

## 数字媒体技术基础作业 2

15331416 赵寒旭

### 1. 实验环境

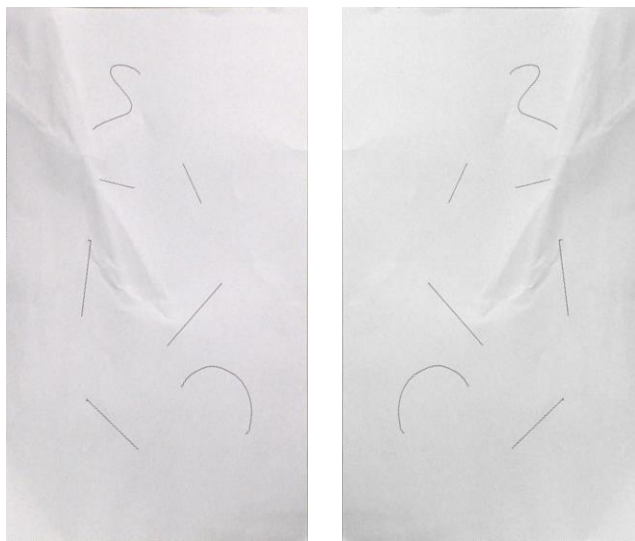
软件：Matlab R2015b

运行系统：windows10

### 2. 实验步骤（结合代码说明）

#### 2.1 通过阈值进行前后景分割（将线条和 A4 纸分开）

##### 1) 将图像转成灰度图

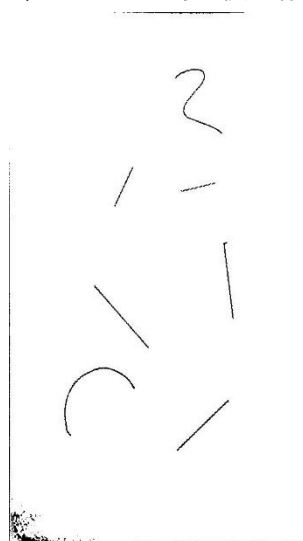


左图为原图（作业 1 输出图像），右图为转换后灰度图像。

实现方法：

```
gray_img = rgb2gray(input_img);
```

##### 2) 通过设定阈值使图像转为二值图



为突出纸上线条的轮廓，设定阈值 0.75，将线条与大部分背景区分开，二值化使灰度值两极化，易于后续轮廓检测工作进行。

可以看到左下角和纸张边缘仍有无法直接去除的部分，但纸张背景大部分已经归为 0 值。

实现方法：

```
thresh_result = im2bw(gray_img, 0.75);
```

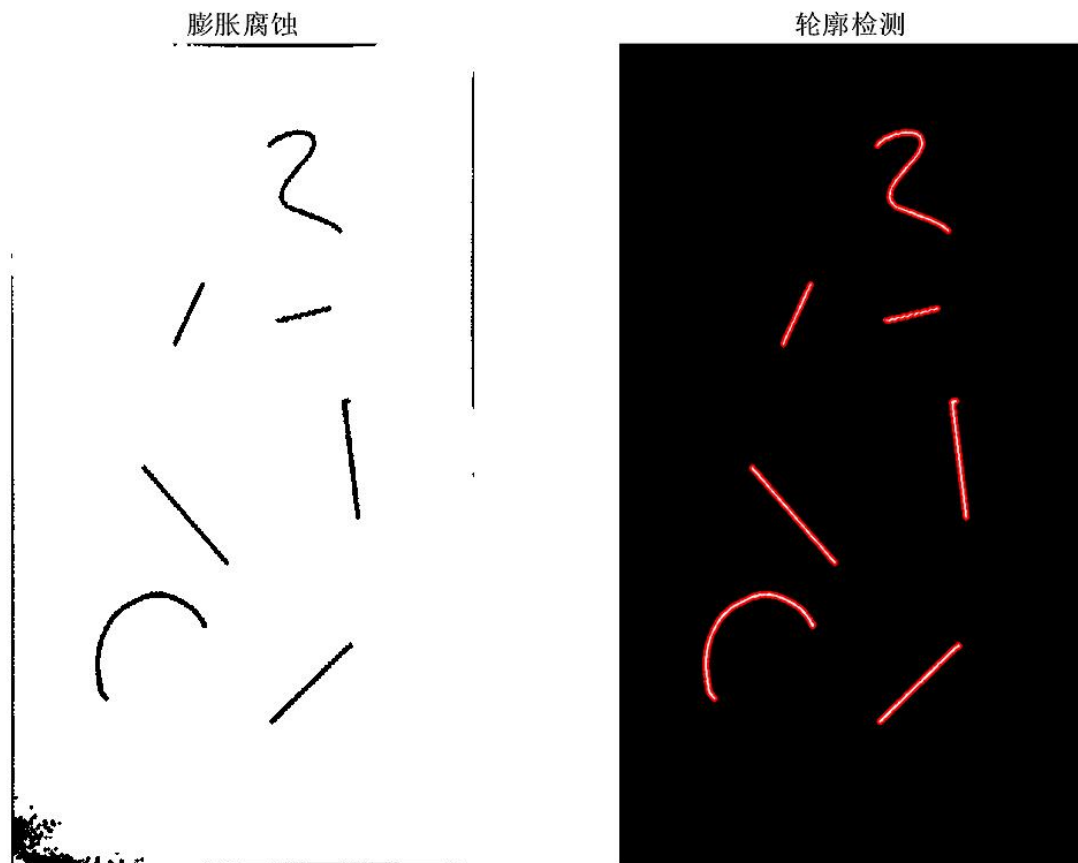
详解：im2bw 使用阈值 (threshold) 变换法把灰度图像 (grayscale image) 转换成二值图像。

```
BW = im2bw(I, level)
```

输出图像 BW 将输入图像 I 中亮度值大于 level 的像素值替换为值 1（白色），其他替换为值 0（黑色）。

经过反复修改测试发现使用 0.75 作为阈值效果较好。

