

基于模糊边缘检测算法的车牌定位

李庆庆, 张燕平

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘 要: 车牌定位是自动车牌识别系统的一个关键步骤, 车牌定位结果直接影响对车牌的最终识别效果。因此为了保证实际应用中车牌的识别准确率, 文中提出了一种新的车牌定位算法, 该算法利用一种改进的快速模糊边缘检测算法来进行车牌图像的边缘检测, 得到整个原车牌图像的的边缘图像, 然后基于边缘图像的丰富的边缘信息设计一个高效、准确的车牌区域定位算法, 检测出车牌区域。实验结果表明: 算法定位速度较快、准确度较高, 具有良好的应用前景。

关键词: 智能交通; 车牌定位; 模糊边缘检测

中图分类号: TP391.4

文献标识码: A

文章编号: 1673-629X(2006)12-0007-02

License Plate Location Based on Fuzzy Edge Detection

LI Qing-qing, ZHANG Yan-ping

(Ministry of Education Key Lab. of Intelligence Computing & Signal Processing, Anhui Univ., Hefei 230039, China)

Abstract: License plate location is an important stage in vehicle license plate recognition for automated transport system, and it keeps itself in the foreground of widespread availability. A new location algorithm to the identification of number plate was proposed. It detected the license plate image with high speed fuzzy edge detection algorithm. And then design a high speed and accurate license plate location algorithm. It has competitive result.

Key words: intelligence traffic system; license plate location; fuzzy edge detection

0 引言

当前, 智能交通系统 (ITS) 在交通信息统计收集、停车场管理、路车间通信、无停车收费等领域应用越来越广泛。车牌自动识别 (ALPR, Automatic License Plate Recognition) 技术作为 ITS 的一个重要的研究领域也越来越受到人们的重视。

文中讨论的是 LPR 中的车牌定位问题。LPR 一般由 3 个部分构成: 车牌定位; 字符分割; 字符识别。其中车牌定位又是相当重要的一个部分, 定位的结果直接影响后续的处理过程。目前, 用于车牌定位的方法有多种, 如: 1) 直接法, 此算法对于干扰大的图像不是很理想; 2) 神经网络法, 此算法在噪声环境下性能稳定, 但计算量大、定位时间长; 3) 多分辨率方法, 此算法可应用于低分辨率图像, 但需要内存大^[1~5]。文中探讨的是基于模糊边缘检测的车牌定位方法。

模糊边缘检测算法最早是由 Pal 和 King 提出, 该算法应用了模糊思想, 检测效果优异。但由于其算法复杂、

速度慢, 继而又有许多学者提出多种改进的快速算法。文中应用一种由 LIANG L R, LOONEY G L 提出的计算复杂度低且效果好的改进算法用于边缘检测^[1], 该算法比 Pal 和 King 的原算法快近 20 倍, 相比其他几种边缘检测算法, 该算法也性能优异, 如: 与 canny 算法相比, 该算法比 canny 算法检测的边缘清晰, 得到的结果图像质量高, 且速度也比 canny 算法快近 10~20 倍^[1,2]。文中利用这种算法的优异性能, 用其作图像的边缘检测, 并在检测结果上应用图像的纵向和横向的跳变点数进行车牌的准确定位。

1 快速模糊边缘检测算法

该算法源于 Pal 和 King 提出的模糊边缘检测算法, 但简化了复杂的计算步骤, 大大降低了处理时间, 从而使其满足了车牌识别这种实时性要求高的系统的要求^[1,3]。该算法首先对每个像素与其 8-领域像素在不同方向上做差值求和, 然后设计一个 6 类分类器, 用隶属度函数将所考查像素划分到 6 类中, 从而将考查像素划为边缘点和非边缘点, 得出原图像的边缘图像。

算法的计算步骤为^[1~3]:

1) 对彩色图像灰度化, 对其进行灰度拉伸。

2) 将图像的每个非边界区域的像素的 8-领域 (即一个 3×3 的区域), 划分为 4 个方向。如图 1 所示。

收稿日期: 2006-03-02

基金项目: 国家自然科学基金资助项目 (60475107); “九七三”计划 (2004CB318108); 安徽省自然科学基金资助项目 (050420208)

