

基于异构多核嵌入式平台的视频监控系统

尹 雷, 卿 粼波, 滕奇志, 付 雄

(四川大学 电子信息学院 图像信息研究所, 四川 成都 610065)

摘 要: 针对传统 DSP 对视频数据处理能力的不足, 以 TI 公司开发的多核异构处理器 TMS320DM8168 为核心, 设计了一套嵌入式视频监控系统, 对视频处理通路做了详细介绍, 提出了一种在 DSP 核心中动态切换算法的方案, 并针对智能监控实现了摄像头遮挡检测与移动侦测应用.

关键词: DM8168; 智能监控; 嵌入式平台; MCFW

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2016)05-0097-05

Video Surveillance System Based on Heterogeneous Multi-core Processor Embedded Platform

YIN Lei, QING Lin-bo, TENG Qi-zhi, FU Xiong

(College of Electronics and Information Engineer,

Image Information Institute, Sichuan University, Chengdu 610065, China)

Abstract: Aiming at the shortage of video data processing ability by traditional DSP, this paper takes the TI's heterogeneous multi-core processor TMS320DM8168 as the center, designs a set of embedded video surveillance system, make a detailed presentation of video processing framework and propose a program of dynamic switchable algorithm in the DSP core. Finally, realizes the camera tamper detection and motion detection for the application of Intelligent Surveillance.

Key words: DM8168; Intelligent Surveillance; embedded platform; MCFW

1 引言

目前, 国内外对实时视频监控课题进行了深入的研究^[1], 基于异构多核嵌入式平台的视频监控系统具有体积小, 功耗低且功能完善等特点, 已经成为目前主流的监控系统. 文献[2]中使用 DM8168 平台完成了多路视频监控系统的开发, 文献[3]也使用了 DM8168 平台实现了视频处理通路并加入遗留物检测算法, 但大多数文献对于如何更好地利用该嵌入式平台进行算法的集成与开发并未进行深入研究. 为此, 本文以 TI 公司达芬奇系列多核异构 DSP TMS320DM8168 处理器^[4]为核心芯片, 设计了一套完整的嵌入式图像处理平台, 完成了视频通路的搭建, 针对原开发环境的不足给出了在原有软件框

架中获取图像帧信息的方法并提出了一种在 DSP 核心中动态切换算法的方案, 最后在 DSP 核心上实现了摄像头遮挡检测与运动目标识别功能. 实验证明本系统具有调试方便, 算法执行效率高等优点.

2 系统方案

本系统需要实现一套完整的视频监控与处理通路, 需要考虑视频的采集、编解码、存储显示与算法处理等问题. 同时由于单一 DSP 核心处理能力有限, 还需考虑系统的可扩展性即可将视频数据通过 PCIE 等高速接口发送至其他处理器进行二次处理^[5]. 根据上述分析, 设计如图 1 所示的方案.

视频源为 PAL 或者 NTSC 制式的视频流, 首先需要解码原视频流数据供数字处理器处理, 图中

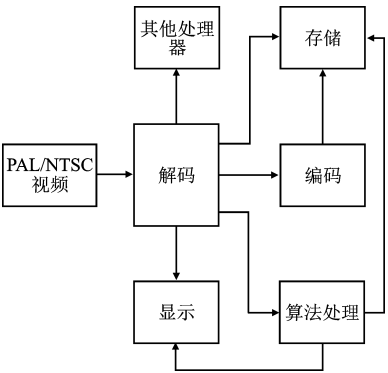


图 1 系统方案框图

经过解码模块后的视频数据均为 YUV 格式. 显示模块与算法处理模块应可直接对 YUV 格式视频进行处理,考虑到单一 DSP 处理能力的不足与调试需要,还需解决 YUV 帧信息物理地址的获取问题以便发送至其他处理器进行处理. 由于直接存储 YUV 数据占用极大地硬盘空间,为了方便视频文件的保存与传输,需要经过编码模块以 H. 264 数据流的形式进行存储. 算法处理模块对 YUV 数据进行处理后可以显示存储并以消息信令传递的方式进行通知,方便使用者观察算法处理后的结果.

