Linux VENC Trouble Shooting

文件标识：RK-PC-YF-A01

发布版本：V0.1.0

日期：2023-04-05

文件密级：□绝密 □秘密 □内部资料 ■公开

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址： 福建省福州市铜盘路软件园A区18号

网址： www.rock-chips.com

客户服务电话： +86-4007-700-590

客户服务传真： +86-591-83951833

客户服务邮箱： fae@rock-chips.com

前言

概述

本文档主要描述多媒体处理VENC模块应用开发过程中遇到的常见问题及解决方法。

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V0.1.0	hogan.wang	2023-04-05	初始版本

目录

1. 各芯片平台对输入格式和编码格式的支持情况

1.1 RV1106/RV1103

	NV12	NV21	I420	NV16	YUYV	UYVY	NV24	RGB565	RGB888	ARGB8888
H264	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK	NG	OK	OK	OK

1.2 RK3588

	NV12	NV21	I420	NV16	YUYV	UYVY	NV24	RGB565	RGB888	ARGB8888
H264	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
JPEG	OK	NG	OK	NG	OK	OK	NG	OK	NG	OK

1.3 RV1126/RV1109

	NV12	NV21	I420	NV16	YUYV	UYVY	NV24	RGB565	RGB888	ARGB8888
H264	OK	OK	OK	OK	OK	OK	NG	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK	NG	OK	OK	OK
JPEG	OK	NG	OK	NG	OK	OK	NG	OK	NG	OK

- 备注：
  - NV12即YUV420SP
  - NV21即YUV420SP_VU
  - I420即YUV420P
  - NV16即YUV422SP
  - NV24即YUV444SP

2. 各芯片平台的分辨率限制

	H264	H265	JPEG
RV1109	64x64 - 3072x1728	64x64 - 3072x1728	96x96 - 8176x8176
RV1126	64x64 - 4096x2304	64x64 - 4096x2304	96x96 - 8176x8176
RK3566	64x64 - 1920x1080	64x64 - 1920x1080	96x96 - 8176x8176
RK3568	64x64 - 1920x1080	64x64 - 1920x1080	96x96 - 8176x8176
RK3588	64x64 - 16384x16384	96x88 - 16384x16384	96x32 - 8176x8176
RV1106	64x64 - 8192x8192	64x64 - 8192x8192	32x32 - 4096x4096
RV1103	64x64 - 8192x8192	64x64 - 8192x8192	32x32 - 2560x1440

3. 各芯片平台的编码性能

	H264	H265	JPEG
RV1109	3072x1728@30fps	3072x1728@30fps	90000000pps
RV1126	4096x2304@30fps	4096x2304@30fps	90000000pps
RK3566	1920x1080@60fps	1920x1080@60fps	90000000pps
RK3568	1920x1080@60fps	1920x1080@60fps	90000000pps
RK3588	单核1920x1080@240fps 有2个H264/H265核	单核1920x1080@240fps 有2个H264/H265核	90000000pps 单核1920x1080@60fps 有4个JPEG核
RV1106	2880x1616@30fps	2880x1616@30fps	2880x1616@5fps
RV1103	2560x1440@30fps	2560x1440@30fps	2560x1440@5fps

4. 各芯片平台对裁剪、镜像、旋转编码功能的支持情况

4.1 RV1106/RV1103

	裁剪	水平镜像	垂直镜像	旋转90度	旋转180度	旋转270度
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

4.2 RK3588

	裁剪	水平镜像	垂直镜像	旋转90度	旋转180度	旋转270度
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

备注：JPEG的水平镜像、垂直镜像、旋转180度通过rockit内部RGA自动处理，编码器本身不支持。

4.3 RV1126/RV1109

	裁剪	水平镜像	垂直镜像	旋转90度	旋转180度	旋转270度
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

