Intelligent Encoding

User Guide

Issue	**00B02**
Date	**2016-10-20**

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes information about H.264 and H.265 intelligent encoding, which mainly involves the following four parts:

- Part 1: group of pictures (GOP) structure. Different GOP structures apply to different scenarios. The GOP structure can be configured dynamically according to the scenario to optimize encoding performance.
- Part 2: input information of the encoder. The encoder input information APIs can interact with other intelligent analysis modules to protect the analyzed regions of interest (ROIs) or important regions by using QpMap or implement better bit rate control based on the customer's algorithms.
- Part 3: output information of the encoder. The customer can provide more reference input information for intelligent analysis algorithms based on the output information of the encoder.
- Part 4: Cyclic Intra Refresh. This technology does not encode the instantaneous decoding refresh (IDR) frames. The I macroblock is encoded periodically in the P-frames to implement smooth bit rate in special application scenarios.

NOTE

Unless otherwise specified, descriptions about the Hi3519 V101 also apply to the Hi3559 V100.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3519	V100R001
Hi3519	V101R001
Hi3516A	V100R001
Hi3516D	V100R001
Hi3518E	V200R001
Hi3518E	V201R001
Hi3516C	V200R001

Product Name	Version
Hi3536	V100R001
Hi3521A	V100R001
Hi3531A	V100R001
Hi3520D	V300R001
Hi3516C	V300R001
Hi3559	V100R003

Intended Audience

This document is intended for:

- Technical support engineers
- Board software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 00B02 (2016-10-20)

This issue is the second draft release, which incorporates the following changes:

The description of **Purpose** in **About This Document** is modified.

Chapter 1 GOP Structure and Application Scenarios

In section 1.9, table 1-3 is modified.

Chapter 2 Input Information of the Encoder

In section 2.1.2, figure 2-2 and figure 2-3 are modified.

Issue 00B01 (2016-06-15)

This issue is the first draft release.

Contents

Tables

1 GOP Structure and Application Scenarios

1.1 GOP Mode Glossary

Table 1-1 GOP mode glossary

GOP Mode	Number of Reference Frames that Can Be Referenced by the P-frame Simultaneously	Remarks
SingleP (NormalP)	1	A P-frame uses one reference frame.
AdvSingleP	1	• A common P-frame uses one short-term reference frame. • A virtual I-frame uses one long-term reference frame.
SmartP	2	A P-frame uses a long-term reference frame and a short-term reference frame.
AdvSmartP	2	A P-frame uses a long-term reference frame and a short-term reference frame.
DualP	2	A P frame uses two reference frames.
BiPredB	2	• A P frame uses two reference frames. • A B-frame uses a forward reference frame and a backward reference frame.

1.2 GOP Structure and Usage in SingleP (NormalP) Mode

NOTE

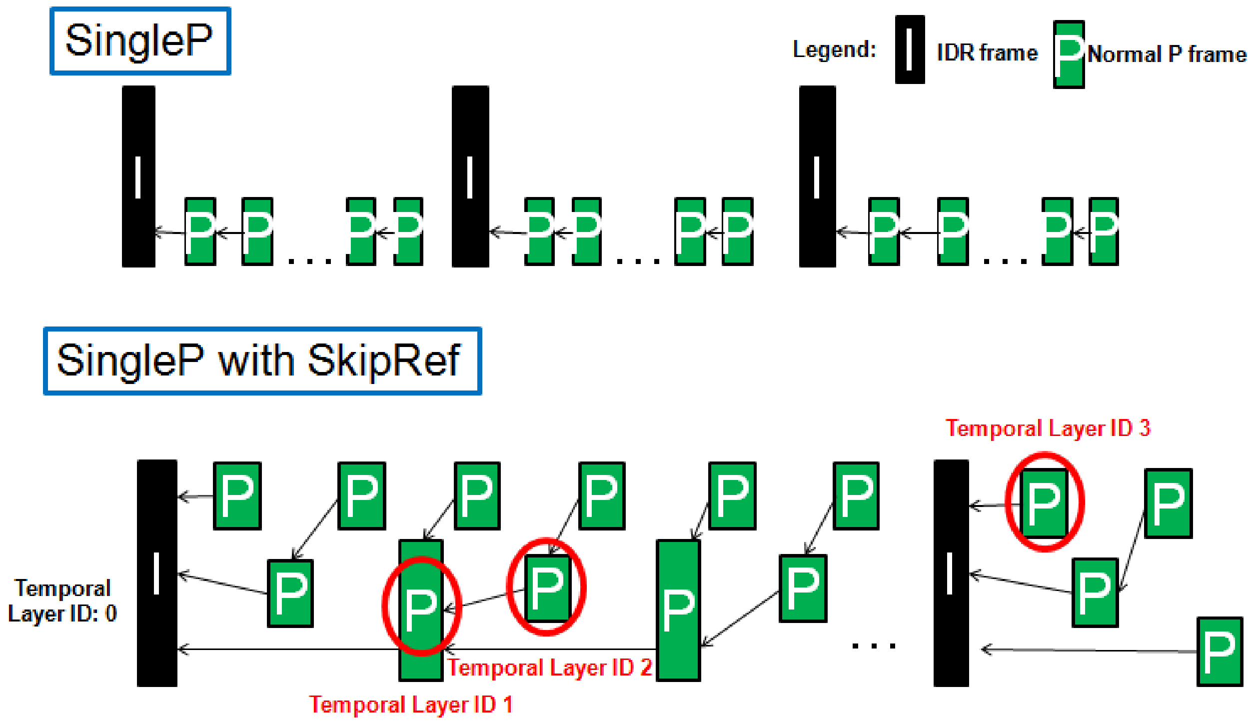
SingleP (NormalP) is a most common GOP structure. All HiSilicon chips support this mode unless otherwise specified.

1.2.1 Structure

- The reference relationship of the SingleP (NormalP) mode is simple. Each P-frame uses one forward reference frame.
- The SingleP (NormalP) mode can be used in any scenarios.

Figure 1-1 shows the GOP structure in SingleP (NormalP) mode.

Figure 1-1 GOP structure in SingleP (NormalP) mode



1.2.2 Usage

- For Hi3519 V100, Hi3519 V101, and Hi3516C V300:

  [API]

  HI_MPI_VENC_CreateChn

  [Parameters]

  VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_NORMALP

  VENC_CHN_ATTR_S::stGopAttr.stNormalP.s32IPQpDelta (The recommended value is 3. A larger value indicates a larger I-frame and better quality of the I-frame.

- For Hi3516A, Hi3516D, Hi3518E V200, Hi3518E V201, Hi3516C V200, Hi3536, Hi3521A, Hi3531A, and Hi3520D V300, the SingleP (NormalP) mode is used by default when an encoding channel is created, and no parameter needs to be configured.

1.3 GOP Structure and Usage in AdvSingleP Mode

NOTE

This mode is supported only by Hi3516A V100, Hi3516D V100, Hi3518E V200, Hi3518E V201, Hi3516C V200, Hi3536 V100, Hi3521A V100, Hi3531A V100, and Hi3520D V300.

During the H.264 encoding of P-frames for Hi3519 V100, two reference frames are not supported. If the GOP mode is set to the SmartP mode, the AdvSingleP mode is used by default.

