第20卷第3期
2001年5月

无锡轻工大学学报
Journal of Wuxi University of Light Industry

Vol.20, No.3
May, 2001

文章编号:1009-038X(2001)03-0315-04

二值图象的细化

王怀群
(北京工业职业技术学院,北京 100042)

摘 要:把数学形态学这一方法应用于数字图象的细化中,提出一种新的二值图象细化算法,与常用的传统细化算法在计算量、处理速度、效果等方面进行相应的比较,结果表明:基于数学形态学的图像处理方法,具有计算简单、运算速度快、并行处理及易于硬件实现等特点.

关键词:数字图像处理;二值图象细化;数学形态学
中图分类号:TP391　　**文献标识码**:A

Fine-Grain Processing of Two Valued Image

WANG Huai-qun
(Beijing Vocational Technology Institute of Industry, Beijing 100042)

Abstract: In this paper, a new algorithm for applying the methods of mathematical morphology to fine-grain processing of digital image was propounded. The comparison of the new algorithm with the conventional one on computing load, processing speed and effects was made. As the result shows, on the basis of image processing methods of mathematical morphology, this new algorithm has the features of simple computation, high calculating speed, parallel processing, and easy realization.

Key words: digital image processing; fine-grain processing of two-valued image; mathematical morphology

在图象处理和模式识别的研究中,常涉及二值图像的细化问题,通过细化而抽取一个二值图形的骨架,是一项极有意义的工作.近几十年来,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.其一是Blum定义的一种细化法,即所谓的中轴变换;其二是Rosenfeld提出的局部差别删除法.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工的形状与连通性等问题,因此,无法在图象处理领域和识别领域获得广泛的应用.

数学形态学的建立,开辟了数学图象处理的新途径.该方法具有算法简单、速度快、并行处理、易于硬件实现等特点.

1 细化算法

所谓细化,就是寻找图形、笔刑的中轴线或骨架,以其骨架取代该图形或笔划.也就是说,细化之后该图形或笔划的象素宽度为1,可以用数学语言严格地描述图形的轴线或骨架.

1.1 一般细化算法

传统细化算法有很多,有串行算法和并行算

收稿日期:2001-03-23;修订日期:2001-04-29
作者简介:王怀群(1965-),男,甘肃兰州人,工学学士,高级讲师.

法.常用的有 Hilditch 细化、Deutch 细化和 Rosenfeld 细化等.

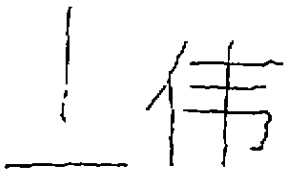
Hilditch 细化算法为串行处理方式,最终得到的是 8 条近邻连接线条(即细化薄线的每个象素都被认为和其周围的 8 个近邻的薄线象素连结).Deutch 细化算法的处理方式为并行处理方式.所得到的线图形形态是不完全的 8-连线,可以看作是 8-连接图形.Resonfeld 8-连接化这种细化算法的处理方式是并行处理方式,最终得到图形是 8-连接图形.

选择两个具有不同结构的图形作为处理对象,如图 1 所示,其中图(1)a 为手工图形,结构简单,仅由一横一竖组成,且不相连;b 为印刷体汉字"伟",由横、竖、撇、勾等多种结构组成,且相互交错.用 Hilkitch、Deutch 和 Rosenfeld 细化算法分别对此两个图形作细化,图 2、图 3、图 4 为细化结果,其中 time 为所用时间.

a 手工图形 b 汉字"伟"

