

文章编号:1007-2780(2018)09-0787-06

基于嵌入式车流量实时检测算法研究与实现

马永杰*, 宋晓凤, 李雪燕, 刘姣姣

(西北师范大学 物理与电子工程学院, 甘肃 兰州 730070)

摘要:针对道路现场实时车流量检测问题,提出了一种改进的帧间差分法的运动车辆检测算法,并将该检测算法成功移植到了嵌入式系统上。将帧间差分法与采用长度、宽度、面积筛选轮廓及用质心距离的车辆跟踪算法结合,实现运动车辆的检测;将 U-Boot 引导程序、Linux 内核、Yaffs2 文件系统和检测算法移植到 S3C6410 上,通过摄像头实时采集交通视频,检测结果由触摸屏显示。复杂交通场景的实时测试结果表明,本系统的检测时间为 0.298 秒/帧,准确率超过 88%,基本能够实现在道路现场的车流量实时检测。

关键词:帧差法;车流量检测;S3C6410;实时检测

中图分类号:TP29 **文献标识码:**A **doi:**10.3788/YJYXS20183309.0787

Research and implementation of real-time vehicle flow detection algorithm based on embedded system

MA Yong-jie*, SONG Xiao-feng, LI Xue-yan, LIU Jiao-jiao

(College of Physics and Electronic Engineering, Northwest Normal University,
Lanzhou 730070, China)

Abstract: Aiming at the problem of real-time traffic flow in the road, an improved algorithm for moving vehicle detection based on frame subtraction was proposed, and the algorithm was successfully transplanted into the embedded system. The Frame subtraction was combined with vehicle tracking algorithm using length, width, area selection contour and centroid distance to realize the detection of moving vehicle. The U-boot program, Linux kernel, Yaffs2 file system and detection algorithm were transplanted to the S3C6410, and the traffic video was collected through the camera, and the detection results were displayed by the touch-screen. The real-time test results of complex traffic scene show that the detection time of the system is 0.298 seconds/frame, and the accuracy rate is over 88%, which can basically realize the real-time vehicle flow detection in the road scene.

Key words: frame subtraction; traffic flow detection; S3C6410; real-time detection

收稿日期:2018-03-29;修订日期:2018-06-08.

基金项目:国家自然科学基金(No.41461078)

Supported by National Natural Science Foundation of China(No.41461078)

* 通信联系人, E-mail: myjmyj@nwnu.edu.cn

1 引言

随着城市的发展,道路堵塞问题越来越严重,准确地检测车流量并将其作为智能交通系统的基础数据,可为有效地缓解道路堵塞状况提供帮助。车流量的检测方法有接触式检测、波频检测、视频分析等。接触式检测是在地下铺设感应装置检测所通过的车辆,但这会破坏路面,且施工复杂,维护困难^[1];波频检测利用多普勒效应对回波进行检测,但检测的准确性较差,功能单一^[2];视频分析法利用机器视觉,设备维护方便,成本低,因此成为目前主流的方法之一。视频分析法中常见的运动目标检测方法有帧差法与光流法^[3]。帧差法可分为背景差分法和帧间差分法^[4]。背景差分法利用当前帧减去背景图像,而帧间差分法利用连续的两或三帧图像做差分。光流法利用运动目标与背景之间的光流差异,计算量大,不具有实时性^[5]。

采用视频分析法检测车流量,先将拍摄的视频帧传给计算机,再由计算机进行处理。嵌入式具有高稳定性、低功耗、低成本、可裁剪等特点^[6],完全可代替计算机在现场处理视频。在众多嵌入式处理器中,Advanced RISC Machines(ARM)处理器功耗低,有较强的事务管理功能,但目前在 ARM 上实现用视频分析法实时检测车流量的系统还很少^[7-8]。

本文对传统帧间差分法进行改进,使其能够进行车辆的跟踪与统计,将改进算法移植到 ARM11 上并经过道路现场的测试,说明本文算法和系统可以很好地实现车流量的实时检测。

