1994年5月 第17卷第3期 重庆大学学报 JOURNAL OF CHONGQING UNIVERSITY Vol. 17, №. 3 May. 1994

# 一种快速的一次扫描细化算法(OSSTA)*

## A Fast One-Step Scanning Thinning Algorithm

吴晓芸 Wu xiaoyun　张太怡 Zhang Taiyi　刘平 Liu Pin

(重庆大学电子信息工程学院,重庆,630044)

**摘 要** 提出了一种快速的一次扫描细化算法,通过骨架点的预测分析,得到骨架点,对非骨架点的像素不作判断,并通过插补连接得到骨架.另给出了细化实例。

**关键词** 图像细化;连通性;骨架;内接圆;邻域

**中国图书资料分类法分类号** O235

**ABSTRACT** A new quick thinning algorithm was presented in this paper, which assigned *different weights to different pixels near the skeleton pixels and rules out the impossible pixels in* the thinning procedure. It needs neither iteration on pixel panel nor detection on every pixel such that the time consumed is greatly lessened. A filling algorithm is also offered for better results. Experiment on variety of binary patterns showed that it could get both a high speed and a good skeleton shape compared with other algorithms. This new algorithm reaches perfect result.

**KEYWORDS** thinning; connectivity; skeleton; inscribed-circle; neighbor

## 0 引　言

在图像处理和模式识别中,细化算法被广泛地使用和研究。细化即是寻找有宽度的图像的骨架,以节省存储空间,利于特征的提取,简化图像形式。

有三种方法可以得到图像模式的骨架。第一种是在保持图像连通性不变的条件下,逐点剥去二值图像的边界点,如 Hildth 算法[1]。这种方法往往经过若干次迭代运算才得到图像的骨架,速度比较慢。第二种方法从内像素点到边界点进行判断,找到中心的点,如 MS (maximal squares)和 MC(maximal circles)[5],但在每个像素的判断上条件是一样的,所以速度提高不大。第三种方法是在每条扫描线上寻找模式分支的中点,然后连接成骨架。这种方法简单,速度快,但对干扰敏感。我们提出的算法特点是只判断可能是骨架点的像素,通过插补连接而得到骨架。试验证明,本方法适合于二值化的文字等模式处理。

## 1 一次扫描细化算法(OSSTA 法)

* 收文日期 1992-11-02

### 1.1 检测方向

这里以二值图像的文字为例。文了以笔划构成。假设一个大小可变的圆在笔划中滚动，则圆心的轨迹必包含于该图像原骨架中，这就是我们寻找骨架点的基本思想。在笔划交叉处找不到内接圆，则用插补的方法进行连接。象 MS 和 MC 方法[5]一样，我们从内像素点判断到边界点，但不是判断方块或圆中所有像素，而只判断8个方向上的像素，如图1所示的方向。

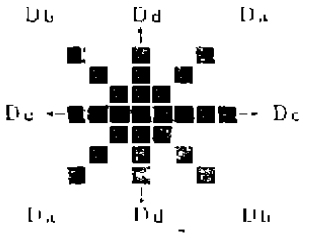


图 1 像素点判断方向

### 1.2 权数的确定

我们把图像的像素点分为三种情况，即骨架点，不必判断的点以及无标志点。我们逐行扫描像素，如果像素是不必判断的点或已是骨架点的话，则进行下一点的判断；若像素是无标志的点则进行基于内接圆原理的细化算法，判断该点是否是骨架点，如果是骨架点并且在8邻域内还存在另一个骨架点的话，则根据骨架的走向对8个方向上的像素进行权数的计算，然后由权数的大小对相应的像素进行细化算法，判断相应的点是不是骨架点，并确定不必判断的点。权数的大小根据骨架走向的正切值来定，如图2所示：

(a) 预测标志图　　(b)67.5° ≤ α < 90° 的预测图　　(c)22.5° ≤ α < 45° 的预测图

$W_v=2$ $W_w=3$ $W_x=1$　　$W_v=2$ $W_w=3$ $W_x=1$

图 2 骨架点预测分析

●：正切计算点　⊙：当前保留骨架点　■：保留骨架点

▣：不必判断点　□：不保留或未检测到的点

因为扫描是从左上角到右下角，所以下一个骨架预选点的象素是 $U,V,W,X,Y$，每次根据细化线走向考虑三个预选点。(1) 式所示为预选集合表达式。这里，$\alpha$ 是细化线走向与水平方向的夹角，集合中的元素按权的大小顺序排列。

$$CS=\begin{cases}\{UVW\} & \text{如果}\quad 0^\circ \leqslant \alpha < 22.5^\circ\\ \{VUW\} & \text{如果}\quad 22.5^\circ \leqslant \alpha < 45^\circ\\ \{VWU\} & \text{如果}\quad 45^\circ \leqslant \alpha < 67.5^\circ\\ \{WVX\} & \text{如果}\quad 67.5^\circ \leqslant \alpha < 90^\circ\\ \{WXV\} & \text{如果}\quad 90^\circ \leqslant \alpha < 112.5^\circ\\ \{XWY\} & \text{如果}\quad 112.5^\circ \leqslant \alpha < 135^\circ\\ \{XYW\} & \text{如果}\quad 135^\circ \leqslant \alpha < 157.5^\circ\\ \{YXW\} & \text{如果}\quad 157.5^\circ \leqslant \alpha < 180^\circ\end{cases} \tag{1}$$

