

# H.264 编解码算法在视频会议中的应用<sup>①</sup>

柳春<sup>1,2</sup>, 廉东本<sup>2</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110168)

**摘要:** 视频压缩是视频会议中的核心技术, 压缩技术的好坏决定着视频会议质量的高低. H.264 是一种高性能的编解码算法, 具有较高的压缩比和较好的网络适配性. 对 H.264 编解码算法中的关键技术进行了介绍, 同时介绍了 Directshow 技术框架. 并且, 通过将开源的 H.264 编解码库 FFmpeg 嵌入到 Directshow 链路中, 实现对用户视频数据的编码压缩及解码播放.

**关键词:** 视频会议; H.264; FFmpeg; 视频编解码; Directshow

## Application of the H.264 En/Decoding Algorithm in Video Conference System

LIU Chun<sup>1,2</sup>, LIAN Dong-Ben<sup>2</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

**Abstract:** Video compression is the core technology for video conference, and its quality depends on the compression technology. The H.264 is a high-performance en/decoding algorithm, which has higher compression ratio and better network suitability. This paper introduces the key technology of the H.264 coding algorithm and the framework of Directshow technology. Besides, through embedding the FFmpeg, the open-source library of H.264, into the Directshow Filter Graph, researchers can encode and decode the video data of users.

**Key words:** video conference system; H.264; FFmpeg; en/decoding algorithm; directshow technology

随着互联网技术和多媒体技术的快速发展, 在互联网上进行单一的语音和文字传输已经远远达不到用户需求. 在用户与用户的交流中更希望加入实时图像, 在这种需求下, 视频会议系统顺势而生. 视频会议系统中整合了文字, 语音及图像等多种多媒体信息, 同时还具有很好的实时性, 使得用户间的交流更加亲切更加接近于面对面的真实体验.

视频会议进行中需要大量的信息在网络中传输, 这些数据会占用大量的带宽, 因此必须采用一定的压缩算法. 对于选取的压缩算法而言, 必须能在降低带宽的同时保证视频会议的质量. 压缩效果的好坏直接影响到视频会议的质量, 因而压缩技术被视为视频会议的关键技术之一.

H.264 由 ITU-T 和 ISO 两个组织联合组建发布, 是一套数字视频编解码标准, 作为 ITU-T 的 H.264 部分,

也是 ISO MPEG-4 标准的第 10 部分<sup>[1]</sup>. H.264 具有更高的编码效率与 H.263 相比能够节约大约 50% 的码率<sup>[2]</sup>. 大量应用证明 H.264 是一个高性能的编解码技术, 在低码率环境中依旧能够提供高质的视频画面. 由于 H.264 的高网络适应性, 它可以工作在如视频会议等的实时通信模式中, 也可工作在视频存储和转码的流媒体服务器中. 同时, H.264 具有较高的复杂度, 性能的提高是通过增加复杂度来实现的. 相比于性能的提高, 在复杂度方面是可以接受的, 现已成为视频会议的中值得信赖的编解码技术.

## 1 H.264 标准介绍

### 1.1 H.264 关键技术

#### (1) 分层技术

H.264 的功能分为两层, 即视频编码层(VCL, Video

① 收稿时间:2014-07-23;收到修改稿时间:2014-08-19

Coding Layer)和网络提取层(NAL, Network Abstraction Layer). VCL 数据即编码处理的输出,它表示被压缩编码后的视频数据序列.在 VCL 数据传输或存储之前,这些已编码的 VCL 数据,先被映射或封装进 NAL 单元中. NAL 负责以网络传输所要求的恰当的方式对数据进行打包和传送.

单个 NAL 单元包括: NAL 头信息、原始字节序列负荷(RBSP). NAL 头信息中需要有字段用来标示当前的 NAL 可否丢弃,以及丢弃后是否会引起错误的扩散. NAL 头的最后,含有字节来说明有效载荷的内容. NAL 的解码需要按序执行,因此在解码端必须实现相应机制在 NAL 乱序时将其恢复原序. NAL 单元可直接作为 RTP 的有效载荷. NAL 单元序列的结构如下:



图 1 NAL 单元序列结构图

(2) 帧内预测编码技术<sup>[3]</sup>

帧内预测是基于频域的预测技术,当前块的一些参数可以通过可以通过相邻块来预测.因为图像相邻块之间有很强的相关性,这种相关性使得在变换后它们的差别不是很大,所以可以通过上方和左方的块来预测当前块.主要分为: 4X4 亮度预测模式、8X8 色度预测模式:

a. 4X4 亮度帧内预测模式

在此模式中 H.264 将一个宏块划分成 16 个 4X4 的子块,然后以每个 4X4 子块为单位进行预测.共有 9 种预测模式.如图 2 所示,共 8 个方向的预测模式(0, 1, 3, 4, 5, 6, 7, 8),模式 2 为直流模式(DC 模式).

