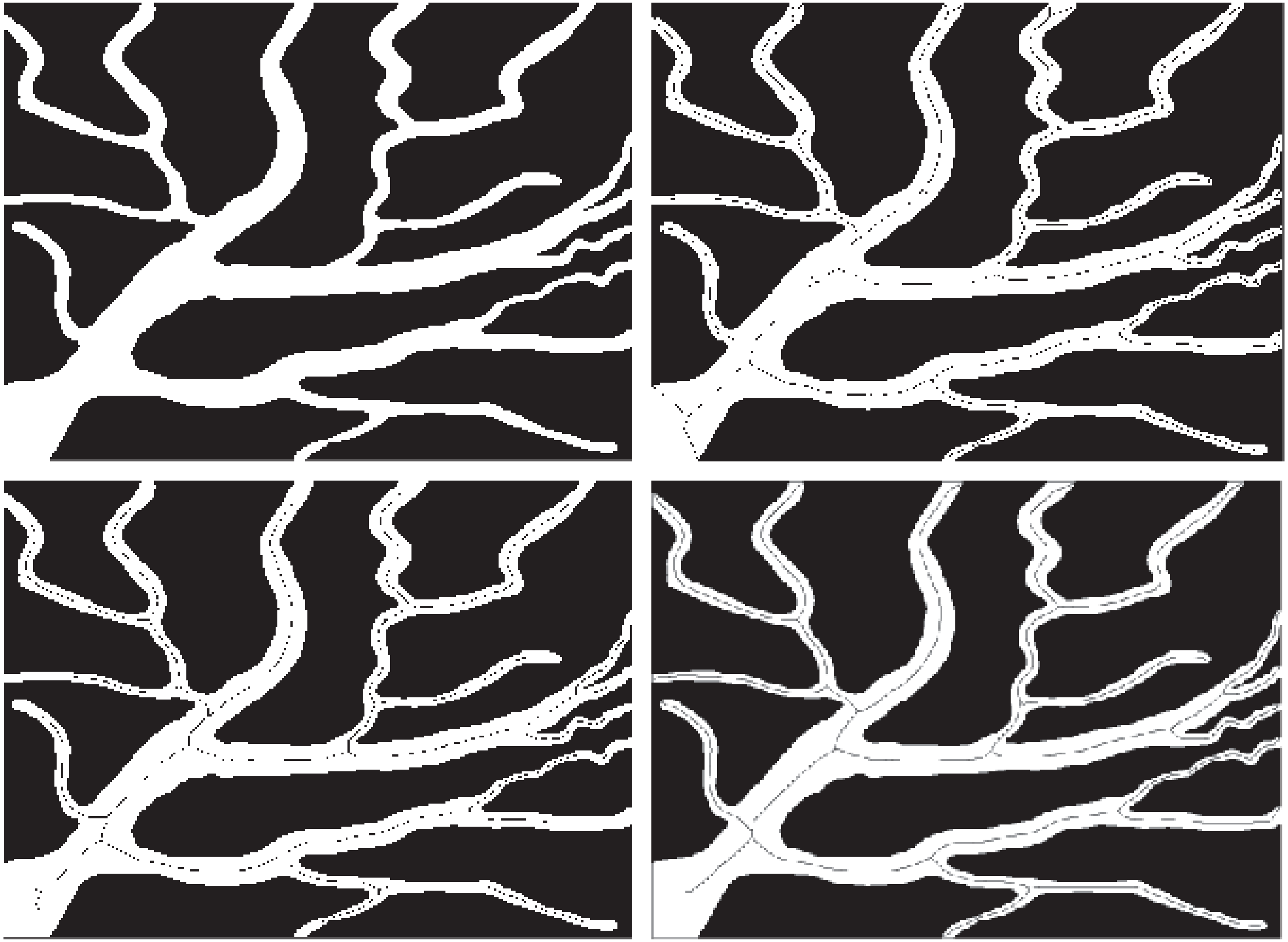


FIGURE 11.14
(a) Thresholded image of blood vessels.
(b) Skeleton obtained by thinning, shown superimposed on the image (note the spurs).
(c) Result of 40 passes of spur removal.
(d) Skeleton obtained using the distance transform.

