

面向仓库监控的改进式动态物体检测系统的设计

陈婧,李光炆,陈鲤文

(福建工程学院 信息科学与工程学院,福建 福州,350118)

摘要: 采用混合高斯模型算法,以实现背景前景的分离,并利用前景目标的二值化图像计算其轮廓特性,对超过阈值的目标物体进行预警,同时系统为实现良好的可移植性及较快的硬件处理速率,采用软件分层的体系来实现动态物体检测,底层通过 V4L2 接口实现硬件视频帧的采集,顶层调用 opencv 视觉库实现混合高斯模型下的前景提取。最后系统实现良好的人机交互界面的设计。

关键词: opencv; 动态物体; 混合高斯模型

中图分类号: TP277

文献标志码: A

文章编号: 1672-4348(2018)04-0336-05

Design of an improved dynamic object detection system for warehouse monitoring

CHEN Jing, LI Guangyang, CHEN Liwen

(School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China)

Abstract: The Gaussian Mixture Model algorithm was used to separate the background and the foreground. The contour features were calculated by using the binary image of the foreground target, and the target object that exceeded the threshold value was forewarned. Meanwhile, in order to achieve good portability and fast processing rate of the hardware, the system adopted a layered software system to realize dynamic object detection. The bottom layer realized the collection of hardware video frames through the V4L2 interface, and the top layer achieved the foreground extraction under the Gaussian Mixture Model by using the opencv visual library. Finally, the system realized the design of a good human-machine interaction interface.

Keywords: opencv; dynamic objects; Gaussian Mixture Model

随着视频监控技术的发展,从最初需要人力监控完成到现在可使用智能化监控系统进行自动监测。智能视频监控的优点在于它几乎不需要人为干预的情况下即可通过分析摄像头采集的图像,对静态背景下动态物体进行识别,并能够处理异常发生的情况^[1]。视频监控系统可应用于普通场合上的仓库管理中。

对于图像序列或者视频流,目前移动物体检测算法主要有利用帧间关系进行目标检测和利用单帧进行统计分析进行目标分类的两大类^[2]。

传统的视频监控系统中,对于监测动态物体的主要算法有:帧差法、光流法、平均背景法、背景建模等;固定背景下的方法常见的有相减法 and 建模法^[3]。Horn 和 Schunck^[4]在 1981 年率先提出物体运动光流的计算方法,将物体的二维运动速度和灰度结合,通过光流的变化来检测运动物体的运动速率,但其拥有明显的缺点就是光流法必须假设检测物体的亮度条件不变^[2,4-6],在仓库管理等自动监控场合中,很难做到此点,因此仅利用光流法将无法进行实时运算。而背景建模法能够很

收稿日期: 2018-07-12

基金项目:福建工程学院青年基金项目(GY-Z13010);泉州市科技局重大项目(2017G012);福建省科技厅项目(2018H0003)

第一作者简介:陈婧(1983-),女,福建福州人,讲师,硕士研究生,研究方向:模式识别与智能系统。

好的避免以上缺点,其可应用在无需提前截取背景帧以及背景光照条件变化的场合,其主要有单高斯模型建模等^[7]。硬件方面,普通的动态监控系统大都直接调用 opencv 视觉库进行运算,但难以适用嵌入式监控的环境^[1,8-9],鉴于以上的问题,在普通仓库管理应用前提下,系统对比传统动态物体检测中帧差法等算法,采用高斯混合算法提取前景目标,再利用前景二值化图像计算其轮廓特征,随后为避免因噪声等因素产生的误差,系统计算其轮廓面积和周长,排除周长及面积较小的噪声点,对于超过一定阈值的动态物体才进行预警。

1 动态物体检测算法

1.1 帧差法

目前流行的帧差法,主要由背景差分法与帧间差分法两种,帧间差分法是一种通过对视频图像序列中相邻两帧作差分运算来获得运动目标轮廓的方法,而背景差分法通常选取视频信号中的第一帧做为背景帧,而后在视频序列中间隔固定的时间选取视频帧与背景帧进行差分比较,最后得到运动目标的轮廓。

