

# 一种有效的二值图像细化算法

吕俊白

(华侨大学计算机科学系, 泉州 362011)

**摘 要:** 基于对图像中目标边缘的分析, 提出了一种行之有效的二值图像细化算法。实验证明, 采用该算法细化后所得的图像骨架不仅对称性好, 而且为单像素宽, 并能够完整地保持原图像的连通性。

**关键词:** 二值图像; 细化; 模板; 骨架

## An Efficient Thinning Algorithm for Binary Images

LV Junbai

(Department of Computer Science, Huaqiao University, Quanzhou 362011)

**【Abstract】** Based on analyzing the edges of the objects in an image, an efficient thinning algorithm for binary images is proposed. Experiments prove that the skeletons of the objects produced by this algorithm have good symmetry, one pixel wide, and can completely preserve the connectedness of the original image.

**【Key words】** Binary image; Thinning; Mask; Skeleton

图像的细化是指在保持原图像拓扑结构的情况下, 尽可能地抽出一个单像素宽的骨架的过程。图像细化是数字图像预处理中的重要一环, 无论是在图像分析还是在图像识别方面都起着重要作用。因此, 图像的细化算法研究一直受到图像处理领域的普遍关注。近年来, 许多学者相继提出了各种细化算法, 依考虑问题的角度不同, 图像细化的方法有两种, 即边缘点删除和内点保留。传统的基于边缘点删除的细化算法在细化过程中, 只对边缘点的可删除性进行判断并作相应处理, 由于受跟踪顺序及所考察邻域的影响容易产生骨架的非对称性; 基于内点保留的细化算法容易使所获得的骨架大于一个像素。本文在分析上述两类细化算法的基础上, 综合运用上述两种细化方法提出一种新的、行之有效的并行细化算法。

### 1 几个基本定义

对于一幅二值图像G, 设G中前景F上的像素记为1, 背景B上的像素记为0。为便于算法的叙述, 先给出以下几个基本定义:

**定义 1** 像素的4-邻点: 对像素(x,y), 将其上、下、左、右4个方向上点的集合称为像素(x,y)的4-邻点。

**定义 2** 像素的8-邻域: 对像素(x,y), 将其上、下、左、右4个像素点及其对角线上的4个像素点构成的集合称为像素(x,y)的8-邻域(如表1)。

表1 像素(x,y)的8-邻域

P3	P2	P1
P4	P	P0
P5	P6	P7

表2 8-邻域的16-环境

P14	P13	P12	P11	P10
P15				P9
P16		P		P8
P17				P23
P18	P19	P20	P21	P22

**定义 3** 8-邻域的16-环境: 8-邻域的四周围的点( $p_8 \dots p_{23}$ )的集合称为8-邻域的16-环境(如表2)。

**定义 4** 交叉数: 对F中的每一个点P, 交叉数定义为点P的8-邻域中灰度值满足条件(1)的像素点的点数之和。

$$g(P_k) - g(P_{k+1}) = 1 \quad (1)$$

式中  $0 \leq k \leq 7$ , 且运算时P的下标模8。

## 2 算法设计

### 2.1 算法的基本思想

任何一个细化算法一般都必须满足以下4点要求:

- (1) 骨架图像必须保持原图像目标边缘的连通性;
- (2) 骨架图像应尽可能为原图像目标边缘的中心线;
- (3) 细化结果应尽可能争取获得一个像素宽度的线条图像;
- (4) 细化速度要快。

据此, 我们提出了一种新的、行之有效的二值图像细化算法, 其基本思想是: 首先, 保留内点及图像中绝对不能被删除的特殊点(如: 交叉点, 拐角点, 等); 其次, 删除多余的像素点; 最后, 去除多余枝线。

### 2.2 算法的实现

第1步: 确定应保留的点。

前景中应保留的点包括以下4类:

- (1) 图像中目标的突出部分;
- (2) 图像中目标边缘的内点;
- (3) 图像中目标边缘的交叉点;
- (4) 细化到宽度为2之前难以判定是否删除的点。

设当前考察点为P(其灰度值为1)。定义P的4-邻域的灰度值之和为 $\Sigma 4$ ; P的8-邻域的灰度值之和为 $\Sigma 8$ ; P的8-邻域的16-环境的灰度值之和为 $\Sigma 16$ ; P点的交叉数为 $N_8$ 。

首先, 为保证细化后的图像能反映原图像中目标的突出部分, 对当前点P考察其8-邻域, 若P的8-邻域之和为1( $\Sigma 8 = 1$ ), 即当前点P的8-邻域仅存在单一一个灰度值为1的点, 则原则上保留该像素点。

其次, 考察当前点P的4-邻点, 若4-邻点的灰度值均为1(即 $\Sigma 4 = 4$ ), 则可断定它是目标边缘的内点应予保留; 接着考察点P的8-邻域, 保留4-邻点之和为1(即 $\Sigma 4 = 1$ )以及交叉数 $N_8$ 大于1(即 $N_8 > 1$ )的点, 以保证细化后的图像的连通性。

基金项目: 福建省自然科学基金资助项目(A0010013)

作者简介: 吕俊白(1969—), 女, 硕士、讲师, 主要研究方向: 数字图像处理, 多媒体技术等

收稿日期: 2002-10-11

最后, 为保留那些细化到宽度为2之前难以判定是否删除的点, 在设计中我们采用图1中的模板进行检测, 若两模板的输出结果之一为1(真), 则保留当前点以确保当边缘细化到双线时不致断裂。

模板 (a)
模板 (b)

图1 宽度为2之前检测模板

为了保证图像连通性的同时, 实现骨架上像素点之间的单点连通, 在设计中我们使用了图2所示的两个模板进行检测, 若模板的输出为1(真), 则删除当前像素点P.

0	1	x
1	P=1	0
x	0	0

x	1	0
0	P=1	1
0	0	x

模板 (a)                      模板 (b)

图2 检测模板

归纳起来, 上述细化算法的具体实现步骤如下:

(5)对于当前点若图3两模板之一输出为真，则保留之，否则，删除当前点：

(3)判断是否扫描完所有像素点, 若否则转(1)。

### 3 实验结果

在实验中我们运用本算法对大量各类二值图像进行细化处理均取得满意的结果。图3为原图，图4为细化后所得的骨架图像。实验表明，该算法在细化的同时能很好地保持原图像的连通性，所得的骨架图像对称性好且为单像素宽。



图3 原图



图4 细化后的骨架图像

## 参考文献

- 1 Kenneth R.Castleman. Digital Image Processing.北京:清华大学出版社,1998-04
- 2 李侠民徐美瑞两个快速的完全的并行细化算法计算机研究与发展,1996,33(7):521-527
- 3 谭柏珠,叶邦彦.一种适用于工程图的快速细化算法.微型机与应用,1999,(10):11-12
- 4 张远鹏,董海,周文灵编著.计算机图像处理技术基础北京:北京大学出版社,1996-09

## 参考文献

- 1 Parsa C, Garcia-Luna-Aceves J J. Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media. In Proceedings of IEEE ICNP '99, Toronto, 1999-10
- 2 C áceres R, Iftode L. The Effects of Mobility on Reliable Transport Protocols. In Proceedings of the 14th International Conference on Distributed Computing Systems, 1994-06
- 3 Balakrishnan H, Padmanabhan V, Seshan S, et al. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. In IEEE/ACM Transactions on Networking, 1997-12