备注：JPEG的水平镜像、垂直镜像、旋转180度通过rockit内部RGA自动处理，编码器本身不支持。

5. RV1106、RV1103平台VENC常见问题

5.1 和VI绑定失败

首先检查一下node是否创建成功：cat /dev/mpi/vsys，对应的rkisp-vir0，rkv_enc node是否创建成功。RK_MPI_VENC_CreateChn不会创建node，因为卷绕、超大帧、Combo等属性还未配置，此时还不能创建node，在RK_MPI_VENC_StartRecvFrame才会创建node，因此需要在RK_MPI_VENC_StartRecvFrame之后，调用系统绑定接口才能完成绑定。

5.2 CreateChn接口的u32BufSize和u32StreamBufCnt的理解

RV1106、RV1103驱动使用一块ringbuf环形buffer来实现码流的输出，区别于RV1126、RV1109、RK3588、RK3566、RK3568等平台：每个输出的I帧或者P帧都是独立的一块buffer。因此u32BufSize在RV1106、RV1103平台理解为可以存储一个I帧和多个P帧的空间大小，在RV1126、RV1109、RK3588、RK3566、RK3568等平台理解为存储单个I帧或者单个P帧的空间大小。u32StreamBufCnt代表用户可获取的最多码流个数，在RV1106、RV1103平台理解为rockit MPI可以操作的buffer handle个数，在RV1126、RV1109、RK3588、RK3566、RK3568等平台理解为实际申请了多少个u32BufSize大小的可以操作的buffer。RV1106、RV1103平台u32BufSize大小可以根据编码类型、编码码率、需要连续存储多少帧来配置，u32StreamBufCnt可以配置为用户预期最多需要hold住多少个buffer handle。RV1126、RV1109、RK3588、RK3566、RK3568等平台u32BufSize大小需要根据编码类型、编码码率配置为I帧预期编码的大小略大一些，u32StreamBufCnt配置为用户需要的buffer数量。

5.3 GetStream失败

首先检查一下绑定状态buffer输入输出是否增加：cat /dev/mpi/vsys，前级的onfa_cnt是否增加，rkv_enc node的infa_cnt是否增加。其次检查一下cat /proc/vcodec/enc/venc_info, ringbuf的ring_fail_cnt是否工作正常，如果ring_fail_cnt一直增加表明ringbuf无输出buffer可用，一直在丢输入的frame。venc_info的strm_cnt代表未被应用取走已经编码的码流，strm_out代表应用取走的码流、还未还给驱动，可以通过这两个值查看目前ring_buf的使用情况。还可以进一步输入dmesg查看内核是否有异常log输出。

5.4 RefBufShare属性

RV1106、RV1103驱动可以通过使能RefBufShare只使用一块buffer实现参考帧和重构帧卷绕，区别于RV1126、RV1109、RK3588、RK3566、RK3568等平台使用了两块独立的buffer用于参考帧和重构帧。使能RefBufShare可以节省参考帧和重构帧buffer开销。使能RefBufShare后不支持超大帧编码、不支持去呼吸效应。只有H264、H265编码支持使能RefBufShare。

5.5 Combo属性

RV1106、RV1103驱动支持H264/H265编码和JPEG编码 combo，理解为H264/H265编码和JPEG编码共用一个输入源，一次硬件完成可以同时输出一帧H264/H265编码码流和一帧JPEG编码码流。JPEG编码通道设置Combo属性后，不再支持和VI等前级绑定。两个combo通道的输入源一样，输入分辨率和输入格式均一样。Combo属性使能可以节省一路输入源，节省系统带宽。

5.6 BufWrap属性

设置编码器的BufWrap卷绕属性，两个绑定的VI、VENC需要同时配置卷绕属性，如果存在combo通道，对应的JPEG combo通道也需要设置卷绕属性。只支持一路VI绑定VENC的卷绕。Buf卷绕行数、Buf卷绕大小配置需要和VI保持一致。BufWrap属性使能可以节省输入buffer内存开销。BufWrap属性使能后不支持超大帧重编、不支持去呼吸效应。

