

一种简单快速的车牌定位算法^①

龚成清

(广东女子职业技术学院 应用设计系, 广州 511450)

摘要: 车牌定位是汽车牌照识别系统的关键技术之一。通过对图像进行预处理, 去除干扰的噪声, 然后利用隔行扫描和矩形窗口搜索法来定位车牌的上下边界和左右边界, 从而提高了车牌定位的速度。提出了利用车牌定位出来的区域面积的变化情况来判断车牌的倾斜情况, 并对倾斜的车牌进行了准确的校正。实验结果表明, 该算法简单高效。

关键词: 车牌定位; 二值化; 隔行扫描; 矩形窗口; 倾斜校正

A Simple and Efficient Algorithm for License Plate Localization

GONG Cheng-Qing

(Department of Application and Design, Guangdong Women Polytechnic, Guangzhou 511450, China)

Abstract: License plate location is one of the key technologies in license plate recognition system. By preprocessing the license plate image to remove the interference noise, this paper uses the method of interlaced scanning and the method of rectangular window searching to location the license plate's top border, bottom border, left border and right border. So it greatly improves the speed of license plate location. This paper also proposes that the changing of location area can judge the tilt condition of the license plate and gives out a correction method of the tilt license plate. The experiments show that this algorithm is simple and efficient.

Key words: license plate location; binarization; interlaced scanning; rectangular window; tilt correction

汽车牌照识别技术(LPR)是智能交通系统的重要组成部分,大量应用于停车管理、自动计费管理、交通运输管理等领域。汽车牌照识别技术包括车牌定位、字符分割和字符识别三大部分^[1-8]。车牌定位就是要从复杂的背景中给出车牌的准确位置,它是汽车牌照识别技术中的关键步骤,其定位的准确性直接影响到后续识别的结果。研究者已经给出了很多有效的车牌定位算法,为研究积累了宝贵的材料。文献[1]提出了基于Hough变换法对车牌进行定位,利用车牌具有矩形边框这一先验知识,通过对矩形边框直线边缘提取,再结合Hough变换检测出车牌边框直线来定位车牌,这种方法对规范的车牌具有很好的识别效果。文献[2,3]提出了基于纹理分析的定位方法,不依赖于车牌的尺寸大小、拥有对光照条件、路面状况、车牌倾斜角度等比较好的免疫作用。文献[4-6]给出了基于边

缘特征和数学形态学的定位方法,该方法通过腐蚀和膨胀以及数学运算来定位车牌,能很好地滤除图像噪声,具有较快的运算速度和较高的准确度。文献[7,8]给出了基于神经网络的定位方法,该方法具有较强的自适应能力,可以在较复杂背景中定位车牌。本文在前人研究的基础上,基于边缘特征利用投影的技术和投影面积的变化对车牌进行了精确的定位。

1 图像灰度化

目前,汽车牌照识别系统通常采集到的都是彩色图片。彩色图像中的每个像素的颜色有R、G、B三个分量决定,每一个像素点可以有1600多万的颜色值,需要大量的存储空间,不利于图像的快速处理,而灰度图像R、G、B三个分量都相同,其每一个像素点的变化范围为255种,大大减少了存储的空间和计算量。

① 收稿时间:2010-12-22;收到修改稿时间:2011-01-18

灰度图像的描述仍然反映了整幅图像的整体和局部的色度和亮度等级的分布和特征,所以在车牌定位处理中一般先将彩色图像转变成灰度图像来提高处理的速度。根据 YUV 的颜色空间中 Y 与 R、G、B 三个颜色分量的对应关系,以 Y 这个亮度值表达图像的灰度值。亮度值 Y 与 R、G、B 三个颜色分量的对应关系如公式 (1):

$$Y=0.3R+0.59G+0.11B \quad (1)$$

利用公式 (1) 把彩色图像转换为灰度图像后,为了更好地突出车牌的区域,需要对灰度图像进行拉伸和图像增强处理,突出对比度。图 1 (b) 是经过拉伸和增强处理后的图片。



(a) 采集到的彩色图像 (b) 灰度图像

图 1 经过灰度处理的图像效果

2 图像二值化

对采集的图像进行灰度处理后需要对图像进行二值化的处理来区分前景和背景。车牌图像的二值化的好坏直接影响着车牌的定位。对灰度图像进行二值化要设定一个阈值,如公式 (2)。根据阈值来判断像素是黑还是白。

$$g(Y) = \begin{cases} 1 & \dots \dots f(Y) \geq T \\ 0 & \dots \dots f(Y) < T \end{cases} \quad (2)$$

