

数字图像处理作业六

夏厚 PB18051031

2021 年 5 月 25 日

1 实验原理

1.1 坎尼边缘检测

坎尼边缘检测器基于三个基本目标：

- 低错误率，所有边缘都应该被找到，并且没有伪响应
- 边缘点被很好的定位
- 单一的边缘点响应，即边缘为单像素宽

坎尼边缘检测算法分为如下基本步骤：

1. 用一个高斯滤波器平滑输入图像。
2. 计算梯度幅值图像和角度图像。
3. 对梯度幅值图像应用非最大检测。
4. 使用双阈值处理和连接分析来检测并连接边缘。

梯度幅度与方向：

$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$
$$\alpha(x, y) = \arctan\left[\frac{g_y}{g_x}\right]$$

非最大抑制：

- 1、寻找最接近 $\alpha(x, y)$ 的方向 d_k 。
- 2、若 $M(x, y)$ 的值至少小于沿 d_k 的两个邻居之一，则抑制 $g_N(x, y) = 0$ 。

双阈值处理

$$g_{NH}(x, y) = g_N(x, y) \geq T_H$$

$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

边缘连接:

- (a) 在 $g_{NH}(x, y)$ 中定位下一个未被访问的边缘像素 p 。
- (b) 在 $g_{NL}(x, y)$ 中将所有弱像素标记为有效像素，并 8 连通到 p 。
- (c) 若 $g_{NH}(x, y)$ 中所有非零像素已被访问，跳到 (d)，否则返回 (a)。
- (d) 将 $g_{NL}(x, y)$ 中未标记边缘像素置零。

1.2 霍夫变换

霍夫变换是将平面中直线 $y = ax + b$ 变换到霍夫投影空间，变换为：

$$\rho = x \cos \theta + y \sin \theta = \sqrt{x^2 + y^2} \cos(\theta + \varphi)$$

将参数空间划分为所谓的累加单元，其中 (ρ_{min}, ρ_{max}) 和 $(\theta_{min}, \theta_{max})$ 是所期望的参数值范围： $-90^\circ \leq \theta \leq 90^\circ$ 和 $-D \leq \rho \leq D$ 。对参数空间的格子进行投票。

1.3 全局阈值处理

当物体和背景像素的灰度分布十分明显时，可以用适用整个图像的单个全局阈值。在大多数应用中，通常图像之间较大的变化，即使全局阈值是一种适合的方法，也需要有能对每幅图像自动估计阈值的算法。下面迭代算法可以用于这一目的：

1. 为全局阈值 T 选择一个初始估计值。
2. 在下式中用 T 分割该图像。这将产生两组像素： G_1 由灰度值大于 T 的所有像素组成， G_2 由所有小于等于 T 的像素组成。

$$g(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases}$$

3. 对 G_1 和 G_2 的像素分别计算平均灰度均值 m_1 和 m_2 。
4. 计算一个新的阈值：

$$T = \frac{1}{2}(m_1 + m_2)$$

5. 重复步骤 2 到步骤 4，直到连续迭代中的 T 值间的差小于一个预定的参数 ΔT 为止。

参数 ΔT 用于控制迭代的次数。通常 ΔT 越大，则算法执行的迭代次数越少。所选的初始阈值必须是大于图像最小灰度级而小于最大灰度级。平均灰度对于 T 来说是一个较好的初始选择。

2 Otsu 最佳阈值处理

阈值处理可视为一种统计决策理论问题，目的在于把像素分配给两个或多个组的过程中引入的平均误差最小。Otsu 方法在类间方差最大的情况下是最佳的。并且该方法完全以一幅图像的直方图为基础执行计算，而直方图是很容易得到的一维阵列。

Otsu 算法如下：

1. 计算输入图像的归一化直方图。使用 $p_i, i = 0, 1, 2, \dots, L-1$ 表示该直方图的各个分量。 $p_i = n_i/MN$, $0, 1, 2, \dots, L-1$ 表示一幅大小为 $M \times N$ 像素的数字图像中 L 个不同的灰度级， n_i 表示灰度级为 i 的像素数。
2. 使用下式，对于 $k = 0, 1, 2, \dots, L-1$ ，计算累积和 $P_1(k)$ 。

$$P_1(k) = \sum_{i=0}^k p_i$$

3. 使用下式，对于 $k = 0, 1, 2, \dots, L-1$ ，计算累积均值 $m(k)$ 。

$$m(k) = \sum_{i=0}^k ip_i$$

4. 计算全局灰度均值 m_G 。

$$m_G = \sum_{i=0}^{L-1} ip_i$$

5. 对于 $k = 0, 1, 2, \dots, L-1$ ，计算类间方差 $\sigma_B^2(k)$ 。

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

6. 得到 Otsu 阈值 k^* ，即使得 $\sigma_B^2(k)$ 最大的 k 值。如果最大值不唯一，用相应检测到的各个最大值 k 平均得到 k^* 。