1.2 背景建模算法

背景建模的主要思想是利用概率学原理,统计视频帧中像素分布的不同从而辨别出前景和背景,其中高斯混合模型算法是对样本的概率密度分布进行估计,而估计采用的模型是几个高斯模型的加权和。每个模型就代表了一个类,然后利用样本数据对模型进行投影,从而得到在各个类上的概率。然后选取概率最大的类为判决结果。混合高斯模型的定义为:

$$p(x) = \sum_{i=1}^K \pi_k p(x|k) \quad (1)$$

其中 K 为模型的个数; π_k 为第 k 个高斯的权重; $p(x|k)$ 则为第 k 个高斯概率密度,若均值为 u_k , 方差为 σ_k 。那么估计概率密度的值就是要求出 π_k , u_k 和 σ_k 各个变量。最后 $p(x)$ 的表达式中,求和式的各项的结果就分别代表样本 x 属于各个类的概率。其中在做参数估计的时候,采用的是最大似然方法,使样本点在估计的概率密度函数上的概率值最大。为避免浮点数下溢,通常将概率值取 \log , 即将目标改写成:

$$\max \sum_{i=1}^N \log p(x_i) \quad (2)$$

也就是最大化对数似然函数,完整形式为:

$$\max \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k N(x_i | \mu_k, \sigma_k) \right) \quad (3)$$

主要的步骤可以分为 2 步:其中第 1 步,假设已知各个高斯模型的参数,然后估计每个高斯模型的权值;第 2 步,基于估计的权值,再次去确定高斯模型的参数。重复这 2 个步骤,直到波动很小,近似达到极值。具体表达如下:

(1) 第 1 步

假设高斯模型的参数和是已知的,对于第 i 个样本 x_i 来说,它由第 k 个 model 生成的概率为:

$$\bar{\omega}_i(k) = \frac{\pi_k N(x_i | \mu_k, \sigma_k)}{\sum_{j=1}^K \pi_j N(x_i | \mu_k, \sigma_j)} \quad (4)$$

(2) 第 2 步

得到每个点的 $\bar{\omega}_i(k)$ 后,对样本 x_i 的 $\bar{\omega}_i(k)$ 值是由第 k 个高斯模型产生,因此在估计第 k 个高斯模型的参数时,使用 $\bar{\omega}_i(k) x_i (i = 1, \dots, N)$ 这些数据采用最大似然法去估计其值:

$$\mu_k = \frac{1}{N} \sum_{i=1}^N \bar{\omega}_i(k) x_i \quad (5)$$

$$\sigma_k = \frac{1}{N} \sum_{j=1}^N \bar{\omega}_i(k) (x_i - \mu_k) (x_i - \mu_k)^T \quad (6)$$

$$N_k = \sum_{i=1}^N \bar{\omega}_i(k) \quad (7)$$

2 基于 V4L2 视频采集流程

系统为适应各种不同硬件的要求及提高视频采集速率,整体采用分层结构来实现。底层使用 V4L2 的驱动编程接口,直接对摄像头数据进行采集,避免了 opencv 库因不同硬件平台摄像头型号的不同而导致的视频帧数据类型无法识别的问题,从而提高了视频帧采集的速率,并提高整体系统监控的兼容性。

V4L2 是在 Linux 的底层视频驱动,系统将其封装成 video_init 自定义类,用于实现视频采集,其配合相关的驱动程序以及摄像头设备,可实现视频采集。V4L2 内部主要是采用相关的回调函数来完成,具体流程如图 1。

