

改进的字符图像细化算法

韩建峰, 宋丽丽
(内蒙古工业大学信息工程学院 呼和浩特 010080)
(hj2003@tom.com)

摘 要: 针对现有并行细化算法中存在的细化不完全、二像素宽斜线细化畸变等问题, 提出一种改进的细化算法. 首先对二值图像构造了一组保留模板和消除模板, 采用算术逻辑运算对图像进行初步细化, 并结合保留模板匹配将二像素宽斜线予以保留; 再利用消除模板匹配进一步删除斜线上的冗余像素, 得到 8 连接的单像素图像. 针对字符图像的实验结果表明, 文中算法能够有效地避免二像素宽斜线的细化畸变, 保持了原图像的拓扑结构, 实现图像的完全细化.

关键词: 细化; 并行算法; 模板匹配; 二像素宽斜线

中图法分类号: TP391

An Improved Thinning Algorithm for Character Image

Han Jianfeng and Song Lili
(School of Information Engineering, Inner Mongolia University of Technology, Huhhot 010080)

Abstract: In order to overcome the shortcomings of existing parallel thinning algorithms, such as thinning incompletely and 2-pixels slash thinning distortion, an improved thinning algorithm is proposed. A set of reservation templates and elimination templates for binary image were constructed firstly. The elementary thinning process was performed by arithmetic and logic operation, while 2-pixels slash was reserved by reservation templates matching. Then the redundancy pixels on slash were deleted by elimination templates matching, and the 8-connected single-pixel image was acquired. Experimental results on character images show that, the proposed algorithm could effectively avoid the 2-pixels slash thinning distortion, maintain the topological structure of original image, and achieve complete thinning result.

Key words: thinning; parallel algorithm; template matching; 2-pixels slash

“骨架”是人们对于字符的抽象认识, 字符的绝大部分信息集中在字符的骨架上. 对二值化目标图像进行细化处理后得到目标字符的细线化图像, 即线宽为单像素, 可形成字符的“骨架”, 能够较容易地提取图像的特征. 因此, 细化效果的好坏直接影响字符识别的速度和准确性, 良好的细化算法对字符的准确识别起着十分重要的作用. 细化算法广泛地应用于图像分析、信息压缩、特征提取及模式识别等.

好的细化算法应满足以下要求:

- 1) 细化图像的连通性必须与原图像保持一致, 保留完整的特征信息;
- 2) 细化后的“骨架”尽可能是原图像的中心线;
- 3) 细化图像应尽可能是单像素, 即完全细化;
- 4) 细化算法的速度应尽可能快.

在图像处理中, 研究应用相对较短的细化时间, 获得完全细化的效果是非常具有实际意义的.

收稿日期: 2012-02-17; 修回日期: 2012-09-24. 基金项目: 内蒙古工业大学科研基金(200602). 韩建峰(1978—), 男, 硕士, 讲师, 主要研究方向为信号与信息处理; 宋丽丽(1980—), 女, 硕士, 讲师, 主要研究方向为信号与信息处理.

现存的细化方法有很多种,文献[1]归纳和分析了多种经典方法,大体上将其分为迭代法及非迭代法,迭代法又分为并行细化和串行细化算法.文献[2]对 15 种算法进行了评测,其中既有扫描算法又有模板匹配算法.从文献[1-2]中看到,众多细化算法各有优缺点,在特定应用中能满足要求的也有多种.ZS 细化算法^[1,3,4]采用算术逻辑运算,是一种典型的并行细化算法,其最突出的优点是对直线、T 型交叉和拐角能够比较精确地保持和原图像一致,同时兼顾图像连续性;其迭代次数少,运行速度快,是目前在字符图像处理中应用最为普遍的算法之一.基于模板匹配的算法广泛应用于指纹图像处理等领域,近年来也出现了很多对经典算法的改进方法,如改进图像模板算法^[5]、改进的一步化细化算法^[6]等.

1 现有细化算法分析

ZS 算法是典型的并行细化算法,它对目标图像进行逐像素扫描,其 8 邻域表示如图 1 所示.

