



智能编码

使用指导

文档版本 00B04

发布日期 2018-11-13

版权所有 © 深圳市海思半导体有限公司 2017-2018。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：support@hisilicon.com



前言

概述

本文档主要介绍 H264 和 H265 协议智能编码相关内容。智能编码主要包含四部分内容：

- 第一部分是 GOP 结构, 不同的 GOP 结构适合不同的场景, GOP 结构可以动态设置, 对不同的场景选择合适的 GOP 结构能够优化编码性能;
- 第二部分是编码器输入信息, 编码器输入信息接口可以和其他的智能分析模块联动, 对智能分析出的感兴趣区域或重要区域, 使用 QpMap 进行保护, 或结合客户自己的算法做出更好的码率控制;
- 第三部分是编码器输出信息, 客户可以根据编码器输出信息, 给智能分析算法提供更多可参考的输入;
- 第四部分是 CyclicIntraRefresh, 此技术不编码 IDR 帧, 在 P 帧中周期性的编码 I 宏块, 在特殊的应用场景实现码率平滑。



说明

- 未有特殊说明, Hi3559CV100 与 Hi3559AV100 内容一致。
- 未有特殊说明, Hi3516DV300 与 Hi3516CV500 内容一致。
- 未有特殊说明, Hi3516EV300/Hi3518EV300 与 Hi3516EV200 内容一致。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3559A	V100ES
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3516C	V500
Hi3516D	V300
Hi3516E	V200



产品名称	产品版本
Hi3516E	V300
Hi3518E	V300

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

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 00B04 (2018-11-13)

添加 Hi3516EV200/Hi3516EV300/Hi3518EV300 相关内容。

文档版本 00B03 (2018-06-15)

第 3 次临时版本发布。

1.5 和 1.8 小节涉及修改。

文档版本 00B02 (2018-01-15)

第 2 次临时版本发布。

1.7.2、1.8 和 2.2.1 小节涉及修改。

2.1.2 小节，更新图 2-2 和图 2-3。

文档版本 00B01 (2017-04-28)

第 1 次临时版本发布。



目 录

前 言.....	i
1 GOP 结构和适用场景	1
1.1 GOP 模式名词解释	1
1.2 NormalP 模式 GOP 结构说明及使用方法.....	1
1.2.1 结构说明	1
1.2.2 使用方法	2
1.3 DualP 模式 GOP 结构说明及使用方法	2
1.3.1 结构说明	2
1.3.2 使用方法	3
1.4 SmartP 模式 GOP 结构说明及使用方法	4
1.4.1 结构说明	4
1.4.2 使用方法	4
1.5 AdvSmartP 模式 GOP 结构说明及使用方法.....	5
1.5.1 结构说明	5
1.5.2 使用方法	6
1.6 BiPredB 模式 GOP 结构说明及使用方法	7
1.6.1 结构说明	7
1.6.2 使用方法	7
1.7 Cyclic Intra Refresh 原理和使用方法.....	8
1.7.1 原理	8
1.7.2 使用方法	8
1.8 GOP 结构内存占用、延时、适用场景及兼容性	9
2 编码器输入信息.....	11
2.1 QpMap/SkipMap 接口定义	11
2.1.2 QPMAP 表内存排布方式	12
2.2 编码接口定义	13
2.2.1 码率控制接口.....	13
2.2.2 编码发送图像接口.....	14
2.3 QpMap/SkipMap 实现自适应 ROI	15
2.4 QpMap/SkipMap 实现外部码率控制	16



3 编码器输出信息.....	17
3.1 SSE 和 PSNR 信息.....	17
3.2 HeaderBits 和 ResidualBits 信息.....	17
3.3 Madi 和 Madp 信息.....	17
3.4 QP Histogram.....	18
3.5 其他上报信息.....	18



表格目录

表 1-1 内存占用、延时及适用场景.....	9
表 1-2 Hisilicon 后端产品兼容性.....	10
表 3-1 其他上报信息表	18



1 GOP 结构和适用场景

1.1 GOP 模式名词解释

GOP 模式	P 帧同时可参考的参考帧帧数	备注
NormalP	1	P 帧只参考一个参考帧。
SmartP	2	P 帧参考一个长期参考帧和一个短期参考帧。
AdvSmartP	2	P 帧参考一个长期参考帧和一个短期参考帧。
DualP	2	P 帧参考两个参考帧。
BiPredB	2	<ul style="list-style-type: none">• P 帧参考两个参考帧；• B 帧参考一个前向参考帧和一个后向参考帧。

