



河南大學

明德新民 止于至善

图像描述与分析

陈小潘

计算机与信息工程学院

目录

- 基于边界的表达
- 基于区域的表达
- 基于边界的描述
- 基于区域的描述

图像表达与描述

- 图像分割实现了将图像中具有不同灰度特征、不同组织特征和不同结构特征的区域分离开的功能。即通过图像分割可得到图像中感兴趣的区域（目标）。
- 为进一步理解图像的内容，需要对这些区域、边界的属性和相互关系用更为简单的文字、数值、符号、数学公式、图等来进行定性或定量表达与描述，称为图像描述（Image Description）。
- 图像描述的目的是保留原图像或图像区域重要信息，减少数据量。

图像表达与描述

图像描述子：按一定的概念或公式从图像中产生文字、数值、符号或图，也称为**图像的特征**；产生它们的过程称为**图像特征提取**。

图像描述子应具有的特点：

- **唯一性**：每个目标必须有唯一的表示。
- **完整性**：描述是明确无歧义的。
- **几何变换不变性**：描述应具有平移、旋转、尺度等几何变换不变性。
- **敏感性**：描述结果应该具有对相似目标加以区别的能力。
- **抽象性**：从分割区域、边界中抽取反映目标特性的本质特征，不容易因噪声等原因而发生变化。

图像表达与描述

两个关键问题



目标表达

如何精确地测量这些
特征?



目标描述

选用什么特征来描述
目标?

图像表达与描述

图像中的区域（目标），可用其**内部**（如组成区域的像素集合）特征表示，
也可用其**外部**（如组成区域边界的像素集合）特征表示。

- 内部特征：所关注区域的性质如灰度、颜色、纹理等。
- 外部特征：所关注区域的形状等，例如几何特征。

图像表达与描述

- 表达是直接具体地表示目标。好的表达方法应具有节省存储空间、易于特征计算等优点。
 - ✓ 基于边界的表达、基于区域的表达。
- 描述是较抽象地表示目标。好的描述应在尽可能满足尺度、平移、旋转不变性、归一化等要求。
 - ✓ 基于边界的描述、基于区域的描述。
- 此外，边界和边界或区域和区域之间的关系也常需要进行描述，

图像表达与描述

- 表达和描述是密切联系的。表达的方法对描述很重要，因为它限定了描述的精确性；而通过对目标的描述，各种表达方法才有实际意义。
- 表达和描述又有区别。表达侧重于数据结构，而描述侧重于区域特性以及不同区域间的联系和差别；表达和描述抽象的程度不同，但其分别的界限是相对的。

目录

- 基于边界的表达
- 基于区域的表达
- 基于边界的描述
- 基于区域的描述

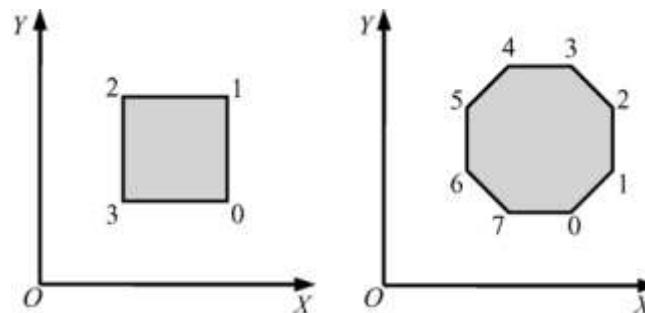
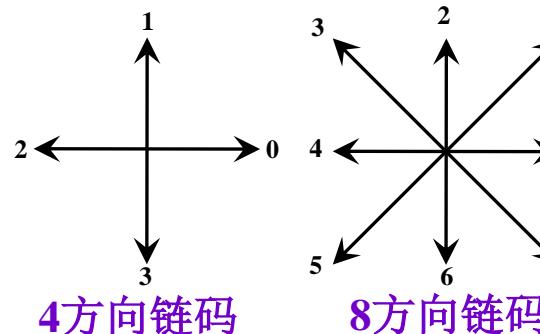
一、链码

- 封闭轮廓对应于目标（区域）的边界，因此边界是目标区域的一部分。
- 边界内的像素点属于该目标区域的点，与边界相邻但位于边界外的像素点不属于该目标区域的点。

一、链码

链码：利用一系列具有特定长度和方向的相连的直线段来表示目标的边界。

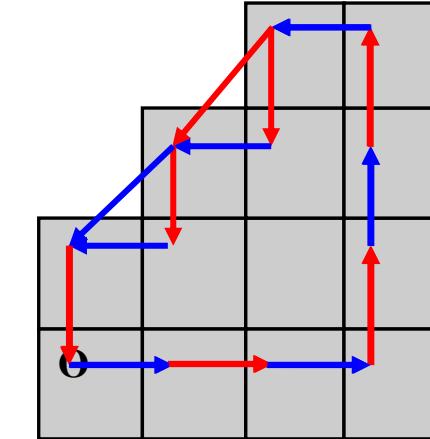
弗里曼（Freeman）链码：每个线段的长度固定而方向数量有限（4-连接或8-连接），所以只有边界的起点需要用（绝对）坐标表示，其余点都可只用接续方向来代表偏移量。



一、链码

以左下角点为起点，其坐标为(0,3)，4方向和8方向链码表示区域边界：

4方向链码：(0,3)0 0 0 1 1 1 2 3 2 3 2 3



8方向链码：(0,3)0 0 0 2 2 2 4 5 5 6

一、链码

■ 特点

- ✓ 由于表示一个方向数比表示一个坐标值所需**比特数少**，而且对每一个点又只需一个方向数就可以代替两个坐标值，因此链码表达可大大**减少**边界表示所需的数据量。
- ✓ 可以很方便地获取相关几何特征，如区域的周长等信息。
- ✓ 隐含了区域**边界的形状信息**。

一、链码

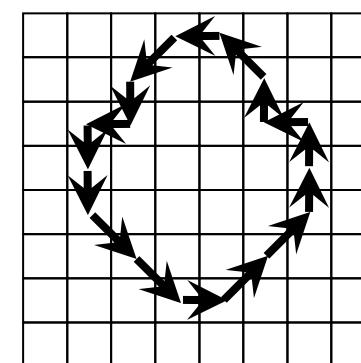
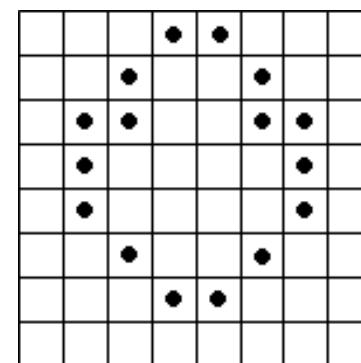
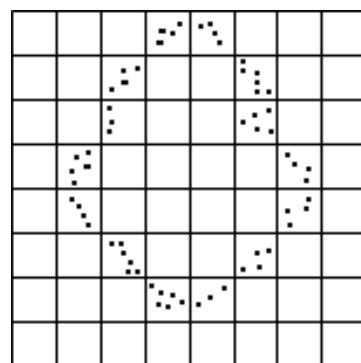
■ 不足

- ✓ 码串比较长；
- ✓ 噪声等干扰会导致小的边界变化，从而使链码发生与目标整体形状无关的较大变动；
- ✓ 目标平移时，链码不变，但目标旋转时，链码会发生变化。

一、链码

■ 多维网格重采样

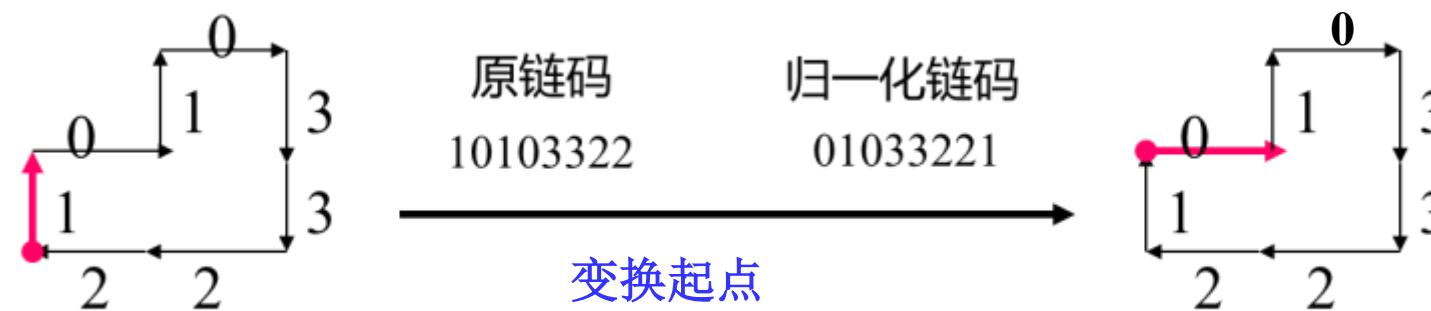
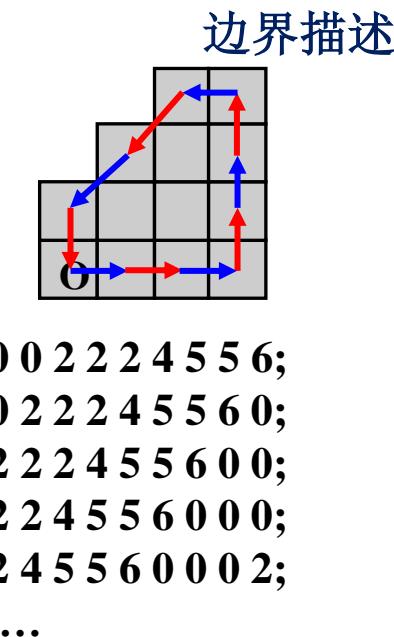
如果对原边界以较大的网格重新采样，并将与原边界点最接近的大网格点定为新的边界点，可以消除目标尺度变化对链码的影响。



一、链码

■ 链码归一化

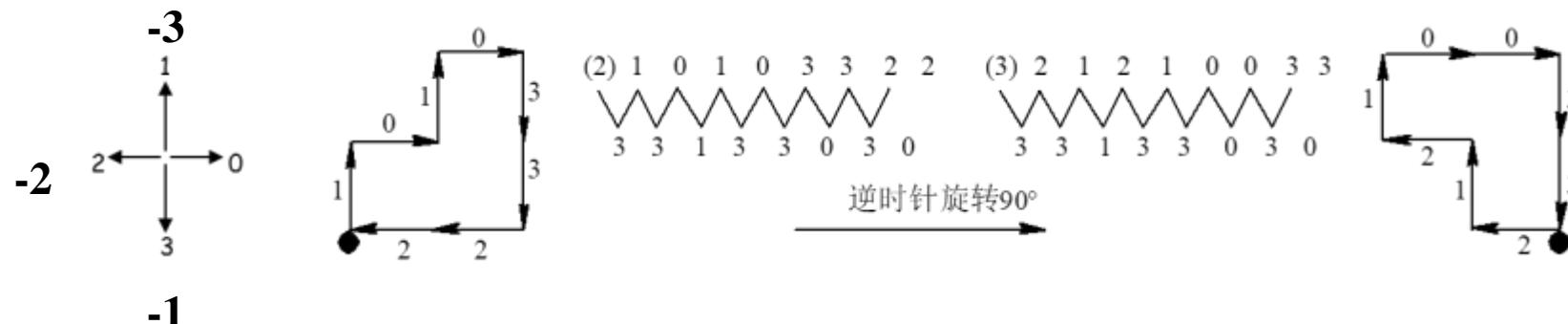
- ✓ 对同一个边界，如用不同的边界点作为链码的起点，得到的链码是不同的。
- ✓ 给定一个从任意点开始产生的链码，这些方向数**固定向一个方向循环**，以使它们所构成的**自然数的值最小**，将这样转换后所对应的链码起点作为这个边界的归一化链码的起点。这称为链码的**起点归一化**。



一、链码

用链码表示给定目标的边界时，如果目标平移，链码不会发生变化，而如果目标旋转则链码会发生变化。为解决这个问题，可利用链码的一阶差分将链码进行旋转归一化。

- 一阶差分可用相邻两个方向数相减（后一个减去前一个，如果得到负值按相反的方向数）得到差分码。
- 差分码不随轮廓旋转而变化。



一、链码



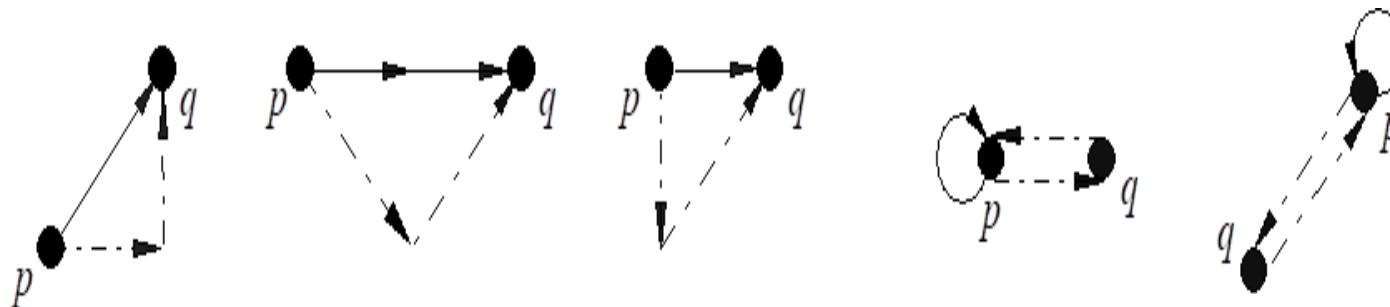
(3) 0 0 0 1 1 1 2 3 2 3 2 3
1 0 0 1 0 0 1 1 3 1 3 1

(0) 1 1 1 2 2 2 3 0 3 0 3 0
1 0 0 1 0 0 1 1 3 1 3 1

上述求一阶差分求解过程中原链码之前的（3）和（0）是分别指该链码最后一位的3和1。当原链码倒数第2位的2减去倒数第3位的3结果为-1。但是因为4方向链码的-1方向的链码为3，所以得出的一阶差分为3。

一、链码

■ **链码平滑**: 将原始的链码序列用较简单的序列代替。



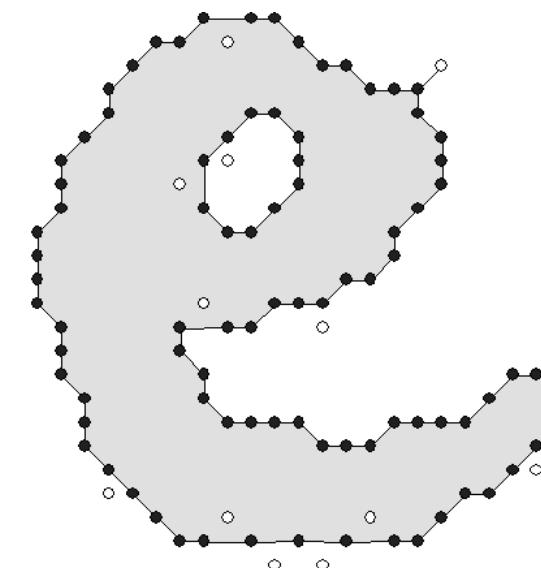
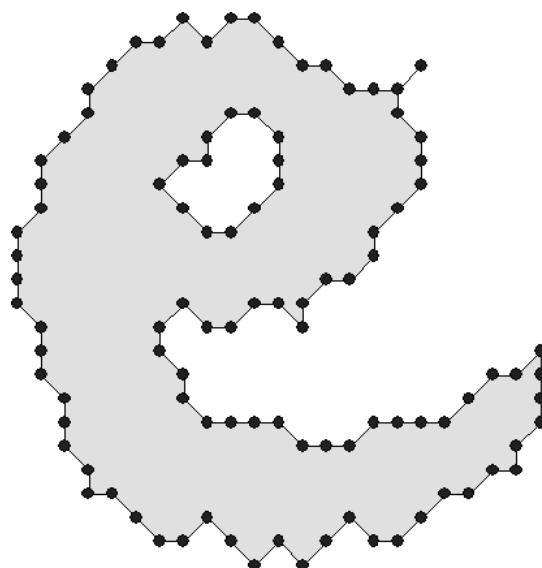
虚线箭头: 原始的在像素 p 和 q 之间的8连通链码。

实线箭头: 用来替换原始序列的新序列。

一、链码

■ 链码平滑示例

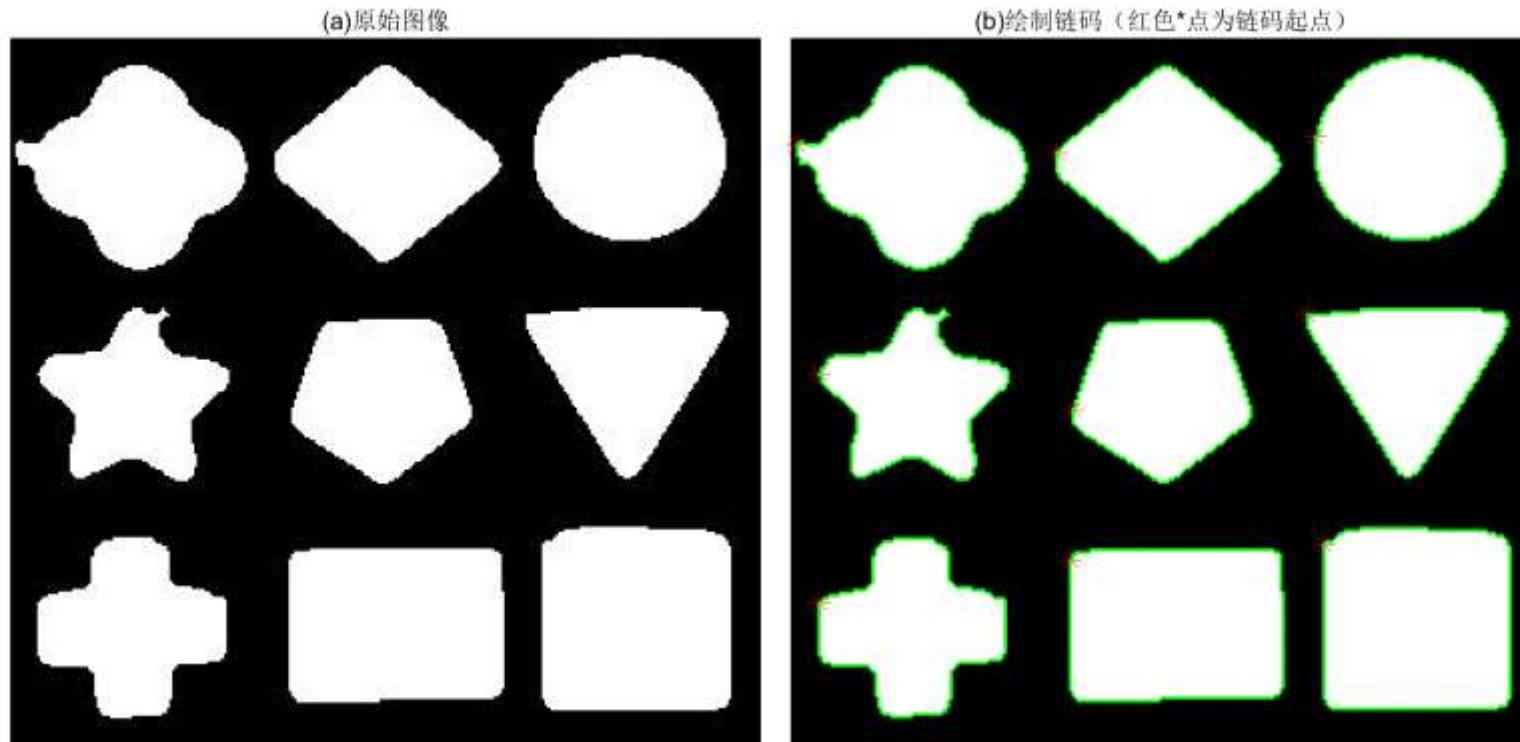
空心圆表示平滑后被除去的原轮廓点。



一、链码

【例】统计边界链码，并利用链码重构目标区域边界。

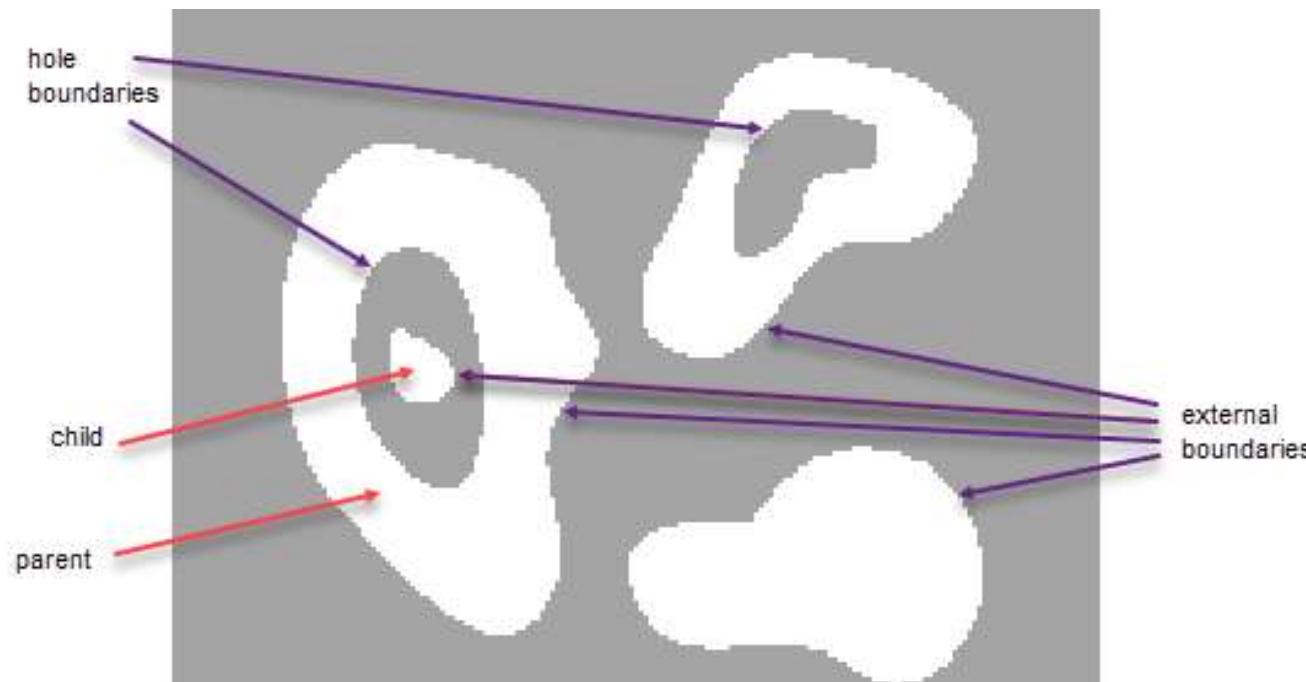
设计思路：二值化图像，统计每个区域的边界点，判断点和点之间位置关系，确定链码。根据链码判断边界上的点，实现目标边界重构。



一、链码

■ bwboundaries

✓ `B = bwboundaries(BW)`: 跟踪二值图像BW中对象的外边界以及这些对象内部孔洞的边界。`bwboundaries`还跟踪父对象完全包围的子对象的外边界和孔洞边界。该函数返回由边界像素位置组成的元胞数组B。



一、链码

■ bwboundaries

- ✓ `B = bwboundaries(BW,conn)` 指定跟踪对象边界时要使用的conn 连通性。
- ✓ `B = bwboundaries(__,options)` 跟踪对象的外边界，并通过将options设置为"holes"或"noholes"来指定是否包括孔洞的边界。
- ✓ `B = bwboundaries(__,Name=Value)` 使用名称-值参量(自R2023a起)指定跟踪样式和返回的顶点坐标的顺序。
- ✓ `[B,L] = bwboundaries(__)` 还返回标签矩阵L，该矩阵用于标记对象和孔洞。
- ✓ `[B,L,n,A] = bwboundaries(__)` 还返回n(找到的对象数量) 和A(邻接矩阵)。

