

# 基于DM642的纹理检测与Adaboost分类器相结合的车牌定位<sup>①</sup>

陈存弟, 刘金清, 刘 引, 蔡淑宽, 何世强, 周晓童, 邓淑敏, 吴庆祥

(福建师范大学 光电与信息工程学院 医学光电科学与技术教育部重点实验室, 福州 350007)

通讯作者: 刘金清, E-mail: [jqliu8208@fjnu.edu.cn](mailto:jqliu8208@fjnu.edu.cn)

**摘 要:** 智能交通系统(Intelligent Transportation System, 简称ITS)是一种实时、准确、高效的综合交通运输管理系统, 而车牌识别(License Plate Recognition LPR)技术是智能交通系统的关键技术之一. 为了在嵌入式系统中实现车牌的实时检测, 研究了基于TMS320DM642的车牌检测、定位与识别技术. 本文提出一种基于纹理检测和Adaboost分类器相结合的车牌定位算法, 针对该算法与DM642的结合进行了系统设计, 并且基于EMCV与Opencv图像处理库进行了代码移植. 该系统很好的避免了单独纹理检测的定位无方向性和单独Adaboost分类的定位不完整性的缺点, 较好的提高了定位准确率, 且定位后的车牌图像能够基于TCP协议发送到PC服务器, 实现远程的车辆车牌的定位和监控.

**关键词:** TMS320DM642; 纹理检测; Adaboost分类; EMCV; Opencv

引用格式: 陈存弟, 刘金清, 刘引, 蔡淑宽, 何世强, 周晓童, 邓淑敏, 吴庆祥. 基于DM642的纹理检测与Adaboost分类器相结合的车牌定位. 计算机系统应用, 2017, 26(7): 56-64. <http://www.c-s-a.org.cn/1003-3254/5853.html>

## Vehicle Plate Location by the Texture Recognition and Adaboost Classifier Based on DM642

CHEN Cun-Di, LIU Jin-Qing, LIU Yin, CAI Shu-Kuan, HE Shi-Qiang, ZHOU Xiao-Tong, DENG Shu-Min, WU Qing-Xiang

(Key Laboratory of OptoElectronic Science and Technology for Medicine of Ministry of Education, College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350007, China)

**Abstract:** The Intelligent transportation system is a real-time, accurate and efficient integrated transportation management system, with the LPR(License Plate Recognition) as one of its key technologies. In order to realize the real-time detection of license plate in the embedded systems, the license plate detection, location and recognition technology based on TMS320DM642 are studied. This paper proposes a kind of license plate location algorithm by combining texture detection with the Adaboost classifier. The system design combining with DM642 is based on EMCV and Opencv image processing library to come true coding porting. The system avoids the non-directional under the condition of only texture detection and the non-integrity under the condition of only Adaboost Classification, it also improves the accuracy of the positioning. In addition, the license plate image which has been located can be sent to the PC server by the TCP protocol. Then the license plate location and monitoring of remote vehicles can be realized.

**Key words:** TMS320DM642; texture detection; Adaboost classifier; EMCV; Opencv

<sup>①</sup> 基金项目: 国家自然科学基金(61179011); 福建教育厅项目(JAS151254); 福建师大项目(I201502019)

收稿时间: 2016-10-30; 收到修改稿时间: 2016-12-05

车牌识别系统分为车牌定位、字符分割和字符识别三个部分,其中车牌定位的准确率直接影响后续的认识.为了提高车牌定位的速度和准确性,国内提出了相关的车牌定位算法,如基于车牌纹理特征<sup>[1-7]</sup>,车牌颜色<sup>[2,4,8]</sup>,车牌固有特征分类<sup>[8,9]</sup>,数学形态学<sup>[10]</sup>等方法.

根据车牌固有的纹理特性,本文提出一种基于纹理检测和Adaboost<sup>[11]</sup>分类器相结合的方法实现车牌定位,其中纹理检测过程中结合图像形态学处理实现车牌的粗定位,而Adaboost分类器是继纹理检测后的多个粗定位区域筛选出真正存在的车牌.最后借助EMCV和Opencv库实现车牌定位算法在DM642的移植,并以DM642平台为客户端,PC为服务器,将车牌定位后的车牌图像以帧方式传向服务器.

## 1 基于纹理检测的车牌

尽管车牌背景复杂多变,但是标准车牌中的7个字符排布具有不变性,绝大部分车牌的图像经边缘提取后都能清晰的看到车牌字符的纹理特性,而且边缘提取后的纹理相对比较密集,字符与字符之间的间隙有规律可寻,依据该特性能够滤除边缘提取后绝大多数的伪车牌区域的白像素,结合形态学处理获取各个粗定位车牌区域的连通域,根据车牌的长宽比等固定特性进一步滤除大部分的伪车牌区域.本文使用的边缘提取算法是基于Canny算子.

如图1所示,纹理检测的具体步骤:

(1) 对灰度化后的车牌图像进行Canny边缘检测.