1.2 NormalP 模式 GOP 结构说明及使用方法



说明

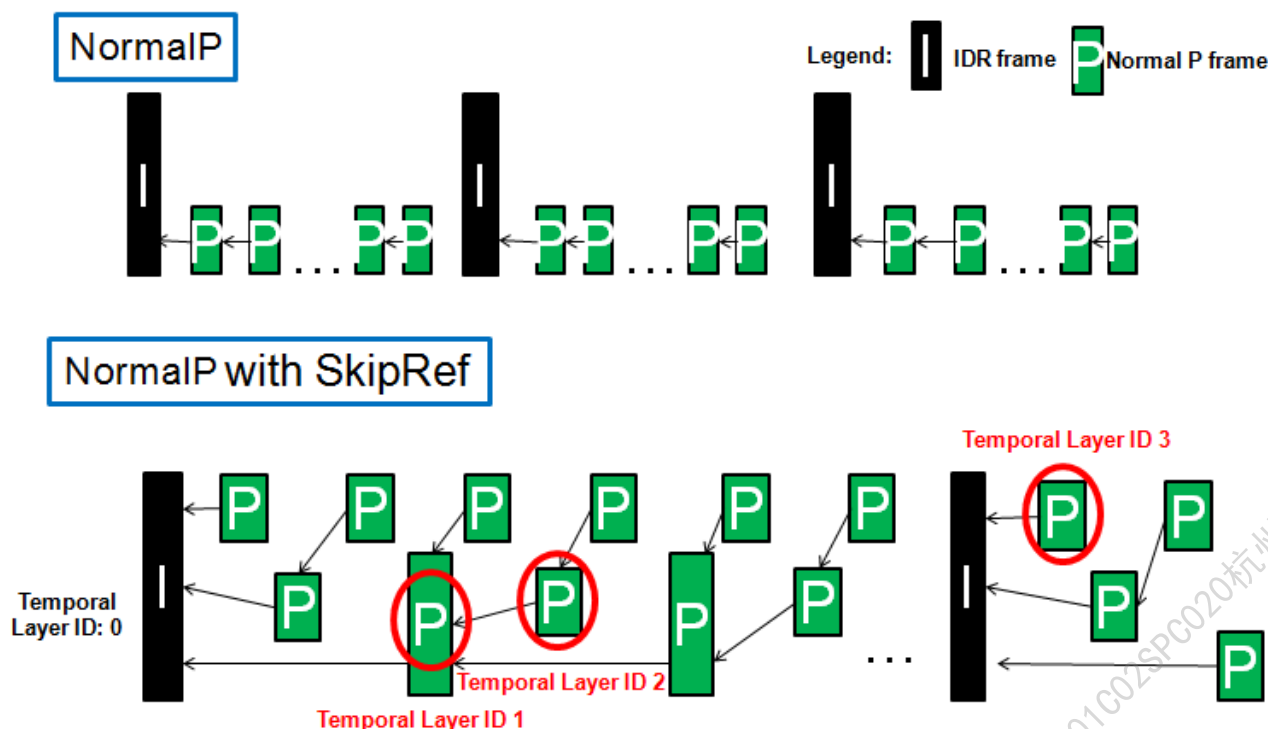
NormalP 是最通常的一种 GOP 结构，如果没有特殊说明，海思所有芯片均支持这一模式。

1.2.1 结构说明

- NormalP 参考关系很简单，每个 P 帧参考一个前向参考帧。
- NormalP 对使用场景没有要求，任何场景都可以使用。

NormalP 模式 GOP 结构，如图 1-1 所示。

图1-1 NormalP 模式 GOP 结构



1.2.2 使用方法

【相关接口】

HI_MPI_VENC_CreateChn

【相关参数】

VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_NORMALP

VENC_CHN_ATTR_S::stGopAttr.stNormalP.s32IPQpDelta 推荐设为 3，值越大 I 帧越大，I 帧质量越好。

1.3 DualP 模式 GOP 结构说明及使用方法



说明

此小节 Hi3559AV100ES 不支持。

1.3.1 结构说明

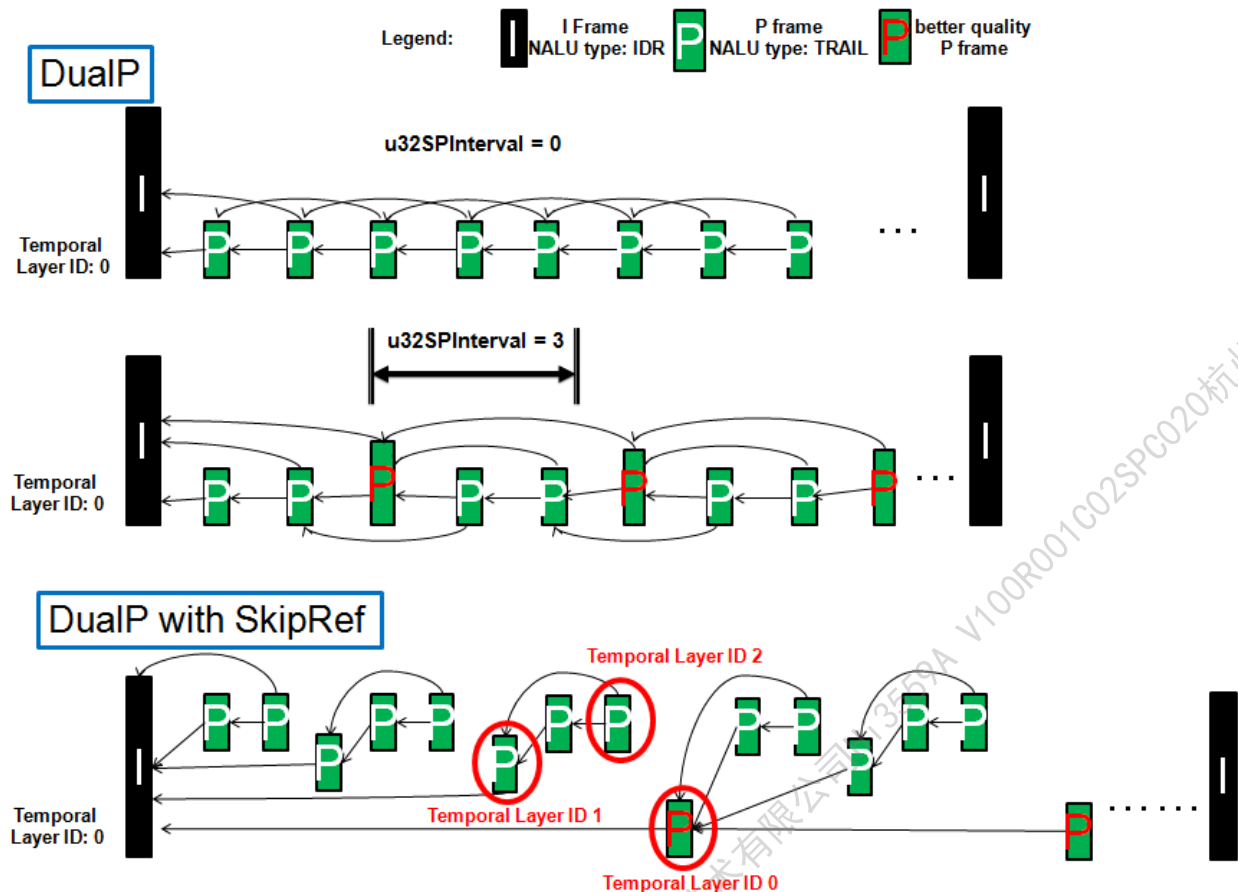
- 其中：SP 指特殊的 P 帧，这里简称 SP 帧，该帧 Qp 值推荐小于其他 P 帧 Qp 值，u32SpInterval=0 指不支持 SP 帧。
- DualP 模式下 P 帧参考就近的两个前向参考帧，能够利用更多参考帧的时域相关性提升编码压缩性能。主要应用在运动且有低延时要求的场景，DualP 压缩性能低于



