

基于 DM6437 的运动目标检测系统*

周建雄, 张笑微

(西南科技大学 信息工程学院, 四川 绵阳 621010)

摘要:采用非参数化的核密度估计完成背景减法, 设计了基于高端数字媒体处理器 KM6437 的实时运动目标检测系统。提出了一种有效的背景更新策略, 使得背景的背景更新更加可靠, 提高了检测的精确度。实验表明该系统能够对目标进行比较实时和准确的检测。

关键词:DM6437; 非参数化的核密度估计; 背景更新

中图分类号: TP391.4 **文献标识码:** A

Moving target detecting system based on DM6437

ZHOU Jian Xiong, ZHANG Xiao Wei

(School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China)

Abstract:A real-time target detecting system was designed based on high-end digital media processor DM6437. Non-parametric kernel density estimation was used during background subtraction. An effective background updating method was proposed to guarantee reliability in updating and precision in detecting. The experiments have shown the system's real-time and accuracy in target detecting.

Key words: DM6437; non-parametric kernel density estimation; background updating

运动目标检测与跟踪是近年来机器视觉领域研究的热点与难点, 准确快速地实现目标的定位是学者们讨论的焦点。运动目标检测与跟踪技术已经广泛应用于智能视频监控、数字视频会议、交通信息检测等领域。运动目标的检测, 通常采用的是差分^[1-3]的方法, 对于背景静止的情况, 采用差分法相对比较简单。通过帧间差分或背景差分的方法就可以提取物体的运动信息, 再辅助以其他特征信息完成特定目标的检测。对于背景是运动的情况, 目标检测通常需要通过全局运动估计^[4]的方法对背景进行补偿, 使得背景在相邻的 2 帧中保持空间位置不变, 再施以差分法得到运动目标。

现有的目标检测算法大部分都是基于 PC 平台, DSP 拥有专用信号处理的硬件结构和流水线技术, 随着 DSP 器件性能的不提高, 在图像处理的某些方面其性能已经超过了 PC 平台。随着嵌入式时代的到来, 人们对

系统体积、性价比、功耗等因素的要求更高, 使得基于 DSP 平台的视频图像处理算法研究和软件开发成为一种趋势。

1 系统硬件平台

TI 公司的 DM6437^[5]是一款在达芬奇技术框架下的面向数字视频设计应用的数字媒体处理器, 采用 C64x+ 核, 相比于 C64x 核, 一方面增加了适合流媒体处理的指令, 提升了 DSP 在流媒体方面的处理能力, 另一方面提升对操作系统的支持的鲁棒性。芯片主频达到 600 MHz, 采用增强型超长指令字结构, 使得单时钟周期能够处理多条指令。芯片含有 8 个功能单元和 64 个通用寄存器平均分配于 A 和 B 2 个寄存器组中, 功能单元中的 6 个算术逻辑单元支持单周期 1 对 32 bit 或 2 对 16 bit 或 4 对 8 bit 数据的算术运算, 2 个乘法单元支持单周期 4 对 16 bit 数或 8 对 8 bit 数据乘法运算。片上存储 32 KB

* 基金项目: 四川科技厅应用基础研究项目(2008ly 0115-2)

图形、图像及多媒体

Image Processing and Multimedia Technology

的L1P, 80 KB的L1D和128 KB的L2都可灵活配置为CACHE和RAM。32 bit的DDR2存储控制器可最大寻址256 MB的存储空间, 最高支持333 MHz总线速率, 因此可以连接DDR2-400 DRAM。64通道增强型的DMA控制器EDMA3。同时拥有2个可以配置的Video-Port。

系统的硬件框图如图1所示。系统首先从摄像头采集PAL制式模拟视频信号, 通过TVP5146解码为YUV422格式的数字图像信号, 为了方便后期图像处理算法, 需要将采集的YUV422格式的数字图像转化YUV420格式, 这里充分利用硬件结构通过EDMA3完成图像格式的转换, 结果存储于DDR2中, DM6437访问DDR2完成数据处理后将结果送出显示器显示。TI公司为开发DSP的外设驱动程序, 推出了DSP/BIOS DDK(Device Driver Kit)^[6], 定义了标准的设备驱动模型, 并提供了一系列针对视频的API接口, FVID驱动就是其中之一, 通过FVID驱动接口函数能够很容易对数据通道及缓存进行配置, 对采集的视频数据进行访问。

