



# 中华人民共和国国家标准

GB/T XXXXX.7—YYYY

## 信息技术 先进音视频编码 第7部分：移动视频

Information technology - Advanced coding  
of audio and video - Part 7: Mobile video  
(送审稿)

××××-××-××发布

××××-××-××实施

国家质量监督检验检疫总局 发布

目 次

前 言 ..... III

引 言 ..... IV

1.1 目标 ..... IV

1.2 应用 ..... IV

1.3 档次和级别 ..... IV

1.4 技术概述 ..... IV

1.5 如何阅读本部分 ..... IV

1.6 相关专利情况说明 ..... V

1 范围 ..... 1

2 规范性引用文件 ..... 1

3 术语和定义 ..... 1

4 缩略语 ..... 7

5 约定 ..... 8

5.1 算术运算符 ..... 8

5.2 逻辑运算符 ..... 8

5.3 关系运算符 ..... 8

5.4 位运算符 ..... 9

5.5 赋值 ..... 9

5.6 数学函数 ..... 9

5.7 比特流语法、解析过程和解码过程的描述 ..... 9

6 编码比特流的结构 ..... 12

6.1 比特流结构 ..... 12

6.2 编码视频序列 ..... 12

6.3 图像 ..... 12

6.4 条带 ..... 13

6.5 宏块 ..... 13

6.6 8×8 块 ..... 13

6.7 4×4 块 ..... 13

7 比特流的语法和语义 ..... 14

7.1 语法描述 ..... 14

7.2 语义描述 ..... 19

8 解析过程 ..... 25

8.1 **k**阶指数哥伦布码 ..... 26

8.2 **ue(v)**, **se(v)**和**me(v)** ..... 26

8.3 **ce4(v)** ..... 28

9 解码过程 ..... 29

9.1 图像头解码 ..... 29

9.2 条带解码 ..... 29

9.3 宏块解码 ..... 29

9.4 块解码 ..... 34

9.5 反量化 ..... 37

9.6 反变换 ..... 38

9.7 帧内预测 ..... 39

9.8 帧间预测 ..... 41

9.9 重建 ..... 43

9.10 环路滤波 ..... 43

附录 A（规范性附录） 变长码表 ..... 47

附录 B（规范性附录） 档次和级别 ..... 52

B.1 档次 ..... 52

B.2 级别 ..... 52

附录 C（规范性附录） 辅助增强信息 ..... 57

C.1 SEI净载语法定义 ..... 57

C.2 SEI 净载语义 ..... 60

附录 D（规范性附录） 字节流格式 ..... 63

D.1 字节流的语法和语义 ..... 63

D.2 字节流的解码过程 ..... 63

附录 E（规范性附录） 虚拟参考解码器 ..... 64

E.1 HRD参数语法 ..... 64

E.2 HRD参数语义 ..... 64

E.3 HRD操作 ..... 64

E.4 对符合本部分的比特流的要求 ..... 65

E.5 对符合本部分的解码器的要求 ..... 65

# 前 言

GB/T XXXXX在《信息技术 先进音视频编码》的总标题下，包括以下九个部分：

- 第1部分：系统；
- 第2部分：视频；
- 第3部分：音频；
- 第4部分：一致性测试；
- 第5部分：参考软件；
- 第6部分：数字媒体版权管理；
- 第7部分：移动视频；
- 第8部分：在IP网络上传输AVS；
- 第9部分：AVS文件格式。