5.7 查看编码单帧耗时

```
echo 0x100 > /sys/module/mpp_vcodec/parameters/mpp_dev_debug
```

6. RK3588、RK3566、RK3568、RV1126、RV1109平台VENC常见问题

6.1 如何排查没有数据输出

- 首先使用rk_mpi_venc_test测试，若测试正常，检查与代码示例的差异。
- 检查s32RecvPicNum是否设置为0，或者通过dumpsys venc查看snap_set是否为非-1并且seq已经等于snap_set数值，此时编码会停止输出。
- 检查输出buffer是否设置太小，导致编码输出异常报错。
- 查看串口日志是否有mpp或者其他相关报错。

6.2 编码输出帧率不足

- 确认帧率是否超过实际规格书定义。
- 确认单独使用rk_mpi_venc_test是否帧率正常。如果test测试也异常，查看芯片温度是否过高导致机器进入温控，查看编码频率是否有降低。如果都正常则串口输入以下命令查看内核编码单帧时间是否过长。

```
echo 0x100 > /sys/module/rk_vcodec/parameters/mpp_dev_debug
```

- 确认是否开启帧率控制导致，可以通过dumpsys venc查看。
- 确认是否是VENC通道后级绑定模块归还Buffer数据慢，或用户获取/释放Buffer的数据慢。
- 若单独编码正常，加上其他业务出现帧率低，此时需分析DDR带宽是否充足，可以尝试提高DDR频率，并结合整体业务分析优化带宽占用。

6.3 输出图像马赛克

- 确认输入图像是否正常
- 确认单独使用rk_mpi_venc_test是否正常。
- 使用rk_mpi_vdec_test解码查看是否正常。
- 检查是否超频使用编码频率，降低对应clk查看是否正常。

```
H264/H265编码：cat /sys/kernel/debug/clk/clk_summary | grep venc
JPEG/MJPEG编码(RV1109/RV1126/RK3566/RK3568)：cat
/sys/kernel/debug/clk/clk_summary | grep vepu
JPEG/MJPEG编码(RK3588)：cat /sys/kernel/debug/clk/clk_summary | grep jpeg
```

- 如果只有JPEG/MJPEG编码输出图像异常，查看是否输入为压缩数据。可以通过dumpsys venc中的afbc_mode查看，如果是0x0为非压缩格式，其余则为压缩格式。
- 尝试使用其他编码器是否正常。如果是MJPEG/H264/H265单独一个编码器异常，查看对应编码器电压供电是否满足需求，尝试抬升25-50mV查看是否有改善。

6.4 编码输出分辨率配置

- 当通过RK_MPI_VENC_SetChnParam设置了缩放功能时，编码器统一按照缩放设置的分辨率输出。
- 编码器默认采用自适应分辨率编码，即当没有开启缩放功能时，按照输入的实际分辨率编码输出，一般JPEG抓图编码可以使用此功能，避免频繁创建销毁编码器。需要注意的是通道输出的Buffer（u32BufSize）要按照最大分辨率的Buffer大小进行配置，避免因为创建时申请的输出Buffer过小导致动态切换大分辨率后无法正常编码。

6.5 码率控制异常问题

- 确认输入数据及输出码流数据都正常，无花屏、马赛克等异常数据。
- 查看码控相关信息设置是否正确，通过dumpsys venc关注venc chn attr查看rc_mode及gop信息，及venc chn rc info的Qp及bps设置值。
- 保存编码输出码流，从码流Qp进行分析，如果Qp在Qp max附近说明视频序列比较复杂，这种情况一般真实的码率会大于设置码率，如果Qp在Qp min附近，说明视频序列偏简单，这种情况一般真实码率会小于设置码率。可以通过调节RK_MPI_VENC_SetRcParam中的Qp min/Qp max（H264/H265）或qfactor（JPEG/MJPEG）进行调整相应Qp。
- 如果输入源为sensor数据，存在超码率问题，需要保证输入图像噪点较小，如果输入图像本身质量较差，码率则一般也会较大。

