

# 基于数学形态学的二值图像细化算法研究

李迎 段汕\*

(中南民族大学 计算机科学学院, 武汉 430074)

**摘要** 将数学形态学这一方法应用于数字图像的细化中, 针对已有细化算法中存在毛刺这种常见现象, 对原有的剪枝结构元素进行改进, 提出了一种新的结构元素消除毛刺. 在 Visual C++ 软件编程实现的基础上, 对这两种结构元素在计算量、处理速度、效果等方面作了相应的测试和比较. 结果表明: 改进后的剪枝结构元素在这3个方面都得到较好的改善, 并获得了理想的处理结果.

**关键词** 数学形态学; 细化; 结构元素

**中图分类号** TP391.4 **文献标识码** A **文章编号** 1672-4321(2005)04-0096-04

## Study of Binary Image Thinning Based on Mathematical Morphology

Li Ying Duan Shan

**Abstract** This paper applies mathematical morphology to digital image thinning, introduces improved structuring elements in view of the traditional thinning algorithm exit "thin\_hair" phenomena in its result skeleton, and compares two kinds of structuring elements in calculation, procession speed and the impression. It is proved to be better than the traditional way. All the data are based on computer programming which realized in Visual C++.

**Keywords** mathematical morphology; thinning; structuring elements

**Li Ying** Master's Candidate, College of Computer Science, SCUFN, Wuhan 430074, China

细化是处理线状二值图像的一种重要技术, 在图形文字识别、图像数据压缩、线状目标自动跟踪以及特征提取与描述等方面均有应用. 细化处理能极大地减少图像中的冗余信息量, 使计算机在分析处理、识别时能很快地接触到本质内容, 这样可减少运算量从而缩短识别的时间. 近几年来, 各国学者为此提出了许多细化算法, 但大多是在以下两类方法的基础上发展起来的. 一是 Blum 定义的一种细化方法, 即所谓的中轴变换; 二是 Rosenfeld 提出的局部差别删除法<sup>[1]</sup>. 由于这些算法在用于较复杂模式时, 都不同程度地存在判断复杂、执行时间长、对噪声敏感、不能很好地保持原模式的形状与连通性等问题<sup>[2]</sup>. 因此, 无法在图像处理和识别领域获得广泛的应用.

20 世纪70年代中期Matheron G 严谨而详尽地论证了随机集论和积分几何, 为数学形态学奠定了理论基础. 与此同时, 形态学细化算法也得到了相应

地发展, 其基本思想是, 通过所给定的一系列具有一定形状的结构元素对(模板)的形态变换, 删除满足击中击不中变换的像素. 本文在已有文献的基础上, 对形态学细化算法中结构元素对(模板)的选取方法进行了一定的研究. 提出改进的结构元素对实现细化. 实验结果证明改进后的结构元素对具有较好的细化效果.

## 1 基本的形态运算

在数学形态学中, 结构元素是最重要、最基本的概念. 设  $N$  维欧氏空间  $E^N$ , 则结构元素可以定义为  $E^N$  或其子空间  $E^N$  上的一个集合, 具有一定的几何形状(如圆、球、有向线段、有向点对等). 在形态细化方法中, 需要应用到腐蚀、击中击不中变换及细化这3种形态运算方法. 对二值图像而言, 形态腐蚀运算  $A \ominus B$  具有收缩输入图像的作用(前提是原点在结构元素的内部<sup>[3]</sup>); 击中击不中变换  $A * B$  可在一次运

收稿日期 2005-05-24 \* 通讯联系人

作者简介 李迎(1981-), 女, 硕士研究生, 研究方向: 数字图像处理, E-mail: zero0361@sina.com.cn

算中同时捕获到内外标记,对于研究图像中目标与背景之间的关系,往往会得到很好的效果;细化  $A \otimes B$  则是用来删除  $A$  中被  $B$  击中的部分,它使图像线划变细,是图像形态细化算法的基础。