(2) 去除图像中由白像素连接的横长直线.因为边缘提取后的车牌区域中的字符极少会含有连续的横长直线.本文算法中规定的横长直线定义为连续的15个像素,如果找到则将图像中的整行白像素置0.

(3) 针对行像素,连接有可能的车牌区域的横直线.步骤(2)结束后,图像中剩下的点都是散布的白像素,其中车牌区域的密度最大.这一步,设置整幅图像每两点不相连白像素允许的最大跳变距离 $T_{max}$ ,还设置不相连的多个两点允许的跳变个数为 $H_{min}$ .目的是将在同一行中两点小于 $T_{max}$ 并且跳变数大于 $H_{min}$ 的初始位置连接起来.关于 $T_{max}$ 的设置,由于在一张具有背景的图片中,车牌区域只是镶嵌在图像中的极小部分,而且标准车牌都有7个字符,每个字符都有相应的笔画,这使得跳变间隔允许设置的非常小,在定位过程中,只需根据特定的车牌识别场所,通过先验数据进行统计来调整

$T_{max}$ 即可得到更好的设置值.关于 $H_{min}$ 的设置,车牌经Canny边缘提取后,车牌区域每两个相邻行跳跃点间距小于 $T_{max}$ 的点数至少有12个(考虑到某个字符笔画会有断裂现象),将 $H_{min}$ 设置为8,是为了找出满足跳变点数大于8的区域,即是可能的车牌区域,直到找到大于 $T_{max}$ 的点即记下该终止点.程序实现过程中,设置一个记录数组不断记录每行的跳跃点,记录前都要初始化数组为0.设检测到的两个跳跃间隔为 $T$ ,检测到跳跃的点数为 $H$ ,若满足 $1 \leq T \leq T_{max}$ 和 $H \geq H_{min}$ 则将始末两点之间像素置为白像素,如图1(d).本文采集样本大小为 $2560 \times 1920$ ,样本摄像头与车牌相距为1.5 m-2 m,为了与DSP板配套的摄像头分辨率相吻合,PC上模拟定位前将图像缩小成 $720 \times 540$ ,经实验统计,取值 $T \in [30, 40]$ , $H \in [8, 10]$ 比较合理.本文设置的 $T_{max}=30$ , $H_{min}=8$ .

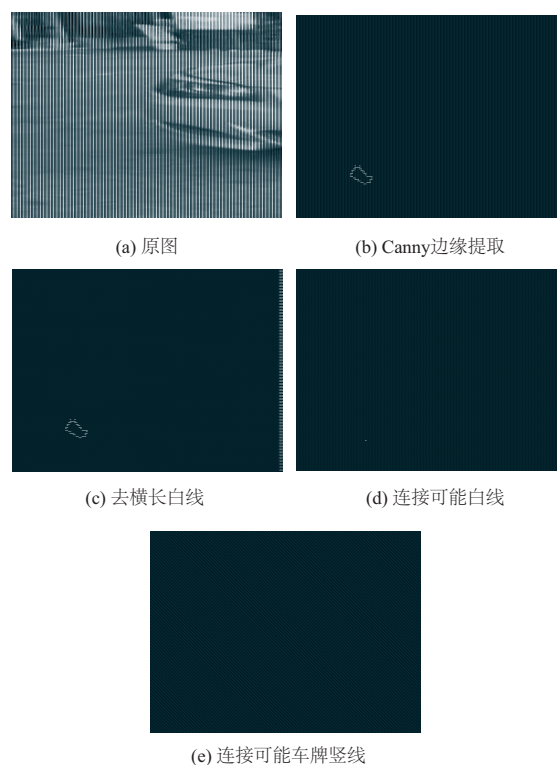


图1 纹理检测图

(4) 针对列像素,连接有可能的车牌区域的竖直线.同样设置每列中两点不相连白像素允许的最大跳变距离 $T_{max}$ ,还设置不相连的多个两点允许的跳变个数为 $H_{min}$ .目的是将在同一列中两点小于 $T_{max}$ 并且跳变数大于 $H_{min}$ 的初始位置连接起来,还消除掉单独的长横直

线. 继步骤(3)后, 车牌区域的白色线条已经非常密集, 上下线条间距基本为1个像素, 设置 $T_{max}=10$ ,  $H_{min}=2$ 已经完全满足上下连线要求, 如图1(e).

(5) 对预留下的区域进行形态学处理. 对区域进行适度的膨胀处理, 并消除长宽比小于3的区域(一般车牌长宽比大于3)和极小与极大的连通区域.

(6) 将最终连通域对应的灰度图像保留下来, 供Adaboost“过滤”, 因为纹理检测的无方向性, 保留下来的区域还不能完全确定为车牌区域.

## 2 Adaboost分类器滤除伪车牌区域

Adaboost算法基本思想是针对同一个训练集训练不同的分类器, 然后把把这些弱分类器(类似于决策树)集合起来构成强分类器. 车牌定位过程中使用Adaboost算法挑选出一些能代表车牌的矩形特征, 按照加权投票的方式将弱分类器构造为一个强分类器, 再将训练得到的若干强分类器串联组成一个级联结构的分类器, 能有效地提高分类器分类的速度.

