Linux VENC Trouble Shooting

ID: RK-PC-YF-A01

Release Version: V0.1.0

Release Date: 2023-04-05

Security Level: □Top-Secret □Secret □Internal ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document primarily presents common issues and the solutions encountered during the application development process of the multimedia processing VENC module.

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V0.1.0	Hogan Wang	2023-04-05	Initial version

Contents

1. Input Formats and Encoding Formats Supported

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

Note:

- NV12 represents YUV420SP
- NV21 represents YUV420SP_VU
- I420 represents YUV420P
- NV16 represents YUV422SP
- NV24 represents YUV444SP

2. Resolution Limitations

	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. Encoding Performance

	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	Single core 1920x1080@240fps dual H264/H265 core	Single core 1920x1080@240fps dual H264/H265 core	90000000pps Single core 1920x1080@60fps quad JPEG core
RV1106	2880x1616@30fps	2880x1616@30fps	2880x1616@5fps
RV1103	2560x1440@30fps	2560x1440@30fps	2560x1440@5fps

4. Cropping, Mirroring and Rotation Encoding Functions

4.1 RV1106/RV1103

	Cropping	Horizontal mirroring	Vertical mirroring	90-degree rotation	180-degree rotation	270-degree rotation
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

4.2 RK3588

	Cropping	Horizontal mirroring	Vertical mirroring	90-degree rotation	180-degree rotation	270-degree rotation
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

Note: Horizontal mirroring, vertical mirroring, and 180-degree rotation for JPEG are automatically processed through the internal RGA in Rockit. The encoder itself does not support these functionalities.

4.3 RV1126/RV1109

	Cropping	Horizontal mirroring	Vertical mirroring	90-degree rotation	180-degree rotation	270-degree rotation
H264	OK	OK	OK	OK	OK	OK
H265	OK	OK	OK	OK	OK	OK
JPEG	OK	OK	OK	OK	OK	OK

Note: Horizontal mirroring, vertical mirroring, and 180-degree rotation for JPEG are automatically processed through the internal RGA in Rockit. The encoder itself does not support these functionalities.

5. Common Issues about VENC on RV1106/RV1103 Platform

5.1 Failed to Bind with VI

First, check if the node is created successfully by running the command: `cat /dev/mpi/vsys`. Verify if the corresponding `rkisp-vir0` and `rkv_enc` nodes are created successfully. Note that the `RK_MPI_VENC_CreateChn` function does not create the node immediately because attributes like wrap, super size frames, and Combo are not configured at that stage. The node is created in `RK_MPI_VENC_StartRecvFrame`. Therefore, the system bind interface should be called after `RK_MPI_VENC_StartRecvFrame` to complete the binding process.

5.2 `u32BufSize` and `u32StreamBufCnt` in the `CreateChn` Interface

For RV1106 and RV1103 drivers, a ring buffer is used to implement stream output. This differs from platforms like RV1126, RV1109, RK3588, RK3566, and RK3568, where each output I-frame or P-frame has its own independent buffer. In RV1106 and RV1103 platforms, `u32BufSize` refers to the space required to store one I-frame and multiple P-frames. In RV1126, RV1109, RK3588, RK3566, and RK3568 platforms, it refers to the space required to store a single I-frame or P-frame. `u32StreamBufCnt` represents the maximum number of streams that can be obtained by users. In RV1106 and RV1103 platforms, it indicates the number of buffer handles that Rockit MPI can operate. In RV1126, RV1109, RK3588, RK3566, and RK3568 platforms, it represents the number of actual buffers of `u32BufSize` size that have been allocated and can be operated on. For RV1106 and RV1103 platforms, `u32BufSize` can be configured based on the encoding type, encoding bitrate, and the number of consecutive frames that need to be stored. `u32StreamBufCnt` can be configured to hold the maximum expected number of buffer handles. For RV1126, RV1109, RK3588, RK3566, and RK3568 platforms, `u32BufSize` should be slightly larger than the expected size of the I-frame for the encoding type and bitrate, and `u32StreamBufCnt` should be set to the desired number of buffers needed by users.