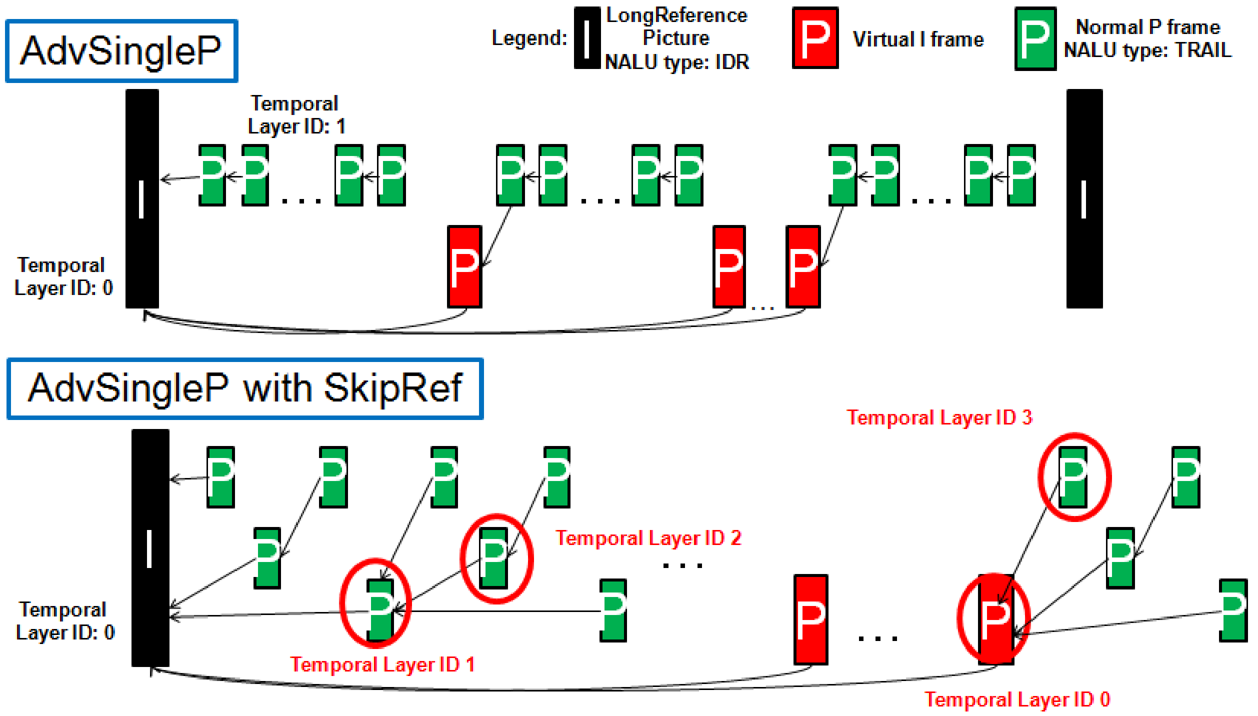
1.3.1 Structure

In AdvSingleP mode, the virtual I-frame is added based on the GOP structure in SingleP (NormalP) mode. The virtual I-frame is actually a common P frame, but it uses the IDR frame but not the previous P-frame as the reference frame. The virtual I-frame is added to resolve some issues caused by enlarging the IDR frame interval in SingleP (NormalP) mode. The caused issues include the following:

- The video cannot be resumed a long time after the frame loss or code errors.
- For video on-demand (VOD) from a certain time point, frames before the VOD time point need to be decoded until an IDR frame, which results in long decoding time and waste in decoding performance.

Figure 1-2 shows the GOP structure in AdvSingleP mode.

Figure 1-2 GOP structure in AdvSingleP mode



1.3.2 Usage

- For Hi3519 V100 H.264:

  [API]
  HI_MPI_VENC_CreateChn
  [Parameters]

  – VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_SMARTP
  – VENC_CHN_ATTR_S::stGopAttr.stSmartP.u32BgInterval = 1200; // 30fps, 40seconds
  – VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32BgQpDelta = 7
  – VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32ViQpDelta = 2
  – VENC_CHN_ATTR_S::stRcAttr.u32Gop = 30; // virtual I interval
  – VENC_CHN_ATTR_S::stRcAttr.u32StatTime = 40; // 40 seconds

- For Hi3516A V100, Hi3516D V100, Hi3518E V200, Hi3518E V201, Hi3516C V200, Hi3536 V100, Hi3521A V100, Hi3531A V100, and Hi3520D V300:

  [API]
  HI_MPI_VENC_CreateChn
  [Parameters]

  – VENC_CHN_ATTR_S::stRcAttr::u32Gop = 1200; // 30fps, 40 seconds
  – VENC_CHN_ATTR_S::stRcAttr::u32StatTime = 40; // 40 seconds

  [API]
  HI_MPI_VENC_SetRefParamEx
  [Parameters]

  – VENC_PARAM_REF_EX_S::bVirtualIEnable = HI_TRUE
  – VENC_PARAM_REF_EX_S::u32VirtualIInterval = 30; // 1 second
  – VENC_PARAM_REF_EX_S::s32VirtualIQpDelta = 2

1.4 GOP Structure and Usage in DualP Mode

NOTE
This mode is supported only by Hi3519 V100 H.265, Hi3519 V101, and Hi3516C V300.