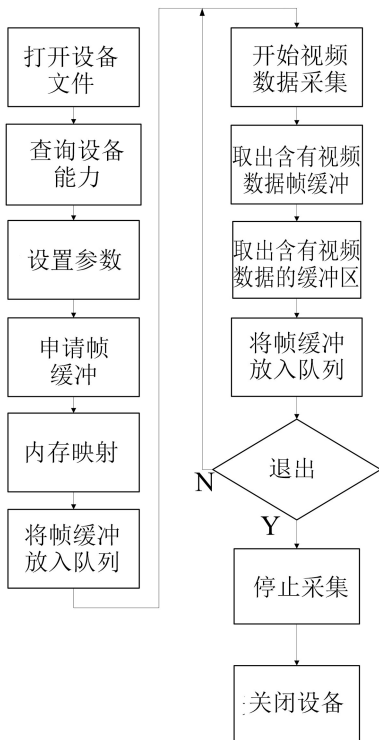


图 1 V4L2 采集视频流程图

Fig.1 Flow chart of video collection through V4L2

利用 V4L2 编程将采集的视频数据的内容封装进自定义的 `video_init` 类中,通过成员函数分别实现:打开设备、设置参数、申请帧缓冲、开始采集、用户内存映射、将帧缓冲放入队列等过程。其中打开设备文件,设备名以设备驱动安装后的设备名“`/dev/video0`”;查询设备主要调用 `VIDIOC_QUERYCAP` 命令实现,并根据开发板硬件要求,设计将视频采集参数设置为 `640 * 480`、编码格式设置为 `MJPEG`,以上的设置需调用 `VIDIOC_S_FMT` 命令通过 `ioctl` 进行设置;设计采用 `VIDIOC_REQBUF` 命令申请 4 个帧缓存,并使用 `mmap` 函数将其映射到用户内存中;并通过 `VIDIOC_QUERY` 命令将申请到的帧缓冲放入采集输出队列;随后可通过 `VIDIOC_STREAMON` 开始视频数据采集,采集到数据后可采用 `VIDIOC_DQBUF` 命令取出帧缓冲,再通过 `VIDIOC_QBUF` 将帧缓冲重新排入输入队列,完成步骤后按照需求停止设备采集及关闭设备。

3 物体检测部分程序流程

系统采用 `opencv` 视觉库联合 V4L2 视频接口分层编程方式,完成视频数据的采集及分析。摄

像头底层数据的采集通过 V4L2 的驱动编程实现,并结合 `qt` 将其封装成自定义的 `video_init` 类,采集视频帧数据,采集后的数据定义为 3 通道、8 位无符号整型数据,即 `CV_8UC3`。采集完的视频数据可采用 `opencv` 的视觉库进行数据类型转换后,调用相应函数循环分离前景图像,并通过腐蚀降噪后,显示于界面上,同时更新背景,流程如图 2。

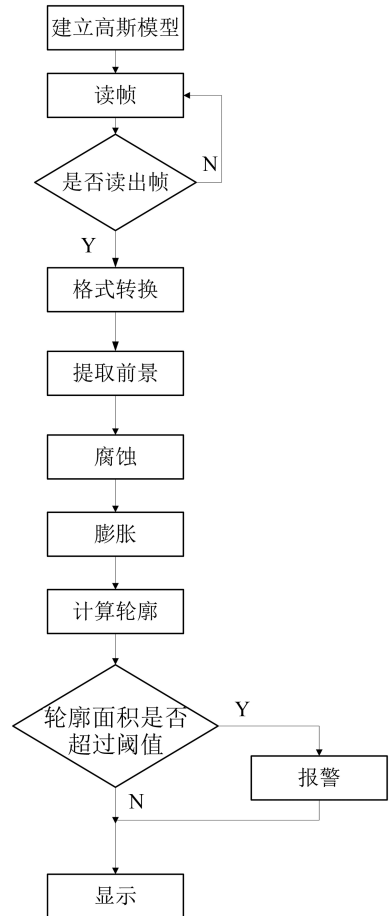


图 2 改进后高斯算法处理流程

Fig.2 Processing flow of the improved Gaussian algorithm

设计在实现过程中,将采集后 `CVmat` 格式的视频帧需先转换成 `mat` 格式,然后调用 `opencv` 库中的 `backgroundmog2` 构造函数分离背景及前景,并将前景信号进行腐蚀、`dilate`、而后对前景图片的二值化结果调用 `cvFindContours` 进行轮廓描述,并使用 `cvArcArea` 以及 `cvArcLength` 计算前景轮廓面积及周长大小,对于面积及周长超过阈值,进行预测。

4 系统调试及分析

系统利用 opencv2.4.9 版本,使用 qt4.7.0 库进行界面程序的设计,对比帧差法以及改进后调用高斯混合模型提取背景的结果如下。

在白天正常光照情况下,见图 3、图 4。

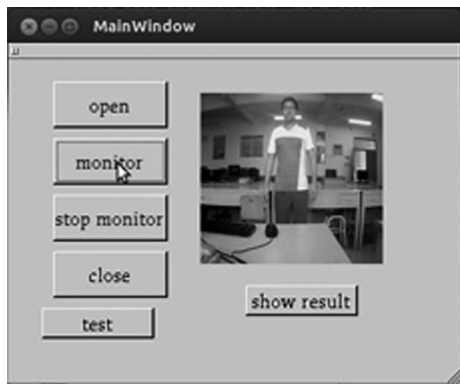


图 3 正常视频监控界面

Fig.3 Normal video monitoring interface



图 4 提取前景图片对比结果

Fig.4 Comparison results of extracted foreground pictures

图 3 为视频采集后播放的结果,图 4 为提取前景的结论对比,左侧是高斯背景提取后的结果,右侧为帧差法提取背景的结果,从图 3、图 4 可知,在日常光照的情况下,改进后的高斯背景能够更完整的提取前景目标。