一、链码

```
%统计边界链码
clc,clear,close all;
image=imread('shapefill.jpg');
subplot(1,2,1);imshow(image),title('(a)原始图像','FontSize',14);
BW=imbinarize(image);
[B,L]=bwboundaries(BW);
len=length(B);
chain=cell(len,1);           %存放各区域链码
startpoint=zeros(len,2); %存放各区域起点
for i=1:len
    boundary=B{i};
    everylen=length(boundary);
    startpoint(i,:)=boundary(1,:);      %记录区域起点
    for j=1:everylen-1
        candidate=[0 1;-1 1;-1 0;-1 -1;0 -1;1 -1;1 0;1 1]; %依8方向链码顺序存放邻点
        y=boundary(j+1,1)-boundary(j,1);
        x=boundary(j+1,2)-boundary(j,2);    %边界线上后一点和前一点的位置差
        [is,pos]=ismember([y x],candidate,'rows'); % 判断相邻关系是8方向中的第几个
        chain{i}(j)=pos-1;                  %给链码赋值
    end
end
```

一、链码

```
subplot(1,2,2);imshow(L),title('(b)绘制链码（红色*点为链码起点）','FontSize',14);
hold on %hold on主要是用于添加新绘图的时候保留当前绘图,
for i=1:len
    x=startpoint(i,2);  y=startpoint(i,1);
    plot(x,y,'r*','MarkerSize',12)      %绘制链码起点
    boundary=chain{i};                  %当前链码
    everylen=length(boundary);
    for j=1:everylen
        candidate=[y x+1;y-1 x+1;y-1 x;y-1 x-1;y x-1;y+1 x-1;y+1 x;y+1 x+1];%候选点
        next=candidate(boundary(j)+1,:);  %根据链码值判断下一个边界点
        x=next(2);y=next(1);
        plot(x,y,'g.');?>
    end
end
```

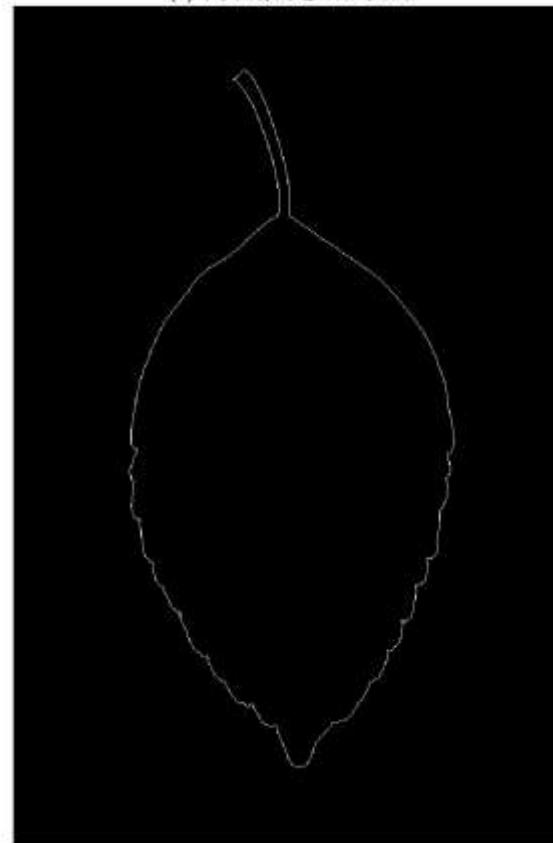
一、链码

【例】利用8方向链码跟踪目标边界Matlab程序。

(a)原叶子的二值图像



(b)叶子图像的边界跟踪结果

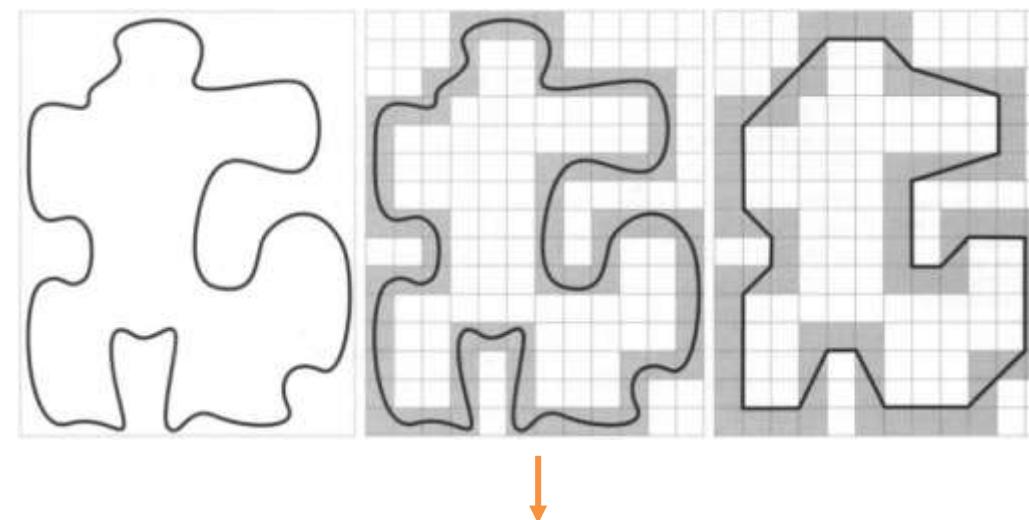


二、多边形拟合

- **多边形拟合**: 用多边形去近似逼近边界。
- 多边形是一系列线段的封闭集合，它可以用来逼近大多数曲线到任意的精度。
- 多边形拟合用**尽可能少**的线段来代表边界，并保持边界的基本形状，这样就可以用较少的数据和较简洁的形式来表达和描述。
- 常用的多边形表达方法:
 - ✓ 基于收缩（Shrink）的最小周长多边形法（MPP）
 - ✓ 基于聚合（Merge）的最小均方差线段逼近法
 - ✓ 基于分裂（Split）的最小均方差线段逼近法

二、多边形拟合

最小周长多边形法 (Minimum-Perimeter Polygons, MPP)：将边界看作是**有弹性的线**，组成边界的像素序列的内外边界是不可通过的墙壁，如果将线拉紧，则可得到最小周长多边形。最小周长多边形法是一种以周长最小的多边形来近似表示目标边界的方法。



(a) 目标边界和包围边界的单元格

(b) 图 (a) 的最小周长多边形

二、多边形拟合

【例】最小周长多边形法（MPP）

```
% 多边形拟合  
clc;clear  
f = imread('Fig1107(a).tif');  
subplot(2,3,1),imshow(f);title('(a)原始图像','FontSize',14);  
B = bwboundaries (f, 4, 'noholes');% 找到边界，4-连接，不考虑内部空洞  
b = B{1};  
[M, N] = size(f);  
bOriginal = bound2im(b, M, N);% 将边界转换为图像  
subplot(2,3,2),imshow(bOriginal);title('(b)原始边界','FontSize',14);  
% 原始边界图像  
[X, Y] = im2minperpoly(f, 2);#收缩橡皮带方法计算二值图像中单个区域的最小周界多边形  
% 最小周长多边形，包围边界单元 2×2  
b2 = connectpoly(X, Y);  
% 连接多边形顶点  
bCellsize2 = bound2im(b2, M, N);  
subplot(2,3,3),imshow(bCellsize2);title('(c)MPP(大小为2的方形单元)','FontSize',14);
```

二、多边形拟合

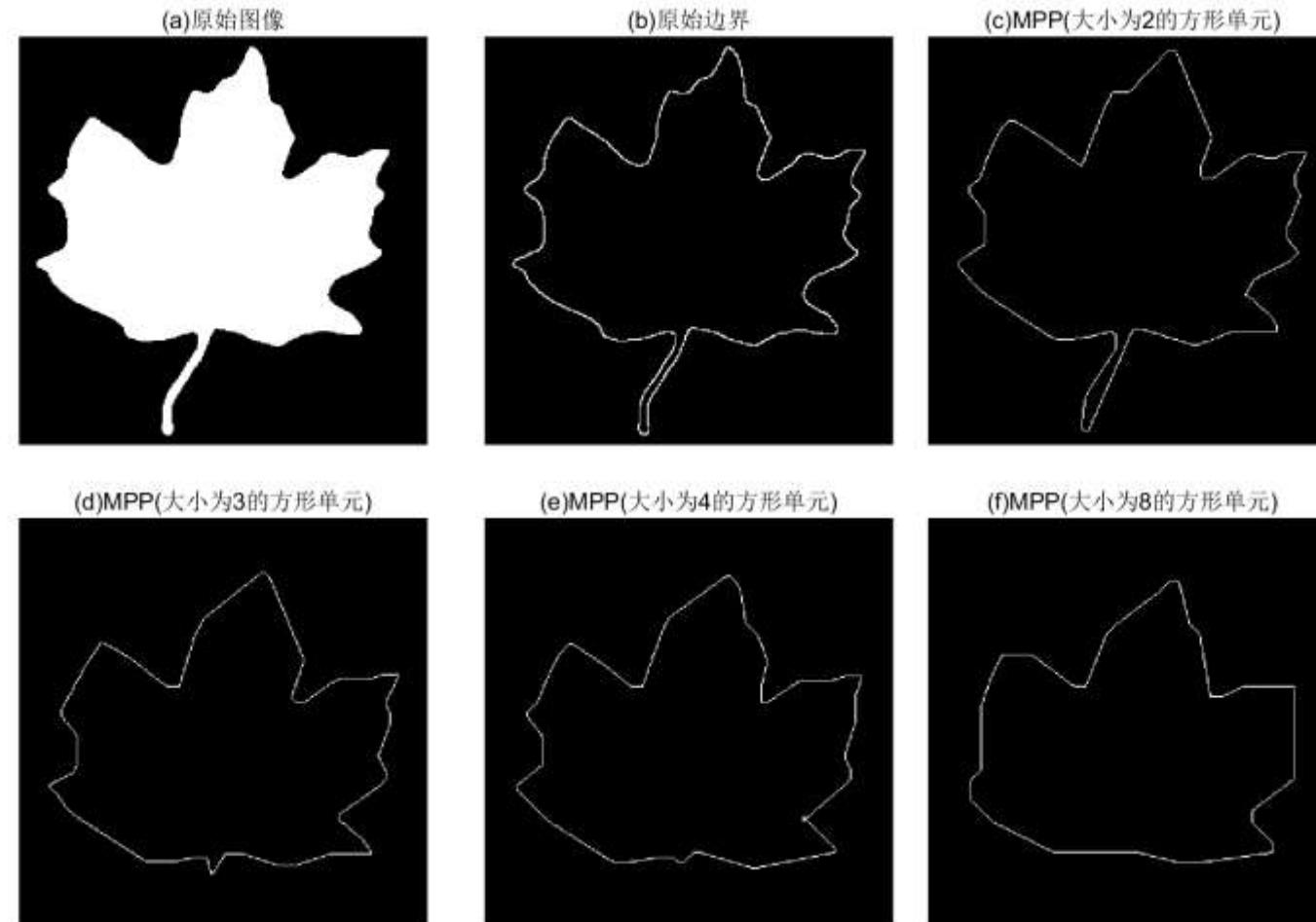
【例】最小周长多边形法（MPP）

```
% 最小周长多边形图像  
[X, Y] = im2minperpoly(f, 3);  
% 包围边界的方形单元大小 3×3  
b3 = connectpoly(X, Y);  
% 连接多边形顶点  
bCellsize3 = bound2im(b3, M, N);  
subplot(2,3,4),imshow(bCellsize3);title('(d)MPP(大小为3的方形  
单元)', 'FontSize', 14);  
[X, Y] = im2minperpoly(f, 4);  
% 包围边界的方形单元大小 4×4  
b4 = connectpoly(X, Y);  
% 连接多边形顶点  
bCellsize4 = bound2im(b4, M, N);  
subplot(2,3,5),imshow(bCellsize4);title('(e)MPP(大小为4的方形  
单元)', 'FontSize', 14);
```

```
[X, Y] = im2minperpoly(f, 8);  
% 包围边界的方形单元大小 8×8  
b8 = connectpoly(X, Y);  
% 连接多边形顶点  
bCellsize8 = bound2im(b8, M, N);  
subplot(2,3,6),imshow(bCellsize8);title('(f)MPP(大小为8的方形单  
元)', 'FontSize', 14);
```

二、多边形拟合

【例】最小周长多边形法（MPP）



二、多边形拟合

【例】最小周长多边形法（MPP）

说明：

- 使用大于 2×2 的像素单元导致了较低的分辨率，因此枫叶的细茎丢失了。
- 枫叶的第二个主要形状特征是其三个主要的裂片。即使使用了大小为8的单元，这些特征仍然可以保留。

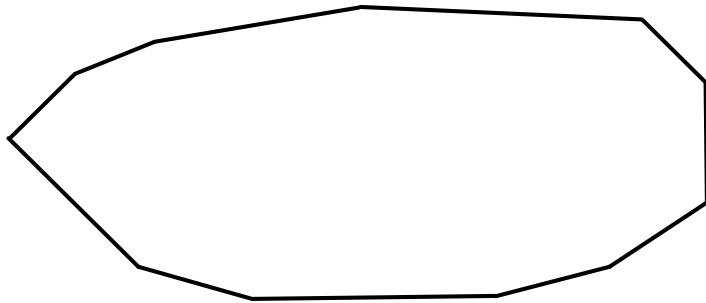
二、多边形拟合

最小均方差线段逼近法：基于聚合（Merge）的方法。

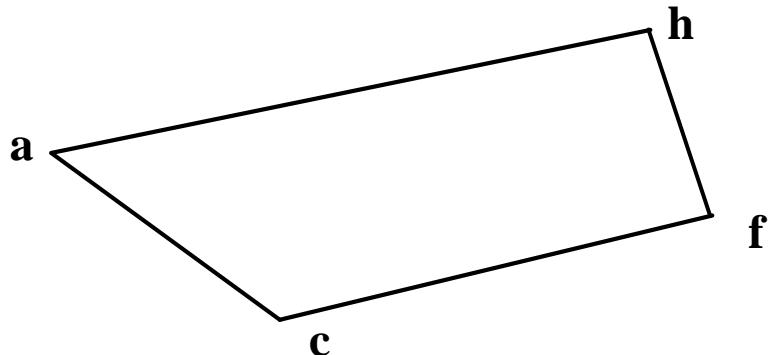
- 首先，选择边界上的任意一点作为直线段的起始端。
- 然后，顺次连接该点与其后的各点，并计算它们所构成的直线与对应边界的拟合误差，当某线段误差**大于**预先设定的阈值时，用该线段前的线段代替其所对应的边界，并将线段的另一端点设为起始点。
- 继续以上各步直到围绕边界一周为止，这样得到的就是与原边界满足一定拟合误差的多边形。

二、多边形拟合

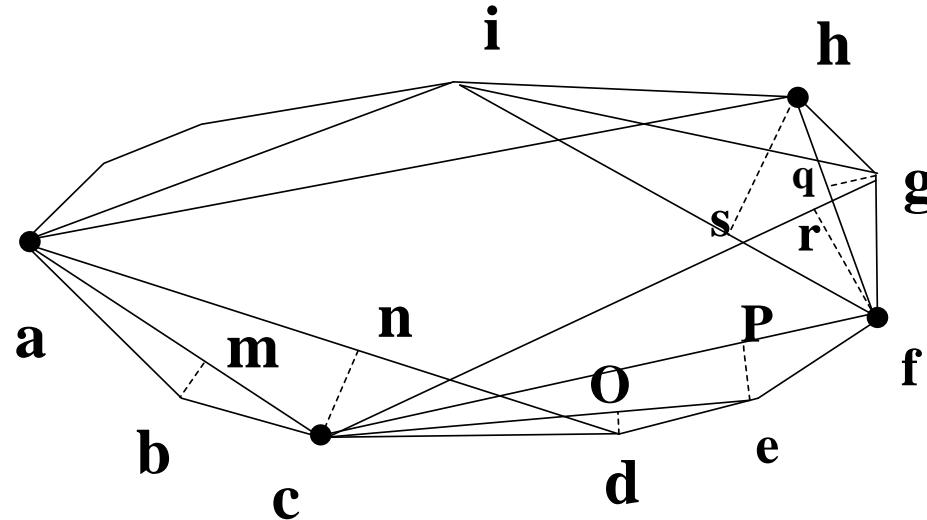
聚合技术一举例：



(a) 目标边界



(c) 表示目标边界的多边形



(b) 用聚合技术表示多边形

图8.7 基于聚合技术的多边形表示法

二、多边形拟合

上页给出了一个利用聚合技术获得多边形表示边界的示意图。其中图8.7(a)为目标边界；图12.7(b)为使用聚合技术进行多边形表示的过程，a为起始点，b、c、d为其后的3个点。为了简化起见，将各直线段到其前一边界点的距离作为误差，如图8.7(b)中所示的 bm 、 cn 。假设 bm 小于预设的阈值， cn 大于预设的阈值，则 c 为多边形的一个端点，以 c 点作为直线的一个端点继续以上的步骤。假设 do 和 ep 小于阈值， fr 大于阈值，则 f 点为多边形的一个端点，同理可得 h 也为多边形的一个端点，得到的多边形如图8.7(c)所示。通过比较图8.7(a)的边界和图8.7(c)的结果可以看出，由这种方法得到的多边形的顶点并不总与原边界的拐点相一致，对于这种情况还需要利用下面介绍的拆分技术加以缓解。

二、多边形拟合

最小均方差线段逼近法：基于分裂（Split）的方法

拆分技术是一种依据一定的准则通过不断拆分边界来得到多边形端点的方法。假设准则是：以边界点到连接边界上最远两点的直线的最大距离不超过一定的阈值。

方法：首先选择边界上**距离最远的两点**作为多边形的端点，并连接两端点得到一条直线；**然后**求边界上的点到该直线的**最大距离**，当距离大于预先设定的阈值时，该点即为多边形的一个顶点；**接着**对拆分后的边界线不断重复上述步骤，就可以确定原边界的多边形表示。

二、多边形拟合

最小均方差线段逼近法：基于分裂（Split）的方法

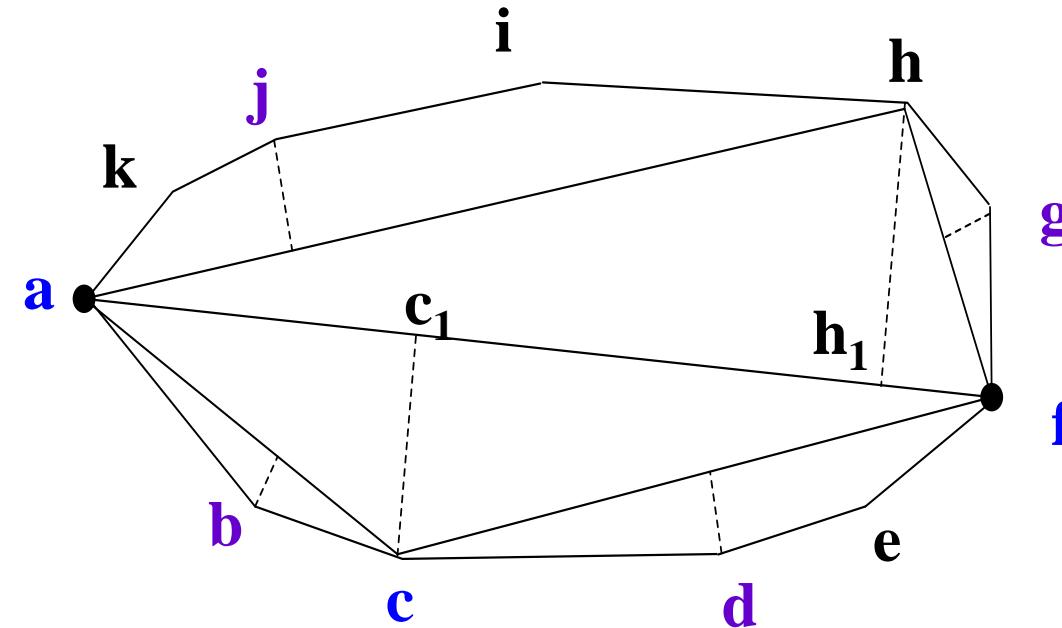


图12.8 基于拆分技术的多边形表示法

二、多边形拟合

最小均方差线段逼近法：基于分裂（Split）的方法

a点和f点是8.8(a)边界上距离最远的两点，分别求各点与af之间的距离，可以得到hh₁和cc₁是边界两侧到该直线最远的点。假设它们均大于预设的阈值。则h和c为确定的多边形上的两个端点；然后对下边界分别求边界段ac上的点到直线ac的最大距离和它与阈值之间的关系，以及边界段cf上的点到直线cf的最大距离和它与阈值之间的关系。可以看出，通过适当地选取阈值就可以较好地得到边界线上所标识的各拐点，这正是该方法与上述基于聚合技术的方法相比具有的优势所在。

三、标记图

- 标记 (Signature) 图：将二维的边界用一维的、较容易描述的函数形式来表达。
- 例如将质心到边界的距离画成角度的函数。
- 一维函数比二维边界更容易描述。

三、标记图

(1) 圆的标记方法

下图为一个二维的圆，如果将圆心看作极点，将圆半径看作极轴，向右的正方向直线看作是数轴；显然，当角度 θ 在 $0\sim 2\pi$ 之间变化时，圆心到圆周的距离始终等于圆的半径A。

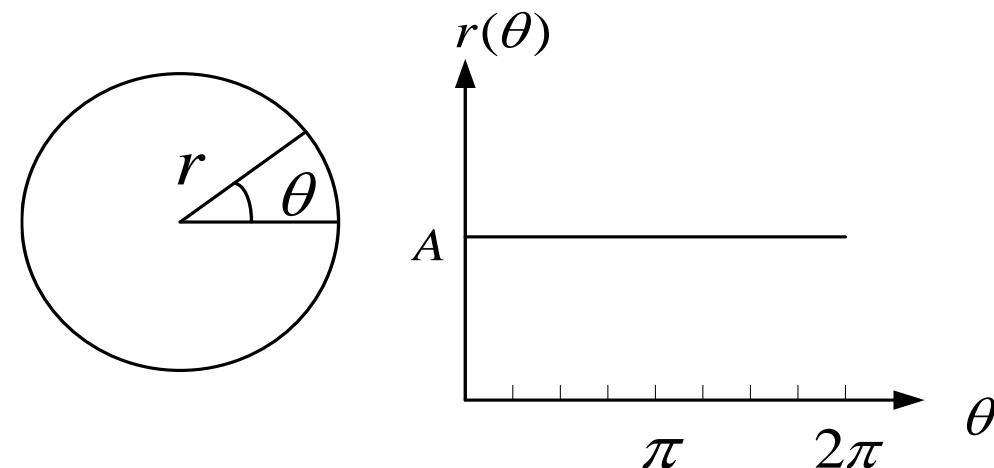


图 二维圆及其对应的一维标记图

三、标记图

(2) 正方形的标记方法

下图 (a) 是一个边长为 $2A$ 的正方形，将正方形的中心看作极点，将极点到正方形四周边的直线看作极轴，向右的正方向直线看作是数轴。当 $\theta=45^\circ$ 时 $r = A/(\sqrt{2}/2) = \sqrt{2} \cdot A$ 。当角度 θ 在 $0\sim 2\pi$ 之间变化时，正方形中心到正方形四周边的距离，正好是图 (b) 描述的自变量 θ 与其取值 $r(\theta)$ 的关系。

