**AVS2视频解码**

**——数据提取说明**

**(Version 1.0)**

国家专用集成电路设计工程技术研究中心

设计二部多媒体组

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改说明 | 修改者 | 修改时间 | 备注 |
| 0.1 | 创建 | 陈皓 | 2016-1-22 |  |
| 0.2 | 增加slice数据定义 | 陈皓 | 2016-1-22 |  |
| 0.3 | 增加mv时域参考定义 | 尔荟东 | 2016-2-22 |  |
| 0.4 | 修改PIC头定义，增加两条 | 尔荟东 | 2016-2-22 |  |
| 0.6 | 评审 | 李玲、尔荟东、陈明书、王执虎 | 2016-3-11 |  |
| 0.7 | 增加1.2定义 | 李玲 | 2016-3-30 |  |
| 0.7 | 增加1.3定义 | 李玲 | 2016-4-1 |  |
| 0.8 | 增加1.3、1.4、1.10、1.11定义 | 李玲 | 2016-4-5 |  |
| 0.9 | 增加SAO说明定义 | 李玲、陈明书 | 2016-5-20 |  |

# 熵解码数据提取格式需求：

从RD14.0中提取熵解码的输入输出数据。

## 1.1 比特流数据

去掉码流中的头信息，从slice起始码开始（码流起始处应为00 00 01），将每个slice的比特流数据写一个文件，文件名为xx/bitstream/slice\_k\_m.txt。其中xx为序列名，k为picture号（k为4位十进制数），m为当前pic内的slice号（m为4位十进制数）。

## 1.2 sequence头信息数据

Sequence头信息数据写一个文件，文件名为xx/sps/sps\_i.txt。其中xx为序列名，i为Picture号（i为4位十进制数）。每个信息占32位，在文件中占一行。

|  |  |  |  |
| --- | --- | --- | --- |
| 行号 | 信号 | **位宽** | **描述** |
| 0 | lcu\_size | 32 | 最大编码单元的大小 |
| 1 | horizontal\_size | 32 | 图像宽度，即水平方向的像素点数 |
| 2 | vertical\_size | 32 | 图像高度，即垂直方向的像素点数 |
| 3 | chroma\_format | 32 | 色度格式，0-4:0:0, 1-4:2:0, 2-4:2:2 |
| 4 | sample\_adaptive\_offset\_enable | 32 | 样值偏移补偿允许标志 |
| 5 | multi\_hypothesis\_skip\_enable | 32 | 多参考跳过模式允许标志 |
| 6 | dual\_hypothesis\_prediction\_enable | 32 | 多参考图像双前向预测模式允许标志 |
| 7 | weighted\_skip\_enable | 32 | 加权跳过模式允许标志 |
| 8 | asymmetric\_motion\_partitions\_enable | 32 | 非对称运动划分允许标志 |
| 9 | nonsquare\_intra\_prediction\_enable | 32 | 非正方形帧内预测允许标志 |
| 10 | nonsquare\_quadtree\_transform\_enable | 32 | 非正方形变换标志 |
| 11 | pmvr\_enable | 32 | 是否使用PMVR技术 |
| 12 | profile\_id | 32 |  |
| 13 | level\_id | 32 |  |
| 14 | progressive\_sequence | 32 | 1-逐行扫描的帧图像；  0-逐行扫描图像或隔行扫描图像 |
| 15 | field\_coded\_sequence | 32 | 1-场图像；0-帧图像 |
| 16 | sample\_precision | 32 | 1-8bits, 2-10bits |
| 17 | encoding\_precision | 32 | 1-8bits, 2-10bits |
| 18 | aspect\_ratio | 32 | 1-SAR 1.0;  2-DAR 4:3;  3-DAR 16:9;  4-DAR2.21:1 |
| 19 | frame\_rate\_code | 32 | 帧率 |
| 20 | bit\_rate | 32 | 比特率；bit\_rate\_upper<<18+ bit\_rate\_lower,单位是400bps |
| 21 | low\_delay | 32 | 低延迟。  1-不包含B图像，不存在图像重排序延时；  0-包含B图像，存在图像重排序延时。 |
| 22 | temporal\_id\_enable | 32 |  |
| 23 | bbv\_buffer\_size | 32 |  |
| 24 | weight\_quant\_enable | 32 | 1-允许使用加权量化；  0-不使用加权量化。 |
| 25 | load\_seq\_weight\_quant\_data\_flag | 32 |  |
| 26 | scene\_picture\_disable | 32 |  |
| 27 | secondary\_transform\_enable\_flag | 32 |  |
| 28 | adaptive\_loop\_filter\_enable | 32 |  |
| 29 | num\_of\_rcs | 32 | 参考图像配置集数量，<=32 |
| 30 | output\_reorder\_delay | 32 |  |
| 31 | cross\_slice\_loopfilter\_enable | 32 | 跨条带环路滤波。  1-可跨条带边界进行DF、SAO、ALF  0-不可跨条带边界进行DF、SAO、ALF |

