

# 一种改进的指纹快速细化算法\*

## An Improved Quick Thinning Algorithm for Fingerprint Image

张显全, 唐莹, 郭明明

ZHANG Xian-quan, TANG Ying, GUO Ming-ming

(广西师范大学计算机科学系, 广西桂林 541004)

(Department of Computer Science, Guangxi Normal University, Guilin, Guangxi, 541004, China)

**摘要:**针对指纹快速细化算法中存在双像素宽的情况,研究快速细化后像素的8邻域像素值的特征,通过计算8邻域像素值的和以及8邻域中相邻像素间的关系,对快速细化算法进行改进,使细化的结果为单像素宽。改进的算法对指纹的细化效果好,速度快,效率高。

**关键词:**快速细化 指纹 图像 单像素 8邻域

中图分类号: TP311.5 文献标识码: A 文章编号: 1002-7378(2006)04-0237-03

**Abstract:** The characteristics of eight-neighbor pixels were studied to solve the problem of double pixels width in the quick thinning algorithm. By computing the sum of eight-neighbor pixel values and researching the relationship between the adjacent eight-neighbor pixels, an improved quick thinning algorithm is proposed. The result in the improved algorithm becomes one pixel width. The experimental results show that the improved algorithm runs quick and efficient.

**Key words:** quick thinning, fingerprint, image, one pixel wide, eight-neighbor

细化处理是指在指纹图像二值化以后,在不影响纹线连通性的基础上,删除纹线的边缘像素,直到纹线为单像素宽为止。细化后理想的结果是纹线骨架应该为原始纹线的骨架中心,并保持纹线细节特征以及纹线的收敛性、拓扑性、细化性和快速性。

目前有很多种对二值图像的细化方法,比较常用的算法一般都是通过考察像素点的四邻域点或八邻域点像素来制订判断规则,从而删除边界点达到细化的目的。文献[1]的方法是基于四邻域的细化算法,对汉字和字母的细化效果显著;文献[2]的方法是基于像素的四邻域点、八邻域点以及16环域来判断是否删除该像素点,对一般的文字和图形都有较好的细化效果。但这些算法应用到指纹图像容易出现较多的断点,影响了纹线的连接性。对于指纹细化比较经典且应用较普遍的是快速细化算法和OPTA算法,采用快速细化算法细化图像,虽可满足收敛

性、拓扑性,并且细化速度非常快,但由于该算法对指纹图像细化后有很多都是双像素宽,对后续处理影响较大,如在指纹的特征提取中,会造成许多虚假特征。文献[3]对快速细化算法进行了改进,虽在保持指纹纹线的细节性方面取得较好的效果,但仍有部分断点的情况出现;文献[4,5]所提出的算法都是在经典图像模板算法的基础上将检测模板加以改进,以适应指纹图像的特性,达到完全细化的目的,但是对于细化纹线较宽的指纹图像容易出现毛刺,且细化过程中由于要检测模板的数目较多,细化的速度较慢。

本文针对快速细化算法存在细化不彻底的情况,对不是单像素宽的特点进行了研究,提出一种改进的指纹快速细化算法,取得了较好的效果。

### 1 指纹图像的细化

指纹图像进行二值化处理后,若像素点 $P$ 为目标像素则 $P = 1$ ,若 $P$ 为背景像素则 $P = 0$ ;对象素点 $P$ 的8邻域像素点为 $P_0, P_1, P_2, P_3, P_4, P_5, P_6$ 和 $P_7$ ,其中 $P_1, P_3, P_5$ 和 $P_7$ 为其4邻域点,如图1所示;

