

## 实验六 图像分割实验

### 一、 实验目的

1. 了解图像分割中的基本概念以及相关知识;
2. 掌握几种经典边缘检测算子的基本原理、实现步骤;
3. 理解阈值分割、区域分割等基本原理、实现步骤;
4. 理解区域生长法的基本原理、实现方法。

### 二、 实验原理

图像分割,简单地说就是将一幅数字图像分割成不同的区域,在同一区域内具有在一定的准则下可认为是相同的性质,如灰度、颜色、纹理等,而任何相邻区域之间其性质具有明显的区别。图像分割是图像处理、模式识别和人工智能等多个领域中一个十分重要且十分困难的问题,是计算机视觉中首要的、重要的关键步骤。图像分割结果的好坏直接影响对计算机视觉中图像的理解。图像分割的主要内容包括:基于边界的图像分割,基于阈值的图像分割,基于区域的图像分割。

#### 1. 基于边缘的图像分割

边缘检测是检测图像特性发生变化的位置。不同的图像灰度不同,边界处会有明显的边缘,利用此特征可以分割图像。边缘检测分割法是通过检测出不同区域边界来进行分割的。边缘总是以强度突变的形式出现,可以定义为图像局部特性的不连续性,如灰度的突变和纹理结构的突变等。图像的边缘包含了物体形状的重要信息,它不仅在分析图像时大幅度地减小了要处理的信息量,而且还保护了目标的边界结构。边缘提取和分割时图像分割的经典研究课题之一,直到现在仍然在不断改进和发展。

边界存在于目标与背景、不同目标之间,是图像分割所依据的重要特征。基于边缘检测的图像分割首先确定图像中的边缘像素,然后将它们连接在一起构成边界。边缘是灰度值不连续的产物,可以利用差分来检测这种不连续性。

一阶差分算子通过寻找图像一阶差分中的最大值来检测边缘,将边缘定位在一阶差分最大的方向;二阶差分算子通过寻找图像二阶差分过零点来定位边缘,最常用的是拉普拉斯过零点。

梯度算子定义在二维一阶导数的基础上,由于数字图像是离散的,常用差分近似偏导数,在像素  $(x, y)$  处的梯度定义为:

$$\nabla f(x, y) = [G_x(x, y), G_y(x, y)]^T$$

式中,  $G_x(x, y)$  和  $G_y(x, y)$  分别表示  $x$  和  $y$  方向上的一阶差分。梯度幅度记为

$\nabla f(x, y)$ :

$$\nabla f(x, y) \approx |G_x(x, y)| + |G_y(x, y)|$$

增加最快的方向称为梯度方向,记为  $\alpha(x, y)$ :

$$\alpha(x, y) = \arctan \left[ \frac{G_y(x, y)}{G_x(x, y)} \right], \quad \alpha(x, y) \in \left( -\frac{\pi}{2}, \frac{\pi}{2} \right)$$

常用的一阶差分算子有 Roberts、Prewitt 和 Sobel 算子，用于近似计算偏导数。

二阶差分算子在图像分割中的主要作用为：二阶差分在边缘处产生零交叉，其过零点可以确定边缘的位置；二阶差分的符号可以确定边缘像素在边缘暗的一边还是亮的一边。

拉普拉斯算子是一个二阶差分算子，也是一个线性算子。4 邻域拉普拉斯算子的计算式为

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

8 邻域拉普拉斯算子的计算式为：

$$\nabla^2 f = 8z_5 - \sum_{i=1,8} z_i$$

高斯差分(DoG)算子也是一种常用的二阶差分算子，顾名思义，它是两个高斯函数之差的形式。将一幅图像  $f(x, y)$  分别与标准差为  $\sigma_1$ ,  $\sigma_2$  的高斯函数  $G_{\sigma_1}(x, y)$ 、 $G_{\sigma_2}(x, y)$  做卷积，可表示为：

$$g_1(x, y) = G_{\sigma_1}(x, y) * f(x, y)$$

$$g_2(x, y) = G_{\sigma_2}(x, y) * f(x, y)$$

高斯差分算子  $\nabla^2 G_{\sigma}(x, y)$  定义为：

$$\begin{aligned} \nabla^2 G_{\sigma}(x, y) &= G_{\sigma_1}(x, y) - G_{\sigma_2}(x, y) \\ &= \frac{1}{\sqrt{2\pi}} \left[ \frac{1}{\sigma_1} \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2}\right) - \frac{1}{\sigma_2} \exp\left(-\frac{x^2 + y^2}{2\sigma_2^2}\right) \right] \end{aligned}$$