### 1.3 确定不必判断点

规则如下：

规则 1，如果　$CS = \{WVX\}$ 或$\{WXV\}$

则　按$W,V,X$或$W,X,V$的顺序求下一个骨架点，并确定不判断点为'$U$'和'$Y$'方向(即水平方向)上的内接圆范围内的点；

规则 2，如果　$CS = \{UVW\}$ 或$\{YXW\}$

则　按$U,V,W$或$Y,X,W$的顺序求下一个骨架点，并确定不判断点为'$W$'方向(即垂直方向)上的内接圆范围内的点；

规则 3，如果　$CS = \{VWU\}$ 或$\{VUW\}$

则　按$V,W,U$或$V,U,W$的顺序求下一个骨架点，并确定不判断点为'$X$'方向(即斜率为 1 方向)上的内接圆范围内的点；

规则 4，如果　$CS = \{XWY\}$ 或$\{XYW\}$

则　按$X,W,Y$或$X,Y,W$的顺序求下一个骨架点，并确定不判断点为'$V$'方向(即斜率为 $-1$ 方向)上的内接圆范围内的点；

这里，内接圆范围由算法中得到当前保留点的迭代次数来确定。

图 3 中用□来表示不必判断的点；用◆和◎表示用基于内接圆的细化算法未得到的骨架点。

图 3　确定不判断点的规则说明

### 1.4 插补算法

用基于内接圆原理的细化算法往往在笔划的交叉部分或极不光滑处得不到骨架点，这时可根据连通性进行插补。

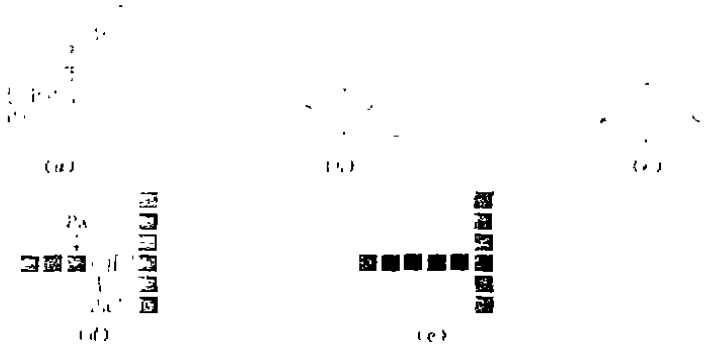

图 4(a)　插补原理图　　(b)　向上插补方向图　　(c)　向下插补方向图

(d)　插补前骨架图　　(e)　插补后骨架图

如图 4 所示，$P_n$是一细化线的终点，插补时考虑$P_n$邻近的 5 个点，取它们中多数的走向为插补的方向，如补上了另一个$P_n$点。

若插补方向是如图 5 的五个方向，则沿细化线的走向连续补完该细化线上所有的点，直

到插补的点到了边界或与别的细化线相交为止。

若插补方向是如图 6 的三个方向，则每补一点都要检查下一行的情况。若下一行有要插补的则补，若没有，则可以再补一点，以此类推。图 7，8 为插补前后的例子。

### 1.5 细化算法原理

在描述算法前，先作如下定义。

| $P_1$ | $P_2$ | $P_3$ |
|---|---|---|
| $P_4$ | $P_0$ | $P_5$ |
| $P_6$ | $P_7$ | $P_8$ |

* 像素是 4 — 邻域内的边界点，如果

$$\sum_{i=1}^{4} P_{2i-1} \leqslant 3 \tag{2}$$

* 像素是4—邻域内的内点，如果

$$\sum_{i=1}^{4} P_{2i-1} = 4 \tag{3}$$

* 像素是8—邻域内的内点，如果

$$\sum_{i=1}^{8} P_i = 8 \tag{4}$$

算法流程图如下：

## 2 试验结果

我们用本文提出的 OSSTA 算法对一些模式进行了试验，并和[2]进行了比较，表1列出了几个模式用两种方法细化所用的时间，图6比较了两种方法得到的细化结果。

算法用 Turbo C 2.0编制，在 IBM PC/286上运行。

表 1 细化时间的比较 s

| 模 式 | 汉字 | “的”字母 | “B”字母 | “X” |
|---|---|---|---|---|
| Zhang and Suen's 算法 | 0.32 | 0.10 | 0.21 | |
| OSSTA 算法 | 0.10 | 0.03 | 0.03 | |