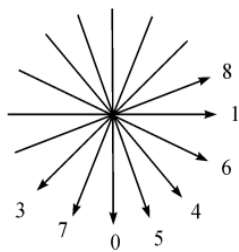


图 2 4X4 帧内

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

图 3 4X4 帧内预测

8 种预测模式

假设待预测的像素块 a-p, 已重建的像素块 A-M, 如图 3. 待预测的像素块根据它上方和左侧相邻的像

素块并选用 9 种模式中的一种来计算得到.

b. 8X8 色度预测模式<sup>[4]</sup>

对于 YUV420 的视频序列而言,与一个 16X16 的亮度宏块对应的色度块大小是 8X8. 因为人眼对色度的感知敏感度弱于对亮度的感知敏感度,所以色度的信息的重要性相比于亮度较差,可以采用较大粒度的方法来对色度信息进行预测. H.264 指出,可以使用 8X8 的预测模式来预测帧内色度信息, U 分量和 V 分量结合起来统一进行预测. 如图 4.

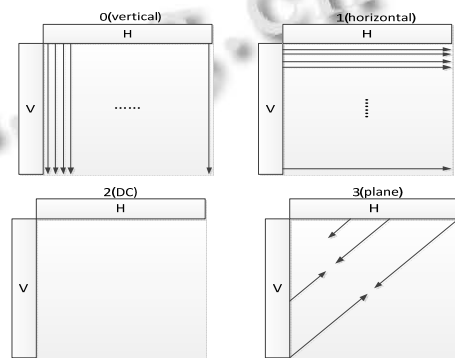


图 4 16X16 预测模式

(3) 帧间预测编码

帧间预测是指通过当前帧之前的一或多帧来预测估计当前帧.采用多个先前帧的作用在于提高运动补偿和运动估计精度及效率.对于一个像素块而言,通过搜索其之前已编好码的一帧或者多帧图像中的相关像素块,以得到其运动矢量.帧间预测的作用是降低图像在时域上的相关性,采用多帧的参考以及运动矢量可以用少量的数据来表示当前帧,减少在网络中的传输量.

1.2 H.264 编解码技术

图 5 给出 H.264 编码结构示意图,图中显示编码所需部件及流程.在开始编码当前帧时,需要对当前帧进行变换、量化、重排、熵编码,最后封装成 NAL 单元.一个视频帧的编码需要经历如上的几个阶段.除此之外,还有帧内预测和帧间预测两种模式的选择.在选择帧间预测时,会用到运动估计和运动补偿参数,而这两个参数的是通过当前帧和参考帧计算出来的;同时参考帧来自于重构帧,重构是当前帧在变换、量化后再进行逆量化和反变换得到的,类似于反编码;对于帧内预测而言不需要运动估计和运动补偿.对于视频帧的解码,是与编码相逆的过程.

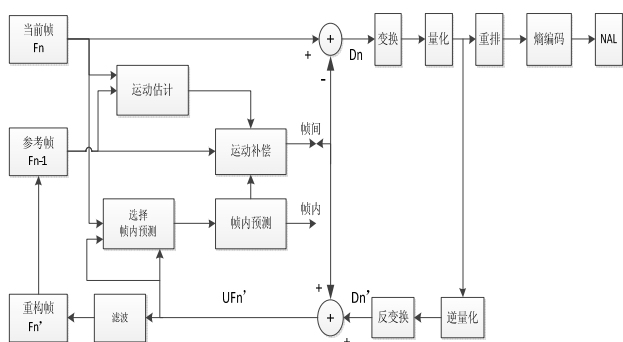


图 5 H.264 编码器结构图

## 2 视频会议中视频处理设计

视频会议中需要对用户视频信息的采集, 编码以及发送, 采用成熟的 Directshow 框架可以完成以上功能. Directshow 提供调用 Windows 平台下媒体设备的 API, 同时还可以在所提供的接口和类的基础上集成开发符合自己个性需求的组件, 使项目开发起来更加高速便捷.