## 2.2 轮廓检测



- 1) 形态学图像处理（膨胀腐蚀）突出待提取线条

```
B = [ 0 1 0
      1 1 1
      0 1 0];
% 第一次腐蚀
output_img = imerode(input_img, B);
% 第二次腐蚀
output_img = imerode(output_img, B);
```

用矩阵 B 进行两次腐蚀，使待提取部分明显突出。  
(右图为原二值图和腐蚀操作后结果对比)



- 2) 二值图取反，方便连通区域选择

```
% 为方便连通区域选择，二值图取反
imdilate_result = ~imdilate_result;
```

- 3) 获取连通区域并显示

- (1) 边界对象抑制

```
% imclearborder:边界对象抑制
IM2 = imclearborder(imdilate_result, 8);
```

由二值图可知边界有一定区域灰度值和待提取线条相同，为避免对连通区域提取造成干扰，使用 `imclearborder` 函数删除边界对象。

(2) 删除小面积对象

`% bwareaopen:删除小面积对象`

`BW2 = bwareaopen(IM2, 60, 8);`

提取的连通区域最小面积为 60。

限制对图中散点的提取。

(3) 边缘提取

`%寻找边缘，不包括孔`

`[B,L] = bwboundaries(BW2, 'noholes');`

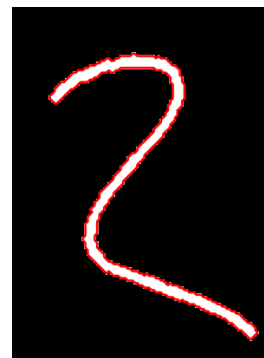
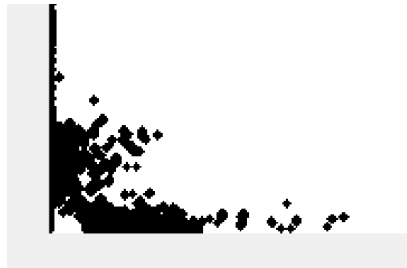
提取纸上线条。