5.3 Failed to GetStream

First, check if the input/output buffers under binding state has increased by running the command: `cat /dev/mpi/vsys`. Verify if the `onfa_cnt` of the previous stage has increased and if the `infa_cnt` of the `rkv_enc` node has increased. Next, check `cat /proc/vcodec/enc/venc_info` and verify if the `ring_fail_cnt` of the ring buffer is working properly. If `ring_fail_cnt` keeps increasing, it indicates that there are no available output buffers in the ring buffer, resulting in continuous frame loss. The `strm_cnt` in `venc_info` represents the encoded streams that have not been taken by the application, while `strm_out` represents the streams taken by the application but not yet returned to driver. These values can be used to check the current usage of the ring buffer. Additionally, you can check the kernel logs by running `dmesg` to see if there are any abnormal log outputs.

5.4 RefBufShare Attribute

In RV1106 and RV1103 drivers, enabling the `RefBufShare` attribute to realize using a single buffer for both reference frames and reconstructed frames wrapping. This is different from platforms like RV1126, RV1109, RK3588, RK3566, and RK3568, which use two separate buffers for reference frames and reconstructed frames. Enabling `RefBufShare` helps save buffer overhead for reference frames and reconstructed frames. However, enabling `RefBufShare` does not support superframes and breathing effect. Only H.264 and H.265 encoding support the `RefBufShare` attribute.

5.5 Combo Attribute

In RV1106 and RV1103 drivers, the combo attribute enables H.264/H.265 and JPEG encoding to share the same input source. It allows hardware to simultaneously output a frame of H.264/H.265 encoded stream and a frame of JPEG encoded stream. Once the JPEG encoding channel is set with the combo attribute, it no longer supports binding with VI or other upstream components. The input source for both combo channels is the same, and the input resolution and format are also the same. Enabling the combo attribute helps save one input source and system bandwidth.

5.6 BufWrap Attribute

Setting the BufWrap attribute of the encoder requires both VI and VENC to be configured with the wrap attribute simultaneously. If a combo channel exists, the corresponding JPEG combo channel also needs to be set with the wrap attribute. Only one VI binding to VENC supports wrapping. The configuration of Buf wrap line count and Buf wrap size should be consistent with VI. Enabling the BufWrap attribute helps save memory overhead for input buffers. However, enabling the BufWrap attribute does not support superframe re-encoding and breathing effect.

5.7 Checking Encoding Time for a Single Frame

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

6. Frequently Asked Questions about VENC on RK3588, RK3566/RK3568 and RV1126/RV1109 Platform

6.1 No Data Output Troubleshooting

- First, use `rk_mpi_venc_test` for testing. If the test is successful, check for differences between your code and the code examples.
- Check if `s32RecvPicNum` is set to 0, or use `dumpsys venc` to check if `snap_set` is not -1 and if the `seq` has already reached the value of `snap_set`. In such cases, the encoding will stop outputting.
- Check if the output buffer size is set too small, causing abnormal encoding output errors.
- Check the serial port logs for any MPP or other related errors.

6.2 Insufficient Encoding Output Frame Rate

- Confirm if the frame rate exceeds the specifications defined in the documentation.
- Confirm if the frame rate is normal when using `rk_mpi_venc_test` alone. If the test also shows abnormal frame rates, check if the chip temperature is too high, leading to thermal control and reduced encoding frequency. If everything is normal, input following command in the serial port to check if the kernel encoding time for a single frame is too long.

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

- Confirm if it is caused by enabling frame rate control, which can be checked `dumpsys venc`.
- Confirm if the later return of Buffer data by the module bound to the VENC channel or later data acquisition/release of Buffers by users.
- If encoding alone works fine but frame rate drops when combined with other tasks, analyze if there is sufficient DDR bandwidth. You can try increasing the DDR frequency and analyze the overall tasks to optimize bandwidth usage.

