

实验三、图像分割及目标检测

时间： AM 10:00-12:00，2018.11.28(星期三)

地点：信软学院西 303，沙河校区

一. 实验目的

- 1、了解图像边缘检测及图像区域分割的目的、意义和手段。
- 2、熟悉各种经典的边缘检测算子、图像分割方法及其基本原理。
- 3、熟悉各种图像特征表示与描述的方法及基本原理。
- 4、熟练掌握利用 MATLAB 工具实现各种边缘检测的代码实现。
- 5、熟练掌握利用 MATLAB 工具实现基本阈值分割的代码实现。
- 6、通过编程和仿真实验，进一步理解图像边缘检测、图像分割及其在目标检测、目标识别及跟踪测量应用中的重要性。

二. 实验内容及原理

1、利用 Sobel 算子进行图像的边缘检测

(1) 实验原理与步骤

数字图像的边缘一般利用一阶/二阶差分算子进行检测。常用的差分算子包括：Roberts 算子（对角算子），Prewitt 算子（一阶），Sobel 算子（一阶），Laplacian 算子（二阶），LoG 算子（二阶）及 Canny 边缘检测算法等。其中，Sobel 算子为常见的一类梯度算子（一阶梯度算子）。其 x, y 方向的梯度算子分别为：

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$

如图 1 所示，一幅数字图像 I 与 S_x 和 S_y 分别做滤波（相关）运算后（可选用多种方式，如 `conv2`, `filter2` 及 `imfilter` 等 MATLAB 函数），可以求得 x, y 两个方向的梯度图像 D_x, D_y ，然后，可以计算得到原图像的梯度幅度，即

$$D = \sqrt{D_x^2 + D_y^2} \quad (2)$$

(2) 进一步执行梯度图像 D 的二值化处理（建议采用 Otsu 阈值，也可考虑其他阈值分割），检测图像的二值化边缘。

(3) 对于与步骤同样的输入图像 I ，利用 MATLAB 工具的 `edge(I,'sobel')` 函数进行处理。试比较处理结果与步骤 (2) 的得到的结果的差异，并分析检测结果存在一定差异的原因。

(4) 画出原图像、原图像的 D_x , D_y , D 图，及最终的边缘检测结果图（即二值化边缘）。



图 1 原始图像（256×256 Lena）

2、利用梯度改进全局阈值分割

实验原理、内容及步骤如下：

① 对图 2 中的目标图像利用图像梯度公式，得到梯度幅度图像；

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y)$$

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)$$

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

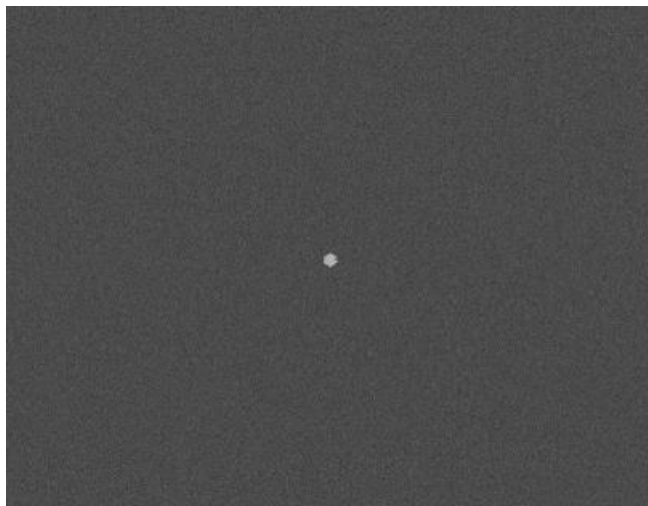


图 2 原始图像 (814×651)

- ② 将梯度图像和原图像进行乘积，得到新的图像；
- ③ 统计非零像素的直方图
- ④ 以统计之后的直方图为基础，利用 **OTSU** 阈值处理方法分割乘积后的图像得到最终二值化结果图像。