BipredB 模式但高于 NormalP 模式。DualP 由于两个参考帧均使用前向参考帧，因此不存在编码和解码延时。

DualP 模式 GOP 结构，如图 1-2 所示。

图1-2 DualP 模式 GOP 结构



1.3.2 使用方法

【相关接口】

HI_MPI_VENC_CreateChn

【相关参数】

- VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_DUALP
- 可以通过周期性的编码质量更好的 P 帧（即 SP 帧），优化图像质量。

SP 帧间隔可以通过 VENC_CHN_ATTR_S::stGopAttr.stDualP.u32SPInterval 参数设置。



1.4 SmartP 模式 GOP 结构说明及使用方法



说明

此小节 Hi3559AV100ES 不支持。

1.4.1 结构说明

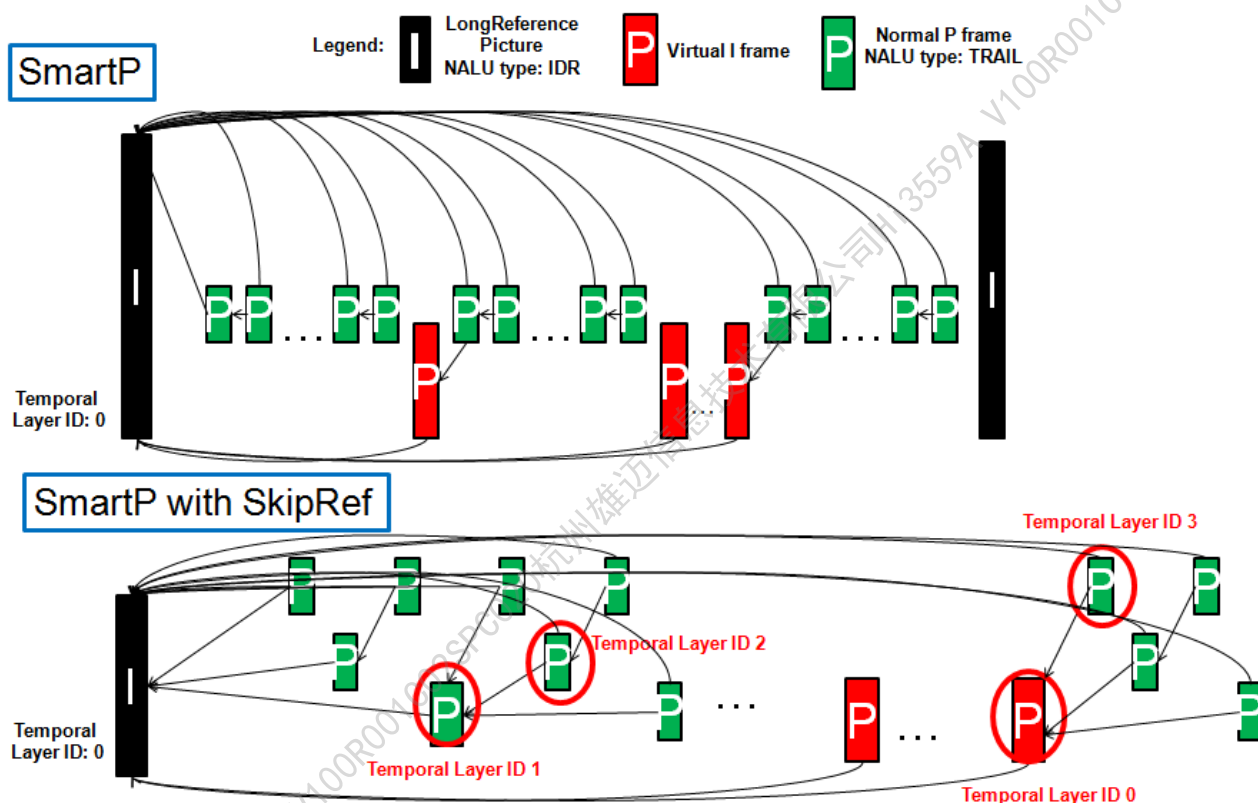
SmartP 模式下 P 帧参考 IDR 帧（长期参考帧）和前向参考帧（短期参考帧），利用两个参考帧的时域相关性提升编码压缩性能。主要应用在监控场景。

此场景的特点是摄像头固定安装，场景中的人和物体有静止有运动。

- 在静止区域，利用长期参考帧和当前帧的时域相关性可以大幅降低码率，并减少呼吸效应和拖尾效应；
- 在运动区域，利用短期参考帧进行运动估计。SmartP 模式拉长 IDR 帧间隔，中间定期插入虚拟 I 帧，能够大幅度降低监控场景的码率并提升图像质量，码率节省可以达到 30%~50%，但对于摄像头运动的场景不适用。

