

一种基于改进后模板的图像快速细化算法

梅园 孙怀江 夏德深

(南京理工大学计算机系, 南京 210094)

摘要 为了快速地进行图像细化,提出了一种新的图像快速细化算法,该算法提出了组合模板的概念,不仅有效地提高了模板的匹配速度,同时对组合模板进行了优化,最后在优化后的组合模板基础上,提出了新的改进算法。大量实验表明,该新算法不但同时具有以往算法的优点,而且细化速度比以往算法提高了3~6倍,是一种较为理想的细化算法。

关键词 细化 指纹 图像处理 模板匹配

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2006)09-1306-06

An Improved Template-based Rapid Thinning Algorithm

MEI Yuan, SUN Huai-jiang, XIA De-shen

(Computer department of Nanjing University of Science & Technology, Nanjing 210094)

Abstract A new improved thinning algorithm is proposed in the paper. In the new algorithm, we advance a series of combination-templates which reinforce the speed of templates matching, and then optimize the combination templates. Finally based on the optimized templates, we bring forward the new algorithm. After doing lots of experiments, we find that the new algorithm plays a good performance in thinning quality, and at the same time, its thinning speed is 3 to 6 times faster than others, so it is a better thinning algorithm.

Keywords thinning, fingerprint, image processing, template matching

1 引言

细化又称为骨架化,即在不影响原图拓扑性的基础上,通过抽取对表达原图形状最关键的点,来使得原图中宽度大于1pixel的线条变成单像素的过程,也就是抽取原图骨架的过程。由于该技术在图像处理中占有重要地位,且是图像分析、信息压缩、特征抽取及模式识别中的常用技术,因此,研究它具有很大的实际价值。

一个好的细化算法应满足以下的要求^[1]:

- (1) 细化图像的连通性必须与原图像保持一致;
- (2) 细化图像中的线条宽度应为1pixel;
- (3) 细化图像中的线条应尽可能为中心线;

- (4) 细化后图像应尽可能保持原图的细节特征;
- (5) 细化算法的速度应尽可能快。

从以上5点要求中可以发现,前4条准则主要集中在对细化质量的评价,最后一条是对细化速度的要求。因此,一个细化算法的优劣,主要是从细化质量和细化速度两个方面进行评价。

文献[1]及文献[2]算法虽然都是较好的细化算法,可获得较好的细化质量,但由于引入了过多的迭代,致使细化速度较慢,文献[3]对文献[1]算法进行了改进,虽然有效地提高了细化速度,但细化质量有所下降,为此,本文在文献[1]算法的基础上,提出了一种新的图像快速细化算法。该算法不仅有效地保留了文献[1]算法的细化效果,同时较之文献[3]算法细化速度更快。

2 已有模板细化算法的比较分析

OPTA 算法^[2] (one-pass thinning algorithm) 是一种典型的基于模板的图像细化算法。本节将 OPTA 算法及其两个较为典型的改进算法^[1,3] 放在一起, 在大量实验的基础上, 从细化质量和细化速度上进行分析比较。大家知道, OPTA 算法的核心是通过应用消除和保留这两套模板(如图 1、图 2 所示)来达到对原图像细化的目的, 其算法过程大致如下:

(1) 对图像中的当前像素点, 将其与图 1 所示的消除模板进行匹配, 若满足消除模板中的某一个, 则转到步骤 (2); 否则转到步骤 (3);

(2) 将其与图 2 所示的保留模板进行匹配。若满足保留模板中的某一个, 则保留该点, 转到步骤 (1); 否则删除该点, 转到步骤 (1);

(3) 保留该点, 转到步骤 (1);

(4) 不断重复步骤 (1) (2) (3), 直至对一张图中所有点扫描完毕, 转入步骤 (5);

(5) 判断在上一次扫描过程中, 有无点被删除(即图像有没有被进一步细化), 若有, 则转入步骤 (1); 进入下一次扫描, 否则细化结束。

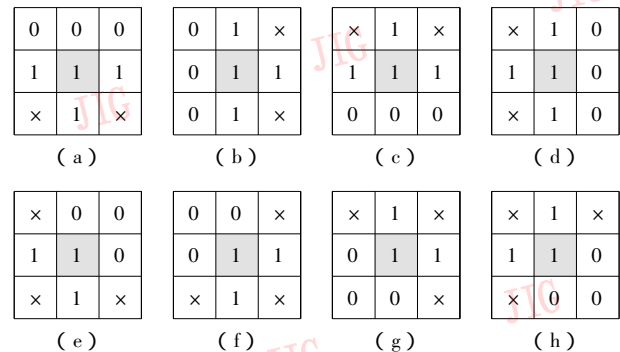


图 1 OPTA 消除模板

Fig. 1 OPTA elimination templates

(注: 图中 1 表示前景点, 0 表示背景点, x 表示可为 0 或 1, 其中阴影部分为当前像素点)