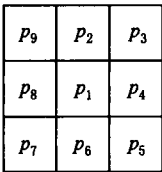


图 1 8 邻域示意图

图 1 中,像素值为 1 的点称为轮廓点或前景点,像素值为 0 的点称为背景点.该算法对轮廓点像素的 8 邻域分步进行算术逻辑运算,依据运算结果判断该像素是否应该删除.其细化过程分为 2 个步骤:

Step1. 对于值为 1 的任意像素 p_1 ,若其 8 邻域点满足以下条件

- a) $2 \leq N(p_1) \leq 6$,
- b) $S(p_1) = 1$,
- c) $p_2 \cdot p_4 \cdot p_6 = 0$,
- d) $p_4 \cdot p_6 \cdot p_8 = 0$;

则标记轮廓点 p_1 为待删除,待本次扫描结束后,将标记的像素删除.其中, $N(p_1)$ 为 p_1 的 8 邻域中非零点的个数,即 $N(p_1) = p_2 + p_3 + \cdots + p_8 + p_9$; $S(p_1)$ 为 $p_2, p_3, \cdots, p_8, p_9$ 顺时针旋转时像素值由 0 变 1 的次数.

Step2. 若 p_1 满足条件 a) 和 b) 以及

- c') $p_2 \cdot p_4 \cdot p_8 = 0$,
- d') $p_2 \cdot p_6 \cdot p_8 = 0$;

同样标记 p_1 待删除,并在本次扫描结束后删除标记的像素.

在 Step1 和 Step2 中分别将标记的轮廓点删除,并反复迭代,直至目标图像中没有可以删除的

点,则细化处理结束,即可生成目标图像的骨架.经过 ZS 算法细化处理后,字符图像的骨架特征信息保存完整,连续性好,且基本为单像素.

ZS 算法的缺点是对斜线的处理不够理想,在斜线上存在冗余像素.另外,对二像素宽斜线的处理也是其缺陷.如图 2 所示,对于二像素宽的斜线,ZS 算法将其细化为单像素短直线,细化结果中出现了丢失特征信息的现象,即所谓的二像素斜线细化畸变.因此,在 ZS 算法基础上有很多基于某种应用的改进和修正,如 LW 细化算法^[1,7]和增强并行细化算法(enhanced parallel thinning algorithm,EPTA)^[7].



图 2 ZS 算法的二像素斜线细化畸变

为解决二像素斜线细化畸变的问题,LW 算法将 ZS 算法的条件 a) 限制放宽,改为 $3 \leq N(p_1) \leq 6$.这样确实可以解决二像素斜线细化畸变的问题,但同时会使冗余像素过多,未能完全细化.EPTA 算法将扫描分为 2 个阶段,第一阶段扫描与 ZS 算法相似,只是针对二像素斜线增加了一个判决条件,即若所有标记删除的像素满足 $N(p_1) = 2$,则将该点保留;在第二阶段利用 2 个条件^[7]删除斜线上的冗余像素.该算法能够在避免二像素斜线细化畸变的同时,减少细化冗余,提高细化效果;但是其解决细化畸变的方法具有局限性.如在针对单幅斜线图像细化时可以避免畸变,如果图像中存在其他轮廓点且所需迭代次数高于纯粹的二像素斜线时,判决条件将失效,会产生斜线细化畸变.如图 3 所示包含二像素斜线和字符“0”的图像,其在细化后出现了二像素斜线细化畸变现象.



图 3 EPTA 算法的斜线细化畸变

同时,由于 EPTA 算法 2 个删除条件的约束过少,第二阶段扫描不能将冗余像素完全删除,从而导致部分图像的细化效果存在细化不完全以及字符末端过长的现象出现.