## 2 形态学细化算法

通俗地讲,细化过程就是对图像逐层剥离的过程。对集合  $A$  进行基于结构元素对序列(以下简称结构对序列)  $\{B\} = \{B_1, B_2, \dots, B_n\}$  的细化,其过程则可以用  $A \otimes \{B\} = (\dots((A \otimes B_1) B_2) \dots) B_n$  这个迭代运算来表示。这样以来随着迭代次数的增加,目标图像将不断细化,且始终保持  $A \otimes B \subset A$ 。形态细化算法往往是基于上述思想进行设计的,且遵循一定的原则,其中包括:(1) 输出骨架应是一条细线(只有一个像素的宽度);(2) 细化结果是原目标图像的中心线;(3) 细化过程中不破坏图像的连通性;(4) 具有好的稳定性。

对于形态运算方法来说,结构元素对序列  $B$  的选择尤为重要,且直接影响图像细化的质量。细化算法将不断重复地剥离二值图像的边界像素。但是,对于边界像素的剥离应保持目标图像纹线的连接性,方向性和特征点不变,还应保持纹线的中心基本不变<sup>[4]</sup>,最终使得变换图像成为单像素宽的图像骨架。

通过常用结构元素对序列  $D = \{D_1, D_2, D_3, D_4\}$  和  $E = \{E_1, E_2, E_3, E_4\}$ ,对基于形态学的细化算法进行测试。其中结构元素对序列  $D$  用来消去西北、东北、东南和西南4方向上的点,如图1所示。结构元素对序列  $E$  则用来消去北、东、南和西4方向上的点,见图2。在图1和图2中,“1”表示目标图像上的点,“0”表示背景图像上的点,“\*”既可以表示目标图像上的点,也可以是背景图像上的点。

0	0	*
0	1	1
*	1	1

$D_1$

*	0	0
1	1	0
1	1	*

$D_2$

1	1	*
1	1	0
*	0	0

$D_3$

*	1	1
0	1	
0	0	*

$D_4$

图1 用来消去西北、东北、东南和西南方向的结构对序列  $D$

0	0	0
*	1	*
1	1	1

$E_1$

1	*	0
1	1	0
1	*	0

$E_2$

1	1	1
*	1	*
0	0	0

$E_3$

0	*	1
0	1	1
0	*	1

$E_4$

图2 用来消去北、东、南和西方向的结构对序列  $E$

利用结构元素序列对  $D$ 、 $E$  细化后的图像存在细化不彻底的问题,所以利用改进的结构元素序列对  $X = \{X_1, X_2, X_3, X_4\}$  (如图3所示)再次进行细化。

0	0	*
0	1	1
*	1	*

$X_1$

*	0	0
1	1	0
*	1	

$X_2$

*	1	*
1	1	0
*	0	0

$X_3$

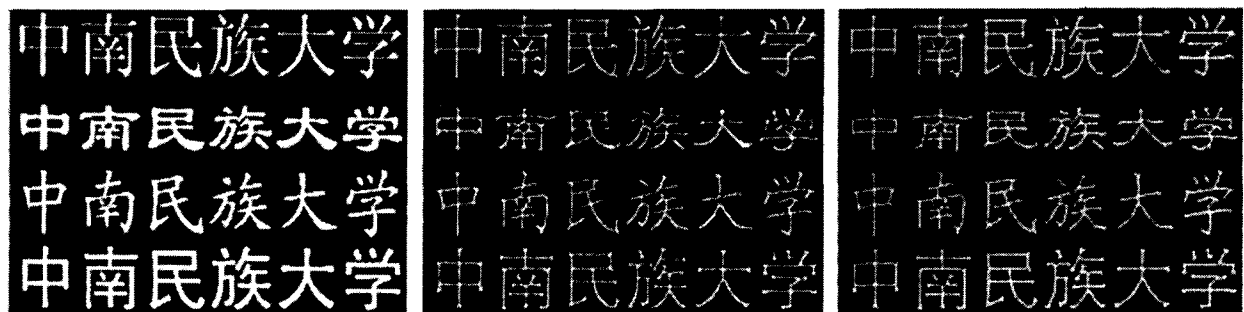
*	1	*
0	1	1
0	0	*

$X_4$

图3 改进后的结构对序列  $X$

在应用 Visual C++ 对细化算法进行实现的过程中,涉及到的几个主要函数都是根据形态学中基本的形态运算思想进行设计,它们包括细化函数  $\text{thin}()$ 、击中不中函数  $\text{hmt}()$  和腐蚀函数  $\text{erosion}()$ 。