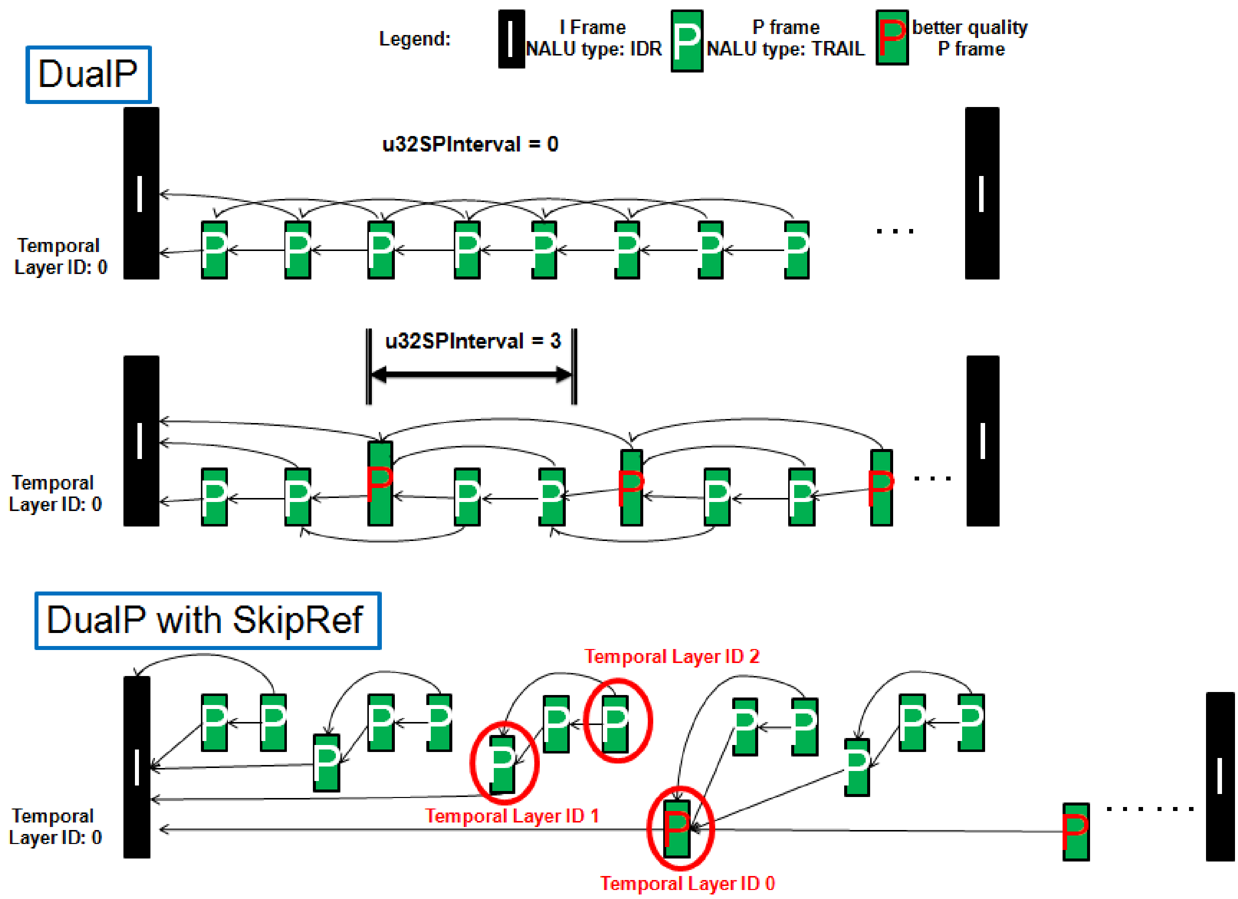
1.4.1 Structure

- SP indicates a special P-frame (also called an SP frame). It is recommended that the QP value of the SP frame be less than that of any other P-frame. If **u32SpInterval** is **0**, the SP frame is not supported.
- In DualP mode, the P-frame uses the nearest two forward reference frames. Using the time domain correlation of more reference frames enhances the encoding compression performance. This mode is mainly used in motion scenarios with low delay requirements.

The compression performance of the DualP mode is lower than that of the BipredB mode but higher than that of the SingleP mode. The DualP mode features no encoding and decoding delay because the two reference frames are forward reference frames.

Figure 1-3 shows the GOP structure in DualP mode.

Figure 1-3 GOP structure in DualP mode



1.4.2 Usage

The DualP mode is supported only by Hi3519 V100, Hi3519 V101, and Hi3516C V300.

[API]

HI_MPI_VENC_CreateChn

[Parameters]

VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_DUALP

P-frames with better quality (that is, SP frames) can be encoded periodically to optimize the picture quality.

The interval between SP frames can be configured by setting **VENC_CHN_ATTR_S::stGopAttr.stDualP.u32SPInterval**.

1.5 GOP Structure and Usage in SmartP Mode

NOTE

- This mode is supported only by Hi3519 V100, Hi3519 V101, and Hi3516C V300.
- The GOP structure in this mode is the same as that in AdvSingleP mode for Hi3519 V100 H.264.

1.5.1 Structure

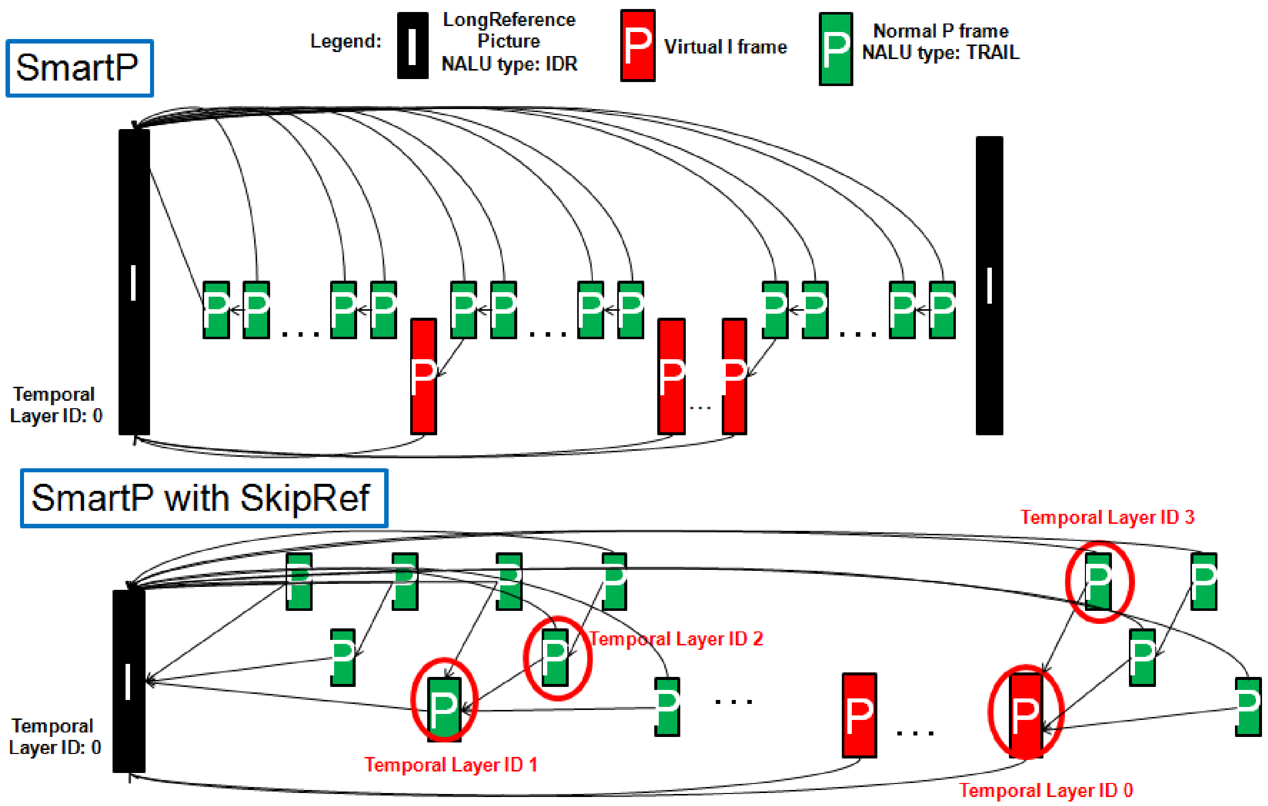
In SmartP mode, the P-frame uses the IDR frame (long-term reference frame) and forward reference frame (short-term reference frame) as reference frames. The time domain correlation of two reference frames is used to improve the encoding compression performance. This mode is mainly used in the monitoring scenario.

In the monitoring scenario, the camera is fixed at a position, and the human and objects in the scenario may be static or moving.

- In static regions, the time domain correlation of the long-term reference frame and the current frame significantly reduces the bit rate and respiratory and smearing effects.
- In motion regions, motion estimation is performed by using the short-term reference frame. In SmartP mode, the IDR frame interval is prolonged, and the virtual I-frame is inserted periodically, which significantly reduces the bit rate by 30% to 50% in the monitoring scenario and improves the picture quality. However, this mode is not applicable to scenarios in which the camera can move.

Figure 1-4 shows the GOP structure in SmartP mode.

Figure 1-4 GOP structure in SmartP mode



1.5.2 Usage

The SmartP mode is supported only by Hi3519 V100, Hi3519 V101, and Hi3516C V300.

