

一种基于屋脊边缘的字符图像细化方法

乐 宁 李晓天 邵 华 翁世修

(上海交通大学 制造技术及自动化研究所 上海 200030)

摘 要 针对字符图像中笔画内部像素点灰度的屋脊型分布特点,提出了一种新的字符图像细化方法.该方法利用字符图像中灰度截面曲线,获取了屋脊型边缘的候选点集,通过对候选点集的进一步筛选和修补,取得了定位较为准确的屋脊边缘点,实现了数字字符灰度图像的可靠细化.

关键词 图像处理,字符细化,屋脊边缘提取
中图法分类号 TP391.4

1 引 言

细化处理能极大地消除光学字符图像中的冗余信息,因而在字符图像的存储、识别、处理、重建等领域具有广泛的应用前景.通过细化处理获得字符图像的中心轨迹,不但实现了线条宽度归一化,而且可以以简单的数据结构表示字符,使得机器分析处理、识别的效率大大提高.研究人员对此进行了广泛的研究^[1-3].目前,大部分细化算法都是通过对边缘像素迭代剥除进行的,根据 Stefaneli^[1] 和 Rosenfeld^[2] 的定义,这样的细化算法又可以分为串行、并行、混合及其它.其中,串行法速度较快,并行法容易实现.

通过对变速器壳体号码的字符图像的大量实验我们注意到:灰度字符图像中的字符笔画区域实际上就是极为典型的屋脊型边缘,本文研究的主要内容就是利用字符灰度图像笔画内部灰度分布的这一结构特点,通过提取字符图像中的屋脊型边缘,来解决字符图像的细化问题.这样做的最大好处是在避免了以往的二值化过程对细化的影响的同时,还最大限度地利用了字符笔画内部的灰度分布信息,不

但能高效可靠地实现字符图像的细化,还可以很好地保证细化后字符结构的完整性,为字符图像的后续处理打下了良好的基础.

2 算法原理

边缘检测在计算机视觉中具有相当重要的意义,也是在计算机视觉领域内研究得较多的内容之一,但目前的研究对象都主要集中在阶跃边缘或阶梯边缘,对屋脊边缘的提取则研究较少.

图像处理中遇到的屋脊型边缘有两大类,上凸型(低—高一低)屋脊边缘和下凹型(高一低—高)屋脊边缘,这两类屋脊型边缘没有实质上的差别.为了讨论方便,此处仅考虑实际字符图像处理中出现频率更高的下凹型屋脊型边缘,即边缘区域内部的像素灰度值相对背景像素灰度值较小的屋脊边缘.下凹型屋脊型边缘的截面的模型如图 1 所示.由图 1 可以看出,屋脊型边缘的最重要的一个参数是边缘的尺度值,字符的屋脊型边缘的各种特征只有在一定的尺度范围内才有可能显现出来^[4],若选取的尺度过小则只能显示屋脊型边缘的某一局部,无法

显示出边缘的整体特征;若选取的尺度值过大,则有可能引入许多无关信息,有时甚至会发生边缘信息被淹没的现象,此时,边缘特征的分析 and 提取将无法进行。

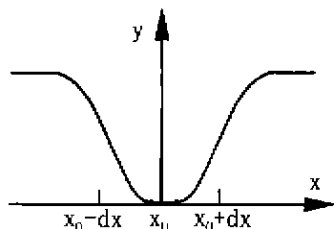


图1 一维屋脊型边缘模型

通常情况下,边缘具有如下这些特征^[5]:(1)具有邻域特征突变性,即在边缘像素所在邻域内像素灰度统计特征发生突变,这是由边缘结构的固有特点决定的;(2)具有方向性,在边缘方向上的像素统计特征与其它方向上的像素的统计特征之间表现出明显的差异,对下凹屋脊型边缘而言,在边缘方向上灰度值较小的像素点将分布比较集中,而在其它方向上的像素点则不具有这一分布特性。

2.1 屋脊边缘候选集的选定

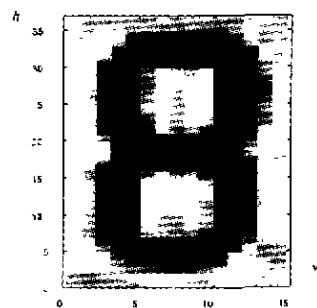
屋脊边缘检测中的主要参数有两个:屋脊边缘区宽度 d 和图像的搜索截断灰度值 G_{limit} , 都可以通过图像的过渡区理论来获得^[6]。

