

基于 HDLC 协议的图像实时压缩传输技术研究

张修建¹, 梁伟伟¹, 王兵¹, 赵茜², 靳硕¹

(1. 北京航天计量测试技术研究所, 北京 100076; 2. 华北计算技术研究所, 北京 100083)

摘要: 针对航天某领域视频监控需求, 解决在图像传输过程中误码率较高, 传输链路可靠性较低的问题, 提出了一种基于 HDLC 协议的图像实时压缩传输系统; 在给出系统体系结构的基础上, 对视频压缩算法、码流控制和环形缓冲区算法进行了研究分析, 介绍了图像实时压缩传输技术的设计和软件实现方法, 采用多线程技术设计了两种视频监控模式, 采取编码器结构优化和压缩算法优化等技术途径, 保证图像压缩的实时性和正确性, 实现了多路视频的单路切换和多路并行两种模式下的视频监控技术; 该系统保证了码流的平稳发送并按照 HDLC 协议传输至测试台控制单元中, 连续运行 5 小时内无图像丢帧, 并能够准确地表达剧烈突变的运动, 性能可靠, 在应用中取得了良好的效益。

关键词: 实时压缩; H. 264; HDLC 协议; 多路并行

Research on Image Real-time Compression and Transmission Technology Based on HDLC Protocol

Zhang Xiujian¹, Liang Weiwei¹, Wang Bing¹, Zhao Xi², Jin Shuo¹

(1. Beijing Aerospace Institute for Metrology and Measurement Technology, Beijing 100076, China;
2. North China Institute of Computing Technology, Beijing 100083, China)

Abstract: According to the video surveillance requirement in a certain field of spaceflight, solve problems in the process of image transmission with high bit error rate, low transmission link reliability, a real-time image compression and transmission system based on HDLC protocol is proposed. Based on the proposed system framework, the image compression algorithm, stream control and ring buffer algorithm are studied and analyzed. The paper introduces the design and software implementation of the real-time image compression transmission technology. To ensure the real-time image compression and correctness, two kinds of video monitoring mode design using multi-thread technology, adopt the encoder structure optimization and compression algorithm optimization approach, has realized the multi-channel video single path switching and multiplex parallel video monitoring technology in the two modes. The system ensures the steady transmission of the bit stream and is transmitted to the center controller in accordance with the HDLC protocol, which has been running continuously for 5 hours without image missing frame, and can accurately express the dramatic changes in the movement. The system is reliable in performance and has achieved good results in applications.

Keywords: real-time compression; H. 264; HDLC Protocol; multiplex parallel

0 引言

视频泛指将一系列静态影像以电信号方式加以捕捉、纪录、处理、储存、传送与重现的各种技术, 随着通信技术的不断发展和网络带宽的快速增长, 视频不断应用于各种传统及新兴行业中^[1]。而视频信息自身具备的直观性和易被视觉接受的特点, 使其不断引入航天、航空、无人机和机器人等需要实时监控的行业中, 方便现场环境和作业对象的实时远程监控。

图像传输具有数据量大, 实时性要求高的特点, 尤其在无线传输中更容易受到外界环境的干扰, 为了降低传输过程中的误码率, 提高传输链路的可靠性, 设计了一种基于 HDLC 协议的传输链路。高速通信协议 HDLC 是一个在同步网上传输数据、面向比特的数据链路控制协议。它具有差错检测功能强大、实时和同步传输、可靠性高等特点^[2]。

收稿日期: 2017-07-03; 修回日期: 2017-08-01。

基金项目: 中国航天科技集团公司计量研究项目; 中国运载火箭技术研究院技术创新基金课题。

作者简介: 张修建(1984-), 男, 山东济宁人, 硕士, 工程师, 主要从事图像压缩编码和测量技术等方向的研究。

为了在图像传输过程中降低误码率, 并且保证其实时性, 本文着重设计了符合项目要求的编码模式、压缩算法和发送协议, 实现了视频压缩装置的开发和实现。

1 视频压缩传输装置原理

视频压缩传输装置的原理如图 1 中所示, 主要分为硬件设计和软件设计。软件是视频压缩装置实现实时压缩和码流平稳传输的核心, 在支持视频采集的硬件电路上, 设计了视频采集、H. 264 编码、码流封装和 HDLC 发送等模块, 各功能模块在 DSP/BIOS 调度机制的作用下统一协作, 共同实现视频的实时压缩和码流的平稳传输功能。