作者简介: 李庆庆 (1981-), 男, 安徽桐城人, 硕士研究生, 研究方向为智能计算; 张燕平, 博士, 教授, 研究方向为人工神经网络、机器学习、人工智能在金融工程中的应用。

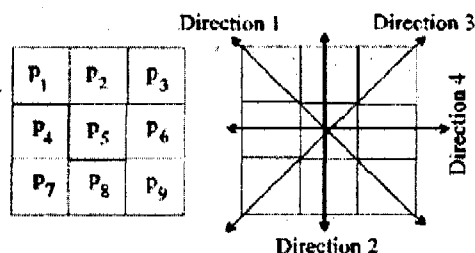


图 1 四方向划分

3) 对每个非边界像素计算四维向量: $x = (d_1, d_2, d_3, d_4)$, 计算方法为:

$$d_1 = |p_1 - p_5| + |p_9 - p_5| \text{ (Direction1)} \quad (1)$$

$$d_2 = |p_2 - p_5| + |p_8 - p_5| \text{ (Direction2)} \quad (2)$$

$$d_3 = |p_3 - p_5| + |p_7 - p_5| \text{ (Direction3)} \quad (3)$$

$$d_4 = |p_4 - p_5| + |p_6 - p_5| \text{ (Direction4)} \quad (4)$$

4) 构造一个六类分类器:

Class 0(background)	$c_0 = (lo, lo, lo, lo)$
Class 1(edge)	$c_1 = (lo, hi, hi, hi)$
Class 2(edge)	$c_2 = (hi, lo, lo, lo)$
Class 3(edge)	$c_3 = (hi, hi, lo, lo)$
Class 4(edge)	$c_4 = (hi, hi, hi, lo)$
Class 5(speckle edge)	$c_5 = (hi, hi, hi, hi)$

其中第 0 类为背景, 第 5 类为噪声, 中间 4 类分别为 4 个方向上的边缘像素。其中的 lo 和 hi 要用户自己设置, 设置的值取决于图像对比度和用户需求, 一般 lo 的范围为 0~5, hi 的范围为 30~50。

5) 对每个像素的四维向量 $x = (d_1, d_2, d_3, d_4)$ 计算 6 个隶属度, 计算公式为:

$$\mu_0(x) = \max \left\{ 0, 1 - \frac{\|x - c_0\|^2}{\omega^2} \right\}$$

取隶属度最大的一类为该像素所属的类。其中 ω 的范围要足够大, 从而使得同一像素对不同类的隶属度有一个较好的区分, 一般其值在 200~256 之间。

6) 如果像素属于 0 类, 则为背景像素, 直接置为 255; 如果属于 5 类, 则暂置为 0, 可能属于噪声点; 如果属于其他 4 类, 则可能分别为各个方向上的边缘, 如: 若像素属于 1 类, 则将其与 1 类所代表方向(方向 1)垂直的方向(方向 3)上的领域点作比较, 如果该像素为局部极大, 则其为所在方向(方向 1)上的边缘点, 置为 0; 否则, 置为 255。

7) 去除噪声点。

至此图像的边缘已检测出来, 此算法检测出的结果边缘清晰, 为后续的分割和识别提供了可靠、优质的图像边缘。此算法的重点在于几个阈值的设定, 文中的值为 256, lo 值为 1, hi 值为 67。不同的阈值有不同的图像检测结果, 有时差别还很大, 如何根据图像动态调整阈值以取得最佳检测结果, 将是下一步的研究方向。

