

# 基于灰度跳变与字符间隔模式的车牌定位方法研究

高飞, 梅凯城, 韩政高, 卢书芳, 肖刚

(浙江工业大学 计算机学院, 杭州 310023)

**摘要:** 车牌定位是车牌自动识别的第一步, 而如何考虑光照影响是车牌定位是否成功的关键; 通过深入分析不同的车牌图像, 提出一种基于灰度跳变与字符间隔模式的车牌定位方法: 首先, 针对不同光照条件下采集到的车牌图像明暗度的不同, 利用多阈值处理方法得到车牌信息不丢失的、最佳的二值图像, 然后, 在其二值图像中首先利用灰度跳变定位车牌的上下边界, 接着对字符垂直投影后的宽度进行统一的调整并以固定的字符间隔特征定位车牌的左右边界, 从而完成车牌定位; 最后, 通过实验验证了该方法的有效性。

**关键词:** 车牌定位; 多阈值二值化; 灰度跳变; 投影调整

## A Method of Car License Plate Location Based on Gray Beep and Char Interval Mode

Gao Fei, Mei Kaicheng, Han Zhengao, Lu Shufang, Xiao Gang

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** Car license plate location is first step of car license plate auto-recognition, how to consider the light conditions is one of the keys whether or not the car license plate can be located successfully. First, a multi-threshold method was employed to obtain the optimal binaryzation image according to different lighting conditions. Second, gray beep was facilitated to determine upper and lower edges of the car license plate. Third, left and right edges of the car license plate were located and then, the full location of car license plate can be finished. Finally, the proposed method was verified through experiment.

**Keywords:** car license plate location; multi-threshold binaryzation; the gray beep; projection adjusting

## 0 引言

车牌识别是计算机视觉在智能交通领域中的一个重用研究课题, 准确而快速的车牌定位算法是车牌识别系统实现的基础。

目前, 车牌定位方法主要有: 基于边缘检测的方法<sup>[1-2]</sup>; 基于数学形态学的方法<sup>[3-4]</sup>; 基于彩色信息的方法<sup>[5-6]</sup>; 基于小波变换的方法<sup>[7]</sup>; 基于灰度跳变和投影特征的方法<sup>[8-9]</sup>等。

车牌定位方法虽比较多, 但每一种方法侧重点不同, 各有各的优势, 也各有各的缺陷和不足。受到采集车牌图像时环境的差异性以及车牌图像背景的复杂性的影响, 很难存在一种完美的方法解决所有问题。本文将对基于灰度跳变和投影特征的方法进行改进, 主要侧重于解决在不同光照环境下采集到的图像的车牌定位, 同时对字符投影间隔距离进行合适的调整可以排除具有同字符投影特征类似的伪车牌区域的干扰, 可有效提高车牌定位应用的实用性和准确性。

## 1 车牌先验知识

根据我国最新机动车驾驶证标准即 2007 年 11 月 1 日正式实施的《中华人民共和国机动车号牌》(GA36-2007) 行业标准, 车牌具有两个特征: 1) 车牌由 8 个字符组成, 具有统一的排列格式:  $X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8$ , 其中  $X_3$  为圆点分隔符号

“·”, 其他字符可能是汉字、英文字母和阿拉伯数字; 2) 每个字符占用宽度为 45 mm, 高度为 90 mm。

## 2 车牌图像的二值化

车牌图像通常是由分布在各十字路口、超速监测点、收费站及其他卡口处的抓拍系统所采集, 图像效果与抓拍时的环境具有直接联系, 不同的抓拍环境会导致截然不同的车牌图像。图 1 表示影响抓拍环境的主要因素, 通常情况下抓拍时的光照条件、车牌图像的明暗程度受到这 3 种因素的不同组合的综合影响。实际情况中, 车牌图像的明暗程度很能达到理想状况, 同一套抓拍系统所拍到的车牌图像之间的亮度并不一致, 具有非均匀性特点。根据明暗程度以及车牌是否可见为依据, 抓拍图像通常可以分为三类: 正常、偏暗和漆黑。正常图像表示车牌信息清晰可见; 偏暗表示整副图像色调的灰度值偏低, 但车牌信息仍可见; 漆黑表示整副图像色调灰度值非常低, 车牌信息很难识别。