6.3 Output Image with Mosaic

- Confirm if the input image is normal.
- Check if `rk_mpi_venc_test` works fine when used alone.
- Decode using `rk_mpi_vdec_test` and check if it is normal.
- Verify if overclocking is causing the issue. Reduce the corresponding clock frequency and check if it resolves the problem.

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

- If only JPEG/MJPEG encoding produces abnormal output, check if the input is compressed data. You can check by the `afbc_mode` in `dumpsys venc`. If it is `0x0`, it indicates a non-compressed format, while other values indicate compressed formats.
- Try using another encoder to see if it works properly. If a specific encoder (MJPEG/H264/H265) has issues individually, check if the voltage supply for the corresponding encoder meets the requirements. You can try increasing it by 25-50mV to see if there is any improvement.

6.4 Encoding Output Resolution Configuration

- When the scaling function is set using `RK_MPI_VENC_SetChnParam`, the encoder will output according to the resolution set for scaling.
- By default, the encoder uses adaptive resolution encoding, which means that when scaling is not enabled, it encodes the input at the actual resolution. This feature is often used for JPEG snapshot encoding to avoid frequent creation and destruction of encoders. Note that the output buffer for the channel (`u32BufSize`) should be configured based on the buffer size of the maximum resolution to avoid issues when dynamically switching to higher resolutions.

6.5 Bit Rate Control Issues

- Confirm that the input data and the output stream data are normal without artifacts or mosaic.
- Check if the rate control settings are correct. Use `dumpsys venc` to check the `rc_mode` and `gop` information in `venc chn attr`, as well as the Qp and bps settings in `venc chn rc info`.
- Save the encoded output stream and analyze the Qp values. If the Qp is near Qp max, it indicates a complex video sequence, and the actual bitrate is generally higher than the set bitrate. If the Qp is near Qp min, it indicates a simple video sequence, and the actual bitrate is generally lower than the set bitrate. Adjust the corresponding Qp by modifying `Qp min/Qp max` (H264/H265) or `qfactor` (JPEG/MJPEG) in `RK_MPI_VENC_SetRcParam`.
- If the input source is sensor data and there are issues with exceeding the bitrate, ensure that the input image has minimal noise. If the input image quality is poor, the bitrate will generally be higher.

6.6 Support for Compressed Format Input

- H264/H265 only supports compressed formats such as YUV420 and YUV422.
- The JPEG/MJPEG hardware encoder itself does not support AFBC compressed format input. Currently, only the RK3588 platform supports compressed format input within the encoding channel through an expansion module. For other platforms, JPEG/MJPEG compressed input needs to be decompressed by the VPSS module before being sent to VENC. This processing will consume additional bandwidth and hardware resources. Therefore, it is only recommended to use compressed input images in scenarios with a small number of frames, such as snapshot encoding. Otherwise, it is recommended to disable compression output in the upstream module.

6.7 Other Considerations

- It is recommended to set dynamic parameters after creating the channel and before starting encoding to avoid frequent settings during the encoding process.
- After calling `RK_MPI_VENC_StopRecvFrame`, it is recommended to call `RK_MPI_VENC_ResetChn` or retrieve the output stream. Otherwise, there may be issues with the upstream module not functioning properly.