### 2.1 DirectShow 技术介绍

Directshow 是由 Microsoft 公司开发, 运行于 Windows 平台之上的流媒体开发体系框架. Directshow 是 DirectX SDK 成员之一, 同样也是完全基于 COM 组件技术. DirectX SDK 是 Microsoft 公司提供的一套在 Wdindows 操作平台上开发高性能图形、声音、输入、输出和网络游戏的编程接口<sup>[5]</sup>. Directshow 中将参与数据处理的各个组件称作 Filter(过滤器), 多个组件连接起来形成一个链路 Filter Graph(过滤器图). 媒体数据的处理和管理由 Filter Graph 完成. Filter 分为 3 类: Source Filter(源过滤器), Transform Filter(转换过滤器), Render Filter(提交过滤器). Directshow 系统结构图如下图 6 所示.

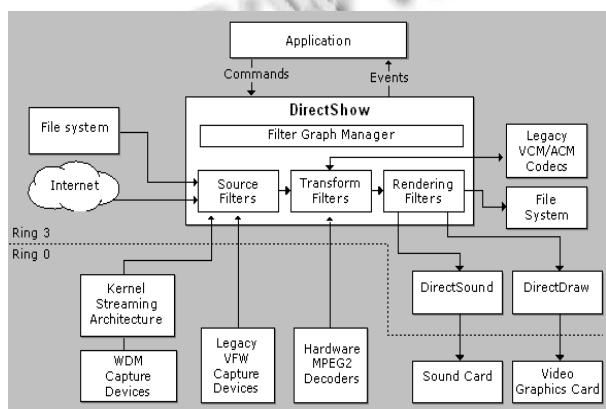


图 6 Directshow 系统架构

SourceFilters: 获取数据; 此数据可以是文件、网络上或者计算机里的摄像头、视频采集卡等; 然后将数据传输给下游 Filter; TransformFilters: 数据的格式转换、传输; RenderingFilters: 数据的最终去向(展现), 可以给声卡、显卡等多媒体设备进行演示, 也可以文件形式存储.

### 2.2 视频处理架构

在视频会议中, 需要将用户的视频信息进行捕获, 然后进行编码. 将编码后的视频信息通过 RTP 模块打包封装成 RTP 包, 通过网络发送给服务器. 同时, 在视频会议中每个用户端还需要对网络中传送进来的 RTP 包进行接收解析得到视频数据, 再将视频数据解码, 最后实时播放展现给用户. 如图 7 所示.

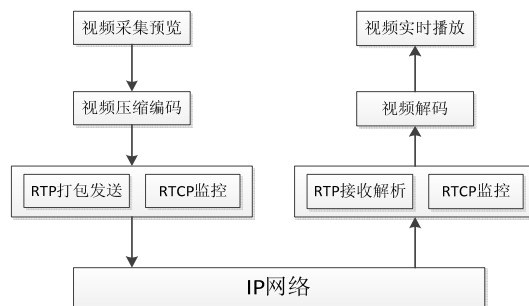


图 7 视频处理结构图

## 3 H.264在视频会议中的嵌入

### 3.1 视频发送接收链路

如图 8 所示. 我们可以看到有 4 个 Filter, 分别负责捕获、编码, RTP 打包发送及视频的预览. 用户的视频信息通过摄像头被获取, 传送给 Code Filter; Code Filter 将视频数据编码成 H.264 数据流, 然后将 H.264 数据流传递给 RTP Filter; RTP Filter 将 H.264 数据流打包成 RTP 数据包, 然后通过网络发送出去. 摄像头的另一路连接 Video Renderer 进行捕获时的预览.

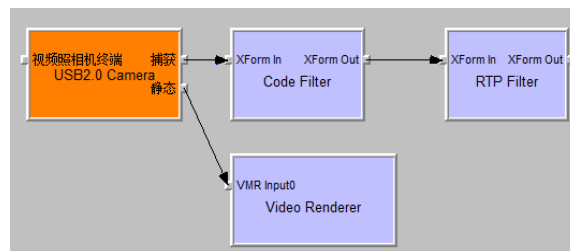


图 8 视频发送链路图

对于视频信息的接收而言, 它的过程正好与发送

相逆。RTP Filter 接收来自网络的 RTP 数据包,在解析后得到 H.264 数据流并将数据传给 Code Filter,此时的 Code Filter 解码功能将 H.264 数据解码成视频帧;视频帧在 Video Renderer 进行播放。