SmartP 模式 GOP 结构，如图 1-3 所示。

图1-3 SmartP 模式 GOP 结构



1.4.2 使用方法

【相关接口】



HI_MPI_VENC_CreateChn

【相关参数】

- 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 second

1.5 AdvSmartP 模式 GOP 结构说明及使用方法



说明

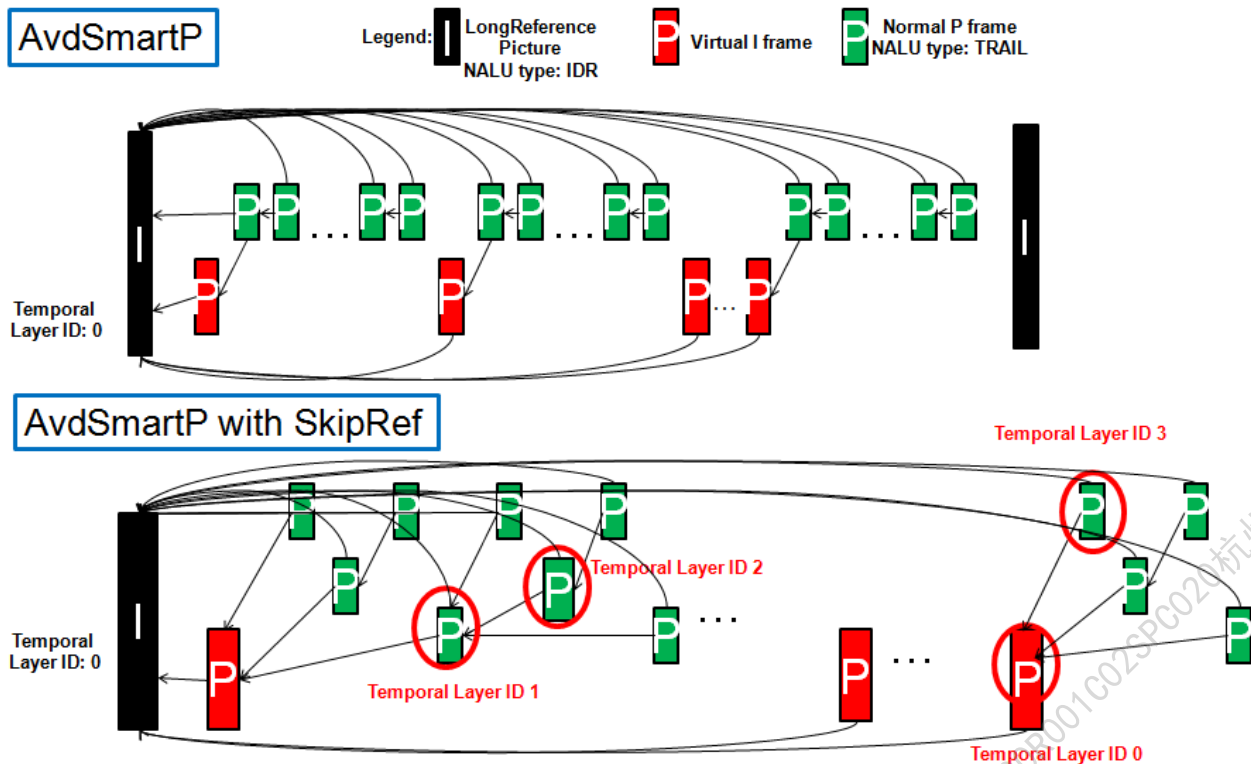
此小节 Hi3559AV100ES/Hi3519AV100/Hi3516CV500/Hi3516DV300/Hi3516EV200 不支持。

1.5.1 结构说明

- 其中：紫色帧编码为 IDR 帧，且用作长期参考帧；红色帧编码为 VI 帧（虚拟 I 帧，本质上是一个普通 P 帧，该帧只参考 IDR 帧，且 Qp 值推荐小于其他 P 帧 Qp 值。
- AdvSmartP 模式与 SmartP 模式 GOP 结构基本一样，差别在于长期参考帧 IDR 帧是编码器内部生成的帧，播放器不显示。

AdvSmartP 模式 GOP 结构，如图 1-4 所示。

图1-4 AvdSmartP 模式 GOP 结构



1.5.2 使用方法

【相关接口】

- HI_MPI_VENC_CreateChn
- HI_MPI_VENC_EnableAdvSmartP

【相关参数】

- VENC_CHN_ATTR_S::stGopAttr.enGopMode = VENC_GOPMODE_ADVSMARTP;
- 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 second



1.6 BiPredB 模式 GOP 结构说明及使用方法

1.6.1 结构说明



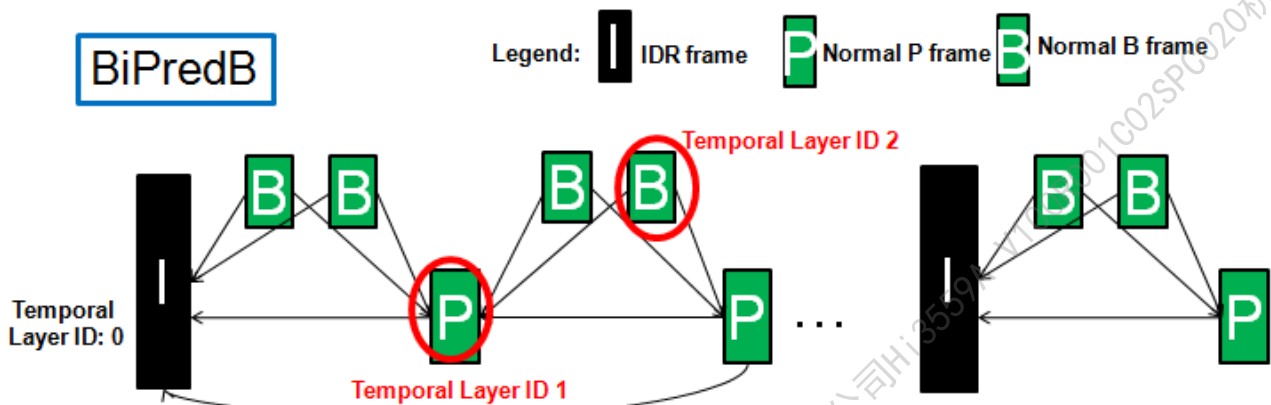
说明

此小节 Hi3559AV100ES/Hi3516CV500/Hi3516DV300/Hi3516EV200 不支持。

- 其中：u32BFrmNum 指 IDR 帧和 P 帧或 P 帧和 P 帧之间 B 帧的个数，如图 1-5 中 u32BFrmNum = 2，每个 Gop 的最后一帧一定是 P 帧，之前的 B 帧个数可能不满足 u32BFrmNum 要求。
- BiPredB 模式相邻两个 P 帧中支持插入 1~3 个 B 帧，B 帧不做参考，使用一个前向参考帧和一个后向参考帧。对于运动场景，双向预测能够更好的做运动估计并且可以支持加权预测，可以提升编码器的压缩性能。