2 算法原理

背景差分法的背景可人为选取或用算法构建^[9],前者不能保障在白天任何时间段内都高效检测,后者计算量大,会降低系统的检测速度;因此,本文在传统的帧间差分法后增加追踪算法,以弥补帧间差分法的不足,准确地检测车辆,但同时保持帧间差分法计算量小的特点,使算法移植到嵌入式系统上能够实时检测。

2.1 帧间差分法

多数设备采集到的道路图像均为彩色,但处

理彩色图像的计算量较大,而灰度图像较三维彩色图像简化了信息量,并且占用了较少的空间,故用加权平均法^[10]对图像灰度化:

$$f(i, j) = \alpha R(i, j) + \beta G(i, j) + \gamma B(i, j), \quad (1)$$

其中,红色像素 $R(i, j)$, 绿色像素 $G(i, j)$, 蓝色像素 $B(i, j)$ 的加权平均系数以取 $\alpha = 0.299, \beta = 0.587, \gamma = 0.114$, 所得到的灰度化的像素值 $f(i, j)$ 对人眼较为合理。

用高斯滤波可有效地对图像平滑处理并抑制噪声。高斯滤波对任意方向的平滑程度均相同,其保留了大部分所需信号的同时去掉了不必要的高频信号。高斯滤波的平滑程度由方差 σ 决定, σ 越窄,频带越窄。用 x 与 y 表示二维坐标系的横纵坐标,则滤波后的输出图像 $f_{\text{out}}(x, y)$ 与输入图像 $f_{\text{in}}(x, y)$ 满足式(2)的卷积关系^[11]:

$$f_{\text{out}}(x, y) = G(x, y) * f_{\text{in}}(x, y), \quad (2)$$

式中 $*$ 为卷积符号,且

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (3)$$

再根据式(4)将两帧做差:

$$S_n(x, y) = |f_n(x, y) - f_{n-1}(x, y)|, \quad (4)$$

式中,第 n 帧图像序列为 $f_n(x, y)$, 其上一帧图像序列为 $f_{n-1}(x, y)$, 这两帧的差为 $S_n(x, y)$ 。

用同样的操作处理二值化图像和灰度图像,二值化图像的计算量较小。将图像二值化的最简单方法就是设定一个阈值,用图像中的每个像素点与阈值做比较^[12]:

$$d_i = \begin{cases} 1, & s_i > T \\ 0, & s_i \leq T \end{cases}, \quad (5)$$

式中: d_i 是二值化之后的第 i 个像素, S_i 是图像序列 $S_n(x, y)$ 的第 i 个像素, T 为阈值。

帧差法容易出现孔洞^[13],为消除孔洞可借助图像形态学膨胀。膨胀是将图形的一部分与核进行卷积,即计算核所覆盖区域像素点的最大值,并把这个最大值给参考点指定的像素^[14]。膨胀原理^[15]如下:

设一个二维离散的欧几里得空间 ζ , 图像 A 与结构元素 B 都是 ζ 的一个子集, $b \in \zeta$ 是欧几里得空间中的一个点, A_b 由 A 中的元素平移到以 b 为原点的坐标系内所对应的元素组成,定义 $A_b = \{a + b | a \in A\}$, 则膨胀运算可定义为

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \cup A_b. \quad (6)$$

2.2 算法的改进

在二值图像中检测出车辆的轮廓,这些检测出的轮廓虽然经过了膨胀处理,但也不能保证完全填充孔洞,而过度的膨胀会将前面未处理的噪声放大,造成比较严重的误检。为避免一辆车的轮廓由于孔洞而被计成多辆车,可根据轮廓间的距离设定阈值,认为小于阈值的为同一辆车的轮廓,这就需要确定每个轮廓的位置。检测出的轮廓可近似看成长方形,可根据式(7)记下每个轮廓的质心坐标,用质心坐标来表示每个轮廓的位置:

$$\begin{cases} C_i(x) = \frac{R_i(x) + R_i(x) + R_i(w)}{2} \\ C_i(y) = \frac{R_i(y) + R_i(y) + R_i(h)}{2} \end{cases}, \quad (7)$$

