

# Windows 平台下应用 FFMPEG 实现 H. 264 视频回放<sup>\*</sup>

刘马飞<sup>1</sup> 曾学文<sup>2</sup> 倪 宏<sup>2</sup>

(<sup>1</sup>中国科学院研究生院 北京 100049

<sup>2</sup>中国科学院声学研究所国家网络新媒体工程技术研究中心 北京 100190)

**摘要:** H. 264 是 ISO/IEC MPEG 和 ITU-T VCEG 联合制定的最新的视频编码标准,其压缩性能和图像质量大大优于先前的视频编码标准。然而,Windows 平台上的 Windows Media Player 需要第三方插件才支持 H. 264 格式视频的播放,很多插件如 ffdshow 等只支持 AVI 容器中的 H. 264 播放,不支持 ASF 容器中的 H. 264 视频播放。因此本文分析 Windows 平台的 DMO 等技术,利用开源程序 FFMPEG,实现在 Windows 平台下与 Windows Media Player 配合的支持 ASF 文件容器中 H. 264 视频回放的插件。

**关键词:** DMO FFMPEG H. 264 视频回放

## Building H. 264 Video Playback Use Ffmpeg on the Windows Platform

LIU Mafei<sup>1</sup>, ZENG Xuewen<sup>2</sup>, NI Hong<sup>2</sup>

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

<sup>2</sup> The National Network New Media Engineering Center of The institute of acoustic of Chinese Academy of Sciences, Beijing, 100190, China)

**Abstract:** H. 264 is a new video coding standard revised by ISO/IEC MPEG and ITU-T VCEG, whose encoding performance is far beyond former video encoding standard. But in windows platform, Windows Media Player needs an additional codec plug-in for playing the H. 264 video. There is some codec plug-in supporting the H. 264 video stream from AVI file container playback under Windows Media Player like ffdshow, but There are no free codec plug-in supporting the H. 264 video stream from ASF container playback. This paper will introduce how to build a codec plug-in with Windows Media Player supporting the H. 264 video stream from ASF container playback, using DMO technique and open source program named ffmpeg.

**Keywords:** DMO FFMPEG H. 264 video playback

H. 264/AVC<sup>[1]</sup> 是 ISO/IEC MPEG 和 ITU-T VCEG 联合制定的最新的视频编码标准。在同等还原图像质量条件下,与 H. 263+ 和 MPEG4 相比,H. 264 能够节省一半的码率。H. 264 正以其高压缩率,优秀的图像质量,低时延以及网络友好等特点,已经得到越来越广的应用。然而 Windows 平台下用户最熟悉的媒体播放器 Windows Media Player 本身并不支持 H. 264 编码格式的视频播放,而是需要另外安装 H. 264 的解码插件。Windows 平台下最普通的媒体文件容器格式莫过于 ASF<sup>[2]</sup> 和 AVI。由于 Windows Media Player 的框架特性,使得编解码插件对视频编码格式的支持,还与文件容器相关。目前能支持 AVI 容器中的 H. 264 编码格式视频播放的免费的插件有 ffdshow 和 CoreAVC 等,但能够配合 Windows Media Player 支持 ASF 容器中的 H. 264 编码格式视频的播放的免费插件则没有。ASF 容器格式是 Windows 平台下非常流行的一种适合流媒

本文于 2008-08-01 收到。

<sup>\*</sup> 基金项目:国家 863 计划。支持三重业务的地面电视无线回传关键技术与双向化方案研究(编号:2007AA01Z230)。

体传输的容器格式,因此编写支持 ASF容器中的 H. 264编码格式视频播放的媒体插件,对于应用来讲非常有必要。

在 Windows平台的流媒体系统中,由于 WindowsMedia Player与 Windows操作系统捆绑以及 Windows操作系统的普及程度,依赖 WindowsMedia Player作为媒体系统的客户端呈现给用户可以使得用户友好性得到增强,也可以节省开发新媒体播放器的成本,所以很多应用于 Windows平台下的流媒体系统都是利用 WindowsMedia Player作为媒体系统的终端。要使得以 WindowsMedia Player作为客户端的 ASF流媒体系统能够采用 H. 264编码视频格式,就必须要有支持 ASF容器中 H. 264视频播放的插件。通过探索,作者寻找到一种利用 FFMPEG,对其进行 DMO<sup>[3]</sup>封装实现配合 WindowsMedia Player支持 H. 264视频播放的插件的方法。