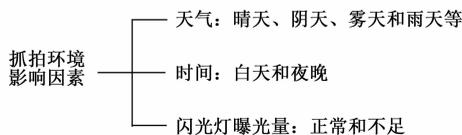


图 1 抓拍环境影响因素

二值化是车牌定位的关键。车牌图像的二值化结果将直接影响是否能成功定位, 而二值化的关键又在于阈值的选择。传统的车牌图像二值化方法通常采用单一阈值(全局阈值、局部阈值或动态阈值)进行处理, 它们对于上述 3 种类型中的正常车牌图像可获得较好的效果, 但对于因 3 种因素组合影响而导致的明暗程度差异较大的车牌图像而言, 其二值化效果并不理想。

收稿日期: 2015-10-26; 修回日期: 2015-12-02。

基金项目: 国家自然科学基金资助项目(C12412135, 61402410); 浙江省自然科学基金资助项目(LY13F020029, LQ14F020004)。

作者简介: 高飞(1974-), 男, 四川邻水人, 博士, 教授, 博士生导师, 主要从事计算机视觉与图像处理方向的研究。

想。为此提出一种多阈值二值化处理方法，算法如下：

1) 对原始彩色车牌图像采用加权平均值法进行灰度化处理：

$$f(x,y) = \frac{299R(x,y) + 587G(x,y) + 114B(x,y)}{1000} \quad (1)$$

其中： $R(x,y)$ 、 $G(x,y)$  和  $B(x,y)$  分别表示  $(x,y)$  处的像素的红、绿、蓝分量， $f(x,y)$  表示像素  $(x,y)$  的灰度值。

2) 分析车牌灰度图像中亮度的分布特征，判断该图像是光照正常或光照异常：

$$M = \frac{\sum_{x=0}^{h-1} \sum_{y=0}^{w-1} f(x,y)}{h \times w} \quad (2)$$

$$D = \frac{\sum F(x,y)}{h \times w} \quad (3)$$

$$F(x,y) = \begin{cases} 1 & \text{if } f(x,y) \leq M_0 \\ 0 & \text{else} \end{cases} \quad (4)$$

$$M_0 = b * M \quad (5)$$

其中： $f(x,y)$  为灰度图像中像素点  $(x,y)$  的灰度值， $h$  和  $w$  表示图像的高度和宽度， $M$  表示灰度图像的平均灰度值， $D$  表示灰度值小于等于  $M_0$  的像素数量占整幅图像的百分比，它是描述图像明暗程度的重要指标，其物理含义为  $M_0$ 。通常取平均灰度值  $M$  的  $b$  倍，如式 (5) 所示。当满足  $M < T_0$  和  $D > D_0$  时，该图像的明暗程度为偏暗或漆黑，判定为光照异常；否则为光照正常。 $T_0$  和  $D_0$  分别为光照是否异常的判定阈值， $T_0$  取值范围为  $[20, 30]$  的整数， $D_0$  取值范围为  $[0.75, 0.9]$ ， $b$  通常取值为 2 或 3。

3) 采用自适应阈值进行二值化处理：

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) \leq f_0 \\ 255 & \text{else} \end{cases} \quad (6)$$

其中： $f_0$  为二值化阈值。根据图像光照是否异常， $f_0$  将作为自适应的选择，即对于光照正常的图像， $f_0$  采用最大类间方差法确定；对于偏暗或漆黑的光照异常的图像，根据经验确定  $f_0$ ，通常取值  $f_0 = M_0$ 。

