

## 轮廓跟踪算法的改进及在字符识别技术中的应用

任民宏

(陕西理工学院 计算机科学与技术系, 陕西 汉中 723003)

(rmh9605@126.com)

**摘 要:**针对图像中字符识别技术的应用需求,提出了轮廓跟踪算法的改进算法,在链码表示法基础上提出了轮廓表存储结构。在字符识别中根据轮廓表生成字符的轮廓特征数据序列,运用其相关值来度量字符的相似程度。实验结果表明提取特征少,识别速度快,准确率高,受字号的影响较小。

**关键词:**轮廓跟踪算法;轮廓表;字符识别

**中图分类号:** TP391.43 **文献标识码:** A

## Improvement of contour following algorithm and its application in character recognition

REN Min-hong

(Department of Computer Science & Technology, Shaanxi University of Technology, Hanzhong Shaanxi 723003, China)

**Abstract:** To fulfill the application requirement of character recognition in image, an improved algorithm for contour following algorithm was proposed and a contour list based on chain code was presented. In character recognition, the character lineament sequence was generated according to contour list and the character similarity was measured by its correlation value. The experimental results show that there are fewer extracted character features, faster recognition and higher degree of accuracy, and it is less affected by font size.

**Key words:** contour following algorithm; contour list; character recognition

在二值图像处理中,轮廓跟踪算法是一种常用的图像处理算法,主要用于连接成分的标记和提取图像的区域形状特征中,在机器人、图像识别、计算机视觉等方面有着重要的应用。字符识别技术属于图像识别的范畴,广泛应用于光符阅读、车牌识别等领域。目前字符识别技术较多,例如人工神经网络识别法、基于特征编码识别法等。人工神经网络识别法容易受到网络的权重初始化不合理而影响识别的速度和准确性。基于特征编码识别法提取的汉字特征较多,影响识别效率。本文针对轮廓跟踪算法,提出了改进的轮廓跟踪算法和轮廓表存储结构,并应用于字符的识别技术中。

### 1 轮廓跟踪算法的改进

在轮廓跟踪之前,必须对图像进行预处理,预处理的主要目的是去除噪声、压缩冗余信息,为轮廓跟踪做好准备,它包括二值化、去噪声、字分割、归一化。

对于一幅二值图像的一条轮廓线,轮廓跟踪的过程是:按照从上到下、从左到右的顺序在图像中搜索,找到的第一个值为 1 的像素作为轮廓线的起点  $S$ ,然后从  $S$  点出发按照一定的“跟踪准则”找到其相邻的轮廓点,直到找到所有的轮廓点。

简单的“跟踪准则”描述为:如果当前轮廓点为  $N$ ,则在  $N$  的 8-邻域中寻找目标点  $M$ ,如果  $M$  的 4-邻域点不都是目标点,则  $M$  是轮廓点,否则不是轮廓点;如果  $M$  和  $S$  重合,则轮廓跟踪结束,否则继续寻找下一点。这种算法要对每个轮廓点 8-邻域内的所有点的 4-邻域点进行判别,计算量比较大,影响跟踪的速度,为此本文提出一种改进算法。

1) 按照从上到下、从左到右的顺序在字符图像中搜索,

找到的第一个值为 1 的像素作为字符轮廓线的起点,记该像素为  $P_1$ ;

2) 考虑一个以  $P_1$  为中心的  $3 \times 3$  模板。将模板内的各像素按图 1 所示标记序号为 0 ~ 7,从序号为 0 的像素开始按顺序检查各像素是否是值为 1 的像素,把最初遇到的值为 1 且不为已跟踪过的轮廓点的像素设为  $P_2$ 。如果从 0 到 7 全是值为 0 的像素,则  $P_1$  为孤立点,转到 4);

$x_0$	$x_7$	$x_6$
$x_1$	$P$	$x_5$
$x_2$	$x_3$	$x_4$

3) 将  $P_2$  作为模板中心像素,按同样的方法搜索  $P_3, P_4, \dots$ 。如果搜索的结果

图 1 以  $P$  为中心的  $3 \times 3$  模板  $P_n = P_1, P_{n+1} = P_2$ , 则表明  $P_1, P_2, \dots, P_n$  已经形成一个闭环,中止本条轮廓线的跟踪。点列  $P_1, P_2, \dots, P_n$  就是我们要找的一条轮廓线。

4) 从  $P_1$  出发按照从上到下、从左到右的顺序在图像中搜索,找到的第一个值为 1 且不在已经找到的轮廓线上的轮廓点作为另一条轮廓线的起点,转到 2),直到找不到新的轮廓线的起点为止。