收稿日期: 2006-07-03

作者简介: 张显全(1964-),男,重庆秀山人,副教授,硕士生导师,主要从事图形图像处理研究。

\* 广西自然科学基金(No. 0447035)资助。

像素  $P$  的 4 邻域点中至少有一个为背景点时该点为边界点。

$P_0$	$P_1$	$P_2$
$P_7$	$P$	$P_3$
$P_6$	$P_5$	$P_4$

图 1 像素的 8 邻域

对于指纹图像,理想的细化结果为纹线端点的 8 邻域中只有 1 个目标像素点,纹线上连续像素点的 8 邻域有且仅有 2 个像素点,分叉点 8 邻域最多只有 3 个目标像素点。设像素  $P$  的 8 邻域像素值的和为:  $n_{sum} = \sum_{i=0}^7 P_i$ , 8 邻域中相邻点像素值差的绝对值的和为  $t_{sum} = \sum_{i=0}^7 |P_{i+1} - P_i|$ ; 快速细化算法是先判断出指纹纹线的边界点,然后再逐步删除这些边界点,直到没有可删除的点为止,具体算法是:如果  $P$  点满足  $t_{sum} = 2, n_{sum} \neq 1$  且  $n_{sum} < 6$  则可将其删除。快速细化算法的运行速度快,但存在缺陷,因为由该算法细化的图像很多不是单像素宽(存在大量的双像素宽的细化结果),即细化不彻底。

根据快速细化算法中删除的条件,如图 2 所示( $X$  代表可为 0 或者 1,黑色像素值为 1)的像素点  $P$

不满足  $\begin{cases} t_{sum} = 2 \\ n_{sum} \neq 1 \\ n_{sum} < 6 \end{cases}$  被删除的条件,因而会保留下来,

从而产生双像素宽的情况,对于像素点  $P$  的 8 邻域像素有  $t_{sum} = 4$ , 且  $\begin{cases} P_1 = 1 \\ P_3 = 1 \end{cases}, \begin{cases} P_3 = 1 \\ P_5 = 1 \end{cases}, \begin{cases} P_5 = 1 \\ P_7 = 1 \end{cases}$  和

$\begin{cases} P_7 = 1 \\ P_1 = 1 \end{cases}$  四个条件至少有一个成立。针对这些情况,

我们通过改进快速细化算法,在细化过程中,找出双像素宽的像素对其进行删除,使细化结果成为单像素宽,同时保持像素间的拓扑结构。

$X$	$P_1$	0	0	0	$X$
0	$P$	$P_3$	0	$P$	$P_5$
0	0	$X$	$X$	$P_3$	0
$X$	0	0	0	$P_1$	$X$
$P_7$	$P$	0	$P_7$	$P$	0
0	$P_5$	$X$	$X$	0	0

图 2 4 种双像素的情况

## 2 改进的快速细化算法

针对快速细化算法细化结果存在大量双像素宽的情况,对快速细化算法进行改进。在快速细化后对图像进行处理,若某个像素  $P$  的  $t_{sum} = 4$ , 则计算  $P_{2i-1} + P_{2i+1}$  ( $i = 1, 2, 3, 4$ , 当  $i = 4$  时  $P_{2i+1} = P_1$ ) 的值,只要其中的一个值为 2, 则该像素产生双像素宽,删除可得单像素宽的细化图像,改进算法如下:

步骤 1: 遍历整个指纹图像,找出纹线的边界点。

步骤 2: 如果边界点  $P$  满足  $\begin{cases} t_{sum} = 2 \\ n_{sum} \neq 1 \\ n_{sum} < 6 \end{cases}$ , 则将其删除;

步骤 3: 如果有未处理的边界点对该点执行步骤 2, 否则执行步骤 4;

步骤 4: 若有可删除的边界点重复执行步骤 1 至步骤 3, 否则转到步骤 5;

步骤 5: 重新遍历图像,对目标像素  $P$ , 当  $t_{sum} = 4$  时,且 ( $i = 1, 2, 3, 4$ , 当  $i = 4$  时  $P_{2i+1} = P_1$ ) 的值中任意一个为 2, 则将其删除;

步骤 6: 对下一像素点继续执行步骤 5, 直到遍历完整个图像。

由改进后算法的步骤可知,新算法在快速细化的基础上对图像进行进一步细化,在保证细化结果只有单像素宽的同时,算法的速度较快。

## 3 实验

为了验证改进后算法的效果,作者选取了大量的指纹样本进行了测试,都取得了比较好的效果。图 3 是其中一幅指纹图像用几种细化方法进行细化的结果,图 3a 是指纹的原始图像;图 3b 是快速细化算法的细化结果,光滑无毛刺,但不是单像素宽;图 3c 是采用文献[5]的算法的细化结果,虽然是单像素宽,但不光滑,有较多毛刺;图 3d 是本文算法得到的细化结果,算法同时解决了这两个问题,既光滑无毛刺,且纹线都是单像素宽。