公式 (2) 中, T 为阈值,当灰度图像的值大于或等于 T 的时候取 1, 否则取 0, 可见图像二值化的关键在于阈值的选取。阈值的选取常用的有全局阈值、局部阈值和自适应阈值三种方法^[3]。典型的全局阈值法是 Otsu 方法^[6]。Otsu 算法是在判别最小二乘法的基础上推导出来的,其基本思想是把图像分为背景和目标两类,按照使类间方差最大的原则来设置阈值。本文采用 Otsu 方法选取阈值对图像进行二值化转换。

由于图像在采集的过程中会受到光照不均、污迹、杂物等的干扰,利用 Otsu 算法对灰度图像进行二值化处理后,会出现一些孤立的像素点。为了减少这些噪声的干扰,需要对图像进行平滑处理。本文采用中值

滤波的方法对图像进行平滑处理,有效地去除了干扰的噪声。经过平滑处理的二值化图像如图 2 所示:



图 2 经平滑处理的二值化图像

3 车牌定位

3.1 车牌上下边界的定位

对经过二值化处理的图像分析发现,车牌区域字符为白色,车牌区域背景为黑色。车牌字符集中在一个区域,从左到右每个字符至少存在左右两次黑白跳变(即由 0 到 1 和由 1 到 0 的跳变),具有鲜明的纹理特征。结合车牌具有连续 7 个字符、规律的字符间距和每个字符至少存在两次跳变等先验知识,从左到右扫描车牌区域,车牌区域的字符至少存在 14 次 0 到 1 之间跳变,则车牌字符区域的每一行满足公式(3):

$$\sum |f(x, y+1) - f(x, y)| \geq 14 \quad (3)$$

利用公式 (3) 对车牌进行水平投影来确定车牌的上下边界。

传统的水平投影是自上而下从左到右逐行扫描车牌图像,这种方法速度是比较慢的。基于大部分车牌都位于图像下半部的先验知识,本文采用自下向上隔行扫描的方式来确定车牌的上下边界。具体的算法为:

① 自下向上隔行扫描车牌图像,设 $\Delta=3$ 为扫描的步长,即每隔 3 行扫描一次图像;

② 把第一次检测到跳变 ≥ 14 的扫描线位置记为 n_1 , 最后一次检测到跳变 ≥ 14 的扫描线位置记为 n_2 ;

③ 把 Δ 改为 1, 从 n_1 开始往后回溯逐行扫描,看是否还有跳变 ≥ 14 的扫描线,如果有把最后一条的位置记为 n_1 ; 如果没有,则 $n_1=n_1$, 即为车牌区域的下边界。从 n_2 开始继续往上逐行扫描,看是否还有跳变 ≥ 14 的扫描线,如果有把最后一条的位置记为 n_2 ; 如果没有,则 $n_2=n_2$, 即为车牌区域的下边界。

采用隔行扫描的方式扫描的次数为 $(n_2/\Delta + b)$, b 为 n_1 、 n_2 逐行扫描的次数, $b \leq 2\Delta$ 。可见,适当地选取 Δ 的值采用隔行扫描的方式可以大大加快投影

的速度。经过扫描，车牌的上下边界如图 3 所示：



图 3 车牌的上下边界

3.4 车牌左右边界的定位

车牌左右边界的定位一般是对车牌区域进行垂直投影，然后分析其投影图的波峰和波谷的位置，结合车牌的先验知识来定位左右边界。这种定位方法实现简单，但对于有噪声干扰的车牌效果误差较大。

本文采用矩形窗口搜索来定位车牌的左右边界。在定位好车牌区域的上下边界后，利用车牌有固定尺寸和固定的长宽比例的先验知识，由采集的图像统计出车牌区域的大小，记为 M_{ixj} 。设置一个与 M_{ixj} 一样大小的矩形窗口，抽取车牌区域上下边界之间的图像 G_{ixm} （设图像有 i 行， m 列），对矩形窗口进行搜索，利用车牌区域中所有的行跳变均 ≥ 14 的事实，计算跳变 ≥ 14 的行数 sum 与总行数 i 的比例是否大于等于 1 来判断矩形区域是否为车牌区域。具体的算法是：