11 Feature Extraction

- Boundary Preprocessing
- Boundary Feature Descriptors
- Regional Feature Descriptors
- Principle Components as Feature Descriptors
- Whole-Image Features
- Scale-Invariant Feature Transform (SIFT)

Simple Descriptors

1. 边界的长度

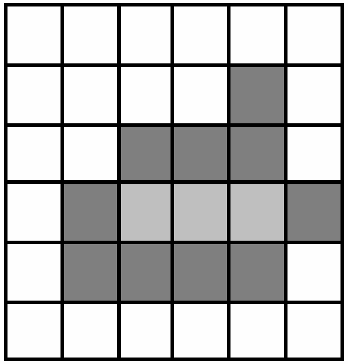
–边界所包围区域的周长

–(1)目标的边界是8连通的，则边界的长度可用八链码长度来近似

- 修正：乘上1.1107

–(2) 8链码中代表斜向的奇数链码长度计为 $\sqrt{2}$ 。这种方法估计的长度比前一种方法误差小些。

- 修正：乘上0.948。



区域的边界

2. 边界的直径

–边界上相隔最远的二点间的距离，即这二点间的直线长度

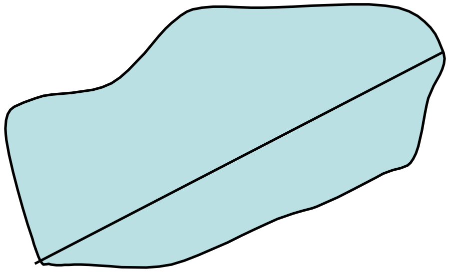
$$\mathrm{Diam}(B) = \max_{i,j}\left[D(p_i, p_j)\right]$$

–也称为边界的主轴或长轴

–其长度和方向是描述边界的一种特征

–对长度的测量常用欧氏距离，也可用街区距离（水平和垂直距离之和）

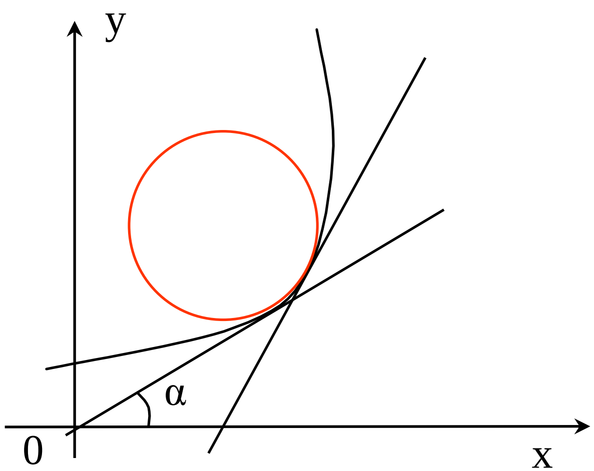
或棋盘距离：D_8（p,q)=max(|X_p-X_q|,| Y_p-Y_q |)



3. 曲率

– 边界的切线方向沿曲线的变化率

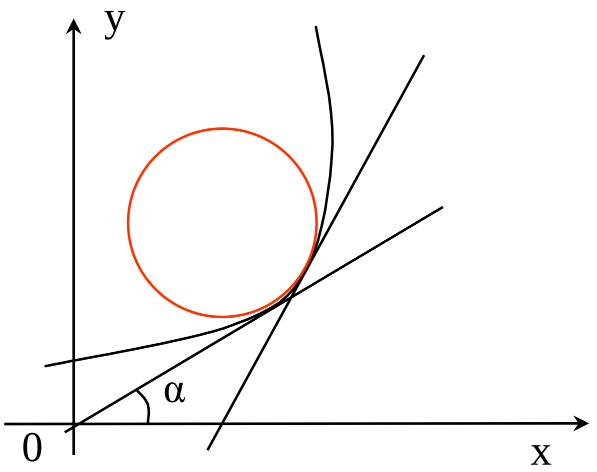
$$K=\frac{d\alpha}{ds}=\frac{y''}{(1+y'^2)^{\frac{3}{2}}}$$



边界的曲率

– 数字图像
  - 可先对边界进行分段拟合，再计算曲率，
  - 可通过估计边界上的切向和弧长来近似计算曲率
– 曲率半径
  - 是相同曲率的圆的半径，表征了边界的弯曲程度
  - 曲率与半径互为倒数 $R=\frac{1}{|K|}=\left|\frac{ds}{d\alpha}\right|$

