

# 计算机视觉第五次作业

朱明杰 15331441

## 一. 实验任务

1. 用手机拍摄 100 张不同的 A4 纸的图像(以压缩包的形式提交)。拍摄要求: 要求得到与前面 A4 纸矫正中用的图像类似, A4 纸上尽量只包含手写/印刷体的数字或者英文字母 (图像中 A4 纸最多占图像大小的  $1/2$ , 即图像中需要一部分背景)。
2. 简要描述“迭代法”和“OSTU”这两个方法的区别。
3. 尝试用 C++(只用 Cimg 图像库)编程实现“迭代法”和“OSTU”这两个图像分割方法(要求把每个方法都各自封装成类)。然后对自己拍摄的 100 张图像做图像分割处理, 以验证算法的正确性, 并给出性能分析(包括准确性和时间复杂度)。在实验报告中给出实验部分实验结果(10 个最好的结果和 10 个最差的结果), 并分析结果好坏的原因。

## 二. 实验工具

Visual studio 2015

Cimg Library

LibJpeg

## 三. 算法流程

### 1. 迭代法。

迭代法本质是 K-means 聚类方法在一维的情形, 两个图像部分的均值对应 K-means 方法中的聚类均值, 阈值对应 K-means 方法中的分类边界。由于 K-means 方法中的初始值会极大地影响 K-means 方法的收敛性以及收敛速度, 类似 K-means++ 的思想, 我们先贪心地取一个阈值, 这里直接取图像灰度的中位数就好。

### 2. Otsu 法。

Otsu 法的目标是计算能将两类分得最开开的最佳阈值, 使得它们的类内方差最小; 由于两两平方距离恒定, 所以即它们的类间方差最大。最佳阈值的计算直接在所有灰度值范围内等间隔采样, 取类间方差最大的那个即可。

### 3. 迭代法与 Otsu 法的区别。

- i) 迭代法与 Otsu 法的执行策略不一样。迭代法本质上是 K-means 算法的一个特例, 是一种启发式算法, 通过对阈值的不断迭代求解其可能陷入局部最优解; Otsu 法优化目标则明确了起来, 通过将阈值可取区间离散化采样并计算类间方差求最值来计算最优解, 只要采样间隔足够小, 解就不会陷入局部最优解;
- ii) 迭代法与 Otsu 法的优化目标函数不一样。

为了表述方便, 记图像为  $\mathbf{A}$ , 共有  $n$  个像素点, 平均灰度为  $u$ , 点  $P$  处像素灰度值为  $I_P$ 。前景共有  $n_1$  个像素点, 像素个数占图像比例为  $w_1$ , 平均灰度为  $u_1$ ; 背景

像素个数占图像比例为 $w_2$ ，平均灰度为 $u_2$ 。灰度划分的阈值为 $t$ 。我们马上有以下的式子：

$$\begin{aligned}
 n &= n_1 + n_2 \\
 w_1 &= \frac{n_1}{n}, w_2 = \frac{n_2}{n} \\
 w_1 + w_2 &= 1 \\
 n_1 &= \sum_{I_P < t} 1, n_2 = \sum_{I_Q \geq t} 1 \\
 u_1 &= \frac{1}{n_1} \sum_{I_P < t} I_P, u_2 = \frac{1}{n_2} \sum_{I_Q \geq t} I_Q \\
 n_1 u_1 + n_2 u_2 &= \sum_{I_P < t} I_P + \sum_{I_Q \geq t} I_Q = \sum_{P \in A} I_P = nu
 \end{aligned}$$

迭代法的目标是最小化类内距离平方和，即找

$$\arg \min_t \left( \sum_{I_P < t} \|I_P - u_1\|^2 + \sum_{I_Q \geq t} \|I_Q - u_2\|^2 \right)$$

即

$$\arg \min_t \left( \sum_{I_P < t} (I_P - u_1)^2 + \sum_{I_Q \geq t} (I_Q - u_2)^2 \right)$$

即

$$\arg \min_t \left( \sum_{I_P < t} I_P^2 - 2u_1 \sum_{I_P < t} I_P + n_1 u_1^2 + \sum_{I_Q \geq t} I_Q^2 - 2u_2 \sum_{I_Q \geq t} I_Q + n_2 u_2^2 \right)$$

即

$$\arg \min_t \left( \sum_{I_P < t} I_P^2 + \sum_{I_Q \geq t} I_Q^2 - 2u_1^2 n_1 + n_1 u_1^2 - 2u_2^2 n_2 + n_2 u_2^2 \right)$$

由于对于一个给定的图像，

$$\sum_{I_P < t} I_P^2 + \sum_{I_Q \geq t} I_Q^2 = \sum_{P \in A} I_P^2$$

是定值，所以迭代法的目标即

$$\arg \min_t (-u_1^2 n_1 - u_2^2 n_2)$$

即

$$\arg \max_t (u_1^2 n_1 + u_2^2 n_2)$$

即

$$\arg \max_t \frac{n^2 u^2 - 2n u n_1 u_1 + n n_1 u_1^2}{n_2}$$

即

$$\arg \max_t \frac{n u^2 - 2u n_1 u_1 + n_1 u_1^2}{n_2}$$

Otsu 法的目标是最大化类间方差，即

$$\arg \max_t (w_1 w_2 (u_1 - u_2)^2)$$

即

$$\arg \max_t (n_1 n_2 (u_1 - u_2)^2)$$

即

$$\arg \max_t \frac{n^2 u^2 n_1 - 2 n^2 u n_1 u_1 + n^2 n_1 u_1^2}{n_2}$$

即

$$\arg \max_t \frac{u^2 n_1 - 2 u n_1 u_1 + n_1 u_1^2}{n_2}$$

可以看到，两个优化目标式子分母仅用  $n_1$  和  $u_1$  两个变量来表示，是不一样的。

#### 四. 算法代码 (Clmg) 实现

见压缩包下代码。

#### 五. 实验结果

A4 纸图像见工程文件下的 Dataset/。

迭代法结果见工程文件下的 Iteration/。

Otsu 法结果见工程文件下的 Otsu/。

#### 六. 分析与评价

算法的正确性得以验证。迭代法的时间复杂度为  $O(Kn)$ ，其中  $K$  为迭代次数；Otsu 法的时间复杂度为  $O(Bn)$ ，其中  $B$  为对灰度范围的采样数。实验的最好 10 个结果为 1、2、3、36、44、45、52、92、96、98，因为字比较简单，而且背景对于灰度的干扰不大；最差的 10 个结果为 12、21、26、27、28、37、49、59、64、66，因为字比较简单，而且背景对灰度的干扰较大，拍照的时候还可能会有抖动模糊的情况。

另外这次作业需要用额外的接口来读写 JPEG 文件，这里我采用了 LibJpeg 库，之后才了解到有用 SIMD 实现加速的 LibJpeg-Turbo 版本，很惭愧。有一个尚未解决的问题是，由于手机重力感应系统的问题，一开始拍出来的照片可能是不正的，这个时候我在手机上或者通过 Win10 的照片应用将图片旋转  $90^\circ$  若干次使得照片是正的。但是通过 Clmg 与 LibJpeg 来处理之后，所导出的图像又是之前不正的方向，现在还没有找到一个解决方法，Google 了一圈貌似也没有找到满意的答案，伤心。