① 图像 G_{ixm} 中，从第一列开始以 M_{ixj} 的矩形窗口截取图像，设置阈值 $T=1$ ；

② 矩形窗口 M_{ixj} 中进行水平扫描，利用公式（4）检测跳变 ≥ 14 的行数累加到变量 sum ：

$$sum = \sum_{x=1}^i count \left(\sum_{y=1}^j |f(x, y+1) - f(x, y)| \geq 14 \right) \quad (4)$$

③ 较 sum/i 与 T 的关系，若 $sum/i \geq T$ ，记下矩形窗口左右边界的位置，并把矩形窗口放到车牌区域候选集合 A 中；否则，选取下一列继续用 M_{ixj} 的矩形窗口截取图像；

④ 矩形窗口遍历完整个图像后，检测集合 A ，如果集合 A 中的矩形窗口个数是 1，则 A 中的矩形窗口的左右边界即为车牌区域的左右边界；如果 A 中的矩形窗口个数大于 1，则增加矩形窗口的宽度（即增加 M_{ixj} 中的列数 j ）跳转步骤②，重新定位；若 A 中的矩形窗口个数为 0，则调整 T ($T = T - 0.1$)，跳转步骤②，重新定位。

利用矩形窗口搜索法，可以有效地去除噪声的干扰，在定位好上下边界的图像上进行搜索也大大减少了运算了，可以快速地车牌进行定位。图 4 是定位

后的结果：



图 4 牌定位的结果

4 倾斜校正

车牌由于悬挂不规范、拍摄角度等问题会出现倾斜变形的情况，采用上述的方法对倾斜的车牌定位会出现较大的误差，如图 5 所示：



图 5 车牌倾斜的几种情况

图 5(a)(b)的车牌在水平倾斜的情况下定位范围明显偏小，(c)(d)在垂直倾斜下字符有明显的变形，影响着后面字符分割的正确性，所以必须要对倾斜的车牌进行校正。文献[1]利用了 Hough 变换，检测车牌图像的边缘直线来确定车牌水平倾斜的角度后进行校正。这种方法对于有明显边缘直线的车牌图像效果很好。但经过图像二值化处理后，很多车牌图像并没有明显的边缘直线了，而且 Hough 变换只能检测出水平的倾斜情况，不能校正垂直方向的车牌倾斜。本文利用倾斜车牌定位出来的车牌区域面积的变化情况来判断车牌倾斜的情况，再根据倾斜的情况进行校正。

4.1 车牌倾斜情况的判断

基于摄像头拍摄角度固定、摄像头与汽车之间的距离固定、摄像头拍摄触发的条件相同、车牌有规定大小等先验知识，统计采集的图像，我们很容易计算出一个正常车牌区域的面积大小，记为 S 。我们把利用投影和矩形窗口法定位出来的车牌区域面积记为 S' 。观察车牌水平倾斜和垂直倾斜定位出来的区域发现，水平方向倾斜的车牌定位出来的车牌区域面积 S' 明显比 S 偏小。垂直方向倾斜的车牌由于字符拉伸变长，利用三角函数可以证明定位出来的车牌区域面积

$S' > S$ 。具体的证明如下：设正常车牌区域的矩形框长为 L ，高为 H ，垂直方向倾斜的角度为 θ ，如图 6 所示。则：

$$S = H \times L \quad (5)$$

$$S' = H \times \cos\theta \times (L + H \times \sin\theta), \quad (0 < \theta < 90^\circ) \quad (6)$$

由 (5) (6) 式可得：

$$S' / S = \cos\theta + \sin\theta \quad (7)$$

由 (7) 两边平方可以得出

$$(S' / S)^2 = 1 + 2 \cos\theta \sin\theta > 1, \quad (0 < \theta < 90^\circ) \quad (8)$$



(a) 正常车牌区域面积 (b) 垂直倾斜车牌区域面积
图 6 车牌区域面积示意图

根据定位车牌区域的面积与正常车牌区域的面积的比值，利用公式 (9) 可以准确地判断出车牌的倾斜类别：

$$f(S') = \begin{cases} 1 \dots \dots S' / S < 1, \text{水平方向倾斜} \\ 0 \dots \dots S' / S = 1, \text{无倾斜} \\ -1 \dots \dots S' / S > 1, \text{垂直方向倾斜} \end{cases} \quad (9)$$

