

doi: 10.3969/j.issn.1009-671X.2011.02.14

嵌入式智能家居安防系统的研究与实现

刘海亮, 曹家年, 郭逢丽

(哈尔滨工程大学 信息与通信工程学院, 黑龙江 哈尔滨 150001)

摘 要: 随着生活水平的提高, 智能家居行业已占据越来越重要的位置, 构建一个舒适和谐的家居环境, 越来越引起人们的关注和重视. 本系统以 ARM9 平台为基础, 应用嵌入式 Web、图像采集与处理、MMS 和嵌入式 Linux 等相关技术, 采用 CGI、SQLite 和 Video4Linux 等编程技术, 并结合 ZigBee、GPRS 和 Internet 网络, 提出了一套完整的智能家居安防系统解决方案. 对现今的家居安防系统提出了一些探索性的创新, 为智能安防监控系统提出了新的方法和思路. 该系统功能强大, 性能稳定并具有很好的扩展性.

关键词: 智能家居; 嵌入式 Linux; ZigBee; GPRS; Boa 服务器

中图分类号: TP277 **文献标识码:** A **文章编号:** 1009-671X (2011) 02-0061-06

The design and implementation of embedded smart home security system

LIU Hailiang, CAO Jianian, GUO Fengli

(College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China)

Abstract: With the development of the living standard, the smart home security system has occupied an increasingly important position. People have attached more and more importance to building a harmonious and comfortable home environment. Based on ARM9 platform, the system uses embedded Web, image acquisition and processing, MMS, and embedded Linux technology, and CGI, SQLite and Video4Linux programming techniques, combines ZigBee, GPRS and Internet network, and thus a complete Smart Home Security System scheme is given. The paper presents an innovative idea for present the smart home security system. The system has powerful function, stable performance and good expansibility.

Keywords: smart home; embedded Linux; ZigBee; GPRS; Boa server

智能家居安防系统是未来家居的发展趋势, 其目的是为人们提供一个安全、舒适、高效和便利的生活环境. 智能安防家居已成为国内外研究的热点, 具有广阔的应用价值和市场需求. 传统的智能家居安防系统功能比较单一, 智能程度低下, 不易扩展, 无法进一步构成网络 and 外界进行交互^[1]. 构建了一个基于 ZigBee 和 GPRS 的智能家居系统, 在系统兼容性和智能化程度都有很大提高, 为未来构建家居安防系统提供了新的参考依据.

1 系统结构和功能简介

设计的嵌入式智能家居安防监控系统主要由 3 部分组成: 用户终端 (手机和 PC), 嵌入式 Web 服务

器和 ZigBee 电器终端, 如图 1 所示.

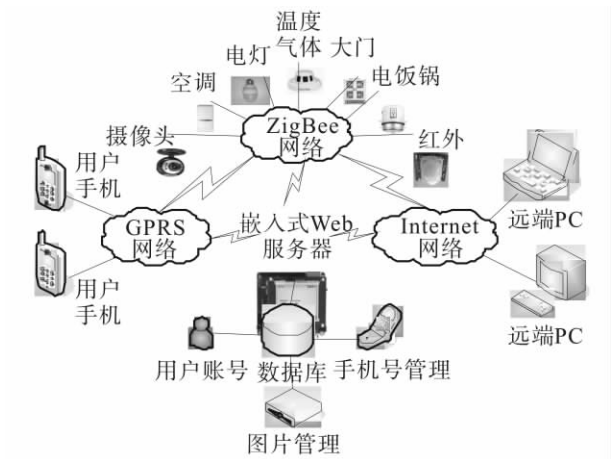


图 1 嵌入式智能家居系统的系统框架
由 ARM9 处理器组成的嵌入式 Web 服务器是整个系统的核心, 是连接 ZigBee、GPRS 无线网络和

Internet 的网关,使得三网之间可以进行无线透明传输,并可省去综合布线的费用和精力.用户在任何时间和任何地点,只需通过联网 PC 打开浏览器登陆服务器,即可对家中进行实时视频监控.系统采用触发式报警方式,当家中有异常情况发生时,监控页面显示异常信息,本地产生声光报警,摄像头采集图像,并为照片打上时间标签,以 MMS(multimedia messaging service) 和 SMS(short message service) 的形式发送到用户手机.用户远程可通过网页或通过手机短信方式直接控制家中电器,本地可使用按键控制.家中电器通过 ZigBee 组成星形网络,接受消息对电器进行控制,并将温度等参数传到嵌入式 Web 服务器上在网页实时显示.系统移植嵌入式数据库 Sqlite 管理和存储图片、用户名密码和手机号等信息,可方便对后期的数据进行检索和查询.且用户只需插入 U 盘或 SD 卡(支持热拔插)即可完成照片和视频的转移,也可将图片和视频通过网络保存到远程 PC 上.