(4) 描边

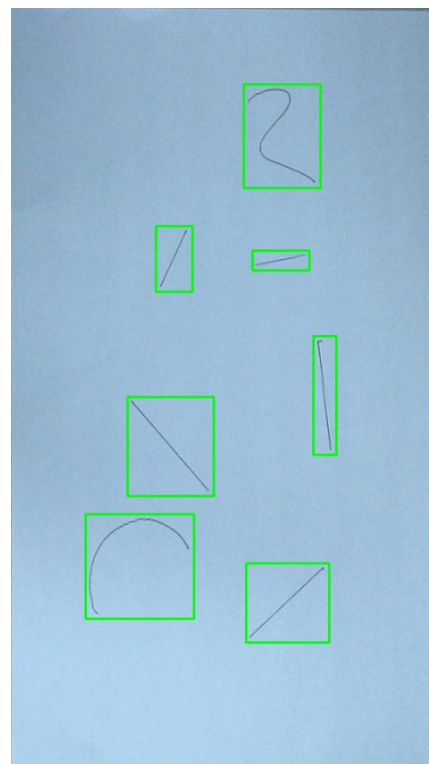
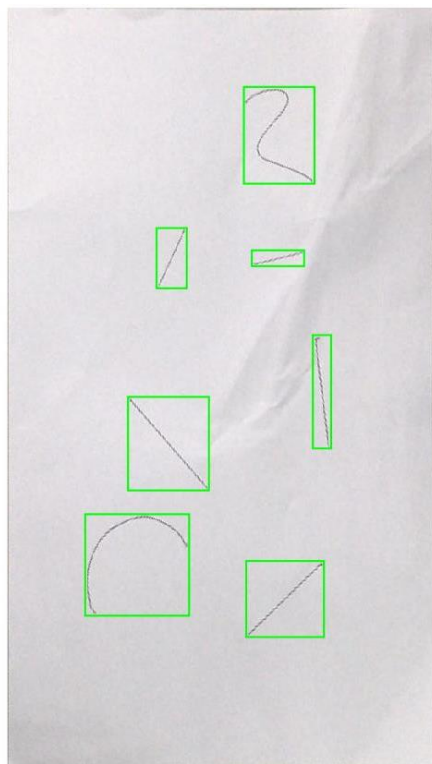
`% 描边`

```
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'r', 'LineWidth', 1)
end
```

用红色对提取处边缘进行描画。



## 2.3 通过 ROI 将每张图片输出



上图中左图为本程序实现的结果，右图 of 需求文档要求的结果，对比可知基本实现了要求，图像色调的区别是输入原图不同导致的。(本程序按照要求选用作业 1 的输出作为输入)

### 1) 获取图像的 basic 属性

主要目的是提取BoundingBox属性。

```
stats = regionprops(BW2,'basic');
```

% 3. 通过ROI将每张图片输出

%获取区域的'basic'属性，'Area'，'Centroid'，and 'BoundingBox'

```
stats = regionprops(BW2, 'basic');
```

### 2) 绘制感兴趣区域 ROI

rectangle 绘制矩形框

'Position' 参数用 [stats(i).BoundingBox]即可把上一步提取的边缘绘制出来。

```
for i=1:size(stats)
    rectangle('Position', [stats(i).BoundingBox], 'LineWidth', 1, 'LineStyle', '-', 'EdgeColor', 'g'),
end
```

### 3) 截取各个区域

循环遍历 7 个区域，对每个区域分别截取：

```
I = imcrop(input_img, stats(i).BoundingBox);
```

```
for i=1:size(stats)
```

```
    I = imcrop(input_img, stats(i).BoundingBox);
```

```
    subplot(2, 4, i);
```

```
    imshow(I);
```