[API]

HI_MPI_VENC_CreateChn

[Parameters]

- VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_SMARTP
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.u32BgInterval = 1200; // 30fps, 40seconds
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32BgQpDelta = 7
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32ViQpDelta = 2
- VENC_CHN_ATTR_S::stRcAttr.u32Gop = 30; // virtual I interval
- VENC_CHN_ATTR_S::stRcAttr.u32StatTime = 40; // 40 seconds

1.6 GOP Structure and Usage in AdvSmartP Mode

NOTE

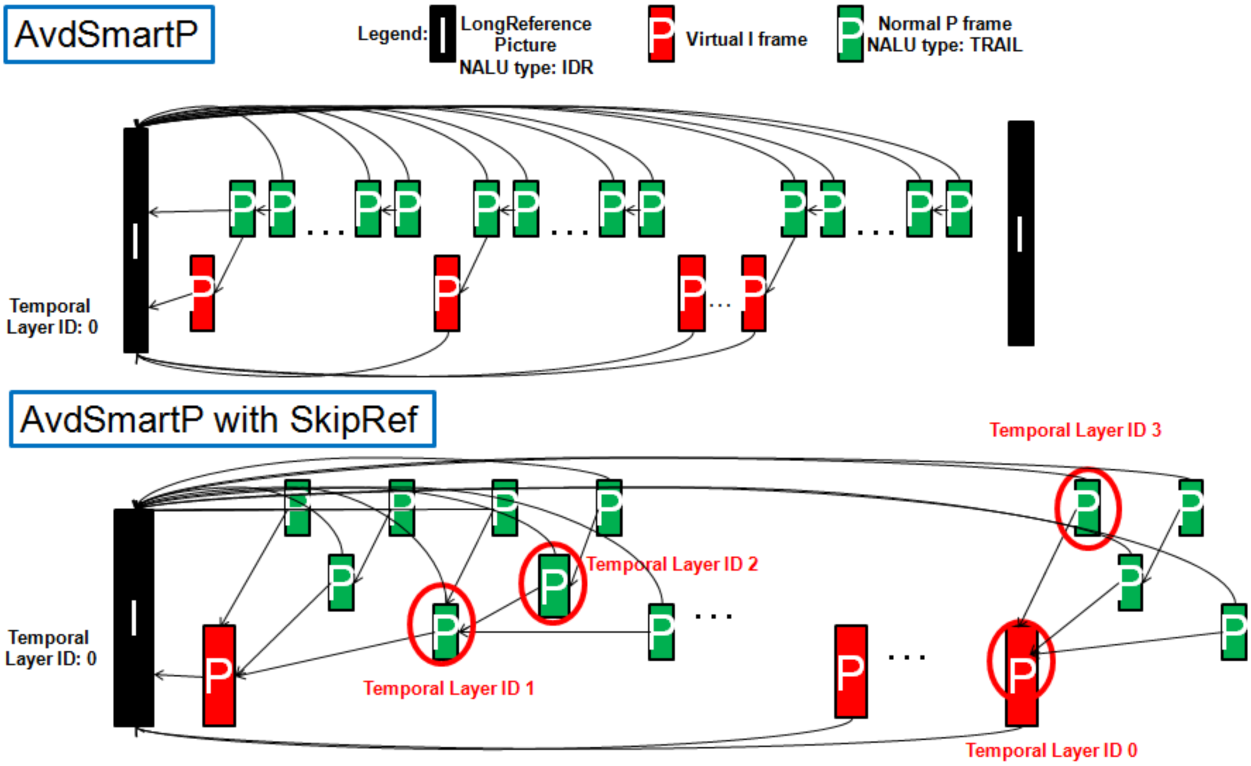
This mode is supported only by Hi3519 V100, Hi3519 V101, and Hi3516C V300 H.265.

1.6.1 Structure

- The purple frames are encoded as IDR frames and used as long-term reference frames. The red frames are encoded as VI frames (virtual I-frames, which are actually common P-frames). The virtual I-frames use only the IDR frames as reference frames, and it is recommended that the QP value of a virtual I-frame be less than that of other P-frames.
- The GOP structure in AdvSmartP mode is almost the same as that in SmartP mode, and the only difference is that the long-term reference frame IDR frame is internally generated by the encoder and therefore not displayed by the player.

Figure 1-5 shows the GOP structure in AdvSmartP mode.

Figure 1-5 GOP structure in AdvSmartP mode



1.6.2 Usage

The AdvSmartP mode is supported only by Hi3519 V100, Hi3519 V101, and Hi3516C V300.

[APIs]

- HI_MPI_VENC_CreateChn
- HI_MPI_VENC_EnableAdvSmartP

[Parameters]

- VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_SMARTP
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.u32BgInterval = 1200; // 30fps, 40seconds
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32BgQpDelta = 7
- VENC_CHN_ATTR_S::stGopAttr.stSmartP.s32ViQpDelta = 2
- VENC_CHN_ATTR_S::stRcAttr.u32Gop = 30;   // virtual I interval
- VENC_CHN_ATTR_S::stRcAttr.u32StatTime = 40;   // 40 seconds

1.7 GOP Structure and Usage in BiPredB Mode

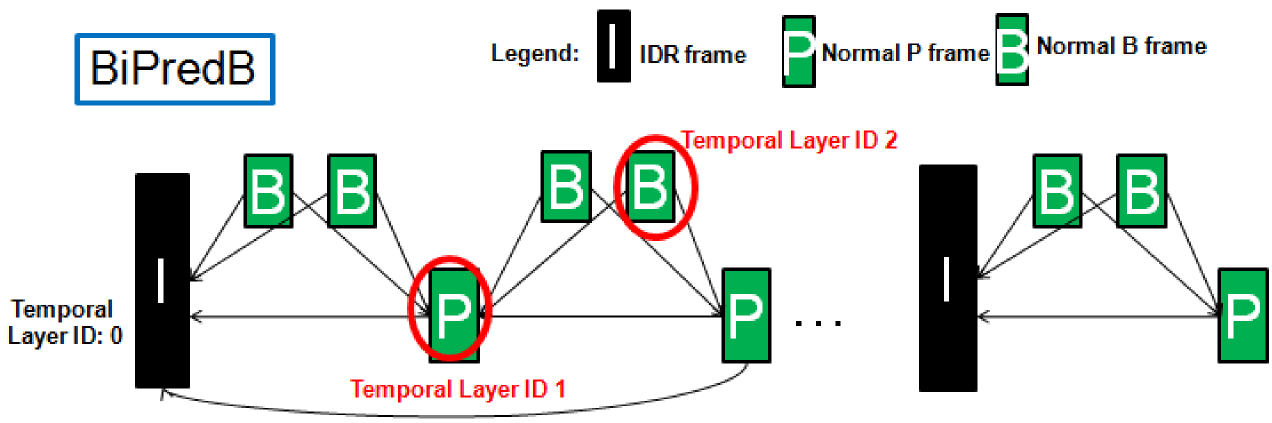
1.7.1 Structure

NOTE
This mode is supported only by Hi3519 V101.

- **u32BFrmNum** indicates the number of B-frames between the IDR frame and P-frame or between the P-frames. For example, in Figure 1-6, **u32BFrmNum** is **2**. The last frame of each GOP must be a P-frame, and the number of B-frames before it may not meet the requirement of **u32BFrmNum**.
- In BiPredB mode, one to three B-frames can be inserted between two adjacent P-frames. The B-frames are not referenced. One forward reference frame and one backward reference frame are used. For motion scenarios, bidirectional prediction implements better motion estimation and supports weighted prediction, and therefore improves the compression performance of the encoder.

Figure 1-6 shows the GOP structure in BiPredB mode.