## 1.3 picture 头信息数据

每个picture头信息数据写一个文件，文件名为xx/PicInfo/picture\_i.txt。其中xx为序列名，i为图像号（i为4位十进制数）。每个信息占32位，在文件中占一行。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 行号 | | 信号 | **位宽** | **描述** |
| 0 | | first\_pic\_sequence | 32 | 当前图形是否是SPS后的第一个图像 |
| 1 | | picture\_start\_code\_flag | 32 | 预测图像起始码为intra时为0，inter时为1 |
| 2 | | picture\_alf\_enable[0] | 32 | 亮度分量ALF允许标志 |
| 3 | | picture\_alf\_enable[1] | 32 | Cb分量ALF允许标志 |
| 4 | | picture\_alf\_enable[2] | 32 | Cr分量ALF允许标志 |
| 5 | | picture\_coding\_type | 32 | 图像编码方式，0-I，1-P，2-B，3-F |
| 6 | | scene\_pred\_flag | 32 | 场景预测标志。不存在时，值为0。  Intra\_picture时：1-G/GB；0-I  Inter\_picture时：1-S；0-P |
| 7 | | fixed\_picture\_qp | 32 | 固定图像量化因子 |
| 8 | | picture\_qp | 32 |  |
| 9 | | RefPicNum | 32 | 当前帧参考图像数量 |
| 10 | | scene\_reference\_enable/ scene\_picture\_output\_flag | 32 | Inter\_picture时：比特流中不存在时，值为0；  Intra\_picture时为场景图像输出标志：1-G；0-GB |
| 11 | | Cur\_PicOrderCntVal | 32 | 当前帧POC |
| 12 | | RefList0[0] | 32 | 参考帧列表中的参考帧POC |
| 13… | |  |  |  |
| 27 | | RefList0[15] | 32 | 参考帧列表中的参考帧POC |
| 28 | | loop\_filter\_disable | 32 | 去块滤波禁用标志 |
| 29 | | AlphaCOffset | 32 | 码流中不存在时，为0 |
| 30 | | BetaOffset | 32 | 码流中不存在时，为0 |
| 31 | | chroma\_quant\_param\_delta\_cb | 32 | 色度量化参数增量Cb，不存在时为0 |
| 32 | | chroma\_quant\_param\_delta\_cr | 32 | 色度量化参数增量Cr，不存在时为0 |
| 33 | | WeightQuantMatrix4x4 | 16x32 | 4x4加权量化矩阵系数 |
| 49 | | WeightQuantMatrix8x8 | 64x32 | 8x8加权量化矩阵系数 |
| 113 | | alf\_filter\_num\_minus1 | 32 | 图像亮度分量样本滤波补偿滤波器个数，取值范围0~15 |
| 114 | | alf\_region\_distance[0] | 32 | 图像亮度分量相邻样本滤波补偿滤波区域个数 |
| 115 | | AlfCoeffLuma[0][j] | 9x32 | 码流中读取的亮度补偿系数 |
| 124 | | alf\_region\_distance[1] | 32 | 图像亮度分量相邻样本滤波补偿滤波区域个数 |
| 125 | | AlfCoeffLuma[1][j] | 9x32 | 码流中读取的亮度补偿系数 |
|  | |  |  |  |
| 264 | | alf\_region\_distance[15] | 32 | 图像亮度分量相邻样本滤波补偿滤波区域个数 |
| 265 | | AlfCoeffLuma[15][j] | 9x32 | 码流中读取的亮度补偿系数 |
| 274 | | AlfCoeffChroma[0][j] | 9x32 | 码流中读取的色度补偿系数 |
| 283 | | AlfCoeffChroma[1][j] | 9x32 | 码流中读取的色度补偿系数 |
| 292 | | bbv\_delay | 32 | BBV演示 |
| 293 | | time\_code\_flag | 32 | 时间编码标志  1-位流中包含time\_code；0-不包含 |
| 294 | | time\_code | 32 |  |
| 295 | | decode\_order\_index | 32 | 解码顺序索引 |
| 296 | | temporal\_id | 32 | 时间层标识 |
| 297 | | picture\_output\_delay | 32 | 图像输出延迟 |
| 298 | | random\_access\_decodable\_flag | 32 | 随机访问正确解码标志。不存在时，值为1 |
| 299 | | bbv\_check\_times | 32 |  |
| 300 | | progressive\_frame | 32 | 逐行帧标志。  progressive\_sequence的值为1时，progressive\_frame为1 |
| 301 | | top\_field\_first | 32 |  |
| 302 | | repeat\_first\_field | 32 |  |
| 303 | | use\_rcs\_flag | 32 | 参考图像配置集标志 |
| 304 | | rcs\_index | 32 | 参考图像配置集索引 |
| 305 | refered\_by\_others\_flag[0] | | 32 | 第0组：  被参考标志 |
| 306 | num\_of\_reference\_picture[0] | | 32 | 参考图像数量 |
| 307 | delta\_doi\_of\_reference\_picture[0][j] | | 7x32 | 参考图像解码顺序偏移量 |
| 314 | num\_of\_removed\_picture[0] | | 32 | 待移出图像数量 |
| 315 | delta\_doi\_of\_removed\_picture[1][j] | | 7x32 | 待移出图像解码顺序偏移量 |
| 322 | refered\_by\_others\_flag[1] | | 32 | 第1组：  被参考标志 |
| 323 | num\_of\_reference\_picture[1] | | 32 | 参考图像数量 |
| 324 | delta\_doi\_of\_reference\_picture[1][j] | | 7x32 | 参考图像解码顺序偏移量 |
| 331 | num\_of\_removed\_picture[1] | | 32 | 待移出图像数量 |
| 332 | delta\_doi\_of\_removed\_picture[1][j] | | 7x32 | 待移出图像解码顺序偏移量 |
|  | …… | |  |  |
| 832 | refered\_by\_others\_flag[31] | | 32 | 第31组：  被参考标志 |
| 833 | num\_of\_reference\_picture[31] | | 32 | 参考图像数量 |
| 834 | delta\_doi\_of\_reference\_picture[31][j] | | 7x32 | 参考图像解码顺序偏移量 |
| 841 | num\_of\_removed\_picture[31] | | 32 | 待移出图像数量 |
| 842-848 | delta\_doi\_of\_removed\_picture[31][j] | | 7x32 | 待移出图像解码顺序偏移量 |
|  | |  |  |  |