7. RC Configuration Example

- 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

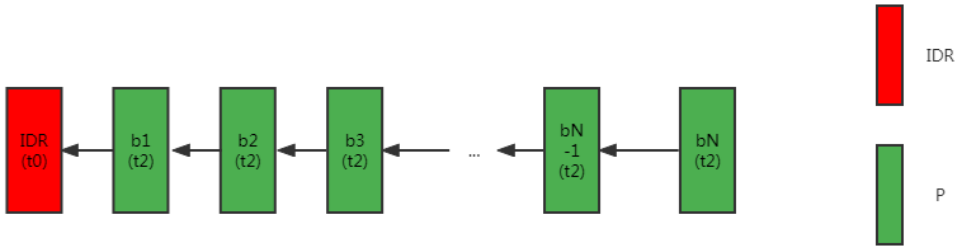
```
VENC_CHN_ATTR_S         stAttr;
memset(&stAttr, 0, sizeof(VENC_CHN_ATTR_S));
RK_MPI_VENC_GetChnAttr(chn, &stAttr);
stAttr.stRcAttr.enRcMode = VENC_RC_MODE_H265CBR;
stAttr.stRcAttr.stH265Cbr.u32Gop = 60;
stAttr.stRcAttr.stH265Cbr.u32SrcFrameRateNum = 30;
stAttr.stRcAttr.stH265Cbr.u32SrcFrameRateDen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

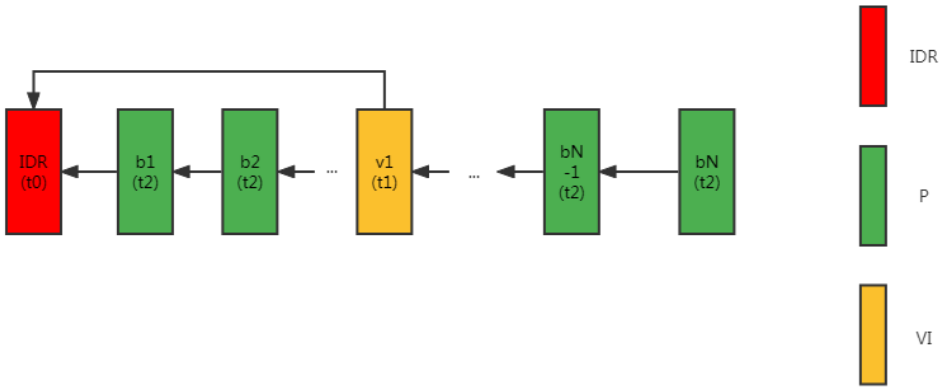
Note:

- `enRcMode` and the corresponding configuration structure should match.
- GOP is recommended to be in integral multiple of the output frame rate.
- Dropping frames is supported, but frame interpolation is not supported.
- Bit rate should be configured based on product requirements.
- `u32MinQp` and `u32MaxQp` correspond to the Qp range for P frames. If the bit rate of P frames is too high, `u32MinQp` can be increased.
- `u32MinIQp` and `u32MaxIQp` correspond to the Qp range for I frames. If the bit rate of I frames is too high, `u32MinIQp` can be increased.
- if `s32DeltIpQp` is negative indicates how much larger the Qp value of I frames is compared to the average Qp of the previous P frames. It can control the bit rate of I frames from becoming too high while sacrificing some quality.
- `s32MaxReEncodeTimes` configures the number of re-encoding attempts. If super-frame re-encoding is enabled and the bit rate exceeds the threshold, the re-encoding process will be triggered. If the number of re-encoding attempts exceeds the set value and the bit rate still exceeds the threshold, the frame will be directly output. The number of re-encoding attempts affects encoding time, so projects with latency requirements should evaluate a reasonable value for the number of re-encoding attempts.

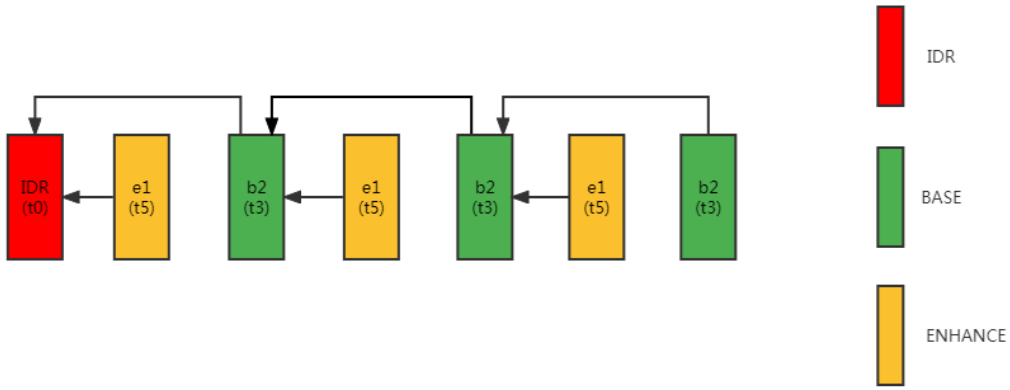
8. GOP Structure



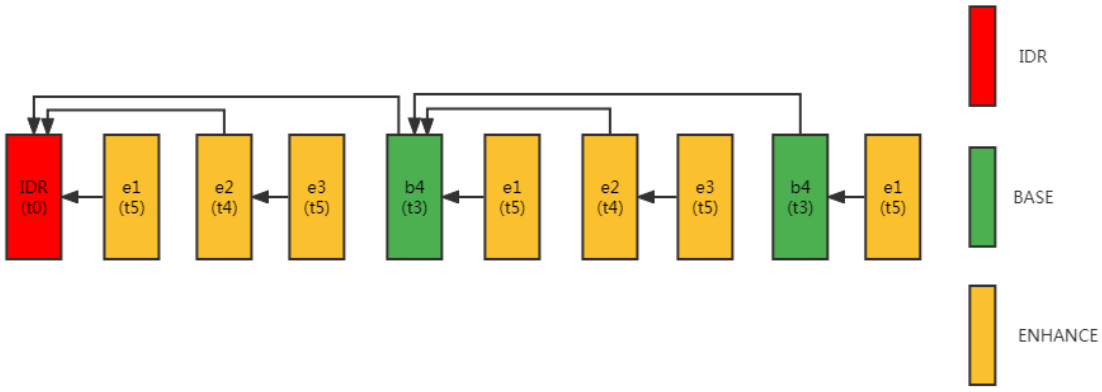
Reference relationship of NORMALP



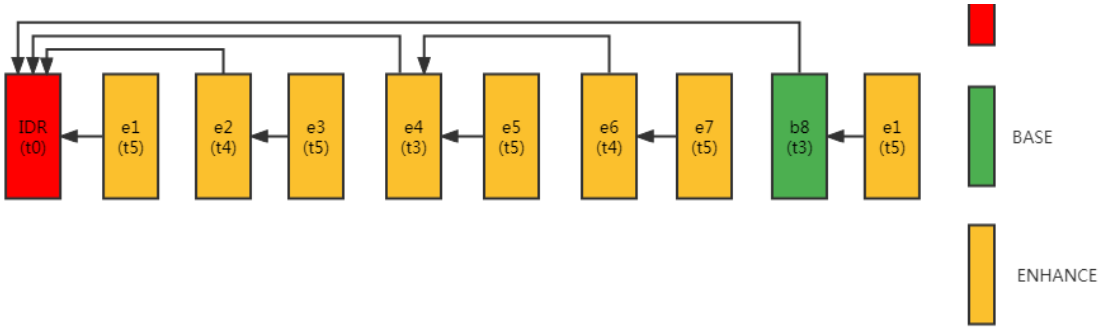
Reference relationship of SMARTP



Reference relationship of TSVC2



Reference relationship of TSVC3



Reference relationship of TSVC4

9. Rate Control

Rate control (RC) is implemented in both hardware and software. The software calculates frame-level quantization parameters (Qp), while the hardware calculates and adjusts Qp at the row and block levels. The collaboration between software and hardware ensures a smooth transition of the bit rate according to requirements. According to information theory, the higher the compression ratio of a video/image, the lower its quality, and vice versa. The bit rate fluctuates depending on the motion between frames and the complexity within frames in different scenes.

9.1 CBR

Constant Bit Rate control(CBR) ensures a smooth transition of the bit rate within the specified time. The stability of the bit rate is evaluated based on two variables:

- Statistical time (`u32StatTime`):