通过 edge 边缘检测函数可以利用各种边缘检测算子检测图像边缘。

- **BW = edge(I)** 采用灰度或一个二值化图像 I 作为它的输入，并返回一个与 I 相同大小的二值化图像 BW，在函数检测到边缘的地方为 1，其他地方为 0。
  - **BW = edge(I, 'sobel')** 自动选择阈值用 Sobel 算子进行边缘检测。
  - **BW = edge(I, 'sobel', thresh)** 根据所指定的敏感度阈值 thresh，用 Sobel 算子进行边缘检测，它忽略了所有小于阈值的边缘。当 thresh 为空时，自动选择阈值。
  - **BW = edge(I, 'sobel', thresh, direction)** 根据所指定的敏感度阈值 thresh，在所指定的方向 direction 上，用 Sobel 算子进行边缘检测。Direction 可取的字符串值为 horizontal(水平方向)、vertical(垂直方向)或 both(两个方向)。
- [BW, thresh] = edge(I, 'sobel', ...) 返回阈值

●

## 2. 基于阈值的图像分割

阈值分割技术是比较经典和流行的图像分割方法之一，也是最简单的图像分割方法。这种方法的关键在于寻找适当的灰度阈值，通常是根据图像的灰度直方图来选取。它是用一个或几个阈值将图像的灰度级分为几个部分，认为属于同一个部分的像素是同一个物体。它不仅仅极大的压缩数据量，而且也大大简化了图

像信息的分析和处理步骤。阈值分割特别适合于目标和背景处于不同灰度级范围的图像。该方法的最大特点是计算简单，在重视运动效率的场合中，它得到了广泛的运用。

基于阈值的图像分割基本原理是通过设定不同的阈值，将图像中的像素分为两类或多类，具有计算简单、容易实现的优点。该方法适用于目标与背景在灰度上有较强对比度，且目标或背景的灰度比较单一的图像；当背景复杂时，该方法会失效。

当一幅图像由亮目标区域和暗背景区域（或反之）组成时，灰度直方图呈现双峰模式。显然，选择一个合适的阈值  $T$  就可以将这两个峰分开，这种方法称为阈值法。

阈值法又分为全局阈值法和局部阈值法，全局阈值法是指利用全局信息对整幅图像求出最优分割阈值，可以是单一或多个阈值；局部阈值法是将整幅图像划分为若干区域，再对各个区域使用全局阈值法分别求出最优分割阈值。

迭代阈值法的基本思想是属于同一类别的像素灰度值具有较大的一致性，使用均值和方差作为均匀性度量的数字指标。设一幅图像的像素总数为  $n$ ，灰度级数为  $L$ ，第  $k$  个灰度级  $r_k$  的像素数为  $n_k$ 。迭代阈值法的具体实现步骤如下：

- 1) 选取灰度均值或灰度中值作为阈值的初始估计值  $T_0$ ；
- 2) 根据像素的灰度值利用阈值  $T_k$  将图像分割成两个区域，记为  $\mathcal{R}_1$  和  $\mathcal{R}_2$ 。 $\mathcal{R}_1$  由灰度值在范围  $[0, T_k]$  内的像素组成， $\mathcal{R}_2$  由灰度值在范围  $[T_k + 1, L - 1]$  内的像素组成；
- 3) 分别计算区域  $\mathcal{R}_1$  和  $\mathcal{R}_2$  内像素的灰度均值  $\mu_1$  和  $\mu_2$ ，即

$$\mu_1(T_k) = \frac{1}{n_1(T_k)} \sum_{k=0}^{T_k} r_k n_k$$

$$\mu_2(T_k) = \frac{1}{n_2(T_k)} \sum_{k=T_k+1}^{L-1} r_k n_k$$

式中， $n_1(T_k) = \sum_{k=0}^{T_k} n_k$ 、 $n_2(T_k) = \sum_{k=T_k+1}^{L-1} n_k$  分别表示区域  $\mathcal{R}_1$  和  $\mathcal{R}_2$  内的像素数。