Figure 1-6 GOP structure in BiPredB mode



1.7.2 Usage

[API]

HI_MPI_VENC_CreateChn

[Parameters]

- VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_BIPREDB
- VENC_CHN_ATTR_S::stGopAttr.stBipredB.u32BFrmNum = 2
- VENC_CHN_ATTR_S::stGopAttr.stBipredB.s32IPQpDelta = 3
- VENC_CHN_ATTR_S::stGopAttr.stBipredB.s32BQpDelta = –2

1.8 Principles and Usage of Cyclic Intra Refresh

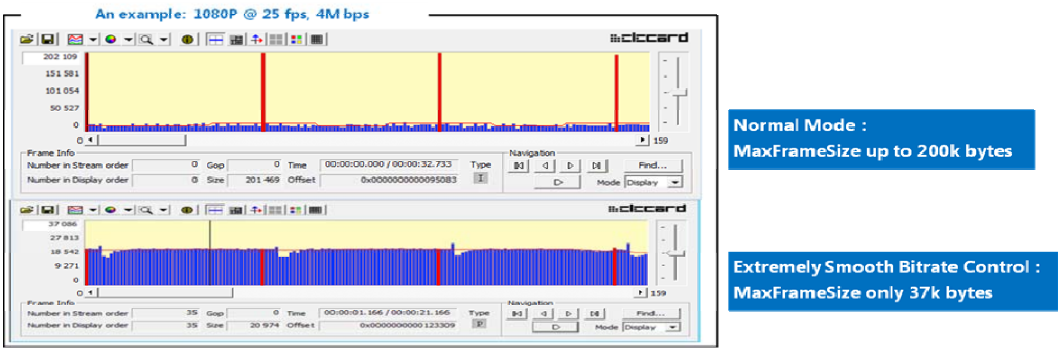
NOTE

- Hi3516A V100, Hi3516D V100, Hi3518E V200, Hi3518E V201, and Hi3516C V200 support only the H.264 protocol.
- Hi3519 V100, Hi3519 V101, and Hi3516C V300 support both the H.264 and H.265 protocols.

1.8.1 Principles

In common scenarios, the size of an IDR frame is many times larger than that of a P-frame, especially in scenarios with small motions, in which the size of an IDR frame is tens or even a hundred times of a P-frame. During network transmission, wireless transmission in particular, IDR frames cause transient network shocks, which result in frame loss and increase the delay. For general encoding technologies, the QP value of the IDR frame is increased to decrease the size of the IDR frame. However, this also reduces the quality of the IDR frame and causes the respiratory effect. The Cyclic Intra Refresh technology does not change the quality of the IDR frame. It distributes the intra largest coding units (LCUs)/macroblocks of an IDR frame into multiple P-frames, which makes the frame sizes relatively even. For example, by using the Cyclic Intra Refresh technology, the maximum frame size in the 1080P@25 fps monitoring scenario with 4 Mbit/s bit rate can be decreased from 200 KB to 37 KB, ensuring smooth bit rate, as shown in Figure 1-7.

Figure 1-7 Cyclic Intra Refresh effect



1.8.2 Usage

[API]

HI_MPI_VENC_SetIntraRefresh

[Parameters]

- VENC_PARAM_INTRA_REFRESH_S::bRefreshEnable = HI_TRUE
- VENC_PARAM_INTRA_REFRESH_S::bISliceEnable = HI_TRUE
- VENC_PARAM_INTRA_REFRESH_S::u32RefreshLineNum; // Number of refreshed macroblock/LCU lines of each frame
- VENC_PARAM_INTRA_REFRESH_S::u32ReqIQp; // QP value for requesting the I-frame

Advantages

- The bit rate is smooth, and the impact on the network is small. This technology applies to the wireless network transmission scenario.
- The encoding, decoding, and network delay is small.
- This technology does not reduce the quality of I-frames or cause serious respiratory effect.

Limitations

- The first few frames of the video are incomplete because the intra LCUs/macroblocks are distributed in multiple P-frames.
- The decoder must support decoding of streams without IDR frames.
- This technology supports the GOP structure of only the SingleP (NormalP) mode.
- This technology mainly applies to scenarios with high requirements on bit rate smoothness. It does not reduce the bit rate and therefore does not apply to low bit rate scenarios.

1.9 Memory Usage, Delay, Application Scenarios, and Compatibility of the GOP Structure

NOTE

- **PicSize** is the size of the frame buffer for reference frames, and **AdvInfoSize** is the size of the information frame buffer (the picture format is YUV420) for AdvSmartp.
- For details about how to calculate **PicSize** and **AdvInfoSize**, see the following section.
- **SrcPicSize** is the size of the source picture before encoding. Some source pictures need to be buffered because B-frames are delayed.
- Delay occurs on both the encoding end and decoding end in BiPredB mode, and it does not occur in other modes.

Table 1-2 Memory usage, delay, and application scenarios

GOP Mode	DDR Usage		Delay		Application Scenarios
	H.264&H.265 Enc	H.264&H.265 Dec	H.264&H.265 Enc	H.264&H.265 Dec	
SingleP	2 x PicSize	2 x PicSize	N/A	N/A	Typical scenarios
AdvSingleP	3 x PicSize	3 x PicSize	N/A	N/A	Monitoring scenarios with fixed cameras; SmartP mode not supported by the chip
SmartP	3 x PicSize	3 x PicSize	N/A	N/A	Monitoring scenarios with fixed cameras

GOP Mode	DDR Usage		Delay		Application Scenarios
	H.264&H.265 Enc	H.264&H.265 Dec	H.264&H.265 Enc	H.264&H.265 Dec	
AdvSmartP	3 x PicSize + AdvInfoSize	3 x PicSize	N/A	N/A	Monitoring scenarios with fixed cameras
DualP	3 x PicSize	3 x PicSize	N/A	N/A	Motion scenarios with moving cameras, such as the event data recorder and handheld DV
BiPredB (*N* indicates the number of B-frames.)	3 x PicSize + *N* x SrcPicSize	3 x PicSize	*N* frames	1 frame	Motion scenarios with moving cameras, such as the event data recorder and handheld DV

Calculating PicSize

- The size of each VB of the encoding frame buffer (reference frame and reconstructed frame) is calculated as follows:

  PicSize = **YHeaderSize** + **CHeaderSize** + **YSize** + **CSize** + **PmeSize** + **PmeInfoSize** + **TmvSize**

- Table 1-3 describes how to calculate the sizes of sub items of the frame buffer for Hi3519 V101, Hi3519 V100, or Hi3516C V300.

Table 1-3 Calculating the sizes of sub items of the frame buffer for Hi3519 V101, Hi3519 V100, or Hi3516C V300

Sub Item of the Frame Buffer		H.264	H.265
YHeaderSize	Compressed	(align(width,256)>>8) x (align(height,16)>>4) x 32	(align(width,64)>>6) x (align(height,64)>>6) x 32
	Uncompressed	0	0
YSize	Compressed	(((align(width,16)>>4)-1)>>2+1)*(((align(width,16)>>4)-1)+1)*4*256	align(width,64) x align(height,64)
	Uncompressed	align(width,16) x align(height,16)	align(width,16) x align(height,16)
CHeaderSize		YHeaderSize	YHeaderSize
CSize		align(width,16)* align(height,16)	YSize/2

Sub Item of the Frame Buffer	H.264	H.265
PmeSize	(align(width,64)>>6) x (align(height,16)>>4)<<6	(align(width,64)>>6) x (align(height,64)>>6)<<8
PmeInfoSize	((align(width,2048)>>4) x (align(height,16)>>4) + 7)>>3	(align(width,512)>>6) x (align(height,64)>>6)<<2
TmvSize	VENC_GOPMODE_BIPREDB: (align(width,64)>>6) x (align(height,16)>>4)<<7 Non VENC_GOPMODE_BIPREDB:0	(align(width,64)>>6) x (align(height,64)>>6)<<7

- Table 1-4 describes how to calculate the sizes of sub items of the frame buffer for Hi3516A V100, Hi3516D V100, Hi3518E V200, Hi3518E V201, Hi3516C V200, Hi3536 V100, Hi3521A V100, Hi3531A V100, or Hi3520D V300.