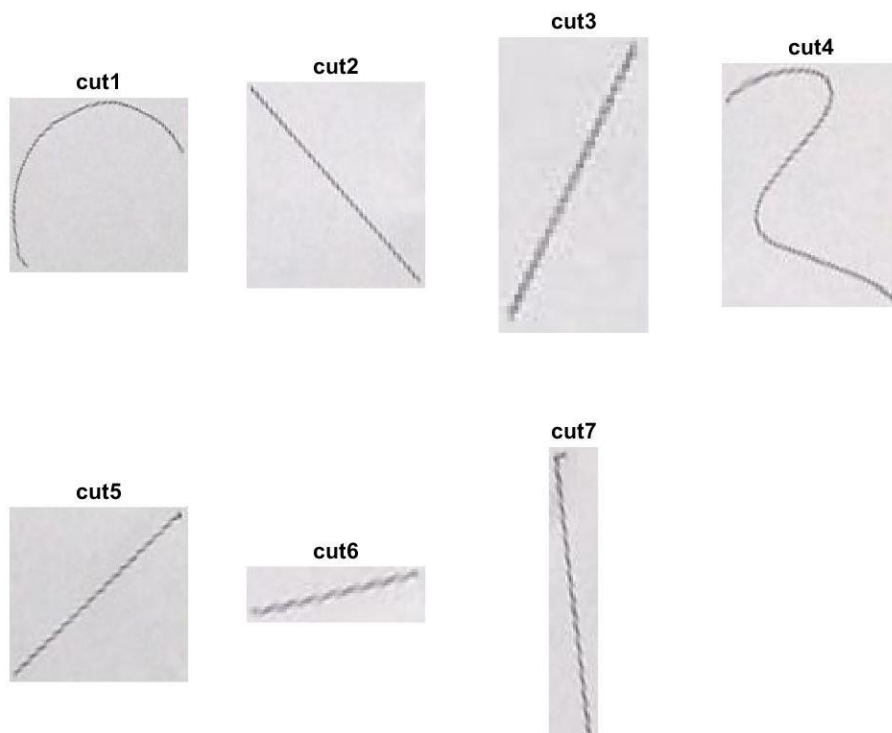
```
    titlename = ['cut', num2str(i)];
```

```
    title(titlename);
```

```
    filename = ['..\output\cut\cut', num2str(i), '.jpg'];
```

```
    imwrite(I, filename);
```

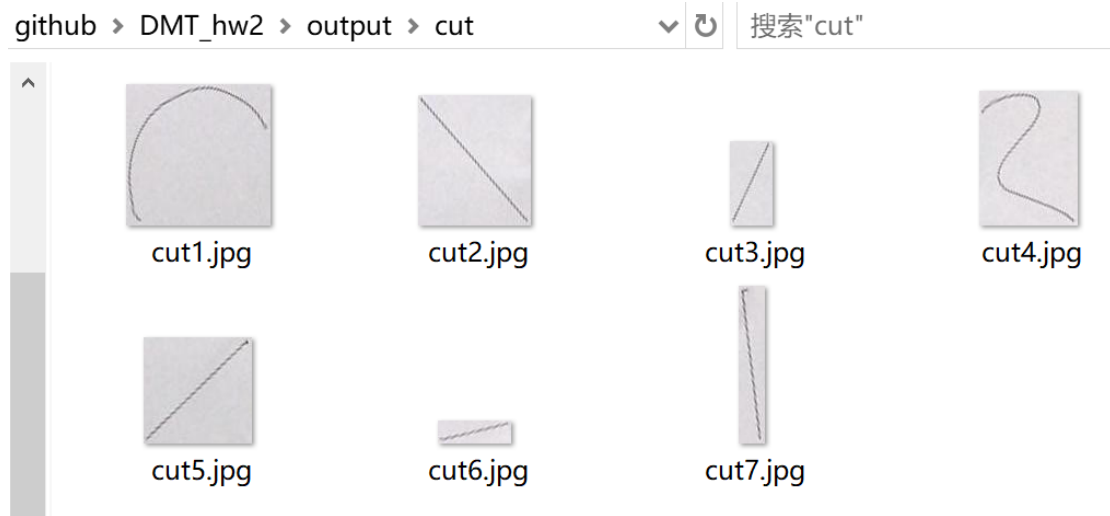
```
end
```



### 3. 实验结果

如上图所示。已按实验要求将 A4 纸上的线条单独识别了出来。

具体结果见根目录下.\output\cut 目录



中间步骤的结果图像也保存在根目录下.\output 目录中。

可以直接在文件夹中查看。