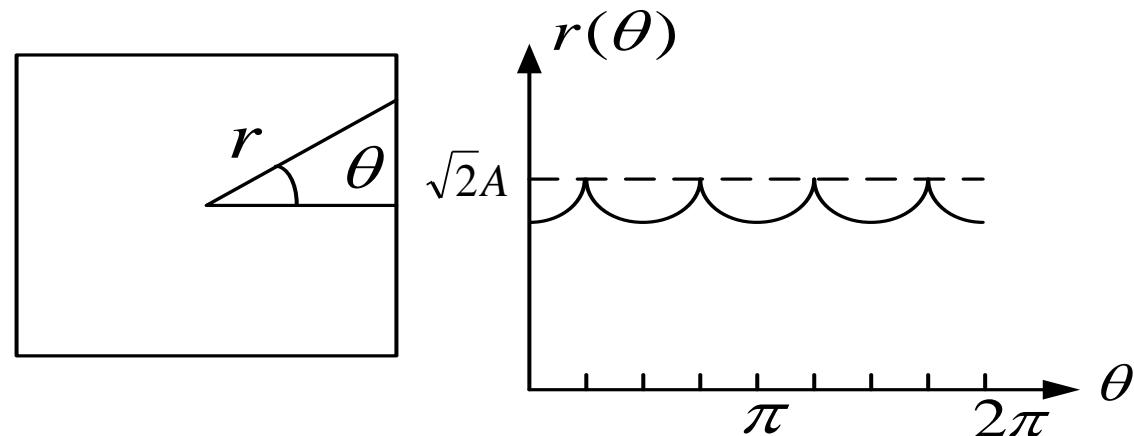


图 二维正方形及其对应的一维标记图

三、标记图

■ 好的标记应该能消除对尺寸的依赖。

%标记

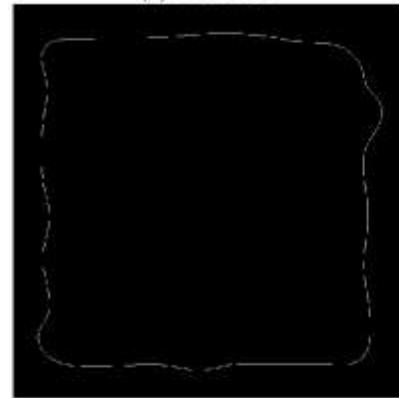
```
fsq = imread( 'Fig1111(a).tif');
subplot(2,2,1),imshow(fsq);title('(a)不规则方形','FontSize',14);
bSq = bwboundaries (fsq, 'noholes' );
% 找到边界，不考虑内部空洞
[distSq, angleSq] = signature(bSq{1});%用来寻找边界的标记。
% 标记，4个峰
subplot(2,2,2),plot(angleSq, distSq);title('(b)不规则方形对应的标记','FontSize',14);

ftr = imread( 'Fig1111(b).tif');
subplot(2,2,3),imshow(ftr);title('(c)不规则三角形','FontSize',14);
btr = bwboundaries (ftr, 'noholes' );
[disttr, angletr] = signature(btr{1});
% 标记，3个峰
subplot(2,2,4),plot(angletr, disttr);title('(d)不规则三角形对应的标记','FontSize',14);
```

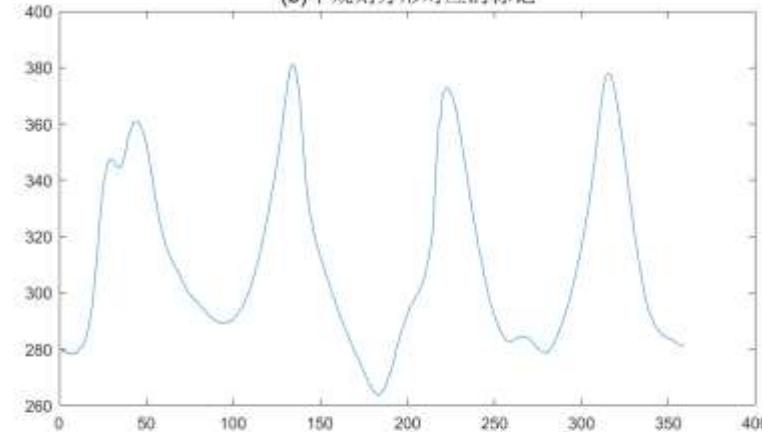
biaoji.m

三、标记图

(a)不规则方形



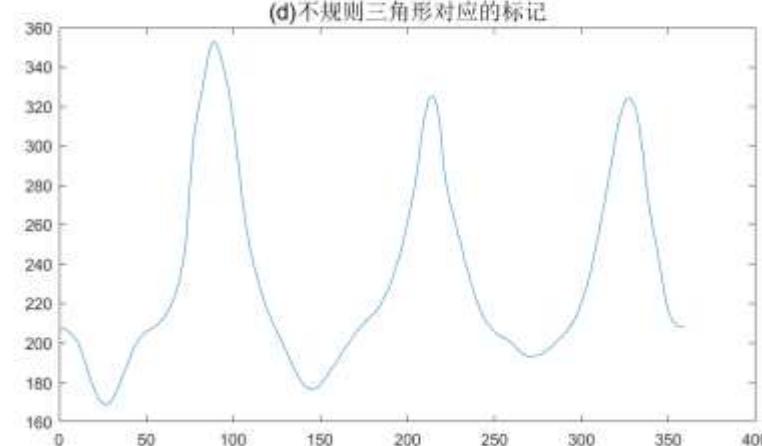
(b)不规则方形对应的标记



(c)不规则三角形



(d)不规则三角形对应的标记



三、标记图

- 与二维表示相比，上述的表示方法较为简单，而且对边界的平移变换也不敏感。
- 由于标记表示方法建立在角度旋转和尺度变换的基础上，因此，为了避免旋转和尺度变换对表示的影响，还需要对标记作相应的旋转和尺度归一化的操作。
- 尺度归一化可以比较简单地将标记的最大幅度值归一化到1来实现，比如圆的标记方法中可以将图中的A归一化为1来实现。

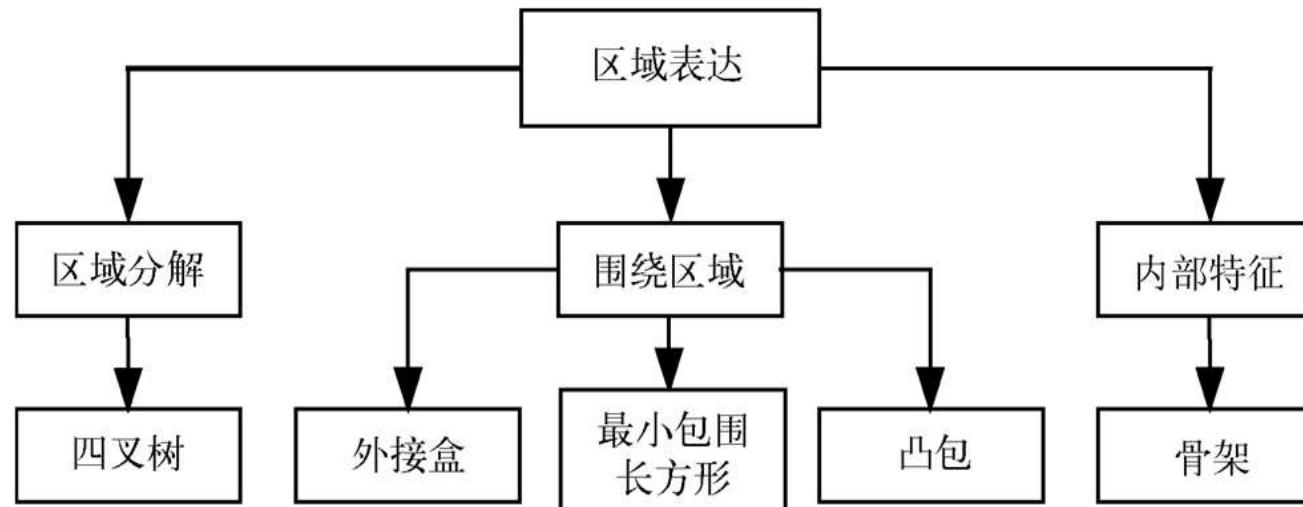
目录

- 基于边界的表达
- 基于区域的表达
- 基于边界的描述
- 基于区域的描述

基于区域的表达

基于区域的表达：

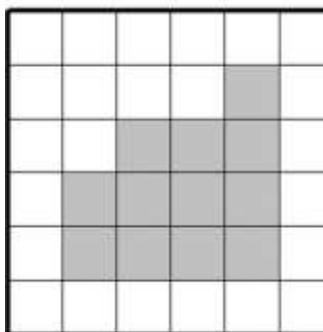
- 在图像处理中，区域是指符合特定特征的像素点集合。
- 区域分解：将目标区域分解为一些简单单元，如四叉树等。
- 围绕区域：用几何基元填充来表达，如外接圆等。
- 内部特征：由区域内部像素获得的集合，如骨架等。



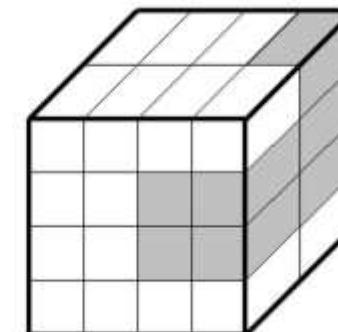
基于区域的表达

■ 空间占有数组

- ✓ 对图像 $f(x, y)$ 中任一点 (x, y) , 如果它位于给定的区域内, $f(x, y)$ 为1, 否则 $f(x, y)$ 为0。
- ✓ 所有 $f(x, y)$ 为1的点组成的集合代表了所要表示的区域。
- ✓ 方便、简单、直观



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



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

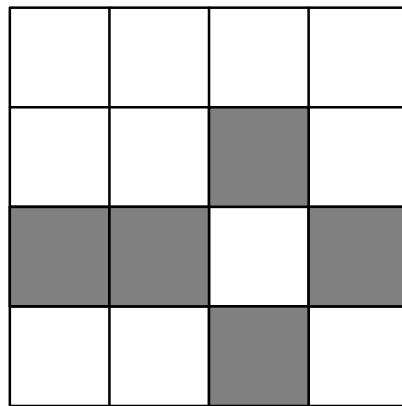
■ 四叉树表示

- ✓ 四叉树是一种有效的区域表示方式。
- ✓ 对于一幅大小为 $2^n \times 2^n$ 的图像，四叉树产生的过程如下：

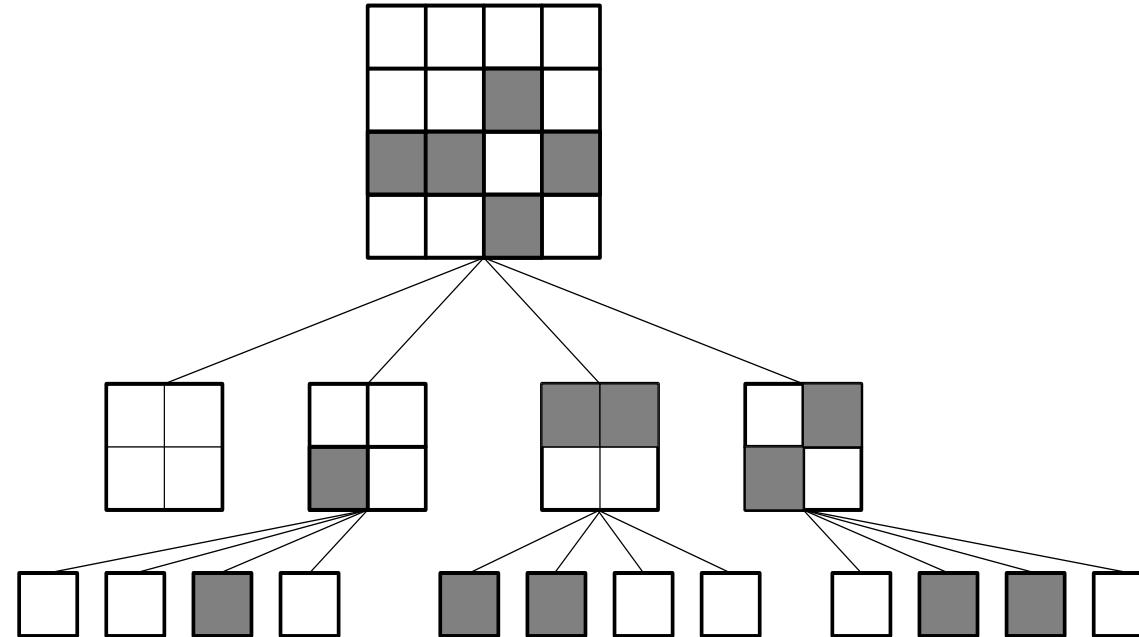
判断图像的一致性：当图像中存在目标区域时，图像中属于目标区域的像素点的值为1，不属于目标区域的像素点的值为0，可以认为图像不具有一致性，此时就将图像平均分成4个子区域，每个区域的大小为 $2^{n-1} \times 2^{n-1}$ ；对各个子区域重复进行一致性判断和分裂过程，直到所有的子区域具有相同的一致性。

基于区域的表达

■ 四叉树表示



(a) 图像及其目标区域



(b) 目标区域的四叉树表示

图 区域四叉树表示

■ 围绕区域

- ✓ 外接盒（Feret box）：包含目标区域的最小的长方形（朝向特定的参考方向）。
- ✓ 最小包围长方形（minimum enclosing rectangle, MER）：也称围盒。包含目标区域的（可朝向任何方向）最小长方形。
- ✓ 凸包（convex hull）：包含目标区域的最小凸多边形。**凸多边形**（Convex Polygon）指如果将一个多边形的所有边中，任意一条边向两方无限延长成为一直线时，其他各边都在此直线的同旁，那么这个多边形就叫做**凸多边形**。

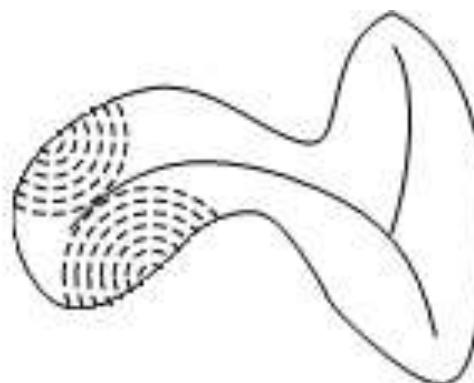


基于区域的表达

■ 骨架 (Skeleton)：可以理解为图像的中轴。骨架是一种细化结构，可以通过中轴变换来获得。图像骨架的提取也称为图像细化。

■ 每个骨架点都保持了其与边界点距离最小的性质。

火烧线法：假设在 $t=0$ 时刻，将目标边界上各点同时点燃，火焰以匀速向目标内部蔓延，当火焰前沿相交熄灭时，由熄灭点组成的集合就构成了该区域的中轴（也即骨架），火焰前沿交会处的这些熄灭点就是骨架点，如下图所示。

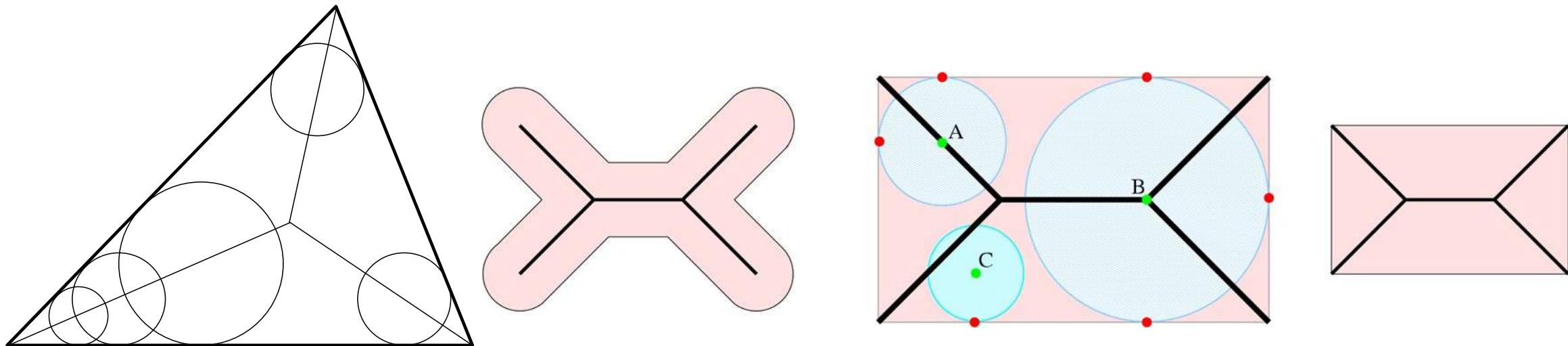


火焰前沿交会处形成的区域骨架

基于区域的表达

最大内接圆法：

最大内接圆盘至少有两点与目标边界轮廓相切。骨架的每个点都对应一个相应最大内接圆盘和半径。用最大圆的概念来描述图像的骨架如图（b）所示。图中三角形区域的骨架，由区域内所有最大内切圆的圆心组成。

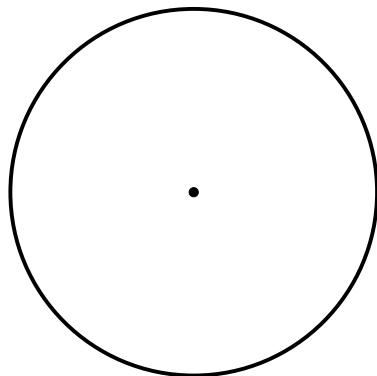


(b) 最大内切圆心组成的骨架

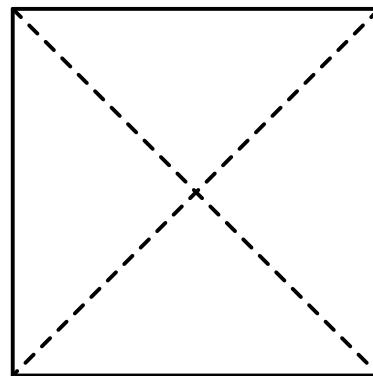
基于区域的表达

骨架的特点：

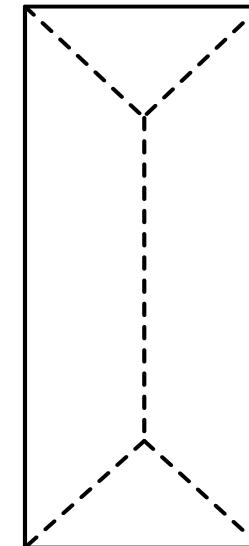
骨架上的任意一点至少与两个不同的边界上的点具有相同的最小距离。或者说每个骨架点都保持了其与边界点距离最小的性质。



(a) 圆形区域的骨架



(b) 方形区域的骨架



(c) 矩形区域的骨架

区域骨架的构建方法

- (1) 利用距离变换为区域中每个点赋以其到区域边界的最小距离值。
- (2) 利用一定的检测算法求出具有局部最大值的点，即为区域的骨架点。

基于区域的表达

Zhang-Suen算法是一种经典的细化算法，后续很多在其基础上进行改进，论文是1984年在IPCV(Image Processing and Computer Vision)发表的，A fast parallel algorithm for thinning digital patterns

基本原理

输入：二值图

输出：细化后图像

思路：删除掉非骨架上的非零像素点

其思路比较简单，重点在于如何判断非零像素点是不是骨架点。

基于区域的表达

Zhang-Suen细化算法通常是一个迭代算法，整个迭代过程分为两步：

Step One: 循环所有非零像素点，对符合如下条件的像素点标记为删除：

- $2 \leq N(p_1) \leq 6$

8邻域中，中心像素 P_1 周围的目标像素（二值中的1）的个数之和在2和6之间。

- $S(p_1) = 1$

按顺时针方向，即从P2 ~ P9 ~ P2像素中出现0~1的累计次数，其中0表示背景，1表示前景。

- $p_2 \cdot p_4 \cdot p_6 = 0$

- $p_4 \cdot p_6 \cdot p_8 = 0$

基于区域的表达

Zhang-Suen细化算法通常是一个迭代算法，整个迭代过程分为两步：

Step One: 循环所有非零像素点，对符合如下条件的像素点标记为删除：

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

1	1	1
1	p_1	1
0	0	0

1	1	1
1	p_1	1
0	0	0

其中 $N(p_1)=5$, $S(P_1)=1$, $P_2 \cdot P_4 \cdot P_6 = 0 \cdot 0 \cdot 0 = 0$, $P_4 \cdot P_6 \cdot P_8 = 0 \cdot 0 \cdot 1 = 0$, 符合条件, 需标记为删除。

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

0 0 1
1 p_1 0
1 0 1

其中 $N(p_1) = 4$, $S(P_1) = 3$, $P_2 \cdot P_4 \cdot P_6 = 0 \cdot 0 \cdot 0 = 0$,
 $P_4 \cdot P_6 \cdot P_8 = 0 \cdot 0 \cdot 1 = 0$, 不符合条件, 无需标记为删除。

基于区域的表达

Zhang-Suen细化算法通常是一个迭代算法，整个迭代过程分为两步：

Step Two: 跟Step One很类似，条件1、2完全一致，只是条件3、4稍微不同，满足如下条件的像素P1则标记为删除，条件如下：

1. $2 \leq N(p_1) \leq 6$
2. $S(P_1) = 1$
3. $P_2 * P_4 * P_8 = 0$
4. $P_2 * P_6 * P_8 = 0$

循环上述两步骤，直到两步中都没有像素被标记为删除为止，输出的结果即为二值图像细化后的骨架。

基于区域的表达

```
% 染色体图像  
f = imread('Fig1113(a).tif');  
subplot(2,3,1),imshow(f);  
f=im2double(f);  
h = fspecial('gaussian', 25, 15); %  $25 \times 25$  高斯平滑，标准差 15  
g = imfilter(f, h, 'replicate');  
subplot(2,3,2),imshow(g);  
g = im2bw(g, 1.5*graythresh(g));  
subplot(2,3,3),imshow(g);  
% 用1.5倍自动阈值进行二值化（进一步抑制噪声）  
s = bwmorph(g, 'skel', Inf); % 提取骨骼，Inf：重复执行直到图像不再发生变化  
subplot(2,3,4),imshow(s);  
% 去除毛刺，重复 8 次  
s1 = bwmorph(s, 'spur', 8);  
subplot(2,3,5),imshow(s1);  
% 去除毛刺，重复 8 次  
s2 = bwmorph(s1, 'spur', 8);  
subplot(2,3,6),imshow(s2);
```

quyugujia.m

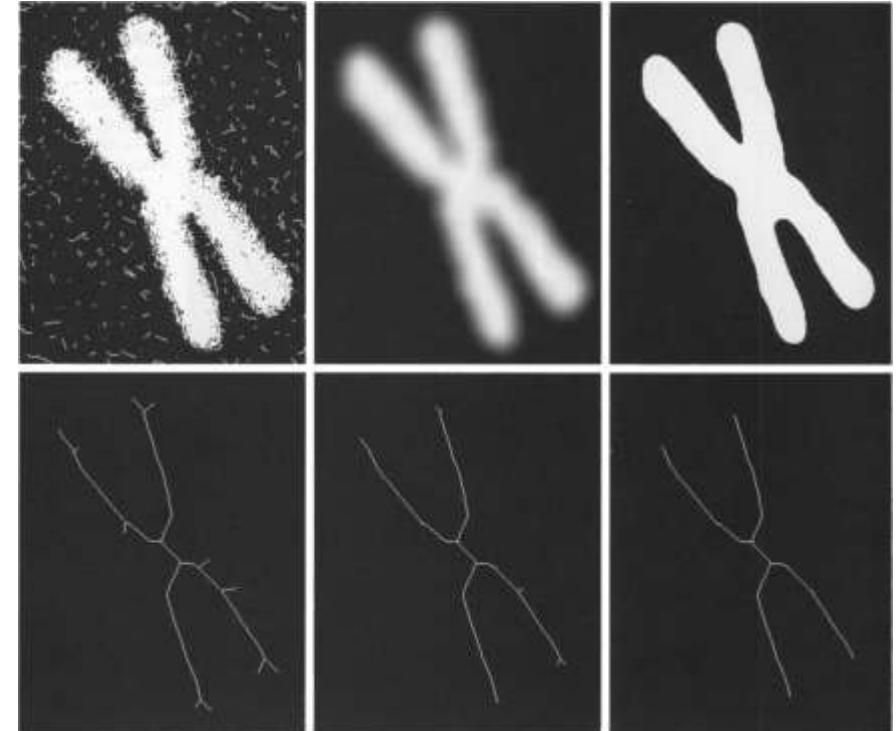


图 骨架

基于区域的表达

■ **fspecial()**

`h = fspecial(type)`

`h = fspecial(type, para)`

`fspecial`是用来生成滤波器（也叫算子）的函数，使用`type`参数来指定滤波器的种类，使用`para`来对具体的滤波器种类添加额外的参数信息。`h`就是生成的滤波器。