2 系统硬件设计

以三星公司 S3C2440 处理器配合 Linux2.6 内核作为家庭网关的平台,在此基础上利用 ZigBee 模块展开无线家庭传感器网络的设计.本系统的硬件框图如图 2 所示.

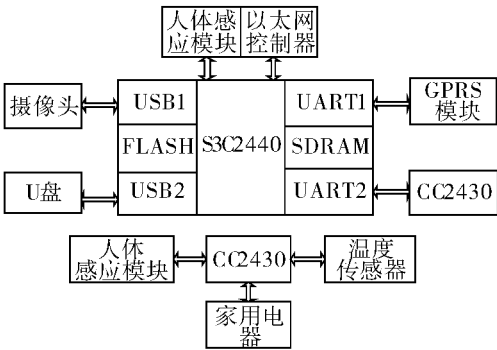


图 2 嵌入式智能家居系统的硬件框图

图中,由 S3C2440 组成的嵌入式 Web 服务器是整个系统的核心;外围存储电路包括 64 M NAND Flash 和 2 片 32M SDRAM;外围接口电路包括 3 个串口、2 个 USB 和网口等;GPRS 模块采用谱泰公司生产的支持彩信协议的 PTW73 模块,GPRS 模块发出报警信息到用户的手机,并且接受用户通过手机

发出的控制信息传送至嵌入式 Web 服务器,最后发送到 ZigBee 网络.

ZigBee 无线收发模块采用 TI 公司生产的 CC2430. CC2430 内部有一个高性能 DSSS 和一颗工业级 8051 控制器,并使用 TI 公司提供的 Z-stack 协议栈,很大程度地减轻了软件开发的工作量. ZigBee 无线网络分为协调器和终端节点,其中带有各种传感器的 ZigBee 终端节点,与嵌入式 web 服务器连接的 ZigBee 协调器构成星形传感器网络,使各个房间都置于监控之中.各终端节点将各项环境参数通过无线发送到 ZigBee 协调器,ZigBee 协调器通过串口传给嵌入式 Web 服务器,并在网页上实时显示.同时 ZigBee 协调器接受来自服务器的控制信息,传送给相应 ZigBee 模块控制各种家电.

摄像头采用内置芯片 ZC301P 的摄像头,ZC301P 芯片采用新一代的 CMOS 感光芯片,内置了 DSP 硬件编解码器,经过 DSP 处理后出来的图像是 JPEG 格式,输出的图像不用再转换,很适合图片的采集以及实时监控的需要.

3 系统软件设计

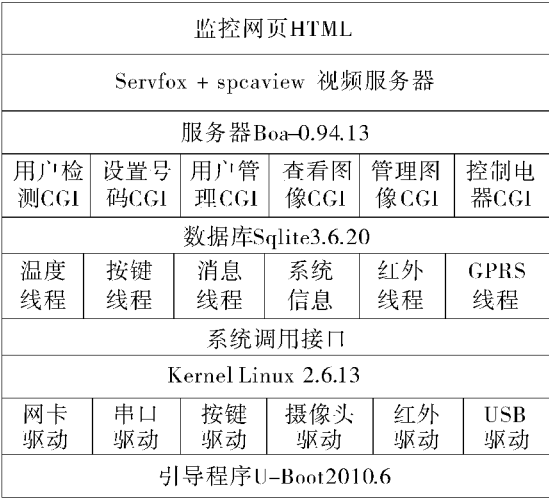


图 3 嵌入式智能家居系统的软件体系结构

本系统的软件框图如图 3 所示.移植最新版本 U-boot2010.6 初始化系统硬件和外围设备;移植 Linux2.6.13 操作系统(新版内核采用 V4L2 编程,而视频采集程序是基于 V4L 编写,故采用此版内核)安排整个系统资源和协调所有程序;应用 Yaffs2

文件系统存储文件和程序,并在此基础上移植和编写网卡、红外和摄像头等驱动;应用程序采用多进程与多线程技术,进程之间使用消息队列与共享内存的方式进行通信;移植嵌入式 Boa 服务器,应用 CGI 编程实现服务器端的相应功能;使用嵌入式数据库 SQLite 管理各种信息;移植采集视频 servfox 程序和流媒体服务进程 spcaview,并用 HTML 编写了本系统的监控页面.