式中, $C_i(x)$ 与 $C_i(y)$ 分别是第 i 个轮廓质心的横纵坐标值, $R_i(x)$ 与 $R_i(y)$ 分别是第 i 个轮廓左下角的横纵坐标值, $R_i(w)$ 与 $R_i(h)$ 分别是第 i 个轮廓的宽与长。

只要车辆未行驶出摄像头的拍摄范围,同一车辆就会在接下来的几帧中的不同位置多次出现,故会被多次计数,可设定感兴趣区域,此区域宽度为路面宽度,长度设置保证在两帧内多数车辆可驶出。在感兴趣区域内统计检测出的轮廓数,并将其作为通过的车辆数;但这样会把自行车或摩托车的轮廓也当成车辆的轮廓计数,所以,根据图像中车辆的长度与宽度设定阈值 T_h 、 T_w ,让轮廓长宽不符合阈值的车辆不被计数,并估算出车辆在图像中所占面积,将其值作为阈值 T_a ,再将不符合的轮廓剔除。轮廓的保留需满足式(8)的条件:

$$\begin{cases} P_A(x, y) < C_i(x, y) < P_B(x, y) \\ R_i(w) > T_w \\ R_i(h) > T_h \\ R_i(a) > T_a \end{cases}, \quad (8)$$

式中, $P_A(x, y)$ 是感兴趣区域矩形框的左上角顶点坐标, $P_B(x, y)$ 是感兴趣区域矩形框的右下角顶点坐标, $R_i(a)$ 是第 i 个轮廓的面积。

虽设置了感兴趣区域,但仍不能避免同一辆车被多次检测,故需对车辆进行跟踪,可如式(9)比较前后两帧感兴趣区域内车辆轮廓的质心间距,若小于设定值 T_{zw} 与 T_{zh} 则认为是一辆,否

则车辆的计数加1,

$$\begin{cases} |C_n(x) - C_{n+1}(x)| < T_{zw} \\ |C_n(y) - C_{n+1}(y)| < T_{zh} \end{cases}. \quad (9)$$

3 基于 ARM11 的检测系统

3.1 检测系统硬件

检测系统硬件由 OK6410A 开发板与 1 200 万像素的 USB 摄像头组成,系统的硬件框架如图 1 所示。OK6410A 的处理器是基于 ARM11 的 S3C6410 芯片;双倍速率同步动态随机存储器(DDR)与 Nand Flash 作为存储介质;4.3 寸薄膜晶体管液晶显示屏(TFT LCD)显示系统所检测的结果;12 MHz 与 27 MHz 的无源晶振分别为处理器及液晶屏提供时钟频率。检测系统的 ARMCLK 为 800 MHz, HCLK 为 133 MHz, PCLK 为 66 MHz。

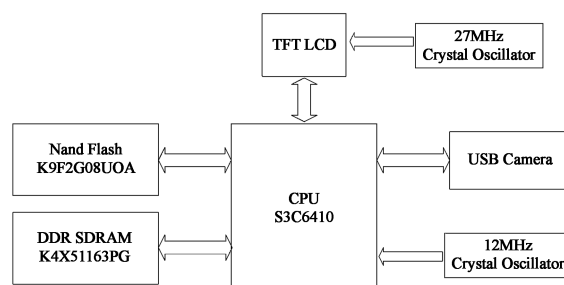


图1 硬件框架

Fig.1 Hardware framework

3.2 检测系统软件

检测系统软件由 Universal Boot Loader(U-Boot)引导程序、Linux 内核、Yaffs2 文件系统、检测算法应用程序组成。嵌入式系统上电后首先运行引导程序,然后调用操作系统内核,最后加载文件系统对文件存储空间进行组织和分配。检测系统软件部分需要在计算机的 Linux 系统上用交叉编译器编译成 ARM 环境下可运行的代码,才可移植到 ARM 处理器上。

