



# 基于 Qt 多媒体技术和编码技术的本地音视频远程传输

■ 文 / 徐姣 赵贵荣 邵德明 南自信息技术有限公司

本文介绍了一种采用 Qt5.0 的 Multimedia 多媒体技术实现音视频采集和本地播放, 采用 FFmpeg 开源库把所采集的原始视频数据进行编码成 H.264 压缩格式, 采用 G.711 编码方式把所采集的原始音频数据编码, 再把编码之后的音视频压缩数据进行网络传输的技术。

Qt 是一个跨平台的 C++ 图形用户界面应用程序框架, 凭借其优良的跨平台特性、良好的封装机制、丰富的 API、大量的开发文档等优点, 得到了很多软件开发者的青睐。Qt 的多媒体技术发展到现在, 使用简单、功能灵活, 已经支持跨平台开发, 支持 QML 以获得更多资源, 它为开发者提供了一系列丰富的接口, 使开发者能够轻松便利的使用平台的、媒体回放以及操作摄像机和收音机设备的多媒体技术。

## 一、音视频编码

音视频编码技术就是采用特定的压缩技术, 将某个音频或视频格式的文件转换成另一种音频或视频格式文件的方式。

视频的编码方式有很多, 目前最为常用的是 MPEG 系列和 H.26X 系

列。其中 MPEG 系列是由 ISO 所制定而发布的视频、音频、数据的压缩标准。H.26X 系列是由国际电信联盟 (ITU-T) 开发的视频压缩编解码标准, 包括 H.261、H.262、H.263、H.263+、H.263++、H.264, 其中 H.264 标准是由国际电信联盟电信标准化部门 (ITU-T) 和 ISO/IEC 共同研究发布。与 MPEG 相比, H.264 具有更高的数据压缩比, 在同等图像质量下, H.264 的数据压缩率比 MPEG-2 高 2 倍 ~ 3 倍, 比 MPEG-4 提高了近 30%。经过 H.264 压缩的视频数据, 提高了存储性能, 在网络传输过程中占用的带宽更少更经济, 极大地抑制了由于摄像机噪声导致的图像失真, 背景流动现象, 使图像质量更加清晰。

G.711 是由国际电信联盟 (ITU-T) 制定的音频编码方式, 又

称 ITU-T G.711, 目前有两种编码方式: A-law 以及 Mu-law, Mu-law 主要运用于北美和日本, A-law 主要用于欧洲和世界其他地区。

## 二、FFmpeg 及 X264 编译

FFmpeg 是一款免费的开源的, 能够跨平台调用的软件, 可以用来记录、转换数字音视频, 并能将数字音视频转化为流。除了包含目前领先的音 / 视频编码库 libavcodec 之外, 还包括了以下几个部分:

libavformat: 用于各种音视频封装格式的生成和解析, 包括获取解码所需信息以生成解码上下文结构和读取音视频帧等功能。

libavutil: 包含一些公共的工具函数, 是 ffmpeg 的基础。

libswscale: 图像格式转换、视频场景比例缩放、色彩映射转换。

libpostproc 图像后期效果处理。

FFmpeg 是基于 Linux 平台开发的,要想在 windows 平台下使用,可以采用 MingGW 编译 FFmpeg 源码,生成动态库文件。但是下载的 FFmpeg 源码默认是不包含 H.264 编解码功能的,需要额外下载 x264 源码,采用 MingGW 编译器把 x264 编译成动态库之后,在编译 FFmpeg 的时候把 H.264 编解码功能添加进去,才能在 FFmpeg 中成功添加 H.264 编解码功能。

### 三、基于 Qt 多媒体技术的音视频采集和编码实现

#### 1. 多媒体模块的导入

Qt5.0 版本中的多媒体模块提供了一组丰富的 QML 类型和 C++ 类以满足多媒体开发的需要,另外还提供了必要的 API 接口来访问照相机和收音机设备。

要在 Qt5.0 以上的版本中使用多媒体模块,首先应该在工程中进行如下的设置:

```
CONFIG += mobility
```

```
MOBILITY = multimedia
```

这样设置之后才能使用 Qt 的多媒体库。

#### 2. FFmpeg 头文件及动态库的添加

把编译好的 FFmpegSDK 文件放到一个文件夹内,首先在工程中设

置好头文件路径以及需要的库文件,在工程中的文件内包含使用的头文件,注意的是,因为 FFmpeg 库的接口都是 C 函数,头文件中也没有 extern “C” 的声明,而 Qt 是 C++ 环境,所以在使用头文件时,需要添加 extern “C”。

#### 3. 摄像机视频的采集和编码

Qt 中的 QCamera 类为系统的摄像机设备提供了接口,它和 QVideoWidget 搭配使用用来显示摄像机视频图像,搭配 QMediaRecorder 可以保存摄像机视频,搭配 QCameraImageCapture 可以抓拍摄像机图像。

本文所介绍的视频采集方法,除了需要实时显示摄像机图像之外,还需要同步获取摄像机视频源数据,所以引入了 QAbstractVideoSurface 这个类。在工程中添加类 QVideoWidgetSurface,该类继承基类 QAbstractVideoSurface:

(1) 重载虚基函数 QList<QVideoFrame::PixelFormat> supportedPixelFormats (QAbstractVideoBuffer::HandleType handleType = QAbstractVideoBuffer::NoHandle) const; 用于设置程序支持的原始视频格式;