### 2.1 Haar-like特征提取<sup>[12,13]</sup>

如图2所示, 车牌提取的特征是用Haar-like特征, 分为三类: 边缘特征、线性特征、中心特征和对角特征, 组合成特征模板. 特征模板内有白色和黑色两种矩形, 设置模板的特征值为白色矩形像素和与黑色矩形像素和之差. Haar-like特征反映车牌图像灰度变化情况. 如车牌字符与车牌背景, 车牌背景与车牌边框等. 由于车牌中存在大量的水平、垂直和对角结构, 而矩形特征对边缘和线段较为敏感, 故能使用于车牌的特征提取.

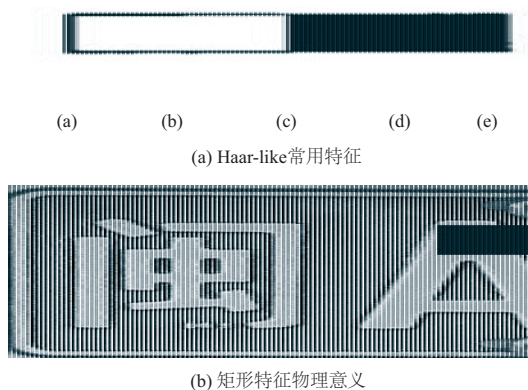


图2 Haar-like特征

### 2.2 Haar-like特征的计算-积分图<sup>[12,13]</sup>

矩形特征的计算量非常大, 特别在图像较大时, 实时性要求较高的情况下, 直接提取像素点灰度值再计算特征值无法满足实时要求, 故引入积分图. 积分图就是只遍历一次图像就可以求出图像中所有区域像素和的快速算法.

积分图的构造方式是位置 $(i, j)$ 处的值是原图像 $(i, j)$ 左上角方向所有像素的和. 以 $FF(m, n)$ 表示源图像,  $I(i, j)$ 表示积分图.

$$I(i, j) = \sum_{m=0}^i \sum_{n=0}^j F(m, n) \quad (1)$$

积分图构造算法:

(1) 用 $S(i, j)$ 表示行方向的累加和, 初始化 $S(i, -1) = 0$ .

(2) 初始化积分图像 $I(-1, i) = 0$ .

(3) 逐行扫描图像, 递归计算每个像素坐标 $(i, j)$ 中第 $i$ 行和 $S(i, j)$ 和积分图 $I(i, j)$ .

$$S(i, j) = S(i, j-1) + F(i, j) \quad (2)$$

$$I(i, j) = I(i-1, j) + S(i, j) \quad (3)$$

(4) 如果 $i$ 还不等于图像宽度, 则返回(3)继续计算积分图; 否则图像积分图就构造好保存在 $I(i, j)$ 中.

如图3所示, 积分图构造好后, 图像中任何矩阵区域的像素累加和都可以通过简单运算得到.

A	B	
C	D	

图3 积分图

设A, B, C, D区域中右下角的坐标分别为 $a, b, c, d$ . 计算D区域的像素和 $D_{inter}$ :

$$D_{inter} = I(d) + I(a) - I(b) - I(c) \quad (4)$$

### 2.3 训练过程

Adaboost分类器算法按规模和功能来划分, 从小到大分为三类: 弱分类器, 强分类器和级联分类器.

#### 2.3.1 弱分类器<sup>[13]</sup>

最初的弱分类器可能只是一个最基本的Haar-like特征, 计算输入图像的Haar-like特征值和最初的弱分类器的特征值比较, 以此来判断输入图像是不是车牌. 其

准确率只要大于50%就可以, 这样对于每一轮训练过后, 根据本轮分类器对所有样本的分类结果修改每个样本的权重, 使分类错误的权重增加, 并投入下一轮的训练过程.

一个弱分类器由下式表示:

$$h_j(x) = \begin{cases} 1 & \text{if } (p_j f_j(x) < p_j \theta_j) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

其中 $x$ 表示图片,  $f_j$ 表示当前特征,  $h_j(x)$ 表示弱分类器,  $\theta_j$ 表示阈值,  $p_j = \pm 1$ 指明不等式的方向. 式中 $f_j$ 和 $\theta_j$ 决定一个分类器, 训练一个分类器, 就是根据当前的权重分布, 找到一个最优的阈值, 使得这个弱分类器对所有训练样本的分类误差和最低. 一个弱分类器仅仅对应一个特征, 那么每一轮弱分类器的训练过程其实就是选择一个最好的将车牌样本和非车牌分离出来的矩形特征. 训练时对于每一个特征都要确定一个最优阈值, 使得该阈值对所有样本的分类效果最好. 每一轮训练过程都可以得到一个在当前权重分布下分类效果最好的特征, 而该特征所对应的弱分类器就是该轮所选出的最优弱分类器.