Table 1-4 Calculating the sizes of sub items of the frame buffer for Hi3516A V100, Hi3516D V100, Hi3518E V200, Hi3518E V201, Hi3516C V200, Hi3536 V100, Hi3521A V100, Hi3531A V100, or Hi3520D V300

Sub Item of the Frame Buffer		H.264	H.265
YHeaderSize	Compressed	(align(width,256)>>8) x (align(height,16)>>4) x 32	(align(width,64)>>6) x (align(height,64)>>6) x 32
	Uncompressed	0	0
YSize	Compressed	(align(width,16) + 48) x align(height,16)	align(width,64) x align(height,64)
	Uncompressed	align(width,16) x align(height,16)	align(width,16) x align(height,16)
CHeaderSize		YHeaderSize	YHeaderSize
CSize		YSize/2	YSize/2
PmeSize		0	(align(width,64)>>6) x (align(height,64)>>6)<<8
PmeInfoSize		0	0
TmvSize		0	(align(width,64)>>6) x (align(height,64)>>6)<<6

Calculating AdvInfoSize

AdvInfoSize = align(Width,16) x align(Height,16) x 3/2 + (align(Width,16)/64 + 1) x (align(Height,16)/64 + 1) x 74 x 16

Compatibility

Table 1-5 describes the back-end product compatibility of HiSilicon.

Table 1-5 Back-end product compatibility of HiSilicon

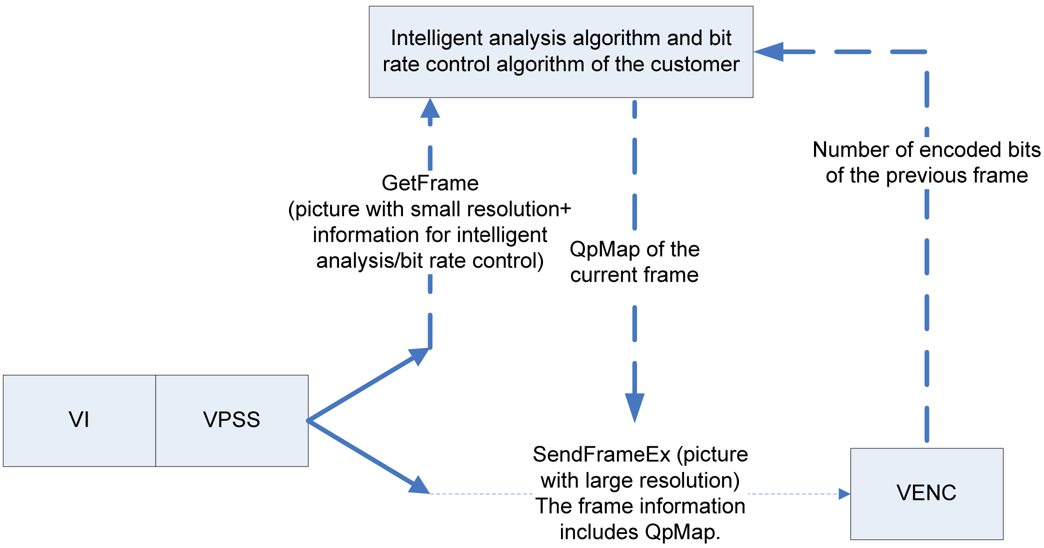
GOP Mode	Hi_H264 Decoder	Hi_H265 Decoder	Hi3536		Hi3531A/Hi3521A/Hi3520D V300	Hi3535
	H.264	H.265	H.264	H.265	H.264	H.264
SingleP	Yes	Yes	Yes	Yes	Yes	Yes
AdvSingleP	Yes	Yes	Yes	Yes	Yes	Yes
SmartP	Yes	Yes	Yes	Yes	Yes	Yes
AdvSmartP	No	Yes	No	Yes	No	No
DualP	Yes	Yes	Yes	Yes	Yes	Yes
BiPredB	No	Yes	Yes	Yes	No	Yes
Cyclic Intra Refresh	Yes	Yes	Yes	Yes	Yes	No

2 Input Information of the Encoder

2.1 QpMap/SkipMap API Definitions

Figure 2-1 shows the overall design system block diagram of the QpMap encoding solution.

Figure 2-1 Overall design of the QpMap encoding solution



NOTE

This solution is supported only by Hi3519 V101 and Hi3516C V300.

The solution implementation details and precautions are described as follows:

- To ensure that SrcPic and QpMap of each frame are synchronized, the solution needs to be implemented in a user-mode App. The App fetches pictures from the VI/VPSS in user mode, calculates the QpMap of each macroblock in a frame by intelligent analysis and frame-level bit rate control, and configures the SrcPic and QpMap by using the SendFrameEx API. The frame rate control and frame-level bit rate control are bypassed within the SDK, and the intelligent analysis and frame-level bit rate control algorithms are developed by the customer.

- There are no picture and stream buffers within the HiSilicon SDK. However, scheduling delay is inevitable. The customer can assess the solution performance and delay.
- For the H.264 and H.265 protocols, QpMap is configured based on the 16x16 block size. However, the encoding coding units (CUs) of H.265 may be greater than 16x16. Therefore, the same CU may correspond to multiple QP values. In this case, HiSilicon provides an API for selecting any of the following three modes:
  - Using the maximum value of multiple QP values
  - Using the minimum value of multiple QP values
  - Using the average value of multiple QP values
- The configuration of QpMap supports the relative QP mode and absolution QP mode.
  - In relative QP mode, the code control (macroblock level) calculated based on the internal logic of the encoder also takes effect. The QP input externally by using the QpMap is a variable Qpdelta, which is overlaid on the basic of the original code control (macroblock level).
  - In absolution QP mode, the QP value of each encoding block is specified by external user input but not the internal macroblock level code control algorithm.
- The internal recoding and frame dropping mechanisms of the encoder may not be used.
- The HiSilicon SDK provides the following information for each picture frame:
  - ISP-related information
  - Number of encoded bytes of the previous frame

2.1.1 QpMap Table Memory Arrangement

The QP arrangement in the QpMap table for H.265 encoding differs from that for H.264 encoding because the basic encoding unit differs. Figure 2-2 and Figure 2-3 show the QP arrangement for H.264 and H.265 encoding respectively.

