OPEN AI LAB

EAI610 IPC RTSP 硬件解码应用指南

2018-11-19

OPEN AI LAB

变更记录
(Reversion Record)

日期 (Date)	版本 (Rev)	说明 (Change Description)	作者 (Author)
2018-10-24	V0.8	初稿	路明
2018-11-16	V0.9	修改流程图重复部分	路明
2018-11-19	V1.0	根据会议review意见定稿	路明

目录(**catalog**)

1 前言

1.1 目的

本文档用于描述 EAI610-P0 平台上对网络摄像机产生的 rtsp 视频码流进行硬件解码的方案

1.2 术语

- **H264**：H.264，同时也是 MPEG-4 第十部分，是由 ITU-T 视频编码专家组（VCEG）和 ISO/IEC 动态图像专家组（MPEG）联合组成的联合视频组（JVT，Joint Video Team）提出的高度压缩数字视频编解码器标准。
- **H265**：H.265 是 ITU-T VCEG 继 H.264 之后所制定的新的视频编码标准。H.265 旨在在有限带宽下传输更高质量的网络视频，仅需原先的一半带宽即可播放相同质量的视频。
- **RTSP：**Real Time Streaming Protocol，RFC2326，实时流传输协议，是 TCP/IP 协议体系中的一个应用层协议，该协议定义了一对多应用程序如何有效地通过 IP 网络传送多媒体数据。

2 概述

EAI610 提供针对网络摄像机应用的 RTSP 硬解码样例代码，能够充分利用瑞芯微 RK3399 的硬件解码能力。

2.1 硬解码支持能力

EAI610 对视频编码仅支持 H264 和 H265 两种编码标准，对于常见的 1080p 网络摄像机支持能力如表 2-1

表 2-1 EAI610 视频硬解码支持能力

H264 硬解码	4 路 1080p@30fps
H265 硬解码	4 路 1080p@60fps

2.2 设备兼容性

EAI610 已经测试过与以下市售网络摄像机的解码兼容性

厂家	型号	分辨率	H264	H265
海康	DS-2CD1221-I3	200 万	OK	NA
大华	DH-IPC-HFW2230M-I2-V2	200 万	OK	OK
大华	DH-IPC-HFW4233F-ZSA	200 万	OK	OK
宇视	IPC242E-IR3-HUF40-C-DT	200 万	OK	OK

2.3 依赖包

IPC RTSP 硬解码示例代码依赖于如下开发包

软件包	版本	名称	功能
librockchip_mpp	1.3.7-3	瑞芯微媒体处理平台库（硬解码）	视频硬解码
librockchip_mpp-devel	1.3.7-3	瑞芯微媒体处理平台库开发包	
librtspclient	1.0-4	RTSP 客户端库及开发包	RTSP 处理
librockchip_rga	1.0.1-6	瑞芯微 RGA 库（硬件图像处理）	图像缩放，旋转，截取，格式转换
librockchip_rga-devel	1.0.1-6	瑞芯微 RGA 库开发包	
fastcv	1.0-1	计算机视觉优化库 FastCV（BladeCV）	图像存储，显示，信息叠加

3 IPC RTSP 硬件解码样例说明

IPC RTSP 硬件解码样例代码包含三个文件：

rtsp-demo.cpp: ipc rtsp 硬件解码样例主程序

ipc_rtsp.hpp: 定义 ipcDecoder 类，用于视频硬件解码

ipc_rtsp.cpp: ipcDecoder 类的实现

3.1 硬件解码主流程

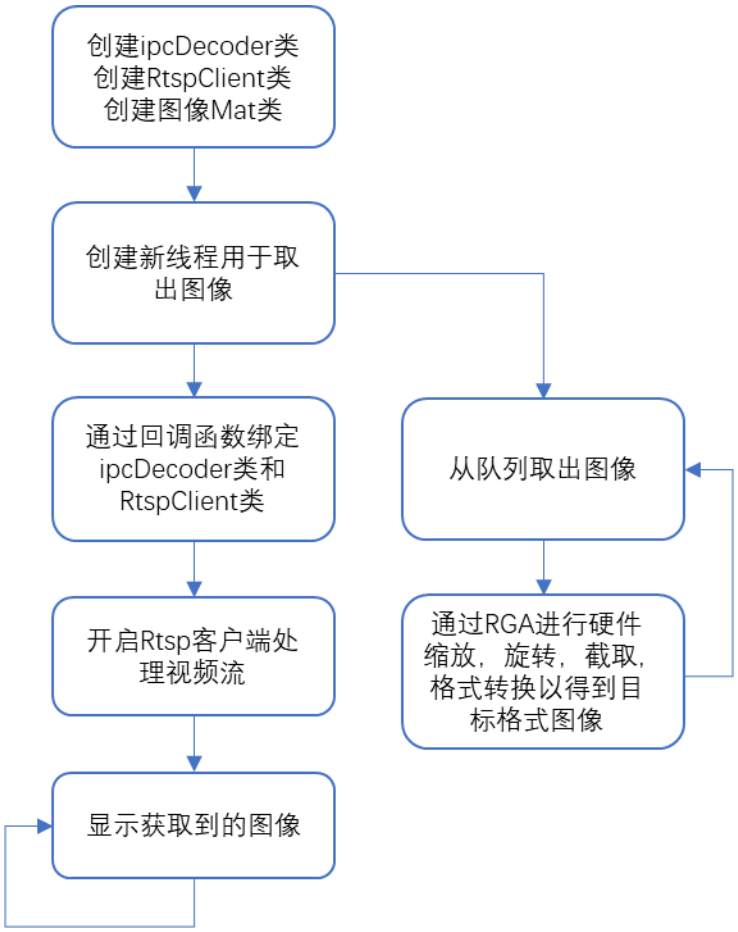

图 3-1 rtsp-demo 主流程

3.2 ipcDecoder 类接口说明

1. 构造函数

```
ipcDecoder(__u32 w, __u32 h, RgaRotate r, int V, int H, __u32 cx,
__u32 cy, __u32 cw, __u32 ch);
```

