## **Thinning**

对于 Thinning 算法,我阅读并学习的所给的算法文档,研究了其中的算法,并将它实践出来。但是结果并没有如算法文档所示那样,结果有所偏差,并且我对算法文档有些疑问,想与您探讨一下,下面是我理解的 Thinning 算法流程:

1. 创建查询表, 该查询表为 50 个 3×3 大小的 template 型数据, 如下图

Amount	Elimination Rules				
0	Never				
1	Never				
2	1 t 1 t t t t t t t t t t t t t t t t t				
3	1 3 1				
4	3 1 5 1 1 1 1 3 3 1 1 1 3 1				
5	1 1 3 1				
6	1 1				
7	1 1				
8	Never				

注意,这里假定 EDGE PIXEL 为高值(e.g., 255),但用黑色表示; 而背景为低值(0), 却用白色表示. 你的 CODE 是如何的?

> 这里应为删除的还是不删除的? 而且还与 EDGE PIXEL 用低值表示

> 还是高值表示有关----必须始终

与作者的假定保持一致才行!

Fig 1. New elimination rules, blank denote the white pixel

- 2. 对目标图像的每一个像素点的8邻域进行遍历并与查询表内的模版进行对比,如不满足任何一模版,符合保留要求,否则进行第三步。如果该点不被删除,则跳到第五步。
- 3. 如果需要删除,还需要判断像素宽度是否为2,如果不为2,则确定删除(将目标点的像素值置为 0 );如果像素宽度为2,则需要步骤4进一步判断是否真的删除
- 4. 创建查询表二,即图二中的7个模版(如图二),如果当前像素满足任何一个模版,则不删除此像素(即使其满足步骤2中的删除条件);如果7个模版都不满足,则可以确定删除该像素。

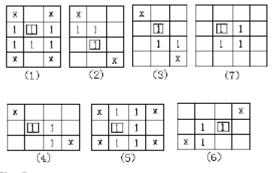


Fig 2. The preserved template used in our algorithm, blank denote 0, "x" are don't care

5. 从步骤 2 开始继续迭代,直到没有像素点需要删除了。

疑问:对于算法文档中的two-pixel-width(我翻译为像素宽度为2),由于算法中没有给出明确的定义,并且在网上也没有找到相应的文章,所以我理解为领域内像素宽度为2,如下表格中的1

	1	1	
	1	1	
	1	1	

但是按我的理解,依旧得不出正确的结果,所以希望你能帮助我解答这个疑惑。给出 two-pixel-width 的定义。

A: 我所给出的样本 CODE 的 THINNING 算法, 请见 MATLAB 中的 THINNING CODE 的注释. 我所给出的样本 CODE 是我长期使用的程序,效果与 MATLAB 中的 THINNING 函数是一致的. 而我所给出的论文则可能是另一种算法.

感觉我所给出的论文里是假定 EDGE 的 PIXEL 为高像素值但却用 black 表示. two-pixel-width 应指的是具有 2PIXEL 宽度的 EDGE. 请检查你实装的 CODE 是否始终与作者的诸假定保持一致?

并将检查结果告诉我好么?

不关怎样,对于我所给出的样本 CODE 的 THINNING 算法(MATLAB LIKE),其算法的道理是很简单的,请先实装它.