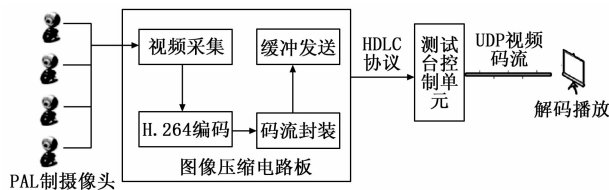


图 1 视频压缩装置原理框图

视频压缩装置的硬件是基于数字信号处理器 (DSP) 和可

编程阵列 (FPGA) 进行设计的, 图 2 为视频压缩装置硬件结构图。DSP 主要完成复合视频解码器的初始化、视频参数设置和视频的实时压缩; FPGA 主要完成时序控制、通信协议实现和码流等数据的传输; 整机工作时可同时最多接入 4 路视频信号的采集。

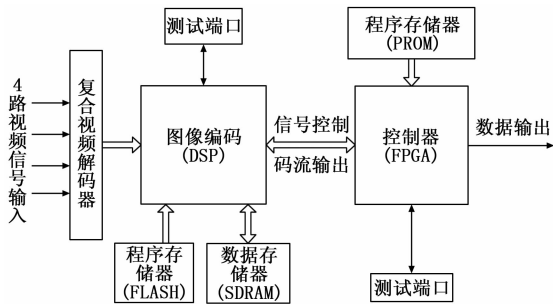


图 2 视频压缩传输装置硬件结构图

2 软件设计

软件设计是整个装置实现视频实时压缩和码流 HDLC 传输的核心, 采用 C 语言和汇编语言设计完成, 主要运行在 DSP 内, 实现视频采集、H. 264 编码、码流封装和 HDLC 发送等功能, 在 FPGA 的进一步时序配合下, 完成数据到下级控制器的传输。

2.1 视频采集设计

参照图 1 可以看出, 视频压缩装置接收摄像头输出的 PAL 制模拟视频信号, 在 DSP 内的视频采集功能模块的控制下, 调用复合视频解码器将视频信号解码成复合 ITU-R 656 格式的视频信号; 同时通过预置参数的作用, 保证捕获到的图像特征如大小、帧场模式等, 符合后端压缩编码器的输入图像信号要求。

在视频采集的设计上, 采用实时操作系统 DSP/BIOS 提供的 FVID 类驱动模型实现。FVID 模型主要由 FVID_create、FVID_control、FVID_alloc、FVID_exchange 和 FVID_delete 函数构成, 提供模块化的读写应用函数, 可以更加友好、高效地实现视频抓取和视频存储空间的分配, 通过简单的应用程序调用, 即可轻松实现视频的循环采集。图 3 显示的是单帧图像的采集过程。

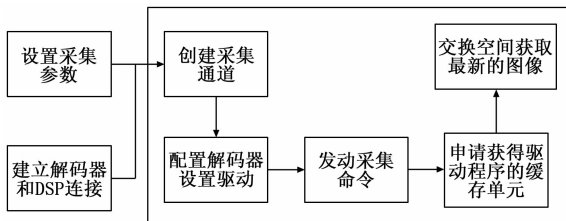


图 3 单帧图像的采集原理

采集参数设置是通过参考并设计 FVID 类驱动模型中的 FVID_frame 和 VPORTCAP_Params 结构体实现的, FVID_frame 结构体中主要包含的是 YUV 图像信号格式的亮度和色度信息, VPORTCAP_Params 结构体包含了捕获的视频信号的详细信息, 如行数、列数、YUV 结构、场频、捕获模式设置、水平视频数据是否放缩、水平色度数据滤波、奇偶场列

的开始和结束位置、FIFO 的门限值等信息。设备驱动根据配置参数计算缓冲区的大小并分配缓冲区, 且至少要为每个 FVID 通道分配 3 个缓冲区, 以确保视频图像的捕获和显示的连续性。视频捕获前, 须正确设置 VPORTCAP_Params 结构体中的相关参数, 以保证获取符合任务要求的视频信号。