BiPredB 模式 GOP 结构，如图 1-5 所示。

图1-5 BiPredB 模式 GOP 结构



1.6.2 使用方法

【相关接口】

HI_MPI_VENC_CreateChn

【相关参数】

- 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

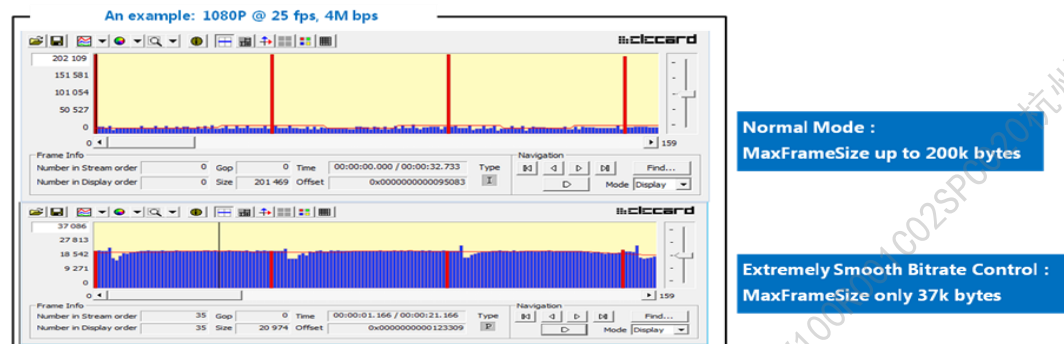


1.7 Cyclic Intra Refresh 原理和使用方法

1.7.1 原理

一般场景 IDR 帧大小会超出 P 帧很多倍，特别是运动比较小的场景，IDR 帧可能达到 P 帧的几十至上百倍。在网络传输特别是无线传输的时候，IDR 帧会引起瞬时网络冲击造成丢帧及增大延时。一般编码技术可以通过增大 IDR 帧的 QP 来降低 IDR 帧大小，但同时会降低 IDR 帧质量引起呼吸效应。CyclicIntraRefresh 不改变 IDR 帧质量，把本来一个 IDR 帧编码的 Intra LCU/宏块分散在若干个 P 帧中，使每一个帧的大小相对平均。如图 1-6 所示，使用 CyclicIntraRefresh 技术，在监控场景下 1080P@25fps 在 4Mbps 码率的最大帧大小可以从 200k 字节降低到 37k 字节。使码率非常平稳。

图1-6 Cyclic Intra Refresh 效果



1.7.2 使用方法

【相关接口】

HI_MPI_VENC_SetIntraRefresh

【相关参数】

- VENC_PARAM_INTRA_REFRESH_S::bRefreshEnable = HI_TRUE
- VENC_PARAM_INTRA_REFRESH_S::enIntraRefreshMode= INTRA_REFRESH_ROW; //宏块/LCU 按行刷新
- VENC_PARAM_INTRA_REFRESH_S::u32RefreshNum; // 每帧刷新宏块/LCU 行数
- VENC_PARAM_INTRA_REFRESH_S::u32ReqIQp; // 请求 I 帧时使用的 QP

优势

- 码率非常平稳，对网络的冲击小，适合用于无线网络传输环境；
- 编码、解码及网络延迟非常小；
- 不会降低 I 帧质量，不会引起严重的呼吸效应。

使用限制

- 由于 Intra LCU/宏块分布在几个 P 帧中，所以播放视频最开始的几帧是不完整的；



- 对解码器有兼容性要求，必须支持没有 IDR 帧的码流的解码；
- 只支持 NormalP 的 GOP 结构，其他 GOP 结构不支持。
- 该技术主要针对码率平稳要求高的场景，并不会降低码率，对低码率场景不适用。
- Hi3559AV100ES/Hi3559AV100 仅支持宏块/LCU 按行刷新。

1.8 GOP 结构内存占用、延时、适用场景及兼容性



说明

- PicSize 为参考帧帧存，AdvInfoSize 为 AdvSmartp 的信息帧存（图像格式为 YUV420 格式）。
- PicSize 和 AdvInfoSize 的计算方法下面将详细描述。
- SrcPicSize 为编码之前的源图像大小，主要是 B 帧有延时需要缓存一部分源图像。
- BiPredB 模式编码端和解码端都有延时，其他模式编码端和解码端都不存在延时

表1-1 内存占用、延时及适用场景

GOP 模式	DDR 占用		Delay		适用场景
	H.264&H.265 Enc	H.264&H.265 Dec	H.264&H.265 Enc	H.264&H.265 Dec	
NormalP	2 * PicSize	2 * PicSize	NA	NA	一般场景
SmartP	3 * PicSize	3 * PicSize	NA	NA	监控场景、摄像头固定不动
AdvSmartP	3 * PicSize +AdvInfoSize	3 * PicSize	NA	NA	监控场景、摄像头固定不动
DualP	3 * PicSize	3 * PicSize	NA	NA	运动场景，摄像头有相对运动。如行车记录仪，手持 DV 等。
BiPredB (N 为 B 帧个数)	3*PicSize+N* SrcPicSize	3 * PicSize	N frames	N frame	运动场景，摄像头有相对运动。如行车记录仪，手持 DV 等。

PicSize 的计算方式

- 编码帧存（参考帧和重构帧）每块 VB 大小计算方式如下：

$$\text{PicSize} = \text{YHeaderSize} + \text{CHheaderSize} + \text{YSize} + \text{CSize} + \text{PmeSize} + \text{PmeInfoSize} + \text{TmvSize} + \text{NbiUpSize}$$
- 帧存大小各子项计算方法请参考文档《HiMPP V4.0 媒体处理软件开发参考》的“视频编码”章节。