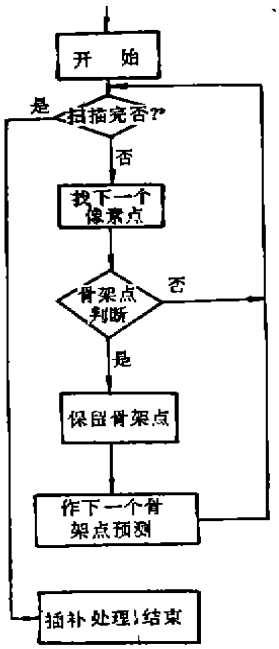


图 5 算法流程图

## 3 结 论

提出的一次扫描细化算法适于二值图像处理，运行算法所要时间短，一次扫描完后即可行到细化骨架。试验结果表明，本算法在运算速度和骨架精度方面取得了很好的效果，故在模式识别和图像处理中有很好的应用前景。

图6的上半部是 OSSTA 算法结果，下半部是 Z. S. TA 算法结果。

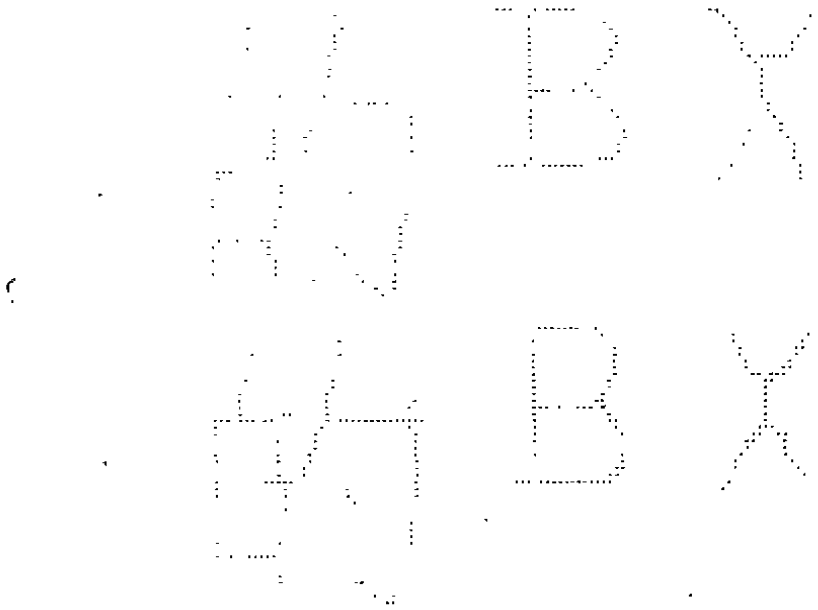

图 6　OSSTA 和 Z.S.TA 的比较

## 参 考 文 献


1 Lam L and Sue C Y. A dynamic shape pdreserving thinning algorithm. Signal Processing, 1991,22(2),199～208

2 Zhang T Y and Suen C Y. A fast parallel algorithm for thinning digital patterns. Comm. ACM,1984,27:236～239

3 Li Bei and Suen C Y. A knowledge-based thinning algorithm. Pattern Recognition, 1991,24(12),1211～1221

4 Yu S S and Tsai W H. A new thinning algorithm for gray-scale images by the relaxation technique. Pattern Recognition, 1990,23(10):1067～1076

5 Abairid M R, Martin G and Suen C Y. Tracing Center-lines of Digital Patterns Using Maximal-Square and Maximal-Cricle Algorthms. Conf. on Vision Interface '90, Nova Scotia, Canada, 1990,76～79

6 Holt C M, Stewart A, Clint M and Perrott R H. An improved parallel thinnig algrithm. Comm. ACM,1987,30(2):156～160

7 Zhang Y Y and Wang P S P. A maxmum algorithm for thinning digital patterns. Proc. 9th Int. Conf. Patt. Recogn., Rome, Italy, 1988, 942～944

8 Hall R W. Fast parallel thinning algorithms. Parallel speed and connectivity pareservation. Comm ACM, 1989, 32(1):124～131

9 Wang P S P, Hui L W and Fleming T. Further improved fast parallrl thinning algorithm for digital patterns. Computer Vision, Image Processing and Communications-systems and Applications, Singgapore, World Scientific, 1986, 37～40

10 Lam L, Lee S W and Suen C Y. Thinning Methodologies- A Comprehensive Survey. IEEE Trans. Patt. Anal. Machine Intell. ,1992,14(9):869～885