■ **imfilter()**

用法：`B = imfilter(A,H)` `B = imfilter(A,H,option1,option2,...)`

或写作`g = imfilter(f, w, filtering_mode, boundary_options, size_options)`其中，`f`为输入图像，`w`为滤波掩模，`g`为滤波后图像。`filtering_mode`用于指定在滤波过程中是使用“相关”还是“卷积”。`boundary_options`用于处理边界充零问题，边界的大小由滤波器的大小确定。

基于区域的表达

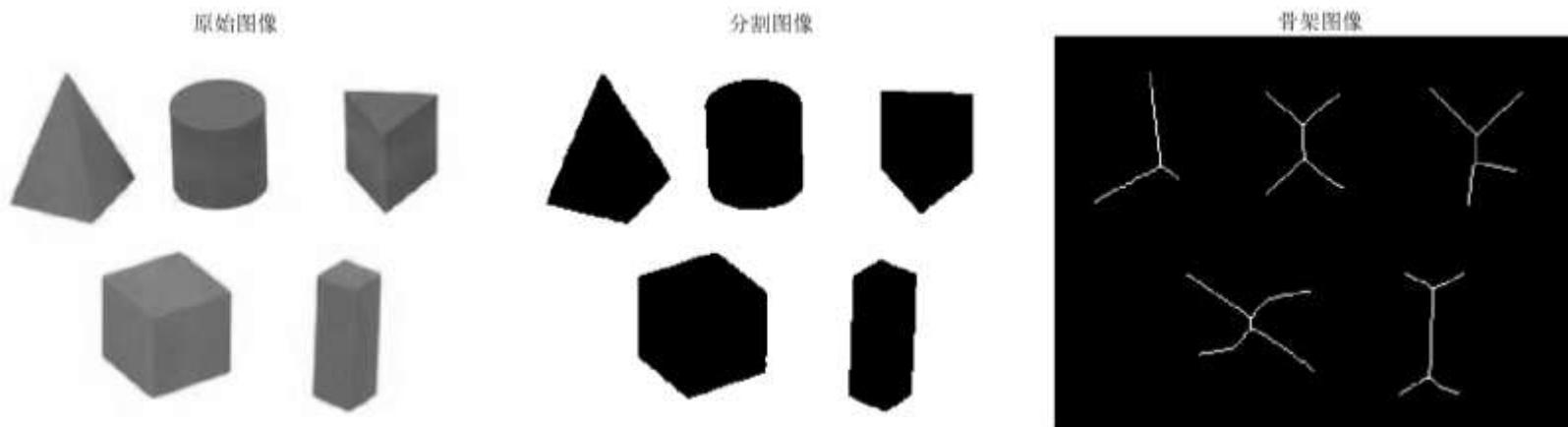
表 imfilter函数的相关参数

	选项	描述
filtering_mode	'corr'	通过使用相关来完成，该值为默认。
	'conv'	通过使用卷积来完成
boundary_options	'X'	输入图像的边界通过用值X（无引号）来填充扩展其默认值为0
	'replicate'	图像大小通过复制外边界的值来扩展
	'symmetric'	图像大小通过镜像反射其边界来扩展
	'circular'	图像大小通过将图像看成是一个二维周期函数的一个周期来扩展
size_options	'full'	输出图像的大小与被扩展图像的大小相同
	'same'	输出图像的大小与输入图像的大小相同。这可通过将滤波掩模的中心点的偏移限制到原图像中包含的点来实现，该值为默认值。

- bwmorph函数用于对二值图像进行形态学操作。它可以根据指定的操作类型（如thin、remove、spur等）对二值图像进行处理，以改善边缘的形状或去除不需要的部分。

基于区域的表达

```
%骨架  
clear,clc,close all;  
Image=rgb2gray(imread('blocks.jpg'));  
subplot(1,3,1),imshow(Image),title('原始图像');  
BW=imbinarize(Image);  
subplot(1,3,2),imshow(BW),title('分割图像');  
result=bwmorph(1-BW,'skel',Inf);  
subplot(1,3,3),imshow(result),title('骨架图像');  
imwrite(result,'tu11-8.jpg');
```



目录

- 基于边界的表达
- 基于区域的表达
- 基于边界的描述
- 基于区域的描述

边界描述子

■ **基于边界的描述：**利用处在目标区域边界上的像素集合来描述边界的特性。

1、边界长度

边界长度是包围区域轮廓的周长。简单情况下，可以用边界上的像素的个数来近似。

2、边界的直径

边界的直径为连接边界上两个距离最远点的线段的长度。边界的直径又称为边界的长轴。

对于某一具体的边界A，其直径定义为： $Diam(A) = \max_{i,j} [D(d_i, d_j)]$

其中， d_i 、 d_j 为边界A上的点， $D(d_i, d_j)$ 表示这两点之间的距离。

边界描述子

距离度量

■ 欧氏 (Euclidean) 距离

$$D_E(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$$

■ 城市街区 (city-block) 距离/曼哈顿距离：曼哈顿距离是标量空间中两点间各维度差的绝对值之和。在二维空间中，曼哈顿距离可以理解为从一个点到另一个点只能沿着水平或垂直方向行走的最小距离。

$$D_4(p, q) = |x - s| + |y - t|$$

边界描述子

距离度量

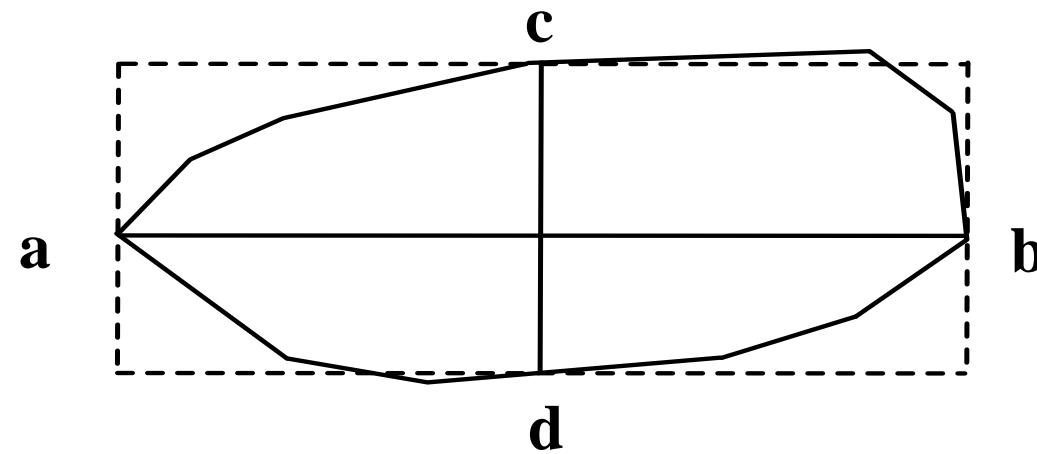
■ 棋盘 (chessboard) 距离、切比雪夫距离

棋盘距离是在直角坐标系中计算两点之间的距离的一种度量方式。它是通过在水平、垂直和对角线方向上的位移进行最大值计算得出的。棋盘距离的几何意义是两点间的路径以45度角的方式进行，类似于象棋棋盘上的走法。

取水平和垂直方向的最大差值： $D_8(p, q) = \max(|x - s|, |y - t|)$

边界描述子

- 与长轴垂直，并与边界相交的两点之间距离最长的线段称为边界的短轴。
- 由边界的长轴和短轴与边界的4个交点确定的矩形称为边界的基矩形（虚线表示的矩形）。
- 边界的长轴和短轴的比值称为边界线的离心率。



3、边界的曲率

曲率是斜率的变化率，用于描述边界上的各点沿边界方向变化的情况。

由于数字图像在边界上离散的像素点，因此仅依据边界上离散的像素点来获得该点的精确曲率是**不可能的**，通常利用相邻边界线段的斜率差来近似代替该点的曲率。

3、边界的曲率

利用曲率可以对边界斜率的变化情况作出判断：

- (1) 当沿着边界顺时针移动，且该边界点的曲率为负时，该点属于凹线段；
为非负时，该点属于凸线段。
- (2) 当曲率小于 10° 时，可近似判断该点属于直线段上的点；当曲率大于 90° 度时，该点应属于拐点。

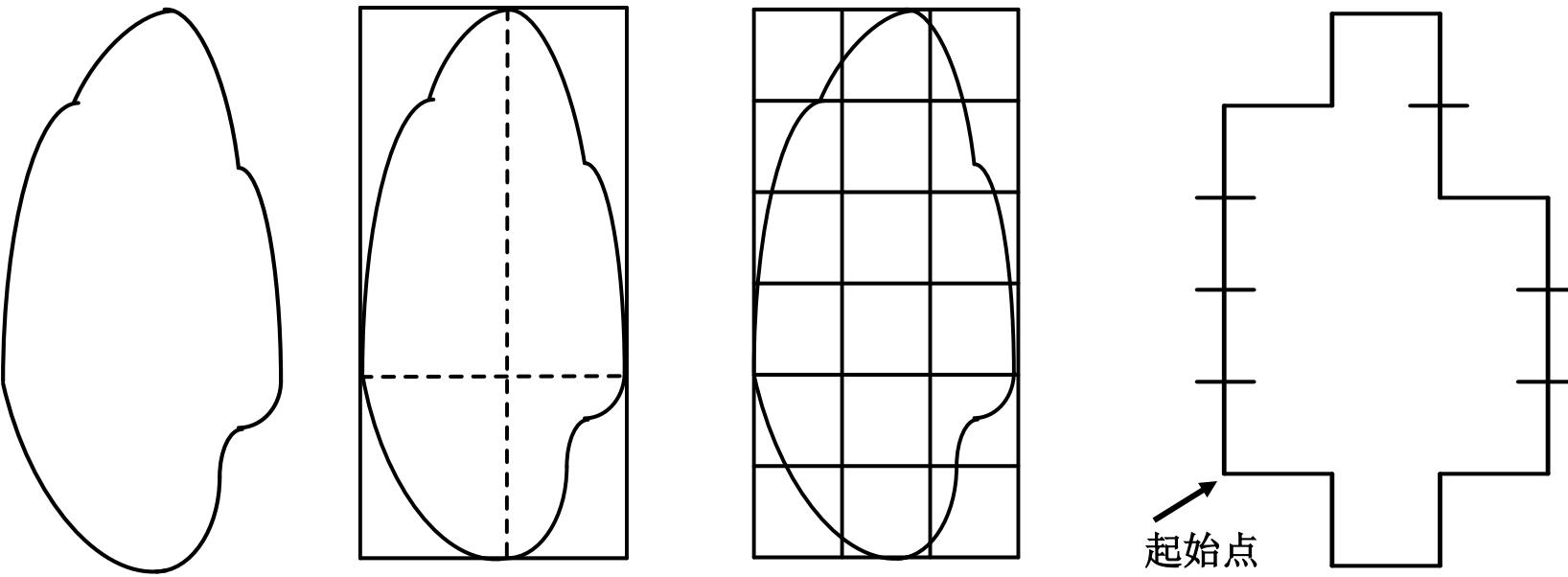
形状数

1、形状数的概念

形状数是一种基于链码反映边界形状的描述子。

形状数定义为具有最小值的原链码的一阶差分码，其值限定了可能的不同形状的数目。形状数是轮廓差分码中其值最小的一个序列。

形状数



(a) 目标的边界 (b) 边界的基本矩形 (c) 边界的方框数和网格 (d) 边界的近似多边形

链码: 1 1 1 1 0 1 0 3 3 0 3 3 3 2 3 2 1 2

一阶差分: 1 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1

形状数: 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1 1

图 边界形状数的生成过程

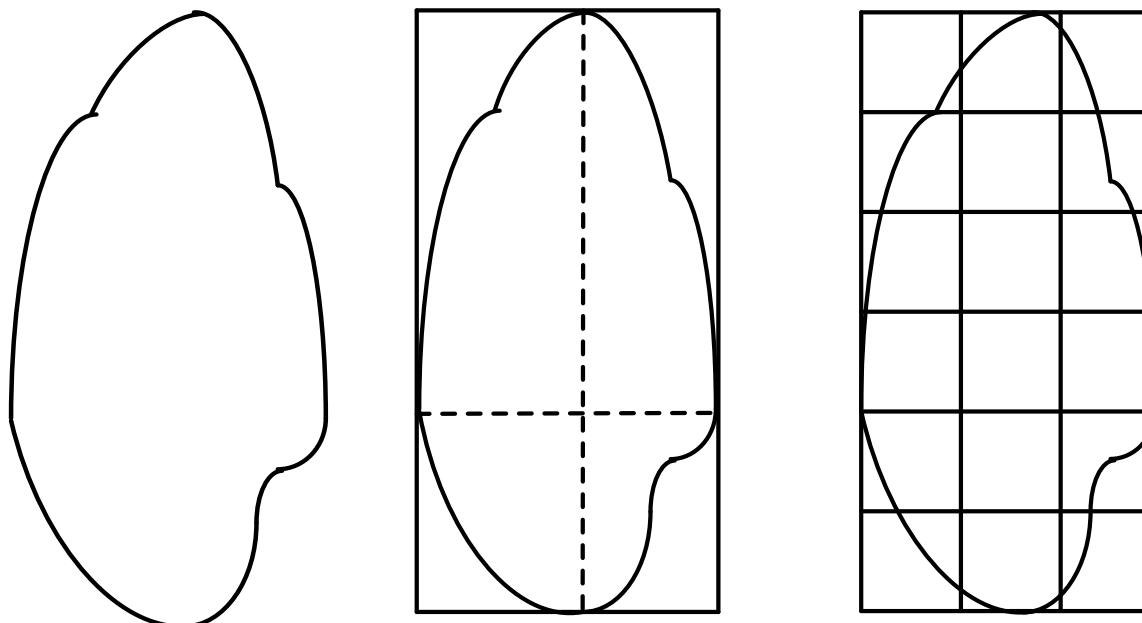
二、形状数

2、确定n阶形状数的一般过程

- (1) 确定阶数为n的方框，它的曲线离心率与边界的基本矩形最近似。
- (2) 依据给定的阶n，确定与之最接近的方框数和确定网格。从所有满足给定阶要求的矩形中选取其长短轴比例最接近的边界矩形。将选出的矩形划分为多个正方形。
- (3) 求出边界的近似多边形。
- (4) 求出多边形的链码和链码的一阶差分。
- (5) 循环差分码使其数串值最小，即该边界的形状数。

3、形状数的求解方法举例

- (1) 确定目标边界的基本矩形，如图 (b)。
- (2) 依据给定的阶 $n=18$ ，确定与之最接近的方框数及网格 ($3 \times 6 = 18$)，如图 (c)。



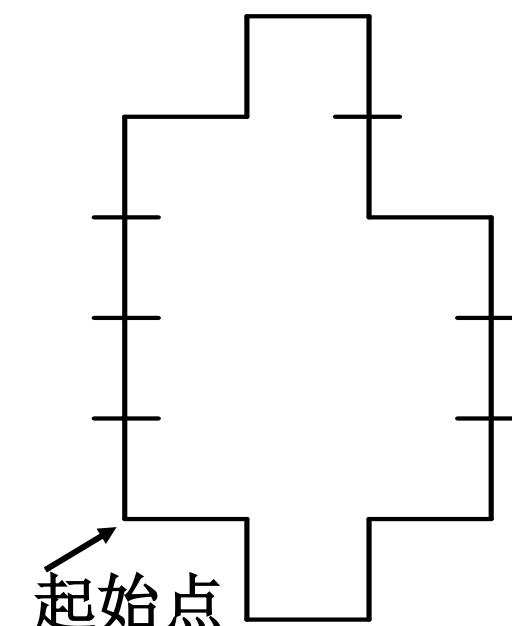
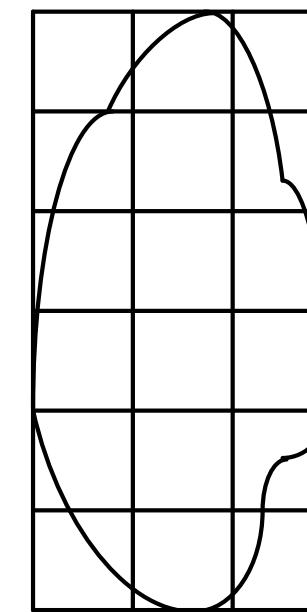
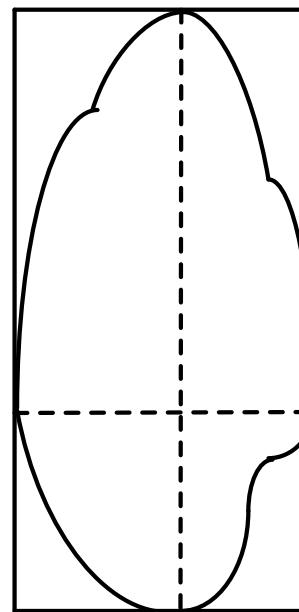
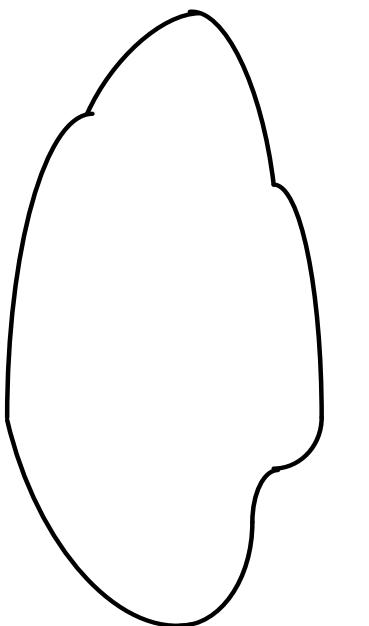
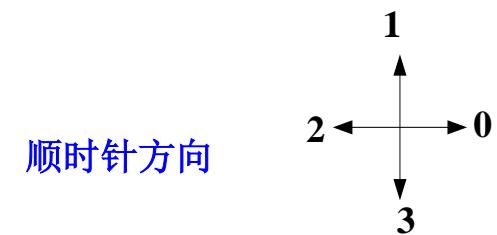
(a) 目标的边界 (b) 边界的基本矩形 (c) 边界的方框数和网格

3、形状数的求解方法举例

(3) 求出边界的近似多边形，如图 (d) 所示。

(4) 求多边形的链码。

链码： 1 1 1 1 0 1 0 3 3 0 3 3 3 2 3 2 1 2



(a) 目标的边界 (b) 边界的基本矩形 (c) 边界的方框数和网格 (d) 边界的近似多边形

3、形状数的求解方法举例

(5) 求链码的一阶差分。

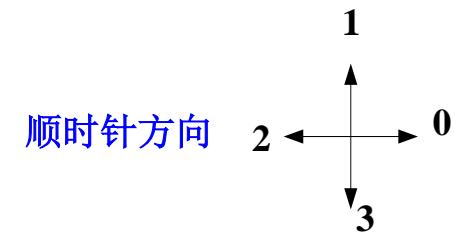
链码: 1 1 1 1 0 1 0 3 3 0 3 3 3 2 3 2 1 2

一阶差分: 1 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1

求解过程:

将链码看成循环序列，将相邻2个方向数按照逆时针方向相减，也即后一位减前一位（右边的位减左边的位），且差分结果的第一位是原链码最后一位和第一位相减的结果。

同时，结果为-1（方向）时，应为3；结果为-3（方向）时，应为1；结果为-2（方向）时，应为0。



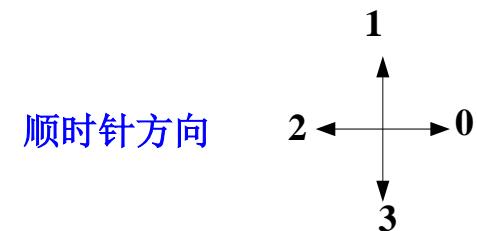
3、形状数的求解方法举例

(6) 求形状数。

链码: 1 1 1 1 0 1 0 3 3 0 3 3 3 2 3 2 1 2

一阶差分: 1 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1

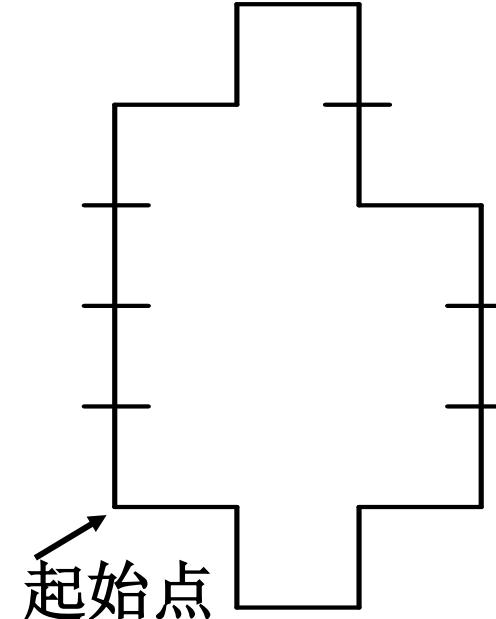
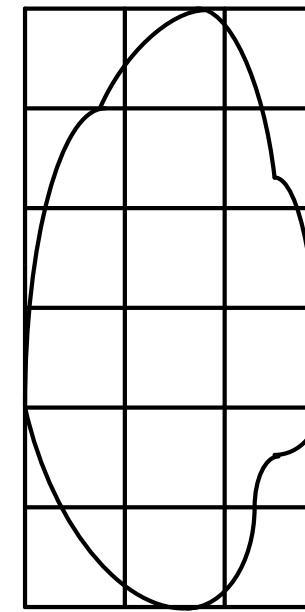
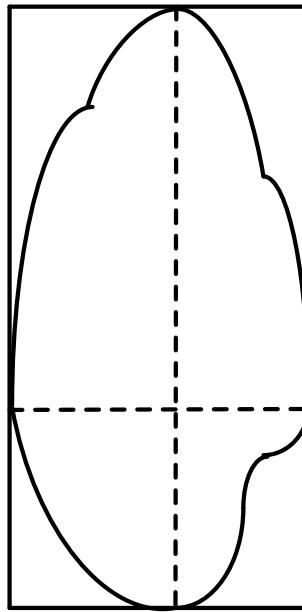
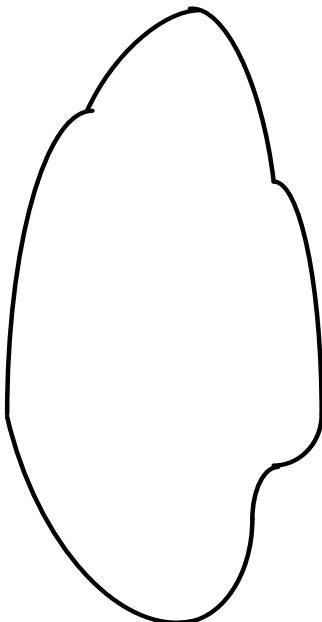
形状数: 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1 1