4.2 水平方向倾斜车牌校正

根据车牌的倾斜情况，要进行不同的校正。在水平方向的车牌校正，我们对图像进行顺时针或者逆时针旋转的方法重新投影来定位车牌区域，如果校正的方向正确，其定位出来的车牌区域会越来越大，也越来越接近正常车牌的区域面积，图片在旋转的过程中也自动地进行了修复。具体的算法为：

① 把原来的车牌图像顺时针旋转 1° ，利用投影方法和矩形窗口搜索法重新定位车牌区域，计算其面积，记为 S'' ，若 $S'' / S \geq S' / S$ ，则说明校正的方向正确；否则回退到初始状态，再逆时针旋转 1° ，同样计算其面积，记为 S'' 。

② 重复步骤①，直到 $S'' / S = 1$ 。

4.3 垂直方向倾斜车牌校正

传统的垂直方向倾斜的车牌校正非常的麻烦，要先利用不同角度的投影计算出最小投影，然后利用最小投影的角度算出倾斜的角度。本文利用垂直方向倾斜车牌定位区域面积的变化可以很快地算出倾斜的角度，垂直方向倾斜车牌的校正不需要一行行地进行移

动，而是一次完成，大大提高了校正的效率。下面以垂直向前倾斜的车牌校正来进行说明：

利用垂直方向倾斜车牌定位区域面积的变化，由公式 (8) 可以直接算出垂直倾斜的角度：

$$\theta = \arccos[(S'' / S)^2 - 1] / 2 \quad (10)$$

以图 6(b) 矩形的左上角的顶点为坐标原点，由于图像倾斜的角度是 θ ，则要校正车牌，车牌像素的每个点都要向左进行平移，每个点平移的距离由公式 (11) 确定。

$$\varepsilon = (H \cos\theta - y) \tan\theta \quad (11)$$

利用公式 (11)，车牌的每个像素向左平移 ε 个元素即可对垂直向前倾斜的车牌进行校正。由于垂直向后倾斜的车牌与垂直向前倾斜的车牌具有对称性，因此垂直向后倾斜的车牌校正只需要把每个像素向右平移 ε 个元素即可。具体校正的算法为：

① 利用公式 (10) 计算 θ ，车牌图像的每个像素向左平移 ε 个元素；

② 对平移校正后的车牌图像重新定位计算其面积，记为 S'' ，若 $S'' / S \leq 1$ ，说明是垂直向前倾斜的车牌校正，校正方向正确，退出程序；否则，说明车牌图像是垂直向后倾斜的车牌，跳转③；

③ 把车牌图像回退到初始状态，对车牌图像的每个像素向右平移 ε 个元素，完成校正。

5 实验结果

上述的算法在 VC++6.0 进行了实现，测试的平台是在 window xp，机器是 P4 2.4GHz、2G 内存。利用本文所提的算法和 Hough 变换法以及数学形态法分别对从某停车收费站收集的 200 个车牌图像样本、网上收集的 100 个车牌图像样本进行了测试，样本中包括了不同光照条件和不同背景等样本和大角度倾斜的车牌样本（倾斜的角度 $> 30^\circ$ ）。测试结果如表 1：

表 1 三种不同算法的测试结果

算法	样本总数	定位正确数	定位失效数	比率
Hough 变换法	300	286	14	95.3%
数学形态法	300	290	10	96.7%
本算法	300	292	8	97.3%

实验的结果表明，算法简单有效，可以快速地对车牌进行准确定位并且可以校正倾斜角度较大的车牌。

(下转第 233 页)

$$\begin{aligned}x_1 &= x_0 + L; & z_1 &= z_0 + L; \\x_2 &= x_0 + 2L; & z_2 &= z_0 + 2L; \\x_3 &= x_0 + 3L; & z_3 &= z_0 + 3L; \\x_4 &= x_0 + 4L; & z_4 &= z_0 + 4L; \\x_5 &= x_0 + 5L; & z_5 &= z_0 + 5L;\end{aligned}$$

