



数字图像处理

Digital Image Processing

信息工程学院

School of Information Engineering

8.5 区域分割

王昱 主讲

8.5 区域分割 (Region Segmentation)

- ◆ 阈值分割法由于没有或很少考虑空间关系，使得阈值分割方法受到限制。
- ◆ 基于区域的分割方法可以弥补这点不足，它利用的是图像的空间性质，该方法认为分割出来的属于同一区域的像素应具有相似的性质，其概念是相当直观的。
- ◆ 传统的区域分割算法有区域生长法和区域分裂合并法。该类方法在没有先验知识可以利用时，对含有复杂场景或自然景物等先验知识不足的图像进行分割，也可以取得较好的性能。但是，空间和时间开销都比较大。

基本公式

形式化地定义如下：令 I 表示图像, H 表示具有相同性质的谓词,图像分割把 I 分解成 n 个区域 $R_i, i = 1, 2, \dots, n$, 满足：

$$1) \bigcup_{i=1}^N R_i = I, R_i \cap R_j = \emptyset, \forall i, j, i \neq j$$

$$2) \forall i, i = 1, 2, \dots, n, H(R_i) = True$$

$$3) \forall i, j, i \neq j, H(R_i \cup R_j) = False$$

- 条件(1)表明分割区域要覆盖整个图像且各区域互不重叠；
- 条件(2)表明每个区域具有相同的性质；
- 条件(3)表明相邻的两个区域性质相异不能合并为一个区域。

1. 区域生长法

区域生长是一种根据事先定义的准则将像素或子区域聚合成更大区域的过程。

区域生长法主要考虑像素及其空间邻域像素之间的关系。

- ✓ 开始时确定一个或多个像素点作为种子。
- ✓ 然后按某种相似性准则增长区域，逐步生成具有某种均匀性的空间区域，将相邻的具有相似性质的像素或区域归并从而逐步增长区域，直至没有可以归并的点或其它小区域为止。

区域内像素的相似性度量可以包括平均灰度值、纹理、颜色等信息。

区域生长

区域增长示例

2	6	9
2	6	9
3	5	5

(a) 初始情形

2	6	9
2	6	9
3	5	5

(b) $T=1$

2	6	9
2	6	9
3	5	5

(c) $T=2.7$

2	6	9
2	6	9
3	5	5

(d) $T=5$

生长准则: 所考虑的像素点的灰度值和种子点区域的平均灰度值的差的绝对值小于或等于某个阈值 T , 就将该像素点归入种子点所在的区域.

区域生长

步骤

- 选择合适的种子点
- 确定相似性准则(生长准则)
- 确定生长停止条件

利用MATLAB函数`imreconstruct`和`bwlabel`进行区域生长的图像分割较简单，相关的例子可以在一些教材中找到。

我们这里给出一个不用这两个MATLAB函数，而是直接按照区域生长的思想对图像进行区域分割的例子。

```
I=imread('coins.png'); %读取图像
if isinteger(I)
I=im2double(I);      %将uint类型转换成double类型
end
figure,imshow(I);    %显示原图像
[M,N]=size(I);       %获取图像的大小
[y,x]=getpts;        %选取种子点
x1=round(x);         %横坐标取整
y1=round(y);         %纵坐标取整
seed=I(x1,y1);       %将种子点灰度值存入seed中
J=zeros(M,N);        %一个全零与原图像等大的图像矩阵J，作为输出图像矩阵
J(x1,y1)=1;          %将J中与所取点相对应位置的点设置为白
sum=seed;             %存储符合区域生长条件的点的灰度值的和
suit=1;              %存储符合区域生长条件的点的个数
count=1;             %记录每次判断一点周围八点符合条件的新点的数目
threshold=0.15;      %阈值，注意需要和double类型存储的图像相符合
```



```
I=imread('coins.png'); %读取图像
if isinteger(I)
I=im2double(I);      %将uint类型转换成double类型
end
figure,imshow(I);    %显示原图像
[M,N]=size(I);       %获取图像的大小
[y,x]=getpts;        %选取种子点
x1=round(x);         %横坐标取整
y1=round(y);         %纵坐标取整
seed=I(x1,y1);       %将种子点灰度值存入seed中
J=zeros(M,N);        %一个全零与原图像等大的图像矩阵J，作为输出图像矩阵
J(x1,y1)=1;          %将J中与所取点相对应位置的点设置为白
sum=seed;             %存储符合区域生长条件的点的灰度值的和
suit=1;              %存储符合区域生长条件的点的个数
count=1;             %记录每次判断一点周围八点符合条件的新点的数目
threshold=0.15;      %阈值，注意需要和double类型存储的图像相符合
```

区域生长

本例中，通过调用函数`getpts`，在最上面的那个硬币上选择一个种子点，如图8.22 (b) 所示，以该种子点按照程序中规定的生长规则进行区域生长分割，得到的分割结果如图8.22 (c) 所示。