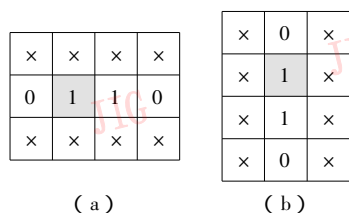


图 2 OPTA 保留模板

Fig. 2 OPTA reservation templates

对大量指纹图像细化的实验结果表明(见文献 [1]), 该算法虽能够保持原图的拓扑性, 但细化后的图像扭曲较大, 不仅毛刺较多, 不够光滑, 而且在三角点处有时不能完全细化。

针对以上的缺点, 冯星奎等提出了一种改进的算法^[1], 该方法对 OPTA 算法中的消除和保留模板都做了很大的改动, 如图 3、图 4 便是文献 [1] 算法中的消除和保留模板。

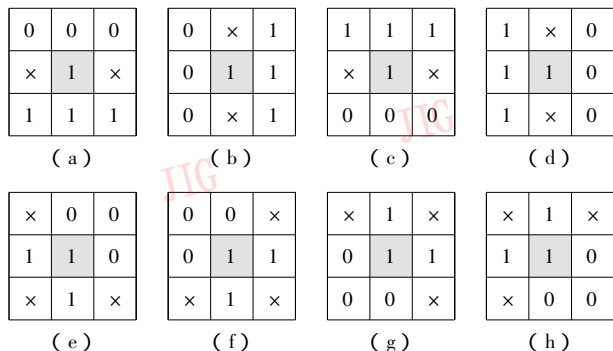


图 3 文献 [1] 算法中的消除模板

Fig. 3 elimination templates in reference [1]

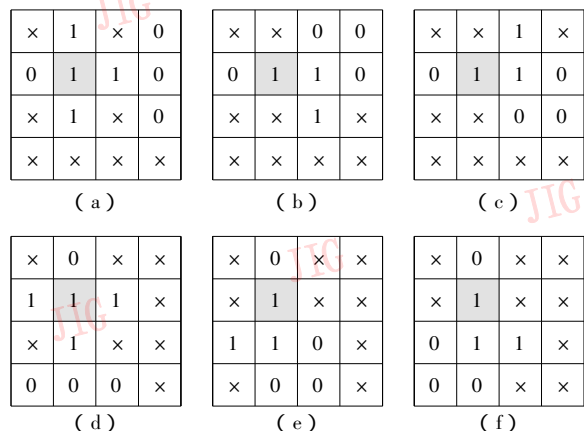


图 4 文献 [1] 算法中的保留模板

Fig. 4 reservation templates in reference [1]

实验结果表明, 该算法不但保持了原图的拓扑性, 而且在很大程度上解决了由 OPTA 算法带来的细化后图像扭曲大、毛刺多、不光滑的缺点, 确实获得了较好的细化结果。

本文在引言中指出, 对一个细化算法的评价, 应该从细化质量和细化速度上综合考虑。文献 [1] 算法在质量上虽已经取得了相对满意的结果, 但其细化速度较慢。

王家隆等认为, 文献 [1] 算法在处理双像素右

斜线(见文献[3])的过程中,由于保留模板的条件过于宽松,导致迭代次数大量增加,从而减慢了细化速度。本文暂且称其为右斜线效应。根据以上分析,在保留文献[1]算法消除模板的基础上,王家隆等重新提出了一套保留模板(如图5所示)。

本文对文献[3]方法做了很多实验,结果发现,该算法虽然明显地减少迭代次数,大幅度提高了细化速度,但破坏了原图的拓扑性。分析其原因,不难发现,文献[3]算法通过增强保留模板的限制条件,很好地解决了右斜线效应,但由于增强了保留条件,显然会使一些不该删除的点被删除。为解决这一缺陷,文献[3]增加了图5(g),图5(h),图5(i)保留模板,但由于这并没有解决问题,从而导致文献[3]算法在细化过程中产生了许多新的断点,破坏了原图的连通性。这对后面的处理,如去除毛刺、特征提取等都会产生较大影响。