U-Boot 主要作用是系统引导,由汇编语言写成的 stage1 和 C 语言写成的 stage2 两部分组成。可在 U-Boot 中设置时钟频率。S3C6410 有 3 个锁相环 APLL、MPLL、EPLL。APLL 生成 ARM 的时钟 ARMCLK。MPLL 生成时钟 HCLKX2、HCLK 和 PCLK, HCLKX2 用于 DDR, HCLK 用

于 AXI 总线与 AHB 总线, FCLK 用于 APB 总线。EPLL 用于特殊用途。APLL 与 MPLL 的输出 F_{out} 与输入 F_{in} 均满足如下关系:

$$F_{out} = \frac{MDIV \times F_{in}}{PDIV \times 2^{SDIV}}, \quad (10)$$

式中 APLL 与 MPLL 的 MDIV、PDIV、SDIV 可通过设置寄存器来确定, F_{in} 为上文提到的外部无源晶振的频率, 即 12 MHz。ARMCLK 超过 533 MHz 时, 必须使用异步模式, 否则 ARM 与计算机串口通讯时, 计算机将显示乱码。同步模式是由 APLL 作为主时钟给 ARMCLK、HCLK、PCLK 提供时钟, 异步模式是由 MPLL 作为主时钟给 HCLK 和 PCLK 提供时钟。HCLK 与分频比 DIV_{HCLKX2} 、 DIV_{HCLK} 有式(11)所示的关系, F_{out} 在同步模式时为 APLL 的输出, 在异步模式时为 MPLL 的输出,

$$F_{HCLK} = \frac{F_{out}}{(DIV_{HCLKX2} + 1)(DIV_{HCLK} + 1)}. \quad (11)$$

内核是一个提供硬件抽象层的系统软件。硬件设备的驱动程序以模块化的形式封装在内核中, 在系统运行时可被装载或卸载。设置 Linux 内核使处理器可支持 Nand Flash、Yaffs2 文件系统、USB 摄像头、触摸屏。

OpenCV 是可以在 Linux 系统上运行的计算机开源视觉库, 但 OpenCV 的一些图形用户接口(GUI)功能无法在 ARM 上实现, 可使用 QT 完成 GUI 的设计。OpenCV 在处理视频时, 需要调用 FFmpeg 的音频/视频编解码库。将 QT、FFmpeg、OpenCV 编译生成的二进制库文件复制到 Yaffs2 文件系统的 lib 中, 就可将文件系统移植到处理器上。

最后, 将检测算法程序与 QT 设计的图型界面程序, 编译成可执行应用程序移植到开发板上。

4 实验结果及分析

在实时检测前用一段道路视频对改进的帧差法与其它算法在 ARM 上进行对比测试。该视频拍摄于下文实时检测所在的天桥上, 在车辆经过天桥下时摄像机会产生抖动, 视频中总共有 32 辆车通过。在读取一帧前与此帧处理结束后添加可返回系统时间的函数, 将这段时间差作为算法的

表 1 检测结果

Tab.1 Detection results

检测方法	误检车 辆数/辆	漏检车 辆数/辆	检测出 车辆数/辆	检测 时间/s
帧差法	88	0	118	0.509
改进帧差法	0	2	30	0.584
Knn 背景建模	3	2	33	0.879
混合高斯背景建模	8	1	39	0.783

执行时间, 检测结果如表 1 所示。

从表 1 中可看出, 帧差法检测速度比较快, 但因摄像机抖动, 环境因素对其影响比较大。本文算法有效地将帧差法检测出的非车辆轮廓滤除掉, 并通过阈值使轮廓因孔洞而断裂的车被检测为一辆, 从而提高了准确度。当视频中有一辆大车通过时, Knn 背景建模与混合高斯背景建模均对其进行了多次检测。Knn 背景建模法排除对大车误检的影响, 其与本文改进帧差法准确度相当。