## 1.4 slice解码数据

每个slice的数据写一个文件，文件名为xx/slice\_info/slice\_k\_m.txt。其中xx为序列名，k为picture号（k为4位十进制数），m为当前pic内的slice号（m为4位十进制数）。

|  |  |  |  |
| --- | --- | --- | --- |
| 行号 | 信号 | **位宽** | **描述** |
| 0 | vertical\_position | 32 | slice垂直位置，以LCU为单位 |
| 1 | vertical\_position\_ext | 32 | slice垂直位置扩展，以LCU为单位 |
| 2 | horizontal\_position | 32 | slice水平位置，以LCU为单位 |
| 3 | horizontal\_position\_ext | 32 | slice水平位置扩展，以LCU为单位 |
| 4 | fixed\_slice\_qp | 32 | slice\_qp是否为定值 |
| 5 | slice\_qp | 32 | 条带量化参数 |
| 6 | slice\_sao\_enable[0] | 32 | Y分量。位流中不存在时，值为0 |
| 7 | slice\_sao\_enable[1] | 32 | Cb分量。位流中不存在时，值为0 |
| 8 | slice\_sao\_enable[2] | 32 | Cr分量。位流中不存在时，值为0 |

## 1.6 mv时域参考信息

每个picture的信息数据写一个文件，文件名为xx/mv\_refinfo/picture\_k.txt，其中xx为序列名，k为picture号（k为4位十进制数）。每个信息占32bits，在文件中占一行，用16进制形式表示。每个picture按16x16块的行扫描顺序写出。