2 基于跳跃点计算的车牌定位

根据所得的检测图像, 根据图像上丰富的边缘信息, 设计如下算法, 通过检测跳跃点数来确定车牌的上下、左

右边界^[4]。

2.1 车牌水平定位

对边缘图像从下到上进行行扫描, 为了提高速度, 扫描的间隔像素数为 15~20, 一般为车牌的高度的一半, 这样可以保证扫描线经过车牌区域。对每一扫描行计算每行的跳跃点数, 当跳跃点数大于 V_1 时, 记下此时的行号。经过第一遍扫描得到车牌的几个候选区域。对每一候选区域(即扫描线), 分别向上和向下检测, 当行跳跃点数小于 V_2 时, 停止此方向上的扫描。此时得到候选区域的高度, 当 $0 < \text{高度} - \text{车牌高度} < V_3$ 时, 确定此为车牌区域。所得到的上下方向的终止行数即位车牌区域的开始行(sline)和结束行(eline)。由此车牌的水平定位结束。其中, V_1, V_2 由车牌区域的字符数而定, 如: 如果 7 个字符边缘全部构造出来则至少会产生 14 个跳跃点, 则 V_1 的值大于等于 14。 V_3 值是为了适应车牌边缘检测不完整而产生的误差。文中的 $V_1 = 14; V_2 = 12; V_3 = 5$ 。

2.2 车牌的垂直定位

检测从开始行(sline)到结束行(eline)之间的一个矩形区域, 计算其在水平线上的像素值投影。其中令 platenlenth 为车牌长度。

第一步, 从第一列开始, 逐列进行检测, 每列的检测范围为从 sline 开始到 eline 结束。

第二步, 对第 N 列, 将从 sline 开始到 eline 结束的每一行的像素值相加得到 $m(n)$ 。

第三步, 重复第二步, 直至最后一行。

第四步, 将每一列的像素值和按公式 $v(n) = m(n) / (\text{eline} - \text{sline} + 1)$ 进行计算, 如果 $v(n) < \partial_1$, 则 $v(n) = 1$; 否则 $v(n) = 0$ (其中 ∂_1 的设定依据原始图像大小和车牌比例而定, 文中为 200)。

第五步, 对于得到的 n 维向量 (0000...111111...100110...), 从第一维开始, 以 5 为步长向后搜索, 至第 $n - \text{platenlenth}$ 维结束(其中, 为保证将车牌区域完整地包含在最后的边界中, 使得 $\text{platenlenth} = \text{platenlenth} + 5$)。

第六步, 对于上面搜索到的每一维 n , 计算从第 n 维到第 $n + \text{platenlenth}$ 维, 共 platenlenth 维的值的和。当上面的搜索结束时, 所得的和最大的即为车牌区域, 车牌的左边界即为开始的维数, 右边界为开始维 + platenlenth 。

3 结 论

文中的试验图片均为 768×576 像素的车辆图片, 图片来源于实地的车辆图像拍摄。对 78 幅采集图片进行定位, 并对其中一幅图片(如图 2 所示, 其中图 3 为图 2 定位后的图像)分别运用文中算法、v_soble 算法、gauss 算法进行边缘检测, 得到如图 4~6 所示的 3 幅车牌区域的边缘检测图片。所有的 78 幅图片中除两幅图片因为车牌周围边缘信息过于集中未能准确定位外, 其余均达要求, 准确定位率为 97%。前面的模糊边缘检测算法平均时间为

(下转第 12 页)

时存在以下几点局限性:

- (1) 不是面向树型结构的;
- (2) 选择/排斥集需要使用另外的方式表达;
- (3) GUI 表现困难;
- (4) 需要用户了解正则表达式的匹配命令。

表 2 正则表达式表示

步骤	规则	资源指定
1	备份	D:/*
2	不备	D:/RECYCLE/*
3	不备	D:/Share/*
4	不备	D:/temp/*
5	不备	D:/test/*
6	不备	D:/BackupApp.rar
7	备份	D:/test/javaxml2/javaxml2/HelloClient.java
8	备份	D:/test/javaxml2/javaxml2/HelloHandler.java
9	备份	D:/test/javaxml2/javaxml2/HelloServer.java

3 结束语

文中针对在周期性任务处理中树型结构数据的定义提出了真前缀标记树的表示方法,该方法使用一个三元组表示用户对节点数据的选择/排斥策略,通过一组变化规则进行树节点的状态转换,并给出一种 GUI 的表现形式,说明真前缀标记树方法不仅易于在 GUI 中表现,也使用户对操作规则的理解更简单、更直观。

(上接第 8 页)