6.6 压缩格式输入支持

- H264/H265仅支持YUV420、YUV422压缩格式输入。
- JPEG/MJPEG硬件编码器本身无法支持AFBC压缩格式输入，目前仅RK3588平台支持编码通道内部通过扩展模块来支持压缩格式输入，其他平台的JPEG/MJPEG压缩输入需要通过VPSS模块进行解压缩后再送入VENC，此处理会额外占用带宽及硬件资源，因此仅建议在抓图等少量帧编码的场景下采用输入压缩图像，否则建议前级模块关闭压缩输出。

6.7 其他注意事项

- 动态参数设置建议在创建通道后、启动编码前设置，避免在编码过程中频发设置。
- 调用RK_MPI_VENC_StopRecvFrame后，建议调用RK_MPI_VENC_ResetChn或取走输出码流，否则有可能前级模块无法正常工作。

7. RC配置示例

- H264

```
        VENC_CHN_ATTR_S         stAttr;
        memset(&stAttr, 0, sizeof(VENC_CHN_ATTR_S));
        RK_MPI_VENC_GetChnAttr(0, &stAttr);
        stAttr.stRcAttr.enRcMode = VENC_RC_MODE_H264CBR;
        stAttr.stRcAttr.stH264Cbr.u32Gop = 60;
        stAttr.stRcAttr.stH264Cbr.u32SrcFrameRateNum = 30;
        stAttr.stRcAttr.stH264Cbr.u32SrcFrameRateDen = 1;
        stAttr.stRcAttr.stH264Cbr.fr32DstFrameRateNum = 30;
        stAttr.stRcAttr.stH264Cbr.fr32DstFrameRateDen = 1;
        stAttr.stRcAttr.stH264Cbr.u32BitRate = 1400;
        RK_MPI_VENC_SetChnAttr(0, &stAttr);

        VENC_RC_PARAM_S         stRcParam;
        memset(&stRcParam, 0, sizeof(VENC_RC_PARAM_S));
        stRcParam.s32FirstFrameStartQp = 35;
        stRcParam.stParamH264.u32StepQp = 2;
        stRcParam.stParamH264.u32MinQp = 29;
```

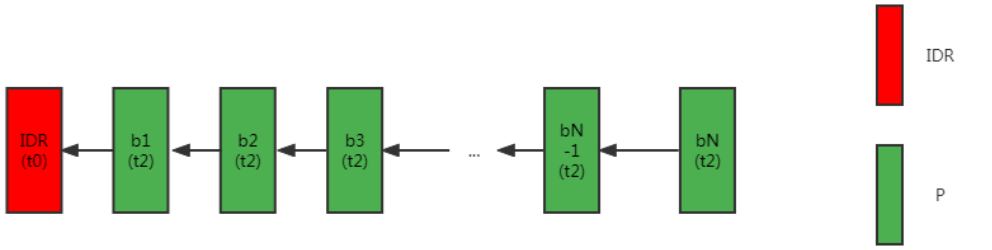
```
stRcParam.stParamH264.u32MaxQp = 48;
stRcParam.stParamH264.u32MinIQp = 34;
stRcParam.stParamH264.u32MaxIQp = 48;
stRcParam.stParamH264.s32DeltIpQp = -2;
stRcParam.stParamH264.s32MaxReEncodeTimes = 2;
RK_MPI_VENC_SetRcParam(0, &stRcParam);
```