### 3.2 嵌入 H.264 编解码算法

在视频信息的发送和接收过程中皆需要图 8 中所示的 Code Filter,而 Code Filter 的编解码算法是 H.264 编解码算法。在本系统我们使用开源的库 FFmpeg<sup>[6,7]</sup>,FFmpeg 为媒体流的编码压缩、解码播放、媒体流转换和媒体流录制提供了一种完美的解决方案<sup>[8]</sup>。FFmpeg 中包含 libavcodec,用来进行音视频编解码的库。除此之外还包括 libavformat,用来对音视频格式化,分离或整合字幕、音频、视频的库;libavutil 为其他库提供帮助的辅助库;libswscale 用来将视频图像进行缩放和视频图像色彩/像素格式进行转换的库。

FFmpeg 编解码流程如图 9,10 所示。

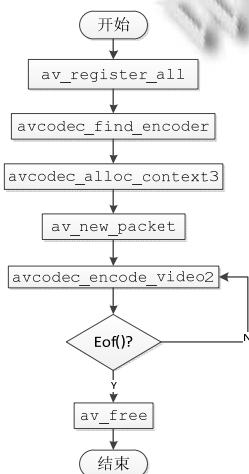


图 9 FFmpeg 编码流程图

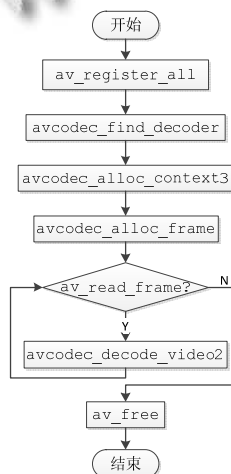


图 10 FFmpeg 解码流程图

由图 9,10 可以看到 FFmpeg 对视频进行编解码的过程。从开始的注册所有编解码器,然后通过参数如 CODEC\_ID\_H264 来找到相应的编码(解码)器和上下文(Context)。Context 中带有当前编码器的参数如码率、编码器类型及 ID,宽度和高度等。第四步分配一个 packet,用来存放编码后的视频数据包;在解码时需要分配一个 Frame 存放解码后的视频帧。在完成上述准备之后就可以对视频帧进行编解码。编解码结束后需要释放所有申请的空间(av\_free)。

### 3.3 实验效果

这里设计一个验证 FFmpeg 编解码成功与否的程

序。如图 11,通过 Directshow 技术获取视频并预览,左侧图像所示;右侧图像为解码后的显示图像。这里没有进行网络传输和接收的部分,将视频捕获存放本地文件,然后将本地文件进行解码播放。可以看到,视频预览编码存储及节目播放都已完全实现。在右侧播放解码的视频中,我们可以看到除一些细节处出现马赛克,整体画面效果与预览视频相差无几。这里还能够调整编码和解码时的参数设置来达到更好的效果,同时这也意味着需要更多的空间资源和时间资源,以及占用更多的带宽资源。



图 11 H.264 实验效果图

## 4 结语

本文介绍了 H.264 编解码中的关键技术,同时使用 Directshow 框架将 H.264 算法嵌入到视频会议客户端中。使用开源的 H.264 库-FFmpeg,并对 FFmpeg 编解码流程进行了分析。实验证明 H.264 可以很好地嵌入到视频会议中,完成视频会议客户端视频数据的获取,压缩,解码播放功能。

### 参考文献

- 1 刘杰.H.264 编解码算法在网络视频传输中的应用[学位论文].西安:西安电子科技大学,2010.
- 2 李超,柴文磊,刘劲松.高清视频会议系统技术浅析.信息安全与技术,2010,8:50-51.
- 3 陈靖,刘京,曹喜信.深入理解视频编解码技术.北京:北京航空航天大学出版社,2012.
- 4 Hu ZY, Peng JH, Wang XA. 2011 Int. Conf. in Electric, Communication and Automatic Control Proc. 2011. 739-748.
- 5 陆其明.DirectShow 实务精选.北京:科学出版社,2004.
- 6 Zerano FFmpeg. <http://ffmpeg.zerano.com>. [2014-7-17].
- 7 FFmpeg. <http://ffmpeg.org>. [2014-7-17].
- 8 李辉.基于 RTP 的 H.264 视频传输系统的设计与实现[学位论文].长春:吉林大学,2013.