文献[8]对于符号图像就 ZS 算法进行了改进, 算法速度在一定程度上有所提高, 但同样未能实现完全细化. 文献[9-10]所提出的均是基于 ZS 算法、结合图像模板匹配的改进方法. 文献[9]算法主要利用 ZS 算法的条件及其镜像条件完成细化, 缺点是无法根除 ZS 算法的斜线冗余; 同时该算法也提出了抑制二像素斜线细化畸变的保留模板, 但模板相对简单, 可能会出现保留过度的现象. 文献[10]算法对采用的消除模板和保留模板进行了完善, 可以删除 ZS 算法处理后斜线的细化冗余, 但未对可能出现的二像素斜线细化畸变加以抑制.

由此可见, 在字符识别应用中, 现有算法或者不能完全细化, 或者增加了迭代次数, 但对于细化效果的改善并不明显, 缺乏快速而且能够完全细化的算法.

本文通过构造一组保留模板和删除模板, 对 ZS 算法进行改进, 在保留经典 ZS 算法优点的同时, 着重解决其斜线存在冗余像素及二像素斜线细化畸变的问题.

2 改进的细化算法

2.1 算法思想

在 ZS 算法的细化结果中, 直线、T 型交叉和拐角细化精确, 能够与原图像保持一致, 仅在斜线位置存在少量冗余像素, 不够平滑; 但其在字符识别的后期处理中会造成虚假的特征信息, 影响字符特征的编码和匹配, 增加识别的难度, 甚至识别错误. 因此, 完全细化的关键在于 ZS 算法细化后删除位于斜线上的冗余像素.

图 4 所示为 ZS 算法细化处理字符‘2’后的放大效果图, 可以看到, 斜线上残留着少量冗余像素. 出现斜线冗余的根本原因是 ZS 算法中条件 b) 的存在, 冗余像素并不满足条件 b), 所以不会被标记删除, 在多次扫描后它仍然留在骨架中.

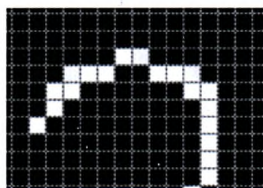


图 4 ZS 算法处理结果中的斜线冗余

本文提出 4 个消除模板, 分别用于消除 4 个方向斜线上的冗余像素, 如图 5 所示, \times 表示任意像素

值. 在用 ZS 算法细化扫描后, 利用消除模板对字符图像进行匹配, 将斜线的冗余像素删除.

\times	1	\times
0	p_1	1
0	0	\times

0	0	\times
0	p_1	1
\times	1	\times

\times	0	0
1	p_1	0
\times	1	\times

\times	1	\times
1	p_1	0
\times	0	0

图 5 消除模板

对于 ZS 算法中存在的二像素斜线细化畸变的问题, 本文采用构造保留模板的方法有针对性地保留二像素斜线. 在构造保留模板时, 若约束条件不够, 即保留过度, 就会出现细化末端过长的问题. 其原因是在字符拐角局部及末端局部有类似于二像素斜线的情况, 保留模板错误地将其保留, 后续处理过程未能全部删除冗余像素. 本文提出了如图 6 所示的保留模板, 能够避免二像素斜线细化畸变, 同时改善细化效果.

0	0	0	0
0	p_1	1	0
0	0	1	1

0	0	0	0
0	1	p_1	0
1	1	0	0

0	0	1	1
0	p_1	1	0
0	0	0	0

1	1	0	0
0	1	p_1	0
0	0	0	0

图 6 保留模板

2.2 本文算法流程

Step1. 扫描目标图像, 若轮廓点 p_1 满足条件 a)~d), 同时不满足保留模板, 则标记之, 本次扫描结束后, 将标记像素删除.

Step2. 扫描目标图像, 若轮廓点 p_1 满足条件 a) b) c') d'), 同时不满足保留模板, 则标记之, 本次扫描结束后, 将标记像素删除.

Step3. 判断删除标记, 若有可删除像素, 转 Step1; 若无可删除像素, 执行下一步.

Step4. 再次扫描目标图像, 若轮廓点 p_1 的 8 邻域与消除模板中的任意一个匹配, 则将该像素删除.