证明



边界的曲率

$$y = f(x)$$

$$tg\alpha = \frac{dy}{dx} = y'$$

$$K = \frac{d\alpha}{ds} = \frac{d\alpha}{d(tg\alpha)}\frac{d(tg\alpha)}{dx}\frac{dx}{ds}$$

$$\frac{d(tg\alpha)}{d\alpha} = 1 + tg^2\alpha = 1 + y'^2$$

$$\frac{d(tg\alpha)}{dx} = \frac{dy'}{dx} = y''$$

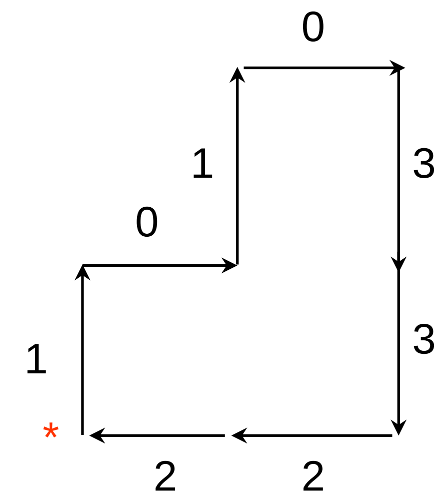
$$\frac{ds}{dx} = \frac{\sqrt{(dx)^2 + (dy)^2}}{dx} = \sqrt{1 + y'^2}$$

$$K = \frac{d\alpha}{ds} = \frac{d\alpha}{d(tg\alpha)}\frac{d(tg\alpha)}{dx}\frac{dx}{ds} = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}}$$

- 如果沿边界的逆时针方向走，则曲率还表征了边界的凹凸性
  - 如果曲率大于零，则边界是向外凸的；（向左弯）
  - 如果曲率小于零，则边界是向内凹的；（向右弯）
  - 如果曲率为零，边界段为直线。

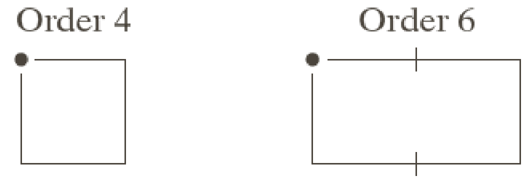
Shape Numbers

- 形状数：组成最小整数的差分链码（归一化）
  - 如图的4方向链码为10103322，差分码为33133030，形状数为03033133
  - 形状数是90度/45度旋转不变且与起点无关的一种边界描述
- 形状数的阶n：形状数序列的长度
  - 对于四链码，闭合曲线的形状数的阶总是偶数；
  - 对于凸形区域，四链码的阶与边界的外接矩形的周长对应