求解方法:

将一阶差分值进行向左或向右的循环移位，当其值最小时，即为所求的形状数。也即，将其中连续零最多的几位移到最左边时，该数的值就最小。

3、形状数的求解方法举例



(a) 目标的边界 (b)边界的基木矩形 (c)边界的方框数和网格 (d)边界的近似多边形

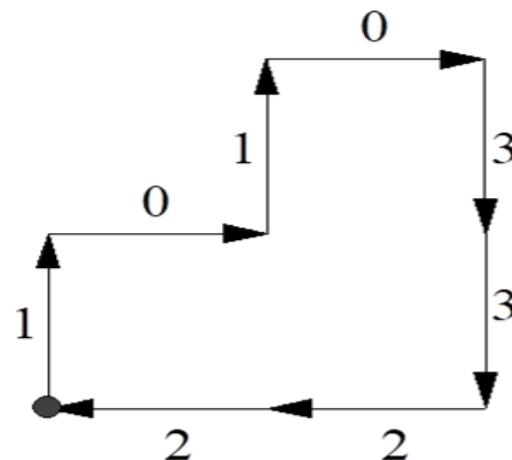
链码: 1 1 1 1 0 1 0 3 3 0 3 3 3 2 3 2 1 2

一阶差分: 1 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1

形状数: 0 0 0 3 1 3 3 0 1 3 0 0 3 1 3 3 1 1

3、形状数的求解方法举例

- 形状数的位数（包括前面的零）称为形状数的阶。
- 对于闭合边界来说，形状数的阶是偶数；对于凸多边形来说，形状数的阶对应于边界的基本矩形的周长。
- 形状数具有旋转、平移不变性。
 - ✓ 4-方向链码为： 10103322
 - ✓ 差分码为： 33133030
 - ✓ 形状数为： 03033133



傅立叶描述子

傅立叶描述子是一种通过对目标边界轮廓进行离散傅立叶变换来定量地描述图像中目标边界形状的图像特征。

傅里叶描述子的基本思想：

用物体边界信息的傅里叶变换作为形状特征，将轮廓特征从空间域变换到频域，并提取频域信息作为图像的特征向量。也即用一个向量代表一个轮廓，将轮廓数字化，从而能更好地区分不同的轮廓，进而达到识别物体的目的。

傅立叶描述子

假设目标的边界上有N个边界点，起始点为 (x_0, y_0) ，按照逆时针方向就可以将边界表示为一个坐标序列：

$$s(k) = [x(k), y(k)] \quad k = 0, 1, 2, \dots, N - 1$$

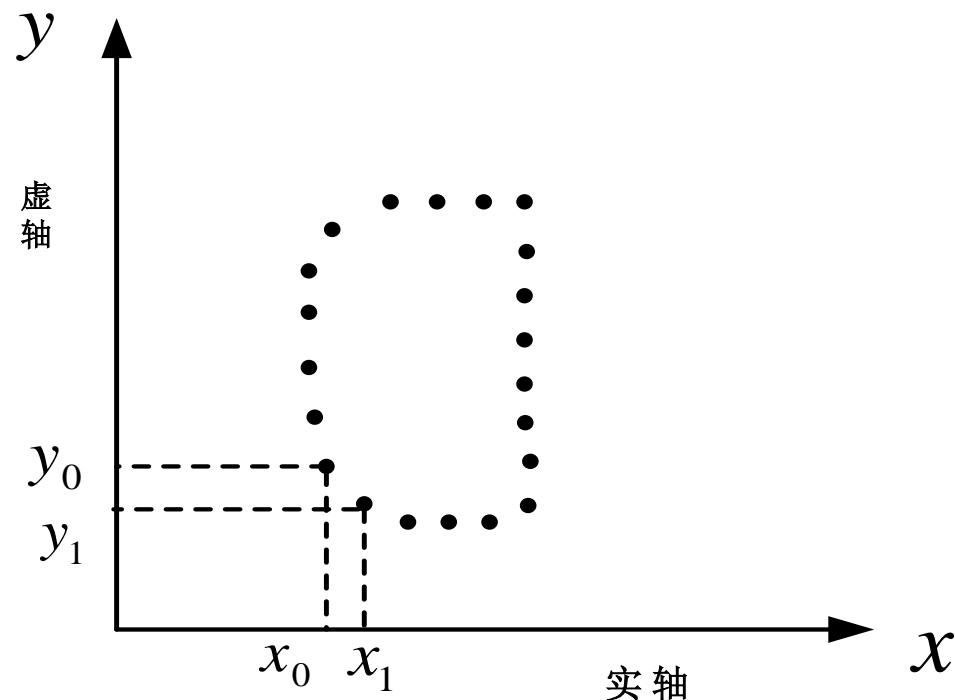
其中， $x(k) = x_k$, $y(k) = y_k$ 。也就是说，在确定了图像中目标边界的起始点和移动方向（顺时针或逆时针）后，就可以用**边界点的坐标对序列来描述边界**。

一般地，如果将目标边界看成是从某一点开始，沿边界逆时针方向旋转一周的周边长的一个复函数，也即将X-Y平面与复平面U-V重合，x轴与实部u轴重合(x坐标为复数的实部)，y轴与虚部v轴重合(y坐标为复数的虚部)。这时，边界点可以用复数表示为：

$$s(k) = x(k) + jy(k) \quad k = 0, 1, 2, \dots, N - 1$$

傅立叶描述子

下图给出了边界点的坐标与复数表示之间的对应关系。虽然通过这种重新定义，边界本身没有发生变化，但边界的表示从二维表达简化为一维表达。



傅立叶描述子

对序列进行傅立叶变换：

$$a(u) = \sum_{k=0}^{N-1} s(k) \exp[-j2\pi u k/N] \quad u = 0, 1, \dots, N-1$$

复系数 $a(u)$ 称为边界的 **傅立叶描述子**。

通过对傅立叶描述子进行傅立叶反变换，可以对边界重建得到边界的各点 $s(k)$ ，用前 N 个傅立叶变换系数近似：对复系数 $a(u)$ 做傅立叶反变换可以重建边界 $s(k)$ 。

$$s(k) = \frac{1}{N} \sum_{u=0}^{N-1} a(u) \exp[j2\pi u k/N] \quad u = 0, 1, \dots, N-1$$

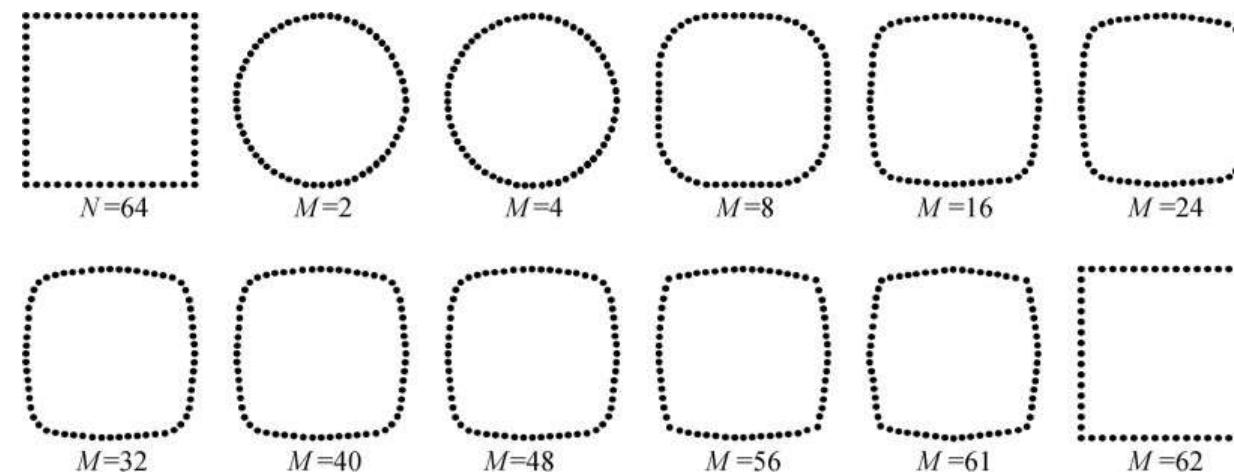
傅立叶描述子

- 低阶系数反映了边界的**大体形状**，随着系数阶数的不断增高，边界的细节特征逐渐变得明显，这与傅立叶变换中低频分量能较好地反映目标的整体形状和高频分量能较好的反映目标的细节特征是相一致的。
- 傅立叶描述子在描述边界时，具有对旋转、平移、尺度变化不敏感的特点。

傅立叶描述子

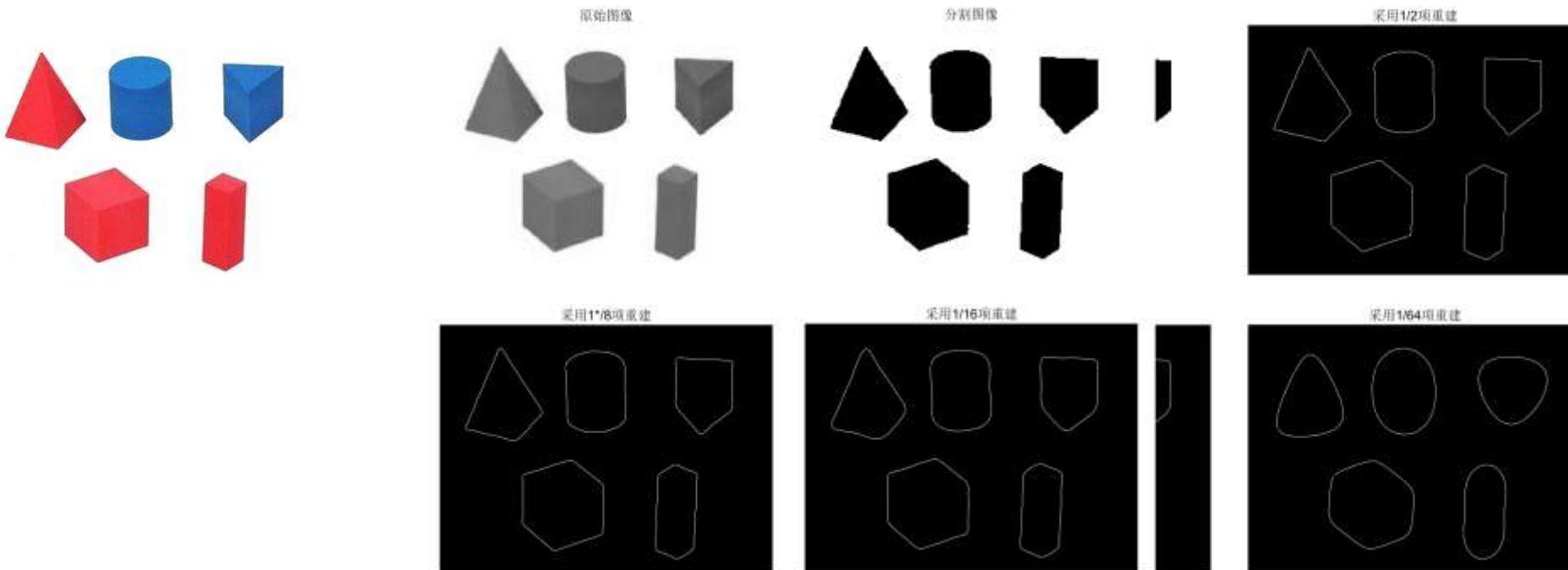
如果在傅立叶反变换时只利用傅里叶变换的前 M 个系数，可得到 $s(k)$ 的近似值（用较少的数据量表达边界的基本形状）。

以一个具有64个边界点的方形目标的边界图像为例，首先求出各边界点的描述子，然后利用前 M 个系数进行边界点的重建，图中分别列出了 M 等于2、4、8、16、24、32、40、48、56、61、62时边界重建的结果。



傅立叶描述子

- 实例：分割积木图像，计算各区域边界点的傅里叶描绘子并重建边界。



傅立叶描述子

■ 实例：分割积木图像，计算各区域边界点的傅里叶描绘子并重建边界。

```
clc,clear,close all;  
Image=rgb2gray(imread('blocks.jpg'));  
subplot(2,3,1),imshow(Image),title('原始图像');  
bw=imbinarize(Image);  
subplot(2,3,2),imshow(bw),title('二值图像');  
S=zeros(size(Image));  
[B,L]=bwboundaries(1-bw);           %二值图像反色，并搜索区域内外边界  
M=zeros(length(B),4);               %M存储重建时采用的点数  
for k=1:length(B)                  %length(B)为分割出的区域数  
    N=length(B{k});                 %N为第k个区域边界点数  
    if N/2~=round(N/2)              %点数非偶数  
        B{k}(end+1,:)=B{k}(end,:); %边界点增加1  
        N=N+1;  
    end  
    M(k,:)=[N/2 N*7/8 N*15/16 N*63/64]; %重建采用的点数为原点数的1/2,1/8,1/16,1/64  
    c={'1/2' '1*8' '1/16' '1/64'};  
end
```

傅立叶描述子

- 实例：分割积木图像，计算各区域边界点的傅里叶描绘子并重建边界。

```
for m=1:4 %四种重建情况
    subplot(2,3,2+m),imshow(S);title(strcat('采用',c(m),'项重建'))
    hold on;
    for k=1:length(B) %每个区域分别处理
        z=B{k}(:,2)+1i*B{k}(:,1); %构建复数点列
        Z=fft(z); %DFT变换
        [Y,I]=sort(abs(Z)); %按模的大小升序排列
        for count=1:M(k,m)
            Z(I(count))=0; %按给定的比例，将较小的项设为0
        end
        zz=ifft(Z); %IDFT
        plot(real(zz),imag(zz),'w'); %重绘边界
    end
end
```

傅立叶描述子

傅立叶描述子的基本性质

变换	边界	傅立叶描述子
原函数	$s(k)$	$a(u)$
旋转变换	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
平移变换	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
尺度变换	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
起点	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2k_0 u/K}$

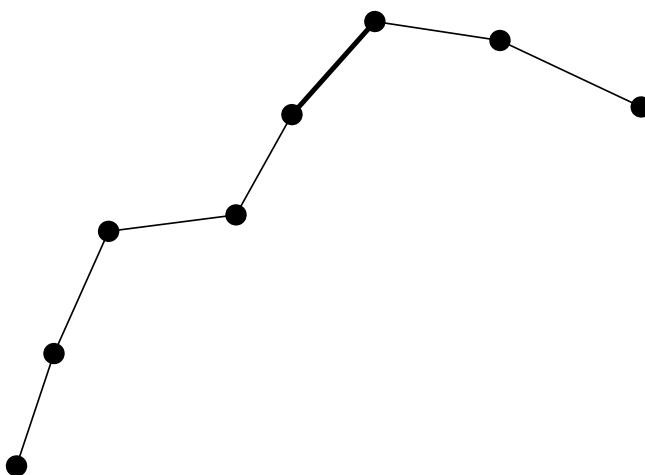
统计矩

统计矩：矩是一个物理量，如均值、方差、高阶矩等。

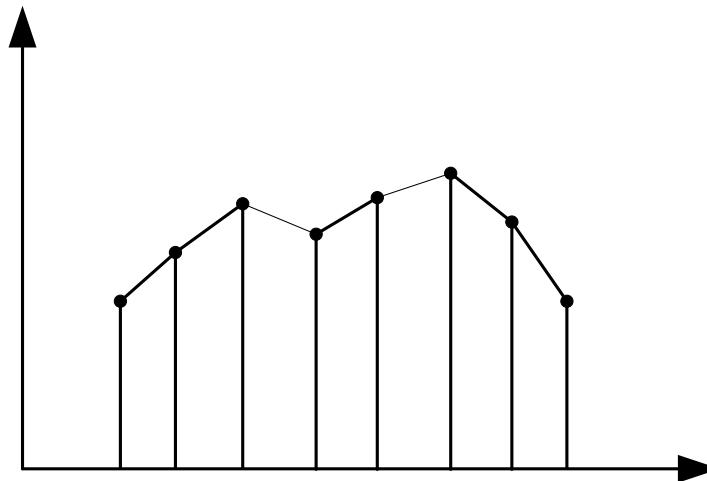
目标的边界可看作由一系列曲线段组成，可将曲线段表示成一个一维函数，并当作直方图来处理。

统计矩

图（a）给出了一个由直线段构成的边界。为了利用统计矩表示边界，需要将边界旋转至由边界上相距最远的两端点确定的线段的水平位置，旋转后的边界可以用变量 r 的一维函数 $g(r)$ 表示，如图（b）所示。



(a) 由直线段构成的边界



(b) 旋转后的边界

图 边界的统计矩描述

统计矩

- 设 m 是 $g(r)$ 的均值，它可以表示为

$$m = \sum_{i=1}^N r_i g(r_i)$$

- 函数 $g(r)$ 对均值 m 的各阶矩定义为

$$\mu_n(r) = \sum_{i=1}^N (r_i - m)^n g(r_i)$$

- 统计矩与 $g(r)$ 的形状有关。利用统计矩对边界进行描述具有对旋转不敏感和与边界空间位置无关的特点。

n 与曲线 $g(r)$ 的形状有直接联系：

- ✓ μ_2 描述了曲线相对于均值的分布（方差）
- ✓ μ_3 描述了曲线相对于均值的对称性（偏度）

目录

- 基于边界的表达
- 基于区域的表达
- 基于边界的描述
- 基于区域的描述

简单区域描述符

- **面积**: 表示面积的方法很多, 用像素数衡量面积最简单。
- **长宽比**: 最小外接矩形长与宽的比值。
- **致密度**: 周长的平方与面积的比。
- **矩形度**: 面积与其最小外接矩形的面积之比。当物体为矩形时, 取得最大值1.0, 圆形物体的取值为 $\pi/4$ 。
- **球形度**: 二维时是内切圆和外接圆半径之比, 两个圆的圆心都在区域的重心上。

简单区域描述符

1、区域面积

区域面积描述区域的大小特征，是区域的基本特性之一。区域面积定义为区域中像素的数目。对于区域R，区域面积表示为：

$$S_R = \sum_{(x,y) \in R} 1$$

其中，等式右侧部分表示当像素在区域R中时，对其进行计数加1。对图像 $f(x, y)$ 中任一点 (x, y) ，如果它在给定的区域内，就取 $f(x, y)$ 为1，否则就取 $f(x, y)$ 为0。所有 $f(x, y)$ 为1的点组成的集合就代表了所要表示的区域

简单区域描述符

2、区域周长

区域周长定义为该区域边界的长度。

3、区域的致密性

设区域R的长度用 L_R 表示，则区域的致密性定义为 L_R^2/S_R 。区域的致密性是一个无量纲的量，当周长固定时，圆形区域的致密性最小（因为当周长固定时，面积最大的平面图形是圆。）。区域的致密性对区域均匀的尺度变化不敏感，对区域的旋转变换也不敏感。

简单区域描述符

4、区域重心

区域重心由所有属于区域中的点计算得到，是区域的一种全局描述子，计算公式如下：

$$\bar{x} = \frac{1}{S_R} \sum_{(x,y) \in R} x$$

$$\bar{y} = \frac{1}{S_R} \sum_{(x,y) \in R} y$$

5、区域圆形性

也称为圆形度或紧凑度，是一个用于衡量一个二维平面区域形状接近完美圆形程度的度量指标。

简单来说，它回答了一个问题：“这个形状有多圆？”

简单区域描述符

5、区域圆形性

区域的圆形性用区域的所有边界点定义的一个特征量来表示，计算公式为

$$C = \frac{\mu}{\sigma}$$

其中： μ 和 σ 分别为区域重心到各边界点距离的平均值和方差。

$$\mu = \frac{1}{N} \sum_{k=0}^{N-1} \sqrt{(x_k - \bar{x})^2 + (y_k - \bar{y})^2}$$

$$\sigma^2 = -\frac{1}{N} \sum_{k=0}^{N-1} [\sqrt{(x_k - \bar{x})^2 + (y_k - \bar{y})^2} - \mu]^2$$

当区域趋向圆形时，特征量C是单调递增趋向无穷的，区域的圆形性不受区域平移、旋转和尺度变的影响。

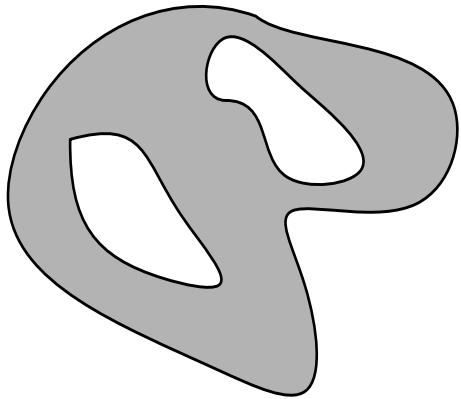
二、拓扑描述子

拓扑学研究图形在没有撕裂和折叠的情况下（又称为橡皮伸展变形），那些不受任何图形变形影响的拓扑性质。拓扑性质既不依赖于距离，又不依赖于基于距离测度的任何性质，因此在图像处理中具有重要意义。孔洞和连通分量是图形的两个重要的拓扑特性。

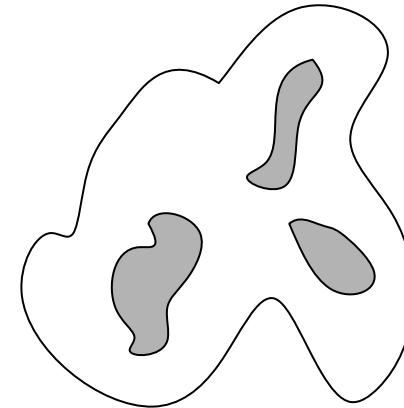
孔指的是不包含感兴趣像素的被封闭边缘包围的区域。

连通分量是指由具有相同像素值（灰度或颜色）且位置相邻的像素组成最大连通区域。连通分量是该拓扑空间中最大的不包含其他点的连通区域。连通分量关注的是感兴趣区域。

二、拓扑描述子



(a) 有两个孔洞的区域



(b) 有3个连通分量的区域，1个孔

二、拓扑描述子

在已知孔洞数目H和连通分量C的基础上，可以进一步定义图形的另一个重要的拓扑特性**欧拉数E**。欧拉数可以用来描述区域的连通性和孔洞的数量。

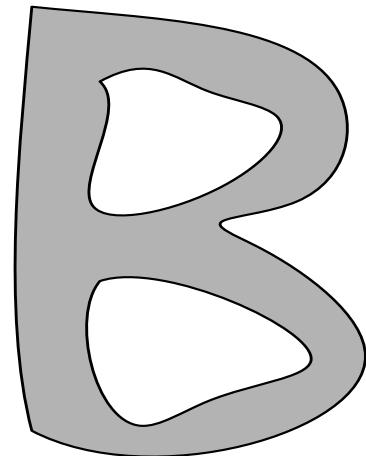
$$E = C - H$$

- 欧拉数描述了区域的连通性。
- H: 区域内的孔数
- C: 区域内的连通组元个数
- 下图中四个字母的欧拉数分别为: -1, 2, 1, 0

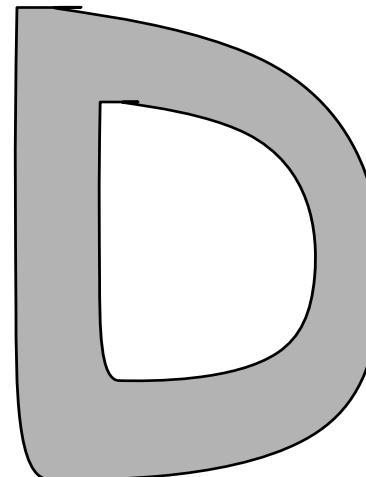
Bird

二、拓扑描述子

例如，图（a）中字母B由于有两个孔洞和一个连通分量，因此字母B的欧拉数为-1；图（b）字母D有一个孔洞和一个连通分量，所以它的欧拉数为0。



(a) 2个孔洞和1个连通分量



(b) 1个孔洞和1个连通分量

二、拓扑描述子

■ 不变矩

- ✓ 不变矩是常用的区域特征描述方法，它具有平移、旋转和尺度等变换的不变性。
- ✓ 若 $f(x,y)$ 为分段连续函数，且它只在平面的有限区域内有非零值，则 $f(x,y)$ 的各阶矩都存在，并且 $f(x,y)$ 唯一地确定一个矩序列 $\{m_{pq}\}$ ；反之，矩序列 $\{m_{pq}\}$ 也唯一确定 $f(x,y)$ 。