Step5. 判断删除标记,若有可删除像素,转 Step4;若无可删除像素,细化结束.

2.3 算法分析

改进算法对目标图像的扫描分为 2 个子过程. 第一个扫描子过程首先使用 ZS 算法的 4 个条件对冗余像素进行判断,然后利用构造的 4 个保留模板对该像素进一步判别,以避免二像素斜线细化畸变. 若该像素满足任意保留模板,则不作标记;若全部模板均不满足则标记,并在本次扫描后将全部标记像素删除. 改进算法没有直接修改 ZS 算法的约束条件,而是在其逻辑运算完成后增加了是否删除的判别过程. 这样做虽然会增加少许处理时间,但优点非常突出,既避免了二像素斜线细化畸变,又可避免放宽 ZS 算法条件约束而留下过多冗余像素.

在第二个扫描子过程中,将图像分别与 4 个删除模板进行匹配,4 个消除模板分别对应删除斜线左下方、左上方、右上方及右下方的冗余像素;处于斜线骨架外的冗余像素会被删除,直至完全细化.

由于在第一个扫描子过程之后剩余的斜线冗余像素很少,因此实际上与 ZS 算法相比,增加的第二个扫描子过程所需的迭代次数很少,通常只需 2 次迭代即可,从而保证了改进算法的快速性.

图 7 所示为改进算法对二像素斜线以及包含“0”字符的二像素斜线细化效果图. 由图 7 可见,本文的保留模板对于抑制斜线细化畸变更具普遍性.

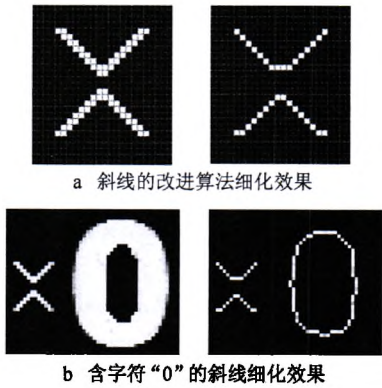


图 7 二像素斜线的改进算法细化效果

3 实验结果与分析

为验证本文改进细化算法的实际效果,在 i3 2.67 GHz, 2GB 内存的 PC 机、MATLAB 环境下应用 ZS 算法、EPTA 算法和改进算法分别对 100 幅 22×28 像素字符图像(含字母及数字)进行细化处

理实验,结果如表 1 所示,并以其中部分实验结果为例进行分析.

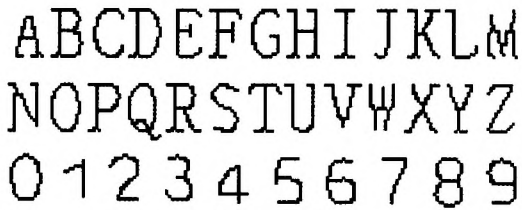
表 1 实验结果

算法	字符图像数	细化时间/s	细化效果
ZS 算法	100	1.05	斜线有冗余
EPTA 算法	100	1.06	部分有冗余、末端长
改进算法	100	1.37	完全细化

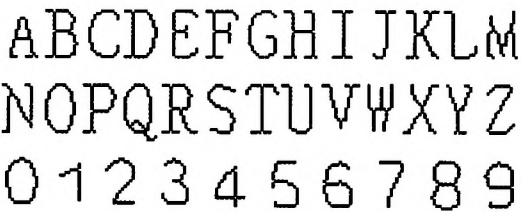
图 8 所示分别为 26 个大写英文字母和 10 个阿拉伯数字的原图像以及 ZS 算法、EATA 算法和本文改进算法的细化效果.