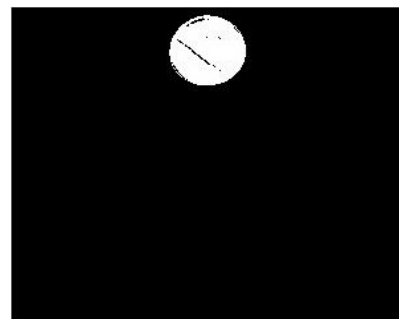
我们也可以在每一个图像块选择一个种子点，即选择多个种子点，生长得到一些互不连通的区域。



(a) 原图



(b) 在最上面那个硬币上
选择标识1个种子点



(c) 以种子点进行
区域生长结果

图8.22 区域增长示例

2. 区域分裂与合并

(1) 区域分裂法

如果区域的某些特性差别比较大,即不满足一致性准则时,则区域应该采用分裂法,分裂过程从从图像的最大区域开始,一般情况下,是从整幅图像开始。

注意：

- 确定分裂准则(一致性准则)
- 确定分裂方法,即如何分裂区域,使得分裂后的子区域的特性尽可能都满足一致性准则。

区域分裂

算法

- (1) 形成初始区域
- (2) 对图像的每一个区域 R_i , 计算 $P(R_i)$, 如果
$$P(R_i) = \text{FALSE}$$
则沿着某一合适的边界分裂区域
- (3) 重复步骤(2), 当没有区域需分裂时, 算法结束。

(2) 区域合并法

基本思想

单纯的区域分裂只能把图像分成许多满足一致性谓词的区域,相邻的具有相同性质的区域并没有合成一体。

- ✓ 合并运算就是把相邻的具有相似性质的区域合成为一个区域
- ✓ 合并算法中最重要的运算是确定两个区域的相似性
- ✓ 评判区域相似性方法有许多,可以基于区域的灰度值,也可以基于区域边界的强弱性等因素。一种简单的方法是比较它们的灰度均值

区域合并

算 法

(1) 使用某种方法进行图像的初始区域分割。

(2) 对于图像中相邻的区域，计算是否满足一致性谓词，若满足则合并为一个区域。

(3) 重复步骤(2)，直到没有区域可以合并，算法结束。

(3) 区域分裂合并

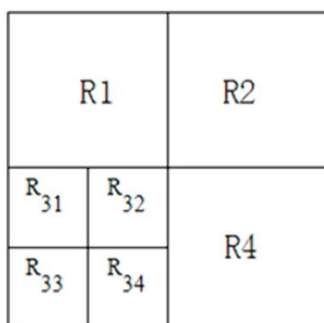
- ◆ 区域分裂合并法无需预先指定种子点,它按某种一致性准则分裂或者合并区域。
- ◆ 可以先进行分裂运算,然后再进行合并运算;也可以分裂和合并运算同时进行,经过连续的分裂和合并,最后得到图像的精确分割效果。
- ◆ 分裂合并法对分割复杂的场景图像比较有效。



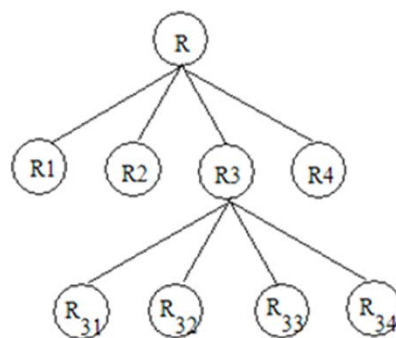
区域分裂合并

数据结构

具体实现时,分裂合并算法可以基于四叉树数据表示方式进行。



(a) 分裂图像

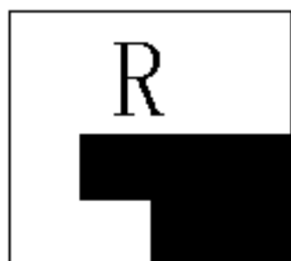


(b) 相应的四叉树结构

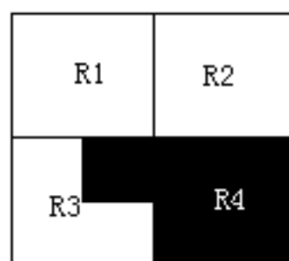
图8-30 图像分裂合并算法

区域分裂合并

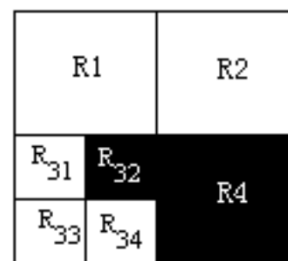
数据结构



(a) 初始图像



(b) 第一次分割



(c) 第二次分割

图 8.31 图像分裂示例

区域分裂合并

算法

(1) 设整幅图像为初始区域。

(2) 对每一区域 R ，如果 $P(R)=FALSE$ ，则把该区域分裂成四个子区域。

(3) 重复上一步，直到没有区域可以分裂。

(4) 对图像中任意两个相邻的 $R1$ 和 $R2$ ，如果 $P(R1 \cup R2)=TRUE$ ，则把这两个区域合并成一个区域。

(5) 重复上一步，直到没有相邻区域可以合并，算法结束。

区域分裂合并

实例



原始图像



分裂最小子
区域大小 16×16



分裂最小子
区域大小 8×8



分裂最小子
区域大小 4×4

图8.13 图像分裂合并



谢谢

THANK YOU