在本系统中前端可通过模拟摄像头同时获取四路模拟视频,经过视频解码器转换成数字信号送入 DSP 中,然后对 DSP 中视频数据进行处理,最后采用视频编码器编码,视频可以存储在本地设备,或通过以太网、PCIE 等接口传输到其他设备.

3 系统软件设计

本文采用了 TI 公司针对达芬奇处理器的多核视频开发工具包 DVRRDK. 在此开发包中,对视频链路的处理采用了 MCFW(多通道视频处理软件框架)方案,本节在 DVRRDK 环境中搭建了视频处理链路,解决了 YUV 帧信息存储问题,提出一种在 DSP 核心上实现算法切换的方案,最后在 DSP 中实现了摄像头遮挡检测与移动侦测应用.

3.1 DM8168 软件框架

DM8168 在对视频进行一系列处理时,采用了层次化的结构来进行软件开发,这种层次化的结构被称为 MCFW(多通道视频处理软件框架). 图 2 所示为 DVRRDK 中针对多通道视频通信的软件框架.

由图 2 中可以看出,主要包含了四个物理核心,分别为 Host A8,VPSS M3,Video M3 以及 DSP,并且任意 2 个核之间都可以通过 Syslink/IPC 来互相通信. 在该框架中,Link 是在视频数据流中最基

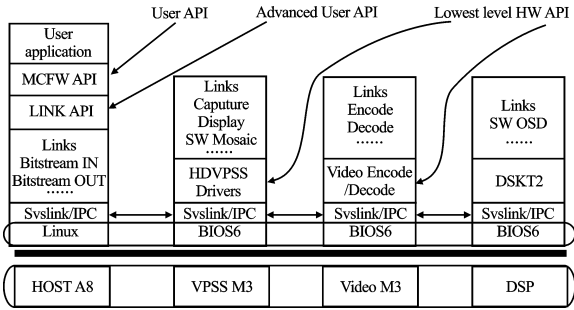


图 2 多核视频软件开发框架

本的单元模块,用户可以直接通过 Link API 层提供的控制函数,组合 Links 层的节点 link,创建用户自定义的应用实例 use-case. 其中节点 link 实际是一个 BIOS6/Linux 独立线程,能够独立处理来自其他节点 Link 中的视频数据. 每个节点 Link 中都有输入输出消息队列,用于与其他节点 Link 交换视频帧或数据流. 值得注意的是,为了节省内存开销,当通过 Link 传递视频帧数据时,只是传递帧数据的 Buf 指针,而帧数据本身并不会被拷贝.

3.2 视频处理通路实现

整个视频的传输通路基于多通道视频处理软件框架,DM8168 中所有核心均被有效使用,VPSS M3 处理器负责视频的采集、去噪、显示等,Video M3 负责将 YUV 格式编码成 H. 264 流文件,C674x DSP 处理器负责实现算法,HOST A8 处理器主要用于控制 PCIE、SATA、以太网、I/O 接口等外设,文件的读写操作也是由 A8 核心实现的. 通过不同 Link 的连接,实现了多处理器间视频数据的传递,本系统具体 Link 连接图如图 3 所示.

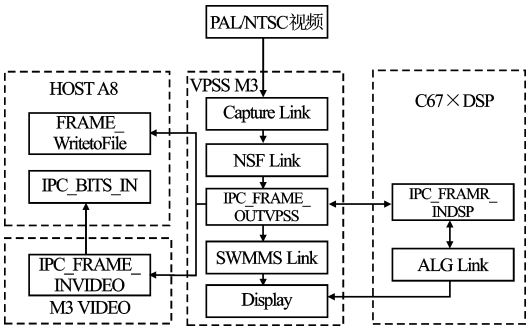


图 3 视频处理流程软件结构图

本系统可同时接收 4 路 D1 格式视频信号,在 VPSS M3 核中通过采集模块 Capture Link 与去噪模块 NSF Link 后,便可将 YUV 格式视频数据发送至其他核心进行处理,也可以直接在本核心中通过 SWMS Link 进行画面的拼接与显示. IPC_FRAME_OUTVPSS 来完成图像数据的提取与同步. 该模