## 1 H. 264简介

H. 264是国际标准化组织 (ISO)和国际电信联盟 (ITU)共同提出的继 MPEG4之后的新一代数字视频压缩格式,它即保留了以往压缩技术的优点和精华又具有其他压缩技术无法比拟的许多优点。

(1) 低码流 (Low Bit Rate):和 MPEG2和 MPEG4 ASP等压缩技术相比,在同等图像质量下,采用 H. 264技术压缩后的数据量只有 MPEG2的 1/8,MPEG4的 1/3;显然,H. 264压缩技术的采用可以大大节省用户的下载时间和数据流量收费。

(2) 高质量的图像:H. 264能提供连续、流畅的高质量图像 (DVD质量);

(3) 容错能力强:H. 264提供了解决在不稳定网络环境下容易发生的丢包等错误的必要工具;

(4) 网络适应性强:H. 264提供了网络适应层 (Network Adaptation Layer),使得 H. 264的文件能容易地不同网络上传输 (例如互联网,CDMA,GPRS,WCDMA,CDMA2000等);

H. 264标准还在不断发展,SVC (scalable video coding)<sup>[4]</sup>和 MVC (multi-view coding)<sup>[5]</sup>添加作为 H. 264的修正的标准化工作正在进行中,因此 H. 264的功能将更加丰富,适用更多应用场景。我们有理由相信,H. 264标准将会受到业界的广泛应用,因此实现支持 H. 264播放的插件对 Windows平台流媒体应用非常有意义。

## 2 DMO介绍及应用

在 Windows媒体应用中,主要包括 DirectShow<sup>[3]</sup>和 DMO。DirectShow是微软公司提供的一套在 Windows平台上进行流媒体处理的开发包,与 DirectX开发包一起发布,为多媒体流的捕捉和回放提供了强有力的支持。

DirectX媒体对象 (DirectX Media Objects,简称 DMOs),是微软提供的另一种流数据处理 COM组件,与 DirectShow filter相比,DMO有以下优点:

(1) DMO比 filter实现的功能要少很多,这使得 DMO“体积”很小;

(2) DMO使用起来比 filter更有灵活性。DMO的使用不需要 filter graph,应用程序可以直接与 DMO交互。而 DMO也可以通过一个 DMO wrapper filter工作于 DirectShow环境。

(3) DMO总是同步处理数据,不像 filter有独立的数据传送线程,需要考虑多线程编程问题;

(4) 与传统的编解码管理器 ACM、VCM相比,用 DMO开发的编解码器是基于 COM的,更易于扩展。并且 DMO支持多个输入和多个输出;

(5) DMO不需要像 filter一样分配数据传送的内存,而有 DMO的使用者负责;

(6) DMO是一个独立功能模块,不需要像 filter一样连接成一条链路;

(7) DMO不需要像 filter一样将数据“推”下去,数据的输入输出都是由 DMO的使用者完成的;

所有这些优点,使得 DMO成为微软对于 Encoder和 Decoder开发的重点推荐模式。DirectX 9.0 SDK

中,微软更是把 DMO 从 DirectShow 中分离出来,而对于一些 transform filter,微软也推荐用 DMO 的方式来替换。因此,本文选择用 DMO 来封装 H.264 视频解码插件。

### 3 FFMPEG

FFMPEG是一套完整的录制、转换和流化音频和视频的开放源代码的解决方案,其包括以下几部分组成:

(1) libavcodec,一个包含了所有 FFmpeg 音视频编解码器的库。为了保证最优性能和高可复用性,大多数编解码器从头开发的。

(2) libavformat,一个包含了所有的普通音视频格式的解析器和产生器的库,该库里边包含解码 H.264 的 codec。

(3) ffmpeg,用 SDL 和 FFmpeg 库开发的一个简单的媒体播放器。

(4) ffmpeg,基于 HTTP 用于实时广播的多媒体服务器。

(5) ffmpeg,视频文件转换命令行工具,也支持经过实时电视卡抓取和编码成视频文件。

FFMPEG是基于 linux 平台上开发的,想要移植到 VC 平台上需要进行很大的改动,比较困难。幸好在 Windows 平台下可以利用 MinGW 编译 FFMPEG 代码,生成 avformat.dll、avcodec.dll、avutil.dll 三个动态链接库,可供应用程序使用。FFMPEG 的高性能设计,以及其包含的经过了优化的 H.264 解码算法,使得 FFMPEG 解码 H.264 视频的速度相当快;在 H.264 开源解码器评测中,FFMPEG 的解码速度最快。因此利用 FFMPEG 实现播放插件,解码速度快,占用内存小。