3.1 系统驱动程序的设计与移植

本系统的设备驱动程序主要基于 Linux2. 6. 13 内核对红外、串口、USB、摄像头和网卡等驱动的移植和编写. 本地服务器采用声光报警方式,驱动部分主要分为蜂鸣器和 LED 驱动. 蜂鸣器驱动通过 S3C2440 内部的 PWM 产生一定的频率来控制不同的异常情况产生不同的声音^[3]. 对报警系统的控制通过 ioctl 函数实现,在驱动中分别定义了 BEEP_ON 和 BEEP_OFF 等宏,在应用程序只需在 ioctl 中调用宏就可以实现对报警系统的声光控制.

本地服务器端外接人体热释电红外感应模块,当红外传感器感应到有人时,在相应的引脚上产生一个跳变. 驱动主要处理电平跳变产生的中断信号,在驱动中利用 kill_fasync() 函数产生信号通知应用程序. 摄像头驱动移植 spca5xx 开源程序,移植摄像头驱动后需在 dev/v4l 目录下创建 video0 设备文件,还应注意内核裁剪时选上 Video For Linux、Support for Host-side USB 和 OHCI HCD support 选项. S3C2440 支持 3 个串口,嵌入式 Web 服务器通过串口与 GPRS 和 ZigBee 模块进行通信;而 Linux 内核只支持 2 个串口,其中 1 个串口是针对红外接口,所以需要修改内核源码(关于摄像头移植和串口驱动修改可查看参考文献[5]).

Linux2. 6 驱动中默认的是将芯片的两 USB 口设置为一主一从. 本系统的 2 个 USB 接口一个连接 USB 摄像头,另一个连接 U 盘. 因此需要修改 USB 驱动,将其改为 2 个主口使用. 在 S3C2440 芯片中,用 MISCCR 寄存器中的 SEL_USBPAD 位来设置的 USB 口,置 1 时为 host,置 0 时为 device. 在 ohci-s3c2410. c 文件中的 usb_hcd_s3c2410_probe() 函数中添加如下代码,重新编译内核即可.

```
unsigned long miscrr;
```

```
miscrr = __raw_readl( S3C24xx_MISCCR );
miscrr |= ( 1 < 3 );
__raw_writel( miscrr, S3C24xx_MISCCR );
```

3.2 系统应用程序的设计

采用多线程和多进程技术,将应用程序划分为多个独立的任务,有效地提高了系统的运行效率. 本系统主要有 2 个进程: 系统主进程和 Web 服务器进程,2 个进程通过消息队列、信号和共享内存的方式进行通信和共享资源的访问,其中 web 服务器进程作为守护进程在后台执行. 系统主进程的流程图如图 4 所示.

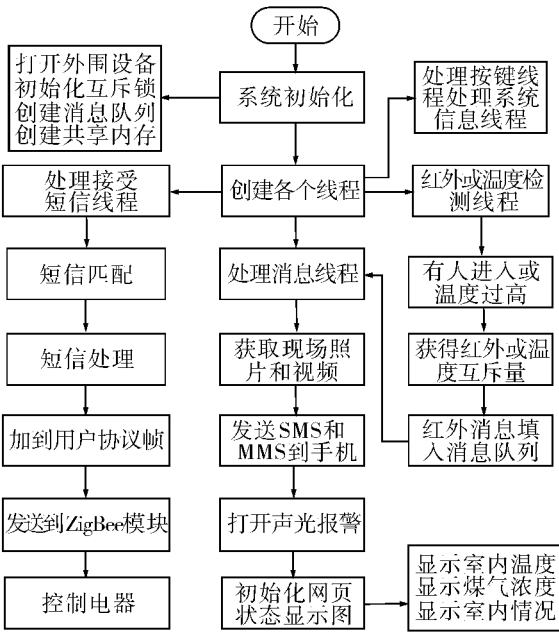


图 4 系统主进程流程

首先进行系统初始化,包括打开蜂鸣器、按键和串口等设备. 由于线程共享进程的资源 and 地址空间,因此对这些资源进行操作时,必须要考虑到各个线程间资源访问的同步与互斥的问题. 文中采用互斥锁机制来控制对共享资源的原子操作,调用 pthread_mutex_init() 函数初始化摄像头、红外和 GPRS 等互斥锁,并调用 msgget() 和 shmget() 函数创建并初始化消息队列和共享内存(关于 LINUX C 编程函数可查阅参考文献[2],这里不再详述).