训练详细过程<sup>[12,13]</sup>:

(1) 给定一组车牌的正负样本图片 $(f_1, g_1), (f_2, g_2), \dots, (f_n, g_n)$ , 其中 $f_i$ 是待训练的图片,  $g_i$ 指出该图片的真伪性, 1为车牌图像, 0为非车牌图像.

(2) 初始化权重 $\omega$ , 对于非车牌样本 $\omega_{1,i} = \frac{1}{2k}$ , 车牌样本 $\omega_{1,i} = \frac{1}{2l}$ , 其中 $k$ 为非车牌样本数,  $l$ 为车牌样本数, 正负样本数 $n=k+l$ .

(3) 对于给定的分类器数目Num, for  $t=1, 2, \dots, \text{Num}$

① 归一化权重:

$$W_{t,i} = \frac{W_{t,i}}{\sum_{j=1}^n W_{t,j}} \quad (6)$$

② 对每一个特征 $j$ 训练一个分类器 $h_j(x)$ , 根据权重计算误差:

$$\varepsilon_j = \sum_i \omega_i |h_j(X_i) - g_i| \quad (7)$$

③ 选择误差最小的弱分类器 $h_t$ .

④ 更新权重:

$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i} \quad (8)$$

当样本 $f_i$ 被正确分类时,  $e_i=0$ , 否则 $e_i=1$ 且 $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ .

### 2.3.2 训练强分类器<sup>[12,13]</sup>

Adaboost分类器包含许多弱分类器, 按照一定的方式将这些弱分类器串联起来就形成一个强分类器. 经过Num此训练后, 得到Num个最优弱分类器. 将这Num个分类器按下列公式组成一个强分类器.

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{\text{Num}} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{\text{Num}} \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

其中 $\alpha_t = \log \frac{1}{\beta_t}$ .

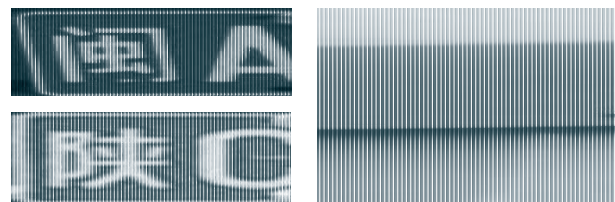
当使用这个强分类器进行检测时, 相当于让该分类器中的多个弱分类器筛选, 再对筛选结果好的弱分类器设置大的权重, 筛选结果差的弱分类器设置较小的权重, 以此来达到好的分类效果.

### 2.3.3 训练级联分类器

多个强分类器级联起来形成一个级联分类器<sup>[14]</sup>, 以达到更高的检测速度和更高的误检率. 每一级分类器都是一个强分类器, 每一个强分类器都能检测出大多数的车牌区域而排除大部分的非车牌区域. 这样在后面的强分类器计算量变少, 因而提高检测速率. 具体训练方法<sup>[12,14]</sup>.

### 2.3.4 训练样本的制作

训练过程中, 需要收集正样本和负样本, 正样本是从含有车牌的图像中剪切出来的子图, 负样本是不包含车牌的任意背景图, 最好选择车辆背景作为样本, 无须归一化和灰度化. 截取2363张正样本, 然后归一化大小为 $50 \times 14$ 的样本并灰度化即可. 训练过程使用opencv库中自带的opencv\_haartraining.exe opencv\_createsample.exe和convert\_cascade.exe文件. 正负样本如图4所示.



(a) 正样本

(b) 负样本

图4 正负样本

结果如图5所示, 本文定位方法综合纹理检测和Adaboost分类器方法, 首先纹理检测能完整包含车牌区域, Adaboost分类器能正确区分出纹理检测后子图的真伪车牌, 起到过滤作用.

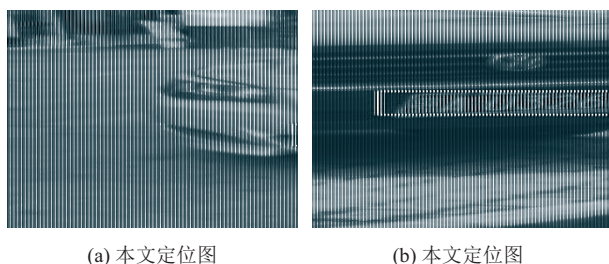


图5 本文算法车牌定位图

### 3 基于DM642<sup>[15]</sup>的车牌检测系统设计实现

#### 3.1 系统构成

车牌定位系统的设计与实现, 本文硬件平台选用的是北京瑞泰公司的ICETEK-DM642开发板, 主要用到开发板外围器件有4路视频端口, 4M\*64\_Bit SDRAM, 存储多达32帧图像, 8-32位Flash, 10-100 M以太网口, 板上JTAG仿真接口. 开发环境是CCS2.2, 编程语言是C/C++语言.

DM642是TI公司C6000系列高性能数字媒体处理器, 时钟频率高达720 MHz, 本文设置时钟频率为600 MHz, 处理能力为4800MIPS. 64个32\_Bit通用寄存器, 8个独立功能单元, 6个算数逻辑单元(ALU 32/40bit), 每个单元支持每时钟周期32bit算术操作, 双16位bit或4个8bit算数操作. 两个乘法器支持每时钟周期4个16x16的乘法<sup>[15]</sup>.