利用结构对序列  $D$ 、 $E$  以及  $X$  对二值图像进行细化的效果见图4,其中图4(a)是原始的二值图像,字体从上到下依次是:宋体、隶书、楷体、黑体,图4(b)是使用结构元素  $D$ 、 $E$  细化后的图像,图4(c)是在  $D$ 、 $E$  细化后再次利用结构元素  $X$  进一步细化后的图像。从测试结果中可以很清楚的看到,结构对序列  $X$  的提出很好地解决了原来细化不彻底的问题,很好地改善了图像的细化质量。



(a) 包含4种字体的原始图

(b) 结构对序列  $D$ 、 $E$  细化后的图像

(c) 结构对序列  $X$  进一步细化后的图像

图4 细化对比效果图

### 3 结构元素的改进

通过实验发现,利用结构对序列 $D$ 、 $E$ 以及 $X$ 细化后的图像存在一个明显缺陷:细化后的图像会产生毛刺.针对这个缺陷,介绍常用的剪枝处理方法以及针对常用剪枝的不足所提出的改进剪枝算法.

#### 3.1 剪枝处理

经过研究发现,毛刺的出现对纹线方向十分敏感,纹线方向角在第二象限的时候容易出现毛刺,特别是纹线近似水平和垂直的时候,毛刺的出现尤其明显.毛刺的产生是由于结构元素的不对称,导致像素点的删除不是对称进行的,所以该细化算法对纹线一定方向上的突出十分敏感,使得纹线上一点小小的突起不能被完全删除,最后演变成了毛刺.

为了能够消除毛刺,可以通过设计消除毛刺的结构元素,来达到剪枝的目的.根据以上分析,结构元素应针对已经细化成单像素的部分进行设计,且应满足8个方向上的剪枝.常用的剪枝结构对序列 $L = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8\}$ ,如图5所示.这8个结构对序列中, $L_q = \{L_2, L_4, L_6, L_8\}$ 为强修剪器, $L_r = \{L_1, L_3, L_5, L_7\}$ 为弱修剪器.

$L_1$	$L_2$	$L_3$	$L_4$
1 0 0	0 1 0	0 0 1	0 0 0
0 1 0	0 1 0	0 1 0	0 1 1
0 0 0	0 0 0	0 0 0	0 0 0
$L_5$	$L_6$	$L_7$	$L_8$
0 0 0	0 0 0	0 0 0	0 0 0
0 1 0	0 1 0	0 1 0	1 1 0
0 0 1	0 1 0	1 0 0	0 0 0

图5 用来消除毛刺的结构对序列 $L$

在剪枝程序实现的过程中要注意对循环次数的设定,避免图像被不断的消减.这一点与前面的细化算法有所不同,前面介绍的细化算法会在结构对之间进行循环,且逐步达到稳定状态.经过多次试验表明,剪枝程序所循环的次数若取为毛刺的长度(像素个数),效果最为理想.

将图5所示的8个方向的结构对序列用同样的细化算法流程执行,实验表明,使用结构对序列 $L$ 对于宋体、隶书、楷体这3种字体可以达到一定的剪枝效果.

#### 3.2 改进的剪枝处理

观察利用剪枝结构对序列 $L$ 剪枝后的图像,发现对比前3种字体,黑体字的剪枝效果十分不理想.如果能够适当减少 $L$ 中击中点(标记为“1”的点)的个数,将结构对序列 $L_r$ 4个角(分别是西北角、东北角、东南角和西南角4个方向)中的“0”改成“\*”,即可得到一种改进的,能够实现进一步剪枝的结构对序列 $T = \{T_1, T_2, T_3, T_4\}$ ,如图6所示.剪枝结构对序列 $T$ 将改善黑体字的剪枝效果.很显然, $T$ 是16种不同的结构对的组合,所以它比常用包含8种情况的剪枝结构对序列 $L$ 有着更多选择,意味着它可以减去更多的分支.

$T_1$	$T_2$	$T_3$	$T_4$
0 0 0	1 * 0	0 * 1	0 0 0
* 1 0	* 1 0	0 1 *	0 1 *
1 * 0	0 0 0	0 0 0	0 * 1

图6 改进的剪枝结构对序列 $T$

与常用的剪枝结构对序列 $L$ 进行比较,发现改进的剪枝结构对序列 $T$ 有4种情况与 $L$ 中的弱修剪器 $L_r = \{L_1, L_3, L_5, L_7\}$ 相同,为了避免不必要的重复,对算法作一下的修改.