每个16x16CU块的信息为：

|  |  |  |  |
| --- | --- | --- | --- |
| 行号 | 信号 | 位宽 | 描述 |
| 0 | RefIdxL0 | 32 |  |
| 1 | MvL0\_x | 32 |  |
| 2 | MvL0\_y | 32 |  |

## 1.7 LCU解码数据

每个LCU信息数据写一个文件，文件名为xx/lcu\_info/pic\_k/lcu\_m\_n.txt，其中xx为序列名，k为pic号（k为4位十进制数），m为当前slice在pic中的序号（m为6位十进制数），n为LCU在slice中的序号（n为6位十进制数）。

每个信息占32位，在文件中占一行。

### 1.7.1 文件结构

LCU解码数据按以下格式输出:

Lcu头信息、sao信息和ALFLcuEnabled

第0个CU解码数据

第1个CU解码数据

第2个CU解码数据

……

第n个CU解码数据

### 1.7.2 格式定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 行号 | 信号 | **位宽** | **个数** | **描述** |
| 0 | lcu\_addr | 32 | 1 | 当前Lcu在图像中的位置，{LCUaddr\_y>>4, LCUaddr\_x>>4}，各占16bits。  LCUaddr\_y，LCUaddr\_x表示像素点坐标。 |
| 1 | SaoMode | 32 | 1 | 高16位为sao\_merge\_type\_index  16’d0：SAO\_NON\_MERGE  16’d1：SAO\_MERGE\_LEFT  16’d2：SAO\_MERGE\_UP  低16位为sao\_mode  16’d0：SAO\_OFF  16’d1：SAO\_INTERVAL  16’d2：SAO\_EDGE |
| 2 | SaoOffset | 32 | 4 | 若sao\_mode为SAO\_OFF，或sao\_merge\_type\_index==SAO\_MERGE\_\*时，为0 |
| 6 | SaoIntervalStartPos | 32 | 1 | SaoMode[i]不等于interval时以0填充 |
| 7 | SaoIntervalDeltaPosMinus2 | 32 | 1 | SaoMode[i]不等于interval时以0填充 |
| 8 | SaoEdgeType | 32 | 1 | SaoMode[i]不等于Edge时以0填充 |
| 9~16 | 同1~8 | 32 | 8 | Cb |
| 17~24 | 同1~8 | 32 | 8 | Cr |
| 25~27 | ALFLCUEnabled | 32 | 3 | 依次输出[0]、[1]、[2] |
| 28 | CUSizeinbit | 32 | 1 | 3,4,5,6 -> 8,16,32,64 |
| 29 | CUPosInLCU | 32 | 1 | 当前CU0在LCU中的位置，  {CUaddr\_y>>3, CUaddr\_x>>3}， 各占16bits。CUaddr\_y，CUaddr\_x表示像素点坐标。 |
| 30 | CuType | 32 | 1 | 帧内/帧间(skip,direct,2N,…)，*具体对应关系见下表* |
| 31 | intra\_chroma\_pred\_mode | 32 | 1 |  |
| 32 | intra\_luma\_pred\_mode | 32 | 1 | PU0 |
| 33 | intra\_luma\_pred\_mode | 32 | 1 | PU1 |
| 34 | intra\_luma\_pred\_mode | 32 | 1 | PU2 |
| 35 | intra\_luma\_pred\_mode | 32 | 1 | PU3 |
| 36 | PuPredMode | 32 | 1 | 0,1,2,3,4 ->  Sym,bipred,dual\_fwd,bck,single\_fwd  PU级 |
| 37 | CuSubType | 32 | 1 | B:10种 F:12种 参见表87/表92，0~9(或0~11)分别与”编码单元子类型”按由上至下的顺序一一对应  CU级 (4bits) |
| 38 | BnxNType (BPuTypeIndex2) | 32 | 1 | 见表90，0~4分别对应 “预测单元类型” 的由上至下的五种情况 |
| 39 | DMH\_mode (dir\_multi\_hypothesis\_mode) | 32 | 1 | Dir Multi Hypo  CU级 (4bits) |
| 40 | ref\_idx | 32 | 1 | PU级 (4bits)，  不存在时输出值为0 |
| 41 | WeightedSkipMode | 32 | 1 | CU级 (4bits) |
| 42 | mvd\_l0\_x | 32 | 1 | 不存在时输出值为0 |
| 43 | mvd\_l0\_y | 32 | 1 | 不存在时输出值为0 |
| 44 | mvd\_l1\_x | 32 | 1 | 不存在时输出值为0 |
| 45 | mvd\_l1\_y | 32 | 1 | 不存在时输出值为0 |
| 46~55 | 同36~45 | 32 | 1 | PU1 |
| 56~65 | 同36~45 | 32 | 1 | PU2 |
| 66~75 | 同36~45 | 32 | 1 | PU3 |
| 76 | transform\_split\_flag | 32 | 1 | CU级 |
| 77 | CuCTP | 32 | 1 | CU级 |
| 78 | CurrentQP | 32 | 1 | 当前Lcu的QP |
|  | 同28~78，下一CU数据 |  |  | 按解码顺序，依次输出当前LCU中所有CU |
| …… | …… |  |  |  |
|  |  |  |  |  |