Figure 2-2 LCU arrangement in the QpMap table for H.265 encoding

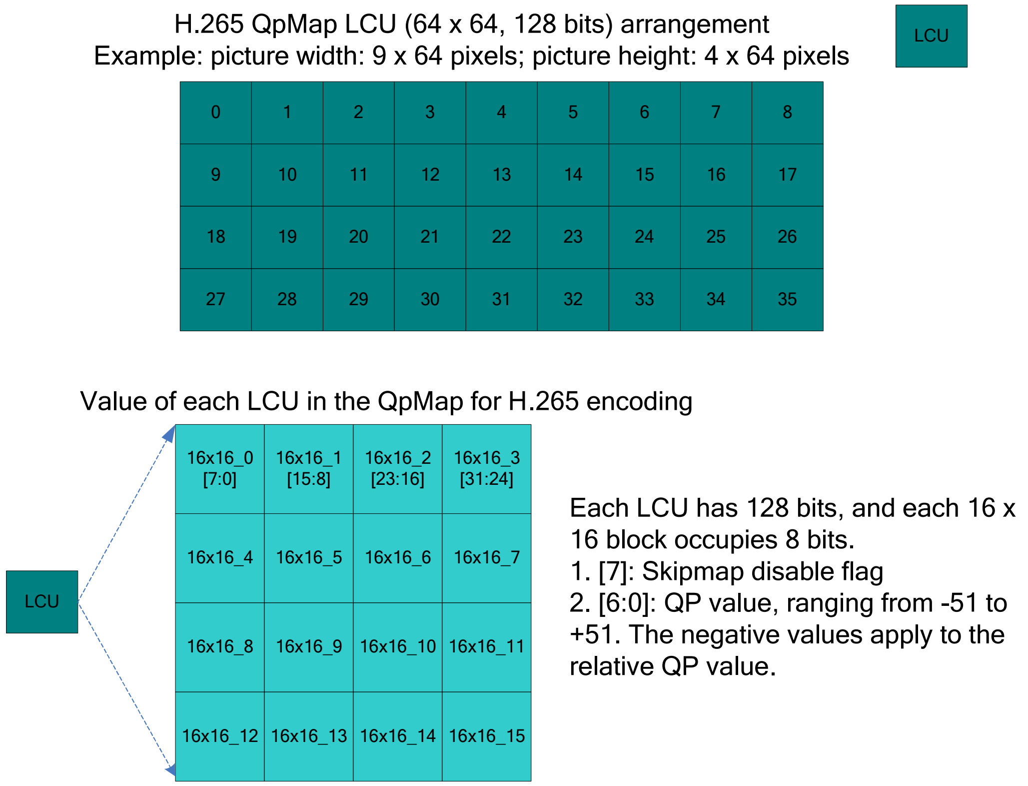
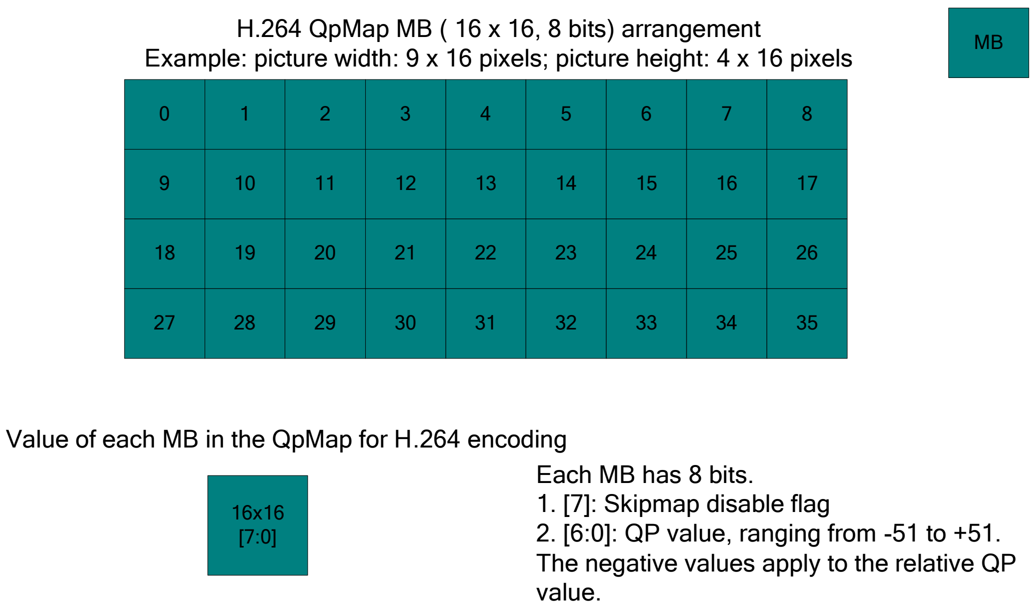


Figure 2-3 MB arrangement in the QpMap table for H.264 encoding



2.2 Definitions of Encoding APIs

2.2.1 Bit Rate Control APIs

```
typedef enum hiVENC_RC_MODE_E
{
   VENC_RC_MODE_H264CBR = 1,
   VENC_RC_MODE_H264VBR,
   VENC_RC_MODE_H264FIXQP,
   VENC_RC_MODE_H264QPMAP,
   VENC_RC_MODE_MJPEGCBR,
   VENC_RC_MODE_MJPEGVBR,
   VENC_RC_MODE_MJPEGFIXQP,
   VENC_RC_MODE_H265CBR,
   VENC_RC_MODE_H265VBR,
   VENC_RC_MODE_H265FIXQP,
   VENC_RC_MODE_H265QPMAP,
   VENC_RC_MODE_BUTT,
}VENC_RC_MODE_E;

typedef struct hiVENC_RC_ATTR_S
{
   VENC_RC_MODE_E enRcMode;   /*the type of rc*/
   union
   {
         VENC_ATTR_H264_CBR_S      stAttrH264Cbr;
      VENC_ATTR_H264_VBR_S    stAttrH264Vbr;
      VENC_ATTR_H264_FIXQP_S      stAttrH264FixQp;
      VENC_ATTR_H264_QPMAP_S  stAttrH264QpMap;
      VENC_ATTR_MJPEG_CBR_S   stAttrMjpegeCbr;
      VENC_ATTR_MJPEG_VBR_S   stAttrMjpegeVbr;
      VENC_ATTR_MJPEG_FIXQP_S stAttrMjpegeFixQp;
      VENC_ATTR_H265_CBR_S    stAttrH265Cbr;
      VENC_ATTR_H265_VBR_S    stAttrH265Vbr;
      VENC_ATTR_H265_FIXQP_S  stAttrH265FixQp;
      VENC_ATTR_H265_QPMAP_S  stAttrH265QpMap;
   };
   HI_VOID* pRcAttr ;     /*the rc attribute which could be specified by user*/
}VENC_RC_ATTR_S;
```

- The QpMap is used as the adaptive ROI mode, the RC mode is set to the constant bit rate (CBR) or variable bit rate (VBR), and the QpMap uses the relative QP mode.
- The QpMap is used as external bit rate control, and the RC mode is set to the QpMap mode. The internal RC algorithm is bypassed, and the frame-level bit rate control and macroblock-level bit rate control are implemented externally in this case.

2.2.2 Encoding Image Transmission API

```
typedef struct hiUSER_RC_INFO_S
{
   HI_U32 u32BlkStartQp;
   HI_U32 u32QpMapPhyAddr;
} USER_RC_INFO_S;
```

```
typedef struct hiUSER_FRAME_INFO_S
{
   VIDEO_FRAME_INFO_S     stFrame;
   USER_RC_INFO_S         stUserRcInfo;
} USER_FRAME_INFO_S;

/*-1:bolck  0:nonblock  >0 : overtime  */
HI_S32 HI_MPI_VENC_SendFrameEx(VENC_CHN VeChn, USER_FRAME_INFO_S *pstFrame ,HI_S32
s32MilliSec);
```