图像采集的具体过程由 FVID 模型中的相关函数实现。其中, FVID_create 函数建立一个 FVID 通道对象并初始化, FVID_control 函数发送控制命令配置复合解码器和采集视频数据, FVID_alloc 函数申请采集空间, FVID_exchange 函数通过交换空间采集新的一帧图像。

2.2 编码器结构及压缩算法设计

视频压缩装置的编码器采用 H. 264 标准设计, 作为新一代压缩标准的 H. 264, 相比以前的标准 (如 MPEG-4、H. 263 等), 在保证相同图像质量的前提下, 可以节约 50% 的码率, 而且对网络传输表现出更好的适应性^[3], 是当下主流的视频编码标准。

H. 264 采用混合编码技术, 在整数 DCT 变换、可变大小的多模式宏块、环路滤波、运动补偿、高效的运动估计算法等方面引入新技术, 通过每个模块的小的技术改进, 带来了系统整体性能的提升, 使运动图像压缩技术上升到了一个更高的阶段。本本采取编码器结构优化和压缩算法优化等技术途径, 保证压缩的实时性和正确性。

2.2.1 编码器结构设计

视频压缩装置的编码器采用混合编码框架设计, 提供了输入图像的宽高、关键帧的间隔、码率和帧率等控制参数, 编码器参数结构定义如下:

```
typedef struct
{
    unsigned int  nWidth; // 图象的宽
    unsigned int  nHeight; // 图象的高
    unsigned int  nIFrame; // 关键帧的间隔
    int  bDeblockingFilter; // 环路滤波的标志
    unsigned int  nQualityCtrlMode; // 质量模式
    unsigned int  nQP; // 量化参数的设置 0-51
    unsigned int  nFPS; // 帧率 (frame/s)
    unsigned int  nBitRate; // 码率 (bit/s)
    unsigned int  nFlag; // 5bit 编码标志位
} YTH264PARAM;
```

编码器中的图像以 16 * 16 像素大小的宏块为单位进行处理, 图 4 描述了对一个宏块进行编码的设计流程, 首先从采集机构输出的视频图像中取一个待编码宏块 F_n , 该宏块以帧内或帧间的模式进行编码, 生成一个预测宏块 P 。

在帧内编码模式下, P 由当前图像中已经编码、解码、重构且未进行去块滤波的宏块 $\mu F'_n$ 经帧内预测得到; 在帧间编码模式下, P 由参考图像进行运动补偿预测得到^[4]。当前宏块 F_n 减去预测宏块 P , 得到残差块 D_n , 对残差块 D_n 进行整数变换、量化后得到一组系数 X , 对系数 X 再进行熵编码, 就完成了宏块的编码过程。经过熵编码的码流加上宏块解码所需的一些信息, 如预测模式、量化步长、描述宏块运动预测补偿的运动矢量信息等, 这就组成了宏块压缩后的码流。

2.2.2 压缩算法设计

视频压缩装置的多处包含决定编码功能和效率的复杂算

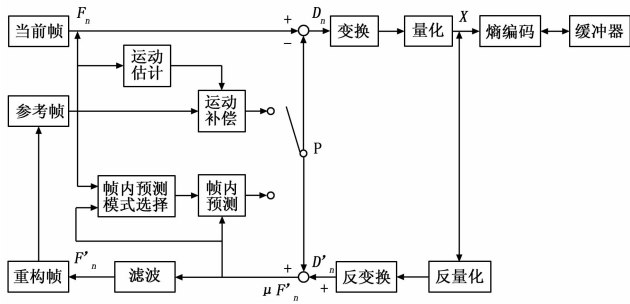


图 4 视频编码器结构图

法。本文只对关键算法进行简单介绍。帧内预测用来清除空间冗余，帧间预测用来清除时间冗余，变换和量化用来消除频率冗余，环路滤波用于消除量化噪声。

帧内预测算法充分利用图像的空间相关性，利用图像块周围的像素来进行帧内预测，提高了编码效率，从而减少编码输出的码率。H. 264 视频编码标准在帧间预测模式中增加了很多关键技术来提升压缩性能，在此不再赘述。