#### 3.2 车牌定位硬件系统的设计

##### 3.2.1 硬件系统框架设计

如图6所示, 本系统硬件以TMS320DM642为核心, 辅以视频解码和视频编码等外设芯片. 图像采集由CCD摄像头采集大小为720\*576的视频信号, 经视频解码器A/D转换为YUV信号, 由EDMA搬运视频数据交予DM642进行相应的车牌定位处理, 处理后得到的定位数据和定位车牌区域图像存在SDRAM中, 最后将定位结果用TCP/IP协议发送到PC服务器, 还传向LCD实现回显功能.

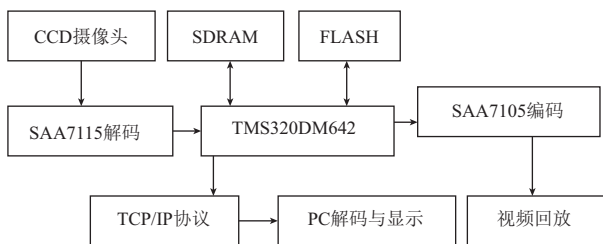


图6 系统硬件框图

##### 3.2.2 基于DSP/BIOS的软件系统框架设计

软件系统框架, 如图7所示.

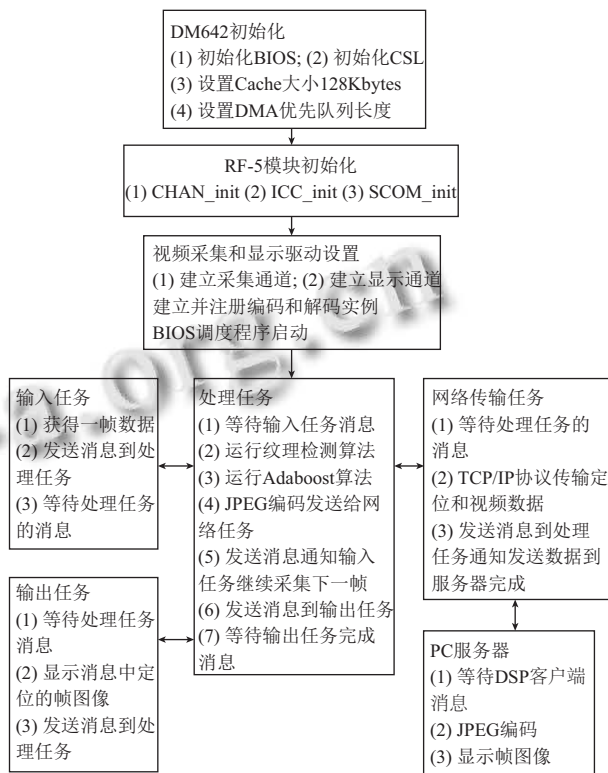


图7 系统软件框图

(1) DSP/BIO: 软件框架的搭建是基于TI公司提供的DSP/BIOS简易实时嵌入式操作系统. 系统基本运行机制类似于UCOS-II嵌入式实时操作系统.

DSP/BIOS可剪裁, 提供诸如任务调度和优先级管理、进程间通信和存储器管理等功能. 具有小型化和模块化的特点使得可以编译连接到应用程序中, 而不是采用独立运行的方式工作, 可以看做是代码库.

(2) Cache分配: DM642片内内存采用两级高速缓存(Cache)结构, 程序和数据拥有各自独立的高速缓存.

片内的第一级程序缓存L1P和数据缓存L1D大小都为16K bytes. CPU与L1P和L1D直接相连<sup>[12]</sup>, 因此这两块Cache都工作在CPU全速访问状态. 程序和数据共享的第二级存储器为L2, 大小为256Kbytes, L2的分段和大小分配可以由用户决定. 本文设置的二级缓存128Kbytes作为缓存, 其余的128Kbytes作为程序和数据的缓存区.

##### (3) RF5结构

RF5是一种DSP软件开发的起步代码参考框架, 它以DSP/BIOS为基础, 主要实现三个功能, 存储管理, 线

程模型和通道封装,共有4个基本数据处理部分:任务(task)、通道(channel)、单元(cell)和XDAIS算法。通常一个任务中可以包括一个或多个通道,每个通道中可以包括一个或多个单元,而每个单元则封装有一个XDAIS算法。单元封装XDAIS算法的作用在于:提供算法与外部世界的一个标准接口,每个单元执行一个简单的ICELL接口,通过该接口执行算法。通道可以执行多个单元,每个任务可以同时处理多个通道。

RF5的线程通信依靠SCOM消息和邮箱(MBX),算法之间的通信基于cell单元。SCOM消息和MBX分别类似于UCOS-II的消息队列和邮箱。SCOM用于任务之间交换信息,有对应的存储空间实现信息在任务间的读取操作。Cell单元同样基于数据缓存区,且存在一个内部cell通信对象,每个cell的输入和输出队列均指向该ICC对象。