图 2 表示采用单一阈值的最大类间方差法进行二值化的示意图，其中图 2 (b)、图 2 (d) 和图 2 (f) 分别是图 2 (a)、图 2 (c) 和图 2 (e) 二值化后的效果，从该示意图看，由于图 2 (c) 和图 2 (e) 光照偏暗，采用该方法进行二值化后，车牌信息均已丢失；图 2 是采用自适应阈值对图 2 (d) 和图 2 (f) 进行二值化处理，其中自适应阈值  $f_0$  取为  $M_0$ ， $b=3$ ， $T_0=20$ ， $D_0=0.8$ ，从效果看，该方法能很好的适应不同明暗程度的车牌图像，原图像中的车牌信息在其二值图像中都较为完整、清晰地得以保留，有利于后续的车牌定位。

### 3 车牌定位

#### 3.1 基于灰度跳变确定车牌上下边界

车牌图像经过二值化处理后，车牌区域变得清晰可见，车牌区域内的字符颜色与背景颜色正好相反，因此，基于灰度跳变法可确定车牌区域的上下边界，算法步骤如下：

步骤 1：用  $i$  表示从二值化车牌图像自底向上开始编号的行号，并令  $i=1$ ；用  $P_T$  和  $P_B$  分别表示车牌上下边界的像数行号， $P_T=0$ ， $P_B=0$ ；

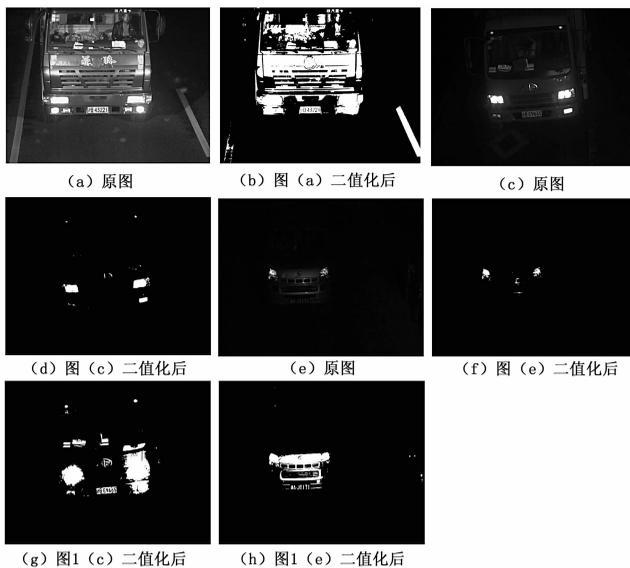


图 2 自适应阈值二值化效果示例

步骤 2：统计第  $i$  行的灰度跳变点总数，即像素灰度值从 0 到 255 或者从 255 到 0 转变的像素点数量，记为  $S[i]$ ，若  $S[i]$  大于等于跳变次数 12，则表示该行为车牌特征行；否则不是车牌特征行；

步骤 3：若  $P_B$  大于 0，则转步骤 4；否则，如果从  $i$  行开始往前连续  $e$  行均为车牌特征行且第  $i-e$  行是非车牌特征行，则  $P_B=i-e-1$  或  $P_B=i-2 * e-1$  并转步骤 4；否则，转步骤 2； $e$  的取值越大则判断误差将越小，但效率会降低， $e$  决定了车牌下边界与车牌中字符的下边界的距离， $e$  值取得越大，则车牌的下边界与字符之间的空隙将越大，反之越小，通常情况下取值  $e \geq 3$ ，在本实施例中， $e=3$ ； $P_B=i-e-1$  时，表示直接取第一个车牌特征行作为下边界； $P_B=i-2 * e-1$  时，表示取第一个车牌特征行往下  $e$  行作为下边界，即在车牌字符与下边界之间留下  $e$  行空隙，可根据实际情况进行选择。