为了进一步节省图像传输码率，需要对图像信号进行压缩，一般采用变换编码来消除图像信号中的相关性及减少图像编码的动态范围。变换编码将图像时域信号通过数学工具变换成频域信号，在频域中，图像信号能量重新分布，且能量大多集中在低频区，相对时域信号，码率会有较大下降。根据统计和经验得知，平坦区域或内容缓慢变化区域占据一幅图像的绝大部分，而细节区域或内容突变区域则占小部分。这样，空间域的图像变换到频域，即所谓的变换域，会产出相关性很小的一些变换系数，并可对其进行压缩编码。视频编码中变换编码的数据源，包括两种：一是图像数据本身，二是图像残差。H. 264 把运动估值和帧内预测的残差结果从时域变换到频域，使用了类似于 4x4 离散余弦变换 DCT (Discrete Cosine Transform) 的整数变换，而不是采用 8x8 DCT 的浮点数变换。这种方法的优点在于：在编码器和解码器中允许精度相同的变换和反变换，便于使用简单的定点运算方式。在 H. 264 中，同时采用整数运算，提高图像压缩的实时性^[5]。

量化编码是把 DCT 系数除以一个常量，经过量化后的结果是量化步长的整数倍或为更多的零值，从而达到了压缩的目的。在反量化时，由于量化过程取整或四舍五入，而无法完全恢复原 DCT 系数，从而产生一定的失真。量化的原理公式如公式 1 所示，反量化的原理公式如公式 2 所示。由于一般的可编程平台没有提供专用的除法指令，所以视频编码标准在量化与反量化的实际应用过程中，采用了查表或移位的方式来实现，量化步长越小，等级越多，则码流大小等级越丰富。

$$q(x, y) = \text{round}(\frac{F(x, y)}{Q} + 0.5) \quad \text{公式 1}$$

$$F(x, y) = q(x, y) * Q \quad \text{公式 2}$$

另外，H. 264 中的 Deblocking Filter 还能够根据图像内容做出判断，对由于块效应产生的像素值跳变进行平滑，对每个 4x4 块边界都要进行滤波操作，以此达到平滑滤波边界的目的。这样可以消除经过反量化和变换后，重建图像中由于预测误差产生的块效应，即块边缘处的像素值跳变，从而一来改善图像的主观质量，二来减少预测误差。

2.2.3 码率控制设计

码率控制算法一般是通过调整 DCT 变换系数的量化参数，来实现达到控制输出码流速率的目的。在本文设计的 H. 264 视频编码中，我们的码率控制算法主要在 GOP 层、Frame 层以及基本单元层 3 个层次上进行。

GOP 层码率控制的主要任务是根据当前的比特率、帧率、缓存区充满度、传输速度以及 GOP 帧数、帧类型来计算每个 GOP 的目标比特数，并且得到各个 GOP 的初始量化参数。其中当前 GOP 目标比特根据网络带宽，帧频以及 GOP 中帧的数目分配。

Frame 层控制的主要任务是根据已编码帧实际消耗的比特数和 GOP 层计算得到的目标比特数，将前两者比特数的差值反应到当前帧的目标比特数，对当前帧的目标比特数进行调整，以便调整该帧的量化参数。具体可分为两步：第一步是考虑缓存容量限制，第二步考虑了网络带宽、帧频、GOP 中剩余可分配的比特等限制。

基本单元一般是指一帧或者一个宏块。若基本单元层是一帧，那么该层码率控制的主要任务就是根据当前帧的目标比特数对该帧的量化参数进行调整。若基本单元层是宏块，那么该层码率控制的主要任务就是在该帧目标比特数确定的前提下，根据每个宏块的图像复杂度，进一步调整该宏块的量化参数。同时，为了满足视觉的平滑性，要保证该基本单元层的量化参数和前一个基本单元层量化参数的差值不能太大。一般来说，选择的基本单元越大，获得的 PSNR 越大，但是比特输出的波动也会变大。反之，选择的基本单元越小，比特输出相对平稳，但是获得的 PSNR 将会变小。