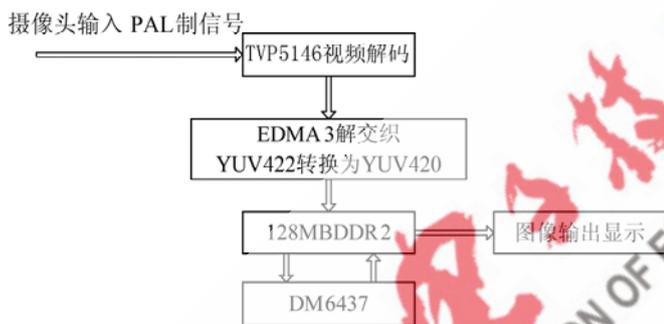


图1 系统硬件框图

2 系统软件设计

系统的软件框图如图2所示。软件系统采集的图像数据为标准PAL格式(720 × 576), 考虑到系统的实时性, 对采集的图像数据的长宽分别进行了1/2缩放, 即处理的图像数据大小为360 × 288。由于算法中需要用到RGB图像数据, 所以需要首先进行图像格式转换, 接下来对3通道的图像数据进行中值滤波去除图像的噪声。通过时间平均的方法得到完成背景初始化, 非参数核密度估计的背景减法得到运动区域, 对运动区域进行阴影检测去除阴影, 对去除阴影的区域进行连通成分标识获取运动区域的位置和形状信息, 最后通过识别得到运动目标。

2.1 目标检测算法

基于背景减法的目标检测方法, 首先需要考虑到背景初始化的问题。通常情况下系统很难得到一帧很干净的背景, 因此可以采用时间平均的方式得到背景的近似值。由于考虑到后面的算法需要建立历史背景图像队列, 算法中采用分段时间平均的方法建立背景

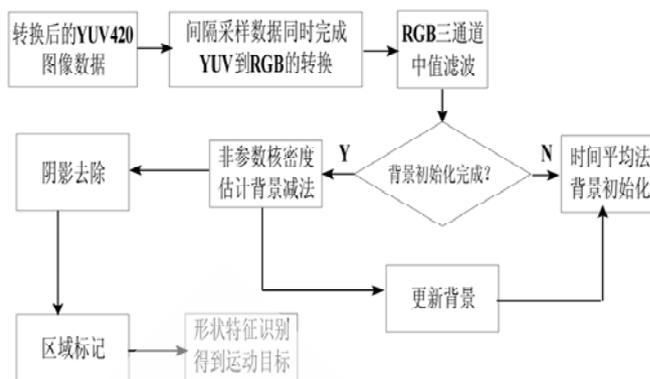


图2 系统软件框图

图像队列, 即队列中的每一幅背景图像都是通过一段时间采集并对图像做平均后得到的, 当背景图像队列建立完成, 背景的初始化也随之完成。

常用的静止背景建模方法是采用参数化的方法, 典型的有单高斯背景模型^[1]和混合高斯背景模型^[2]。对于单高斯背景模型来说假设背景中的每个像素服从单个高斯分布, 通过统计一段时间内每个像素的均值和方差来建立高斯模型。如果当前像素值与背景的差值大于一定的阈值则将该像素判断为运动点, 否则判为背景点, 对于背景参数的更新通常采用固定速率。而通常情况下对背景中的每个像素做单高斯分布的假设是不合理的, 例如背景中有树枝的晃动或水纹的波动, 同时背景的更新速率也是很难控制的。对于基于混合高斯分布的背景建模, 用多个高斯分布拟合背景中的每一个像素, 但是经常可能出现像素的概率分布无法用参数形式的高斯分布拟合的情况, 如何根据实际情况正确地选择高斯分布的个数也是一个很难解决的问题。同时该方法的计算量非常大, 很难满足嵌入式系统实时性的要求。

相比于参数方法的背景建模, 非参数的方法不需要确定概率密度函数, 也不需要考虑如何设置模型的参数, 实现起来比较简单, 算法复杂度低, 非常适合嵌入式系统的实现。常用的方法是核密度估计的方法。非参数的核密度估计方法早在20世纪50~60年代就被提出, 对概率密度函数 $f(x)$ 的估计, 通过一系列样本点来表示, 如下式所示。

