

# 图像测量实验报告

## 目录

实验一、图像平滑和图像增强.....	3
1 图像平滑.....	3
1.1 实验原理.....	3
1.2 实验内容.....	3
1.3 实验步骤.....	4
1.4 实验结果.....	4
2 图像增强.....	5
2.1 实验原理.....	5
2.2 实验内容.....	5
2.3 实验步骤.....	6
2.4 实验结果.....	6
实验二、图像的边缘检测.....	7
1 图像边缘检测.....	7
1.1 实验原理.....	7
1.2 实验内容.....	7
1.3 实验步骤.....	8
1.4 实验结果.....	8
实验三、图像的二值化和细化.....	10
1 图像二值化.....	10
1.1 实验原理.....	10
1.2 实验内容.....	11
1.3 实验步骤.....	12
1.4 实验结果.....	12
2 图像细化.....	13
2.1 实验原理.....	13
2.3 实验步骤.....	15
2.4 实验结果.....	16
实验四、图像测量应用实例——面积测量.....	17
1 面积测量.....	17
1.1 实验原理.....	17
1.2 实验内容.....	18
1.3 实验步骤.....	18
1.4 实验结果.....	19
实验五、图像测量应用实例——区域边界抽出和周长计算.....	20
1 区域边界抽出和周长计算.....	20
1.1 实验原理.....	20
1.1.1 区域边界抽出.....	20
1.1.2 区域边界周长.....	20
1.2 实验内容.....	20
1.3 实验步骤.....	21

1.4 实验结果.....	21
参考文献.....	23
代码链接.....	23

## 实验一、图像平滑和图像增强

**摘要:** 本实验包括图像平滑和图像增强, 分别利用空间邻域平均法、中值滤波法、最大值和最小值滤波四种方法实现对采集的灰度图像进行平滑处理, 利用空域中的图像锐化方法对图像进行锐化操作(其中 $\alpha$ 为可选择的用于控制锐化程度的因子), 以去除噪声并增强图像的轮廓边缘等细节信息。

### 1 图像平滑

#### 1.1 实验原理

图像的平滑处理就是为了消除图像中存在的噪声而对图像施加的一种处理。也称为图像的去噪声处理。

施行平滑化处理的方法大致分为两类:

频域法: 相应处理在频域进行。

其中主要的两种方法是空间邻域平均法和中值滤波法。

空域法: 相应处理在空域进行。

主要有理想低通滤波器和巴特沃思低通滤波器。

还有一种多图像平均法。

(1) 空间邻域平均法的一种简单实现: 在图像中以当前像素 $f(i, j)$ 为中心切出一个 $3 \times 3$ 像素组成的图像块, 如下图所示, 设当前像素 $f(i, j)$ 的灰度值为 $g(i, j)$ 时, 则:

$$g(i, j) = \frac{f(i, j) + f(i-1, j-1) + f(i, j-1) + f(i+1, j-1) + f(i-1, j) + f(i+1, j) + f(i-1, j+1) + f(i, j+1) + f(i+1, j+1)}{9}$$

(2) 中值滤波法的一种实现: 中值滤波是指在图像中以当前像素 $f(i, j)$ 为中心切出一个 $N \times M$  (例如 $3 \times 3$ )像素组成的图像块, 如上图所示的那样, 设当前像素 $f(i, j)$ 的灰度值为 $g(i, j)$ 时, 则 $g(i, j)$ 取 $N \times M$ 个像素灰度值中的中值。

中值(定义): 将一些数排序之后, 正中间的一个数(奇数个数字), 或者中间两个数的平均数。

最大(小)值滤波: 最大(小)值滤波是指在图像中以当前像素 $f(i, j)$ 为中心切出一个 $N \times M$  (例如 $3 \times 3$ )像素组成的图像块, 设当前像素 $f(i, j)$ 的灰度值为 $g(i, j)$ 时, 则 $g(i, j)$ 取 $N \times M$ 个像素灰度值中的最大(小)值。