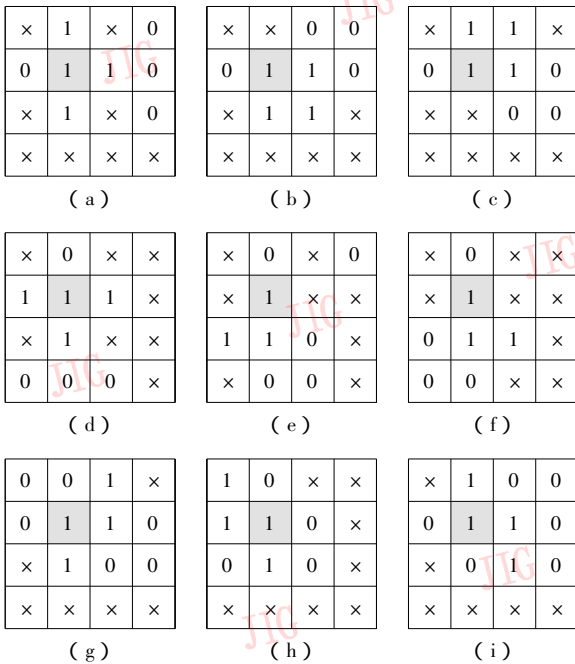


图5 文献[3]算法中的保留模板

Fig. 5 reservation templates in referenced [3]

为了更好地说明问题,本文对实验过程中遇到的一种情况进行了分析,以求简洁明了地阐述该算法存在的缺陷。如图6所示,图中黑点表示前景点,其余为背景点,图7为其局部放大后的效果图。

在这种情况下,当文献[3]算法扫描到图6中a点时,由于其符合图3(g)消除模板,且不符合图5中任何保留模板,故被删除。同样,由于图6中b点符合图3(e)消除模板,但不符合图5中任何保留模

板,因此亦被删除。图8为图6细化后的效果图,从图中可以看出,由于图6中a、b两点同时被删除,因而产生了新的断点。

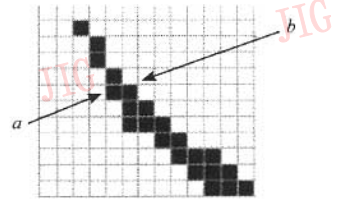


图6 待细化图

Fig. 6 The image to be thinned

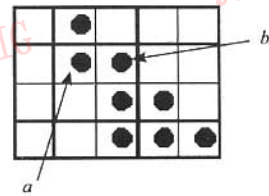


图7 局部放大后图

Fig. 7 The local magnified image

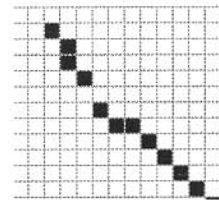


图8 细化后效果图

Fig. 8 The thinned image

从以上的分析中可以发现,文献[1]算法虽提高了细化的质量,但引入了多余的迭代,因而减慢了细化的速度;文献[3]算法虽有效地减少了迭代次数,却破坏了细化的质量。

3 一种优化后的细化算法

在经过深入的研究后,本文提出了一种新的细化算法,该算法不仅具有上述两种算法的优点,而且相较之文献[3]算法细化速度更快。新算法在文献[1]算法的基础上主要做了以下3个方面的工作:(1)在文献[1]算法的消除模板和保留模板的基础上提出一套组合模板;(2)优化组合模板;(3)基于优化后的组合模板提出新算法。下面是对这3个方面进行的详细阐述。

3.1 提出组合模板

在对文献 [1] 中的消除模板和保留模板(见图 3, 图 4)深入观察后发现:对于任意当前点

- (1)若满足图 3(a)消除模板,则只会满足图 4(b)保留模板或者图 4(d)保留模板;
- (2)若满足图 3(b)消除模板,则会只满足图 4(a)保留模板或者图 4(f)保留模板;
- (3)若满足图 3(c)消除模板,则会只满足图 4(c)保留模板;
- (4)若满足图 3(d)消除模板,则只会满足图 4(e)保留模板;
- (5)若满足图 3(e)消除模板,则只会满足图 4(e)保留模板或者图 4(f)保留模板;
- (6)若满足图 3(f)消除模板,则只会满足图 4(b)保留模板或者图 4(c)保留模板或者图 4(e)保留模板或者图 4(f)保留模板;
- (7)若满足图 3(g)消除模板,则只会满足图 4(b)保留模板或者图 4(c)保留模板;
- (8)若满足图 3(h)消除模板,则可以直接删除该点。