CuType对应关系

0 : Skip

1 : Direct

2 : 2N

3 : 2N\_H

4 : 2N\_V

5 : 2N\_HU

6 : 2N\_HD

7 : 2N\_VL

8 : 2N\_VR

9 : NxN

10 : I\_2N

11 : I\_N

12 : I\_nNxN

13 : I\_NxnN

## 1.8 LCU残差系数信息

每个LCU信息数据写一个文件，文件名为xx/lcu\_coeff/pic\_k/lcu\_m\_n.txt，其中xx为序列名，k为pic号（k为4位十进制数），m为当前slice在pic中的序号（m为6位十进制数），n为LCU在slice中的序号（n为6位十进制数）。【需进一步商讨格式】

每个信息占32位，在文件中占一行。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 行号 | 信号 | **位宽** | **个数** | **描述** |
| 0 | cuIdx | 32 | 1 | 当前CU0在LCU中的位置，  {CUaddr\_y>>3, CUaddr\_x>>3}， 各占16bits。CUaddr\_y，CUaddr\_x表示像素点坐标。 |
| 1 | blk\_idx | 32 | 1 | 0,1,2,3-luma; 4-Cb, 5-Cr |
| 2 | CGPos | 32 | 1 | 当前CG在blk内部的z序 |
| 3~18 | coeff\_level | 32 | 16 | 当前CG的16个变换系数，按z扫描顺序，从15到0依次输出 |
|  | 同0~18 |  |  | 下一个CG数据 |
| …… | 按解码顺序输出当前LCU所有CG |  |  | 不限定行数范围 |

## 1.9 MV解码数据

每个LCU的解码数据写一个文件，文件名为xx/lcu\_mv/pic\_k/lcu\_mv\_m\_n.txt，其中xx为序列名、k为picture号（k为4位十进制数），m为当前slice在当前picture中的slice号（m为6位十进制数），n为LCU号（n为6位十进制数）。每个信息占32bits，在文件中占一行，用16进制形式表示。每个CU有43行信息。每个LCU内按CU的解码顺序，依次存放CU中PU的解码信息。