#### 1.2 实验内容

图像平滑: 利用空间邻域平均法、中值滤波法、最大值和最小值滤波四种方法实现对采集的灰度图像进行平滑处理。

1.3 实验步骤

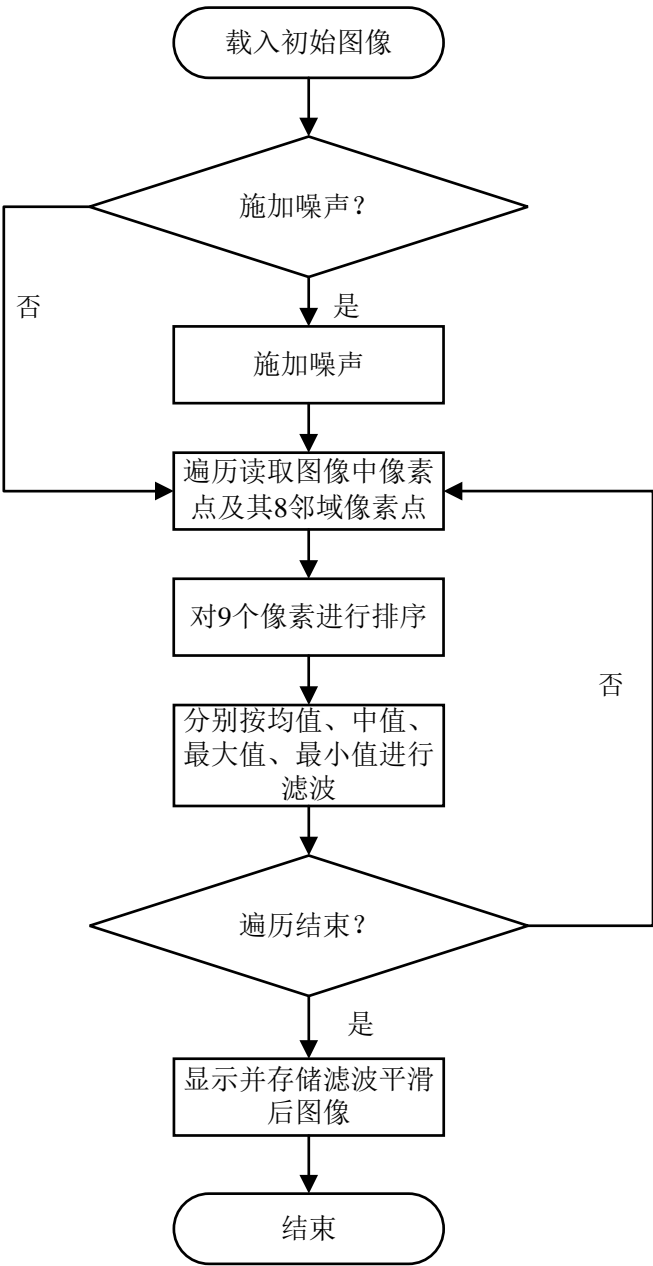


图 1-1 算法流程图

1.4 实验结果



图 1-2 原图



图 1-3 原图 (a) 空间邻域平均 (b) 中值滤波 (c) 最大值滤波 (d) 最小值滤波



图 1-4 加噪声后图



图 1-5 加噪声后 (a) 空间邻域平均 (b) 中值滤波 (c) 最大值滤波 (d) 最小值滤波

## 2 图像增强

### 2.1 实验原理

图像增强也称为图像锐化，其目的是为了加强图像中的轮廓边缘等细节信息。与图像平滑一样，也可分别在空域和频域进行图像增强。

在空域中进行图像锐化：将原图像与  $\alpha \Delta^2 f(i, j)$  相减可以得到经过锐化处理的图像  $g(i, j)$ ，即：

$$\begin{aligned} g(i, j) &= f(i, j) - \alpha \Delta^2 f(i, j) \\ &= f(i, j) - \alpha [f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j)] \\ &= f(i, j) + 4\alpha \left[ f(i, j) - \frac{1}{4} \sum \sum f(i, j) \right] \end{aligned}$$

这里， $\alpha$  为可选的用于控制锐化程度的因子，其中， $1/4 \sum \sum f(i, j)$  是被锐化处理点的周围 4 邻点之灰度的平均值。