- H265

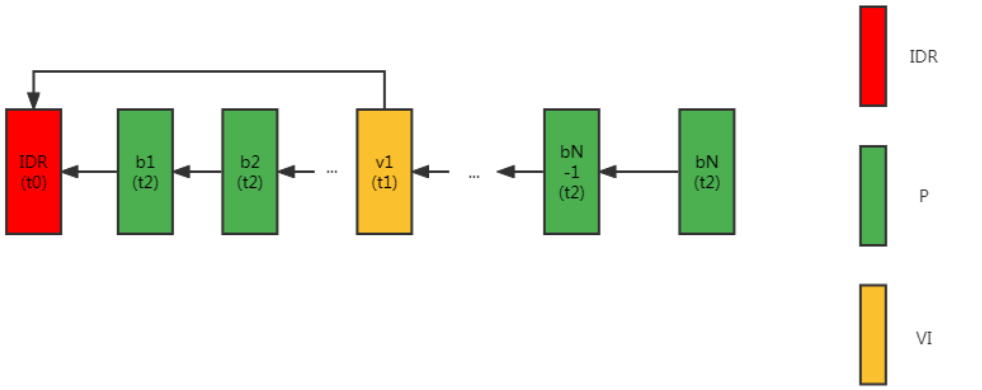
```
VENC_CHN_ATTR_S         stAttr;
memset(&stAttr, 0, sizeof(VENC_CHN_ATTR_S));
RK_MPI_VENC_GetChnAttr(chn, &stAttr);
stAttr.stRcAttr.enRcMode = VENC_RC_MODE_H265CBR;
stAttr.stRcAttr.stH265Cbr.u32Gop = 60;
stAttr.stRcAttr.stH265Cbr.u32SrcFrameRateNum = 30;
stAttr.stRcAttr.stH265Cbr.u32SrcFrameRateDen = 1;
stAttr.stRcAttr.stH265Cbr.fr32DstFrameRateNum = 30;
stAttr.stRcAttr.stH265Cbr.fr32DstFrameRateDen = 1;
stAttr.stRcAttr.stH265Cbr.u32BitRate = 1400;
RK_MPI_VENC_SetChnAttr(chn, &stAttr);

VENC_RC_PARAM_S         stRcParam;
memset(&stRcParam, 0, sizeof(VENC_RC_PARAM_S));
stRcParam.s32FirstFrameStartQp = 35;
stRcParam.stParamH265.u32StepQp = 2;
stRcParam.stParamH265.u32MinQp = 29;
stRcParam.stParamH265.u32MaxQp = 48;
stRcParam.stParamH265.u32MinIQp = 34;
stRcParam.stParamH265.u32MaxIQp = 48;
stRcParam.stParamH265.s32DeltIpQp = -2;
stRcParam.stParamH265.s32MaxReEncodeTimes = 2;
RK_MPI_VENC_SetRcParam(0, &stRcParam);
```

- 注意事项：
  - enRcMode和对应配置结构体要对应。
  - GOP建议为输出帧率的整数倍。
  - 支持输出丢帧，不支持输出插帧。
  - 码率设置根据产品需求来配置。
  - u32MinQp、u32MaxQp对应P帧的Qp范围，P帧码率过大可以加大u32MinQp。
  - u32MinIQp、u32MaxIQp对应I帧的Qp范围，I帧码率过大可以加大u32MinIQp。
  - s32DeltIpQp为负表示期望I帧Qp值比前几帧P帧的平均Qp大多少，可以控制I帧的码率不至于太大，对应的I帧的质量相对差一点。
  - s32MaxReEncodeTimes配置重编次数。如果使能超大帧重编，码率超过阈值会进入重编流程。重编超过设置的次数，码率依旧超过阈值，会直接输出该帧。重编次数会影响编码耗时，对时延有需求的项目需求评估下重编次数的合理值。