然后调用 pthread_create() 函数创建各个线程. 其中接受短信线程将短信内容进行处理,并加入到协议帧发到 ZigBee 模块控制家中电器. 温度和红外检测线程通过 read() 函数读取 ZigBee 模块发送过

来的温度值和室内的状况,保存到主进程的全局变量.处理系统信息线程通过共享内存的方式将温度值和室内状况传给 Web 服务器,服务器通过相应 CGI 程序在浏览器上实时显示.如果温度过高或有人进入,则通过 msgsnd() 函数,将相应消息加入消息队列.处理消息线程通过 msgrcv() 函数接收到消息,调用声光报警线程,并完成拍照和发送彩信.

3.3 嵌入式 Web 视频服务器程序的设计

本系统移植了 Boa 服务器,Boa 服务器是一个单任务的小巧而高效的服务器.作为一个单任务 Web 服务器,Boa 依次完成用户的请求,而不会 fork 出新的进程来处理,且 Boa 支持 CGI 编程,能够为 CGI 程序 fork 出一个进程来执行.

CGI 是通用网关接口的简称,其建立了 Web 服务器与 Web 浏览器之间的交互.服务器部分主要是使用 C 语言编写相应 CGI 程序,CGI 进程与主进程并行执行,应用消息队列和共享内存的方式与主进程进行通信.CGI 程序只需通过交叉编译为后缀名是 .cgi 的形式即可.

系统移植 SQLite 数据库管理家居的环境参数,存储用户的账号密码、手机号和图片视频等信息.SQLite 体积小,提供了对 ANSI SQL92 的支持.使用数据库方式,使得数据和应用程序相互独立,通过事务来进行调度和并发控制,有效地对数据进行了存取、查询等共享操作,确保系统具有较好的整体性能^[4].本系统通过调用 CGI 程序访问 SQLite 数据,将环境参数和照片等信息显示到网页上.

视频服务器通过移植 servfox 和 spcaview 服务建立,servfox 程序建立在 Web 服务器之上,使用 Linux 下的 V4L 函数和 MJPEG 图像压缩器完成视频信号获取和压缩.Spcaview 流媒体进程和 TCP 网络传输协议完成了视频图像远程的传输.视频服务器不断监听端口,如有连接请求则建立 TCP 连接,之后 servfox 将不断向端口提供采集的数据流^[5].

其中关于 Boa、SQLite 和 servfox 等的移植方法其他资料讲得已经介绍很清楚,下面介绍单击拍照按钮实现图像的采集与处理的整个流程(如图 5 所示),详细介绍 CGI、SQLite 和图片采集的编程方法,系统其他功能与此流程类似,不再介绍.

首先用户点击“拍摄照片”按钮向服务器发出

HTTP 请求(CGI 请求).Web 服务器收到该请求后创建拍摄照片的 CGI 进程,并将该请求数据变成环境变量,启动 CGI 程序.该部分代码如下所示:

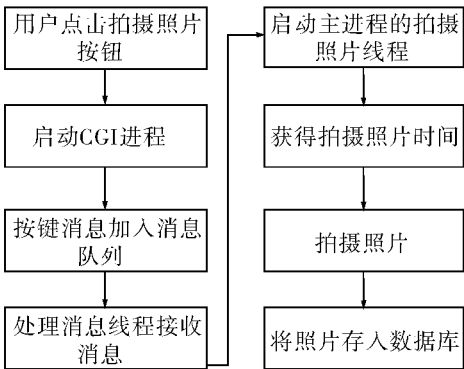


图 5 拍摄照片处理流程

```

fprintf( stdout, "Content - type: text/html; charset
= utf - 8\n\n" );
//打印 HTML 头
fprintf( stdout, "<html> <title> device_test <
title> \n" );
fprintf( stdout, "<body> " );
char * cntl = getenv( "QUERY_STRING" ); //获
得环境变量
mqid = msgget( ftok( MQ_KEY,0 ),0 ); //打开
消息队列
struct Mesg * ptr_mesg;
ptr_mesg = ( struct Mesg * ) malloc( sizeof( struct
Mesg ) );
if( strcmp( cntl, "camera_capture" ) == 0)
ptr_mesg -> mesg_type = MSG_CAMERA_CAP-
TURE;
msgsnd( mqid, ptr_mesg, 1, 0 ); //添加消息
队列
free( ptr_mesg );
fprintf( stdout, "</body> </html> " );
该 CGI 程序完成了对环境变量的处理,并向主
进程发送拍摄照片的消息.下面是主程序中消息处
理线程的部分代码:
struct Mesg * mesg_buff;
mesg _ buff = ( struct Mesg * ) malloc ( sizeof
( struct Mesg ) );
msgrcv ( mqid, mesg _ buff, 1, 0, IPC _