3 软件实现

本软件采用多任务系统，采用 DSP/BIOS 线程调度机制进行任务的管理^[6]，设计多个线程同时运行，如图 5 所示，每个线程有独立的指令流，完成一个独立的功能模块，允许优先级高的线程抢占优先级低的线程，线程间可通信和同步处理，通过环形缓冲区进行数据的访问和存取。

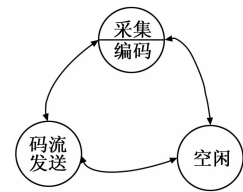


图 5 多线程运行原理图

3.1 视频采集与编码线程设计

在此线程中，基于图 4 的视频采集原理，设计了可同时采集 4 路输入视频的采集架构。基于产品化设计要求，对于每一路视频进行灵活控制，可直接压缩编码，也可以根据 CPU 分配的时间片，在特定的时间片内进行压缩编码。根据使用需求，设计了两种视频压缩模式：单路切换监控模式和 4 路并行监控模式。在实际工作应用场景中，可灵活配置视频压缩装置的视频监控路数和单路的视频监控时间。

送入编码器的视频经压缩处理后形成 H. 264 码流，在标准的 H. 264 码流头部加入特殊的识别标志、序列号、码流大小等关键信息进行封装，形成具有独特识别标识的码流，保证码流在传输过程中的安全，方便解码软件的分析 and 播放。

H. 264 码流暂存环形缓冲区中, 等待后续的发送和 HDLC 传输。

3.2 码流发送线程设计

当 DSP/BIOS 运行发送线程时, 暂存环形缓冲区中的 H. 264 码流按包发送并进行 HDLC 传输, 单包的数据量过大, 积累的码流数据量过多, 解码出来的图像卡顿; 单包的数据量过小, 每次接收到的码流数据量少, 会导致解码的图像花屏, 因此, 在理想情况下, 每帧图像压缩形成的码流, 需要及时地传输到解码端进行解码播放, 才能保证视频播放的连续性。

为了 HDLC 同步传输的连续性, 在不发送 H. 264 码流的空闲时间段内, 循环发送 4 个字节的无效数据, 提高了视频压缩装置的整体数据发送速率, 并把无效数据重复有效数据的概率降低到 2×10^{-10} 数量级, 避免了有效数据在解码端的错误丢弃, 保证了视频的质量和流畅播放。

3.3 环形缓冲区设计

在主程序中, 视频编码线程与码流发送是两个独立的线程, 它们之间的数据共享是通过环形缓冲区来完成的。由于是长时间运行, 通常对内存的使用有很高的要求, 分配频繁, 释放内存容易产生内存碎片, 必然会影响软件长期运行的稳定性, 在这种情况下, 使用像动态环形队列这种内存性能比较好的数据结构就非常合适^[7]。用环形队列的缓冲区, 解决多个线程同时读写缓冲区的问题, 同时根据数据的重要性可选择添加互斥访问保护机制。一般来讲, 队列 FIFO, 从底部添加数据, 从顶部弹出数据。添加新元素的那一端被称作队尾 nTail, 而弹出元素的那一端被称作队首 nHead。环形队列的特点是: 不需要进行动态的内存释放和分配, 使用固定大小的内存空间反复使用。在实际的队列插入和弹出操作中, 是不断交叉进行的, 当 push 操作时, tail 会增加, 而当 pop 操作时, head 会增加。

1) 如果 push 的速度快的时候, 有可能追上 head, 这个时候说明队列已经满了, 不能再进行 push 的操作了, 需要等待 pop 操作腾出队列的空间。

2) 如果 pop 的操作快的时候, 使得 head 追上 tail, 这个时候说明队列已空了, 不能再进行 pop 操作了, 需要等待

.....

(上接第 202 页)

参考文献:

[1] Tian B, Yao Q, Gu Y, et al. Video processing techniques for traffic flow monitoring: A survey [A]. IEEE International Conference on Intelligent Transportation Systems [C]. 2011, 1103 - 1108.

[2] Perkasa O, Widyantoro D H. Video-based system development for automatic traffic monitoring [A]. IEEE International Conference on Electrical Engineering & Computer Science [C]. 2014, 240 - 244.