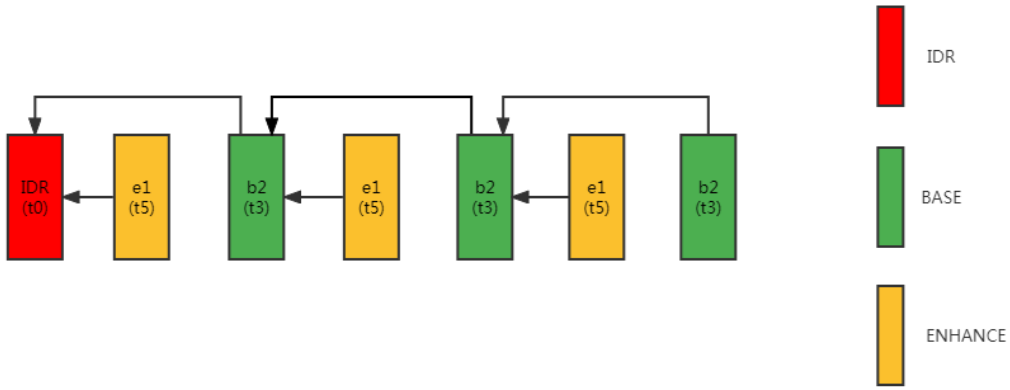
8. GOP结构



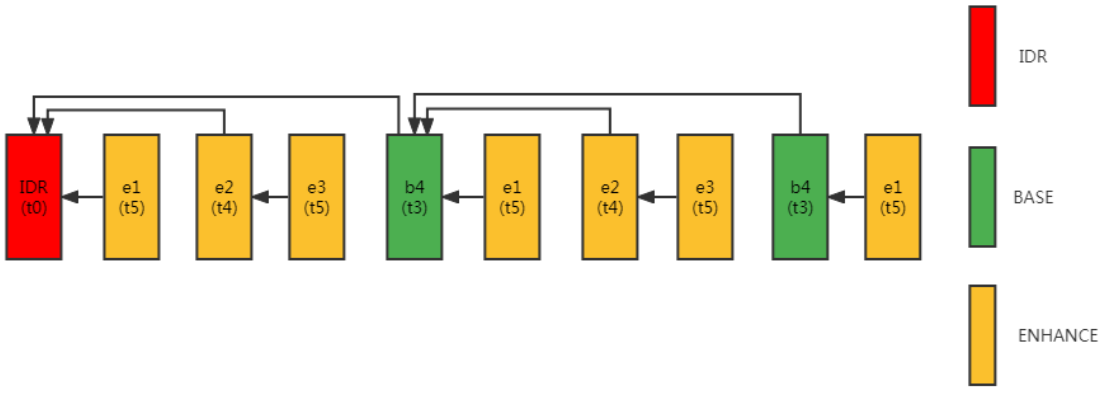
NORMALP参考关系



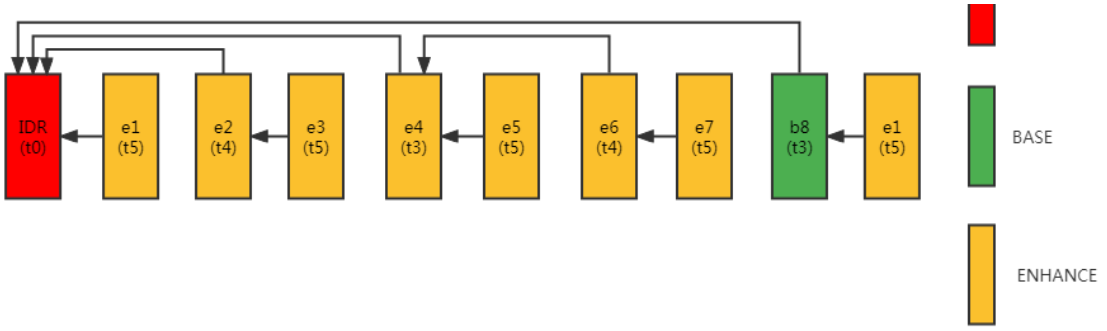
SMARTP参考关系



TSVC2参考关系



TSVC3参考关系



TSVC4参考关系

9. 码率控制

码率控制（后文简称：码控或RC）分硬件、软件两部分实现：软件端实现帧级量化参数（quantization parameter，后文简称：Qp）的计算，硬件端实现行级、块级Qp的计算、调整，软件和硬件配合使得码率能按需求平稳过渡。根据信息论可知，视频/图像的压缩比越大，视频/图像的画质越差；反之，视频/图像的压缩比越小，视频/图像的画质越好。不同场景的帧间运动量、帧内的复杂度不同，码率会随着场景不同而波动。