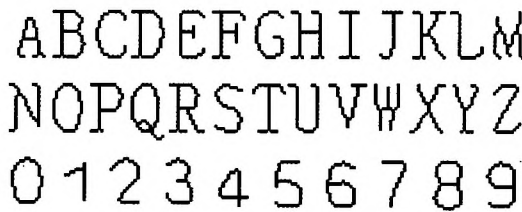
a 原字符图像



b ZS算法细化效果



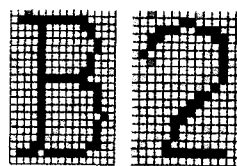
c EPTA细化效果



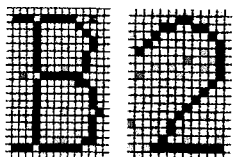
d 改进算法细化效果

图 8 字符图像细化实验示例

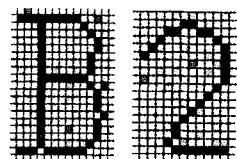
显然,ZS 算法和 EPTA 算法都存在细化不完全的现象. 以字符“B”,“2”为例,它们的细化效果放大图如图 9 所示. ZS 算法在斜线上留有冗余像素;EPTA 算法对斜线上的冗余像素有明显改善,但也不是单像素,在个别位置仍存在冗余像素;而改进算法则不存在上述问题,得到了保持 8 连接的、完全细化的字符骨架.



a ZS算法细化效果放大图



b EPTA算法细化效果放大图



c 改进算法细化效果放大图

图 9 细化效果放大图

改进算法较 ZS 算法和 EPTA 算法的处理时间稍长,增加的处理时间主要是用于保留二像素斜线,以避免细化畸变以及删除斜线上的残留冗余像素。

如图 10 所示,本文改进细化算法对于汉字的细化效果同样显著。由图 10 可见,对于汉字中经常出现的斜线,ZS 算法和 EPTA 算法细化后均有冗余像素,而本文改进算法能够完全细化至单像素,且保留了完整的特征信息。

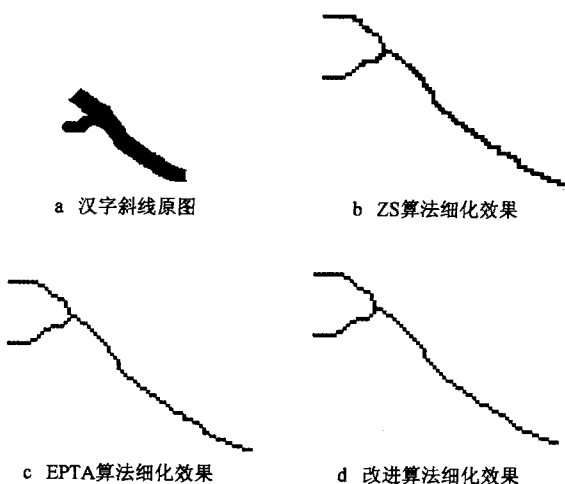


图 10 汉字斜线细化实验示例

4 结 论

本文在对现有并行细化算法的深入研究基础上,针对现有算法不能完全细化和二像素斜线细化畸变等问题,基于经典的 ZS 算法,提出了改进的字

符细化算法。实验证明,改进算法增加了少许处理时间,实现了对字符图像的完全细化,细化结果满足 8 连接条件,完整地保留了字符图像的特征信息,是一种较为快速、实用的细化算法。

参考文献 (References):