本部分为GB/T XXXXX的第7部分。

本部分的附录A～附录E为规范性附录。

本部分由中华人民共和国信息产业部提出。

本部分由中国电子技术标准化研究所（CESI）归口。

本部分由XXXX起草。

本部分主要起草人：XXXX。

# 引 言

## 1.1 目标

GB/T XXXXX的本部分是为了适应数字存储媒体、网络流媒体、多媒体通信等应用中对运动图像压缩技术的需要而制定的。

## 1.2 应用

本部分适用的范围包括但不限于下述领域：

交互存储媒体；

宽带视频业务；

多媒体邮件；

分组网络的多媒体业务（MSPN, Multimedia services on packet networks）；

实时通信业务（视频会议，可视电话等）；

远程视频监控。

## 1.3 档次和级别

本部分能支持多种比特率、分辨率和质量的视频压缩。考虑到应用本部分时的互操作性，定义档次和级别。

档次是本部分规定的语法、语义及算法的子集。

级别是在某一档次下对语法元素和语法元素参数值的限定集合。

## 1.4 技术概述

本部分采用了一系列技术来达到高效率的视频编码，包括帧内预测、帧间预测、变换、量化和熵编码等。帧间预测使用基于块的运动矢量来消除图像间的冗余；帧内预测使用空间预测模式来消除图像内的冗余。再通过对预测残差进行变换和量化消除图像内的视觉冗余。最后，运动矢量、预测模式、量化参数和变换系数等用熵编码进行压缩。

### 1.4.1 预测技术

帧内预测不需要参考其它图像，某些采用帧内预测编码的图像可作为编码码流的随机访问点。

帧间预测需要参考先前已解码的图像。帧间预测中运动矢量的精度能达到1/4像素，运动矢量采用预测编码。

### 1.4.2 图像分块

本部分中视频解码过程的基本处理单元是宏块。一个宏块包括一个16×16的亮度样值块和对应的色度样值块。宏块可进一步划分到最小4×4的样本块来进行预测和变换。

### 1.4.3 变换和量化

变换的单元是4×4的样本块。变换系数进行标量量化。

## 1.5 如何阅读本部分

建议读者从第1章（范围）开始，然后转到第3章（术语和定义）。第6章定义了编码比特流结构。第7章（语法和语义）定义了比特流的语法和语义，第7.1条是语法描述，定义了比特流中语法元素的出现顺序，第7.2条是语义描述，也即语法元素的范围、限制和条件。第8章定义了语法元素的解析过程。最后，第9章（解码过程）定义了语法元素如何映射到解码样值。在阅读本部分的过程中，读者还可阅读第2章（规范性引用文件）、第4章（缩略语）、第5章（约定）及附录。

## 1.6 相关专利情况说明

本部分的发布机构提请注意如下事实，声明符合本部分时，可以通过“AVS专利池”使用部分本部分涉及的相关专利。

本部分的发布机构对于专利的范围、有效性和验证资料不提出任何看法。

专利持有人已向本部分的发布机构保证，他愿意同任何申请人在合理和非歧视的条款和条件下，就使用授权许可证进行谈判。这方面，该专利持有人的声明已在本部分的发布机构备案。

本部分涉及的专利的许可遵循本部分起草组织者数字音视频编解码技术标准工作组制定并经会员签署同意的《知识产权政策》，通过AVS专利池进行许可，具体许可条款由专利池管理委员会决定。标准实施者可通过专利池管理委员会或其授权机构获得专利池中所有专利的许可权。

专利池管理委员会由数字音视频编解码技术标准工作组推动发起。有关事宜可联系数字音视频编解码技术标准工作组秘书处，联系方法如下：

联系人：黄铁军（数字音视频编解码技术标准工作组秘书长）

通讯地址：北京2704信箱31分箱

邮政编码：100080

电子邮件：tjhuang@ict.ac.cn

电 话：+86 10 58858303，+86 10 58858300-303

传 真：+86 10 58858301

网 址：<http://www.avs.org.cn>（可从本网址获得最新联系方法）

# 信息技术 先进音视频编码 第7部分:移动视频

## 1 范围

GB/T XXXXX规定了数字音视频的压缩、解压缩、处理和表示的技术方案，适用于高分辨率和标准分辨率数字电视广播、数字存储媒体、网络流媒体、多媒体通信等应用。

GB/T XXXXX的本部分规定了多种比特率、分辨率和质量的视频压缩方法，适用于交互式存储媒体、多媒体邮件、分组网络的多媒体业务、实时通信业务、远程视频监控等应用中的视频编码。