块作为 IPC FRAME LINK 的一个实例,除了能在不同模块间传递数据外,还有支持不同处理器核心对同一帧的数据进行同步操作的功能. 为了方便视频文件的保存与传输,YUV 数据需经过 M3 VIDEO 的编码,以 H. 264 流数据的格式存储 SATA 硬盘中. 算法模块可通过设置 IPC_FRAME_OUTVPSS 的 PROCESS LINK 属性,指定由哪个模块处理当前 Link 中的数据. DSP 核上的 dspAlg 模块可实现帧图像的修改,这种修改会同步地反映在 IPC_FRAME_OUTVPSS 模块传送的数据中. 这就是在 IPC_FRAME_OUTVPSS 与 IPC_FRAME_INDSP 之间使用双向箭头的含义. 处理后的图像帧数据还可以通过 HOST A8 进行文件存储操作,方便使用者观察算法处理结果.

在系统调试的过程中,用户经常需要获得图像帧信息用于调试^[6],同时考虑单一 DSP 处理能力有限,当遇到复杂应用时需要把本系统进行扩展,即把采集到的视频以 YUV 格式发送至其他处理能力更强的处理器中进行算法处理,所以需要构建从 VPSS M3 到 HOST A8 的直接通路用于帧信息的存储. MCFW 的 API 中并没有能提供帧图像数据的输出功能,本文通过以下方式实现了 YUV 的直接存储.

由于 API 中只提供了图像帧信息的物理地址,首先需要在 MCFW 框架添加字符设备文件 mem 和系统函数 mmap(),将设备内存映射到系统内存中,在得到数据的虚拟地址后,在 A8 核心中开辟 fwrite()线程对文件进行写操作,图像在内存中的排列如图 4 所示,图像数据通过帧宽度与帧长度这种二维方式对齐排列. 在内存的实际存储中,真实行宽并不等于帧图像的宽度 frameWidth,而等于 framePitch 帧间隔. 在 DM8168 的软件框架内图像帧信息都是以该结构在内存中进行存储,只有当外部 API 进行调用时才会进行剪裁. 因此在计算图像数据的大小时,不能简单地用帧宽度乘帧长度,还需数考虑在内存中的帧间隔以及起始坐标. 这样便可得

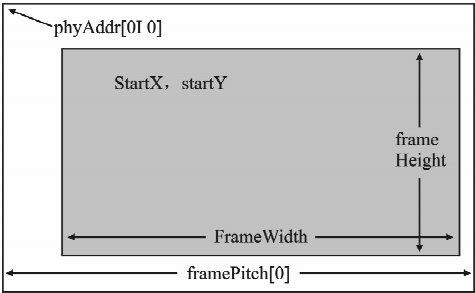


图 4 图像亮度信息数据在内存中的分布

到一帧图像的虚拟地址与实际大小,通过写文件线程实现了帧信息的存储.

3.3 算法集成与智能监控算法实现

在实际的开发调试过程中发现,在 MCFW 框架中虽然提供了 ALG LINK 用于使用者集成算法,但是并没有介绍如何将用户自己算法进行集成. 同时根据例程,ALG LINK 只提供了一个算法接口,这样给那些想在多个算法间进行切换的用户带来了很大不便. 基于这种需求,本文实现了算法函数在 MCFW 框架中的集成并在原有框架的基础上进行了改进,实现了多个算法的动态切换.

首先,在 MCFW 框架中,需要调用 PROCESS LINK 将 ALG LINK 引入到原有框架中. PROCESS LINK 只能用于与 ipcFrameOutHLOS 或 ipcFrameOutRTOS 进行连接,所以本文在采集显示通路中的 NSF LINK 后加入 IPC_FRAME_OUTVPSS,并将 PROCESS LINK 连接到 IPC_FRAME_OUTVPSS 后,PROCESS LINK 后是 IPC_FRAME_INDSP Link,视频帧数据先通过 PROCESS LINK 从前向 Link 导入到 DSP 中,DSP 处理完成后再通过 PROCESS LINK 将数据传递到后向 Link 中. 这样便在原有视频通路的基础上增加了算法链路.