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x_i - x}{h}\right)$$

其中 n 表示样本的数目, h 为核带宽, 表示样本点的参与程度, h 越大参与的样本点越多, $K(\cdot)$ 为核函数, K 的不同代表了不同的样本点加权方式。参考文献[7]将核密度估计的方法引入了目标检测的背景减法中, 算法的本质就是对当前时刻以前的历史数据进行开窗处理, 消除原始的陈旧数据对当前判决的影响, 对背景中的每一个像素点的概率密度的估计采用窗口内的数据点加权的方

图形、图像及多媒体

式得到, 算法中唯一需要确定的参数就是核带宽的大小, 核带宽通常可以根据经验值得到。本文采用了RGB空间的核密度估计背景减法。写成向量的形式为:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^3 \frac{1}{h_j} K\left(\frac{x_j - X}{h_j}\right)$$

其中 $x = (R, G, B)^T$, 算法中假设 R 、 G 、 B 三个通道相互独立, 并且具有相同的核带宽。为了省去不必要的浮点运算, 核函数选择了三角核, 当 n 和 h 确定时, $\frac{1}{nh}$ 和 $\frac{1}{h}$ 为常数项, 去掉 $\frac{1}{nh}$ 和 $\frac{1}{h}$ 不影响最终判断结果, 则上式可以进一步简化为:

$$\hat{f}_R(x^R) = \sum_{i=1}^n K(x_i^R - x^R)$$

$$\hat{f}_G(x^G) = \sum_{i=1}^n K(x_i^G - x^G)$$

$$\hat{f}_B(x^B) = \sum_{i=1}^n K(x_i^B - x^B)$$

$$\text{其中 } K(x) = \begin{cases} h-x & x \leq h \\ 0 & x > h \end{cases}$$

最终的背景减法为如下公式:

$$D(x, y) = \begin{cases} \text{foreground} & \min(\hat{f}_R(x^R), \hat{f}_G(x^G), \hat{f}_B(x^B)) < \theta \\ \text{background} & \text{else} \end{cases}$$

实验中数据窗选取为 8, 核带宽 h 选取为 15; 阈值 θ 取 10。

2.2 背景更新策略

背景模型的更新对于实现稳定、准确的目标检测和跟踪十分重要, 盲目更新会将运动区域也作为背景添加到背景当中来, 这样在接下来的目标检测过程中会出现误检测、漏检测的问题。本文提出了一种新的背景更新的方法。其流程如图 3 所示。

算法中的背景由 8 个背景图像序列构成, 8 个背景图像序列组成 FIFO 队列, 队头存放较新的背景数据, 队尾存放较旧的背景数据, 背景更新一次队列向队尾方向移动一次, 最陈旧的背景从队尾被剔除, 队首存放最新更新的背景。

考虑背景不应该对场景中突然的变化干扰很敏感,

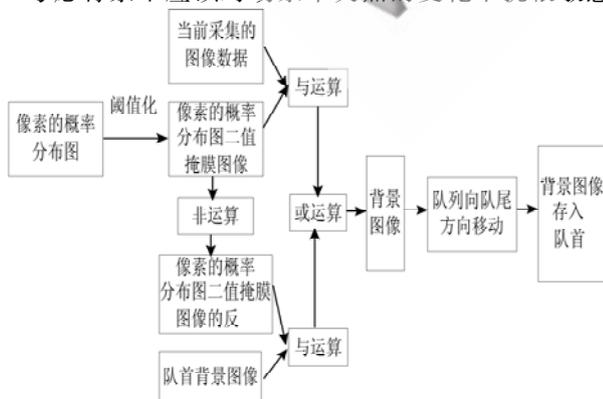


图 3 背景更新的流程

算法中并没有每一帧都对背景进行更新, 而是建立了一段时间内图像中每个像素化分为运动前景和背景的概率分布图, 通过对概率分布图设置阈值来判断该点是划分为运动点还是背景点, 当概率分布大于或等于设定阈值时认为该点为运动点, 反之为背景点。如下式所表示:

$$p_map(x, y) = \frac{\sum_{i=1}^n \text{binary_image}_i(x, y)}{n \times 255}$$

其中 p_map 为一段时间内图像中每个像素化分为运动前景和背景的概率分布图, binary_image_i 为第 i 幅通过检测得到的二值图像, n 表示背景更新的间隔, 以图像帧表示, 假设二值图像中运动点值为 255, 背景点值为 0。