必须明确指出的是,这里希望提取的像素点是图像中的屋脊结构的顶点集,而不是通常意义下的图像中的一阶差分滤波的局部极值点的集合。屋脊边缘候选集的选取是指利用字符图像中笔画内部的像素灰度的屋脊型分布特点,给出边缘的顶点集合或相应的候选点的过程。由于在字符图像中屋脊边缘走向上的多样性,在任何方向上都不可能全部确定出边缘的顶点集。但是,可以得出其这些顶点的替代点——顶点候选点。

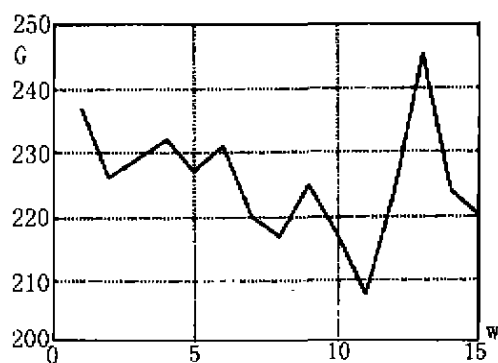
考虑到在高度方向上往往更能反映出字符图像之间的差别,这里将主要从高度方向上入手,逐行确定屋脊型边缘的顶点或候选点。以下将以数字8的图像为例,逐步说明屋脊边缘候选集的确定过程。

对于图2所示的数字8图像,图中的 h 轴为字符图像的高度, w 轴为字符图像的宽度。经过计算得到的尺度参数 $d = 4$, 搜索截断灰度值 $G_{limit} = 100$ 。在不同高度上的典型灰度截面图如图2所示。对于图2(a)所示的字符图像,高度值为2(完全为背景区)、高度值为9(存在着两个屋脊型边缘点)及高度值为19(截面方向与边缘方向基本平行)时图像的灰度截

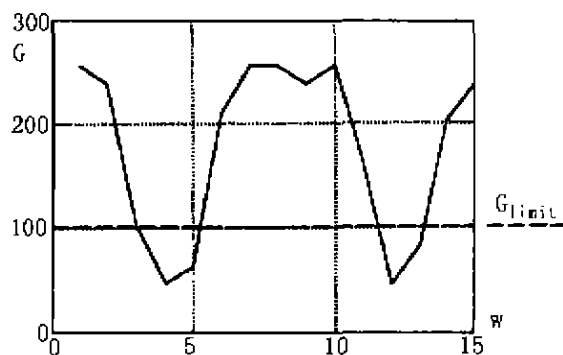
面图如图2(b)、(c)和(d)所示。



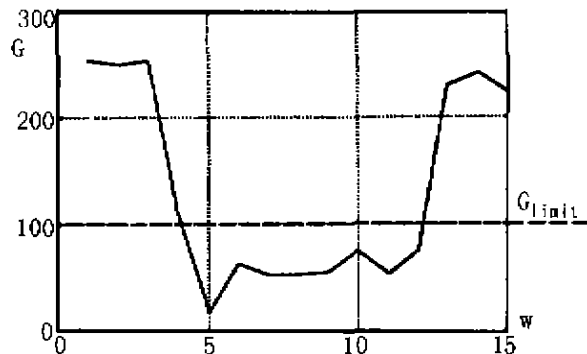
(a) 示例图像



(b) 高度值 $h = 2$



(c) 高度值 $h = 9$



(d) 高度值 $h = 19$

图2 字符图像及其灰度截面曲线

由图 2(b) 可以看出, 当给定字符图像中某一行上不存在屋脊边缘时, 其中的全部像素的灰度值均远大于搜索截断值 G_{limit} . 因此, 若在搜索过程中遇到这种情况, 即可直接判定出在该行范围内不存在屋脊型边缘.

图 2(c) 中的灰度截面曲线则表明, 当给定行上存在与字符高度方向平行或接近平行的屋脊边缘时, 会在灰度截面曲线上反映出明显的灰度峰值, 且其峰值的个数与该行范围内的边缘点个数相一致. 若以搜索截断值 G_{limit} 作为搜索的起点同样可以较为方便地寻找到各峰值的所在位置.