lcu\_mv\_m\_n.txt文件内容：

CU0信息

CU1信息

……

CUp信息

其中每个CU的信息格式定义如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 位号 | 信号 | 位宽 | 说明 |
| 0 | LCU\_addr | 32 | 当前LCU在图像中的位置，{LCUaddr\_y>>4, LCUaddr\_x>>4}，各占16bits。  LCUaddr\_y，LCUaddr\_x表示像素点坐标。 |
| 1 | last\_cu | 32 | 当前CU是否是CTU的最后一个CU |
| 2 | CU\_index | 32 | 当前CU在LCU中的位置，{CUaddr\_y>>3, CUaddr\_x>>3}，各占16bits。CUaddr\_y，CUaddr\_x表示LCU内的像素点坐标。 |
| 3 | log2CbSize | 32 | 取对数后CU的大小 |
| 4 | pred\_mode\_flag | 32 | 0-inter, 1-intra |
| 5 | part\_mode | 32 | CU划分类型，同1.4中CuType定义 |
| 6 | predFlagL0 | 32 | [predFlagL1,predFlagL0]  01-list0; 10-list1; 11-Bi/F/sym;  依次表示PU0、PU1、PU2、PU3的predFlagLX的值  AVS2中PuPredMode定义如下：  Sym,bipred,dual\_fwd,bck,single\_fwd对应[11, 11, 11, 10, 01] |
| 7 | predFlagL1/predFlagL0\_snd | 32 |
| 8 | predFlagL0 | 32 |
| 9 | predFlagL1/predFlagL0\_snd | 32 |
| 10 | predFlagL0 | 32 |
| 11 | predFlagL1/predFlagL0\_snd | 32 |
| 12 | predFlagL0 | 32 |
| 13 | predFlagL1/predFlagL0\_snd | 32 |
| 14 | RefIdxL0 | 32 | PU0的list0，不存在时默认为-1 |
| 15 | RefIdxL1 | 32 | PU0的list1，不存在时默认为-1 |
| 16 | RefIdxL0 | 32 | PU1的list0，不存在时默认为-1 |
| 17 | RefIdxL1 | 32 | PU1的list1，不存在时默认为-1 |
| 18 | RefIdxL0 | 32 | PU2的list0，不存在时默认为-1 |
| 19 | RefIdxL1 | 32 | PU2的list1，不存在时默认为-1 |
| 20 | RefIdxL0 | 32 | PU3的list0，不存在时默认为-1 |
| 21 | RefIdxL1 | 32 | PU3的list1，不存在时默认为-1 |
| 22 | MvL0\_x | 32 | PU0的MvL0，不存在时默认为0 |
| 23 | MvL0\_y | 32 |  |
| 24 | MvL1\_x | 32 | PU0的MvL1，不存在时默认为0 |
| 25 | MvL1\_y | 32 |  |
| 26 | MvL0\_x | 32 | PU1的MvL0，不存在时默认为0 |
| 27 | MvL0\_y | 32 |  |
| 28 | MvL1\_x | 32 | PU1的MvL1，不存在时默认为0 |
| 29 | MvL1\_y | 32 |  |
| 30 | MvL0\_x | 32 | PU2的MvL0，不存在时默认为0 |
| 31 | MvL0\_y | 32 |  |
| 32 | MvL1\_x | 32 | PU2的MvL1，不存在时默认为0 |
| 33 | MvL1\_y | 32 |  |
| 34 | MvL0\_x | 32 | PU3的MvL0，不存在时默认为0 |
| 35 | MvL0\_y | 32 |  |
| 36 | MvL1\_x | 32 | PU3的MvL1，不存在时默认为0 |
| 37 | MvL1\_y | 32 |  |
| 38 | intra\_luma\_pred\_mode | 32 | PU0的intra luma pred mode |
| 39 | intra\_luma\_pred\_mode | 32 | PU1的intra luma pred mode |
| 40 | intra\_luma\_pred\_mode | 32 | PU2的intra luma pred mode |
| 41 | intra\_luma\_pred\_mode | 32 | PU3的intra luma pred mode |
| 42 | intra\_chroma\_pred\_mode | 32 |  |

## 1.10 LCU级的BS解码信息

每个picture的所有LCU的bs解码数据写一个文件，文件名为xx/bsinfo/pic\_k.txt，其中xx为序列名、k为picture号(k为4位十进制数)。每个信息占32bits，在文件中占一行，用16进制形式表示。

按照YCbCr次序，以帧级来打印，先垂直边界，后水平边界。

打印垂直边界时，从上往下，从左往右；打印水平边界时，从左往右，从上往下；

假设图像宽为W，高为H，LCU为LxL，则亮度垂直边界有edgeVerY=W/8条，每条有pointVerY=((H+L-1)/L)\*L个数值，即不满足LCU整数倍时，向上扩展到整数倍。亮度水平边界有edgeHorY=H/8条，每条有pointHorY =((W+L-1)/L)\*L个数值，即不满足LCU整数倍时，向上扩展到整数倍。

色度垂直边界有edgeVerCb=edgeVerY/2条，每条有pointVerCb=pointVerY/2个数值，即不满足LCU整数倍时，向上扩展到整数倍。色度水平边界有edgeHorCb=edgeHorY/2条，每条有pointHorCb = pointHorY/2个数值，即不满足LCU整数倍时，向上扩展到整数倍。色度会相应的减少一半。