图 1 细化原始图象

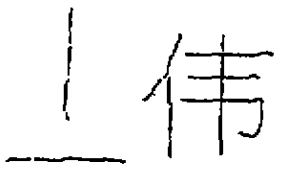
Fig.1 Original fine-grained image



t = 564 s t = 1044 s

图 2 Hilditch 细化结果

Fig.2 The result of hilditch fine-grained processing



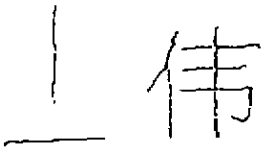
t = 329 s t = 605 s

图 3 Deutch 细化结果

Fig.3 The result of deutch fine-grained processing

1.2 基于数学形态学的细化算法

对图象的细化要保持图象的连通性.这是一条细化的最基本准则.无论是采用经典的算法还是应用数学形态学的方法,都毫不例外地要求遵守这一原则.同时我们看到:细化过程也就是对图象逐层剥落的过程,随着细化的进行,图象有规律地缩小.由数学形态学的基本知识可以知道,基本形态运算中的薄化,即 $A \bigcirc B$,可以使图形按一定规律不断缩小,并始终保持 $A \bigcirc B$ A.因此,可选取薄化运算作为进行细化的基本运算形式.



t = 630 s t = 1193 s

图 4 Rosenfeld 细化结果

Fig.4 The result of rosenfeld fine-grained processing

运算方法确定后,结构元素 B 的选择尤为重要,成为影响整个细化问题的关键.

1.2.1 确定结构元素 B

结构元素 B 首先应使得图形在细化过程中保持连通性.更具体地说,基于数学形态学的细化运算是由腐蚀和膨胀两种变换合成的,在腐蚀过程中,不断移动结构元素 B 的中心点,使它与图象 A 中各点重合,若当 A 中某点(灰度为 1)与 B 的中心重合,该点的邻点(在此取 8 邻域点)也恰好与 B 的其它点结构特征相同,则可将该点的灰度由 1 变为 0,即从 A 中删去该点,将中心点有 1 变为 0 后,A 的连通性不变.其次就是孤立点、一条线段的端点、非边界点的结构形式不能选取.因为孤立点已经不存在进一步细化的问题;一条线段没有"厚度",因此,不必细化;非边界点不属于被腐蚀或"剥落"的对象.所示结构元素如图 5 所示.

0	0	0
*	1	*
1	1	1

(B1)

0	0	0
*	1	*
1	1	1

(B2)

1	*	0
1	1	0
1	*	0

(B3)

1	1	1
*	1	*
0	0	0

(B4)

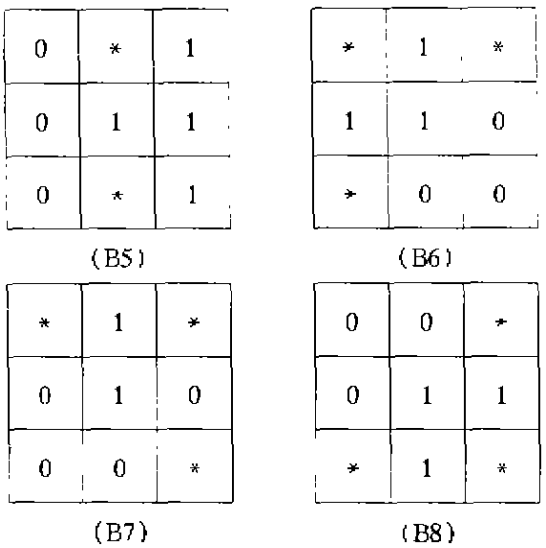


图 5　结构元素特征

Fig.5　The feature of structure elements

其中，符号“＊”表示即可取灰度 1，又可取灰度 0 的像素．然后将以上两种结构元素旋转 90 度、180 度、270 度得出 B3……B8 共 6 种结构元素，共有 8 种结构元素．这 8 种元素分别对应着东、西、南、北及东北、西北、西南及东南 8 个方向上的边界点．