二、拓扑描述子

■ 不变矩

图像 $f(x,y)$ 的 $(p+q)$ 阶矩定义为：

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

$(p+q)$ 阶中心矩定义为：

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

其中， $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$

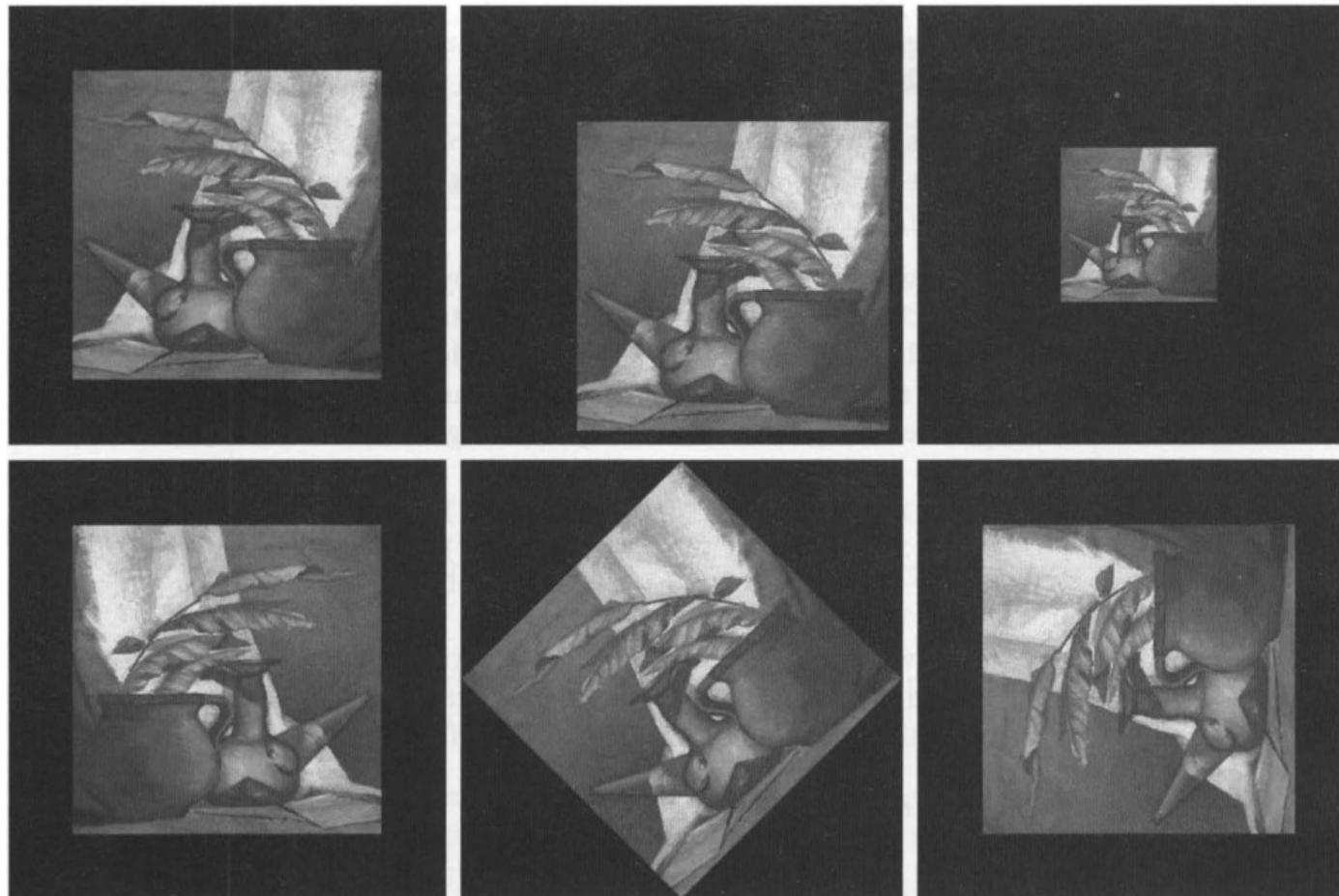
二、拓扑描述子

不变矩

Moment order	Expression
1	$\phi_1 = \eta_{20} + \eta_{02}$
2	$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
3	$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
4	$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
5	$\begin{aligned}\phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\& - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\& [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned}$
6	$\begin{aligned}\phi_6 = & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\& + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned}$
7	$\begin{aligned}\phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\& - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03}) \\& [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned}$

二、拓扑描述子

不变矩性质演示



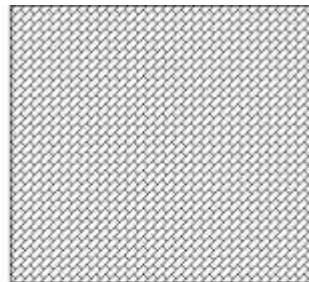
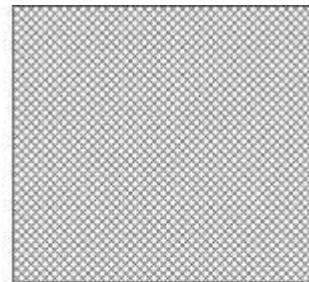
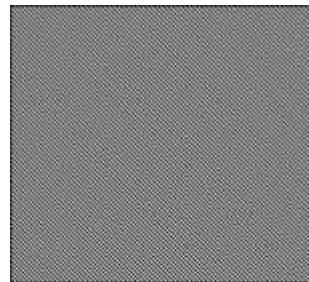
二、拓扑描述子

不变矩性质演示

Moment Invariant	Original Image	Translated	Half Size	Mirrored	Rotated 45°	Rotated 90°
ϕ_1	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
ϕ_2	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
ϕ_3	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
ϕ_4	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
ϕ_5	21.3674	21.3674	21.3924	21.3674	21.3663	21.3674
ϕ_6	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
ϕ_7	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809

纹理

- 纹理 (Texture)：灰度（颜色）在空间以一定的形式变化而产生的图案（模式）。
- 任何物体的表面，如果一直放大下去进行观察的话一定会显现出纹理。
- 对单个像素来说讨论纹理是没有意义的。
- 人工纹理与自然纹理：



纹理

特点：灰度分布一般具有某种**周期性**，即便灰度变化是随机的，也具有一定
的统计特性，周期长的纹理显得粗糙，周期短的纹理细致。

纹理的分析方法

- **统计法：**利用灰度分布的统计规则描述纹理。
- **结构法：**根据描述几何关系的排列规则来描述纹理基元。
- **频谱法：**根据傅里叶频谱的分布来描述纹理的全局周期性质。

纹理

■ 统计法是基于灰度直方图的特性计算纹理描述子，如均值、标准差、平滑度、三阶矩、一致性、熵等。

■ 关于均值的第n阶矩公式 $\mu_n = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i)$

式中 z_i 表示灰度的一个随机变量。 $p(z)$ 是一个区域中灰度级的直方图， L 是可能的灰度级数。 $m = \sum_{i=0}^{L-1} z_i p(z_i)$ 表示平均灰度。

纹理

表 8.2 基于灰度直方图的纹理描绘子

矩	表达式	纹理的测度
均值	$m = \sum_{i=0}^{L-1} z_i p(z_i)$	平均灰度测度
标准差	$\sigma = \sqrt{\mu_2} = \sqrt{\sigma^2}$	平均对比度测度
平滑度	$R = 1 - 1 / (1 + \sigma^2)$	区域中灰度的相对平滑度测度。对于恒定灰度区域, R 为 0; 对于其灰度级的值的最大偏离区域, R 近似为 1。实践中, 该测度中使用的方差 σ^2 被归一化到区间[0, 1], 方法是将它除以 $(L-1)^2$
三阶矩	$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$	直方图偏斜度的测度。对于对称直方图, 该测度为 0; 关于均值右偏的直方图, 该测度为正; 关于均值左偏的直方图, 该测度为负。通过将 μ_3 除以 $(L-1)^2$ (归一化方差时使用了相同的除数), 该测度的值可化到与其他 5 个测度相比较的取值范围
一致性	$U = \sum_{i=0}^{L-1} p^2(z_i)$	一致性测度。当所有的灰度值相等时(最大一致), 该测度最大, 然后减小
熵	$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$	随机性测度

纹理

```
%统计纹理  
clc  
clear  
f = imread('Fig1128(a)[original].tif');  
imshow(f)  
title('原始图像')  
t=statxture(f)
```

119.3463 %平均值
53.3071 %标准差
0.0419 %平滑度
1.7436 %三阶矩
0.0105%一致性
6.8022 %熵



灰度共生矩阵

灰度共生矩阵（Gray-level Co-occurrence Matrix, GLCM），是一种用于纹理分析的统计方法，它可以捕捉图像中的空间结构和纹理信息。

应用广泛：

1. 遥感和地理信息系统。土壤分析：通过分析卫星或航空图像来识别和分类土壤类型。

植被分析：用于分析植被的纹理特征，以帮助识别和分类不同类型的植被。

2. 工业检测。产品质量控制：通过分析产品表面的纹理来进行质量控制。缺陷检测：

用于检测产品上的缺陷或损坏。

3. 计算机视觉和图像处理。对象识别：通过分析对象的纹理特征来进行对象识别。图

像分割：用于将图像分割成具有不同纹理特征的区域

灰度共生矩阵

■ 定义：对图像所有像素进行统计调查，描述其灰度分布的方法。

取图像中点 (x, y) 及偏离点 $(x + \Delta x, y + \Delta y)$ ，假设该点对的灰度值为 (f_1, f_2) ，移动点 (x, y) ，得到各种 (f_1, f_2) 值。设灰度值级数为 L ，则 f_1 与 f_2 的组合共有 L^2 种。对于整幅图像，统计出每一种 (f_1, f_2) 值出现的次数，排列成方阵，归一化为 (f_1, f_2) 出现的概率 $p(f_1, f_2)$ ，称方阵为联合概率矩阵，也称为灰度共生矩阵。

灰度共生矩阵

灰度共生矩阵

- 也可以通过设定方向 θ 和距离d来确定灰度对(f_1, f_2)，进而生成灰度共生矩阵。
- 偏离值($\Delta x, \Delta y$)取不同的值，可以形成不同的灰度共生矩阵。通常，
($\Delta x, \Delta y$) 根据纹理周期分布特性选择：变化缓慢的图像，($\Delta x, \Delta y$) 较小时，
 f_1 与 f_2 一般具有相近的灰度，体现在共生矩阵中，矩阵对角线及其附近的
数值较大；变化较快的图像，矩阵各元素的取值相对均匀。

灰度共生矩阵

灰度共生矩阵

- 灰度共生矩阵能反映图像灰度关于方向、相邻间隔、变化幅度等信息。
- 灰度共生矩阵用两个位置像素的联合概率密度来定义。
- 灰度共生矩阵的第*i*行、第*j*列元素表示在图像中，从灰度值为*i*的像素出发，沿着固定方向（ θ 角度）和距离（ δ ）移动到灰度值为*j*的像素的出现次数或概率。

灰度共生矩阵

2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2
14	2	6	10	14	2	6
2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2

(a) 原图像

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

(b) 灰度级减为4级

图a为原图像，最大灰度级为16。这里将灰度级数压缩为4级，此时的GLCM是4阶方阵(GLCM的阶数等于灰度的等级数)。图a变为图b的形式。

这样(f_1, f_2)取值范围为[0,3]。取不同的位置关系，将(f_1, f_2)各种组合出现的次数排列起来，就可得到不同的灰度共生矩阵。

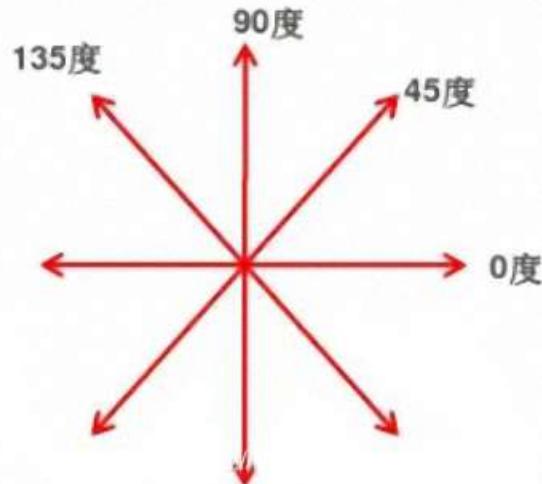
灰度共生矩阵

假设一对像素的空间位置关系表示为 $\delta=(a,b)$, (a,b) 的取值不同，灰度共生矩阵中的值不同。

- a 和 b 的取值要根据纹理周期分布的特征来选择，对于较细的纹理，选取(1,0), (1,1), (2,0)等这样的值是有必要的。
- a, b 取值较小对应于变化缓慢的纹理图像，其灰度共生矩阵对角线上的数值较大。
- 纹理的变化越快，则对角线上的数值越小，而对角线两侧的值增大。

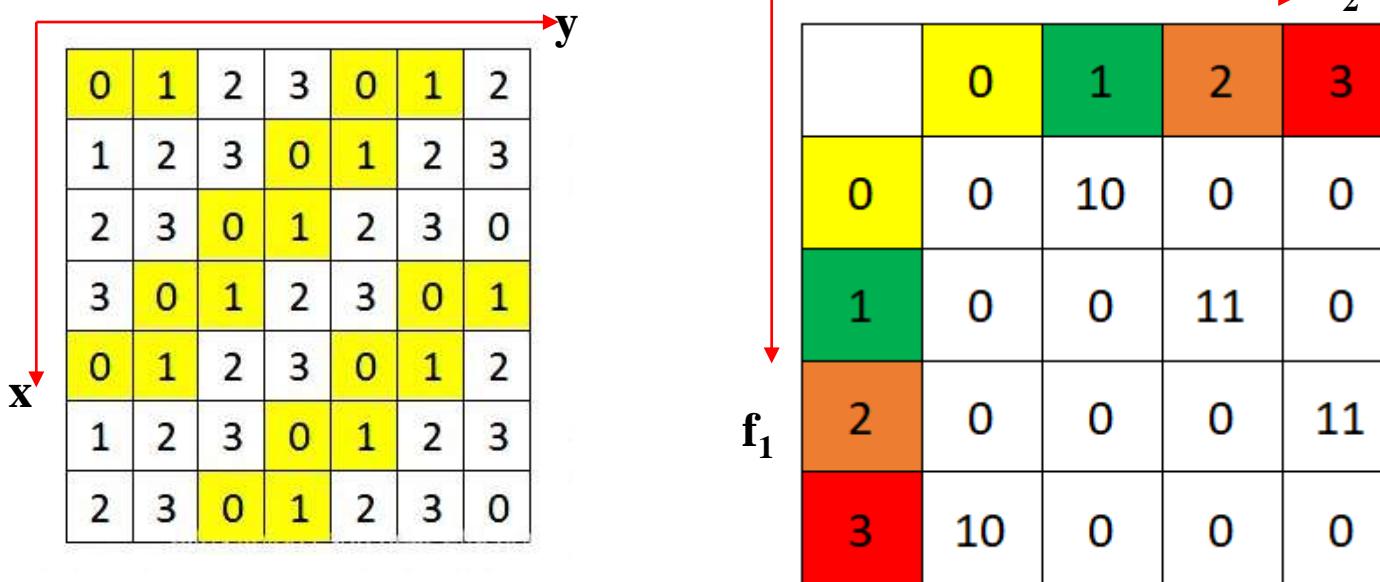
灰度共生矩阵

- (a,b)=(1,0), 像素对是水平的, 即0度扫描;
- (a,b)=(0,1), 像素对是垂直的, 即90度扫描;
- (a,b)=(1,1), 像素对是右对角线的, 即45度扫描;
- (a,b)=(-1,1), 像素对是左对角线, 即135度扫描。



灰度共生矩阵

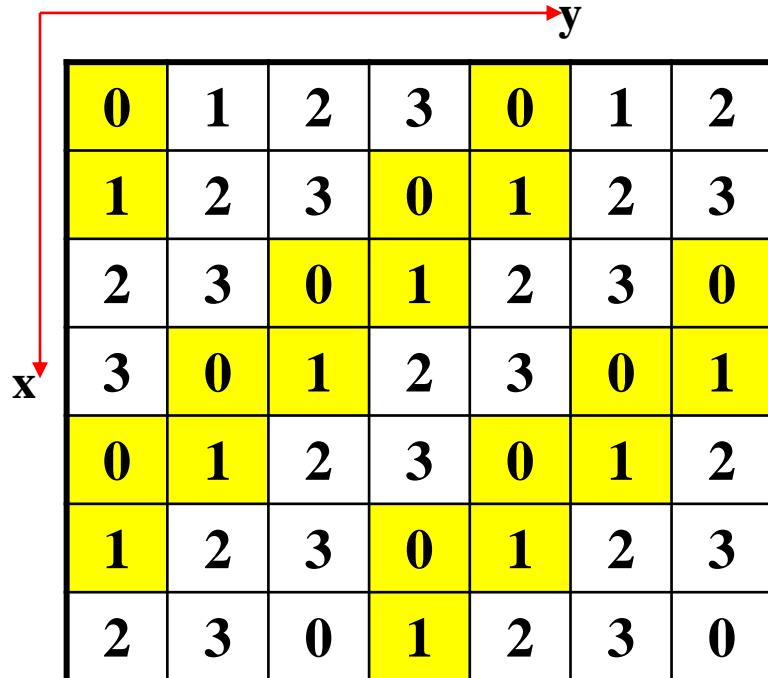
(x,y) 与偏离它的 $(0,1)$ 构成点对时，即比较 (x,y) 与 $(x,y+1)$ 这两个点， $(f_1, f_2) = (0,1)$ 取值的情况共10种(图a中填充部分)。



灰度共生矩阵

■ 示例

2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2
14	2	6	10	14	2	6
2	6	10	14	2	6	10
6	10	14	2	6	10	14
10	14	2	6	10	14	2



原图像

灰度级减为4级

f_1, f_2 分别取值为 0、1、2、3，将 (f_1, f_2) 各种组合出现的次数排列起来，得联合概率矩阵。右图中 $42 = 10 + 11 + 11 + 10$ ，即当前 (f_1, f_2) 各种组合出现的总次数。

$$\begin{array}{l} f_2 \\ \hline \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 0 & 0 & 10 & 0 & 0 \\ 1 & 0 & 0 & 11 & 0 \\ 2 & 0 & 0 & 0 & 11 \\ 3 & 10 & 0 & 0 & 0 \end{array} \end{array} \quad (\Delta x = 1, \Delta y = 0)$$

$$\begin{pmatrix} 0 & 10/42 & 0 & 0 \\ 0 & 0 & 11/42 & 0 \\ 0 & 0 & 0 & 11/42 \\ 10/42 & 0 & 0 & 0 \end{pmatrix}$$

灰度共生矩阵

$(\Delta x, \Delta y)$ 取不同的数值组合，得到不同的联合概率矩阵：

$$\begin{array}{c} f_2 \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \left(\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 11 \\ 10 & 0 & 0 & 0 \end{array} \right) \begin{array}{c} 0 \\ 10/42 \\ 0 \\ 11/42 \\ 0 \end{array}$$
$$\begin{array}{c} f_1 \\ 1 \\ 2 \\ 3 \end{array} \left(\begin{array}{cccc} 0 & 10/42 & 0 & 0 \\ 0 & 0 & 11/42 & 0 \\ 0 & 0 & 0 & 11/42 \\ 10/42 & 0 & 0 & 0 \end{array} \right)$$

$(\Delta x = 1, \Delta y = 0)$ 联合概率矩阵

$$\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \left(\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 10 \\ 9 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \end{array} \right) \begin{array}{c} 0 \\ 0 \\ 0 \\ 10/36 \\ 9/36 \end{array}$$

$(\Delta x = 2, \Delta y = 0)$

灰度共生矩阵

$$\begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 8 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 9 \end{pmatrix} \quad \begin{pmatrix} 8/36 & 0 & 0 & 0 \\ 0 & 9/36 & 0 & 0 \\ 0 & 0 & 10/36 & 0 \\ 0 & 0 & 0 & 9/36 \end{pmatrix} \end{matrix}$$

$(\Delta x = 1, \Delta y = 1)$

$\Delta x, \Delta y$ 的取值要根据纹理周期分布的特性来选择，较细的纹理选取较小的差分值。

灰度共生矩阵

■ 共生矩阵二次统计量

作为纹理分析的特征量，一般不是直接应用计算的灰度共生矩阵，而是在灰度共生矩阵的基础上再提取纹理特征量，称为**二次统计量**。ASM（Angular Second Moment, 角二阶矩）是灰度共生矩阵（GLCM）中的一个纹理特征，通常用于衡量图像的均匀性或平滑性。它是通过计算灰度共生矩阵中每对灰度级像素的值的平方和来量化图像的纹理细节。

$$\text{角二阶矩(能量): } ASM = \sum_{f_1} \sum_{f_2} [p(f_1, f_2)]^2$$

$p(f_1, f_2)$ 是灰度共生矩阵中，灰度对 (f_1, f_2) 出现的概率。

灰度共生矩阵

■ 共生矩阵二次统计量

若所有图像具有相同的灰度级 f , $p(f,f) = 1$ 且 $p(f_1,f_2) = 0$ ($f_1 \neq f$ 或者 $f_2 \neq f$) , 则ASM=1; 如具有所有可能的像素对, 且像素的灰度级具有相同的概率, 则ASM等于这个概率值; 区域越不平滑, $p(f,f)$ 越均匀, 则ASM越低。

- ✓ **ASM越大:** 表示图像中的纹理更加均匀, 变化较小。图像中大部分像素的灰度值分布较为集中, 即图像的纹理较为平滑、规则。
- ✓ **ASM越小:** 表示图像中的纹理更加复杂, 灰度值变化较大。图像中有更多的纹理变化或不规则性。

灰度共生矩阵

■ 共生矩阵二次统计量

- ✓ 在一些平坦的图像中，如大面积的单一颜色区域，灰度共生矩阵的值会较为集中，因此ASM较大。
- ✓ 在一些含有丰富细节或纹理变化的图像中，如复杂的纹理或自然图像，灰度共生矩阵的值分布更分散，ASM较小。
- ✓ ASM是图像纹理分析中的一个重要特征，常用于图像分类、识别和模式分析等领域。

灰度共生矩阵

■ 共生矩阵二次统计量

对比度：

$$CON = \sum_k k^2 \left[\sum_{f_1} \sum_{f_2} p(f_1, f_2) \right]_{k=|f_1-f_2|}$$

$$\frac{1}{30} \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad (\Delta x = 1, \Delta y = 0)$$

对角线取值大，亮度变化慢， CON 小

图像像素值变化很快，则
 CON 会有较大取值

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

$$\frac{1}{30} \begin{pmatrix} 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 \\ 7 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \end{pmatrix} \quad (\Delta x = 1, \Delta y = 0)$$