Advsmartp 背景帧帧存的计算方式

在 advsmartp 模式下，额外需要一帧背景帧存，大小需要如下：

$$\text{BgModelsize} = \text{YSize} + \text{CSize} + \text{ExtYSize} + \text{ExtCSize} + \text{InfoSize}$$

内存子项名称		H.265
YSize		$\text{align}(\text{Width}, 64) * \text{align}(\text{Height}, 16)$
CSize		$\text{YSize} / 2$
ExtYSize	8bit	0
	10bit	$\text{align}(\text{Width}, 64) * \text{align}(\text{Height}, 16) / 4$
ExtCSize	8bit	0
	10bit	$\text{align}(\text{Width}, 64) * \text{align}(\text{Height}, 16) / 8$
InfoSize	8bit	$(\text{align}(\text{Width}, 64) >> 6) * (\text{align}(\text{Height}, 64) >> 6) * 76 * 128$
	10bit	$(\text{align}(\text{Width}, 64) >> 6) * (\text{align}(\text{Height}, 64) >> 6) * 100 * 128$

兼容性

HiSilicon 后端产品兼容性，如表 1-2 所示。

表1-2 HiSilicon 后端产品兼容性

-	Hi_H 264 Deco der	Hi_H 265 Deco der	Hi3536		Hi3531A /Hi3521 A/Hi352 0DV300	Hi35 35	Hi3559AV100 ES/Hi3559A V100		Hi3519AV1 00		Hi3516EV200	
	H.264	H.265	H.2 64	H.2 65	H.264	H.26 4	H.26 4	H.265	H.26 4	H.2 65	H.26 4	H.265
NormalP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SmartP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AdvSma rtP	No	Yes	No	Yes	No	No	Yes	Yes	No	No	No	No
DualP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
BiPredB	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No
CyclicIn traRefres h	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes

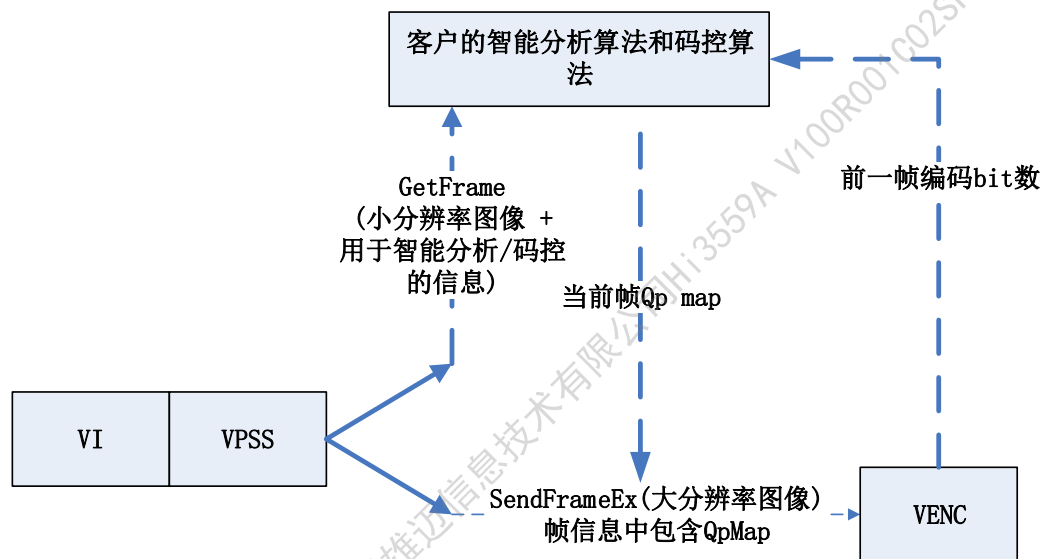


2 编码器输入信息

2.1 QpMap/SkipMap 接口定义

QpMap 编码方案总体设计系统框图，如图 2-1 所示。

图2-1 QpMap 编码方案总体设计



方案实现细节和注意事项：

- 为了保证每帧的 SrcPic 和 QpMap 能够保持同步，方案需要在用户态 App 完成，App 通过用户态从 VI/VPSS 取走图像，经过智能分析和帧级码率控制，计算一帧内每个宏块的 QpMap，再把 SrcPic 和 QpMap 通过 SendFrameEx 接口配置下来；SDK 内部 Bypass 帧率控制和帧级码率控制；智能分析和帧级码率控制算法客户自己开发。
- 海思 SDK 内部不会有图像和码流缓存，但是调度的延迟不可避免；方案的性能和延迟客户自行评估。
- H.264、H.265 都按照 16x16 的块大小配置 QpMap，但 H.265 的编码 CU 可能大于 16x16，因此同一个 CU 可能对应有几个不同 QP 值，海思提供接口，在这种情况下支持从以下 3 种模式中选择一种。



- (1) 取多个 QP 的最大值
- (2) 取多个 QP 的最小值
- (3) 取多个 QP 的平均值
- QpMap 的配置，支持相对 Qp 和绝对 Qp 两种模式。
 - 相对 Qp 指的是基于编码器内部逻辑计算的码控（宏块级）仍然生效，外部通过 QpMap 输入 Qp 大小是一个变化量 Qpdelta，在原始码控（宏块级）基础上，叠加该 Qpdelta。
 - 绝对 Qp 指的是每一个编码块的 Qp 完全是由用户外部输入确定，不依赖于内部宏块级码控算法。
- 编码器内部的重编码和丢帧机制可能无法使用。
- 海思 SDK 可以提供每帧图像的如下信息。
 - ISP 相关信息；
 - 前一帧编码字节数。