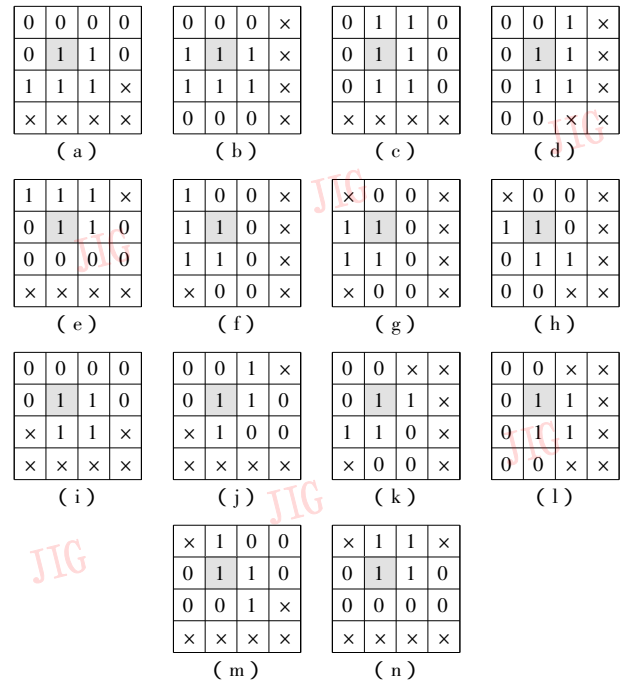


图 9 文献 [1] 算法中同时满足消除和保留模板的所有组合模板列举

Fig. 9 All combination-templates that both satisfy with elimination templates and reservation templates in reference [1]

(例如,图 9(a)表示既满足图 3(a)消除模板,又符合图 4(b)保留模板,依次类推,图 9(n)表示既满足图 3(g)消除模板,又符合图 4(c)保留模板)

从以上的过程可以看出,这样无疑增强了保留模板匹配的针对性。这很有意义,因为在该方法中,80%左右的时间都用来匹配模板,而模板的匹配又是一个耗费时间的处理过程。所以增强匹配的针对性,可以有效地减少匹配数量,提高匹配速度。以上(1)~(8)8种情况,可以用图形表示出来(如图 9 所示)图 9(a)~图 9(n)一起构成了文献 [1] 算法中既满足消除模板又符合保留模板的所有组合模板。本文将图 9 中的模板称为组合模板。

3.2 优化组合模板

在大量实验的基础上,本文发现将图 9(m)组合模板改成图 10(m)组合模板,可以消除右斜线效应。为了更好地说明问题,本文沿用了文献 [3] 中的图(如图 11 所示)。在这种情况下,图 11 中 d 点由于符合图 3(g)消除模板且不符合任何保留模板而被删除,图 11 中 c 点亦满足图 3(g)消除模板,但由于满足图 9(m)组合模板而需要被保留下来,这样就产生了右斜线效应。此时,若将图 9(m)组合模板改成图 10(m)组合模板,则 c 点便可以删除了,这样右斜线效应便迎刃而解。

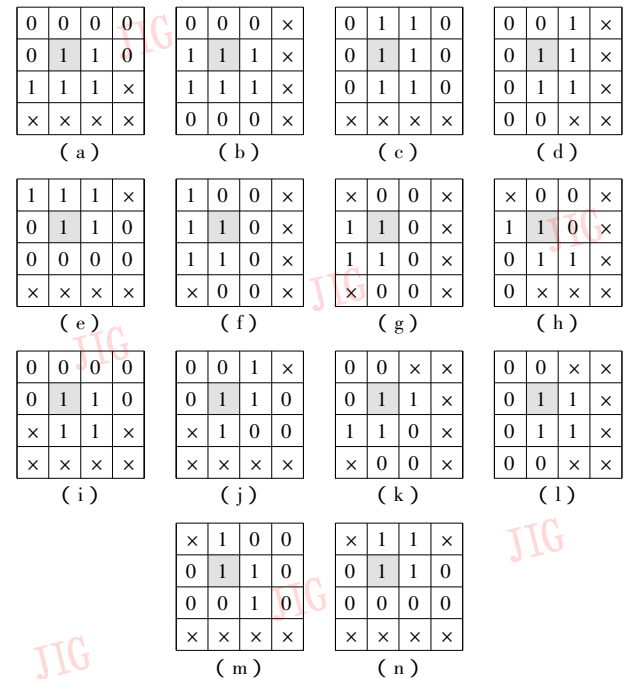


图 10 优化后的组合模板

Fig. 10 The optimized combination-templates

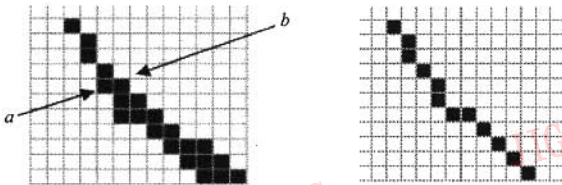
但这样做,由于弱化了保留条件,在某些情况下就会产生新的断点。不妨把图 6 再拿过来看一下,当扫描到 a 点时,由于其满足图 3(g)消除模板,但不满足图 10(m)组合模板和图 10(n)组合模板,故