图 2(d) 则表明当某一行上存在与字符高度方向垂直或接近垂直的屋脊型边缘时, 其灰度截面曲线上会反映出一定范围的灰度平台(或在这一范围内只有较小幅度的灰度波动), 以搜索截断值 G_{limit} 作为搜索的起点则可以较为方便地确定出这一灰度平台的大致范围.

屋脊边缘候选集的选定正是基于对字符图像的灰度截面图基本形状的分析. 由此, 有如下屋脊边缘候选集的搜寻算法;

Step 1 利用过渡区理论计算给定图像的搜索灰度截断值 G_{limit} 和边缘区宽度值 d ;

Step 2 从第一行开始逐行扫描字符图像, 并作出各像素行的灰度截面曲线;

Step 3 以搜索灰度截断值 G_{limit} 为起点, 搜寻整个灰度截面曲线的最小值 G_{Min} , 并纪录最小值的位置 w_{Min} ;

Step 4 对于单个数字字符而言, 在其灰度截面曲线上最多只存在两个理想的极小值点; 但对于汉字、英文字母之类的字符而言, 此类极小值点的个数往往要多于 2 个. 由于此处为数字字符, 故仅需判断是否存在第二个极小值点. 类似地, 以搜索灰度截断值 G_{limit} 为起点, 在范围 $(w_{Min} - d/2, w_{Min} + d/2)$ 以外搜寻灰度截面曲线的第二个极小值点 G_{Sec} , 若存在第二个极小值点, 则纪录该极小值点的位置 w_{Sec} , 然后转入第 6 步, 否则转入下一步;

Step 5 若在搜索灰度截断值 G_{limit} 以下只有一个极小值点 G_{Min} , 此极小值所在的像素点即为该行上唯一的屋脊边缘候选点, 通常情况下该候选点就是屋脊边缘上的点, 纪录该点的位置 w_{Min} 并加注标记“0”后转入第 9 步;

Step 6 若按第 4 步搜寻到两个极小值 G_{Min} 和 G_{Sec} , 则察看灰度截面图上位置 w_{Min} 和 w_{Sec} 之间是否有像素点的灰度值低于搜索灰度截断值 G_{limit} , 若

有这样的像素点则立即转入第 7 步, 若不存在这样的像素点则转入第 8 步;

Step 7 两个灰度极小值间没有像素点的灰度值低于搜索灰度截断值 G_{limit} 的像素点, 表明在灰度截面曲线上确实存在两个单独的灰度峰值, 这就意味着在这一行上存在两个屋脊边缘候选点, 纪录这两点的位置 w_{Min} 和 w_{Sec} , 并加注标记“0”后转入第 9 步;

Step 8 两个灰度极小值点之间存在多个像素点的灰度值低于搜索灰度截断值 G_{limit} , 表明在灰度截面曲线上实际并不存在第二个单独的灰度峰值, 这就意味着在灰度截面图上存在一个灰度平台(其上仅有较小幅度的灰度波动), 此时, 应以 w_{Min} 和 w_{Sec} 为起点分别向左向右搜寻灰度截面曲线上最后一个位于截断值 G_{limit} 以内的点, 并纪录这两点的位置 w_{Left} 和 w_{Right} , 将该像素行上这两个位置间的所有像素点均作为屋脊边缘候选点, 并加注标记“1”后转入下一步;

Step 9 计算下一行像素的灰度截面曲线, 重复以上各步, 直至结束.

可以看出, 在上述算法中, 针对字符图像中灰度截面曲线的几种典型情况, 分别对其进行处理. 在整个处理过程中, 最核心的内容就是灰度截面曲线中峰值的判定和检出, 无需进行其它运算, 因而整个算法的计算速度很快, 可以较快确定出字符图像中候选点集的位置.