3、数字图像中目标区域测量及计算

实验原理、内容及步骤如下：

- ⑤ 对图 2 中的黑色形状目标进行阈值分割，得到二值化的图像；

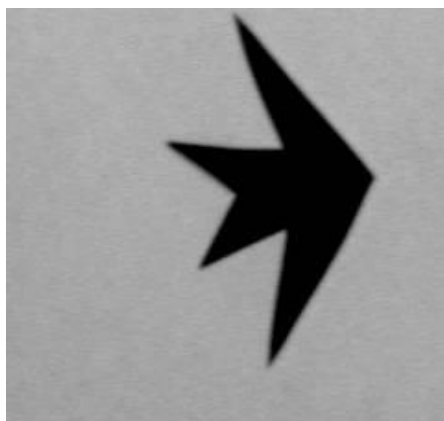


图 3 原始图像 (240×240)

- ⑥ 计算目标形状的面积（以像素单位表示）；
- ⑦ 计算图中黑色形状目标的形心位置，并在原图上进行位置标记（可以计算位置为圆心，以一定半径 r 画一个红色小圆圈）。

$$\begin{cases} x_c = \frac{1}{A} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} xf(x, y) \\ y_c = \frac{1}{A} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} yf(x, y) \end{cases} \quad (3)$$

其中，A 为目标区面积， $f(x, y)$ 在背景区域值为 0，目标区域值为 1。

。

三、实验报告及要求

1、原始音频及图像数据可在课程网站“实验指导”中下载，也可根据需要自行选择合理的待处理图像。

2、简述实验原理。

3、根据各个实验内容分别叙述其实验步骤、程序设计流程图（建议用 Visio 软件），并对实验结果进行必要的分析和总结。

4、要求提交 MATLAB 源代码，并注意程序代码书写的规范性；涉及实验核心内容的代码需要自己编写，不允许直接调用 MATLAB 库函数。

5、严格按电子科技大学《实验报告》的相关要求，撰写实验报告，并按时提交纸质版实验报告。实验报告中还需要回答和完成以下的思考题。

四、思考题

1、利用梯度算子与图像进行滤波（相关）运算后，为什么还需要给定阈值进行二值化处理？

2、Laplacian算子检测边缘为什么会产生双边效果？为什么不能检测出边的方向。

3、相对其他边缘检测算子，Canny边缘检测算法的主要优势体现在哪里？

附：必要的提示

1、边缘检测 MATLAB 函数：

% Sobel 边缘检测函数。

BW = edge(I,'sobel')

BW = edge(I,'sobel',thresh)

BW = edge(I,'sobel',thresh,direction)

```
[BW,thresh] = edge(I,'sobel',...)
```

```
% Prewitt 边缘检测函数。
```

```
BW = edge(I,'prewitt')
BW = edge(I,'prewitt',thresh)
BW = edge(I,'prewitt',thresh,direction)
[BW,thresh] = edge(I,'prewitt',...)
```

```
% Roberts 边缘检测函数。
```

```
BW = edge(I,'roberts')
BW = edge(I,'roberts',thresh)
[BW,thresh] = edge(I,'roberts',...)
BW = edge(I,'log')
BW = edge(I,'log',thresh)
BW = edge(I,'log',thresh,sigma)
[BW,threshold] = edge(I,'log',...)
```

```
% 零交叉边缘检测函数。
```

```
BW = edge(I,'zerocross',thresh,h)
[BW,thresh] = edge(I,'zerocross',...)
```

```
% Canny 算子边缘检测函数。
```

```
BW = edge(I,'canny')
BW = edge(I,'canny',thresh)
BW = edge(I,'canny',thresh,sigma)
[BW, threshold] = edge(I,'canny',...)
```

2、图像阈值分割

```
BW= im2bw(I, thresh);      % 设定阈值分割
level = graythresh(I);     % OTSU 自动阈值分割
```

3、图像区域测量

```
imfill(Ibw,'holes');       % 空洞填充
regionprops(Ilabel,'centroid'); % 形心计算
```

4、指定位置画矩形、圆形标记

```
rectangle('Position',[x,y,w,h]) % 给定起点[x,y] 矩形宽 w 高 h, create 2-D rectangle object
                                   % 该函数功能可创矩形、圆形、椭圆等标记。
[lat,lon] = scircle1(lat0,lon0,rad) % 在指定位置生成一个小圆形。
```

注：以上各个函数的参数说明及具体用法，可查阅相关的 MATLAB 帮助文件。获取“XXX()”函数的用法，可在命令窗口（Command Window）键入：“doc XXX”获得详细说明。