图 11 文献 [3] 中的例图

Fig. 11 Examples from reference [3]

此点需要删除,扫描到点 b 时,因为其满足图 3(e) 消除模板,且不符合图 10(g),图 10(h) 组合模板,故 b 点也需删除,这样就产生了新断点。经研究发现若将图 9(h) 组合模板改成图 10(h) 组合模板,此时 b 点符合图 10(h) 组合模板,被保留下来,从而避免了新断点的出现(如图 12 所示)。大量实验表明,用图 10(h),图 10(m) 组合模板分别代替图 9(h),图 9(m) 组合模板,不但很好地解决了右斜线效应,而且完全符合原图的连通性,不会产生新断点。图 10 即为最终优化后的一整套组合模板。



(a) 待细化图 (b) 新方法细化后图像

图 12 图 6 及其用新方法细化后图像

Fig. 12 Fig. 6 and its thinned image with new algorithm

3.3 提出新算法

本文算法是在优化后组合模板的基础上提出来的。新算法保留了文献 [1] 算法中的消除模板(见图 3),但不引入保留模板的概念。为了便于叙述,本文规定当前点 P 的邻域如图 13 所示。改进后的算法与文献 [1] 算法实现过程的框架大体一致,但核心部分作了较大改动。下面简单给出该算法核心部分的实现过程:对于当前点 P

- (1) 若其满足图 3(a) 消除模板,则只需判断是否满足 $(q_4 = 0 \& q_5 = 0 \& q_6 = 1 \& q_7 = 0)$ 或 $(q_5 = 1 \& q_6 = 1 \& q_{12} = 0 \& q_{13} = 0 \& q_{14} = 0)$, 其中“&”表示逻辑运算“与”,若满足其中之一,则保留 P 点,否则删除该点;
- (2) 若其满足图 3(b) 消除模板,只需判断是否满足 $(q_2 = 1 \& q_9 = 1 \& q_4 = 0 \& q_7 = 0 \& q_{11} = 0)$ 或者 $(q_2 = 0 \& q_9 = 0 \& q_{12} = 0 \& q_{13} = 0)$, 若满足其中之一,

则保留 P 点,否则删除该点;

(3) 若其满足图 3(c) 消除模板,则只需判断是否满足 $(q_5 = 0 \& q_6 = 1 \& q_7 = 0 \& q_{11} = 0)$ 若满足,则保留 P 点,否则删除该点;

(4) 若其满足图 3(d) 消除模板,则只需判断是否满足 $(q_2 = 0 \& q_9 = 1 \& q_{13} = 0 \& q_{14} = 0)$ 若满足,则保留 P 点,否则删除该点;

(5) 若其满足图 3(e) 消除模板,则只需判断是否满足 $(q_8 = 1 \& q_{10} = 0 \& q_{13} = 0 \& q_{14} = 0)$ 或 $(q_8 = 0 \& q_{10} = 1 \& q_{12} = 0)$, 若满足其中之一,则保留 P 点,否则删除该点;

q_1	q_2	q_3	q_4
q_5	P	q_6	q_7
q_8	q_9	q_{10}	q_{11}
q_{12}	q_{13}	q_{14}	q_{15}

图 13 P 点邻域图

Fig. 13 The adjacent points of P

(6) 若其满足图 3(f) 消除模板,则只需判断是否满足 $(q_3 = 0 \& q_4 = 0 \& q_7 = 0 \& q_{10} = 1)$ 或 $(q_3 = 1 \& q_7 = 0 \& q_{10} = 0 \& q_{11} = 0)$ 或 $(q_8 = 1 \& q_{10} = 0 \& q_{13} = 0 \& q_{14} = 0)$ 或 $(q_8 = 0 \& q_{10} = 1 \& q_{12} = 0 \& q_{13} = 0)$, 若满足其中之一,则保留 P 点,否则删除该点;