算法链路添加完成后,需要在链路里集成图像处理函数. 首先,需要在框架中创建初始化函数 AlgLink_algCreate(),该函数中首先申请内存空间用来存储算法中用到的变量,该空间可被不同图像处理函数共享. 然后调用系统函数 System_linkGetInfo()获取前向 Link 的基本信息来判断是否有帧数据达到. 初始化后,在函数 AlgLink_algProcessFrame()中调用 System_getLinksFullFrames()函数从前向 Link 获取新数据. 在新数据到达时,通过使用函数指针指向不同子函数来进行数据处理. 函数指针指向的改变可以通过 A8 核心向 ALG LINK 传递不同消息信令来完成. ALG LINK 主控线程获取相应消息流程如图 5 所示.

为了便于不同算法的灵活切换,不同算法函数参数设置应与默认的帧数据处理函数 AlgLink_algProcessFrame()一致. 输入参数应为结构体 AlgLink_Obj 与 FVID2_FrameList. AlgLink_Obj 中存储 Link ID、Link 中队列数量等基本信息,每个子算法均可使用. FVID2_FrameList 中包含视频帧信息的实际地址、时间戳、通道 ID 等重要信息. 执行不同图像处理函数时,都需先调用 Link 管理核心函数 System_getLinksFullFrames(),该函数内部会发送

消息到对应的前向 Link, 触发该 Link 中已经向管理模块注册的回调函数 System_LinkGetOutputFramesCb() 将帧数据传递给 ALG LINK, 处理完成后使用 system_putLinksEmptyFrames() 将数据交还给前向 Link 用于后续处理。

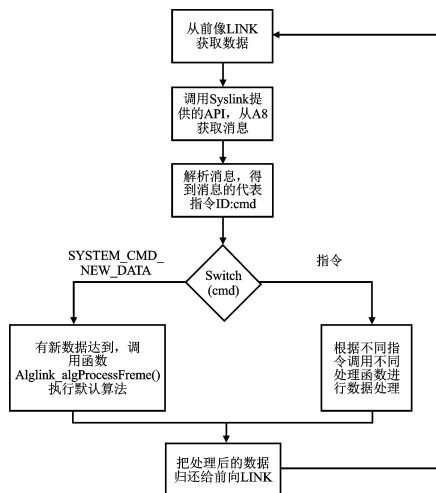


图5 ALG LINK 主控线程获取相应消息流程

这样就在 MCFW 框架中集成了图像处理算法函数并可实现不同算法的动态切换。用户可直接将不同用途的图像处理算法加入到系统中, 并可在系统运行的过程中进行动态切换, 在很大程度上增加了系统的可用性。

在上文基础上, 本文在 DSP 核心中添加场景改变侦测算法^[7], 主要针对输入视频信息实现了以下两项功能:

- (1) 摄像头遮挡检测;
- (2) 移动侦测。

首先将输入的帧数据分割成 CIF 分辨率或者更小分辨率的小块, 这样在一帧画面中可以得到以横纵坐标表示位置的 N 个小块。在遮挡检测应用中需要对完整的一帧进行监测, 而在移动侦测算法中监测者可以选择单个块或者多个块内的区域进行监测。具体如图 6 所示, 实现步骤如下所示。

(1) 初始化

令 x 为小块坐标中的横坐标, y 为小块坐标中的纵坐标, t 表示时间。定义当前帧中某一小块用 $b(x, y, t)$ 表示, 前一帧中相同坐标小块用 $b(x, y, t-1)$ 表示。

(2) 创建场景建模

根据场景复杂度, 设定前 M 帧作为学习样本并对该 M 帧进行建模求取亮度及色度平均值, 求得样本模型:

$$m(x, y, t-1) = \sum_{t=1}^M \frac{b(x, y, t)}{M} \quad (1)$$

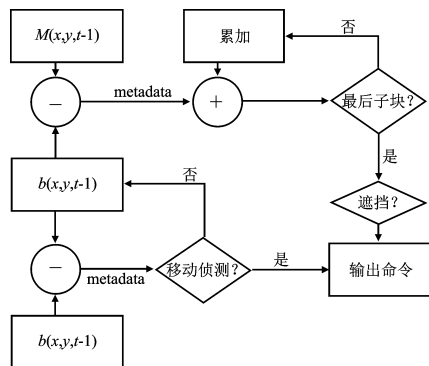


图6 遮挡检测与移动侦测实现框图

(3) 帧差法求取结果

摄像头遮挡检测应用中, 从采集的视频中每秒抽取 5 帧作为 ALG LINK 的输入, 每一帧中需从第一个子块开始与建模子块进行帧差求得子块像素差数据 metadata, 将所有子块做差后的 metadata 进行求和求得一帧数据的像素差 Frame_metadata, 最后将 5 帧图像数据的 frame_metadata 求平均得到最终的 Fina_metadata。当 Fina_metadata 大于某一设定阈值时, 即判断产生遮挡, 通过核间通信模块向 HOST A8 发送 VSYS_EVENT_TAMPER_DETECT 命令, 后期可以利用该命令触发 GPIO 中断等进行报警。

$$\text{metadata} = b(x, y, t) - m(x, y, t-1) \quad (2)$$

$$\text{Frame_metadata} =$$

$$\sum_{\text{子块}=1}^N (b(x, y, t) - m(x, y, t-1)) \quad (3)$$

$$\text{Fina_metadata} = \frac{\sum_{\text{frame}=1}^5 \text{Frame_metadata}}{5} \quad (4)$$

在移动侦测应用中, 只需对选择的一个或者多过子块进行做差同样以每秒 5 帧的速度对前后帧进行做差。当利用式(2)求得 metadata 大于一设定阈值时, 即判断产生了移动。为了针对用户不同需求, 为算法设立了七个等级的灵敏度来改变处理器的判断阈值, 用户可以在监测过程中动态的改变该灵敏度参数, 当将灵敏度调整为最高时, 便可以检测到视频中更细微的变化, 相反如果将灵敏度调整为最低, 两帧之间需要有较大的差异才能判断为移动。当移动侦测事件发生时, 通过核间通信模块向 A8 发送 VSYS_EVENT_MOTION_DETECT 命令, 后期可以利用该命令触发视频的自动存储等功能。

在摄像机被遮挡或者摄像机连接线被移除的情况

下,Linux 系统控制台中显示如图 7(a)所示,在 A8 端显示 TAMPER DETECTED 命令并记录遮挡次数与遮挡时间;图 7(b)显示当有移动侦测发生时,在 A8 端显示 MOTION DETECTED 并记录侦测时间,成功实现场景改变侦测算法并带有算法结果输出功能。

```
yinlei@ubuntu:~$
[m3vpss ] 29459: DISPLAY: Start in progress !!!
[m3vpss ]
[c6xdsp ] 26506: IPC_BITS_OUT : Create Done !!!
[m3vpss ] *** UTILS: DM_TIMER[2] KHz = 27000 KHz ***
[m3vpss ]
[m3vpss ] 29509: DISPLAY: Start Done !!!
[m3vpss ] 29509: DISPLAY: SDOAC(SECI): 20 fps, Latency (Min / Max) = ( 255 /
[m3vpss ] 29509: SNMS: Start in Progress !!!
[m3vpss ] 29509: SNMS: Start Done !!!
[m3vpss ] 29546: CAPTURE: Start in progress !!!
[m3vpss ] 29546: CAPTURE: Start Done !!!
[m3vpss ] 29671: CAPTURE: Fields = 0 (fps = 0), Total Resets = 0 (Avg 0 ms pe)
[TAMPER DETECTED] 18817: SCD CH <1> CAP CH = 0
the num of count=1
[c6xdsp ] 41466: SYSTEM: Opening Msg0 [HOST_MSGQ] ...
[TAMPER DETECTED] 31411: SCD CH <1> CAP CH = 0
the num of count=2
[TAMPER DETECTED] 32609: SCD CH <1> CAP CH = 0
the num of count=3
[TAMPER DETECTED] 33208: SCD CH <1> CAP CH = 0
the num of count=4
[TAMPER DETECTED] 33807: SCD CH <1> CAP CH = 0
the num of count=5
```