- [1] Lam L., Lee S W., Suen C Y. Thinning methodologies—a comprehensive survey [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(9): 869–885
- [2] Vincze M., Kovári B. Comparative survey of thinning algorithms [OL]. [2012-02-17]. [http://uni-obuda. hu/ conferences/cinti2009...](http://uni-obuda.hu/conferences/cinti2009...)
- [3] Zhang T Y., Suen C Y. A fast thinning algorithm for thinning digital patterns [J]. Communications of ACM, 1984, 27(3): 236–239
- [4] Gonzalez R C., Woods R E. Digital image processing [M]. 2nd ed. Beijing: Publishing House of Electronics Industry, 2002: 650–653
- [5] Wang Jialong, Guo Chengan. An improved image template thinning algorithm [J]. Journal of Image and Graphics: A, 2004, 9(3): 297–239 (in Chinese)
(王家隆, 郭成安. 一种改进的图像模板细化算法[J]. 中国图象图形学报: A 版, 2004, 9(3): 297–301)
- [6] Mei Yuan, Sun Huaijiang, Xia Deshen. An improved template-based rapid thinning algorithm [J]. Journal of Image and Graphics, 2006, 11(9): 1306–1311 (in Chinese)
(梅 园, 孙怀江, 夏德深. 一种基于改进后模板的图像快速细化算法[J]. 中国图象图形学报, 2006, 11(9): 1306–1311)
- [7] Bao Jianjun, Fan Jing. Robust parallel thinning algorithm for binary images [J]. Computer Aided Engineering, 2006, 15(4): 43–46 (in Chinese)
(包建军, 樊 菁. 鲁棒的二值图像并行细化算法[J]. 计算机辅助工程, 2006, 15(4): 43–46)
- [8] Yu Qingcang, Su Bin, LI Huaqiang. Improved parallel thinning algorithm of ideograph image [J]. Computer Engineering and Design, 2009, 30(3): 723–762 (in Chinese).
(喻擎苍, 苏 斌, 李华强. 改进的符号图像并行细化算法[J]. 计算机工程与设计, 2009, 30(3): 723–724)
- [9] He Jigang, Yang Xiaowei, Wu Guangchao. Fast parallel thinning algorithm based on reserved templates [J]. Computer Applications and Software, 2007, 24(12): 26–28 + 103 (in Chinese)
(贺继刚, 杨晓伟, 吴广潮. 基于模板保留的快速并行细化算法[J]. 计算机应用与软件, 2007, 24(12): 26–28+103)
- [10] Wang Peng, Zhang Youguang, Zhang Shuo. A synthetically thinning algorithm for fingerprint image [J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(2): 179–182 + 189 (in Chinese)
(王 朋, 张有光, 张 烁. 指纹图像细化的综合化算法[J]. 计算机辅助设计与图形学学报, 2009, 21(2): 179–182 + 189)

作者: 韩建峰, 宋丽丽, Han Jianfeng, Song Lili
作者单位: 内蒙古工业大学信息工程学院 呼和浩特010080
刊名: 计算机辅助设计与图形学学报 ISTIC EI PKU
英文刊名: Journal of Computer-Aided Design & Computer Graphics
年, 卷(期): 2013, 25(1)

参考文献(10条)

1. [Lam L;Lee S W;Suen C Y Thinning methodologies-a comprehensive survey](#)[外文期刊] 1992(09)
2. [Vincze M;Kovdri B Comparative survey of thinning algorithms](#) 2012
3. [Zhang T Y;Suen C Y A fast thinning algorithm for thinning digital patterns](#) 1984(03)
4. [Gonzalez R C;Woods R E Digital image processing](#) 2002
5. [王家隆;郭成安 一种改进的图像模板细化算法](#)[期刊论文]-[中国图象图形学报A](#) 2004(03)
6. [梅园;孙怀江;夏德深 一种基于改进后模板的图像快速细化算法](#)[期刊论文]-[中国图象图形学报](#) 2006(09)
7. [包建军;樊菁 鲁棒的二值图像并行细化算法](#)[期刊论文]-[计算机辅助工程](#) 2006(04)
8. [喻擎苍;苏斌;李华强 改进的符号图像并行细化算法](#)[期刊论文]-[计算机工程与设计](#) 2009(03)
9. [贺继刚;杨晓伟;吴广潮 基于模板保留的快速并行细化算法](#)[期刊论文]-[计算机应用与软件](#) 2007(12)
10. [王朋;张有光;张烁 指纹图像细化的综合化算法](#)[期刊论文]-[计算机辅助设计与图形学学报](#) 2009(02)

引证文献(1条)

1. [陈梅. 王健 图像并行细化算法改进研究](#)[期刊论文]-[中国印刷与包装研究](#) 2014(1)

引用本文格式: 韩建峰, 宋丽丽, Han Jianfeng, Song Lili 改进的字符图像细化算法[期刊论文]-计算机辅助设计与图形学学报 2013(1)