```

```
NOWAIT); //读取消息
long mtype = mesg_buff -> mesg_type;
switch ( mtype) {
case MSG_CAMERA_CAPTURE:
if ( pthread_mutex_trylock( &camera_mutex) =
= 0)
pthread_create( &tid_camera, NULL, camera_cap-
ture_webpage, NULL);
..... //其他消息的处理 }
处理消息线程循环接收 web 服务器发来的消
息,经过判断之后启动拍摄照片线程,进行拍照,最
后将照片信息保存到 SQLite 数据库,便于以后查询
管理. 下面是拍照线程部分代码,首先通过 get_pic-
ture_name() 获得时间标签作为照片名字,该函数如
下所示:
get_picture_name( char * path, char * jpeg-
name, int type) {
time_t t; //获取日历时间
struct tm * timetm;
timetm = gmtime( &t); //化为格林威治时
间
strftime( year, 5, "%Y", timetm);
strftime( month, 5, "%m", timetm);
..... //存为 2010 - 06 - 26 10: 17: 20. jpg
格式
store_to_database( jpegname, year, month, day,
hour, minute, seconds, t); //存入 SQLite 数据库 }
接下来需将图片信息存入 SQLite 数据库,首先
调用 sqlite3_open() 打开数据库,然后调用 sqlite3_
exec() 函数执行 SQL 语句,最后使用 sqlite3_close
() 函数关闭该数据库.
int ret;
char c [] = “;”;
char end [] = “)”;
char sql [200] = “insert into pictures values
( NULL, ”;
strcat( sql, path);
strcat( sql, c);
strcat( sql, year);
strcat( sql, end);
```

```
..... //根据 SQL 语句格式填写完整
int rc = sqlite3_open( PATH, &db); //下有出
错处理
rc = sqlite3_exec( db, sql, 0, 0, &ErrMsg);
sqlite3_close( db);
本系统图像采集程序使用 Video4Linux 编程,
V4L 是 Linux 对图像传感器之类的图像设备支持文
件,V4L 包含了一系列的 API 和图像语音驱动. 首先
使用 init_videoIn() 函数初始化 vdIn 结构,然后通过
调用 v4lGrab() 函数就可以拍摄照片了.
void take_one_picture( char * path) {
struct vdIn videoIn;
char videodevice [] = “/dev/video0”;
int format = VIDEO_PALETTE_JPEG;
int width = 352;
int height = 288;
memset( &videoIn, 0, sizeof( struct vdIn) );
if ( init_videoIn( &videoIn, videodevice, width,
height, forma, grabmethod) == 0)
v4lGrab( &videoIn, path); //拍摄照片}
发送中文短信需通过访问 SQLite 数据库查出要
发送的手机号,然后通过 AT 指令发送. 文中使用的
GPRS 模块支持 MMS 协议,只需将 JPEG 格式照片
通过处理,通过 AT 指令即可完成彩信的发送.
3.4 ZigBee 协调器及其终端程序的设计
ZigBee 的家庭网络需要采集家居信息,实时的
进行监控,并将这些信息传送给 web 服务器上. 为
了有效地管理这些数据,需要执行统一用户协议. 下
面介绍该系统使用的通用数据格式,如图 6 所示.
帧头 地址 操作类 二值 整形 累加 帧尾
标志 信息 型码 参数 参数 和 标志
帧头 数据 帧尾
图 6 用户协议帧格式
用户协议帧由帧头、数据和帧尾组成. 其中帧头
由 0xAA 表示. 地址信息是所访问模块的短地址信
息. 操作类型码使用十六进制数分别表示设备开关
操作、参数查询操作、调节控制操作和设备配置操
作. 二值参数表示设备的状态. 整形参数表示要发
送的数据,根据不同操作设置不同的含义. 累加和用校
```

验数据的正确性. 帧尾标志为 0xFF^[6].