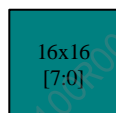
2.1.2 QPMAP 表内存排布方式

QPMAP 表，H.265 和 H.264 编码基本单元不同，所以 QP 排列方式不同。H.264 编码和 H.265 编码 QP 值的组织方式如图 2-2 和图 2-3 所示。

图2-2 H.265 QPMAP LCU 排放位置



16x16块QPMAP值：



每一个16x16块qpmap值由8bits表示，其中：

- 1、[7]: 16x16块skip使能标志，1: skip; 0: 非skip;
- 2、[6]:绝对qp 标志位，1: 绝对qp; 0: 相对qp;
- 2、[5:0]: qp值，绝对qp: [0,51]; 相对qp: [-32,31]

每一个LCU包含16个16*16小块，CU的qp值有以下3种方式：

- 1、取所含小块qp值的平均值。
- 2、取所含小块中最大qp值。
- 3、取所含小块中最小qp值。



图2-3 H.264 QPMAP MB 排放位置

H.264 QPMAP MB(16*16,8bits) 排放位置:
举例: 图像宽: 9*16像素, 图像高: 4*16像素

MB

0	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35

H.264 每一个MB QPMAP值:

16x16
[7:0]

每一个MB为8bits, 其中:

- 1、[7]: 16x16块skip使能标志, 1: skip; 0: 非skip;
- 2、[6]:绝对qp 标志位, 1: 绝对qp; 0: 相对qp;
- 2、[5:0]: qp值, 绝对qp: [0,51]; 相对qp: [-32,31];

2.2 编码接口定义

2.2.1 码率控制接口

```
typedef enum hiVENC_RC_MODE_E
{
    VENC_RC_MODE_H264CBR = 1,
    VENC_RC_MODE_H264VBR,
    VENC_RC_MODE_H264AVBR,
    VENC_RC_MODE_H264FIXQP,
    VENC_RC_MODE_H264QPMP,
    VENC_RC_MODE_MJPEGCBR,
    VENC_RC_MODE_MJPEGVBR,
    VENC_RC_MODE_MJPEGFIXQP,
    VENC_RC_MODE_H265CBR,
    VENC_RC_MODE_H265VBR,
    VENC_RC_MODE_H265AVBR,
    VENC_RC_MODE_H265FIXQP,
    VENC_RC_MODE_H265QPMP,

```



```

VENC_RC_MODE_BUTT,

}VENC_RC_MODE_E;
typedef struct hiVENC_RC_ATTR_S
{
    VENC_RC_MODE_E enRcMode;
    union
    {
        VENC_H264_CBR_S    stH264Cbr;
        VENC_H264_VBR_S    stH264Vbr;

        VENC_H264_FIXQP_S  stH264FixQp;
        VENC_H264_QPMAP_S  stH264QpMap;
        VENC_H264_AVBR_S    stH264AVbr;
        VENC_MJPEG_CBR_S    stMjpegCbr;
        VENC_MJPEG_VBR_S    stMjpegVbr;
        VENC_MJPEG_FIXQP_S  stMjpegFixQp;

        VENC_H265_CBR_S    stH265Cbr;
        VENC_H265_VBR_S    stH265Vbr;
        VENC_H265_AVBR_S    stH265AVbr;
        VENC_H265_FIXQP_S  stH265FixQp;
        VENC_H265_QPMAP_S  stH265QpMap;
    };
}VENC_RC_ATTR_S;

```

- QpMap 使用 Adaptive ROI 方式，RC 模式使用 CBR 或 VBR，同时 QpMap 使用相对 QP 模式；
- QpMap 使用外部码率控制，RC 模式使用 QPMAP，bypass 内部 RC 算法，此时帧级码率控制和宏块级码率控制都由外部实现。

2.2.2 编码发送图像接口

```

typedef struct hiUSER_RC_INFO_S
{
    HI_BOOL bQpMapValid;
    HI_BOOL bSkipWeightValid;
    HI_U32 u32BlkStartQp;
    HI_U64 u64QpMapPhyAddr;
    HI_U64 u64SkipWeightPhyAddr;
    VENC_FRAME_TYPE_E enFrameType;
} USER_RC_INFO_S;
typedef struct hiUSER_FRAME_INFO_S
{
    VIDEO_FRAME_INFO_S stUserFrame;
}

```



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

2.3 QpMap/SkipMap 实现自适应 ROI

感兴趣区域，如图 2-4 所示。

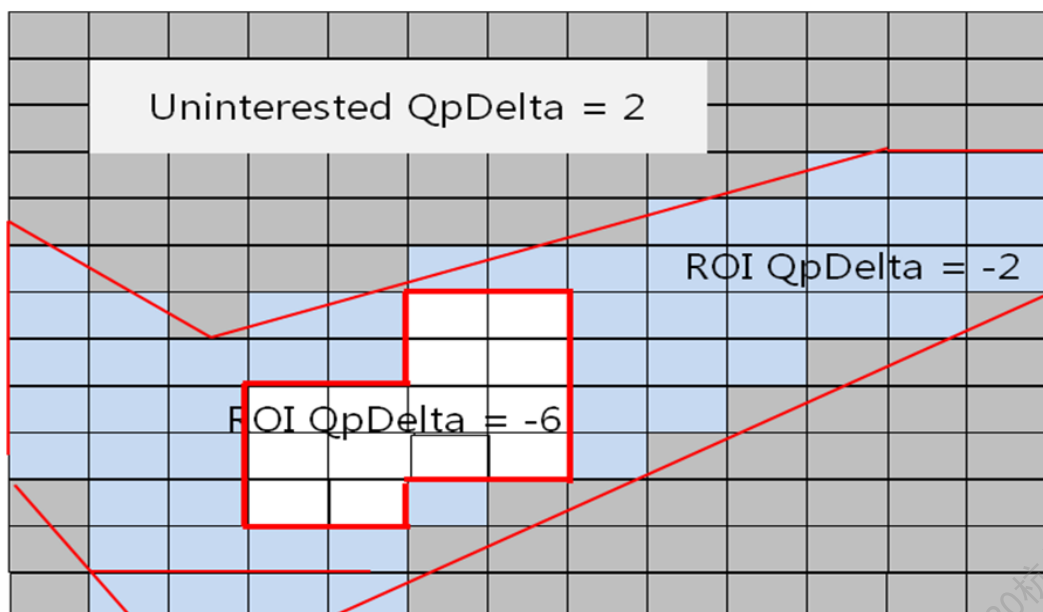
图2-4 感兴趣区域示意图



使用 QpMap 调整感兴趣区域和非感兴趣区域 QP，如图 2-5 所示。



图2-5 使用 QpMap 调整感兴趣区域和非感兴趣区域 QP



通过 QPMap 可以画出任意形状及任意个数的感兴趣区域，并通过降低 QP 提升感兴趣区域图像质量，或提升 QP 降低非感兴趣区域质量。此模式下需要使用相对 QP 调节方式。