原图像经过锐化处理后，灰度变化较平坦的区域，变化不大，而包括轮廓点在内的灰度变化较剧烈的区域，灰度差别加大，即图像细节得到增强。

也可以在频域进行图像的锐化处理。

### 2.2 实验内容

利用空域中的图像锐化方法对图像进行锐化操作，选择合适的控制锐化程度的因子。

## 2.3 实验步骤

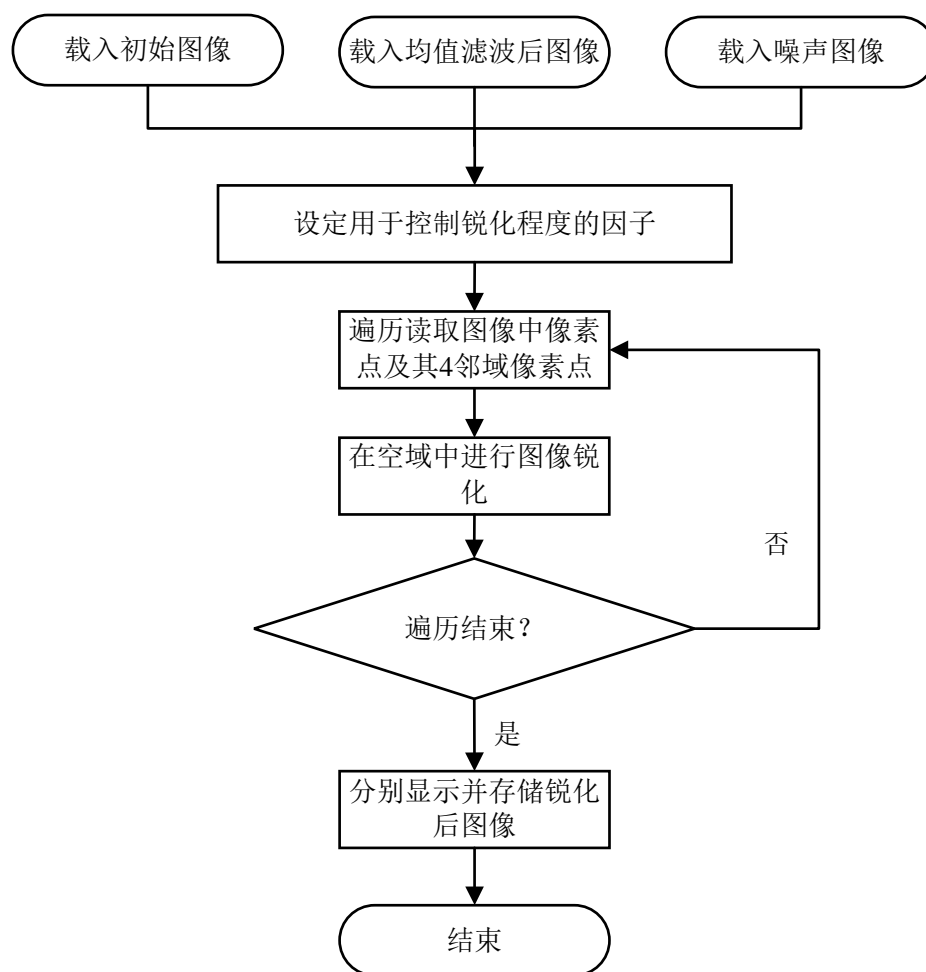


图 1-6 算法流程图

## 2.4 实验结果

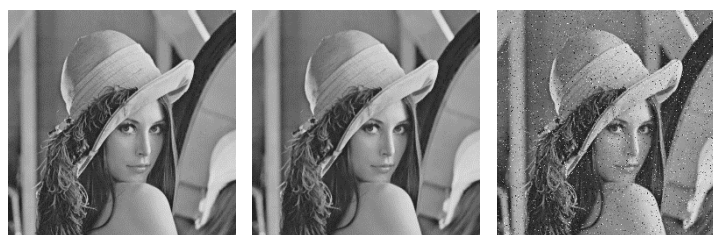


图 1-7 (a) 原图 (b) 原图中值滤波 (c) 加噪声后图



图 1-8 锐化后 (a) 原图 (b) 原图中值滤波 (c) 加噪声后图

## 实验二、图像的边缘检测

**摘要：**本实验即图像的边缘检测，分别使用一阶微分算子 `priwitt` 算子和 `smoothed` 算子对实验一中经过图像平滑和增强的灰度图像进行边缘检测。

### 1 图像边缘检测

#### 1.1 实验原理

在灰度图像的情况下，所谓的边缘检测可以看成是基于图像像素灰度值在空间的不连续性对图像做出的一种分割。边缘可以用方向和幅度两个特性来描述。一般而言，沿边缘走向方向其幅度值变化较平缓，而沿垂直于边缘走向其幅度值变化较剧烈。

边缘检测主要有以下几种方法：

- 1、一阶微分算子法
- 2、梯度算子法
- 3、二阶微分算子法
- 4、LOG 算子法

`smoothed` 算子是一个  $3 \times 3$  算子，说明如下：

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, D_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$D = (D_x^2 + D_y^2)^{\frac{1}{2}} \text{ 或 } D = \text{Abs}(D_x) + \text{Abs}(D_y)$$

$$D_x(i, j) = f(i-1, j+1) + f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - f(i, j-1) - f(i+1, j-1)$$

$$D_y(i, j) = f(i-1, j-1) + f(i-1, j) + f(i-1, j+1) - f(i+1, j-1) - f(i+1, j) - f(i+1, j+1)$$

#### 1.2 实验内容

对前次实验采集到的并经过图像平滑和增强的灰度图像分别使用 `priwitt` 算子和 `smoothed` 算子对图像进行边缘检测，给出经过边缘检测处理以后的图像。

### 1.3 实验步骤

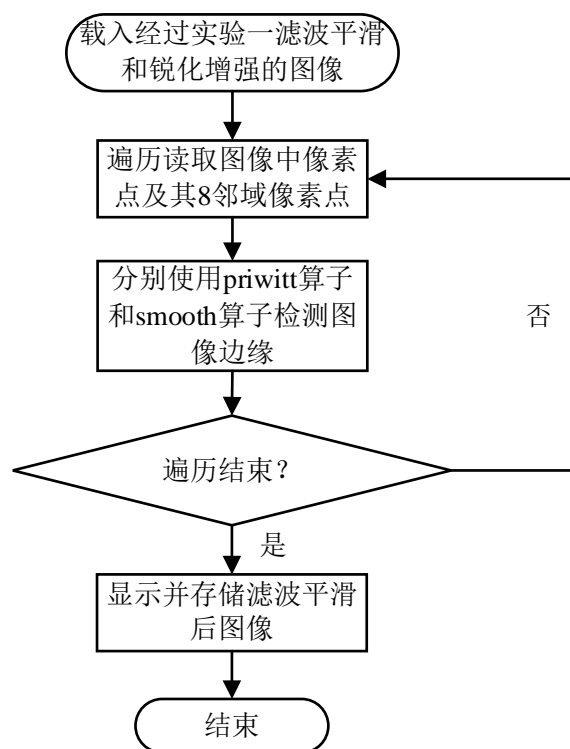


图 1-1 算法流程图

### 1.4 实验结果



图 1-2 经过实验一中值滤波图像平滑和增强的图像





图 1-3 (a) priwitt 算子 (b) smoothed 算子

## 实验三、图像的二值化和细化

**摘要：**本实验即图像的二值化和细化，主要是利用图像的灰度直方图以确定相应的分割阈值，从而将图像划分为物体和背景。在本实验中主要利用 S.Watanabe 方法选择阈值从而将图像二值化，并对已经二值化的图像使用 Deutsch 法进行细化操作。