## 2 规范性引用文件

下列文件中的条款通过GB/T XXXXX的本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T XXXXX.8—YYYY 信息技术 先进音视频编码 第8部分:在IP网络上传输AVS

## 3 术语和定义

下列术语和定义适用于GB/T XXXXX的本部分。

### 3.1

**保留 reserved**

定义了一些特定语法元素值用于将来对本标准的扩展。这些值不应出现在符合本标准的比特流中。

### 3.2

**比特串 bit string**

有限位比特的有序序列，其最左边比特是最高有效位（MSB），最右边比特是最低有效位（LSB）。

### 3.3

**比特流 bitstream**

码流

编码图像所形成的二进制数据流。

### 3.4

**比特流缓冲区 bitstream buffer**

存储比特流的缓冲区。

### 3.5

**比特流顺序 bitstream order**

编码图像在比特流中的排列顺序，与图像解码的顺序相同。

### 3.6

**变长编码 variable length coding**

一个可逆的熵编码过程，它将短的码字分配给出现频率较高的符号，将长的码字分配给出现频率较低的符号。

### 3.7

**变换系数 transform coefficient**

变换域上的一个标量。

### 3.8

**编码表示** encoding presentation

数据编码后的形式。

### 3.9

**编码过程** encoding process

产生符合本标准比特流的过程。

注：本部分不规定该过程。

### 3.10

**编码器** encoder

编码过程的实现。

### 3.11

**编码视频序列** coded video sequence

编码比特流的最高层语法结构，包括一个IDR访问单元和所有后续的非IDR访问单元，直到但不包括下一个IDR访问单元或直到比特流结尾。

### 3.12

**编码图像** coded picture

一帧图像的编码表示。

### 3.13

**标志** flag

一个二值变量。

### 3.14

**补偿** compensation

求由语法元素解码得到的样本残差与其对应的预测值之和。

### 3.15

**残差** residual

样本或数据元素的重建值与其预测值之差。

### 3.16

**参考索引** reference index

解码图像缓冲区中参考图像的编号。

### 3.17

**参考图像** reference picture

参考帧

解码过程中用于后续图像帧间预测的图像。

### 3.18

**层** layer

比特流中的分级结构，高层包含低层。编码层由高到低依次为：序列、图像、条带、宏块和块。

### 3.19

**档次** profile

本标准规定的语法、语义及算法的子集。

### 3.20

**访问单元** access unit

包含一帧编码图像的一系列NAL单元。除编码图像头和条带数据外，一个访问单元还可包含其它数据单元。

### 3.21

**非参考图像** non-reference picture



非参考帧

解码过程中不用于后续图像帧间预测的图像。

### 3.22

分量 component

图像的三个样值矩阵（亮度和两个色度）中的一个矩阵或矩阵中的单个样值。

### 3.23

反变换 inverse transform

将变换系数矩阵转换成空域样值矩阵的过程。

### 3.24

反量化 dequantization

对量化系数缩放后得到变换系数的过程。

### 3.25

块 block

一个M×N的样值矩阵或者变换系数矩阵（M列N行）。

### 3.26

块扫描 block scan

量化系数的特定串行排序方式。

### 3.27

亮度 luma

表示亮度信号的样值矩阵或单个样值。

注：亮度的符号是Y。

### 3.28

量化参数 quantization parameter

在解码过程对量化系数进行反量化的参数。

### 3.29

量化系数 quantization coefficient

反量化前变换系数的值。

### 3.30

光栅扫描 raster scan

将二维矩形光栅映射到一维光栅，一维光栅的入口从二维光栅的第一行开始，然后接着扫描第二行、第三行，依次类推。光栅中的行从左到右扫描。