1.2.2　构造细化步骤

结构元素确定以后，就可设计细化的运算方法．由前面的内容知道，形态学中的薄化运算可用来对图像进行细化，薄化运算为

$$A \bigcirc B = A - (A \circledcirc B)$$

$$A \circledcirc B = (A \ominus Ba) - (A \oplus \check{B}b)$$

其中 $B = Ba \cup Bb$，$Ba \cap Bb = \Phi$　$\check{B}b$ 为 Bb 的反射集（Bb 为图象 B 被点 b 平移的结果）．若 $Bb = \Phi$，即 B 中不包含 Bb 则

$$B = Ba \qquad A \circledcirc B = A \ominus Ba$$

$$A \bigcirc B = A - (A \ominus Ba)$$

在这里 Ba 为 B1、B2、B3、B4、B5、B6、B7、B8 共 8 种结构元素．式(6)是细化运算的主要依据．它实际上是薄化运算的一种“退化”形式．通常情况下采取的细化步骤为：

(1)将需要细化的目标图像 A 送入 Y　Y←A；

(2)对 $I = 1, 2, 3, 4, 5, 6, 7, 8$ 分别执行　$A = A \bigcirc Bi$，即分别用 B1～B8 所述的 8 种结构元素对 A 实行薄化运算各一次．

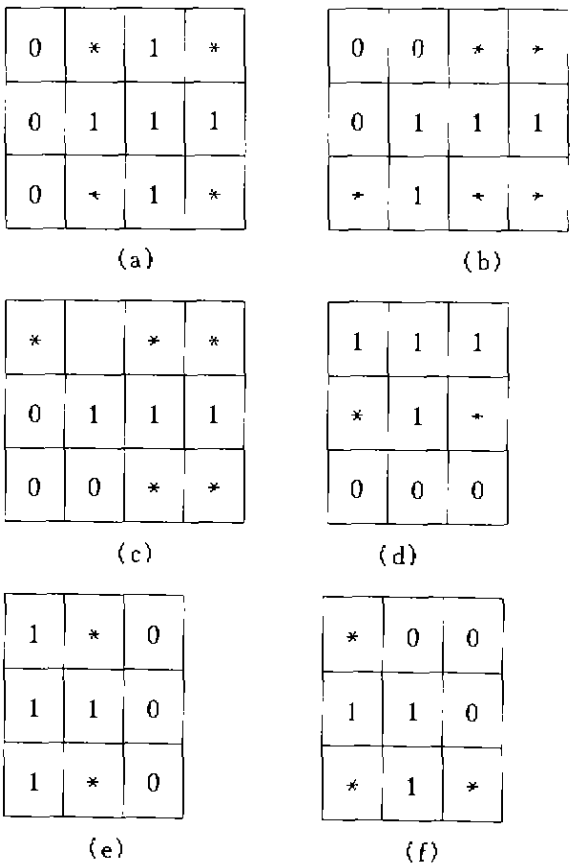
(3)若有条件 $A = Y$，迭代结束；否则执行步骤(4)．

(4)$Y \leftarrow A$，返回(2)．

从以上细化过程可以知道：步骤(2)可用 B1～B8 中结构元素分别对图像进行 $A \bigcirc B$ 运算，所以对图像的扫描次数较多；同时还存在着“运算浪费”现象，即当用某一结构元素执行 $A \bigcirc Bi$（$I = 1, 2 \cdots 8$）运算时，此结构元素只能对某一个方向上的像点作出是否删除的判断，对其它上的像点则无能力，但薄化运算还要持续到整幅图像结束，因而这种细化的处理速度不够理想．

为解决细化速度问题，把步骤(2)再作一分析．步骤(2)$A \bigcirc B$ 运算的实质主要是判断哪些点删除，那些点保留，即与结构元素 B 匹配的象素为边缘点，被删除；反之，则保留．因此，可以把 $B1 \sim B8$ 的 8 种结构元素并列起来，同步作用于某一二值图像，以检出边界点．如果某个像点及其邻点与 8 中结构元素之一相匹配就可以去除所有的边缘点（保护点除外），使用细化速度大大提高．

