

# Homework - Advanced Morphological Operations

杨铭

5130379022

2015 年 11 月 12 日

## 1 简述

主要代码在 `Morp.py` 中，在 `imglab.py` 中使用了在网上学习的一种 thinning 算法对土豆图进行细化。使用 python 编写，调用了 python 的 `copy` 库来做输出数组的深度复制，以及 `opencv` 的 python 接口实现对各种格式图像的读取和写入

## 2 输入格式

使用 `opencv` 的 `cv2.imread` 函数读取图片，支持 `.bmp`, `.jpg`, `.png` 等格式

## 3 实现方式

实现了以下几个算法

- hit-or-miss transform
- difference set of image
- union set of image
- thinning
- thickening

**hit-or-miss transform** 击中不击中变换，根据定义我对算法进行了优化

$$A \text{ hit-or-miss } B = (A \ominus B_1) \cap (A \ominus B_2)$$

实际是对结构体 B 在原图上进行严格匹配，即如果遍历原图，将结构体的中心放置于原图的点上，然后查看原图周围的点是否和结构体的颜色一致。这样优化以后，省去了两次腐蚀的操作，变为了一次匹配，能提高运算的效率

**difference and union set of image** 两个图片的差集和并集

没什么好说的，遍历一边出结果

**diffImg**  $A - B$

**unionImg**  $A \cup B$

**thinning** 细化由于作业第一题没有说明使用什么算法，我一开始上网查找了大量细化的算法。使用其中 ZhangThinning 和根据一个点四周的颜色计算权重的算法对土豆图提取骨架，但是效果并不好。最后老老实实用了 hit-or-miss 的细化算法来提取骨架。

算法

$$A - (A \text{ hit-or-miss } B)$$

结构体 其实就是 ppt 上那 8 个

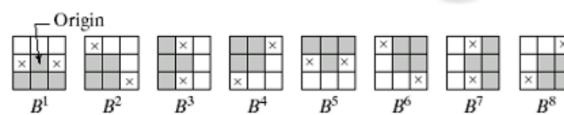


Figure 1:

**thickening** 增厚

类似于细化，这里不做赘述

## 4 结果



Figure 2: 原图

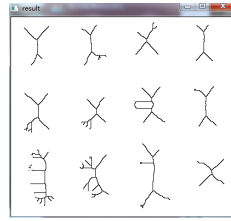


Figure 3: 骨架

**分析** 由于之前所忽略的二值化准确度，导致结果有些瑕疵。土豆中间的缺痕在二值化后变为背景色，导致细化后图片上出现异物。并且每一次和某个结构体进行 thinning 运算，只能消除外围的一个像素的，而 python 语言的效率较低，导致我跑了 5,6 分钟才出来比较理想的结果。