7. 计算可分性测度

$$\eta(k^*) = \frac{\sigma_B^2(k^*)}{\sigma_G^2}$$

2.1 分块可变阈值分割

由于噪声或者光照不均匀等等一系列问题，导致无法直接进行阈值分割，图像平滑和边缘信息有益于阈值处理，但是在更经常的情况下，上述的两种做法效果不明显，只能使用可变阈值来进行解决。可变阈值处理的简单方法之一就是在一幅图像分成不重叠的矩形。这种方法用于补偿光照或反射的不均匀性。选择的矩形要足够小，以便每个矩形的光照都近似是均匀的。

3 实验内容

3.1 霍夫变换：直线检测

实验中实验 5×5 的高斯滤波器对原图进行平滑，并进行坎尼边缘提取。其中梯度计算使用了 sobel 算子计算水平方向和竖直方向的一阶导数。然后将边缘图像映射到霍夫空间中，寻找霍夫空间最大的四个点，也就是对应实际边缘图像中的四条单像素宽的直线。并在最后将直线叠加到原图上。

实验 MATLAB 代码分为若干函数,主要是高斯平滑滤波器函数 *myAverage.m*、提取边缘赋值和方向函数 *GetHandA.m*、非最大抑制函数 *NonMaxMum.m*、双阈值处理与提取函数 *DoubleThold.m*、霍夫变换函数 *Hough.m* 和霍夫反变换函数 *iHough.m*。代码较多，实验报告中就不予贴出，见同目录下.m 文件。

复现 10.34 结果如图 1:

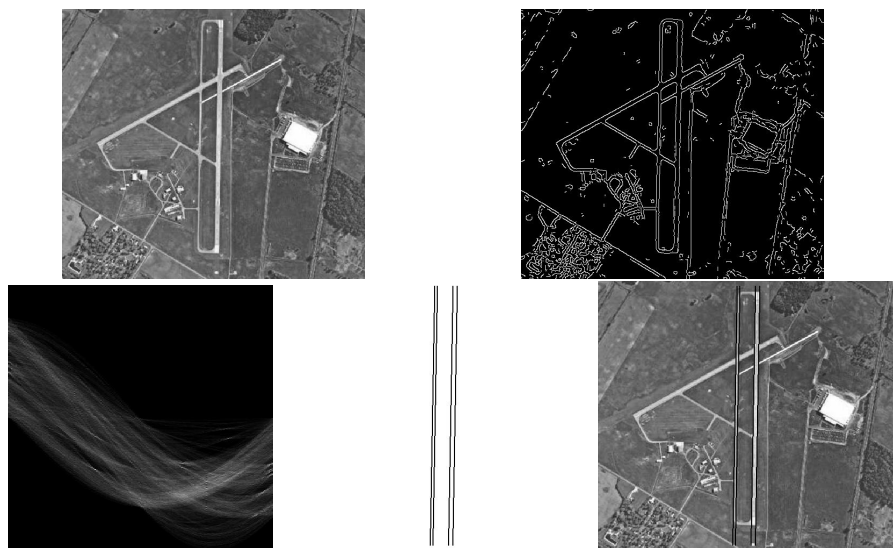


图 1: 分别为原图、坎尼算法提取边缘图像、霍夫参数空间图像、图像对应平面中强调的直线、直线叠加到原图

3.2 基本的全局阈值处理

对原图采用实验原理所述的算法以阈值估计为基础进行分割。使用 $T_0 = m$ （平均图像灰度）开始，并令 $\Delta T = 0$ ，进行迭代之后得到阈值 $T = 125.4$ 。使用阈值 $T = 125$ 来分割原图像得到的结果。

复现图 10.38 结果如图 2：

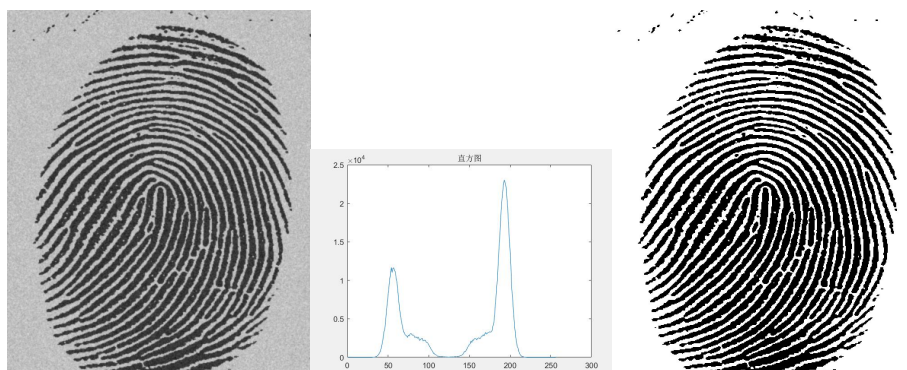


图 2: 分别为带噪声的指纹图像、直方图、全局阈值分割结果图像

3.3 Otsu 阈值分割

对原图统计其直方图，并使用前一实验内容中的全局阈值处理算法进行处理，得到分割结果。从直方图可以看到直方图并没有明显的波谷，且背景和物体间的灰度差别很小，所以全局阈值处理算法未能完成期望的分割。使用 Otsu 方法进行处理，得到阈值为 181，更接近图像中定义的细胞较亮区域。