步骤 4：若从  $i$  行开始往前连续  $v$  行均是非车牌特征行且第  $i-v$  行是车牌特征行，则  $P_T=i$  或  $P_T=i-v$  并转步骤 5； $v$  的取值越大则判断误差将越小，但效率会降低， $v$  决定了车牌的上边界与车牌中字符的上边界的距离， $v$  值取得越大，则车牌的上边界与字符之间的空隙将越大，反之越小，通常情况下取值  $v \geq 3$ ，在本实施例中， $v=3$ ； $P_T=i-v$  时，表示直接取最后一个车牌特征行作为上边界； $P_T=i$  时，表示取最后一个车牌特征行往上  $v$  行作为上边界，即在车牌字符与上边界之间留下  $v$  行空隙，可根据实际情况进行选择；

步骤 5：结束，此时  $P_T$  和  $P_B$  分别表示车牌上下边界的像素行号。

之所以跳变法可有效定位上下边界，其基本原理在于：车牌固有的特征导致字符边界和背景之间存在灰度跳变，以二值化后的黑白字车牌为例，逐行扫描遇见字符时，像素点的灰度值会从 0 到 255 有跳变，而离开字符时，像素点的值又会从 255 到 0 有跳变。此外，考虑车牌区域包括至少 7 个字符（不算圆点分隔符），每个字符至少存在两次跳变，同时，在车牌字符二值化过程中，特殊字符“1”的某一行或某几行可能存在断裂的情况，因此在扫描到该字符的这些行时将没有跳变，

因此, 跳变次数的临界值  $S_0$  通常保守的取值为 12。

为提高算法的效率, 在逐行扫描时, 通常自下而上进行, 这是因为在一幅车牌图像里, 车牌区域以上的纹理特征较多, 而车牌以下纹理稀少, 此外, 车牌区域通常位于图像的下半部分, 因此自下而上扫描通常能用最少的扫描时间定位到车牌的上下边界。

### 3.2 基于字符间隔模式确定车牌左右边界

根据我国车牌先验知识, 字符间隔存在一定的固有模式, 为此, 以白字黑底的二值化车牌图像为例说明基于字符间隔模式确定左右边界的算法步骤如下:

步骤 1: 标记特征列。自左向右逐列扫描由上下边界  $P_U$  和  $P_L$  所形成的车牌区域, 采用数组  $C []$  记录扫描的列情况, 当第  $j$  列出现白色像素点时, 将该列标记为特征列并记  $C [k] = j, k=1, 2, \dots, m, m$  即为扫描结束后特征列的总数;

步骤 2: 标记候选车牌字符。令车牌高度为  $Height = |P_T - P_B|$ , 扫描数组  $C []$ , 对每组连续相邻的  $N (0.1 * Height \leq N \leq 0.5 * Height)$  列特征列, 为不失一般性, 假设这  $N$  列特征列的列号为  $C [k], C [k+1], \dots, C [k+N-1]$ , 则这组连续相邻的  $N$  列标记为候选字符列, 这  $N$  列与上下边界  $P_T$  和  $P_B$  所形成的区域即为候选车牌字符。令  $LL [r] = C [k]$ , 它表示这组候选字符列的最左边的列号, 也即候选字符的左边界,  $LR [r] = C [k+N-1]$ , 它表示这组候选字符列的最右边的列号, 也即候选字符的右边界, 其中  $r=1, 2, \dots, Q, Q$  表示候选字符的总数, 它是在扫描数组  $C []$  结束之后确定的;



图 3 车牌字符间隔调整

步骤 3: 调整特殊字符“1”的投影宽度并去除“.”的投影数据。如图 3 (a) 所示的车牌 1 和车牌 2 中的 8 个字符间隔可分别用序列  $S_1 S_1 S_1 S_2 S_2 S_1 S_1$  和  $S_1 S_1 S_1 S_1 S_1 S_1 S_2$  来表示, 然而, 由于车牌字符中“1”和“.”的投影宽度要比其他字符小, 同时, 由于“1”出现的位置具有随机性, 导致字符之间的间隔不统一, 因此  $S_2$  在该序列中出现的位置同样具有随机性, 为了采用统一的模式序列去表达车牌字符间隔, 须对“1”进行调整以使其宽度与其他字符一致; 另一方面, “.”经过二值化处理后存在丢失的可能性, 因此, 对“.”进行去除处理, 从而使得字符之间的间隔序列可以采用固定统一的模式序列去表达, 如图 3 (b) 所示, 经过调整后车牌 1 和车牌 2 中的 7 个字符可采用统一的序列模式进行表示:  $S_1 S_2 S_1 S_1 S_1 S_1$ 。