图 5 为轮廓描述后的结果,左侧为混合高斯模型法提取出的前景图片,右侧为描述出的前景目标的轮廓,在描述轮廓的过程中,系统通过计算轮廓的周长以及面积,将小面积和周长的噪声滤除,得出前景目标的轮廓。

系统在黑夜或者光照条件弱的情况下,见图 6、图 7。



图 5 轮廓描述结果

Fig.5 Contour description results

图 6 为光照弱的情况下视频帧采集后播放的结果,图 7 为提取前景的运算结果,其中左侧为帧差法经过二值化后的结果,右侧图片为采用混合高斯背景分离后提取的前景图像,经过对比可以得到在光照弱的情况下,混合高斯背景法能够更好的提取前景图像。

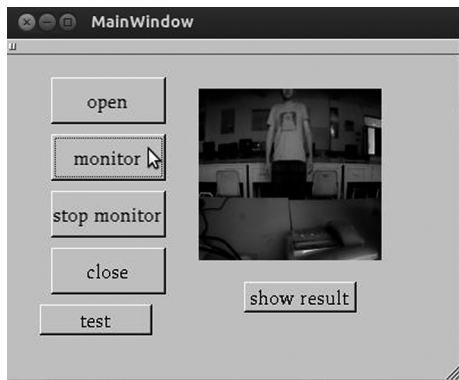


图 6 正常视频监控界面(光照弱)

Fig.6 Normal video monitor interface (weak illumination)



图 7 提取前景结果对比(光照弱)

Fig.7 Comparison results of extracted foreground pictures(weak illumination)

经过高斯背景提取出的前景图像经过轮廓描

述(见图 8),可得到完整的轮廓描述结果,同时将计算其面积和周长并报警。



图 8 描述前景轮廓的结果

Fig.8 Contour description results of foreground objects

5 结语

基于仓库存储应用背景下设计的自动监控系统,采用分层的软件结构完成,无论在光照条件好或者差的情况下,通过混合高斯背景模型截取的前景目标都更为完整。混合高斯模型虽然能较好的提取前景目标,但开始监测时需要经过 10 帧的计算时间,并在光照条件强的情况下会出现较多噪点,因此系统在后续的轮廓提取过程中通过计算轮廓而将噪声滤除。

参考文献:

- [1] 刘升. 基于 V4L2 的嵌入式视频监控系统[J]. 微计算机应用, 2011, 32(1): 37-42.
- [2] 曲国志, 肖作江. 基于 Matlab 的动态目标模拟发生技术研究[J]. 长春理工大学学报(自然科学版), 2010, 33(3): 130-132.
- [3] 马森. 动态背景下的动态物体检测[M]. 哈尔滨: 哈尔滨工业大学, 2015.
- [4] HORN B, SCHUNCK B. Determining optical flow[C]//Technical Symposium East, 1981: 319-331.
- [5] SUN D, ROTH S, BLACK M. Secrets of optical flow estimation and their principles[C]//IEEE Computer Society Conference on Computer Comparison of foreground results Vision and Pattern Recognition, 2010: 2432-2439.
- [6] SUN D, ROTH S, LEWIS J P, et al. Learning optical flow[C]//European Conference on Computer Vision(ECCV), 2008: 83-97.
- [7] 张文强, 路红, 陈义东, 等. 动态场景中的自适应背景建模研究[J]. 中国图象图形学报, 2009, 14(12): 2627-2630.
- [8] 李观金. 基于计算机互联网技术的智能远程视频监控系统设计[J]. 现代信息科技, 2017(6): 83-85.
- [9] ZHANG X, YANG Y H, HAN Z, et al. Object class detection: a survey[J]. Association for Computing Machinery Computing Surveys(CSUR), 2013, 46(1): 1311-1325.
- [10] 韩国强. 浅谈智能视频监控技术及其主要应用[J]. 计算机与网络, 2014(2): 62-65.

(特约编辑: 黄家瑜)