  It is measured in seconds (s). A longer statistical time reduces the impact of bit rate fluctuations per frame, resulting in slower bit rate adjustment and less video quality fluctuation. A shorter statistical time makes single-frame bit rate fluctuations have a larger impact on the overall bit rate, leading to more sensitive adjustment and more noticeable video quality fluctuations.

- Row-level Qp adjustment range (`u32RowQpDelta`)

  It represents the maximum range of row-level adjustment within a frame, based on macroblock rows/CTU rows. A larger adjustment range allows for greater row-level Qp fluctuations, resulting in a more stable bit rate. However, excessive row-level Qp adjustment can affect image quality in scenes with uneven complexity distribution.

With CBR, the resulting bitstream generally stays within ±5% of the target bit rate range. A longer statistical time provides higher stability, and in scenarios with frequent content changes, the average bit rate can usually be controlled within ±10%. This represents a good trade-off between bit rate stability and video quality stability.

9.2 VBR

Variable Bit Rate(VBR) control allows fluctuations in the bit rate within the statistical time to ensure a smooth transition of the encoded video sequence's quality. The encoder provides adjustable parameters such as MaxQp, MinQp, MaxBitrate, and ChangePos.

- MaxQp: The maximum Qp value for the video sequence.
- MinQp: The minimum Qp value for the video sequence.
- MaxBitRate: The maximum bit rate within the statistical time.
- u32BitRate: The baseline bit rate for Qp adjustment.

When the encoding bit rate is lower than u32BitRate, the image's Qp will gradually adjust towards MinQp. Conversely, when the bit rate is higher than u32BitRate, the image's Qp will gradually adjust towards MaxQp.

VBR control is generally effective in reducing file size in simple scenes with minimal or no motion and simple textures. However, it can also ensure that the output bit rate stays within ±10% of the set maximum bit rate in complex scenes with more motion and complex textures.

9.3 AVBR

Adaptive Variable Bit Rate(AVBR) control allows fluctuations in the bit rate within the statistical time to ensure a smooth transition of the video sequence's quality. The AVBR mode internally detects the current scene's motion and stillness states and actively adjusts the bit rate based on the degree of motion, achieving a smooth transition of video quality.

- MaxQp: The maximum Qp value for the video sequence.
- MinQp: The minimum Qp value for the video sequence.
- MaxBitRate: The maximum bit rate within the statistical time.
- u32BitRate and MinStillPercent work together, u32BitRate * MinStillPercent represents the minimum bit rate in stillness conditions. The target bit rate is adjusted between the maximum and minimum bit rates based on the degree of motion. Note: MinStillPercent is a fixed value within the driver and cannot be externally configured. Its mention here is to highlight the difference from the VBR mode.

9.4 SVC

Smart Video Coding(SVC) allows fluctuations in the bit rate within the statistical time to ensure a smooth transition of the video sequence's quality. Internally, it dynamically adjusts the target bit rate based on the motion complexity and image complexity. The smaller the motion and simpler the image complexity, the lower the bit rate; the more complex the motion and higher the image complexity, the higher the bit rate.