在道路现场检测是使用摄像头采集视频, 这样避免了从内存中读取视频, 因此, 加快了算法的执行时间。上述 3 种算法实时检测时间如表 2 所示。

表 2 实时检测时间

Tab.2 Real-time detection execution time

本文改进的 帧差法/s	Knn 背景 建模/s	混合高斯 背景建模/s
0.298	0.347	0.337



图 2 道路现场实时检测

Fig.2 Real-time detection of road scene

图 2 所示是现场检测环境与系统的情况, 图 3 是系统正在工作时的检测界面, 也是 TFT 液晶



图3 TFT液晶屏的显示

Fig.3 TFT LCD screen display

屏上的显示情况,其中黑色的矩形框就是感兴趣区域。

道路现场实时检测的结果如表3所示。其中部分检测的显示结果如图4所示。

表3 实时检测数据结果

Tab.3 Real-time detection of data results

实际车辆 数/辆	误检车辆 数/辆	漏检车辆 数/辆	检测出车 辆数/辆	准确率/%
31	2	0	33	93.5
28	1	1	28	92.9
21	0	1	20	95.2
27	2	0	29	92.3
27	3	0	30	88.9



图4 部分检测结果的显示

Fig.4 Partial detection result display

本文准确率 A_{cc} 、实际车辆数 A_{ct} 、漏检车辆数 M_c 、误检车辆数 M 的关系如式(12):

$$A_{cc} = \frac{A_{ct} - M_c - M}{A_{ct}} \times 100\% \quad (12)$$

从表3中可看出,改进帧差法的准确率比较高。误检是由于算法会将大型车辆检测成多辆车,且摄像头的抖动也会引入误检,而当光照使车辆颜色接近路面颜色或车速比较快时,系统会漏检。

5 结 论

本文对传统的帧差法进行改进,使其可检测、跟踪、统计车辆,并移植到 ARM11 上,在道路现场进行实时车流量检测测试,获得了较好的实时性和准确率,算法执行时间为 0.297 秒/帧,准确率可达 88% 以上,表明算法改进和嵌入式实现的有效性。在车速比较快的地方,未来需要进一步提高算法的执行速度和准确率。

参 考 文 献:

- [1] 谢印庆,姜莉,王鲁云,等. 一种快速车流量检测算法[J]. 哈尔滨理工大学学报,2016,21(4):19-24.
XIE Y Q, JIANG L, WANG L Y, et al. A fast vehicle detection algorithm [J]. *Journal of Harbin University of Science and Technology*, 2016, 21(4): 19-24. (in Chinese)
- [2] 孙朝云,阳红,高怀刚. 交通雷达测速仪误差分析与改进[J]. 仪器仪表学报,2003,24(4):418-420,440.
SUN C Y, YANG H, GAO H G. The error analysis and improving method of traffic radar speed gun [J]. *Chinese Journal of Scientific Instrument*, 2003, 24(4): 418-420, 440. (in Chinese)
- [3] 王静静,秦世引. 互补增强式空间运动目标高精度检测与分割[J]. 哈尔滨工业大学学报,2016,48(3):26-32.
WANG J J, QIN S Y. High accuracy detection and segmentation of space moving target by complementary enhancement [J]. *Journal of Harbin Institute of Technology*, 2016, 48(3): 26-32. (in Chinese)
- [4] 闵卫东,郭晓光,韩清. 改进的 ViBe 算法及其在交通视频处理中的应用[J]. 光学精密工程,2017,25(3):806-811.
MIN W D, GUO X G, HAN Q. An improved ViBe algorithm and its application in traffic video processing [J]. *Optics and Precision Engineering*, 2017, 25(3): 806-811. (in Chinese)
- [5] 魏洪涛,李瑾,吴细秀. 融合帧差和 ViBe 的运动目标检测算法[J]. 计算机应用研究,2017,34(5):1565-1567,1571.
WEI H T, LI J, WU X X. Moving object detection algorithm using ViBe combined with frame-difference [J]. *Application Research of Computers*, 2017, 34(5): 1565-1567, 1571. (in Chinese)
- [6] 吴斌,叶东,张鑫,等. 非合作目标姿态测量的嵌入式算法[J]. 光学精密工程,2016,24(11):2804-2813.
WU B, YE D, ZHANG X, et al. Embedded algorithm for relative pose measurement between non-cooperative tar-