### 1 图像二值化

#### 1.1 实验原理

图像二值化的目的是将图像一分为二，即将图像划分为物体和背景两个部分，利用图像的灰度直方图以确定相应的分割阈值是实现图像二值化的一个方法。该方法是基于如下的假设，即图像中的每一个区域都是由许多灰度级相似的像素所组成的，而物体和背景以及不同物体之间其灰度级则存在明显的差别。此时，图像的灰度直方图中将出现明显的峰值。这样，只要能根据直方图的形状适当选取灰度阈值，即可实现对图像的分割。

一般而言，图像的二值化的阈值处理方式如下：

$$\begin{aligned} f(i, j) &= 1; f(i, j) \geq t \\ f(i, j) &= 0; f(i, j) < t \end{aligned}$$

通常，用  $f(i, j)=1$  的部分表示图像，用  $f(i, j)=0$  的部分表示背景，确定  $t$  的方法称为阈值选择。下面介绍两种算法：

##### 1、类判别分析法

用该方法寻找到的阈值得到的二值图像，有较好的分离特性。

计算步骤：

- 1) 计算输入图像的灰度级直方图（用灰度级的概率函数  $Phs(i)$  来表示）。
- 2) 计算灰度均值(Ave)

$$Ave = \sum_{i=0}^{255} (i-1) Phs(i)$$

- 3) 计算灰度类均值(Aver(k))和类直方图和(W(k))

$$Aver(k) = \sum_0^K (i+1) Phs(i)$$

$$W(k) = \sum_{i=1}^K Phs(i)$$

- 4) 计算类分离指标  $Q(k) = [Ave \times W(k) - Aver(k)]^2 / [W(k) \times (1 - W(k))]$

- 5) 求使  $Q$  最大的  $k$

最佳阈值：  $T=k-1$ 。

##### 2、S.Watanabe 方法

阈值计算的第 2 种方法是根据 S.Watanabe 提出的方法以灰度直方图为依据，为图像二值化选择阈值。

设：  $X(i, j)$  为输入图像中点  $(i, j)$  的灰度级；

$Y(i+k_1, j+k_2)$  为点  $(i, j)$  的 8 邻域像素的灰度级；

$X(i, j)$  同  $Y(i+k_1, j+k_2)$  的关系如下图所示：

$Y(i-1, j-1)$	$Y(i-1, j)$	$Y(i-1, j+1)$
$Y(i, j-1)$	$X(i, j)$	$Y(i, j+1)$
$Y(i+1, j-1)$	$Y(i, j+1)$	$Y(i+1, j+1)$

则灰度级  $X$  的差定义为：

$$S(i, j, X) = \sum_{k_1=-1}^1 \sum_{k_2=-1}^1 (D(i+k_1, j+k_2))$$

其中：

当  $X(i, j) - Y(i+k_1, j+k_2) > 0$  时， $D(i, j) = X(i, j) - Y(i+k_1, j+k_2)$ ，

当  $X(i, j) - Y(i+k_1, j+k_2) < 0$  时， $D(i, j) = 0$ 。

将图像中的每个灰度级的所有  $S(i, j)$  分别叠加，可得到灰度差的直方图。即：

$$C(x) = \sum S(i, j, x)$$

$\Sigma$  表示所有灰度为  $X$  的像素的  $S$  取和。

对应于最大的  $C(x)$  的灰度级即为输出的阈值  $Y$ 。

如若  $D(i, j)$  的定义变为：

当  $X(i, j) - Y(i+k_1, j+k_2) < 0$  时， $D(i, j) = X(i, j) - Y(i+k_1, j+k_2)$ ，

当  $X(i, j) - Y(i+k_1, j+k_2) > 0$  时， $D(i, j) = 0$ 。

那么输出的阈值将变为  $Y = (Y(+)) - (Y(-)) / 2$ ，

其中  $Y(+)$  表示第一种定义，而  $Y(-)$  表示  $D(i, j)$  第二种定义。

计算图像的阈值的目的是要二值化图像，所谓二值化图像就是将原来的灰度图像转换成只有黑和白两种颜色的图像。对于大多数灰度图像来说，图像中的物体和背景是有明显的区别。通过选择阈值，区分图像和背景，以便对物体进行处理。