如图8所示,车牌定位系统有4个主要任务,每个任务都有静态缓存区,TSK\_process有一个通道,通道内有一个cell单元,存放的是JPEG算法。

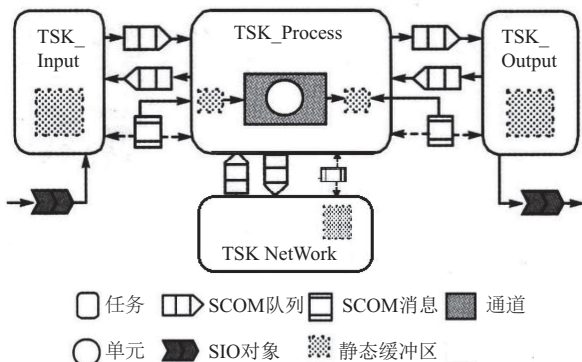


图8 车牌定位系统的RF5视频传输框架

(4) 视频采集和显示驱动能确保摄像头采集过来的图像数据运行良好。通过对输入视频数据的存储管理,输入的视频数据能够由DM642的Vport口解码后通过EDMA存储到SDRAM中,这个过程通过输入视频接口的设备驱动程序完成。在配置输入视频接口的设备驱动程序时,开设3个视频缓存区,设备驱动程序管理这些缓冲区用于实时视频数据的采集。输出视频数据的存储管理也需要3个缓冲区。初始化是将视频口驱动程序准备好,并从视频口驱动程序申请一个输入缓冲区和一个输出缓冲区;进入主循环后,首先通过DMA调用将输入缓冲区的数据读出、处理,再将数据存储到输出缓冲区中,然后向视频口驱动程序切换输

入和输出缓冲区。输入和输出视频数据的存储管理是实际上采用“乒乓”模式。

(5) 一旦BIOS、CSL、DMA、RF5框架结构和视频驱动初始化完成后, BIOS便调度程序启动系统,按程序中线程的优先级运行各个任务。输入任务从SDRAM中读取一帧视频数据,然后通过SCOM传给处理任务,处理任务接受到视频数据后,便开始运行车牌定位算法。由于视频数据传送过来的是YUV图像数据, Y分量表示的是图像的亮度图,即是灰度图,而车牌定位算法中正好是采用灰度图进行信息提取处理。经过车牌定位的纹理检测和Adaboost算法处理后,将定位出来车牌区域图像编码经JPEG编码发送到网络任务,同时将未经过编码的图像发送到输出任务。网络任务主要完成TCP/IP协议的搭建,通过socket套接字实现TCP/IP协议的接口调用。TI推出NDK(Network Develop's Kit)开发套件,目的是提供一个完整的TCP/IP功能环境。基于NDK模块,视频数据得以发送到基于VS平台搭建的PC网络服务器中经JPEG解码后显示。输出任务接收到视频数据后回显车牌区域的帧图像。

### 3.3 系统软件工具

TI公司的DSP集成开发环境CCS2.2提供了基于C语言系统的编译、链接和调试工具,并能进行系统仿真和实时跟踪,为DSP程序的设计和开发提供了极大的方便。

### 3.4 系统代码移植

由于CCS平台调试程序需要通过JTAG将代码下载到DM642后才能查看算法运行的有效性,这无非给调试带来很大的困难,而且会花费很多时间在程序下载上,会大大降低开发速度。本文先在PC环境下,调用计算机视觉Opencv函数库,在Visual Studio 2010下仿真实验。算法的实验效果达到预期的要求后,便开始着手移植到DSP。CCS2.2支持C/C++编程,而Opencv函数库的源代码正是开源的C/C++代码,这无非给开发带来方便。但是Opencv还不能完全适用于特定的DSP开发环境,故有必要对其进行裁剪和优化。EMCV是Opencv分离出来的一个适合TI DM64x系列DSP上运行的计算机视觉库,提供了跟Opencv完全一致的函数接口,移植EMCV能够把Opencv中少量的算法移植到DSP,但是在车牌定位过程中使用到Canny边缘检测算法,图像的缩小放大(Resize)算法和Adaboost算法等,这

些算法在EMCV库中并不带有,因此在移植EMCV后还要在Opencv中裁剪出EMCV中没有的算法.本文使用的是Opencv1.0版本结合EMCV移植到DSP,最后裁剪出来车牌定位真正需要的算法,这样就能大大降低程序的代码量.

在程序移植过程中,由于车牌定位采用的Adaboost分类器文件是.xml格式文件,在CCS中无法读取该文件,因此本文读取分类器的方式改为读取.txt文件.移植后,读取Adaboost分类器使用cvLoadHaarClassifierCascade()函数,检测车牌区域使用cvHaarDetectObjects()函数.