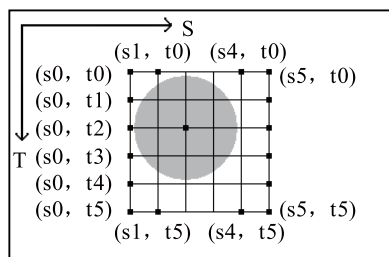


图 12 光环网格顶点的纹理坐标

上面公式中, $\lfloor \cdot \rfloor$ 符号表示向下取整, 例如 $\lfloor 8.9 \rfloor = 8$, $\lfloor -8.1 \rfloor = -9$ 。由纹理坐标原理及图 12 可知光环网格各顶点的纹理坐标公式为:

$$\begin{aligned}s_0 &= - (x_c - x_2) / (4L); \\t_0 &= - (z_c - z_2) / (4L); \\s_1 &= s_0 + 0.25; & t_1 &= t_0 + 0.25; \\s_2 &= s_0 + 0.25 \times 2; & t_2 &= t_0 + 0.25 \times 2; \\s_3 &= s_0 + 0.25 \times 3; & t_3 &= t_0 + 0.25 \times 3; \\s_4 &= s_0 + 0.25 \times 4; & t_4 &= t_0 + 0.25 \times 4; \\s_5 &= s_0 + 0.25 \times 5; & t_5 &= t_0 + 0.25 \times 5;\end{aligned}$$

如果光环网格有由 $n \times n$ 个单元格组成(n 为奇数), 则各顶点的位置平面坐标为:

$$x_0 = \lfloor x_c / L \rfloor \times L - \lfloor n / 2 \rfloor \times L;$$

$$z_0 = \lfloor z_c / L \rfloor \times L - \lfloor n / 2 \rfloor \times L;$$

.....

$$x_k = x_0 + kL; \quad (k > 0 \text{ 且 } k \leq n)$$

$$z_k = z_0 + kL; \quad (k > 0 \text{ 且 } k \leq n)$$

$$s_0 = - (x_c - x_2) / [(n - 1) L];$$

$$t_0 = - (z_c - z_2) / [(n - 1) L];$$

.....

$$s_k = s_0 + [1 / (n - 1)] \times k; \quad (k > 0 \text{ 且 } k \leq n)$$

$$t_k = t_0 + [1 / (n - 1)] \times k; \quad (k > 0 \text{ 且 } k \leq n)$$

上面公式中, $\lfloor \cdot \rfloor$ 符号表示向下取整, 例如 $\lfloor 8.9 \rfloor = 8$, $\lfloor -8.1 \rfloor = -9$ 。

4 结束语

以上便是“地面光环”特效的实现算法, 利用这种算法, 还可以制作人物影子等游戏效果。

参考文献

- 1 Lake A. Game Programming Gems8. America: Course Technology PT, 2010.140-203.
- 2 Jacobs S. Game Programming Gems7. America: Charles River Media, 2008. 255-336.
- 3 Walsh P. Advanced 3D Game Programming with DirectX 9.0.1thed.Texas:Wordware Publishing, 2003. 413-481.
- 4 黄蓝泉. 游戏引擎中特效的实时渲染技术研究[硕士学位论文].成都:电子科技大学,2008.
- 5 赵明. 基于 DirectX 的三维游戏特效技术的研究与实现[硕士学位论文].哈尔滨:哈尔滨工程大学,2009.
- 3 柴晓荣,刘锦高. 基于纹理分析的精确车牌定位算法. 计算机系统应用,2010,19(2):160-163.
- 4 王文涛,马启新,杜鹏飞. 基于边缘检测和数学形态学的车牌定位. 中南民族大学学报(自然科学版),2008,27(4):84-87.
- 5 熊春荣,黄文明. 基于复合型数学形态学边缘提取的车牌定位算法. 微计算机信息,2010,26(6):188-190.
- 6 姜谊,严京旗. 车牌定位及倾斜校正方法. 微型电脑应用,2010,26(4):45-47.
- 7 李会民,张仁津. 基于改进 BP 网络的车牌字符识别方法研究. 计算机工程与设计,2010,31(3):619-621.
- 8 宋文强,马义德,何胜宗. 脉冲耦合神经网络在车牌定位中的应用. 计算机工程,2010,8(16):174-179.

(上接第 217 页)

6 结语

本文的创新点是利用水平投影和窗口搜索法定位出来的车牌区域面积的变化来判断车牌倾斜的情况和进行相应的校正和定位。算法的实现简单、定位迅速、可以有效地去除干扰的噪声, 对倾斜变形的车牌具有很好的免疫作用, 有较好的实用价值。

参考文献:

- 1 林俊,杨峰,林凯. Hough 变换与先验知识在车牌定位中的新应用. 计算机与数字工程,2009,37(8):138-140.
- 2 贾小军,喻擎苍,谭召均. 纹理谱描述子及其在车牌定位中的应用. 计算机应用研究,2007,24(3):215-217.