- gets [J]. *Optics and Precision Engineering*, 2016, 24(11): 2804-2813. (in Chinese)
- [7] THE P N, HOANG N A N, NGUYEN T A, *et al.* Real time ARM-based traffic Level of Service classification system [C]//Proceedings of the 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. Chiang Mai, Thailand: IEEE, 2016: 1-5.
- [8] 王艺瑶, 严锡君. 基于 S3C6410 目标搜索飞行器的设计与研究[J]. *电子设计工程*, 2015(16): 130-133.
WANG YY, YAN X J. S3C6410-based design and research of target searching vehicle [J]. *Electronic Design Engineering*, 2015(16): 130-133. (in Chinese)
- [9] 于渊博, 张涛, 郭立红, 等. 卫星视频运动目标检测算法[J]. *液晶与显示*, 2017, 32(2): 138-143.
YU Y B, ZHANG T, GUO L H, *et al.* Moving objects detection on satellite video [J]. *Chinese Journal of Liquid Crystals and Displays*, 2017, 32(2): 138-143. (in Chinese)
- [10] KUMARK, LI J P, ZAIN-UL-ABIDIN, *et al.* Content based image retrieval using gray scale weighted average method [J]. *International Journal of Advanced Computer Science and Applications*, 2016, 7(1). DOI: 10.14569/IJACSA.2016.070101.
- [11] 陈超, 罗小华, 陈淑群, 等. 基于现场可编程门阵列的高斯滤波算法优化实现[J]. *浙江大学学报(工学版)*, 2017, 51(5): 969-975.
CHEN C, LUO X H, CHEN S Q, *et al.* Optimizing implementation of Gaussian filter based on field programmable gate array [J]. *Journal of Zhejiang University (Engineering Science)*, 2017, 51(5): 969-975. (in Chinese)
- [12] 徐芳, 刘晶红, 曾冬冬, 等. 基于视觉显著性的无监督海面舰船检测与识别[J]. *光学精密工程*, 2017, 25(5): 1300-1311.
XU F, LIU J H, ZENG D D, *et al.* Detection and identification of unsupervised ships and warships on sea surface based on visual saliency [J]. *Optics and Precision Engineering*, 2017, 25(5): 1300-1311. (in Chinese)
- [13] 屈晶晶, 辛云宏. 连续帧间差分与背景差分相融合的运动目标检测方法[J]. *光子学报*, 2014, 43(7): 0710002.
QU J J, XIN Y H. Combined continuous frame difference with background difference method for moving object detection [J]. *Acta Photonica Sinica*, 2014, 43(7): 0710002. (in Chinese)
- [14] 加里·布拉德斯基, 阿德里安·克伯勒. 学习 OpenCV[M]. 于仕琪, 刘瑞祯, 译. 北京: 清华大学出版社, 2009: 135-143.
BRADSKI G, KAEHLER A. *Learning OpenCV* [M]. YU S Q, LIU R Z, Trans. Beijing: Tsinghua University Press, 2009: 135-143. (in Chinese)
- [15] 杨琨, 曾立波, 王殿成. 数学形态学腐蚀膨胀运算的快速算法[J]. *计算机工程与应用*, 2005, 41(34): 54-56.
YANG K, ZENG L B, WANG D C. A fast arithmetic for the erosion and dilation operations of mathematical morphology [J]. *Computer Engineering and Applications*, 2005, 41(34): 54-56. (in Chinese)

作者简介: 马永杰(1967—), 男, 甘肃灵台人, 博士, 教授, 硕士生导师, 主要从事进化算法、测控技术及应用方面的研究。

E-mail: myjmyj@nwnu.edu.cn