在实际中可能出现一种临界的情况, 即在更新时刻之前图像中的某个像素点都被判为背景点, 而从更新时刻或离更新时刻很近的时刻起该点变为运动点, 这时如果简单地按照上述概率统计的方法很可能真正的运动点也被判为了背景点。因此引入了参量 i^λ , 对概率分布图的统计进行加权, 离更新时刻越近, 统计的结果对最终的判别影响越大。重新改写上式得到:

$$p_map(x, y) = \frac{\sum_{i=1}^n i^\lambda \times \text{binary_image}(x, y)}{n \times 255 \times \sum_{i=1}^n i^\lambda}$$

在算法中选择每 20 帧更新一次背景, 此时 λ 取 5。考虑到式中含有指数运算, 为了降低算法复杂度, 通过离线表格化 i^λ , 最终通过查表方式实现上述运算, 大大降低了算法复杂度。

2.3 阴影检测

阴影的存在给目标的准确检测带来了很大的困难。如果目标与阴影粘连在一起, 将对后续的基于形状的目标识别带来很大的误差, 同时如果多个目标通过阴影粘连在一起将可能被系统识别为一个目标。因此需要对差分后的图像进行阴影检测, 去除阴影。参考文献[7]提出了基于 HIS 空间的阴影检测算法, 其中 H 分量体现了图像的颜色信息, S 分量体现了图像的饱和度信息, 而 I 分量体现了图像的亮度信息, 3 个分量彼此正交, 更符合人对色觉的反应。通过下式完成阴影的检测。

$$S(x, y) = \begin{cases} \text{shadow} & a \leq \frac{C_I(x, y)}{B_I(x, y)} \leq \beta \ \& \& \\ & C_S(x, y) - B_S(x, y) \leq T_S \ \& \& \\ & |C_H(x, y) - B_H(x, y)| \leq T_H \\ \text{foreground} & \text{else} \end{cases}$$

其中 S 为结果图像, C 为当前采集的图像, (x, y) 表示像素在图像中的位置, C_H 、 C_S 、 C_I 分别代表当前采集的图像的 H 、 S 和 I 分量, B_H 、 B_S 、 B_I 分别代表背景图像的 H 、 S 和 I 分量。

2.4 区域标记及目标识别

通过标记算法可以得到运动区域的位置、区域面积、区域的长度和宽度。在实验中选择行人作为跟踪目标,识别特征选择目标的面积、长宽比、目标面积与外接矩形框面积的比值3个参数作为目标的特征,当标记的运动区域满足上述特征时,系统将其识别为行人并通过外接矩形框表示。

3 代码优化

代码的优化一般可以按照以下三步实现:

(1)首先根据算法原理编写C代码,保证算法的正确性。通过DSP/BIOS接口函数CLK_gettime()可以得到系统时间,从而可以测试出算法中每个模块的运行时间,对于比较耗时的模块进入(2);

(2)C代码级优化。打开编译优化开关,通常采用-k, -mt, -o3, -mw的选项组合。其中-k选项用于保留编译产生的汇编语言文件;-mt告知编译器程序中不使用别名技术;-o3最大程度地优化,执行各种优化,如软件流水、循环展开和单指令多数据流,同时使用各种文件级的优化。-mw是在编译产生的汇编语言文件加入软件流水线信息。优化时应该着重对循环体进行优化,因为图像处理算法中的循环次数一般比较多,只要一次循环得到一点改善,对于整个循环来说将得到很大的改善。通过分析循环核找到冗余部分,调整C代码。通过反复分析编译产生的汇编代码,调整C代码,完成C代码级的优化。同时C代码的优化还有一些小的技巧:

①避免在C代码中将int和long型混用,因为long型数据使用40bit操作,需要占用2个寄存器;

②对于循环计数器尽量使用unsigned int或int类型,以免不必要的符号扩展指令;

③使用字访问短型数据,可以减少存取时间;

④使用intrinsic数据打包指令,DSP拥有32bit的功能寄存器和通用寄存器,1个时钟周期支持1对32bit或2对16bit或4对8bit数据的算术运算,4对16bit数据乘法或8对8bit数据乘法,利用这些指令能够使代码效率得到极大的提高;

⑤对于乘法和除法,尽量通过移位操作来实现,在算法设计时应该尽量考虑到使乘除法中的常数为2的幂次;

⑥对于某些算法可以通过查表的方式来实现,这将给代码效率带来成倍的提高;