本系统选用了 TI 公司的 Z-Stack 协议栈. Z-stack 符合 ZigBee 2006 规范标准, 是 ZigBee 协议的具体实现. Z-Stack 运行在 OSAL (operation system abstraction layer) 的操作系统上, 本部分的主要任务是对 OSAL 的应用层进行开发, 主要分为 2 部分: 协调器和终端设备的应用开发. 协调器开发主要包括发起网络, 分配网络地址, 实现绑定, 接受/发送信息等事件. 终端设备开发主要包括加入网络, 接受控制命令和送环境参数给协调器等. 系统融合了数据采集和串口通信等功能. 因此以 Z-stack 中 SimpleApp 为模板进行无线传输, 加入 SerialApp 程序, 添加新的任务用于串口通信.

4 结束语

智能家居安防监控系统是一个复杂的系统, 涉及到通信、计算机、自动化、建筑等多个领域. 本系统具有无需布线, 多种报警方式, 多种控制家电方式,

系统稳定可靠等特点, 为智能家居系统提出了新的方法和思路. 但本系统对产品化还有一定的距离, 对硬件的扩展和软件的设计还需进一步的改善, 从而实现整个系统的高度集成化和智能化.

参考文献:

- [1] 龙明桂, 陈书开. 基于 GPRS 的远程安防监控系统 [J]. 微计算机信息, 2008, 24(12/1/2): 46-48.
- [2] 陈健, 宋健健. LINUX 程序设计 [M]. 3 版. 北京: 人民邮电出版社, 2007: 16-28.
- [3] 韦山东. 嵌入式 Linux 应用开发完全手册 [M]. 北京: 人民邮电出版社, 2009: 40-58.
- [4] 刘琼, 马海波. SQLite 在嵌入式 Web 服务器中的应用研究 [J]. 微计算机信息, 2010, 26(3): 44-46.
- [5] 龚尧非, 金静. 嵌入式 Linux 下的视频服务器的设计 [J]. 电子测量技术, 2010, 33(4): 72-74.
- [6] 韩双双. 基于 ZigBee 无线网络的智能家居系统关键技术研究 [D]. 长春: 吉林大学通信工程学院, 2009: 35-36.

(上接第 60 页)

参考文献:

- [1] SPAGNOLO P, ORAZIO T D, LEO M, et al. Moving object segmentation by background subtraction and temporal analysis [J]. Image and Vision Computing, 2006, 24: 411-423.
- [2] YUE Hao, SHAO Chunfu, ZHAO Yi, et al. Study on moving pedestrian tracking based on video sequences [J]. J Transpn Sys Eng & IT, 2007, 7(4): 47-51.
- [3] CARMONA E J, MARTINEZ-CANTOS J, MIRA J. A new video segmentation method of moving objects based on blob-level knowledge [J]. Pattern Recognition Letters, 2008, 29: 272-285.
- [4] ZHOU D, ZHANG H. Modified GMM background modeling and optical flow for detection of moving objects [J]. 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005, 3(10/11/12): 2224-2229.
- [5] WEI Zhiqiang, JI Xiaopeng, WANG Peng. Real-time moving object detection for video monitoring systems [J]. Journal of Systems Engineering and Electronics, 2006, 17(4): 731-736.
- [6] KIM J B, KIM H J. Efficient region-based motion segmentation for a video monitoring system [J]. Pattern Recognition Letters, 2003, 24: 113-128.
- [7] FORESTI G L, MICHELONI C, PICIARELLI C. Detecting moving people in video streams [J]. Pattern Recognition Letters, 2005, 26: 2232-2243.
- [8] CELIK T, DEMIREL H, OZKARAMANLI H, et al. Fire detection using statistical color model in video sequences [J]. Journal of Vision Communication & Image Representation, 2007, 18(2): 1-9.
- [9] 陈韩锋, 戚飞虎. 视频对象分割中基于 Gibbs 随机场模型的时空分割结合方法 [J]. 电子学报, 2004, 32(1): 34-37.
- [10] BICEGO M, CRISTANI M, MURINO V. Unsupervised scene analysis: a hidden Markov model approach [J]. Computer Vision and Image Understanding, 2006, 102(1): 22-41.
- [11] ZHANG Wei, FANG Xiangzhong, YANG Xiaokang. Moving vehicles segmentation based on Bayesian framework for Gaussian motion model [J]. Pattern Recognition Letters, 2006, 27: 956-967.