这个改进算法不仅简化了判断轮廓点的计算,而且可以跟踪一幅图像的所有轮廓线。

### 2 链码表示法的改进

二值图像的一条轮廓线可以用直角坐标表示法和链码表示法表示。在轮廓跟踪算法中通常采用链码表示法表示。例如,图 2 所示的二值图像轮廓线,用链码表示法可写成 A55432245566。

很明显,这种链码表示法必须存储轮廓线上每一个轮廓

点相对于上一个轮廓点的方向。直接用链码表示法来存储字符的轮廓线,用于字符识别中是不行的,因为要识别的字符与特征字符库中的标准字符的笔画长短不一定一致,因而对链码表示法加以改进,使其用于字符识别技术中。

要使链码表示法能用于字符识别技术中,必须压缩轮廓线上轮廓点信息,可以采用如下定义的轮廓表来存储。轮廓表的表头节点应由起点域和指针域构成,如图 3(a)所示,起点域中存储一条轮廓线的起点坐标,指针域指向第一个轮廓拐点节点。轮廓

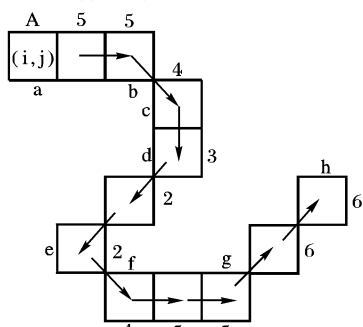


图2 链码的例子

拐节点节点应由方向域、点数域和指针域构成,如图 3(b)所示,方向域中存储该轮廓线上当前轮廓拐点相对于上一个轮廓拐点的方向值,点数域中存储该轮廓线上上一个轮廓拐点与当前轮廓拐点之间的轮廓点个数,所谓拐点就是轮廓线上下一个相邻轮廓点方向相对发生变化的轮廓点,指针域指向下一个相邻轮廓拐节点。一幅图像的轮廓线若有多条,则将所有轮廓表的表头节点用一个一维数组组织。

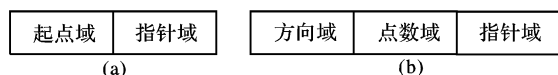


图3 轮廓表的表头节点和轮廓拐点节点

例如,图2中的轮廓线包括起点和终点在内,一共有8个拐点,分别是a、b、c、d、e、f、g、h。其轮廓表如图4所示。

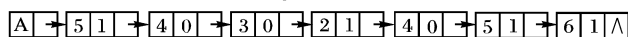


图4 图2中的轮廓线的轮廓表

### 3 字符识别

对于经过预处理的字符图像,必须先用上述改进的轮廓跟踪算法建立轮廓表。为了提高字符的识别速度和准确率,字符的轮廓特征数据序列只取字符轮廓表中各个轮廓拐节点的方向值,点数值只在判断轮廓线起点时有用,对字符识别

无用。例如图5中的“只”字,用上述改进的轮廓跟踪算法,可得到四条轮廓线。第一条轮廓线是a→b→c→d→a,a是起点,b、c、d、a是拐点。第二条轮廓线是e→f→g→h→e,e是起点,f、g、h、e是拐点。第三条轮廓线是i→j→k→l→i,i是起点,j、k、l、i是拐点。第四条轮廓线是m→n→o→p→m,m是起点,n、o、p、m是拐点。建立的轮廓表如图5(b)所示。

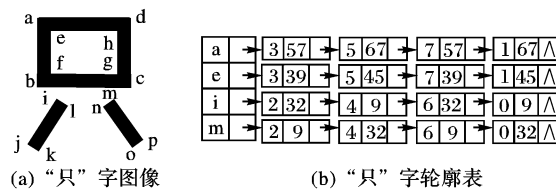


图5 “只”字图像及其轮廓表

根据图5(b)中“只”字的轮廓表,“只”字的轮廓特征数据序列是3,5,7,1,3,5,7,1,2,4,6,0,2,4,6,0。

设字符的轮廓特征数据序列为 $f_1, f_2, \dots, f_n$ ,标准字符特征数据序列为 $f'_1, f'_2, \dots, f'_n$ ,则可以按下面公式计算两个序列的相关值。

$$E = \frac{1}{N} \sum_{i=1}^N f_i, \sigma = \frac{1}{N} \sum_{i=1}^N (f_i - E)^2 \quad (1)$$

$$E' = \frac{1}{N} \sum_{j=1}^N f'_j, \sigma' = \frac{1}{N} \sum_{j=1}^N (f'_j - E')^2 \quad (2)$$

$$\rho = \frac{\frac{1}{N} \sum_{i=1}^N (f_i - E)(f'_i - E')}{\sqrt{\sigma \sigma'}} \quad (3)$$