[3] 张 玲, 陈丽敏, 何 伟, 等. 基于视频的改进帧差法在车流量检测中的应用 [J]. 重庆大学学报: 自然科学版, 2004, 27 (5): 31 - 33.

[4] 张 咏, 李太君, 李枚芳. 利用改进的背景差法进行运动目标检测 [J]. 现代电子技术, 2012, 35 (8): 74 - 77.

[5] 杨国亮, 王志良, 牟世堂, 等. 一种改进的光流算法 [J]. 计算机工程, 2006, 32 (15): 187 - 188.

[6] Han X, Gao Y, Lu Z, et al. Research on moving object detection algorithm based on improved three frame difference method and optical flow [A]. IEEE International Conference on Instrumentation

push 进来数据。

4 结论

本文以该技术背景为前提给合在航天领域视频监控的应用, 探索并实现了基于 HDLC 协议的图像实时压缩传输系统设计。视频采集与编码线程和码流发送线程之间的数据共享通过对环形缓冲区来完成的, 视频采集与编码线程输出码流到环形缓冲区中, 在 DSP/BIOS 线程调度机制下, 码流发送线程从缓冲区中取出一定字节大小的码流进行发送, 按照 HDLC 传输协议送至通信链路中, 然后通过 UDP 协议送入主机进行解码播放。

在实际测试环境下, 输入视频的分辨率为 352×288 , 视频压缩传输装置以 25 帧/s 的实时速率压缩编码, 以 512kbps 的码率宽带输出码流, 经过中转后送入主机实时解码播放。通过对帧头时间标签的计算得出压缩与处理环节的平均总延时为 170ms, 4 路视频的平均切换时间为 65ms; 采用 ITU-T 提供的测试序列, 计算得出重建图像序列平均峰值信噪比 PSNR 值为 37.83dB; 连续运行 5 小时内无图像丢帧, 并能够准确的表达剧烈突变的运动, 达到了预期的设计目的和控制效果。

参考文献:

[1] 张纪宽, 彭 力, 陈志勇. 动态复杂背景下的智能视频监控系统设计及实现 [J]. 计算机测量与控制, 2016, 24 (7): 100 - 104.

[2] 张文博, 左 勇, 徐浩然. 支持 HDLC 协议的紫外通信系统设计 [J]. 光通信技术, 2012, 36 (9): 53 - 56.

[3] 金艳梅. 高速视频会议系统 H. 264 编码的优化 [J]. 兰州理工大学学报, 2015, 41 (3): 108 - 111.

[4] 柳 春, 廉东本. H. 264 编解码算法在视频会议中的应用 [J]. 计算机系统应用, 2015, 24 (4): 144 - 147.

[5] 安向阳, 沈庭芝. 基于 DSP TMS320DM642 的 H. 264 视频编码的实现和优化 [J]. 微计算机信息, 2005 (20): 128 - 130.

[6] 彭启琼, 管 庆. DSP 集成开发环境——CCS 及 DSP/BIOS 的原理与应用 [M]. 北京: 电子工业出版社, 2004.

[7] 詹 英, 吴春明, 王宝军. 基于 RCSW 的数据流速度异常检测算法研究 [J]. 电子学报, 2012, 40 (4): 674 - 680.

[7] Abdullah T, Anjum A, Tariq M F, et al. Traffic monitoring using video analytics in clouds [A]. IEEE/ACM 7th International Conference on Utility and Cloud Computing [C]. 2014: 39 - 48.

[8] Anjum A, Abdullah T, Tariq M, et al. Video stream analysis in clouds: an object detection and classification framework for high performance video Analytics [J]. IEEE Transactions on Cloud Computing, 2016 (99): 1 - 14.

[9] Dean J, Ghemawat S. Simplified data processing on large clusters [J]. Proceedings of Operating Systems Design and Implementation (OSDI), 2004, 51 (1): 107 - 113.

[10] Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing [A]. Usenix Conference on Networked Systems Design and Implementation [C]. USENIX Association, 2012.

[11] Yang S, Wu B. Large scale video data analysis based on spark [A]. IEEE International Conference on Cloud Computing and Big Data [C]. 2015: 209 - 212.