2.2 屋脊型边缘的获取

图 3 是通过以上算法获得的字符图像屋脊边缘候选点所组成的图像. 由图 3 可以看出, 字符图像经

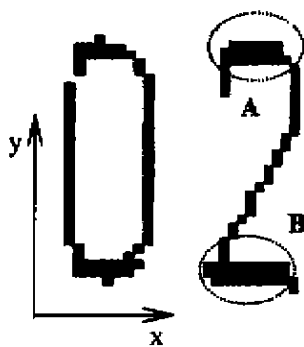


图 3 候选点集组成的图形

过上述处理以后, 获得的候选点要远远多于实际的屋脊边缘点. 这主要是由于算法中采用的是高度方向上的灰度截面曲线, 因此, 字符图像中与高度方向平行或接近平行的屋脊边缘可以较好地反映在灰度截面曲线上, 因而在上述算法中可以直接得到这类

边缘点(在算法中这类像素点被标记为“0”),而其它方向上的屋脊边缘点则无法直接在灰度截面曲线上反映出来,只能给出这类边缘点的大致分布范围,这类屋脊边缘候选点被标记为“1”,有必要对这类候选点进行更进一步地校核和删除,以便获得这些屋脊边缘点的准确位置.

2.2.1 候选点集的筛选

在边缘的候选点集中,标记为“1”的候选点所限定的区域给出的是除垂直方向以外的屋脊边缘的大致范围.通常情况下,在字符图像所在区域内往往会有多个这样的候选区域,且由于字符结构本身的原因,区域与区域之间是互不相连的,图 3 中字符“2”上的 A 和 B 两个区域就是一个典型例子.在每个候选区域内,图像的初始灰度值仍然是典型的屋脊结构,只不过需要从其它方向来检测.因此,对于候选点集中标记为“1”的候选点,有如下的处理过程:

Step 1 将整个候选点集分解为各自独立的候选区域;

Step 2 从上面第一个候选区域开始,寻找区域中的宽度最大的像素行,并纪录该行的行号 y_{Max} ;

Step 3 以该行上的每一像素点 (x_i, y_{Max}) 为搜索的起点,沿字符图像的列向进行搜索,找出各点在候选区域范围内列向上的灰度峰值点 (x_i, y_{peak}) ,并纪录这些点的位置;

Step 4 进入下一个候选区域,重复 2、3 两步,直至结束.

经过以上这些处理步骤以后,获得的点集组成的图形将是笔画只有单像素宽的字符图像,且其中的每一点都是原始图像中字符笔画上的屋脊边缘点,能较好地反映出字符中的笔画走向和字符的结构特征,具有很好的代表性.但是,由于算法中仅仅进行了横向和纵向两个方向的搜索而没有考虑其它方向,而实际上这类屋脊边缘是大量存在的,这时就有可能还有许多边缘点没有检测出来,反映在图形上,就是字符的拐角或连接处中断,对字符图像的结构完整性造成了很大的破坏(如图 4 所示),此时就

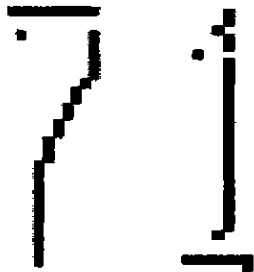


图 4 初步提取结果

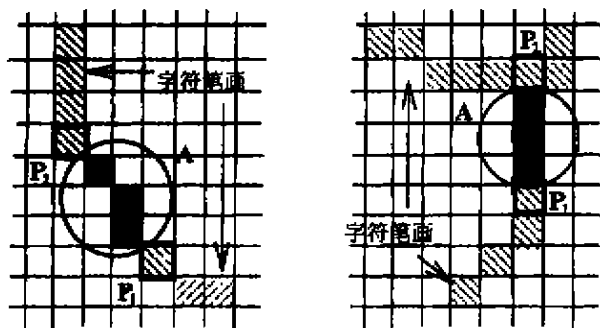
需要对细化字符进行修补.

2.2.2 图形的修补

字形结构完整性是字符图像最重要的特征之一,字形完整性的丧失就意味着字符自身拓扑形状的变化,这将极大的影响字符识别过程的可靠性,因此在候选点集筛选后必须进行字符图形的修补.

目前经常采用的修补方法有两种:模板法和插值法.在修补过程中,必须注意的是,修补点要在尽可能接近原始边缘点的同时,要确保修补后的字符笔画宽度仍为 1 个像素,否则修补过程将没有实际意义.模板法是目前应用最为广泛的修补方法^[3],这里不再详细介绍,但当其应用于笔画裂缝较大的场合时,则显出没有足够灵活性的弱点,这时将不得不求助于插值修补法.

所谓插值修补,是指在字符图像细化后的笔画断裂处的两端点之间做线性插值,并将插值点置为细化后的前景点的过程.在字符笔画修补中,需要插值法处理的情况有如图 5 所示的两种,应对这两类情况应分别进行处理.