9.1 CBR

固定码率控制（Constant Bit Rate）。在码率统计时间内保证码率能平稳过渡。码率的稳定度由两个变量来评估：

- 码率统计时间 u32StatTime

单位为秒（s），码率统计时间越长，每帧图像的码率波动对于码控影响越小，码率的调节会越缓慢，视频序列质量波动越小；码率统计时间越短，单帧图像码率对整体码率的波动影响越大，图像码率调节越灵敏，视频画质波动越明显。

- 行级码控调整幅度 u32RowQpDelta

行级码控调整幅度是一帧内行级调节的最大范围，其中以宏块行/CTU行为单位。调节幅度越大，允许行级Qp波动越大，码率越平稳。对于图像复杂度分布不均的场景，行级码控调整的过大会影响画质。

固定码率控制得到的码流，一般会保证在设置的目标码率范围±5%以内，统计时间越长，稳定性越高，在一些经常出现内容变化的场景，最后的平均码率一般会保证在±10%以内，即属于码率平稳性和画质平稳性的一个很好的折中。

9.2 VBR

可变码率控制（Variable Bit Rate）。在码率统计时间内允许码率波动，用于保证编码视频序列质量平稳过渡。编码器提供用户可设置的MaxQp, MinQp, MaxBitrate和ChangePos 。

- MaxQp视频序列最大的Qp值
- MinQp视频序列最小的Qp值
- MaxBitRate限制统计时间内最大的码率
- u32BitRate用来控制开始调整Qp的码率基准线

当编码码率小于u32BitRate时，图像Qp会逐步向MinQp调整；反之，图像Qp会逐步向MaxQp调整。

可变码率控制，一般会在简单的场景（运动较少或不运动，纹理简单）能有效的节省文件大小，但在复杂场景（运动较多且纹理复杂）也可以保证输出的码率在设置的最大码率的±10%以内。

9.3 AVBR

自适应可变码率控制（Adaptive Variable Bit Rate）。在码率统计时间内允许码率波动，保证视频序列质量的平稳过渡。AVBR模式内部会检测当前场景的运动、静止状态，根据运动程度主动的抬升码率，实现视频画质的平稳过渡。

- MaxQp视频序列最大的Qp值
- MinQp视频序列最小的Qp值
- MaxBitRate限制统计时间内最大的码率
- u32BitRate和MinStillPercent共同作用，u32BitRate * MinStillPercent表示静止情况下的最小码率。根据运动程度的不同目标码率会在最大和最小码率间调整。请注意：MinStillPercent是驱动内部固定住的值，外面不可配，此处是为了说明跟VBR模式的不同而增加MinStillPercent的说明。

9.4 SVC

智能编码（Smart Video Coding）。在码率统计时间内允许码率波动，保证视频序列质量的平稳过渡。内部会根据运动复杂度、图像复杂度动态的调整目标码率。运动越小、图像复杂度越简单，码率就越小；运动越复杂、图像复杂度越高，码率就越大。

码控过程，内部根据图像复杂度、运动复杂度对I帧做约束，该约束适合所有码率控制。当前I帧前面的统计帧内运动复杂度较高、图像复杂度较高，需要把当前I帧的帧级Qp限制在I帧的帧级Qp最小值或以上；P帧的帧级Qp限制在P帧帧级Qp最小值或以上，避免突然间出现的运动导致码率过渡消耗出现模糊、马赛克。