- 4) 利用  $\mu_1(T_k)$  和  $\mu_2(T_k)$  更新阈值，即

$$T_{k+1} = \frac{1}{2} [\mu_1(T_k) + \mu_2(T_k)]$$

- 5) 若连续两次迭代的阈值  $T_{k+1}$  与  $T_k$  之差小于预设限  $\epsilon$ ，即

$$|T_{k+1} - T_k| < \epsilon$$

终止迭代；否则，返回步骤 2。

需要使用到的 **matlab** 函数有：

- (1) **graythresh**: 利用 Otsu 算法（最大类间方差）获取全局阈值。
- (2) **im2bw**: 将灰度影像转换为二值图像。

3. 基于区域的图像分割

区域生长法的基本思想是：在待分割区域内确定一个或多个像素作为种子像素，根据某种相似性准则，由内向外合并具有相同或相似性质的相邻像素，逐步扩展区域，将扩展区域内的所有像素作为新的种子像素，继续合并具有相同或相似性质的相邻像素，直至扩展到整个区域。

区域生长法要求待分割区域具有相同或相似的性质，且是连通的。区域内像素的相似性度量通常利用灰度、颜色、纹理等特征。区域可分为 4 连通区域和 8 连通区域。区域标记是将一幅图像中的同一个连通分量标记为同一符号，因而也称为连通分量标记。根据区域可分为 4 连通和 8 连通区域，区域标记包括 4 连通和 8 连通区域标记。通常利用区域生长法提取二值图像中的所有连通分量，使用不同的符号标记不同的连通分量，输出为区域标记矩阵。

区域生长法的实现主要包括 3 个步骤：

- (1) 选取合适的种子像素；
- (2) 确定像素合并的相似性准则；
- (3) 确定终止生长过程的准则。

区域生长法从待分割区域内的种子像素出发，按照规定的顺序判断相邻像素是否满足相似性准则。区域生长法常用堆栈结构实现，种子像素的 4 连通区域生长法的实现步骤描述如下：

- 1) 初始化一个堆栈，将种子像素入栈；
- 2) 若堆栈非空，则栈顶像素出栈，标记该像素属于待分割区域；
- 3) 按照“左！ 上！ 右！ 下”的顺序查看其 4 邻域内的像素，若相邻像素满足相似性准则，则属于同一区域，将该像素入栈。
- 4) 重复上述过程直到堆栈为空。此时，标记了待分割区域内的所有像素。

对于种子像素的 8 连通区域生长，按照“左！ 左上！ 上！ 右上！ 右！ 右下！ 下！ 左下”的顺序查看其 8 邻域内的像素。4 连通区域生长法的问题是不能通过狭窄区域，因而有时不能填满区域；而 8 连通区域生长法的问题是当边界是 8 连通时会填出区域的边界。如图 1 所示，白色表示边界像素，黑色表示区域内部像素，p 表示种子像素，数字标记了填充的顺序。图 1(a)显示了 4 连通区域的生长过程，而图 1(b)显示了 8 连通区域的生长过程。

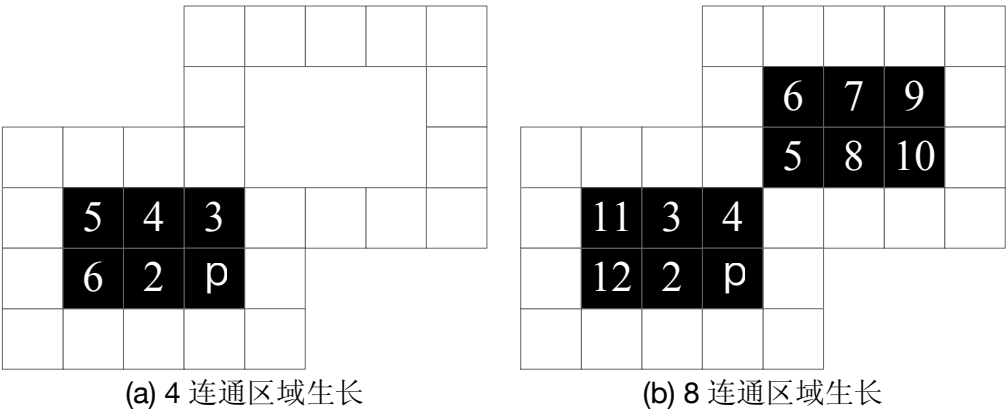


图 1 区域生长过程示意图

#### 4、霍夫变换

Hough 变换是 Paul Hough 于 1962 年提出，是图像处理技术中用于识别几何形状的一种常用方法，它实现了一种从图像空间到参数空间的映射关系。Hough 变换的基本原理是利用点线间的对偶性，将原始图像空间的给定曲线通过曲线表达形式变换成为参数空间的一个点。