### 4 代码实现

代码实现的主要任务是将 FFMPEG 封装在 DMO 对象中,由 DMO 对象与 Windows Media Player 协商媒体类型、对象连接以及传递媒体数据,而 FFMPEG 负责将输入到 DMO 中的未解码的 H.264 视频帧解码成可以显示的 RGB 格式的视频帧,然后交由 DMO 输出端传予 Windows Media Player 进行显示。

在微软的 DirectX SDK 中,为开发 DMO 提供了一个 MediaObjectImp 模板类,使得开发 DMO 比较容易。实现程序定制的 DMO,只需派生此模板类,实现 17 个以 Internal 开头的函数。

#### 4.1 FFMPEG 库的使用

使用 FFMPEG 库,首先应该将 avcodec.dll、avformat.dll、

avutil.dll 加入到工程中,并且在文件头中需要包含 avcodec.h 头文件;

另外使用 FFMPEG 库的函数之前,需要对 FFMPEG 库进行初始化,因此本实现中是在 DMO 对象的构造函数中实现 FFMPEG 库的初始化。通常的操作是通过 avcodec\_init() 以及 avcodec\_register\_all() 初始化并注册 FFMPEG 所有的编码解码器;然后通过 CODEC\_D\_H264 指示码寻找到本 DMO 使用的 H.264 解码器,即:

```
codec = avcodec_find_decoder(CODEC_D_H264);
```

然后做编解码器上下文初始化及申请空间保存解码图像,即: c = avcodec\_alloc\_context();

```
picture = avcodec_alloc_frame();
```

最后是根据得到的 H.264 codec 以及解码上下文结构去打开 H.264 解码器,使其能够工作,即:

```
avcodec_open(c, codec);
```

具体代码可以参照 ffmpeg 源代码中的 output\_example.c 中的示例代码。

#### 4.2 InternalCheckInputType 函数

该函数的主要目的是检查 DMO 对象是否适合于此输入的媒体类型的解码,如果检查通过那么该 DMO 才可以用于此种类型视频的解码。本设计的 DMO 中实现的是 H.264 视频解码插件,因此 DMO 接收的输入

媒体主类型是视频,而媒体 subtype则是 H.264格式;所以只有当媒体类型符合此限制时,函数返回 TRUE,检查通过。

#### 4.3 InternalCheckOutputType函数

该函数的目的是检查 DMO对象是否能够产生此种媒体类型的视频输出,如果检查通过那么该 DMO可以用于产生此类视频输出。解码器的目的就是将压缩的格式解码,成为无压缩可以直接显示的格式;在这里,我们选择最普通的 RGB24格式。

#### 4.4 InternalProcessInput函数

该函数的作用是输入一个未解码的图像帧,需要做的工作就是把帧对象保存,以待应用程序在 DMO输出端请求图像输出时,对此输入图像帧执行解码操作,输出解码图像。因为图像帧在图像中是通过 C++的 COM对象 MediaBuffer来保存的,因此我们只需要 MediaBuffer的接口函数 GetBufferAndLength得到输入的图像存放的内存块指针和大小,然后增加 COM对象的引用计数,最后设置相关的时间戳即可。

#### 4.5 InternalProcessOutput函数

该函数的作用是将上次输入 DMO的未解码的帧调用 FMPEG中的解码函数进行解码,输入 YUV格式的视频,然而我们最终要求输出的视频格式是 RGB格式的,因此还必须将 YUV格式的视频转换成 RGB格式的视频。幸好的是 FMPEG中有高度优化的函数 YUV2RGB供我们使用。使用本 DMO的应用程序会向本函数请求输出解码图像,因此在这里我们调用 FMPEG中的解码函数对输入图像进行解码,把解码重建后的图像输出,交由应用程序进行播放。具体的代码如下:

```
len = avcodec_decode_video(c, picture, &got_picture, outbuf_ptr, size); //调用 FMPEG媒体库的解码函数进行解码
```

```
YUV2RGB(picture -> data[0], picture -> data[1], picture -> data[2], pOutData, pic_width, pic_height); //调用 FMPEG媒体库的格式转换函数将 YUV格式视频转换成 RGB格式
```