同步遍历候选字符列的左右边界数组, 即  $LR []$  和  $LL$

$[],$  如果满  $LR [r] - LL [r] < 0.3 * Height$ , 即候选字符的宽度小于  $0.3 * Height$  个像素, 则该候选字符为“1”或“.”, 为了进一步区分是“1”或“.”, 令  $Mid$  表示候选字符中间列的列号, 即  $Mid = (LL [r] + LR [r]) / 2$ , 统计第  $Mid$  列的上下边界  $P_T$  和  $P_B$  之间的白色像素的总数  $S_{Mid}$ , 如果  $S_{Mid} \geq 0.5 * Height$ , 则表示  $Mid$  列为特殊候选字符“1”的中间列, 此时调整该字符的宽度也就是调整其左右边界, 令  $\Delta = (0.5 * (P_T - P_B) - (LR [r] - LL [r])) / 2$ , 则  $LL [r] = LL [r] - \Delta$  以及  $LR [r] = LR [r] + \Delta$ ; 否则, 说明  $Mid$  列为“.”的中间列, 此时去除该字符的投影数据, 即  $LL [r..Q-1] = LL [r+1..Q]$  及  $LR [r..Q-1] = LR [r+1..Q]$ , 详细调整算法如图 4 所示。

```
void AdjustProjection()
for(r=1; r<=Q; r++){
    if((LR[r]-LL[r])<8){
        Mid=(LR[r]+LL[r])/2;
        //计算上下边界 PT 和 PB 之间的 Mid 列
        //的白色像素总数
        Smid=TotalPixels(Mid, PT, PB);
        if(Smid>=0.5*Height){//字符“1”
            LR[r]=LR[r]+Δ;
            LL[r]=LL[r]-Δ;
        }else{//字符“.”
            LR[r.Q-1]=LR[r+1.Q];
            LL[r.Q-1]=LL[r+1.Q];
            Q--;
        }
    }
}
```

图 4 调整特殊字符“1”和“.”的投影宽度算法

步骤 4: 去除“.”的投影后, 车牌字符数应该为 7 个, 因此, 存在  $Q \geq 7$ , 为了定位整个车牌的左右边界, 令  $D []$  表示实际的字符间隔序列模式, 则  $D [s] = LL [s+1] - LR [s], s=1, 2, \dots, Q-1$ , 寻找  $t (1 \leq t \leq Q-6)$  以满足下面两个公式:

$$D_0 = \sum_{k=t, k \neq t+1}^{t+5} D[k] / 5 \quad (7)$$

$$\frac{|D[k] - D_0|}{D_0} \leq \delta, \quad k = t, t+2, t+3, t+4, t+5 \quad (8)$$

其中:  $\delta$  为事先选定的允许误差, 取  $\delta \in [0.1, 0.2]$ , 则车牌的左右边界分别为  $LL [t]$  和  $LR [t+6]$ , 记为  $P_L = LL [t], P_R = LR [t+6]$ 。

至此, 车牌定位完成, 其上下的像素边界行号分别为  $P_T, P_B$ , 左右边界的像素边界列号分别为  $P_L, P_R$ 。

## 4 实验结果与分析

针对不同光照明暗的情况, 选取了 500 张车牌图像进行定位实验, 如图 5 所示。结果表明, 定位成功率在 95% 以上, 尤其是光照偏低的情况仍然能达到这一比例, 符合国家关于车牌识别率的要求。

```
void AdjustProjection()
for(r=1; r<=Q; r++){
    if((LR[r]-LL[r])<8){
```