调用样例：

```
ipcDecoder *ipc = new ipcDecoder(width, height, rotate, vflip,
hflip, cropx, cropy, cropw, croph);
```

参数说明：

1) width, height：目标获取图像的宽和高
2) rotate：对视频获取到的图像需要做的旋转角度，支持 90，180，270 或不旋转
3) vflip：为 1 时，竖直镜像旋转
4) hflip：为 1 时，水平镜像旋转
5) cropx, cropy, cropw, croph：定义从获取到的图像进行截取的区域

[cropx, cropy]为区域左上角坐标，cropw, croph 为截取区域宽和高。

2. 初始化

```
int init(DecodeType type);
```

调用样例：

```
ipc->init(type);
```

参数说明：

type：解码类型，支持 H264(DECODE_TYPE_H264)和 H265(DECODE_TYPE_H264)两种

返回值：

0：正常

-1：RGA 创建失败

-2：MppDecoder 创建失败

3. 视频码流数据入队

```
int enqueue(unsigned char *buf, size_t len);
```

调用样例：

```
ipc->enqueue(buf, len);
```

参数说明：

1) buf：视频码流数据地址
2) len：数据长度

返回值：

0：数据正常入队

其他：数据未入队

4. 解码数据出队

```
DecFrame *dequeue(void);
```

调用样例：

```
frame = ipc->dequeue();
```

返回值：

获取到 Decframe 格式的帧

5. 释放帧

```
void freeFrame(DecFrame *frame);
```

调用样例：

```
ipc->freeFrame(frame);
```

参数说明：

Frame：释放 frame 占用的内存

6. RGA 处理

```
void rgaProcess(DecFrame *frame, __u32 dstFormat, Mat* mat);
```

调用样例：

```
ipc->rgaProcess(frame, V4L2_PIX_FMT_RGB24, mat);
```

参数说明：

1) frame：解码后的图像数据

2) dstFormat：目标图像格式
3) mat：目标图像指针

7. RGA 格式转换

```
rgaConvertFormat(Mat& src, Mat& dst, __u32 srcFormat, __u32
dstFormat, __u32 width, __u32 height);
```

调用样例：

```
ipc-> rgaConvertFormat (srcImage, dstImage, srcFormat, dstFormat,
w,h);
```

参数说明：

1) src：源图像
2) dst：目标图像
3) srcFormat：源图像格式
4) dstFormat：目标图像格式
5) width, height：图像宽高（目标和源图像尺寸相同）

8. 视频流接收回调函数

```
void onStreamReceive(unsigned char *buf, size_t len);
```

调用样例：

```
client.setDataCallback(std::bind(&ipcDecoder::onStreamReceive, ipc,
std::placeholders::_1, std::placeholders::_2));
```

参数说明：

1) buf：接受到的数据
2) len：数据长度

3.3 RtspClient 类接口说明

1. 构造函数：

```
RtspClient(std::string url, std::string username = "", std::string
password = "");
```

调用样例：

```
    Rtspclient client("rtsp://192.168.1.100/video1", "username",
"password");
```

参数说明：

1) url：IPC 摄像头的 RTSP 网络地址
2) username：IPC 摄像头的用户名，默认为空
3) password：IPC 摄像头的密码，默认为空

2. 设置回调函数：

```
setDataCallback(FRtspCallBack callBack);
```

调用样例：

```
    client.setDataCallback(std::bind(&priv_class::onRstpHandle,
priv_point, std::placeholders::_1, std::placeholders:_2));
```

参数说明：

1) priv_class 类回调函数：

```
void onRtspHandle(unsigned char *buf, size_t len)
```

说明：Rtspclient 每接收到一帧都会回调此函数；

其中 buf 存放 H264 图像数据，len 为图像的大小。

2) priv_point：类指针
3) _1&&_2：占位符

3. 开始获取 RTSP 流：

```
enable();
```

调用样例：

```
client.enable();
```

4. 停止获取 RTSP 流：

```
disable();
```

调用样例：

```
client.disable();
```

3.4 rtsp-demo 说明

1. 参数说明：

-d, --decoder: 指定解码格式，h265 或 h264，缺省为 h264

-w, --width: 指定目标图像宽度，缺省为 640

-h, --height: 指定目标图像高度，缺省为 360

-r, --rotate: 目标图像相对解码原图像旋转角度，有效输入为 0，90，180，270，缺省 0

-V, --vflip: 目标图像相对解码原图像做竖直镜像，缺省不做镜像旋转

-H, --hflip: 目标图像相对解码原图像做水平镜像，缺省不做镜像旋转

-c, --crop: 指定目标图像从解码原图中截取部分，格式为 cropx, cropy, cropw, croph

2. 视频码流 URL 设定

视频码流的 URL 设定放在 rtsp-demo.cpp 中，以宇视网络摄像机举例如下。

```
const string ipcUrl="rtsp://192.168.1.100/video1";
const string ipcUser="admin";
const string ipcPassword="12345678";
```

其他品牌或型号的网络相机 URL 请查阅相关产品用户手册。

3. 示例

```
./rtsp-demo
```

说明：解码采用缺省 H264 方式，目标图像宽高为缺省的 640、360，不做旋转和截取

```
./rtsp-demo -d h265 -c 0,0,1920,1080
```

说明：解码采用 H265 方式，截取解码后图像的区域为原图左上角[0,0]为原点，宽 1920，高 1080 的区域，目标图像宽高为缺省的 640、360。