During the rate control process, constraints are applied to I-frames based on image complexity and motion complexity. This constraint is applicable to all rate control modes. The frame-level Qp of the current I-frame is limited to the minimum frame-level Qp or above, considering the higher motion complexity and image complexity in the preceding statistically analyzed frames. Similarly, the frame-level Qp of P-frames is limited to the minimum frame-level Qp or above to avoid blurriness or mosaic caused by sudden motion.

The Smart Video Coding process is currently well-supported by RV1103/RV1106 and includes three modes:

- Bitrate Priority
- Quality Priority
- Balanced mode between bit rate and quality.

10. Common Issues Analysis

10.1 Mosaic and Blurriness in Certain Time Segments of the Video Sequence

In some outdoor or indoor scenes, when motion gradually increases, such as when a moving object approaches the camera and its imaging size becomes larger, accurate detection is required. The frame-level Qp of the I-frame should be limited within u32FrmMinIQp to appropriately reduce the bitrate consumption of the I-frame and distribute it to the subsequent P-frames. This helps avoid excessive increases in the frame-level Qp of the P-frames, which can lead to mosaic and blurriness. During the debugging process, it's important to note that in certain resolutions and bitrate constraints, such as 1440p 25fps HEVC with a target bitrate of 3Mbps, it may not be possible to ensure optimal image quality. In such cases, consider reducing the image complexity from the ISP output in combination with encoding to reduce the occurrence of mosaic and blurriness.

10.2 Mosaic Caused by AE Adjustments

Mosaic artifacts can occur when there are variations in lighting or when individuals move in front of the camera, causing AE adjustments. If the AE adjustments are too abrupt, it is advisable to smooth the AE transition. By utilizing the frame-level average pixel values obtained from the VEPU_PP module, it is possible to detect light changes between consecutive frames. In order to alleviate mosaic issues, the frame-level Qp of the I-frame should be limited within u32FrmMinIQp, and the frame-level Qp of the P-frames should be limited within u32FrmMinQp. This allows for a smooth transition in bitrate allocation and helps mitigate mosaic problems.

10.3 Motion Smearing in Regular Scenes

Motion smearing caused by the encoder is mainly due to suboptimal block mode selection on the edges of moving objects, resulting in residual content from the previous frame. To mitigate or resolve motion smearing, the RK_MPI_VENC_EnableMotionDeblur function interface can be called to enable motion deblurring or residue removal.

10.4 Handling of Super Frames