(a)遮挡检测结果显示

```
yinlei@ubuntu:~$
4: Display Settings
5: Audio Settings

t: Print detailed system information

e: Stop Demo

Enter Choice: [MOTION DETECTED] 79413: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 80612: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 80611: SCD CH <1> CAP CH = 0
[m3vpss ] DM_TIMER_TEST:TotalElapsed:36208:NumUpdates:999
[MOTION DETECTED] 81218: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 81888: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 82487: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 83066: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 83605: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 84204: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 84803: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 85402: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 86001: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 86599: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 87199: SCD CH <1> CAP CH = 0
[MOTION DETECTED] 87797: SCD CH <1> CAP CH = 0
Ctrl-A 2 for help | 9600 RN1 | NOR | Minicom 2.5 | VT102 | Offline
```

(b)移动侦测结果显示

图 7 检测与侦测显示

4 测试结果及分析

本文以分辨率为 720×576、每秒 25 帧视频的采集、H264 编码、存储、显示作为视频的传输通路,并在其中加入了摄像头遮挡检测与移动侦测算法,经多次实验取平均值后的 CPU 利用率如表 1 所示。

表 1 CPU 利用率情况表

处理器	操作系统	用途	CPU 利用率/%
HOST A8	Linux	系统设置、程序加载、外设控制等	21
VPSS M3	BIOS6	控制视频采集、去噪、显示等	11
VIDEO M3	BIOS6	控制视频压缩、解压(H264 编码、H264 解码)	5
C674x DSP	BIOS6	执行算法	4

通过对该实验中 CPU 利用率分析可知,由于 DM8168 硬件上采用多核异构芯片组成方案,软件上通过多通道视频处理软件框架完成对视频链路的处理,在 DSP 核心中针对算法处理采用了本文提出

的动态切换方案,这种方法能极大地提高整个平台的并行处理能力,使每个处理器都可以均衡的工作,减轻了 ARM 处理器的负载,具有效率高、实时性强等优点,同时 C674×DSP 高达 1.1 GHz 的主频也为实现复杂度更高的算法提供了载体。

5 结束语

多核异构处理器是近年来研究的热点课题,本文针对 DM8168 处理器,设计了一套嵌入式智能监控平台,针对 MCFW 的缺陷进行了改进,方便使用者进行测试,拓展了 MCFW 的应用范围,对使用 DVRRDK 环境进行开发的开发人员有一定的借鉴意义。在此基础上,实现了多路视频的采集显示、H. 264 编解码处理、数据存储等常用视频应用,同时在智能监控方面添加了场景改变侦测算法,具有一定的应用价值。

参考文献:

[1] 李德明,全盛程,叶进. 基于 SC 的多核处理器并行仿真机制的研究[J]. 微电子学与计算机,2014,31(2): 23-26.

[2] 郑亮. 基于 DM8168 多路视频监控系统研制[D]. 杭州:杭州电子科技大学,2014.

[3] 李新文,张璐. 基于 DM8168 的遗留物体检测算法设计[J]. 现代电子技术,2014,37(15):113-116.

[4] TEXAS INSTRUMENTS INCORPORATED, TMS320-DM816x DaVinci Digital Media Processors Technical Reference Manual[M]. Literature: sprugx,2011.

[5] Yuanpeng W, Hongxu J, Huirong Y. A multi-mode high-speed video data capture system based on DSP+FPGA[C]// Multimedia Technology (ICMT), 2011 International Conference on. Hangzhou: IEEE, 2011: 5230-5233.

[6] 纪娜. 多线程存储 YUV 数据在安防系统中的应用[J]. 信息技术,2014,38(7):167-170.

[7] 吴江. 基于移动预测的水下传感器网络可扩展定位方法[J]. 微电子学与计算机,2014,31(12):157-160.

作者简介:

尹雷 男,(1992-),硕士.研究方向为嵌入式系统、图像处理. E-mail:510693455@qq.com.

卿粼波 男,(1982-),博士,副教授.研究方向为图像处理、数字通信.

滕奇志 女,(1961-),博士,博士生导师.研究方向为计算机应用与图像识别.

付雄 男,(1989-),硕士.研究方向为嵌入式系统、图像处理.