对一般的灰度图像来说，若像素的灰度值大于阈值则取 255，否则就取 0。

## 1.2 实验内容

利用上面介绍的算法对采集并经过处理的图像进行二值化。

1.3 实验步骤

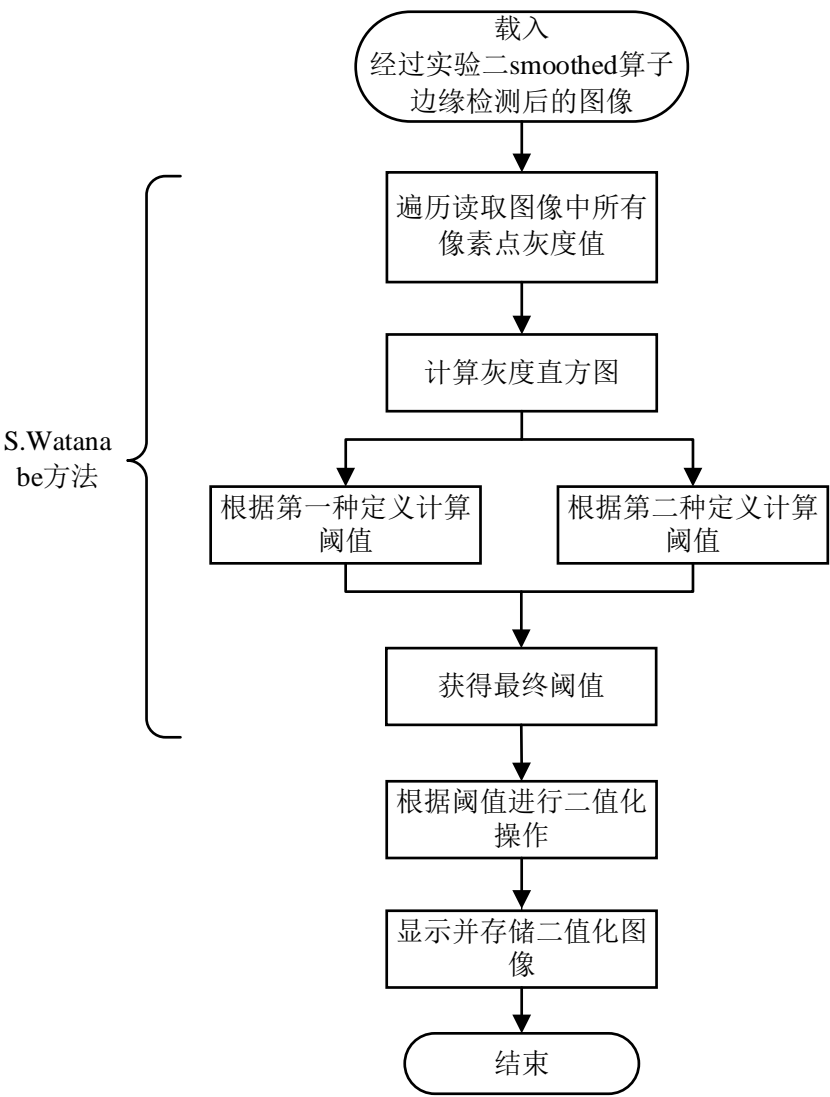


图 1-1 算法流程图

1.4 实验结果



图 1-2 经过实验二 smoothed 算子边缘检测后的图像



图 1-3 二值化后图像

## 2 图像细化

### 2.1 实验原理

许多种形状，特别是细长的形状，可以用沿着目标肢体中轴的细化连线表示。单位宽度，位于目标中轴上的连线，称为目标的骨架。细化过程与骨架有密切关系。可以暂时把细化过程定义为：一个连续剥离形体最外层元素直到获得单位宽度的连通线(骨架)的过程，因此细化算法就是重复地剥掉二值图像的边界像素，特别是  $0 \rightarrow 1$  在变化处的像素的算法，但是剥离边界像素时必须保持目标的连通性，因而。细化算法要满足下面的条件：

- 1、在每次重复过程中，要保持连通性。
- 2、不能减小形状的长度。

**Deutsch 细化算法：**使用 Deutsch 法对值为 1 和 0 的二值图像进行细化处理。其中 1 为待细化的图像，0 为背景。算法如下：

$f(4)$	$f(3)$	$f(2)$
$f(5)$	$f(0)$	$f(1)$
$f(6)$	$f(7)$	$f(8)$

如上图所示，设  $f(0)$  为输入图像的任意像素， $f(k)$  为  $f(0)$  的 8 邻域像素灰度值，位置如图所示。定义  $f(0)$  的交叉数为

$$X = \sum_{k=1}^8 |f(k+1) - f(k)|$$

如果下述条件满足，则点将被删去：

- (1) 当  $X = 0$  或  $X = 2$  或  $X = 4$  时， $f(9) \equiv f(1)$ ；
  - (2)  $\sum_{k=1}^8 f(k) \neq 1$ ；
  - (3)  $f(1) \wedge f(3) \wedge f(5) = 0$  且  $f(1) \wedge f(3) \wedge f(7) = 0$ ；
  - (4) 若  $X = 4$  则应满足以下条件之一：
    - (a)  $f(1) \wedge f(7) = 1$  且  $f(2) \vee f(6) = 1$  且  $f(3) \vee f(4) \vee f(5) \vee f(8) = 0$
    - (b)  $f(1) \wedge f(3) = 1$  且  $f(4) \vee f(8) = 1$  且  $f(2) \vee f(5) \vee f(6) \vee f(7) = 0$
- 式中，“ $\wedge$ ”表示二进制“逻辑与”运算，“ $\vee$ ”表示二进制“逻辑或”运算。

为了对称地进行细化，在根据（1）（2）（3）（4）遍历后，应再根据（1）（2）（5）（6）进行遍历。

（5）  $f(3) \wedge f(5) \wedge f(7) = 0$  且  $f(5) \wedge f(7) \wedge f(1) = 0$ ；

（6） 若  $X = 4$  则应满足以下条件之一：

（a）  $f(5) \wedge f(3) = 1$  且  $f(6) \vee f(2) = 1$  且  $f(1) \vee f(4) \vee f(7) \vee f(8) = 0$

（b）  $f(7) \wedge f(5) = 1$  且  $f(8) \vee f(4) = 1$  且  $f(1) \vee f(6) \vee f(3) \vee f(2) = 0$