非对角线取值大，亮度变化快， CON 较大

灰度共生矩阵

■ 共生矩阵二次统计量

倒数差分矩:

$$IDM = \sum_{f_1} \sum_{f_2} \frac{p(f_1, f_2)}{1 + |f_1 - f_2|}$$

熵:

$$ENT = - \sum_{f_1} \sum_{f_2} p(f_1, f_2) \log_2 p(f_1, f_2)$$

平滑图像熵值小

相关系数:

$$COR = \frac{\sum_{f_1} \sum_{f_2} (f_1 - \mu_{f_1})(f_2 - \mu_{f_2}) p(f_1, f_2)}{\sigma_{f_1} \sigma_{f_2}}$$

$$\mu_{f_1} = \sum_{f_1} f_1 \sum_{f_2} p(f_1, f_2)$$

$$\mu_{f_2} = \sum_{f_2} f_2 \sum_{f_1} p(f_1, f_2)$$

$$\sigma_{f_1}^2 = \sum_{f_1} (f_1 - \mu_{f_1})^2 \sum_{f_2} p(f_1, f_2) \quad \sigma_{f_2}^2 = \sum_{f_2} (f_2 - \mu_{f_2})^2 \sum_{f_1} p(f_1, f_2)$$

灰度共生矩阵

■ 共生矩阵二次统计量

打开一幅灰度图像，生成联合概率矩阵并计算参数

%灰度共生矩阵

clc,clear,close all;

f=rgb2gray(imread('texture.bmp'));%读取纹理图像

[g1,SI1]=graycomatrix(f,'G',[]); %生成灰度共生矩阵

status1=graycoprops(g1) %计算属性矩阵status1

f=filter2(fspecial('average',min(size(f))/8),f); %对原图高强度滤波

[g2,SI2]=graycomatrix(f,'G',[]); %对滤波后的图像生成灰度共生矩阵

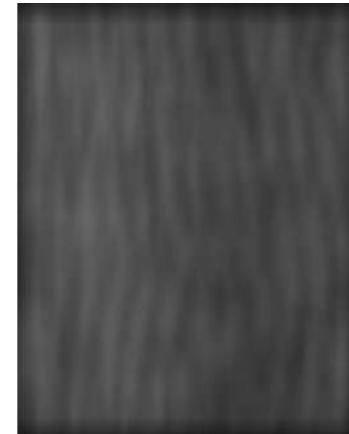
status2=graycoprops(g2) %计算属性矩阵status2

灰度共生矩阵

■ 实例



status1 =
 Contrast: 0.6303
 Correlation: 0.7874
 Energy: 0.0901
 Homogeneity: 0.7628



status2 =
 Contrast: 0.0893
 Correlation: 0.9606
 Energy: 0.2396
 Homogeneity: 0.9553

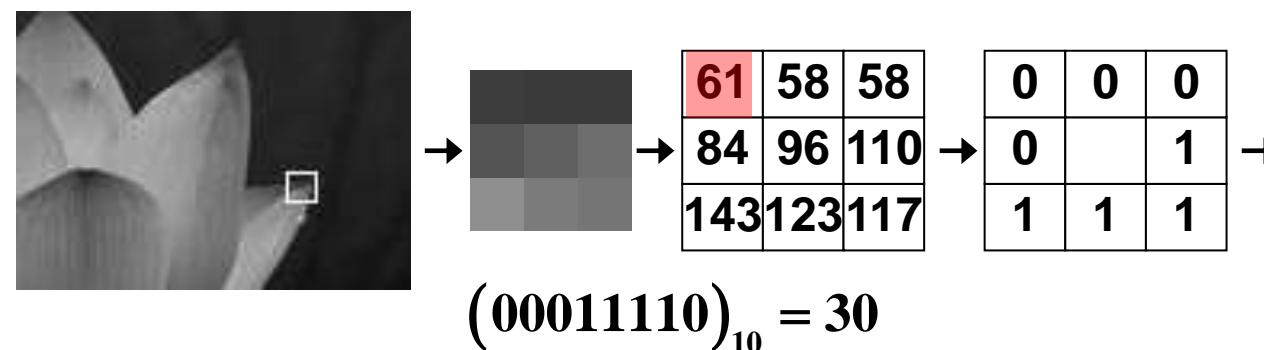
平滑后图像对比度降低，自相关性增强，角二阶矩和倒数差分矩增大。

LBP特征

LBP（Local Binary Pattern），局部二元模式，一种用于纹理分析的图像处理算法，它可以用来描述图像中的局部纹理特征。该算法最初由T. Ojala等人于1996年提出。LBP不是一个完整的特征描述子，具有旋转不变性和灰度不变性。LBP常应用于人脸识别和目标检测中。

■ LBP特征提取

3×3的窗口内，灰度值大于中心像素的位置记为1，否则为0，产生8位无符号二进制数，转换为十进制数，即为该窗口中心像素点的LBP值



LBP特征

- **灰度不变性：**指的是LBP特征对于图像的光照变化不敏感。这是因为LBP是通过比较像素与其邻域内的像素值来定义的，而不是直接使用像素的绝对灰度值。具体来说，在计算LBP时，每个像素点的邻域（通常是 3×3 或更大的邻域）内的像素值与该中心像素点的灰度值进行比较，生成一个二进制模式。由于比较的是相对值，因此，当整个图像的光照发生变化时，这种相对关系保持不变，从而使得LBP特征具有灰度不变性。
- **旋转不变性：**指LBP特征对于图像的旋转具有不变性。原始的LBP算子并不是旋转不变的，因为当图像旋转时，像素的位置会发生变化，导致LBP值发生变化。因此学者提出了**LBP旋转不变模式**。

LBP特征

通常将图像分为 $n \times n$ 的子区域，对子区域内的像素点计算LBP值，并统计其直方图，以直方图作为其判别特征。

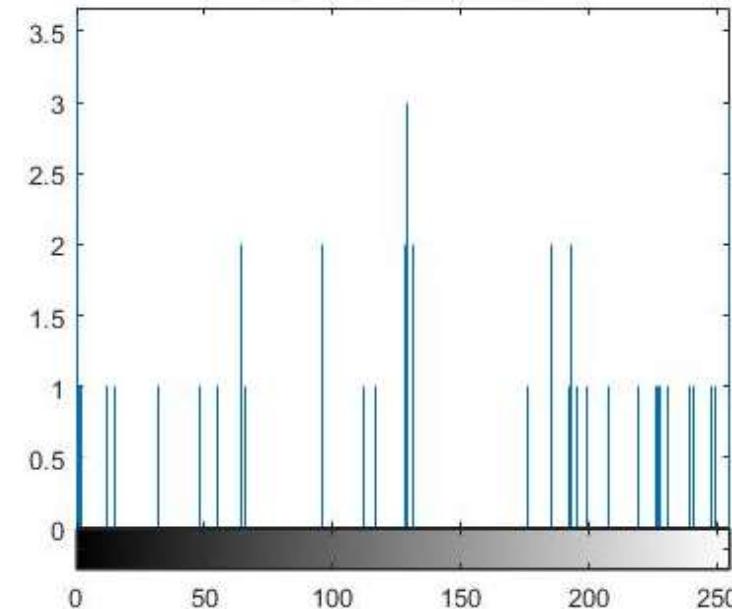
计算Lena图像的LBP特征图

```
image=imread('lena.bmp');
[N,M]=size(image);
lbp=zeros(N,M);
for j=2:N-1
    for i=2:M-1
        neighbor=[j-1 i-1;j-1 i;j-1 i+1;j i+1;j+1
i+1;j+1 i;j+1 i-1;j i-1];
        count=0;
        for k=1:8
            if image(neighbor(k,1),neighbor(k,2))>image(j,i)
                count=count+2^(8-k);
            end
        end
        lbp(j,i)=count;
    end
end
lbp=uint8(lbp);
figure,imshow(lbp),title('LBP特征图');
subim=lbp(1:8,1:8);
imhist(subim),title('第一个子区域直方图');
```

LBP特征



LBP特征图



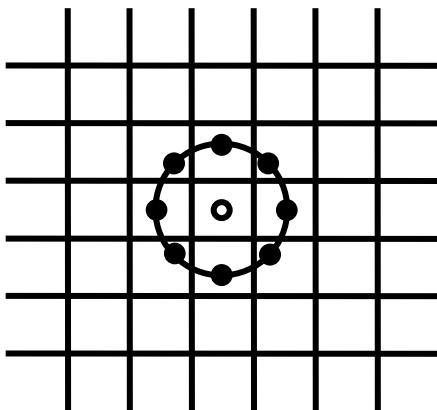
第一个 8×8 子区域
LBP直方图

■ 圆形LBP算子

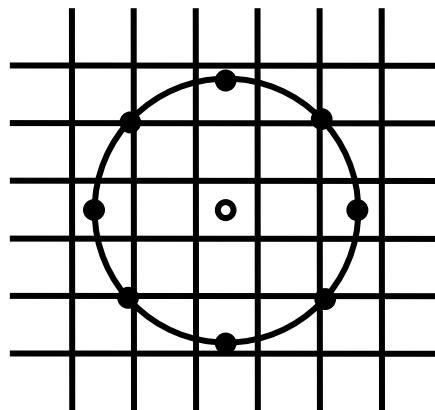
原始的LBP算子只覆盖了一个固定半径范围内的小区域，**不能满足尺寸和频率纹理的需要**。为了适应不同尺度的纹理特征，并且依然能够满足旋转不变性和灰度不变性，对原始的LBP算子进行改进，**将 3×3 的邻域扩展到任意的邻域，并用圆形替代了原始的正方形邻域**，改进后的LBP算子允许在半径为R的圆形邻域内有P个采样点，表示为 LBP_P^R 。

LBP特征

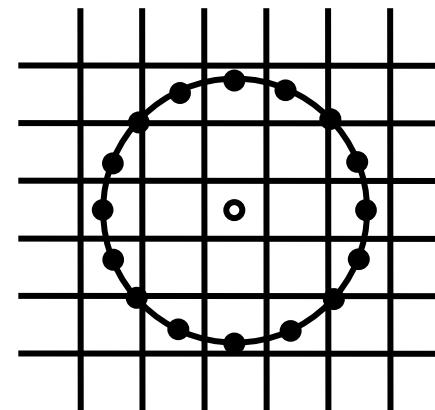
圆形LBP算子



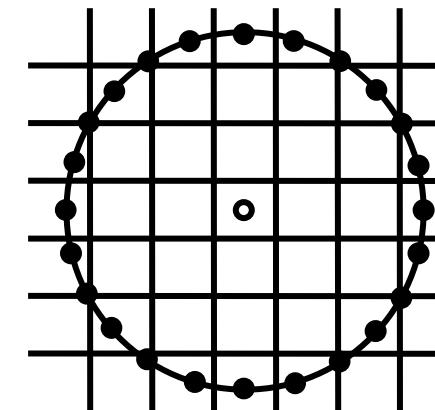
LBP_8^1



LBP_8^2



LBP_{16}^2



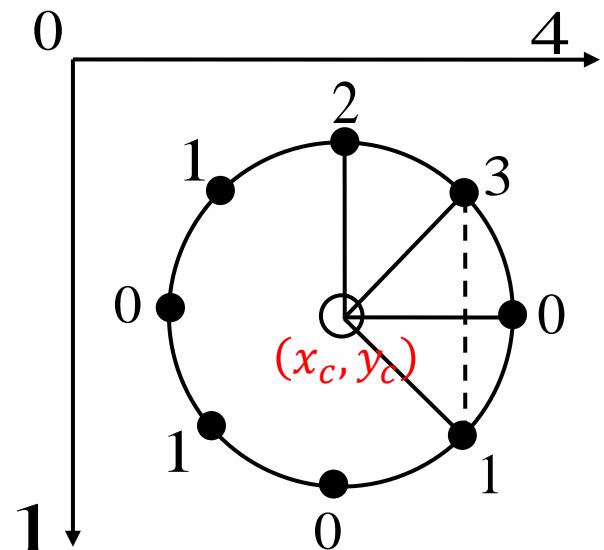
LBP_{24}^3

LBP特征

圆形LBP算子

设中心像素点为 (x_c, y_c) , 黑色采样点为 (x_k, y_k) , $k = 0, 1, \dots, P - 1$, 采样点为非整数像素, 用插值方法确定其像素值。

$$\begin{cases} x_k = x_c + R \times \cos\left(\frac{2\pi k}{P}\right) \\ y_k = y_c + R \times \sin\left(\frac{2\pi k}{P}\right) \end{cases}$$

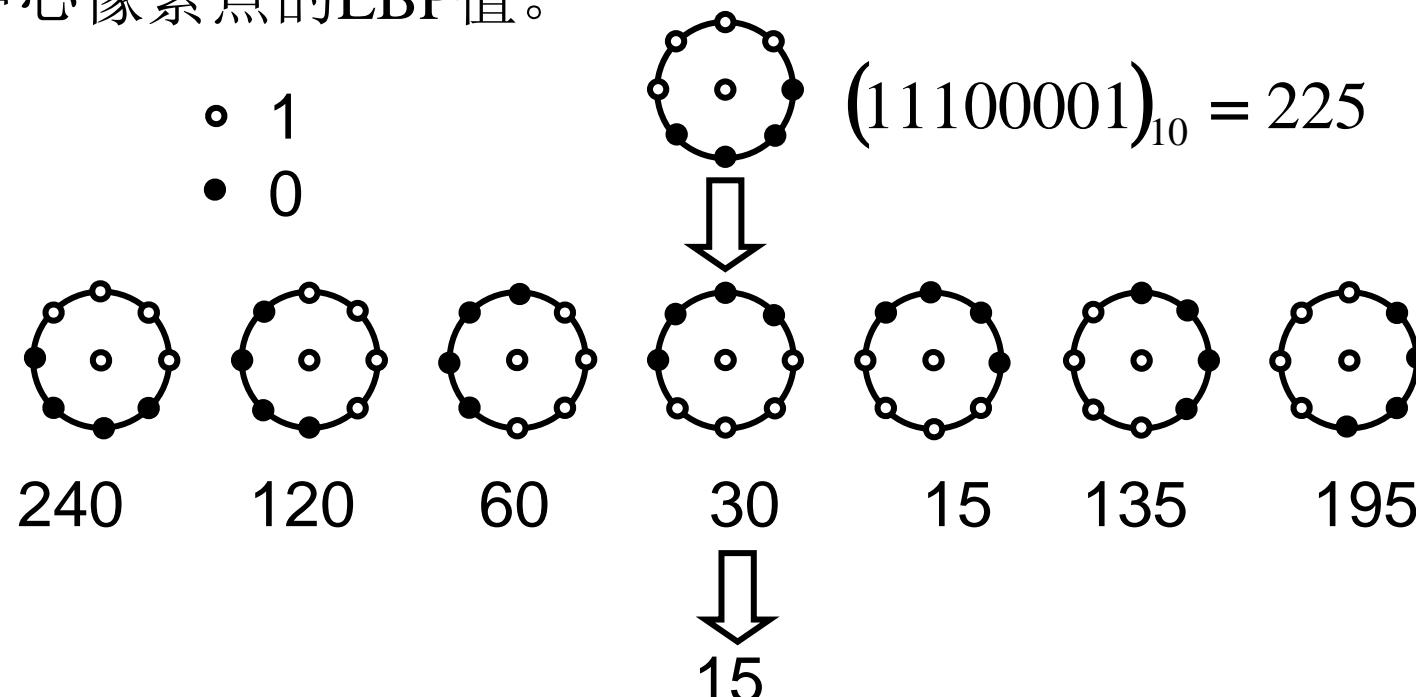


圆形LBP采样点

LBP特征

■ LBP旋转不变模式

圆形LBP特征具有灰度不变性，但还**不具备旋转不变性**，因此提出了具有**旋转不变性**的LBP。不断旋转圆形邻域内的LBP特征，得到一系列LBP值，选择**值最小**的作为中心像素点的LBP值。



LBP特征

■ 实例 计算Lena图像的LBP旋转模式特征图

```
image=imread('lena.bmp');
[N,M]=size(image);
P=8; R=2;
clbp=zeros(N,M);
for j=1+R:N-R
    for i=1+R:M-R
        count=0;
        for k=0:P-1
            x=i+R*cos(2*pi*k/P);    y=j+R*sin(2*pi*k/P);
            Lowx=floor(x);Highx=ceil(x); Lowy=floor(y);Highy=ceil(y);
            coex=x-Lowx; coey=y-Lowy;
```

LBP特征

```
a=image(Lowy,Lowx)+coex*(image(Lowy,Highx)-image(Lowy,Lowx));  
b=image(Highy,Lowx)+coex*(image(Highy,Highx)-image(Highy,Lowx));  
pixel=a+coey*(b-a); %双线性插值  
if pixel>image(j,i)  
    count=count+2^(P-1-k);  
end  
end  
lbp=dec2bin(count);  
mincount=count;
```

LBP特征

```
for k=1:P-1  
    lbp = circshift(lbp',1)'; % 循环移位函数  
    count=bin2dec(lbp);  
    if mincount>count  
        mincount=count;  
    end  
end  
clbp(j,i)=mincount;  
end  
end  
figure,imshow(uint8(clbp));  
title('LBP旋转模式特征图 '');
```



LBP_4^3



LBP_8^2



LBP_8^1



LBP_{16}^2

频谱法

- 频谱法：借助傅立叶频谱的频率特性来描述二维图像的方向性
- 频谱法基于信号处理理论，将纹理看作一个信号，通过对其频谱进行分析来提取和描述纹理的特征。
- 频谱法可以将纹理分解为一系列的频率成分，从而捕捉到纹理的结构和细节。这些频率成分可以用于纹理分析、纹理合成和纹理识别等应用。

■ 频谱法的基本思想：

将纹理图像转换为频域，通过对频域进行分析来提取纹理的特征。在频谱分析中，常用的工具是傅里叶变换。傅里叶变换可以将一个信号在频域上的成分分解出来，将其表示为一系列的频谱分量，并通过频谱图像来观察纹理的特征。

频谱法

借助傅里叶变换得到傅里叶频谱，傅里叶频谱有三个合适描述纹理的性质：

(1) 傅里叶频谱中突起的峰值对应纹理模式的主方向

(2) 频率平面中的尖峰位置对应模式的基本空间周期

(3) 利用滤波将周期性成分除去，留下非周期分量，并利用统计方法描述剩下的非周期性部分。

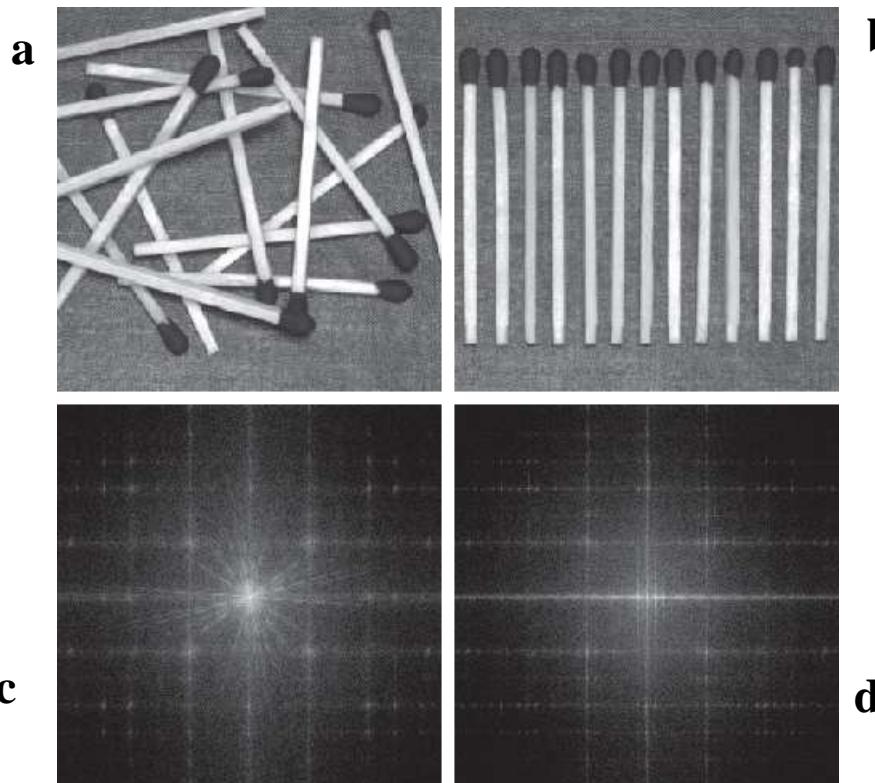
频谱法

一般情况，使用极坐标频谱函数 $S(r, \theta)$ 来表示。其中 r 和 θ 是坐标中的半径和角度。对每个方向 θ 而言， $S(r, \theta)$ 可视为一维函数 $S_\theta(r)$ 。对每个频率 r 来说， $S_r(\theta)$ 也可视为一维函数。当 θ 固定时，通过分析 $S_\theta(r)$ ，可得沿原点辐射方向的频谱特性。相反，当 r 固定时，通过分析 $S_r(\theta)$ ，可得以原点为圆心的圆形的特性。对离散变量求和，可得到全局纹理的描述。

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \quad S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)$$

式中 R_0 是以原点为圆心的圆半径， $S(r)$ 是离圆心距离为 r 的图像频谱值的总和， $S(\theta)$ 是旋转角度为 θ 时的图像频谱值的总和。对极坐标的每一对 (r, θ) ， $[S(r), S(\theta)]$ 构成了对整个区域的纹理频谱能量的描述。

频谱法

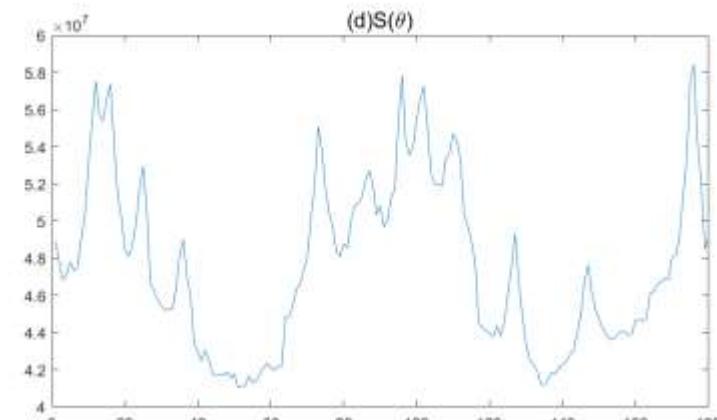
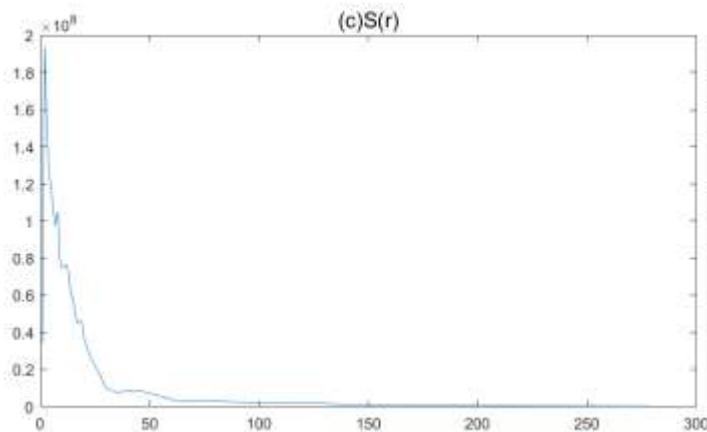
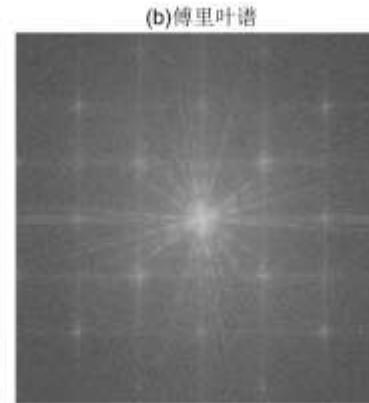