230ms,第二部分的车牌定位算法平均时间为 12ms,相比同类算法亦有提高,且算法简单,边缘清晰、定位准确。只是由于某些原因(如汉字模糊),未能清晰构画出汉字的边界。

文中算法改进在于将模糊边缘检测算法运用到车牌

参考文献:

- [1] Tansley D. Linux and Unix Shell Programming[M]. [s. l.]: Addison Wesley/Pearson, 2000.
- [2] Bray T, Paoli J, Sperberg-McQueen C M, et al. Extensible markup language (XML) 1.0 (2nd Edition)[EB/OL]. W3C Recommendation, 2000. <http://www.w3.org/TR/2000/REC-xml-20001006>.
- [3] Clark J, DeRose S. XML Path Language (XPath), Version 1.0[EB/OL]. W3C Recommendation, 1999-11. <http://www.w3.org/TR/xpath>.
- [4] Boag S, Chamberlin D, Robie J, et al. XQuery 1.0: An XML Query Language[EB/OL]. W3C Candidate Recommendation, 2005-11-03. <http://www.w3.org/TR/xquery/>.
- [5] 王强,武港山.对 XPath 模式定位能力的扩充[J].计算机研究与发展,2001(6):674-678.
- [6] 吴恒山,吴亚辉,班鹏新.XML 查询优化的面向路径可扩展模型[J].华中科技大学学报:自然科学版,2004(3):62-64.
- [7] 王静,孟小峰,王宇,等.以目标节点为导向的 XML 路径查询处理[J].软件学报,2005,16(5):827-837.
- [8] Yazdani N, Min P S. Prefix Trees: New Efficient Data Structures for Matching Strings of Different Lengths[C]//2001 International Database Engineering & Applications Symposium (IDEAS '01). Grenoble, France: IEEE Computer Society, 2001:76-85.

定位中,并利用其优异的性能,节省了计算时间,得到了清晰的边缘图像,为后面的分割和识别节省了计算时间,提供了可靠的边缘信息。并且根据边缘信息,设计了相应的车牌定位算法,定位准确、时间短,适用于车牌识别这种适时性要求高的系统,具有良好的应用前景。

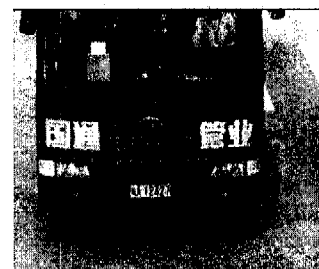


图 2 原始图像

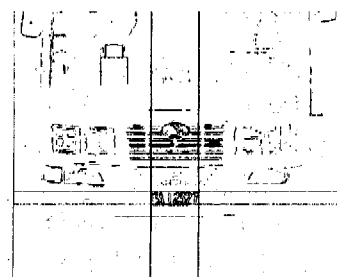


图 3 定位后的图像

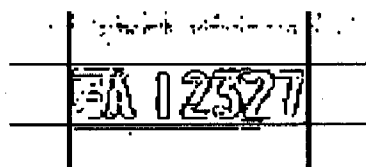


图 4 文中算法检测的边缘

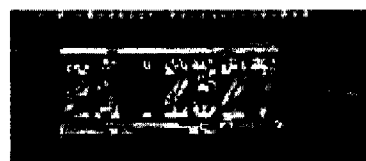


图 5 v-soble 算法检测的边缘



图 6 gauss 算法检测的边缘

参考文献:

- [1] LIANG L R, LOONEY G L. Competitive fuzzy edge detection[J]. Applied Soft Computing, 2003(3):123-137.
- [2] 段震,姚芳兵,张铃.基于构造性学习方法的车牌定位[J].微机发展,2004,14(8):41-43.
- [3] 鲁继文,张二虎.基于分类器的图像模糊边缘检测快速算法[J].计算机应用,2005(10):2374-2375.
- [4] Zheng Darian, Zhao Yannan, Wang Jiaxin. An efficient method of license plate location[J]. Pattern Recognition Letters, 2005(26):2431-2438.
- [5] Li Xiaobo, Liu Zhi-Qiang, Lenug Ka-Ming. Detection of vehicles from traffic scenes using fuzzy integrals[J]. Pattern Recognition, 2002(35):967-980.