## 2.2 实验内容

利用上面介绍的方法对进行过边缘检测而且已经二值化的图像进行细化操作。

## 2.3 实验步骤

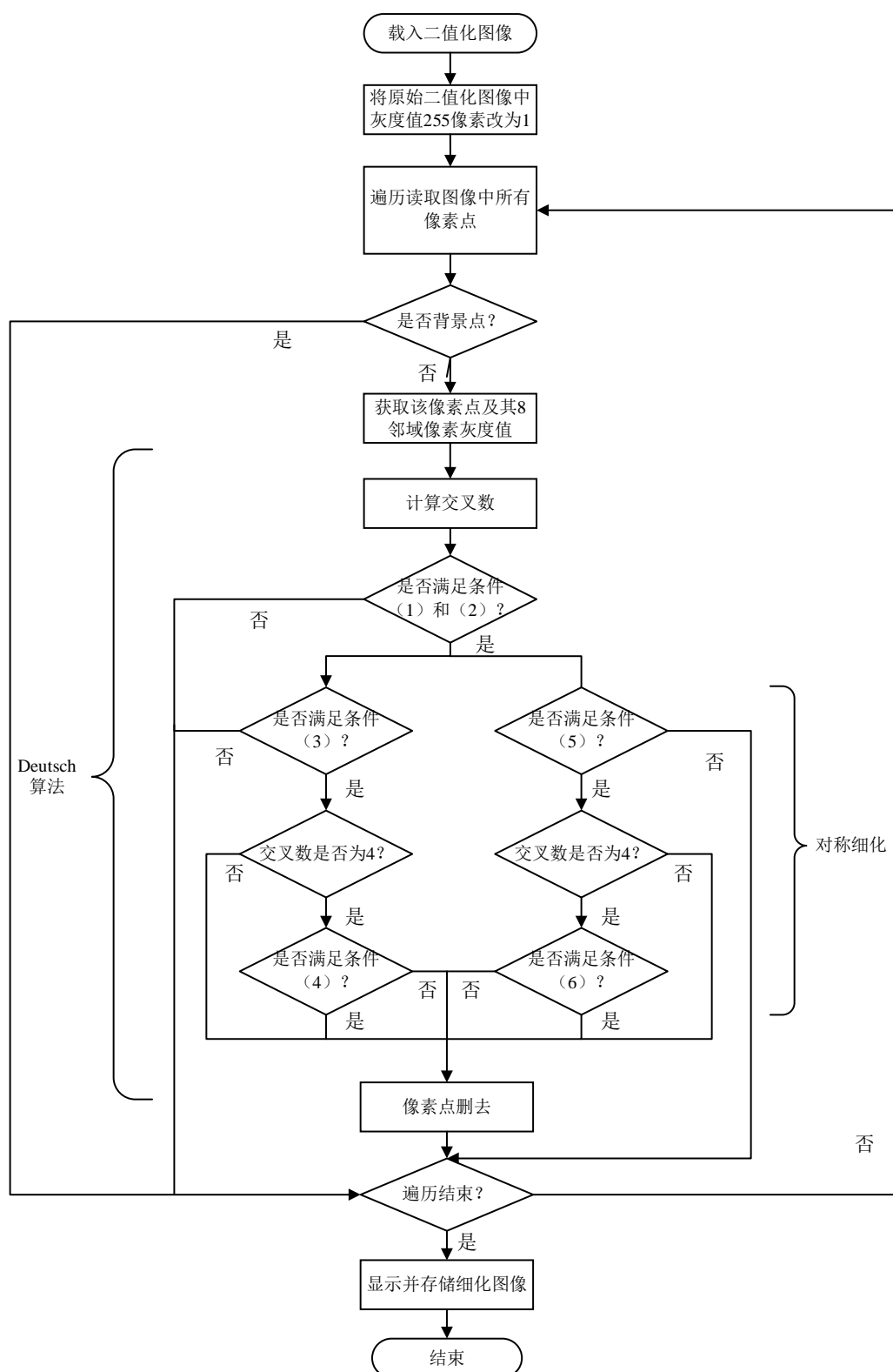


图 1-4 算法流程图

## 2.4 实验结果



图 1-5 细化后图像



## 实验四、图像测量应用实例——面积测量

**摘要：**本实验即图像测量应用实例之一的面积测量，主要是利用标号法计算图像中的局部面积。本实验中采用扫描标号法，利用 8 连通意义下的贴标号算法对图像中的各个图形贴标号，从而进行面积测量。

### 1 面积测量

#### 1.1 实验原理

面积测量是图像测量的基本操作之一，利用面积测量可以计算图像的局部面积，对于图像的参数测量具有重要的意义。

一般采用标号法进行面积计算。所谓标号法，就是图像中不同的物体都有唯一识别的号数。在同一物体中，所有的像素点的标号都是一致的，而不同物体之间是完全不同的。在图像面积测量中，首先必须对图形进行分割。为了区分互不连通的图形以便分别计算其面积，需要对图形进行标号操作。

对不同图像进行标号操作的方法有很多种，这里介绍两种最基本的方法。

##### 1) 扫描标号法

扫描标号法的算法如下：

- 首先进行从左到右，从上到下扫描。在同一行中不连通的行程（灰度级相同的点）标不同的号，不同的列标不同的号。
- 从左上到右下扫描，如果两个相邻的行中右相连通的行程则下行的号改为上行的号。
- 从右下的左上扫描，如果两个相邻的行中有相连通的行程则上行的号改为下行的号。
- 再对标过的号进行排列。