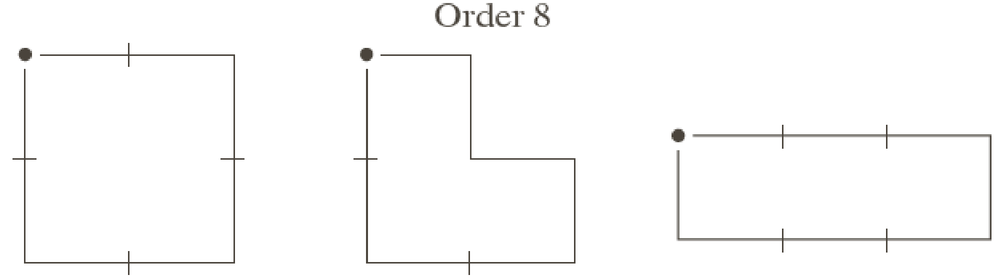


4方向链码示例

- All shapes of order 4, 6, and 8

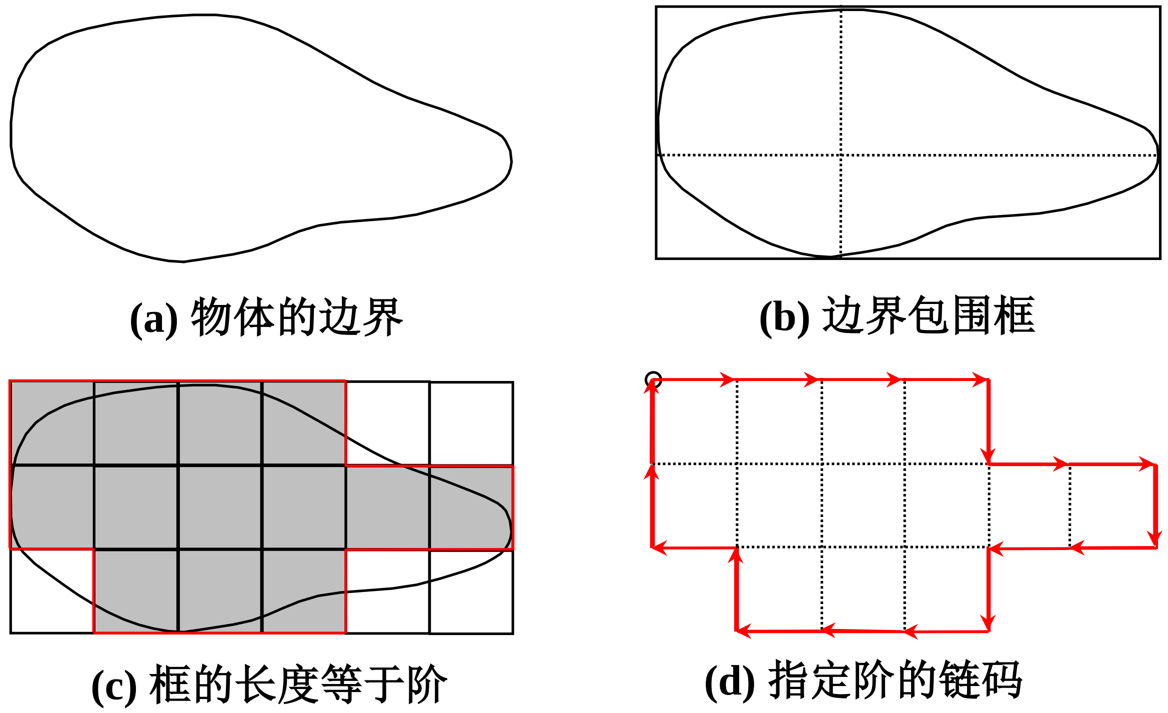


	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3



Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

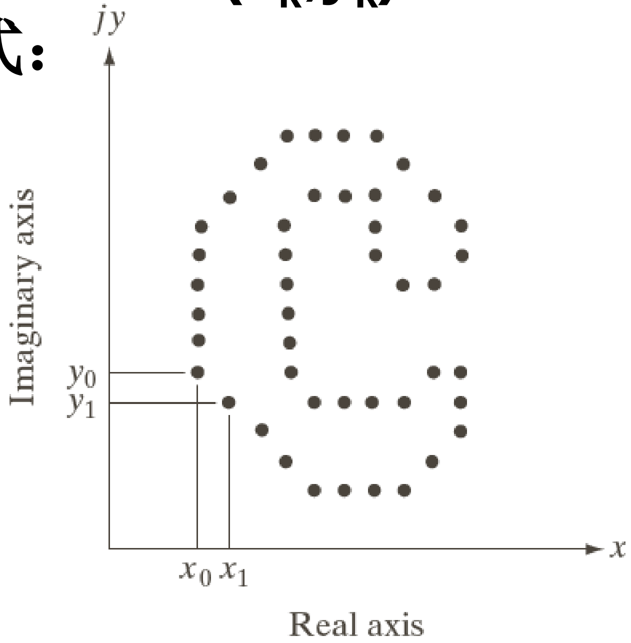
- 用网格近似原边界包围的区域，则区域的边界可得到：
  - 阶为18的链码：000030032232221211
  - 差分码为：300031033013003130
  - 形状数为：000310330130031303