MATLAB 中常用的关于 Hough 变换的函数有：hough、houghpeaks 和 houghline。

(5) hough: Hough 变换函数。

1.hough 函数

语法: [H, theta, rho] = hough(BW)

[H, theta, rho] = hough(BW, ParameterName, ParameterValue)

描述: [H, theta, rho] = hough(BW) 计算二值图像 BW 的标准 hough 变换。函数返回的 H 称为 hough 转换矩阵。theta 和 rho 是一维数组，theta 记录径，rho 记录角度。BW 可以是逻辑型，也可以是数值型。

(6) houghpeaks: Hough 变换峰值识别。

peaks = houghpeaks(H, NumPeaks)

peaks = houghpeaks(..., 'Threshold', val1, 'NHoodSize', val2)

其中，"..."指出来自默认语法和 peaks 的输入是持有峰值行和列坐标的 Q×2 大小的矩阵。Q 的范围是 0 到 NumPeaks，H 是霍夫变换矩阵。参数 val1 是非负的标量，指定了 H 中的什么值被考虑为峰值；val1 可以从 0 到 Inf 变化，默认值是 0.5\*max(H(:))。参数 val2 是奇整数的两元素矢量，指定量围绕峰值的邻域大小。

(7) houghline: 基于 Hough 变换提取线元。

lines = houghlines(f, theta, rho, peaks)

lines = houghlines(..., 'FillGap', val1, 'MinLength', val2)

其中，theta 和 rho 是来自函数 hough 的输出，peaks 是函数 houghpeaks 的输出。输出 lines 是结构数组（可能检测到多条直线），长度等于找到的线段数。结构中的每个元素可以看成一条线，并含有下列字段：

point1: 两元素向量[r1, c1]，指定了线段起点的行列坐标。

point2: 两元素向量[r2, c2]，指定了线段终点的行列坐标。

theta: 与线相关的霍夫变换的以度计量的角度。

rho: 与线相关的霍夫变换的ρ轴位置。

其他参数如下：

val1 是正的标量，指定了与相同的霍夫变换相关的两条线段的距离。当两条线段之间的距离小于指定的值时，函数 houghlines 把线段合并为一条线段(默认的距离是 20 个像素)。参数 val2 是正的标量，指定合并的线是保留还是丢弃。如果合并的线比 val2 指定的值短，就丢弃(默认值是 40)。

### 三、 实验内容与要求

1. 分别用 Roberts、Prewitt、Sobel（参考函数：BW1 = edge(I,'sobel')）

和 LOG 算子对灰度图像 `house.tif` 进行边缘检测，比较这几种算子的处理结果，并总结它们各自的优劣。

2. 使用 Hough 变换对 `circuit.tif` 图像进行直线检测，并显示最长的 10 条直线。

3. 对 `coins.png` 图像使用双峰阈值法行图像分割。实现方法：读取图像，显示其灰度直方图，观察直方图，选择两个波峰之间的波谷对应的灰度值作为最优阈值，通过 `im2bw()` 函数进行图像分割。

4. 对 `cell.tif` 灰度图像采用 Otsu 算法（`graythresh` 函数获取图像分割的最优阈值）和基本全局阈值法（迭代阈值法）进行图像分割，显示分割结果，并分析这两种方法各自的优劣。

5. 实现基于局部统计的可变阈值处理算法，并对图像 `chromosome.tif` 进行阈值分割。

#### 四、 思考题

请在实验报告最后进行简答：

- (1) 一阶和二阶边缘检测算子的异同点是什么？
- (2) LoG 算子相对于拉普拉斯算子的改进是什么？
- (3) 简答霍夫变换的步骤。

#### 五、 实验报告要求

请同学们完成上述实验：描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果和源代码，并进行必要的讨论，必须包括原始图像及其计算/处理后的图像以及相应的解释。