通过对图像进行标号后，可对不同的物体的面积进行测量。主要是对相同号的点进行累加，之后得到物体的像素点的总和。总和再乘以系数将得到图像中物体的相近的面积。如果对总和的大小进行限定，还可以祛除图像中的粒子。

##### 2) 递归搜索法

利用递归搜索法可以区分图像上不同的物体，特别是对于一些形状特殊的物体效果明显。假设物体的颜色值为黑（即灰度值为 0），背景的颜色值为白（即灰度值为 255），并且设置同图像数组对应的标号数组 HAO()（初始值都为 0），号的初值为 0。其算法如下：

- 如果图像点的值为 0（即是物体的点）而且其对应的标号数组的值为 0（即未标过号），那么号的值增 1 且调用标号递归子程序。
- 递归子程序。

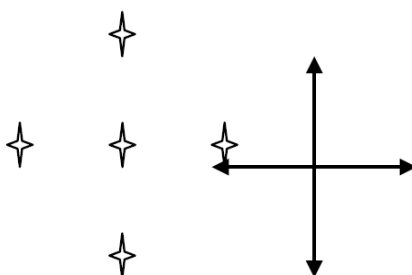


图 1-1 示意图

如上图所示，中间的点沿着四个箭头的方向向左邻域点、向右邻域点、向上邻域点和向下邻域点递归。递归的出口示图像点的值为 255（即背景的点）或者图像点对应的标号叔祖的值非 0（即标过号）。

说明：这种方法的缺点是当测量面积太大时，系统的堆栈会发生溢出，所以只能适应小的图像。

## 1.2 实验内容

利用上述的算法之一，对前景为几个简单几何图形的二值图像进行面积测量。要求分别给出每个图形的面积。

## 1.3 实验步骤

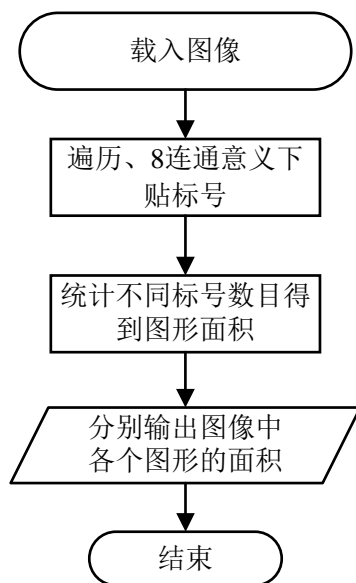


图 1-2 算法流程图

## 1.4 实验结果

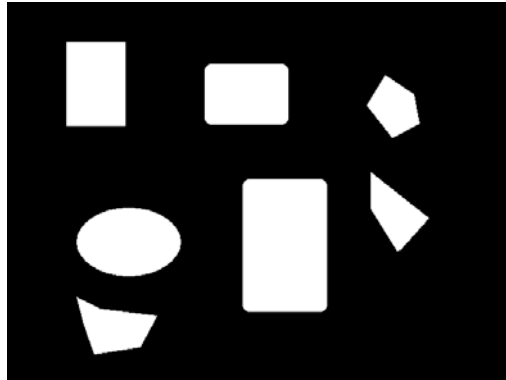


图 1-3 测试图

测量结果:

area0 253263

area1 8025

area2 8094

area3 3180

area4 3584

area5 17908

area6 8956

area7 4190

## 实验五、图像测量应用实例——区域边界抽出和周长计算

**摘要：**本实验即应用实例之一的区域边界抽出和周长计算，分别利用四邻域法和八邻域法对二值化图像进行边界抽出，并分别进行边界周长的计算。

### 1 区域边界抽出和周长计算

#### 1.1 实验原理

抽出图像区域边沿并测量某些图像边沿的长度时图像测量中常遇到的问题。本节介绍区域边界抽出和周长计算的方法。

##### 1.1.1 区域边界抽出

边界抽出有三种简单的方法：

##### 1) 四邻域法

对目标像素和它的四邻域像素进行探测，若值都相等，则对目标像素给以标记，然后清除。

##### 2) 八邻域法

对目标像素和它的八邻域像素进行探测，若值都相等，则对目标像素给以标记，然后清除。

##### 3) 全图检测法

对整幅图像从左到右、从上到下进行检测，若相邻两点的像素值为1和0时，则对值为1的点进行标记后保留，其余的点清除。

##### 1.1.2 区域边界周长

对于二值图像（记黑为1，白为0）中周长的测量，通常的方法是对图像的边缘像素作出标记，然后累计所标记的像素个数，所得就是图像的周长。对边缘像素作标记可用下面两种算法：