#### 4 系统优化

由于系统实时性要求特别高,因此需要针对DM642的硬件特性对算法和配置进行相应的优化.本文首先将EMCV库和Opencv库文件的C++文件改为C文件,然后主要针对C语言和CCS提供的编译器优化选项进行优化,方法如下:

(1) 浮点转定点运算<sup>[12,16]</sup>: 由于DM642是定点型DSP,将程序中的浮点运算转为定点运算后,程序运行速度大大提高.

(2) EDMA搬移视频数据<sup>[12]</sup>: EMCV可以在CPU不参与的情况下完成数据的复制.

(3) 使用内联函数<sup>[12]</sup>: 频繁的函数调用将导致系统额外开销的增多,将一些功能简单的函数写成内联函数能大大增加系统运行速度.

(4) 使用编译器选项进行优化<sup>[12,17]</sup>: 在编译器中提供了若干等级和种类的自动优化选项.

- O3: 表示可得到的最好程度的优化,编译器将执行各种优化循环的方法,包括软件流水、循环展开

- pm: 联合所有源程序文件在程序级优化代码

本文采用了-O3和-pm优化选项,使车牌定位速度大大提升.

#### 5 算法的比较

(1) 结果如图9所示,单独纹理检测定位,由于车牌背景复杂多变,且易受环境和光照等影响,难免会检测出多个误判的车牌区域,本文将纹理检测方法所带造成的误判称为无方向性,但是优点是检测出的车牌区域中,能够包含整体的车牌,如图9(a),而不会只检测到一部分的车牌区域.带有伪区域张数定义为在车牌定

位过程中的误检区域,如图9(b)单独纹理检测剪切出的伪区域.

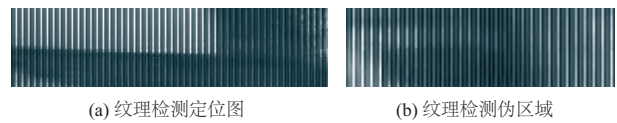


图9 单独纹理检测

(2) 结果如图10所示,如果只单独使用Adaboost分类器方法,就有以下缺点:

① 虽然Adaboost将强分类器级联起来提高运行速度,但这是相对于不将强分类级联起来比较的.一帧720x576的图像数据,Adaboost分类器检测时大部分的时间都花在积分图上,这势必增加检测时间.

② Adaboost分类器虽然能够准确的识别某个区域有车牌,但是有时候无法完全把车牌区域标识出来,所以把这样的定位视为失败,也就是说会导致车牌区域定位不完整,这让接下来的车牌分割与识别顺利完成增加难度.

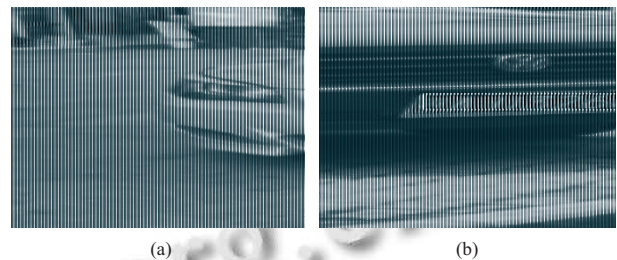


图10 单独Adaboost检测

(3) 纹理检测和Adaboost算法相结合

将纹理检测得到的包含有车牌的小部分区域图像输入到Adaboost算法中进行过滤处理,既能够克服单独Adaboost分类的定位区域不完整性,而且能够避免单独检测的无方向性,大大提高了车牌定位的准确性.

定位过程,纹理检测与Adaboost相结合的方法相对于单独Adaboost方法稍微快,因为单独Adaboost检测需要计算整幅图像的卷积图,而本文算法只是对纹理检测后的子图计算卷积图,并且纹理检测后的子图远远小于原图.纹理检测与Adaboost相结合的方法检测一帧图像需要300 ms左右,而单独使用Adaboost算法定位需要400 ms左右.整个系统运行后,纹理检测与Adaboost相结合的方法1秒能够运行3-4帧,而单独使用Adaboost算法定位1秒能够运行2-3帧.

## 6 DM642上检测结果显示和PC服务器显示结果

车牌在DM642上检测结果在PC服务器上 and LCD显示器上显示,如图11所示。

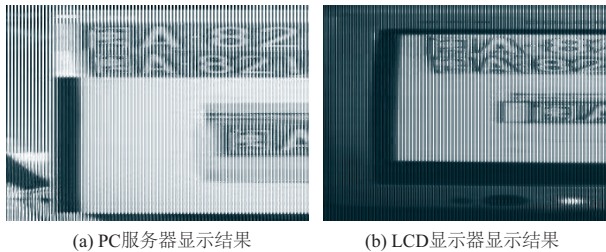


图11 定位显示结果

本文的定位准确率评判标准:不仅能够定位出车牌位置,而且定位出来的矩形区域必须具有完整性.定位的准确率与带有伪区域张数是两个独立的量,因为有些图像能准确定位到车牌却带有一些误检的子图,误检的子图可以在车牌字符分割后得到的字符个数来确定是否是车牌,故本文将带有伪区域但能正确定位到车牌的图像设为正确定位图像.本文在PC机上模拟定位,样本采自福建师范大学校内201张不同场景下的车牌图像,比较本文的定位算法与单独使用纹理检测方法和单独使用Adaboost算法的准确率,结果如表1所示。

表1 实验结果比较

	准确率/%	带有伪区域/张
纹理检测定位	96.5	58
Adaboost定位	87.5	54
本文定位方法	96.5	2

本文算法定位方法中错误定位7张,其中2张是摄像头与车牌间距比较近,导致纹理检测错误,进而Adaboost分类算法无法过滤,5张是车牌定位有断裂现象,如图12(a).剩下的194张中有2张定位正确,但是带有伪区域,是因为Adaboost分类算法误检所致,如图12(b).