复现 10.39 结果如图 3:

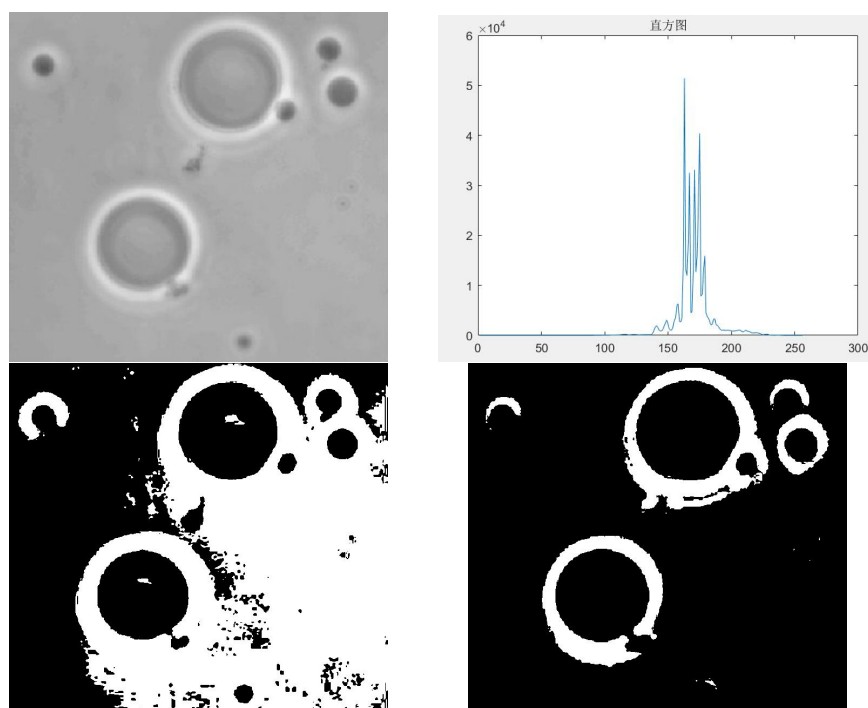


图 3: 分别为原图、直方图、基本全局阈值分割结果、Otsu 方法得到的结果

3.4 图像的分块可变阈值处理

原图的直方图显示使用一个全局阈值不能分割该图像。使用全局阈值处理分割和 Otsu 方法分割图像。可以看到这两种方法分割出现了大量的错误。将原图像细分为 6 个矩形区域，对每一个矩形区域使用 Otsu 方法分割，尽管分割结果中仍然有一些错误，但在一幅分割起来十分困难的图像上，图

像细分产生了合理的结果。

复现图 10.46 如图 4:

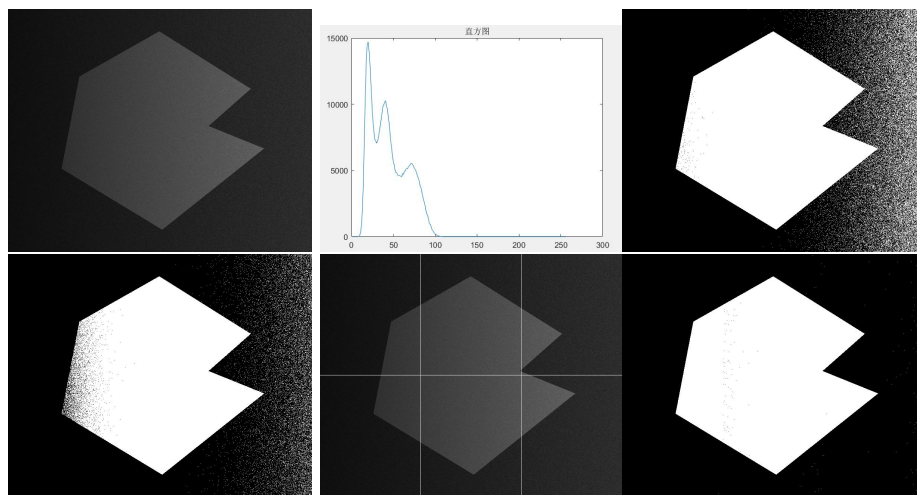


图 4: 分别为原图、直方图、基本全局阈值分割结果、使用 Otsu 方法得到的结果, 分为 6 幅子图像后的图像, 对每幅子图应用 Otsu 方法的结果

4 总结

- 吸取上次实验未将各功能模块独立成函数的教训, 本次实验将每一个步骤尽量独立成函数。使得主函数简洁易读, 程序移植性也更好。