In video sequences, certain frames, especially I-frames, may exceed the predefined threshold, such as a single frame exceeding 1024KB. This can lead to bitrate overshooting, frame loss, and subsequent screen flickering issues. To address this, the RK_MPI_VENC_SetSuperFrameStrategy API interface function can be used. In the wrap mode, frames exceeding the threshold (e.g., single frame exceeding 1024KB) are directly discarded, and the encoding proceeds to the next frame, thereby avoiding frame loss and screen flickering problems.

10.5 Color Shift Caused by Color-to-Black-and-White Transition

During the color-to-black-and-white transition, color frames are typically used as references for black-and-white frames, or the OSD (on-screen display) in black-and-white frames may be in color, resulting in color shifts, especially when Qp is high. To address this issue, the color-to-black-and-white transition can be detected as follows:

- The first black-and-white frame can be encoded as an all I-blocks
- Appropriate Qp adjustments can be made to avoid instantaneous bitrate overshoot, and a smooth Qp transition can be achieved for a few frames following the first black-and-white frame.

- If the OSD needs to switch from color to black-and-white, and if the OSD must be in color, appropriate Qp protection can be applied to the OSD area.

10.6 Chroma Invasion in Regular Scenes

Chroma components have fewer details compared to the luminance component and occupy a larger proportion of low frequencies. Therefore, adjusting the Qp value for the chroma component can be considered by setting the `chroma_qp_offset` to a negative value, typically ranging from -6 to -12. After adjusting the Qp value for the chroma component, the bitrate is less likely to overshoot compared to adjusting the Qp value for the luminance component.

10.7 Breathing Effect

In RV1103/RV1106's wrap mode, it is important to ensure a smooth transition of Qp values between I-frames and P-frames, which can effectively alleviate the breathing effect in relatively static scenes. It is recommended to set `qp_delta_ip` to a value between 2 and 4. Additionally, `u32FrmMinIQp` and `u32FrmMinQp` should be adjusted accordingly. It is generally recommended to set `u32FrmMinQp` between 28 and 33 and `u32FrmMinIQp` between 21 and 26. These adjustments help ensure a smooth transition of video quality between simple and complex scenes and alleviate the breathing effect when transitioning from motion scenes to static scenes.

RV1103/RV1106's offline mode (non-wrap mode) has special handling for the breathing effect, and it can provide better improvement compared to the wrap mode.

10.8 Noise Preservation in Weak Texture or Flat Areas

In static areas, especially those with weak textures or flat regions, visible noise can occur, particularly in low-light conditions. The morphological characteristics of the noise can easily be altered after encoding, and human eyes are particularly sensitive to abnormalities (sudden changes) in weak textures or flat regions. To address this, it is necessary to allocate a portion of the bitrate to these areas by adjusting the Qp values. Additionally, efforts should be made to maintain consistency between these areas and the surrounding regions. Reducing the occurrence of I-blocks can also help ensure more natural noise in these areas.