对 30 幅不同的指纹图像,在 P4 2.0G 256M 的微机上进行细化处理,从表 1 中可知,本文算法处理的时间少于文献[5]算法处理所用的时间,比快速细化算法处理的时间略有增加,但时间增加的幅度很小。



图3 细化算法对比实验结果

a. 原始图像; b. 快速细化算法结果; c. 模板细化算法结果; d. 本文算法细化结果

表1 3种算法的运行时间

算法	运行时间(ms)
快速细化算法	5.782
文献[5]算法	9.148
本文算法	6.506

## 4 结束语

本文在快速细化算法的基础上提出了一种新的快速细化算法。从实验结果可见,本文的细化算法不仅速度快,同时还保证了对图像的完全细化,细化效果较快速细化算法和改进的图像模板算法有一定提高。细化后的图像能够很好的保持原图像的特征,并且没有出现双像素宽的情况,是一种有效的指纹快速细化方法,有较好的应用价值。

参考文献:

- [1] 张昊,徐刚.基于四邻域的二值图像细化算法[J].信息技术与信息化,2004,6(3):24-27.
- [2] 吕俊白.一种有效的二值图像细化算法[J].计算机工程,2003,29(18):147-148.
- [3] 穆国燕,田俊霞,陈树中.一种有效的指纹图像分割和细化方法[J].计算机工程,2002,28(10):82-83.
- [4] 冯星奎,李林艳,颜组泉.一种新的指纹图像细化算法[J].中国图像图形学报,1999,4A(10):835-838.
- [5] 王家隆,郭成安.一种改进的图像模板细化算法[J].中国图像图形学报,2004,9(3):297-301.

(责任编辑:韦廷宗)

(上接第233页)

于局部优点的常见弊病,在科学与工程优化问题中具有良好的应用前景。

作为新型的进化算法,PSO算法与蚁群算法的研究时间较短,理论基础有待进一步研究。尤其是对于求解连续空间的优化问题,算法的收敛性和参数选定都需要探讨,这些都值得进一步研究。

参考文献:

- [1] 王凌.智能优化算法及其应用[M].北京:清华大学出版社,2001:154-157.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization; proceeding of the IEEE International Conference on Neural networks, IV[C]. Piscataway: IEEE Service Center, 1995:1942-1948.
- [3] SHI Y, EBERHART R C. A modified particle swarm optimizer; proceedings of the IEEE Congress on Evolutionary Computation, Piscataway [C]. USA: IEEE Service Center, 1998:6973.
- [4] COLORMI A, DORIGO M, MANIEZZO V. Distributed

optimization by ant colonies; proc Europ Conf Artif Life[C]. Paris: Elsevier Publishing, 1991:134-142.

- [5] ROCCO P. Stability of P D control for industrial robot arms [J]. IEEE Trans Robot Automat, 1996, 12(4): 606-614.
- [6] MANIEZZO V, COLORMI A. Ant system applied to the quadratic assignment problem [J]. IEEE Trans Knowl Data Eng, 1999, 11(5): 769-778.
- [7] 吴庆洪,张纪会,徐心和.具有变异特征的蚁群算法[J].计算机研究与发展,1999,36(10):1240-1245.
- [8] 杨勇,宋晓峰,王建飞,等.蚁群算法求解连续空间优化问题[J].控制与决策,2003,18(5):62-65.
- [9] 陈焯.用于连续函数优化的蚁群算法[J].四川大学学报,2004,36(6):119-122.
- [10] 熊伟清,余舜浩,魏平.用于求解函数优化的一个蚁群算法设计[J].微电子学与计算机,2003(1):23-25, 30.

(责任编辑:邓大玉)