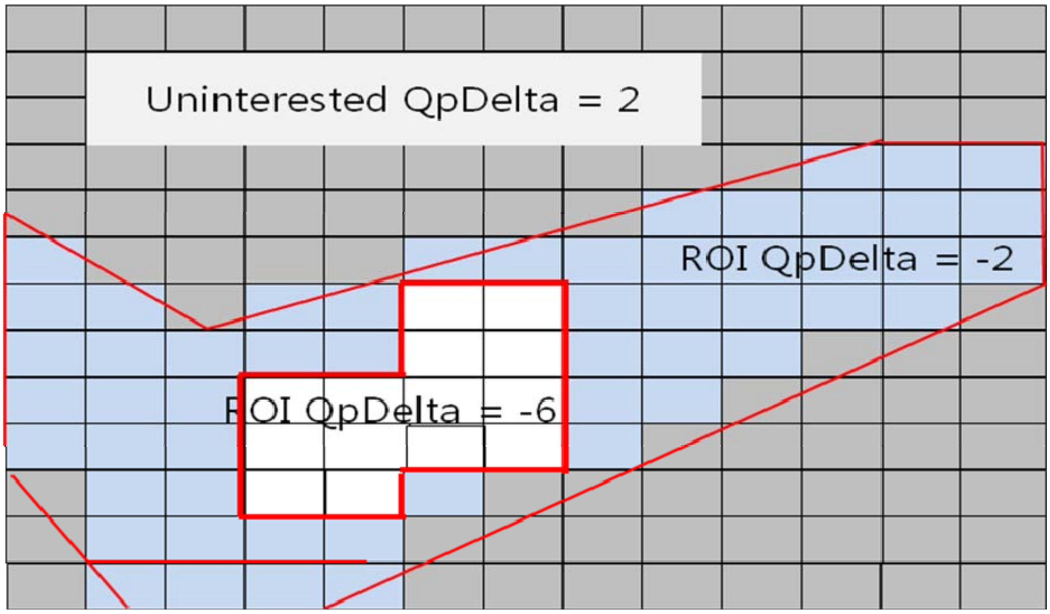
2.3 Adaptive ROI by Using the QpMap/SkipMap

Figure 2-4 shows an example of an ROI.

Figure 2-4 ROI example

The QpMap is used to adjust the QP values of the ROI and non-ROI, as shown in Figure 2-5.

Figure 2-5 Adjusting the QP values of the ROI and non-ROI by using the QpMap



You can draw ROIs with any shape and any quantity by using the QpMap. You can also improve or reduce the picture quality in ROIs by reducing or increasing the QP values respectively. In this mode, the relative QP adjustment mode is required.

2.4 External Bit Rate Control by Using QpMap/SkipMap

During external bit rate control by configuring the QpMap, the QP values can be configured in relative or absolute mode. The QP values of all LCUs/MBs are externally specified, and the frame-level bit rate control does not take effect.

3 Output Information of the Encoder

3.1 SSE and PSNR Information

NOTE

The SSE and peak signal to noise ratio (PSNR) information is supported only by Hi3519 V101 H.265 and Hi3516C V300.

- PSNR indicates the signal-to-noise ratio of the entire frame picture. A larger value indicates better picture quality.
- Besides measuring the PSNR of the entire frame picture, the encoder also supports the output of SSE information that reflects local picture quality. The encoder supports the output of SSE of a maximum of eight regions. You can calculate the region SNR based on the SSE and the size of the eight regions.

3.2 HeaderBits and ResidualBits Information

NOTE

The **HeaderBits** and **ResidualBits** information is supported only by Hi3516C V300.

HeaderBits and **ResidualBits** are CU-level information. **HeaderBits** contains the block type and inter-frame and intra-frame prediction information, and **ResidualBits** contains the residual information after transformation and quantization.

- A larger **HeaderBits** value indicates:
  - More picture textures and finely divided blocks
  - More complex motions, more relative motions, and more MV information.
- A larger **ResidualBits** value indicates:
  - More irregular motions or excessive motions (which are beyond the range of the search window), and more residuals after motion compensation
  - More complex spatial textures and more residuals after intra-frame prediction
  - Larger noises of the original picture

3.3 Madi and Madp Information

NOTE
The **Madi** and **Madp** information is supported only by Hi3519 V101 and Hi3516C V300.

Madi is used to measure the spatial texture complexity of the current frame, that is, the texture variation based on pixel values within a block. The block size can be 16x16, 32x32, or 64x64.

Madi is calculated as follows:

$$f = \frac{1}{N}\sum_{n=0}^{N-1} |Pix_n - \overline{Pix}|$$

$$\overline{Pix} = \frac{1}{N}\sum_{n=0}^{N-1} Pix_n$$

- *N* indicates the number of pixels in a 16x16, 32x32, or 64x64 block. $\overline{Pix}$ is the average luminance value within a block, and **f** is the luminance complexity within a block.
- The encoder supports the reporting of frame-level Madi statistic information (average Madi values of all LCU units within a frame). The reporting of this information is supported only by the H.265.

Madp is used to measure the temporal motion complexity of the current frame. It is the average sum of absolute difference (SAD) after motion search and compensation on all blocks (unit: a 16x16 block) in a frame picture (which is a 1/4 down-sampling picture of the original picture). It measures the average encoding temporal complexity of all blocks. Its value reflects the encoding pressure of a frame picture. The reporting of this information is supported only by the H.265.

Madp is calculated as follows:

$$SAD = (\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} |\,\mathrm{Pix\_ori(n,m)} - \mathrm{Pix\_pre(n,m)}\,|)$$

$$f = (\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} |\,\mathrm{Pix\_ori(n,m)} - \mathrm{Pix\_pre(n,m)}\,|) / M * N$$

SADs are classified into block-level SADs and frame-level SADs. For the block-level SAD (16x16 block), *M* and *N* are 16; for the frame-level SAD, *M* is the picture height, and *N* is the picture width. **Pix_ori(n,m)** indicates the original pixel, **Pix_pre(n,m)** indicates the predicted pixel based on the ME search, and **f** indicates the Madp.

3.4 QP Histogram

NOTE
The QP histogram is supported only by Hi3519 V101 and Hi3516C V300.

The QP histogram is a histogram that displays the QP values used by all blocks (unit: 4x4 block). The QP values are actual QP values used by the luminance component (Y) during the encoding process in skip mode or non-skip mode. The actual QP values are inferred from the frame-level, line-level, and CU-level bit rate control. The histogram has 52 levels (supported by both H.264 and H.265).

3.5 Other Reporting Information

NOTE

Unless otherwise specified, the reporting information described in Table 3-1 is supported by the product versions to which this document applies.

Table 3-1 Other reporting information

Name		Description
MeanQp		Average QP value for encoding the entire frame picture. It reflects the encoding pressure during the encoding process. **Note that this information is supported by Hi3519 V100, Hi3519 V101, and Hi3516C V300.**
StartQp		Starting QP of the current frame
H.264	Inter16x16MbNum	Number of macroblocks to be encoded in inter16x16 prediction encoding mode in the current frame
	Inter8x8MbNum	Number of macroblocks to be encoded in inter8x8 prediction encoding mode in the current frame
	Intra16MbNum	Number of macroblocks to be encoded in intra16 prediction encoding mode in the current frame
	Intra8MbNum	Number of macroblocks to be encoded in intra8 prediction encoding mode in the current frame
	Intra4MbNum	Number of macroblocks to be encoded in intra4 prediction encoding mode in the current frame
H.265	Inter64x64CuNum	Number of CUs to be encoded in inter64x64 prediction mode in the current frame
	Inter32x32CuNum	Number of CUs to be encoded in inter32x32 prediction mode in the current frame
	Inter16x16CuNum	Number of CUs to be encoded in inter16x16 prediction mode in the current frame
	Inter8x8CuNum	Number of CUs to be encoded in inter8x8 prediction mode in the current frame
	Intra32x32CuNum	Number of CUs to be encoded in intra32x32 prediction mode in the current frame
	Intra16x16CuNum	Number of CUs to be encoded in intra16x16 prediction mode in the current frame

Name		Description
	Intra8x8CuNum	Number of CUs to be encoded in intra8x8 prediction mode in the current frame
	Intra4x4CuNum	Number of CUs to be encoded in intra4x4 prediction mode in the current frame