理想情况下,两个特征数据序列如果相似,相关值 $\rho$ 的绝对值应等于1,因而在识别时选取相关值 $\rho$ 的绝对值最大时对应的标准字符作为识别的结果。

### 4 实验结果及分析

采用上面提出的方法,对52个英文字母的大小写和10个阿拉伯数字、不同结构的汉字共150个字符进行了识别。由于篇幅所限,仅列出各类字符中的一个字符在轮廓跟踪算法改进前和改进后所用时间、轮廓特征数据序列、识别结果,如表1所示。

表1 各类字符的实验数据

识别对象	算法改进前 所用时间(s)	算法改进后 所用时间(s)	轮廓特征数据序列	识别结果	备注
F	0.230	0.032	3531357175357131754601	F	大写英文字母
f	0.208	0.029	12121231353135717571765653570101	f	小写英文字母
4	0.224	0.025	25313571757171	4	阿拉伯数字
呈	0.582	0.078	3575357602103571457531457531457502175021	呈	上下结构
草	0.635	0.082	31457531036753145753675012175367601216750121760317603257135713571	草	上中下结构
计	0.376	0.054	34570314575367501217604575324602760121	计	左右结构
班	0.630	0.086	21457531457532460275323567132012175013246745753145753145750121750121750121760	班	左中右结构
这	0.514	0.059	3435703435702145752024207601214575324646206435706501	这	半包围结构
因	0.619	0.089	367536760210357131357532356457075021760	因	全包围结构
十	0.254	0.037	3145367501217601	十	独体字

实验中算法在VC++6.0下实现,所用机器配置P4 2.0G CPU,256M RAM。上述的实验数据均以在photoshop 6.0下做的字符图像作为识别对象,图像中字符的字体均为宋体,字号为72pt。为了节约篇幅,表1中的识别对象均用相应的字符代替。

实验结果表明采用本文中改进的轮廓跟踪算法可以有效

地降低轮廓跟踪所用的时间。采用轮廓表存储字符的轮廓要比直接用链码表示法存储的轮廓点少,起到了压缩轮廓数据的目的。采用轮廓表可以方便地得到字符的轮廓特征数据序列。算法的正确识别率在96%以上,字符识别时间为0.5s以内。

(下转第2382页)

$$\begin{cases} \text{secondp. } x + \Delta x = \text{newp. } x \\ \text{secondp. } y + \Delta y = \text{newp. } y \end{cases} \quad (6)$$

式中,  $\text{newp. } x$  表示后续图像在按照位移量恢复之后的点的坐标。然后取出基准图像的第  $n+1$  个角点, 它的坐标是  $\text{firstp. } x_{n+1}, \text{firstp. } y_{n+1}$ , 在后续图像中按照排序的顺序, 依次检测, 找到后续图像中与第  $n+1$  个角点相匹配的点, 判断的标准入(7)式,  $t$  表示阈值:

$$\begin{cases} \text{firstp. } x_{n+1} - \text{newp. } x \leq t \\ \text{firstp. } y_{n+1} - \text{newp. } y \leq t \end{cases} \quad (7)$$

4) 从基准图像中取出第  $n+2$  个角点的坐标  $\text{firstp. } x_{n+2}, \text{firstp. } y_{n+2}$ , 按照上面的步骤3) 的做法在后续图像中找出与第  $n+2$  个角点匹配的。接下来取出第  $n+3 \cdots \text{end}$  一直到基准图像的最后一个角点, 完成上面的步骤3)。

5) 判断如果基准图像和后续图像的第  $n$  个角点之间的位移量  $\Delta x, \Delta y$  恢复到和后续图像上之后, 基准图像第  $n$  个角点之后的所有角点( $n+1, n+2, n+3 \cdots \text{end}$ ) 之中, 有半数以上的角点都在后续图像上找到对应的点, 那么第  $n$  个角点算匹配成功。

如果匹配成功, 记录下点对的位置。如果匹配不成功, 删除这个点。开始重复执行操作第二步, 比基准图像的第  $n+2$  角点是否有点能够匹配后续图像的第  $n+2$  个角点。直到完成基准图像之中所有点的匹配。