找指定阶的形状数

Fourier Descriptors

- 设边界上有N个点，按逆时针方向，坐标依次为(x_k,y_k)，k=0,1,…,N-1，我们将其表示成复数形式：



$$s(k) = x_k + jy_k$$ **k=0,1,…,N-1**

– **s(k)**的离散傅立叶变换是

$$S(u) = \frac{1}{N}\sum_{k=0}^{N-1} s(k)\exp[-j2\pi uk/N]$$

u=0,1,…,N-1

– **S(u)**称为边界的傅立叶描述。反变换可得到边界的坐标

$$s(k) = \sum_{u=0}^{N-1} S(u)\exp[j2\pi uk/N]$$ **k=0,1,…,N-1**

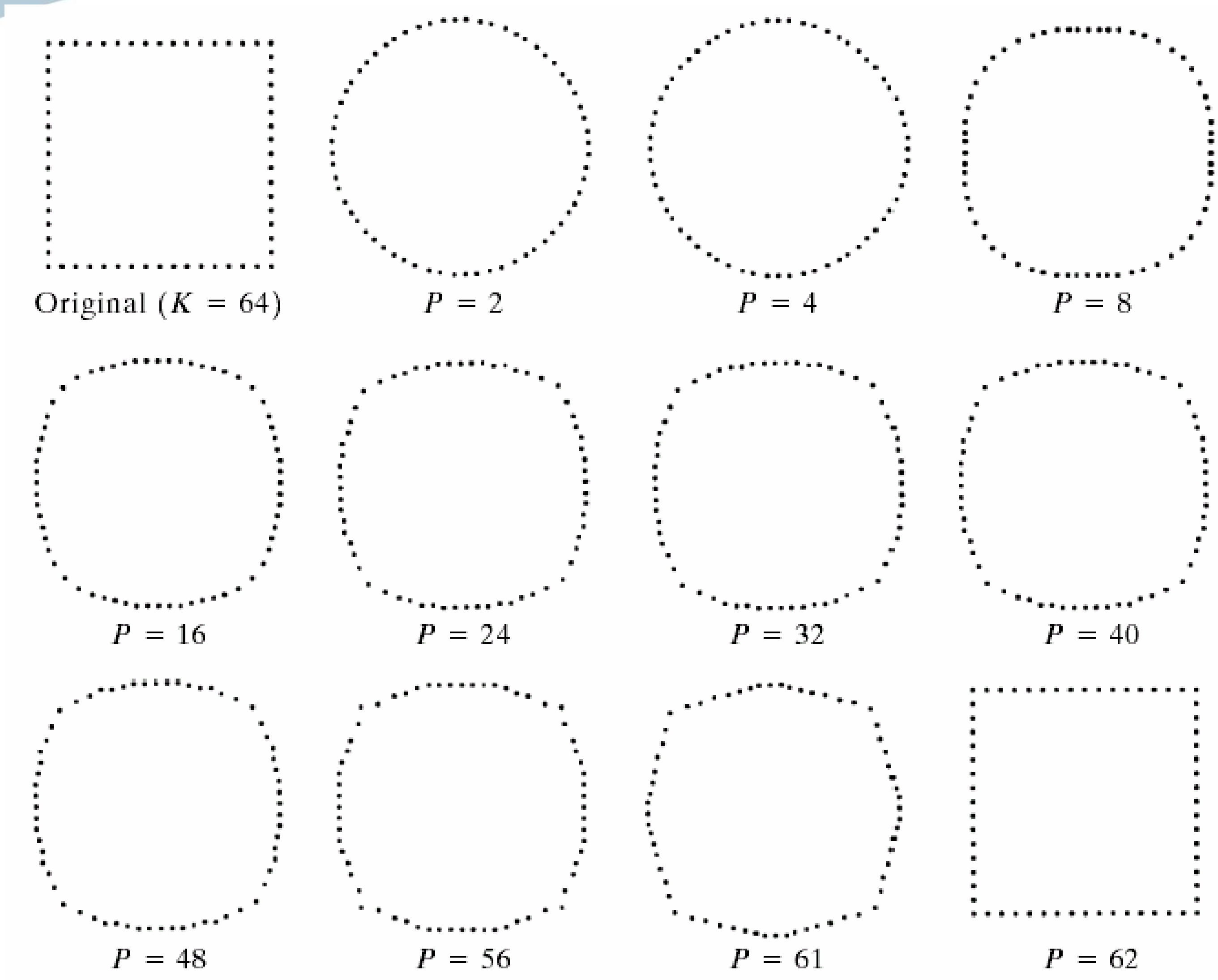
- 如果只利用傅立叶描述的前P个值，即舍去边界上的高频成分，则

$$s(k)=\sum_{u=0}^{P-1}S(u)\exp\left[j2\pi uk/N\right]$$ ，k=0,1,…,N-1

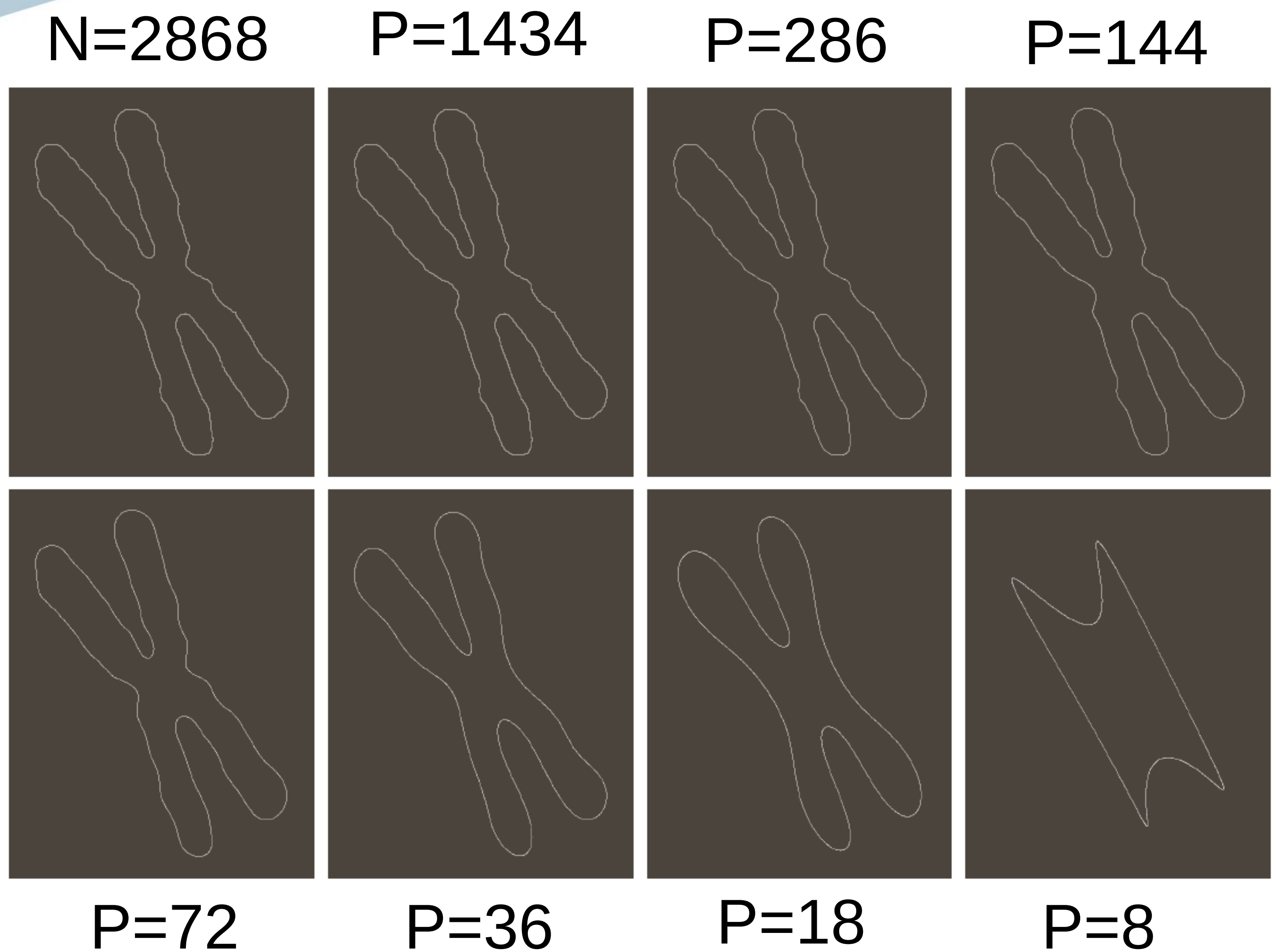
是对边界的近似

- Attention (DIP4E P837)

Because the transform is centered, we set to 0 half the number of coefficients on each end of the transform to preserve symmetry.