(a)随机目标图像, (b)有序目标图像, (c)随机图像的傅里叶谱, (d)有序图像的傅里叶谱

频谱法

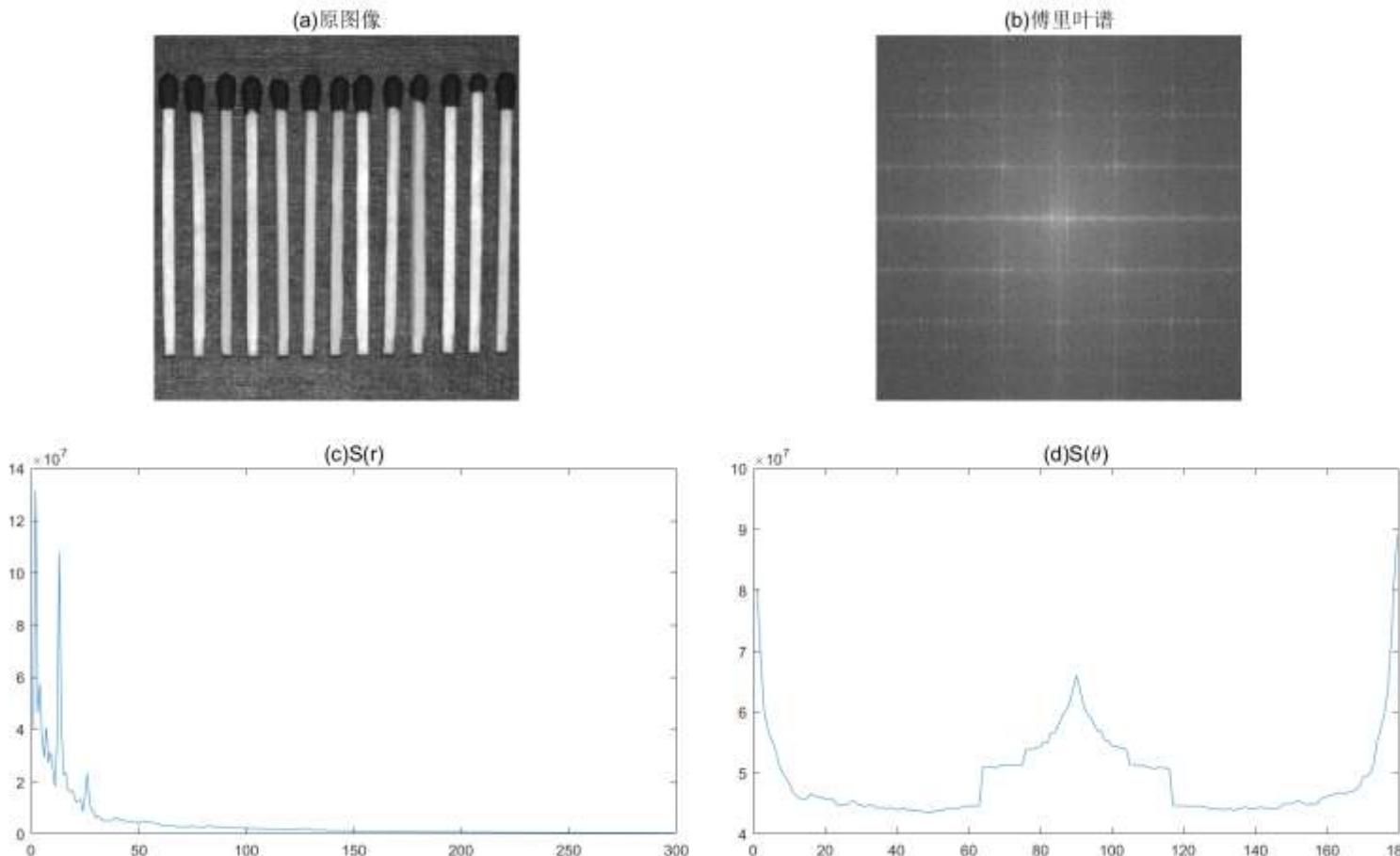
```
%% 例11.11 计算频谱纹理 specxture  
clc,  
clear,  
close all,  
A = imread('Fig1121(a)(random_matches).tif');  
subplot(2,2,1),imshow(A);title('(a)原图像','FontSize',14);  
[srada, sanga, Sa] = specxture(A);  
subplot(2,2,2),imshow(Sa);title('(b)傅里叶谱','FontSize',14);  
subplot(2,2,3), plot(srada);title('(c)S(r)','FontSize',14);  
subplot(2,2,4), plot(sanga);title('(d)S(\theta)','FontSize',14);
```

频谱法



随机目标的 $S(r)$ 图形没有强周期分量（除了原点有一个直流分量峰，谱中没有其他主峰）

频谱法



有序目标的 $S(r)$ 图形在 $r=15$ 附近有一个强峰，在 $r=25$ 处有一个弱峰直流分量峰，谱中没有其他主峰，它们分别对应与图像中亮（目标）区域和暗（背景）区域的周期性重复。

主分量描述

主分量分析（Principal Component Analysis, PCA），又称主成分分析，将关联性较强的变量变换为数量较少的不相关的变量的方法（降维）。

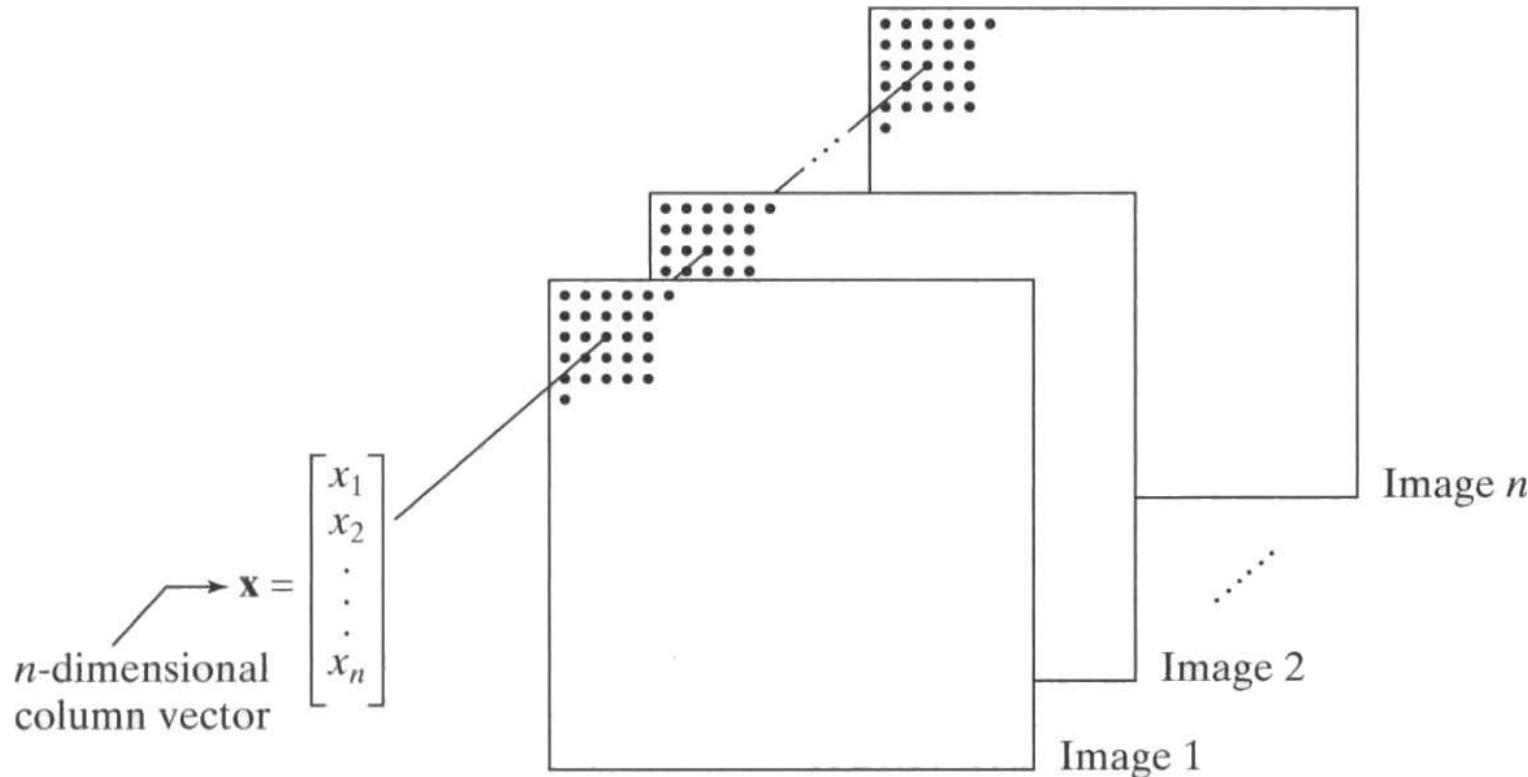
PCA是对多维信号进行降维操作的一种重要手段，主成分分析从多维信号中提取出低维的主要信息。主成分描述就是使用主成分来描述原来的高维复杂信号。主成分分析先计算输入的多维信号的协方差矩阵，然后得到该协方差矩阵的特征向量和特征值，将特征向量按照它们所对应的特征值大小进行排序后，挑选出比较重要的特征向量(对应的特征值较大)，最后将原始信号投影到挑选出的特征向量上，从而构建出一个低维信号。

主分量描述

- 在图像领域，使用不同角度、不同手段通常会得到多幅图像，如果忽略图像的空间特性将图像看成一维序列，则 n 幅图像就构成了 n 维信号。**主成分描述的作用就是将多幅已配准的图像进行融合，提炼出比较重要的若干幅图像来对原始采集到的多幅图像进行描述，从而达到降维的目的。**
- 在通常情况下，主成分分析仍然会得到一个 n 维信号，但是该 n 维信号已经按照其**重要性进行了排列**，并且其重要性也有确切的参数来描述，可以选择其中最重要的 k 维作为最终的主成分描述。通常，需要设定一个值，使得这个**k维信号的重要性之和占全部重要性之和的百分比大于该阈值**。

主分量描述

将多幅图像中的每个像素看成一个高维向量，多幅图像由高维空间中的一些点组成。



主分量描述

设图像尺寸为 $M \times N$ ，在n幅图像中，n维向量X有 $K=M \times N$ 个，它们的均值为：

$$m_x = \frac{1}{K} \sum_{k=1}^K x_k$$

其协方差矩阵为： $C_x = \frac{1}{K-1} \sum_{k=1}^K (x_k - m_k)(x_k - m_k)^T$

C_x 为实对称矩阵，求出其特征向量及对应的特征值，并按降序对特征值排序，以对应次序将特征向量从上到下写出矩阵A，用A作霍特林（Hotelling）变换（或KL变换）：

$$Y = A(X - m_x)$$

主分量描述

Y 的均值向量 m_y 为零向量，它的协方差矩阵为： $C_y = AC_xA^T$

C_y 是 Cx 的特征值按降序方式 ($\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$) 排列的对角矩阵：

$$C_y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

因为矩阵 A 的行是正交的，所以 $A^{-1} = A^T$ ，可以执行反变换来恢复 X ： $X = A^T Y + m_x$

主分量描述

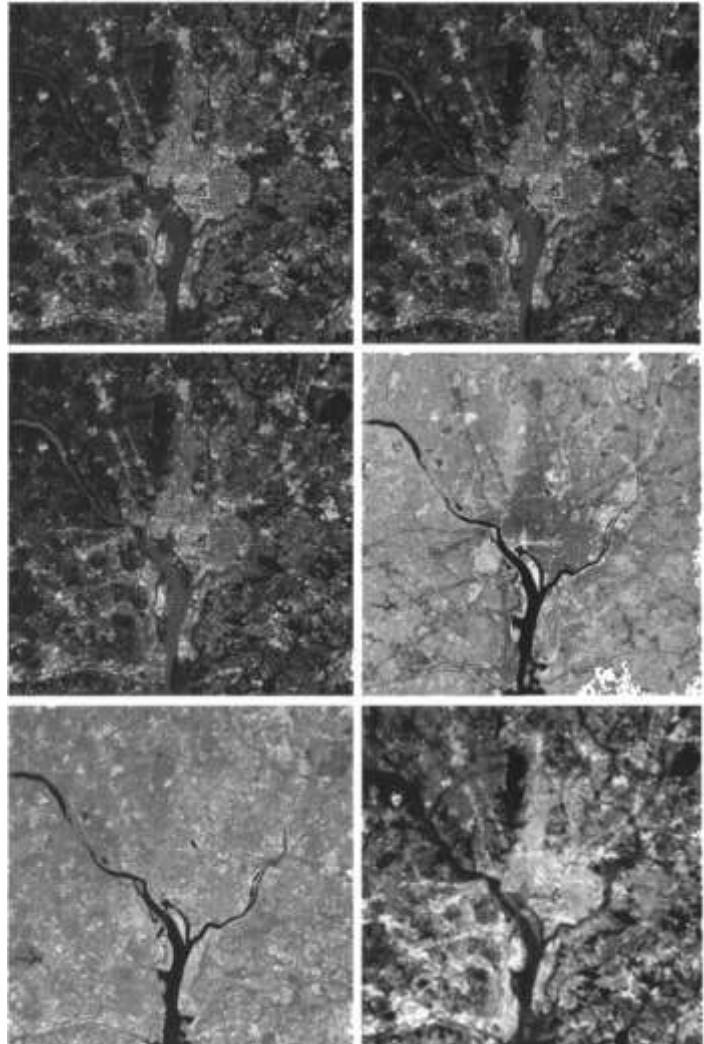
- 如果仅使用 q 个特征向量 ($q < n$) 来恢复 X , 重建的近似值为: $\hat{X} = A_q^T Y + m_x$
- \hat{X} 与 X 的均方误差为: $e_{ms} = \sum_{i=1}^n \lambda_i - \sum_{i=1}^q \lambda_i = \sum_{i=q+1}^n \lambda_i$
- 选取最大特征值的 q 个特征向量, 可以使误差最小。
- 在最小均方误差 (Mean Square Error, MSE) 的意义下, 主分量变换是最佳的。
- 协方差矩阵为对角矩阵, 使各分量不相关。

主分量描述

MATLAB中提供了PCA函数，需要安装 Statistics and Machine Learning Toolbox

- [COEFF,SCORE,LATENT]=pca(data)
- [COEFF,SCORE,LATENT]= princomp(data)
 - ✓ COEFF: 各个主成分
 - ✓ SCPRE: 主成分上的投影
 - ✓ LATENT: 各个特征值

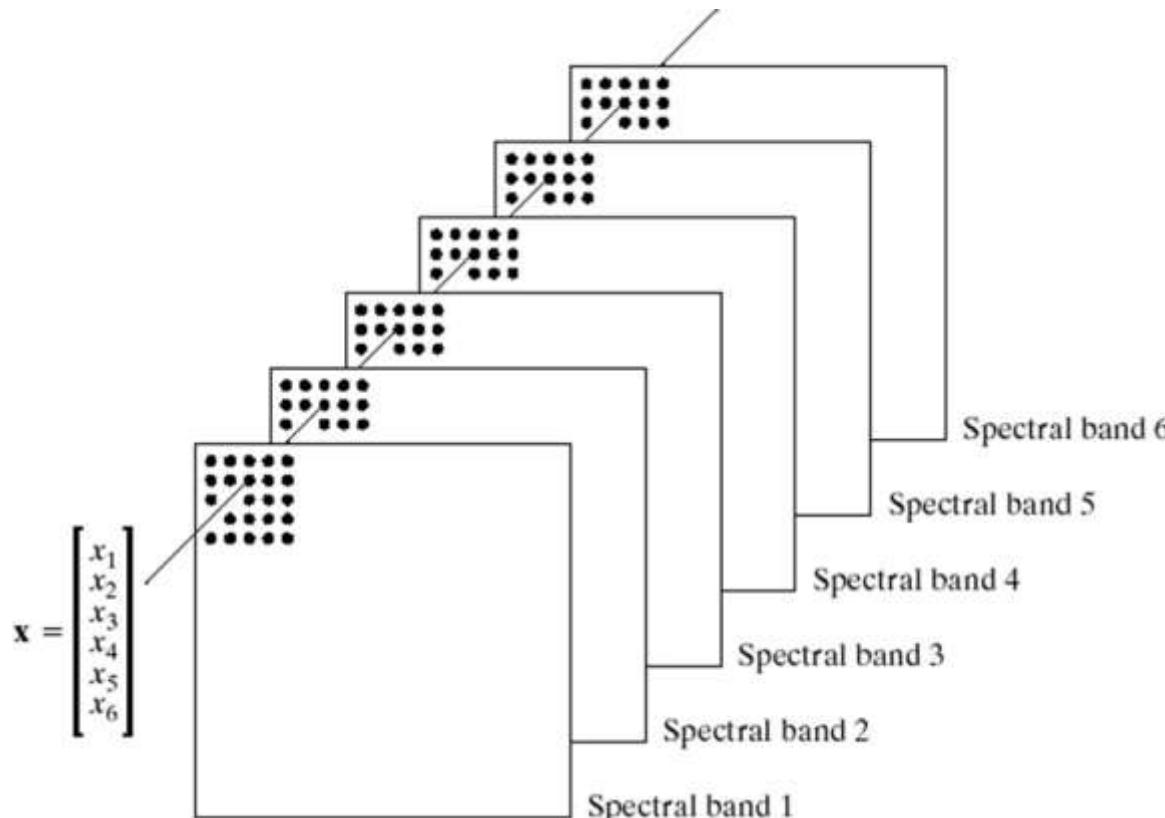
主分量描述



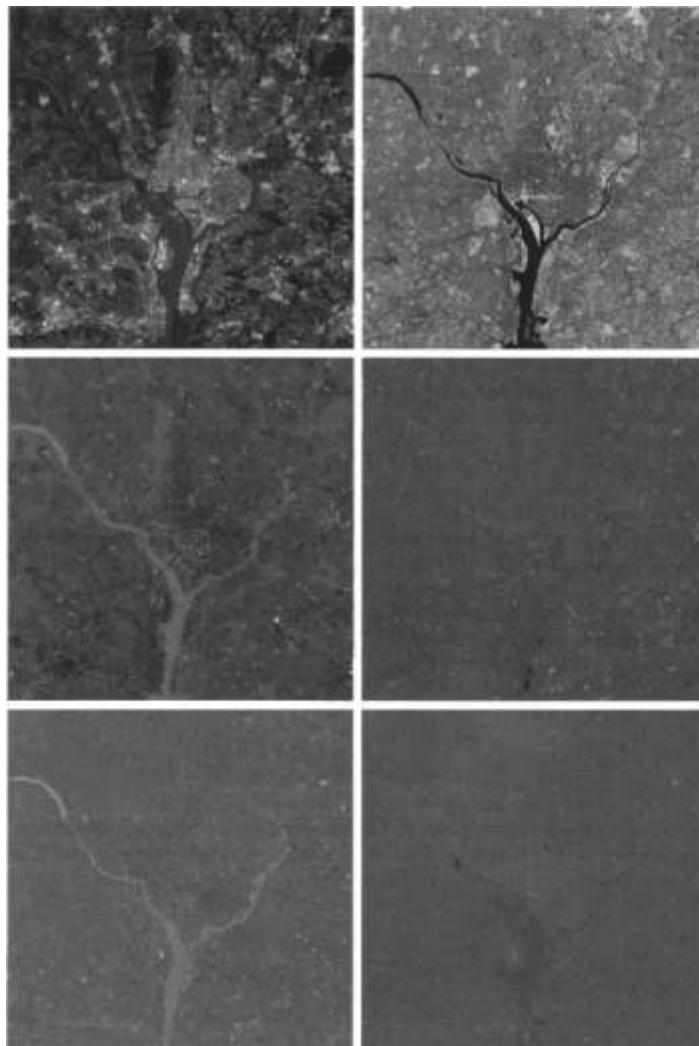
将6幅多光谱图像组织成堆叠图像

$S = \text{cat}(3, f1, f2, f3, f4, f5, f6);$

$X = \text{imstack2vectors}(S);$



主分量描述



■ 获得6幅主分量图像

$P = \text{principalcomps}(X, 6);$

■ 第一幅主分量图像生成并显示，其余5幅类似：

$g1 = P.Y(:, 1); \% \text{主成分分量}$

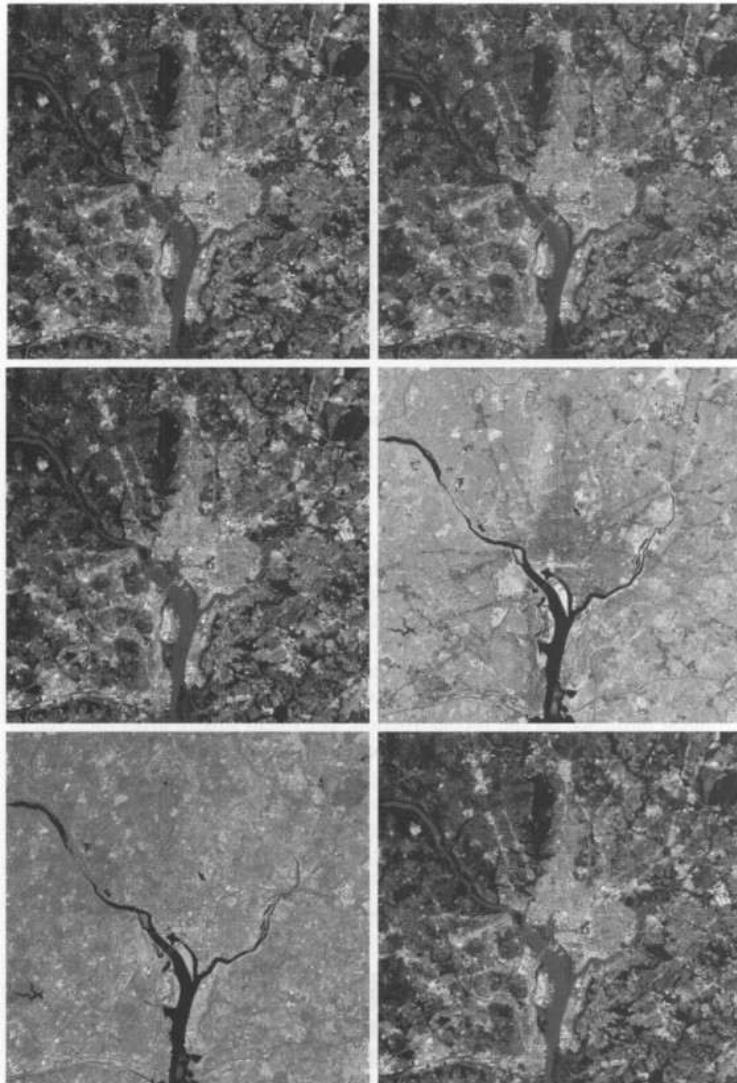
$g1 = \text{reshape}(g1, 512, 512);$

$\text{imshow}(g1, []);$

提取特征值（主对角线元素）：

$d = \text{diag}(P.Cy); \% \text{协方差矩阵}$

主分量描述



- 仅基于两幅主分量图像进行图像重建

$P = \text{principalcomps}(X, 2);$

- 重建第一幅图像，其余 5幅类似：

$h1 = P.X(:, 1); \% \text{从主成分重构的向量}$

$h1 = \text{mat2gray}(\text{reshape}(h1, 512, 512));$

THE END
Thank You.