例如：W=832,H=480,L=64时，

edgeVerY=104，pointVerY=512，edgeVerCb=52，pointVerCb=256；

edgeHorY=60，pointHorY=832，edgeHorCb=30，pointHorCb=416；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 行号 | 信号 | 位宽(bits) | 个数 | 描述 |
|  | bs\_vertical | 32 | 1 | 亮度Y垂直方向的BS值，从上到下，从左到右。  pointVerY\*edgeVerY个Bs值 |
|  | bs\_horizontal | 32 | 1 | 亮度Y水平方向的BS值，从左到右，从上到下。  pointHorY \*edgeHorY个Bs值 |
|  | bs\_vertical | 32 | 1 | 色度的垂直方向的BS值，从上到下，从左到右。  pointVerCb\*edgeVerCb个Bs值 |
|  | bs\_horizontal | 32 | 1 | 色度Cb的水平方向的BS值，从左到右，从上到下。  pointHorCb \*edgeHorCb个Bs值 |
|  | bs\_vertical | 32 | 1 | 色度Cr的垂直方向的BS值，从上到下，从左到右。  pointHorCb \*edgeHorCb个Bs值 |
|  | bs\_horizontal | 32 | 1 | 色度的水平方向的BS值，从左到右，从上到下。  pointHorCb \*edgeHorCb个Bs值 |

## 1.11 预测和重建数据

存放数据以64bit为一行，一行中可以有8个或4个像素点，这取决于像素点是8bit还是10bit，当像素点为10bit时，按每个像素点16bit存放。

其中，数据存放采用小端序方式.

例：0x12345678

像素点顺序12345678

对于YUV各图像分量，地址以64bit对齐，即每个分量的起始均为新的一行。

对于行像素点不满64bit整数倍的数据以0填充

### 1.11.1 预测数据

帧内和帧间预测得到的图像数据。每个picture的预测数据写一个文件，文件名为xx/img/pred\_pic\_k.txt，其中xx为序列名、k为picture号(k为4位十进制数)。先存放Y分量，再存放U分量，最后存放V分量。

### 1.11.2 反变换反量化后的数据

反变换、反量化后得到的图像数据。每个picture的预测数据写一个文件，文件名为xx/img/tq\_pic\_k. txt，其中xx为序列名、k为picture号(k为4位十进制数)。先存放Y分量，再存放U分量，最后存放V分量。

### 1.11.3 Deblock filter后的数据

Deblock filter后得到的图像数据。每个picture的去块滤波后的数据写一个文件，文件名为xx/img/df\_pic\_k. txt，其中xx为序列名、k为picture号(k为4位十进制数)。先存放Y分量，再存放U分量，最后存放V分量。

### 1.11.4 SAO后的数据

SAO滤波后得到的图像数据。每个picture的去块滤波后的数据写一个文件，文件名为xx/img/sao\_pic\_k. txt，其中xx为序列名、k为picture号(k为4位十进制数)。先存放Y分量，再存放U分量，最后存放V分量。

### 1.11.4 ALF后的数据/最终的重建数据

ALF滤波后得到的图像数据。每个picture的去块滤波后的数据写一个文件，文件名为xx/img/alf\_pic\_k. txt，其中xx为序列名、k为picture号(k为4位十进制数)。先存放Y分量，再存放U分量，最后存放V分量。

## 1.12 LCU级的SAO参数信息

每个picture的所有LCU的SAO参数数据写一个文件，文件名为xx/saoinfo/pic\_k.txt，其中xx为序列名、k为picture号(k为4位十进制数)。每个信息占32bits，在文件中占一行，用16进制形式表示。每个LCU占20行。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | lcu\_addr | 32 | 1 | 当前Lcu在图像中的位置，{LCUaddr\_y>>4, LCUaddr\_x>>4}，各占16bits。  LCUaddr\_y，LCUaddr\_x表示像素点坐标。 |
| 1 | SaoMode | 32 | 1 | 16’d0：SAO\_OFF  16’d1：SAO\_INTERVAL  16’d2：SAO\_EDGE |
| 2 | SaoOffset | 32 | 4 | 若sao\_mode为SAO\_OFF，或sao\_merge\_type\_index==SAO\_MERGE\_\*时，为0 |
| 6 | SaoIntervalStartPos | 32 | 1 | SaoMode[i]不等于interval时以0填充 |
| 7 | SaoIntervalDeltaPosMinus2 | 32 | 1 | SaoMode[i]不等于interval时以0填充 |
| 8 | SaoEdgeType | 32 | 1 | SaoMode[i]不等于Edge时以0填充 |
| 9~16 | 同1~8 | 32 | 8 | Cb |
| 17~24 | 同1~8 | 32 | 8 | Cr |

# 相关定义

# 3、参考文献

1. AVS2-P2(征求意见稿)-20160107.pdf
2. AVS2熵解码报告-陈明书