(1) 将剪枝结构对序列 $L = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8\}$ 由原来的8个方向的剪枝改成4个方向的剪枝 $L_q = \{L_2, L_4, L_6, L_8\}$ .

(2) 然后使用改进的剪枝结构对序列 $T = \{T_1, T_2, T_3, T_4\}$ 进一步剪枝.

图7给出不同结构对组合的剪枝效果图.其中图7(a)为仅仅使用剪枝结构对序列 $L$ (也即 $L_q + L_r$ )的剪枝效果图,图7(b)为按照改进算法的步骤一使用强剪枝结构对序列 $L_q$ 的剪枝效果图,而图7(c)则是按照改进算法的第2个步骤,组合使用 $L_q$ 以及改进后的剪枝结构对序列 $T$ (也即 $L_q + T$ )的剪枝效果图.从图7中可以看到,改进后的算法不仅使得其他字体得到更好的剪枝,且使得黑体字也在原先的结构对序列 $L$ 无法剪枝的情况下得到很好的剪枝,达到理想的效果.

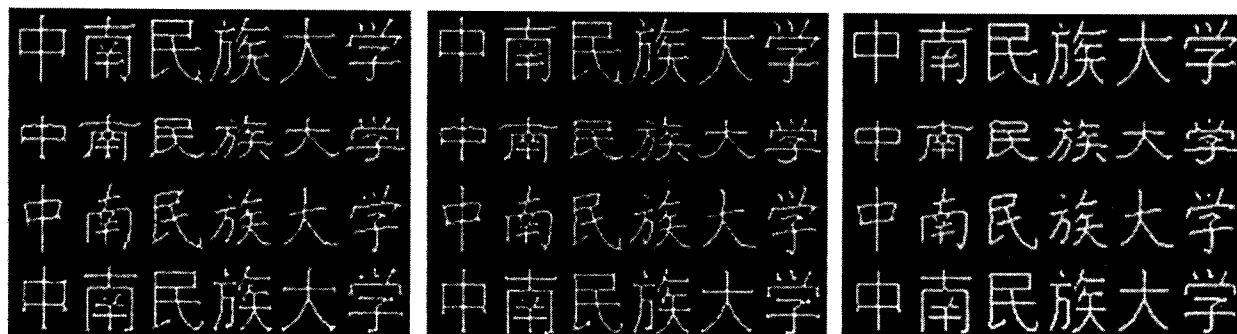
### 4 总结

本文中,笔者提出了一种利用数学形态学对二值图像进行细化的算法,它是通过结构元素对的组合,利用击中击不中变换对二值图像进行细化的.结构元素在二值图像细化的过程中有着很重要的作用,它没有固定的形状和大小,是根据二值图像的信

息和所需的结果来设定的,结构元素的选择非常灵活,不同的二值图像,需要不同的结构对和处理方法,选择的恰当与否将直接影响到二值图像的细化结果,可以这样说,结构元素是形态图像细化算法优越于其他图像细化算法的关键所在,但同时也是运

用形态学方法进行图像细化的难点.

本文在形态细化方法的基础上,通过对结构对的改进,较为理想地完成了对目标图像的细化处理.解决了之前常用的剪枝结构对序列针对黑体字剪枝效果不理想的问题.



(a) 利用  $L$  剪枝后的图像

(b) 利用  $L_q$  剪枝后的图像

(c) 利用  $L_q$  以及  $T$  剪枝后的图像

图7 剪枝前后效果对比图

#### 参 考 文 献

- [1] 张英琦,张庆林. 数学形态学应用于二值图像的细化[J]. 焦作工学院学报,1997,16(4):39~43
- [2] Lam L, Suen C Y. An evaluation of parallel thinning algorithms for character recognition[J]. IEEE Trans

on Pattern Analysis and Machine Intelligence,1995,17  
(9):914~919

- [3] 崔 屹. 图像处理与分析——数学形态学方法及应用[M]. 北京:科学出版社,2002. 16
- [4] 冯星奎,李林艳,颜祖泉. 一种新的指纹图像细化算法[J]. 中国图像图形学,1999,4(10):835~838