(a) 情况一

(b) 情况二

图 5 插值算法的处理对象

图 5(a) 所示的情况是指字符笔画断裂处(图中圆圈 A 所包含的部分)的两端像素(图中为 P_1 和 P_2) 均为盲端,即该像素点只与一个前景点相连接,而图 5(b) 所示的情况是指字符笔画断裂处(图中圆圈 A 所包含的部分)的两端像素(图中为 P_1 和 P_2) 中只有一个是盲端(图中为 P_1),另一个端则为字符笔画上的点(图中为 P_2).

插值算选用的是较为成熟的一次线性插值,运算速度和精度均不存在问题.因此,在插值法中最为关键的步骤是如何可靠地获得插值的两个端点(图 5 中为 P_1 和 P_2).事实上,图 5 中所示的两种情况之间的主要差别就是端点的类型不同.

由于字符笔画的盲端只与一个前景点相连,而在此类图像中的其它前景点至少与两个前景点相连

接,因而很容易识别.字符图像中笔画盲端的具体判别方法如下:

Step 1 从第一行开始逐个扫描待处理图像中的每一个像素,若其为背景点,则跳过该点转向下一个像素点,若为前景点则转入第 2 步;

Step 2 在该像素点的 3×3 邻域内,统计前景点的个数 N . 若 $N = 1$, 转入第 3 步; 若 $N > 1$, 表明该像素点为字符笔画内部的点, 立即返回第 1 步;

Step 3 $N = 1$ 表明该像素点为字符笔画的盲端, 纪录该点的位置和笔画的连接方向;

Step 4 重复 2、3 两步, 直至结束.

确定出字符图像的所有盲端以后, 应当解决的首要问题就是对这些盲端进行分类和组合. 从插值修补的角度来看, 可以将字符中的盲端分为三类: 第一类是字符本身固有的盲端, 字符“5”、“7”及“9”中就会出现此类情况, 此类盲端必须综合考虑多种因素, 在插值修补之前将其剔除; 第二类是与另一个盲端相组合的盲端(图 5(a)); 第三类是与字符笔画内部一点相组合的盲端(图 5(b)). 经过反复实践, 我们发现, 以上三类盲端实际上可以通过距离最小原则来加以区分. 具体实现过程如下:

Step 1 对于给定的盲端序列中的点 P_1 , 找出其周围距离为 D 的范围内除了 P_1 点所在的笔画以外的所有前景点, 并将其归入集合 P . 若点集 P 为空集则表明其为字符上固有的盲端, 将其标记为“1”, 然后转入第 5 步, 否则转入下一步;

Step 2 找出点集 P 中与盲端 P_1 距离最小的像素点 P_2 . 若 P_2 点同样为盲端, 则转入下一步; 若 P_2 为字符笔画内部的点, 则对两点之间的像素点进行一次线性插值, 将 P_1 标记为“1”后转入第 4 步;

Step 3 同样地, 在 P_2 点周围距离为 D 的范围内除了 P_2 点所在笔画以外的所有前景点, 找出其中与 P_2 距离最小的点 P_3 . 若 P_1 与 P_3 为同一个点, 则对两点之间的像素点进行插值, 将 P_2 标记为“1”后转入下一步; 若 P_1 与 P_3 为不同点, 则应对点 P_2 和 P_3 之间的像素点进行插值, 将 P_2 和 P_3 标记为“1”后转入下一步;

Step 4 在插值直线的垂直方向上, 在字符的灰度曲面上寻找距离插值点最近的灰度截面曲线上的极小值点, 并纪录各极小值点的位置, 并用这些极小值点取代相应的线性插值点;

Step 5 将所有标记为“1”的盲端从整个盲端序列中剔除, 重复以上各步, 直至结束.

可以看出, 在以上过程中, 最为关键的一个参数

就是距离 D , 这个参数将直接影响到插值后的字符修补效果. 距离 D 的选定将主要取决于字符笔画的宽度 d , 通常可以取 $D = 0.8d + 1$. 修补后的字符图像如图 6 所示.