方法1：

##### 1) 定义一个二维数组 $a(i, j)$ ；

2) 对整幅图像从上到下扫描，比较相邻两点的值，如为1和0，则记  $a(i, j)=1$ ， $(i, j)$  为值为1的点的坐标；

3) 对整幅图像从左到右扫描，比较相邻两点的值，如为1和0，也记  $a(i, j)=1$ ， $(i, j)$  为值为1的点的坐标；

4) 累计整幅图像中  $a(i, j)=1$  的像素即为周长。

方法2：

##### 1) 定义一个二维数组 $a(i, j)$ 。

2) 对整幅图像从上到下扫描，比较相邻两点的值，如为1和0，则记  $a(i, j)=1$ ， $(i, j)$  为值为1的点的坐标；

3) 对整幅图像从左上到右下扫描，比较相邻两点的值，如为1和0，也记  $a(i, j)=1$ ， $(i, j)$  为值为1的点的坐标；

4) 对整幅图像从右下到左上扫描，比较相邻两点的值，如为1和0，也记  $a(i, j)=1$ ， $(i, j)$  为值为1的点的坐标；

5) 累计整幅图像中  $a(i, j)=1$  的像素，即为周长。

#### 1.2 实验内容

利用上述的算法之一，对前景为几个简单几何图形的二值图像进行区域边界

抽出和周长计算。要求给出区域边界抽出以后的图像，并给出区域的周长。

1.3 实验步骤

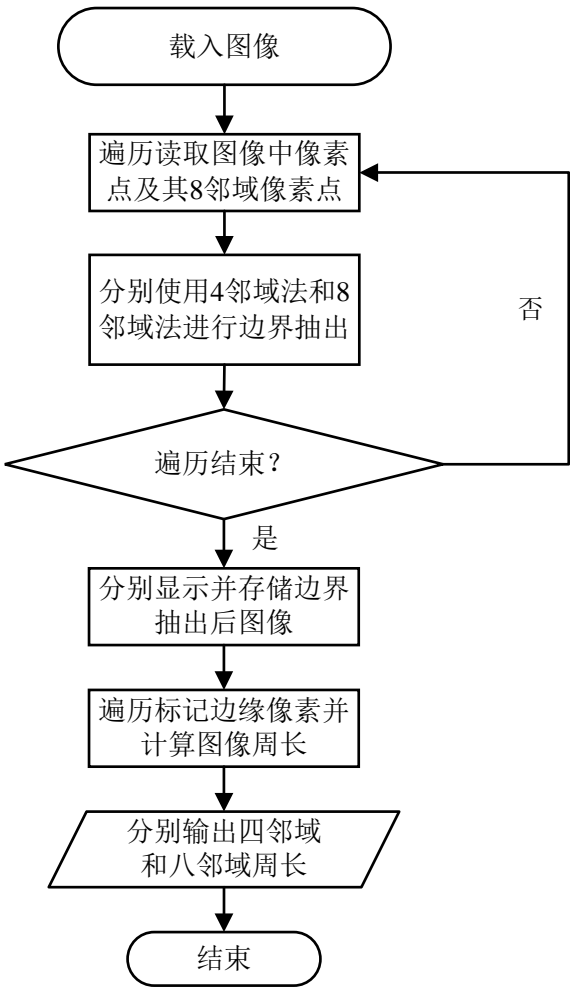


图 1-1 算法流程图

1.4 实验结果

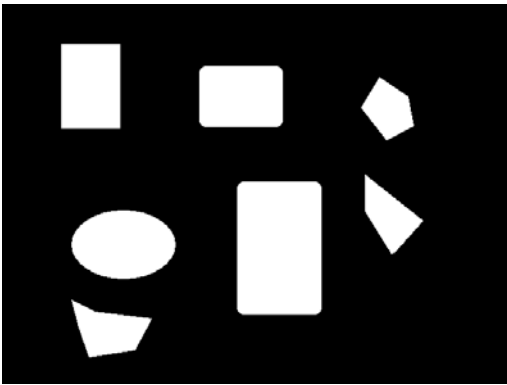


图 1-2 测试图

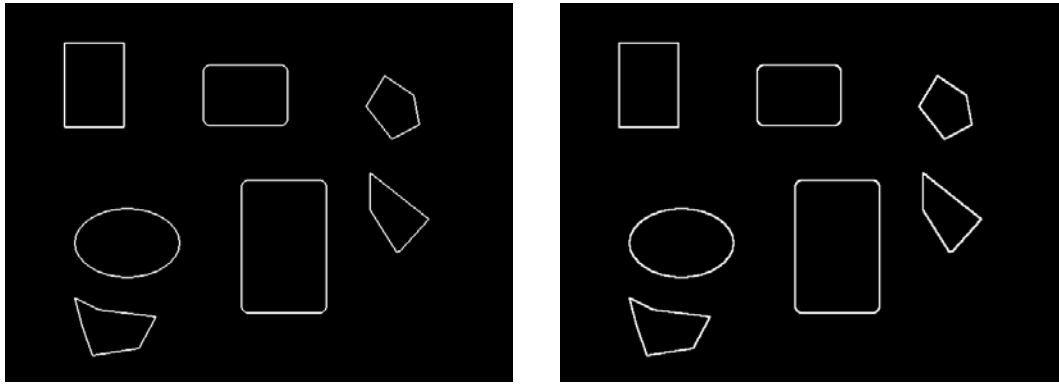


图 1-3 (a) 四邻域法边界抽出 (b) 八邻域法边界抽出

周长测量结果:

four\_neighborhoods: 4456

eight\_neighborhoods: 5412

## 参考文献

- [1] 中国科学技术大学自动化系. 图像测量实验指导书[Z]. 2018.
- [2] Szeliski R. Computer vision: algorithms and applications[M]. Springer Science & Business Media, 2010.
- [3] Forsyth D A, Ponce J. Computer vision: a modern approach[M]. Prentice Hall Professional Technical Reference, 2002.
- [4] Wilson J N, Ritter G X. Handbook of computer vision algorithms in image algebra[M]. CRC press, 2000.

## 代码链接