(7) 若其满足图 3(g) 消除模板,则只需判断是否满足 $(q_3 = 0 \& q_4 = 0 \& q_7 = 0 \& q_{10} = 1 \& q_{11} = 0)$ 或 $(q_3 = 1 \& q_7 = 0 \& q_{10} = 0 \& q_{11} = 0)$, 若满足其中之一,则保留 P 点,否则删除该点;

(8) 若其满足图 3(h) 消除模板,则直接删除该点。

4 实验结果比较与分析

为验证本文算法的效果,本文对文献 [1], 文献 [3] 及改进新算法使用了统一的编程思路,并采用 FVC2000 中经过二值化后的指纹图像进行了大量的对比实验,指纹图像大小为 300×300 pixels。

为了对比上述 3 种算法的图像细化效果,本文进行了大量实验,在实验过程中,对每一张图都采用以上 3 种算法进行细化,同时,为了更好地验证细化后是否产生了新断点,本文又对每张细化图进行了相同的去毛刺处理,若细化图像中存在大量断点,则去毛刺后,这些断点就会被扩大化,从而出现很明显的断裂。所有实验结果都反映了改进算法的细化结果不仅解决了文献 [3] 算法中产生新断点问题,并

达到了与文献 [1] 算法相同的细化效果。图 14 为本文从大量实验中随机挑选的一组细化结果图。



(a) 待细化图像



(b) 文献 [1] 算法细化后图像



(c) 图 14(b) 去毛刺后图像



(d) 文献 [3] 算法细化后图像



(e) 图 14(d) 去毛刺后图像



(f) 改进算法细化后图像



(g) 图 14(f) 去毛刺后图像

图 14 细化效果对比图

Fig. 14 The thin-effects with different algorithms

为了对比 3 种算法的细化速度, 本文随机地抽取了其中 10 张图像的细化时间结果。实验使用的

计算机硬环境为 Pentium(R) 4 CPU 2.40GHz, 256M 内存, 软环境为 Microsoft Windows XP Professional 操作系统, Microsoft Visual C++ 6.0 编程语言。实验结果如表 1 所示。从表中可以看出, 改进后的算法在迭代次数上和文献 [3] 算法几乎一致。同时, 由于改进算法中抛弃了保留模板, 只是在每个消除模板上稍加了一些判断条件, 因此, 改进算法较文献 [3] 算法更快。

表 1 实验结果

Tab. 1 Experiment results

图像编号	文献 [1] 算法		文献 [3] 算法		改进后算法		改进后算法编码实现	
	时间 (ms)	迭代次数	时间 (ms)	迭代次数	时间 (ms)	迭代次数	时间 (ms)	迭代次数
1	140	24	93	11	62	11	15	11
2	125	18	93	10	62	10	16	10
3	93	13	93	9	62	10	16	10
4	109	15	78	7	46	8	15	8
5	94	16	62	8	47	8	15	8
6	78	12	62	8	47	8	15	8
7	125	21	93	9	46	9	15	9
8	62	7	78	7	46	8	15	8
9	93	14	78	7	47	7	16	7
10	109	17	78	8	47	9	16	9

另外, 在实验过程中, 本文亦用编码查表法对改进后算法进行了实验, 从中可以看出, 该方法大大加快了细化速度, 同时也说明了模板的匹配是一个很耗时的过程。

5 结 论

本文在对文献 [1], 文献 [3] 算法进行深入研究的基础上, 提出了一种改进的新的细化算法。大量实验证明, 该算法不但能获得较好的细化效果, 而且在很大程度上简化了运算, 从而提高了细化速度, 是一种比较理想的图像细化方法。

参考文献 (References)

- 1 Feng Xing-kui, Li Lin-yan, Yan Zu-quan. A New Thinning Algorithm for Fingerprint Image [J]. Journal of Image and Graphics, 1999, 4(10): 835 ~ 838. [冯星奎, 李林艳, 颜祖泉. 一种新的指纹图像细化算法 [J]. 中国图象图形学报, 1999, 4(10): 835 ~ 838.]
- 2 Chin R T, Wan H K, Stover D L, et al. A one-pass thinning algorithm and its parallel implementation [J]. Computer Vision Graphics Image Processing, 1987, 40(1): 30 ~ 40.
- 3 Wang Jia-long, Guo Chen-an. An Improved Image Template Thinning Algorithm [J]. Journal of Image and Graphics, 2004, 3(9): 297 ~ 301. [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, 2004, 3(9): 297 ~ 301.]