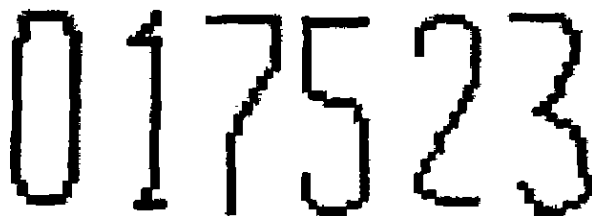


图 6 修补后的字符

3 总 结

本文研究的核心内容是利用字符图像笔画内部灰度的屋脊型分布特点, 通过提取这种屋脊结构的顶点集, 即字符笔画的屋脊型边缘, 来实现字符的细化. 这种算法直接处理灰度图像, 中间无需进行字符图像的二值化处理, 从而避免了因二值化阈值选取不当对字符细化过程的影响, 使细化过程更加稳定可靠. 过渡区理论的运用不但最大限度地利用了字符图像中的各种先验知识, 还有效地减小了判别过程中的搜索区域, 加快了算法的处理速度. 利用模板法和插值法对细化后的字符图像进行修补将会有效地保证字符自身的结构完整性, 这为其后的字符识别打下了良好的基础. 在变速器壳体号码自动识别系统中的实际应用表明, 本方法将会对识别系统的两项最重要性能指标——识别速度和识别率都有相当大的提高, 因而在字符识别和处理领域有着广阔的应用前景.

参 考 文 献

- [1] Chen Y S, Hsu W H. A Systematic Approach for Designing 2-Subcycle and Pseudo 1-Parallel Thinning Algorithm. Pattern Recognition, 1989, 22(3): 267-282
- [2] Stefanelli R, Rosenfeld A. Some Parallel Thinning Algorithm for Digital Pictures. Journal of Computer Association, 1971, 18: 255-264
- [3] 朱学芳, 石膏云. 一种自适应细化方法. 模式识别与人工智能, 1997, 10(2): 140-146
- [4] 杨 恒, 梁德群. 工业图像中屋脊边缘多尺度检测方法. 红外与毫米波学报, 1998, 17(12): 411-416
- [5] 杨 恒, 梁德群. 多尺度边缘检测中滤波尺度自适应调整方法. 西安交通大学学报, 1998, 32(11): 20-23
- [6] 章毓晋. 过渡区和图像分割. 电子学报, 1996, 24(1): 12-17

A CHARACTER THINNING METHOD BASED ON DETECTION OF RIDGES OF THE STROKES

Le Ning, Li Xiaotian, Shao Hua, Weng Shixiu

(Institute of Manufacturing Technology and Automation, Shanghai Jiaotong University,
Shanghai 200030)

ABSTRACT

A new method for character thinning is presented based on ridge features of character strokes. By analyzing the curve of gray values at each cross section in character area, the set of possible ridge edge points is obtained. Then the criteria for selecting ridge points with high location accuracy from the candidate set are discussed. This method has been proved reliable and effective by experiment results.

Key Words Image Process, Character Thinning, Ridge Edge Extraction

2001 年中国智能自动化会议(CIAC'2001)征文通知

会议主办单位

- 中国自动化学会智能自动化专业委员会
- 昆明船舶设备集团公司
- 云南自动化学会
- 中国人工智能学会计算机视觉及智能控制学会
- IEEE 控制系统学会北京分会
- 国际 RoboCup 联合会中国分会筹备委员会

会议地点时间

昆明 2001 年 8 月 13~16 日

会议组织机构

名誉主席:李衍达

主 席:张 钹

副 主 席:涂序彦 于景元 戴冠中 王 珏 孙增圻 吴启迪 徐文立 李 涛 廖伯瑜

秘 书 长:孙增圻 杜继宏

程序委员会主席:涂序彦

组织委员会主席:蒋楚明

组织委员会副主席:许善昆

征文范围

·人工神经网络·模糊系统·进化计算·计算智能及软计算·智能控制·先进控制方法和技术·离散事件系统与混合系统·机器人·多智能体系统·人工智能及应用·智能信息处理·混沌、分形与小波·智能管理与决策·智能设计与制造·智能建模与仿真·智能故障诊断·数据挖掘与知识发现·智能技术在通信与网络中的应用·智能人机交互技术·虚拟现实及多媒体技术·计算机视觉·模式识别与图像处理·智能测量及多传感器信息融合·智能自动化装置·智能交通系统·智能自动化系统及应用·其它

论文要求

·在国内外杂志或会议上未曾发表过。

(下转第 356 页)