#### 4.6 DMO注册函数

Windows Media Player播放视音频时,会根据视音频的格式到系统注册表中寻找到相应的解码器,如果找不到合适的解码器,Windows Media Player则播放不正常甚至无法播放。所以 DMO程序需要在系统中进行注册,而 DMO本身是一个 COM程序,COM程序的注册在 DllRegisterServer中进行。因此在 DllRegisterServer函数中通过接口函数 DMORegister在操作系统注册表中注册本 DMO的信息。设置 DMO的信息为只有一个输入端,输入的是 H.264格式的视频;设置 DMO的信息为只有一个输出端,输出的是 RGB格式的视频。如此设置以后,当 Windows Media Player播放 ASF中的 H.264视频时可以顺利地寻找到本插件,成功播放 ASF容器中 H.264格式的视频。

```
//定义输入媒体格式
```

```
DMO_PARTIAL_MEDIA_TYPE mt_input;
```

```
mt_input.type = MEDIA_TYPE_Video;
```

```
mt_input.subtype = MEDIASUBTYPE_H264;
```

```
//定义输出媒体格式
```

```
DMO_PARTIAL_MEDIA_TYPE mt_output; mt_output.type = MEDIA_TYPE_Video;
```

```
mt_output.subtype = MEDIASUBTYPE_RGB24;
```

```
DMORegister(L"DMO Sample", //DMO对象名
```

```
CLSID_Sample, //DMO对象的序列 ID DMOCATEGORY_VIDEO_DECODER, //视频解码器目录
```

```
0,
```

```
1, //输入媒体类型个数
```

```
&mt_input, //输入媒体类型格式
```

1, / 输出媒体类型个数

&mt\_output); / 输出媒体类型格式

到此,利用FIMPEG封装在DMO对象中构造一个能支持Windows Media Player播放ASF容器中的H.264视频播放插件所需要做的工作已经完成。

## 5 结束语

鉴于目前缺乏配合Windows Media Player播放ASF媒体容器格式的H.264视频播放的免费插件,本文介绍了一种简单实现此插件的方法,依赖于DMO高效性以及FIMPEG编解码库中性能优化的H.264解码算法,使得插件解码速度快,内存损耗小。利用此插件,Windows平台上的H.264流媒体系统可以利用Windows Media Player作为媒体系统的客户端,节省开发新播放器的成本,大大增强了系统的用户友好性。

## 参 考 文 献

- 1 Draft ITU - T recommendation and final draft international standard of joint video specification (ITU - T Rec. H.264/ISO/IEC 14496 - 10 AVC) in Joint Video Team (JVT) of ISO/IEC MPEG and ITU - TVCEG, JVTG050, 2003.
- 2 Microsoft Corporation. Advanced Systems Format(ASF) Specification, Revision 01.20.03, December 2004[5].
- 3 陆其明. Directshow开发指南. 北京:清华大学出版社, 2003.
- 4 T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz and M. Wien, eds. Joint Draft 7, Joint Video Team. Doc. JVT - T201, Klagenfurt, Austria, July 2006.
- 5 Joint Draft 6.0 on Multiview Video Coding. JVT - z209, Antalya, Turkey, Jan 2007.

论文降重，论文修改，论文代写加微信:18086619247或QQ:516639237

论文免费查重，论文格式一键规范，参考文献规范扫二维码：



[相关推荐：](#)

[基于FFmpeg的H.264视频解码器的研究与实现](#)

[基于PNX1700平台的H.264视频编码器优化实现](#)

[H.264视频编码研究与应用](#)

[H.264视频编码及其DSP实现](#)

[Symbian平台下基于FFmpeg的H.264解码器的移植](#)

[基于CUDA的H.264视频编码实现](#)

[H.264标准及其卫星视频应用](#)

[Windows平台下应用FFMPEG实现H.264视频回放](#)

[基于嵌入式平台的h.264视频编码器的实现](#)

[H.264视频编码器的VLSI实现](#)