## 2 实验结果及其分析

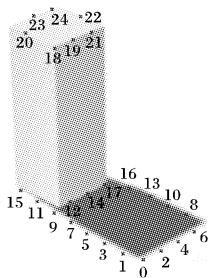


图3 原始图片1的角点效果

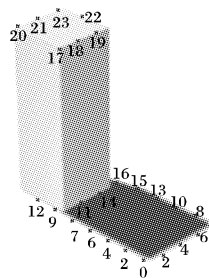


图4 原始图片2的角点效果

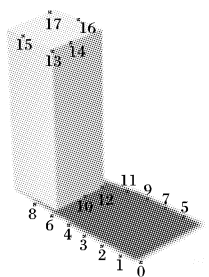


图5 图片1配准后的效果

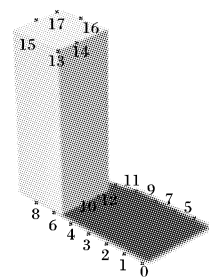


图6 图片2配准后的效果

根据本文的方法, 在 Windows 环境下, 使用 VC++ 6.0 编写程序来实现。图3、图4是两幅不同视角的同一物体的

照片, 首先采用了本文的方法对图像的角点进行检测, 图3找到的角点有24个, 图4找到的角点有23个, 都用的“X”表示。角点找到以后, 按照本文的算法进行图像匹配, 从匹配结果图5、图6可以看出, 没有匹配成功的点就消失了, 证明算法的正确性。图像的位移是  $\begin{cases} \Delta x = -57 \\ \Delta y = -29 \end{cases}$ 。

## 3 结语

本文应用 SUSAN 算法, 对刚体图片进行角点检测, 然后根据刚体的几何不变性对图像直接进行配准。由于采用 SUSAN 算法找出角点, 角点的数量不大, 对于后续的图像配准的实现带来相当大的速度优势。在图像配准中, 仅仅对找到的少量角点进行比较, 采用两重循环的简单排序办法, 其算法的时间复杂度为  $O(n^2)$ , 由于角点数量不大, 对于时间复杂度为  $O(n^2)$  还是很快的。由于 SUSAN 算法和其他角点检测算法相比较具有精确度高, 抗噪声能力强和速度快的特点, 因此适合于很多图像的角点检测。当然本文的算法也有待改进, 由于以快速检测为目标而忽略了检测角点的精确度, 配准算法适用的范围比较窄, 这些都是以后有待改进和提高的地方。

### 参考文献:

- [1] LI H, et al. A contour-based approach to multisensor image registration[J]. IEEE Transaction on Image Processing, March, 1995, 4(3): 320-334.
- [2] JURGEN B. Automated Inspection Systems for Industry [M]. UK: IFS.
- [3] ROSENFELDA K. K Gray\_level corner detection[J]. Pattern Recognition letters, 1982, 3(1): 95-102.
- [4] ZUNIGA OA, HARALICK R. Corner detection using the facet model[A]. In Proceedings of CVPR'83[C]. Arlington, VA, USA, June 1983. 30-37.
- [5] SMITH SM, BRADY JM. SUSAN-a new approach to low level image processing[J]. Journal of Computer Vision, 1997, 23(1): 45-78.
- [6] MEDIONI G, YASUMOTO Y. corner detection and curve representation using cubic B-Spline[J]. Computer Vision, Graphics, Image Process., 1987, 39(3): 267-278.
- [7] HARRIS C, STEVEN M. A combined corner and edge detector[A]. In: Proc 4th Alvey Vision Conf[C]. Manchester, UK, 1988. 147-151.
- [8] Smith SM, BRADY M. SUSAN-a new approach to low. level image processing[J]. International Journal of Computer. 1997, 23(1): 45-78.
- [9] 周鹏, 谭勇, 徐守时. 基于角点检测图像配准的一种新算法[J]. 中国科学技术大学学报, 2002, 32(4): 455-461.
- [10] TRAJKOVIC M, HEDLEY M. Fast corner detection Image and Computing[J]. 1998, 16: 75-87.

(上接第 2379 页)

## 5 结语

本文针对轮廓跟踪算法, 提出了改进的轮廓跟踪算法和轮廓表存储结构, 并应用于字符的识别技术中。在存储字符的轮廓线时运用了轮廓表, 从而生成字符的轮廓编码, 不仅能压缩轮廓特征数据以提高字符匹配速度, 而且可以避免字符笔画长短与标准字符笔画长短不一致所引起的误差。

### 参考文献:

- [1] 杜彦蕊, 李珍, 宋卫宏. 基于特征编码的手写字符识别技术[J]. 计算机工程, 2004, 30(5): 156-158.
- [2] 赵丹青, 孙德宝. 字符识别在车牌识别中的应用[J]. 中南民族大学学报(自然科学版), 2002, 21(3): 33-35.
- [3] 张远鹏, 董海, 周文灵. 计算机图像处理技术基础[M]. 北京: 北京大学出版社, 1999.
- [4] 朱学芳. 多媒体信息处理与检索技术[M]. 北京: 电子工业出版社, 2002.