2.4 QpMap/SkipMap 实现外部码率控制

通过配置 QPMap 实现外部码率控制，支持相对和绝对方式配置 QP。此模式下所有 LCU/MB 的 QP 均由外部指定，帧级码率控制将不起作用。



3 编码器输出信息

3.1 SSE 和 PSNR 信息

- PSNR 代表整帧图像的信噪比数据，数据越大代表图像质量越好；
- 除了衡量整帧图像的 PSNR 外，编码器还支持统计局部质量的 SSE 信息输出。编码器支持输出最多 8 个区域的 SSE。用户可以根据这 8 个区域的大小及 SSE 算出区域信噪比，Hi3559AV100ES 暂不支持。

3.2 HeaderBits 和 ResidualBits 信息

HeaderBits 和 ResidualBits 都是描述 CU 级的信息，HeaderBits 包含块类型、帧内帧间预测信息等；ResidualBits 包含变换量化后的剩余残差信息。

- HeaderBits 较大一般代表：
 - (1) 图像纹理比较多，块划分比较细碎；
 - (2) 运动比较复杂，相对运动比较多，mv 信息比较多。
- ResidualBits 较大一般代表：
 - (1) 运动不规则或运动过大超出搜索窗范围，运动补偿后剩余残差较多；
 - (2) 空间纹理比较复杂，帧内预测后剩余残差较多。
 - (3) 原始图像噪声较大。

3.3 Madi 和 Madp 信息

Madi 用于度量当前帧的空域纹理复杂度，一个块内基于像素值的纹理的变化，块的大小可以是 16x16，32x32 或 64x64。

Madi 复杂度计算公式为：

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



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

- 其中 N 代表一个 64x64、32x32 块或 16x16 块的像素点的个数。 $\overline{\text{Pix}}$ 对应一个块内的亮度均值， f 对应一块内亮度复杂度计算公式。
- 编码器支持帧级上报 Madi 统计信息（一帧内所有 LCU 单元 madi 的平均值）。该上报信息仅 H.265 支持。

Madv 用于度量当前帧的时域运动复杂度，以 16x16 块为单位，对一帧图像（该图像是原始图像的一个 1/4 下采样图像）中所有块进行运动搜索和补偿后残差绝对值（SAD, Sum of Absolute Difference）的平均值。其度量了所有块平均编码的时域复杂度，通过其值的大小，可以反馈一帧图像的编码压力，该上报信息仅 H.265 支持。

Madv 复杂度计算公式：

$$\text{SAD} = \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |\text{Pix_ori}(n, m) - \text{Pix_pre}(n, m)| \right)$$

$$f = \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |\text{Pix_ori}(n, m) - \text{Pix_pre}(n, m)| \right) / M * N$$

其中 SAD 分为块级和帧级，若对于块级 SAD（16*16 块）， $M=N=16$ ，对于帧级，则 M =图像高， N =图像宽， $\text{Pix_ori}(n, m)$ 为原始像素， $\text{Pix_pre}(n, m)$ 基于 ME 搜索预测像素， f 为 Madv 。

3.4 QP Histogram

QP 直方图：

以 4x4 图像块为单位，统计所有小块使用的 Qp 的直方图，其 Qp 是编码过程中亮度分量（Y）使用的真实 Qp，无论是 skip 模式，还是非 skip 模式。该真实 Qp 一般是在结合帧级，行级和 Cu 级上面推断出来的。共 52 级（H.265/H.264 都支持）。

3.5 其他上报信息



说明

如无特殊说明，下表中描述的上报信息与本文档相对应的产品版本均支持。

表3-1 其他上报信息表

名称	描述
MeanQp	编码整帧图像使用的 Qp 的平均值。可以反馈编码过程中，当前编码压力的大小。
StartQp	当前帧的起始 QP。



H.264	Inter16x16MbNum	编码当前帧中采用 Inter16x16 预测模式的宏块数。
	Inter8x8MbNum	编码当前帧中采用 Inter8x8 预测模式的宏块数。
	Intra16MbNum	编码当前帧中采用 Intra16 预测模式的宏块数。
	Intra8MbNum	编码当前帧中采用 Intra8 预测模式的宏块数。
	Intra4MbNum	编码当前帧中采用 Intra4 预测模式的宏块数。
H.265	Inter64x64CuNum	编码当前帧中采用 Inter64x64 预测模式的 CU 块数。
	Inter32x32CuNum	编码当前帧中采用 Inter32x32 预测模式的 CU 块数。
	Inter16x16CuNum	编码当前帧中采用 Inter16x16 预测模式的 CU 块数。
	Inter8x8CuNum	编码当前帧中采用 Inter8x8 预测模式的 CU 块数。
	Intra32x32CuNum	编码当前帧中采用 Intra32x32 预测模式的 CU 块数。
	Intra16x16CuNum	编码当前帧中采用 Intra16x16 预测模式的 CU 块数。
	Intra8x8CuNum	编码当前帧中采用 Intra8x8 预测模式的 CU 块数。
	Intra4x4CuNum	编码当前帧中采用 Intra4x4 预测模式的 CU 块数。