用付立叶描绘子重建边界的示例



用付立叶描绘子重建边界的示例

- 傅立叶描绘子在平移、旋转、尺度变换及起点不同时的影响

变　换	边　界	傅立叶描述
平移（x_0,y_0）	$s_t(k)=s(k)+x_0+jy_0$	$S_t(u)=S(u)+(x_0+jy_0)\cdot\delta(u)$
旋转（θ）	$s_r(k)=s(k)\exp(j\theta)$	$S_r(u)=S(u)\exp(j\theta)$
尺度（C）	$s_c(k)=C\cdot s(k)$	$S_c(u)=C\cdot S(u)$
起点（k_0）	$s_p(k)=s(k-k_0)$	$S_p(u)=S(u)\exp(-j2\pi k_0u/N)$

- 傅立叶描述的另一种形式是用曲线的曲率~边长函数进行傅立叶变换
  - 边界的方向

$$\phi(k) = tg^{-1}\left(\frac{y_k - y_{k-1}}{x_k - x_{k-1}}\right)$$

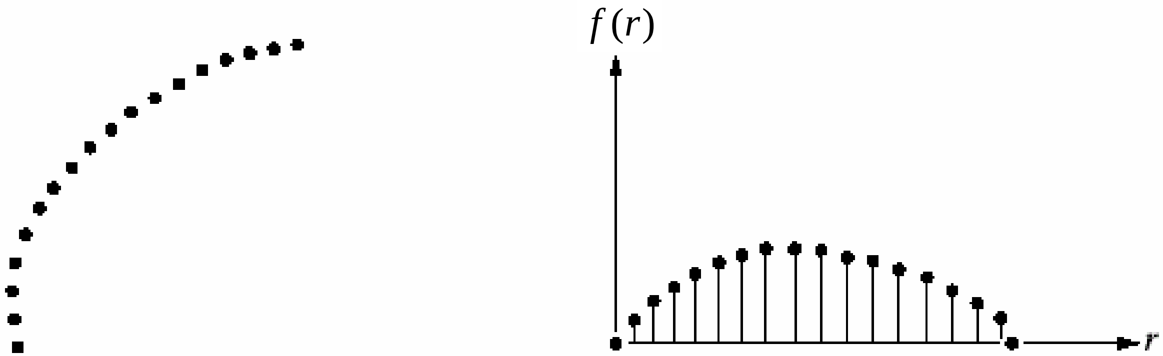
  - 曲率

$$K(k) = \phi(k) - \phi(k-1)$$

Statistical Moments

- 对任一段边界曲线，均可将其表示成一维函数$f(r)$的形式，其中**r**是任意可用的变量

  如：边界上的点到两端所成直线的距离可作为函数值



(a)边界线段　　**(b)**一维函数表示

边界线段的函数表示

- **设边界共有L点，如果将r当作随机变量，上图(b)看成是r的直方图，$f(r)$当作r的概率，则r的均值为**

$$m=\sum_{r=1}^{L} r_i f(r_i)$$

r的一阶矩

- **r的n阶矩为** $\mu_n=\sum_{r=1}^{L}(r_i-m)^n f(r_i)$

μ_n 与函数 $f(r)$ 的形状有直接关系，如，二阶矩描述了曲线相对于均值的分布，三阶矩描述了曲线相对于均值的对称性。

这种方法由于进行了边界的旋转，故有旋转不变性，也可通过 r 和 $f(r)$ 的缩放达到尺度归一化。

Assignments

- 11.2, 11.13, 11.15, 11.16

课后作业题目请对照参考第4版英文原版

- 第6次编程作业

从Laboratory Projects_DIP3E.pdf的Proj11-xx中选做1个题目。也可针对DIP4E Chapter 11内容，自拟任务。

浙江大学信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

Assignments

每个编程作业要求递交1份实验报告，命名“学号姓名_prjX.pdf”，内容提纲包括：

- 实验任务： 描述本次实验的任务，即所选择的ProjXX-xx题目，或自拟题目。
- 算法设计：理论上描述所设计的算法。
- 代码实现：描述编程环境，给出自己编写的核心代码。
- 实验结果：描述具体的实验过程，给出每个小实验的输入数据、算法参数和实验结果，并对结果做简要的讨论。
- 总结：简要总结本次实验的技术内容，以及心得体会。