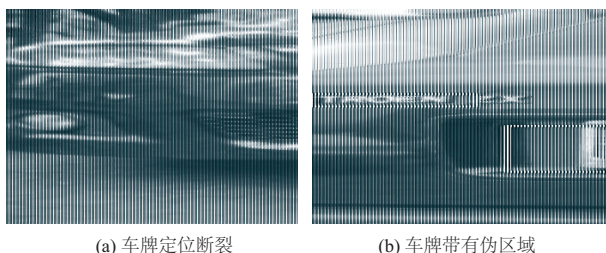


图12 本文定位断裂图(a)和带伪区域图(b)

实验表明,纹理检测和Adaboost相结合的方法很好的避免了纹理检测的无方向性和单独Adaboost分类定位的不完整性,能大大剔除伪区域.单独纹理检测过程,为了保证车牌区域不被滤除,将检测条件设置稍微宽松,主要能保证在检测到的所有子图中存在车牌即可,但这会造就带有的伪区域张数特别多(58张),又由于本文算法是基于纹理检测后用Adaboost进行过滤,故准确率与单独纹理检测的准确率相同,但非常大的优点能把伪区域张数剔除.单独Adaboost定位,虽然能绝大部分定位到车牌区域,但出现车牌不完整现象很严重,导致准确率降到87.5%,而且带有54张伪区域.若将带有伪区域的图片视为定位错误,完全定位到有192张,正确率也能达到95.5%,而单独纹理检测和Adaboost定位正确率大大下降。

## 7 结束语

本文介绍了基于DM642的车牌实时检测系统,虽然本文只实现了车牌定位部分,但是,整个系统框架完全适用于车牌的整个识别系统.定位部分提出一种基于车牌纹理特性检测和Adaboost分类器相结合的车牌定位方法,该方法相对于单独纹理检测和单独Adaboost分类检测,定位准确率大大提高,并且保证了车牌定位的完整性,并且采用EMCV与Opencv图像处理库方法将代码移植在DM642上,使得本定位算法能够实现实时定位的效果.在确保车牌定位准确率的情况下,能为接下来的车牌分割和车牌识别的顺利进行提供了保障。

## 参考文献

- 丁亚男. 基于图像分解的车牌定位算法. 电子科技, 2014, 27(1): 42-44.
- 万燕, 徐勤燕, 黄蒙蒙. 复杂背景中基于纹理和颜色的车牌定位研究. 计算机应用与软件, 2013, 30(10): 259-262, 316. [doi: 10.3969/j.issn.1000-386x.2013.10.089]
- 郭延祥, 陈耀武. 基于边缘检测和颜色纹理直方图的车牌定位方法. 计算机科学与探索, 2014, 8(6): 719-726.
- 孙红, 郭凯. 融合字符纹理特征与RGB颜色特征的车牌定位. 光电工程, 2015, 42(6): 14-19.
- 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.
- 穆长江, 苑玮琦. 基于纹理特征的车牌定位方法. 控制工程, 2004, 11(6): 574-576.
- 潘勇, 赵佳乐, 徐志刚, 等. 基于纹理特征的低照度环境下车



- 牌定位与识别算法. 公路交通科技, 2015, 32(7): 140–148.
- 8 张晓娜, 何仁, 陈士安, 等. 基于主动学习Adaboost算法与颜色特征的车牌定位. 交通运输工程学报, 2013, 13(1): 121–126.
  - 9 曾毅. 遗传算法在车牌定位中的应用. 福建电脑, 2015, (7): 88–90.
  - 10 陈佳. 基于形态学的车牌定位方法的研究与应用[硕士学位论文]. 衡阳: 南华大学, 2015.
  - 11 Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, 55(1): 119–139.
  - 12 聂江浩. 基于DM642的人脸检测系统的设计与实现[硕士学位论文]. 重庆: 重庆大学, 2009.
  - 13 Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Kauai, HI, USA. 2001. I-511–I-518.
  - 14 Lienhart R, Kuranov A, Pisarevsky V. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Lecture Notes in Computer Science*. Berlin, Heidelberg. Springer. 2003. 297–304.
  - 15 TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor (SPRS200J). Texas Instruments, 2005.
  - 16 C Fast RTS Library User Guide (Rev 1.0). Texas Instruments, 2008.
  - 17 TMS320C6000 Optimizing Compiler User's Guide (SPRU187L). Texas Instruments, 2004.