智能编码过程，目前只有RV1103/RV1106支持得比较完善，包括三个模式：

- 码率优先
- 质量优先
- 码率和质量的折中

10. 常见问题分析

10.1 视频序列中间某些时间段有马赛克、模糊

一些户外或室内场景，运动逐步加大的过程，例如朝镜头运动过程运动目标成像由小及大，此时需要内部做准确检测，把I帧帧级Qp限制在u32FrmMinIQp以内，即适当降低I帧的码率消耗分摊到后续的P帧，避免造成P帧的帧级Qp飚得过高导致马赛克、模糊。调试过程需要注意，有些运动过大在一定分辨率、码率约束下，如1440p 25fps的HEVC，如果要求码率在3Mbps，则不一定可以很好的保证图像画质，可以考虑适当降低ISP出来的图像复杂度，配合编码一起减少马赛克、模糊的出现。

10.2 AE调整导致的马赛克

灯光变化，或人物在镜头前走动导致的AE调整，如果AE调整过于剧烈，可以先考虑让AE平滑过渡，同时通过VEPU_PP模块出来的帧级像素平均值，判断出前后两帧有光线变化，限制I帧的帧级Qp在u32FrmMinIQp内，限制P帧的帧级Qp在u32FrmMinQp内，让码率分配平稳过渡，缓解马赛克的问题。

10.3 常规场景下的运动拖影

编码器会导致运动拖影，主要原因在于运动物体边缘有些块模式选择不合理，导致有上一帧图像内容的残余，调用RK_MPI_VENC_EnableMotionDeblur函数接口，使能去模糊或去残余的功能，缓解/解决运动拖尾问题。

10.4 超大帧处理

视频序列中有些帧特别是I帧，可能超过预设值，比如单帧超过1024KB，此时容易导致码率过冲而丢帧，丢帧继而引发花屏问题。可以通过调用超大帧处理的API接口函数RK_MPI_VENC_SetSuperFrameStrategy进行处理，在卷绕模式下，会把超过阈值（如单帧超过1024KB）的帧直接丢弃，紧接着编码下一帧，避免丢帧、花屏的问题。

10.5 彩色切换黑白导致色偏

彩色转黑白过程，一般会因为黑白帧参考前一帧彩色帧，或者黑白帧的osd是彩色，且Qp较大时会出现色偏，可以通过检测彩色切换为黑白过程这一行为，针对性处理：

- 检测第一帧黑白帧并编码为全I块
- 调整合适的Qp，避免码率瞬间过冲，并让第一帧黑白帧之后的少数帧的Qp平滑过渡
- osd从彩色换成黑白，如果osd必须是彩色则对osd区域做适当的Qp保护

10.6 普通场景下的色度侵染

因为色度分量相对于亮度分量细节较少，低频占比较大，因此可以考虑适当的调整色度的Qp值，即调整chroma_qp_offset的值为负值，一般推荐选择在-6到-12之间。色度分量Qp调整之后，码率也不至于像亮度分量Qp调整那样，色度分量的Qp调小，不容易导致过冲。

10.7 呼吸效应

RV1103/RV1106的卷绕模式，需要合理的让I帧和P帧的Qp合理过渡，在相对静止场景下，呼吸效应能得到有效的缓解。建议qp_delta_ip值设置为2~4，一般推荐选择2 。也要配合调整u32FrmMinIQp和u32FrmMinQp一起使用，通常推荐u32FrmMinQp为28~33之间，u32FrmMinIQp选择在21~26之间，尽量保证在简单场景切换到复杂场景过程的画质平稳过渡，也尽量保证运动场景切换到静止场景过程呼吸效应得到较好的缓解。

RV1103/RV1106的离线模式（非卷绕模式）有针对呼吸效应做特殊处理，效果相对于卷绕模式能有较好的改善。

10.8 噪声形态保留

在静态区域特别是弱纹理或平坦区域，在低照度下容易出现显性的噪声，且噪声形态经过编码之后很容易被改变，人眼对弱纹理或平坦区域的异常（编码导致的突兀）特别敏感，因此需要倾斜部分码率到这块区域（通过调整Qp的方式），且也要尽量使这块区域跟周围区域有效的保持一致性，也可以减少I块的出现，保证这块区域的噪声更加自然。