在这里还有一个需要注意的问题是：细化后图形的“厚度”为 1 个像素宽，但对被处理的数字图形来说，无论是沿着横向计算，还是沿着纵向计算，即可能是偶数行（列），也可能是奇数行（列）．如果是奇数行（列），细化后的图形正好处于原图形的中心；如果是偶数行（列），就会把最后二个像素宽的垂直线、水平线、甚至拐角处的像点删掉，这样细化后的图形就会出现断线现象，也就是说，细化后的图形不能保持原图形的连通性，不符合细化的基本准则．为此，需要对 B1～B8 的 8 中结构元素加以改进，改进后的 8 种结构元素如图 6 所示．



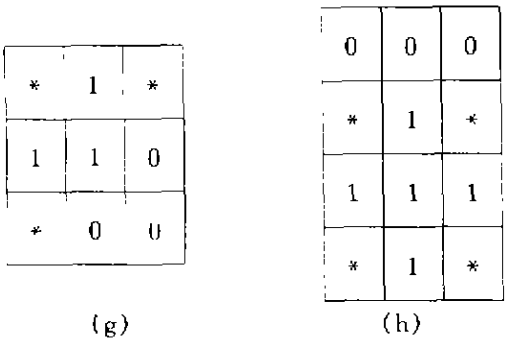


图6 结构元素特征

Fig.6 The feature of structure elements

用改造后的结构元素(a)～(h),对图1中的a图和b图进行细化处理.注意在步骤(2)中,8种结构元素是并行同步参与运算,图7为细化结果.

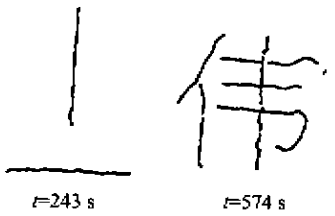


图7 数学形态学细化结果

Fig.7 The result of fine-grained processing of mathematical morphology

2 结果与讨论

从上述和各种细化方法中可以看到:经典的细化方法,无论是串行法,还是并行法,它们对判断某一象素是否边缘点,是否被删除,所用的判据甚多,如Hilditch的细化方法为6条,Deutch细化方法为5条,Resonfeld9-连接细化达10条之多.而用数学形态学的方法对二值图象进行细化,虽然要用到多个结构元素,但它们之间是相互独立并列的关系,每次只用到某中一个判据.也就是说,只要判断出图像中的某一象素及其邻点,与众多结构元素中的一个相匹配,那么该象素就是可以删除的边缘点,而不象其它方法那样需要几个条件同时满足才能删除.因此基于数学形态学的细化方法简单明了.

表1为Hilditch方法、Deutch方法、Resonfeld8-连接方法和数学形态学方法,对图1中两个不同的处理对象,分别进行细化的对比情况.

从表1可以明显看出这两种细化方法的处理速度存在着较大的差别,数学形态学的细化方法最快,比Hilditch快47%,比Resonfeld快53%,比Deutch快11%.(在P166机上,用C语言编程实现).

再从迭代次数来看,数学形态学方法也是最少的,且一次迭代所用时间(即细化时间除以迭达代次数)为21.7 s,而Hileitch方法为57.4 s,Rosenfeld方法为79.2 s,Deutch为21.5 s.也就是说,基于数学形态学的细化方法,扫描图像的速度较快.

表1 几种细化方法对比

Tab.1 Comprison of some fine-grained processing

细化方法	处理对象	处理时间/s	迭代次数	细化效果
Hilditch	1	564	10	B
	2	1 040	18	A
Deutch	1	329	16	A
	3	605	28	A
Resonfeld	1	630	8	D
	3	1 193	15	A
数学形态学	1	243	12	A
	3	574	27	B

从表1还可以看出:基于数学形态学的细化方法,对横线条图形或竖线条图形的处理速度,明显优于其它细化法,但当被处理图像中含有拐角时,处理速度减慢.

参考文献:

[1] 唐常表. 数学形态学方法及应用[M]. 北京:科学出版社,1990.

[2] 周新伦. 数学图象处理[M]. 北京:国防工业出版社,1990.

(责任编辑:朱 明)