⑦将浮点运算转换为定点运算来实现。大多数DSP都是定点DSP,如果进行浮点运算需要调用浮点运算库来实现,将同时打断软件流水,运行效率非常低,所以应该尽量避免浮点运算;

⑧利用TI提供的算法库dsplib和imagelib,算法库

中包含高质量的代码,效率相当高。

如果此时还不能满足要求,则进入(3)。

(3)汇编级优化。当上述方法都不能满足要求时,就必须对C代码的关键部分编写线性汇编。编写线性汇编需要对汇编指令相当熟悉,因其需要耗费大量的时间和精力。

通过步骤(2)的优化,系统已经能够满足实时性的要求,达到20帧/s的处理速度。所以没有进行汇编级的优化和后续的CACHE优化。表1列出了几个比较耗时的核心模块的优化,对于二值图像标记算法来说,连通区域的个数直接影响了算法运行的时间,所以选择了连通区域个数在100左右进行测试。

表1 代码优化结果

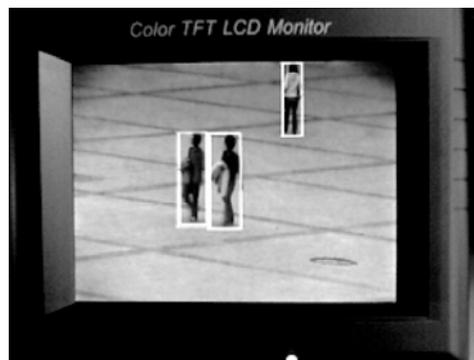
	优化前 (时间/ms)	优化后 (时间/ms)	采用的方法
背景的时间 平均初始化模块	16.72	5.48	intrinsic数据打包处理,将除法改为移位操作。
核密度估计 背景减法模块	33.86	14.45	intrinsic数据打包处理,消除存储相关。
二值图像标记 模块	43.567	27.45	消除存储相关,将临时标记的等价表放置于LID中,加快了等价表的整理速度。

4 实验结果

实验选择了行人作为对象,在图4的(a)和(b)中,LCD显示器显示了2个不同场景中对行人的检测结果。基于



(a)场景一



(b)场景二

图4 行人检测

(下转第61页)

(上接第 57 页)

核密度估计的背景减法在实验中体现了良好的效果,同时本文提出的背景更新策略进一步提高了目标检测的精度和准确度。系统能够完整地检测出进入视野的行人,同时达到了 20 帧/s 的处理速度。

本文采用了 TI 公司的 DM6437 数字媒体处理器搭建了实时运动目标检测系统,并对算法代码进行了优化。算法中采用了基于非参数的核密度估计背景减法算法,该算法非常适合嵌入式平台,算法复杂度低,容易优化,并且效果比较理想。同时提出了一种新的背景更新策略,使得背景的更新更加可靠和准确。设计和实验中也遇到了一些问题。在算法方面:二值图像标记算法的运行效率与标记区域的个数及区域的复杂程度有很大的关系,如何减小这种依赖程度。在代码优化方面:如何将算法中的部分模块设计为并行执行,这样可以借助 DMA 实现片外数据的后台片上搬运,减少 DSP 访问片外数据所需的存取等待时间,进一步提高系统的运行效率。

参考文献

[1] OMAR J, KHURRAM S, MUBARAK S. A hierarchical approach to robust background subtraction using color and gradient information [A]. Proceedings of the Workshop on Motion and Video Computing [C]. 2002:22-27.

《信息化纵横》2009 年第 12 期

《电子技术应用》www.ChinaAET.com
[2] STUDEFELT C, GRIMSON W. E. L. Adaptive background mixture models for real-time tracking [A]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, [C]. 1999, (2):246-252.

[3] ELGAMMAL A, HARWOOD D, DAVIS L. Non-parametric model for background subtraction. IEEE FRAME-RATE Workshop, 1999:246-252.

[4] DUFAUX F, KONRAD J E. Robust and fast global motion estimation for video coding [J]. IEEE Transactions on Image Processing, 2000:497-501.

[5] Texas Instruments Incorporated. Texas instruments. TMS320DM6437 digital media processor, 2008.

[6] Texas Instruments Incorporated. Texas instruments. TMS320C6000 DSP/BIOS application programming interface (API) reference guide, 2004.

[7] CUCCHIARA R I. Shadow suppression in moving object detection with HSV color information [J]. Proceedings of Intel. Transportation Systems Conference, 2001.334-339.

(收稿日期:2009-03-21)