(2) 重载基类的 bool present(const QVideoFrame &frame) 函数,用于获取当前帧的视

频源数据;

(3) 重载基类的 bool start(const QVideoSurfaceFormat &format) 函数,用于启动视频表面及进行相应的窗口设置;

(4) 重载基类的 void stop() 函数,用于停止当前的视频表面和释放调用 start 时所产生的资源。

(5) 添加 void paint(QPainter \*painter) 函数,用以绘制视频图像,以在本地窗口中输出。其次,新建一个 QCamera 对象 m\_pCamera,新建一个 VideoWidgetSurface 对象 m\_pVideoSurface; 通过 m\_pCamera->service()->requestControl<QVideoRendererControl\*>() 方法获取 QVideoRendererControl 的指针 pControl,如果该指针存在,则调用 pControl 的 setSurface 关联到 m\_pVideoSurface。这样当 m\_pCamera 执行完 Start() 后,就可以在 m\_pVideoSurface 对象的 present 函数中得到每一帧的 QVideoFrame 视频数据。在测试程序中获取的视频数据格式为 RGB32,需要先转为 YUV240P 的格式,然后再用 FFmpeg 的 H.264 编码功能实现视频数据编码。

#### 4. 麦克风音频的采集和编码

Qt 中的 QAudioInput 类可以进行本地的麦克风、收音机等设备的音





频采集,而QAudioOutput类可以实现把本地的音频数据输出到音频输出设备,实现音频的播放功能。具体的步骤如下:

(1) 获取声音的输入输出设备,可以直接获取默认的设备,也可以在程序中查找声音输入输出设备,然后指定,本文采用直接获取默认设备的方式,调用QAudioDeviceInfo::defaultInputDevice()方法得到QAudioDeviceInfo的指针m\_audioInputDevice;调用QAudioDeviceInfo::defaultOutputDevice()方法得到QAudioDeviceInfo的指针m\_audioOutputDevice。

(2) 设置QAudioFormat格式,QAudioInput和QAudioOutput对象的QAudioFormat格式应一致,否则播放的声音就与输入的声音不同。新建一个QAudioFormat对象format,本例中设置如下:采样率(SampleRate):8000;信道数(ChannelCount):1;样本大小(SampleSize):16;编码方式(Codec):"audio/pcm";字节序(ByteOrder):QAudioFormat::LittleEndian;采样类型(SampleType):QAudioFormat::SignedInt。

(3) 分别按照上述的声音输入输出设备和声音参数信息创建QAudioInput对象m\_audioInput

和QaudioOutput对象m\_audioOutput。

(4) 调用m\_audioInput和m\_audioOutput的start接口,打开声音输入输出设备,将分别返回QIODevice指针m\_audioInputIODevice和m\_audioOutputIODevice。

(5) 当声音输入设备采集到声音数据后,将触发m\_audioInputIODevice的readyRead()信号,所以将此信号与该类的captureDataFromDevice()函数连接,即可在此函数内处理得到的原始数据。

(6)在captureDataFromDevice()槽函数中调用m\_audioInputIODevice的read函数得到原始的音频数据。

(7) 声音播放:把得到的原始音频数据写入缓存,然后调用m\_audioOutputIODevice的write函数把缓存内的数据写入声音输出设备即可听到声音。

(8) 原始音频数据采用G.711编码方式进行编码,就可以获得压缩后的音频数据,即可进行网络传输。

#### 四、开发过程中注意事项

1. 要在本地计算机设备中显示摄像机视频,在显示的窗口类重载paintEvent函数,在该函数内调用VideoWidgetSurface的paint函数。

2.FFmpeg的初始化只需要一次,不能在每次进行编码时都初始化和重新设置H.264编码器。

#### 五、结语

本文基于Windows操作系统,Qt5.1开发平台,充分利用Qt的多媒体模块和FFmpeg及G.711的编码技术,实现了本地摄像头、麦克风等设备的音视频采集和编码功能,不仅能够在本机实现音视频的预览,而且为实现音视频的网络传输做好了充分的准备。本例在win7操作系统中测试通过,同样也可应用于ipad等移动设备。

#### 参考文献:

[1] <http://qt-project.org/doc/qt-5.0/qtmultimedia/qtmultimedia-index.html>

[2] 黄诗文. 基于ffmpeg的高性能高清流媒体播放器软件设计[D]. 浙江:浙江大学[硕士论文],2012

[3] 司马飞,善学文,倪宏. Windows平台下应用FFMPEG实现H.264视频回放[J]. 微计算机应用,2008,29(11):61-65

[4] 张国庆. 基于FFmpeg的视频转码与保护系统的设计与实现[D]. 湖北:华中师范大学[说是论文],2011

# 基于Qt多媒体技术和编码技术的本地音视频远程传输

作者: [徐姣](#), [赵贵荣](#), [邵德明](#)  
作者单位: [南自信息技术有限公司](#)  
刊名: [中国安防](#)  
英文刊名: [China Security & Protection](#)  
年, 卷(期): 2014(8)

引用本文格式: [徐姣](#), [赵贵荣](#), [邵德明](#) [基于Qt多媒体技术和编码技术的本地音视频远程传输](#)[期刊论文]-[中国安防](#) 2014(8)