### 3.31

宏块 macroblock

包括一个16×16的亮度样值块和对应的色度样值块。

### 3.32

宏块地址 macroblock address

从图像左上角的宏块开始，沿光栅扫描的顺序编号，起始号为0。

### 3.33

宏块行 macroblock line

从编码的图像左边界到右边界在相同的垂直位置连续的宏块，宏块行的高度是16个样本。

### 3.34

宏块位置 macroblock position

图像中一个宏块的二维坐标，表示为 $(x, y)$ 。图像左上角的宏块 $(x, y) = (0, 0)$ 。对每个宏块列，从左到右 $x$ 依次加1。对每个宏块行，从上到下 $y$ 依次加1。

## 3.35

后向预测 backward prediction

用显示顺序上将来的参考图像对当前图像进行预测。

## 3.36

划分 partitioning

将一个集合分为子集的过程。集合中的每个元素属于且只属于某一个子集。

## 3.37

级别 level

在某一档次下对语法元素和语法元素参数值的限定集合。

## 3.38

即时解码刷新图像 IDR picture

IDR图像

一种帧内解码图像，该图像后的所有图像（解码顺序）的解码过程都不会用到该IDR图像前的图像（解码顺序）进行帧间预测。一个编码视频序列的第一帧应为IDR图像。

## 3.39

交流系数 AC coefficient

AC系数

二维变换域上索引号不全为0的变换系数。

## 3.40

解码处理 decode processing

包括解析过程和解码过程。

## 3.41

解码过程 decode process

由语法元素产生解码图像的过程。

## 3.42

解码器 decoder

完成解码处理的实体。

## 3.43

解码顺序 decoding order

解码过程根据图像之间的预测关系，对每帧图像解码的顺序。与比特流顺序相同。

## 3.44

解码图像 decoded picture

解码器根据比特流重建的图像。

## 3.45

解码图像缓冲区 decoded picture buffer

保存解码图像并用于预测、输出重排序和输出定时的缓冲区。

## 3.46

解析过程 parse

由比特流获得语法元素的过程。

## 3.47

禁止 forbidden

定义了一些特定语法元素值，这些值不应出现在符合本标准的比特流中。禁止某些值的目的是通常是为了避免在比特流中出现伪起始码。

## 3.48

某档次解码器 x-level decoder

能够解码符合某档次规定的比特流的解码器。

3.49

起始码前缀 start code prefix

长度为三个字节的序列，值为0x000001。起始码前缀包含在字节流中，作为每个网络抽象层单元的前缀。

3.50

前向预测 forward prediction

用显示顺序上过去的参考图像对当前图像进行预测。

3.51

前向帧间解码图像 forward inter decoded picture

P帧

帧间预测中只使用前向预测解码的图像。

3.52

色度 chroma

两种色差信号的一种的样值矩阵或单个样值。

注：色度的符号是Cr和Cb。

3.53

输出处理过程 output processing

由解码图像得到输出帧的过程。

3.54

输出顺序 output order

输出解码图像的顺序，与显示顺序相同。

3.55

随机访问 random access

从某一点而非比特流起始点开始对比特流解码并恢复出解码图像。

3.56

随机访问点 random access point

比特流中能进行随机访问并能够正确恢复解码图像的点。

3.57

条带 slice

按光栅扫描顺序的若干连续宏块。

3.58

条带头 slice header

编码的条带的一部分，是条带中宏块公用数据元素的编码表示。

3.59

跳过的宏块 skipped macroblock

除“跳过”指示外，无其它编码数据的宏块。

3.60

图像参数集 picture parameter set

一种语法结构，包含适用于零个或更多编码图像的语法元素。

3.61

图像重排序 picture reordering

如果解码顺序和输出顺序不同，解码图像进行重排序的过程。

## 3.62

网络抽象层单元 NAL unit

NAL单元

一种语法结构，包含数据类型和原始字节序列净载（RBSP）数据。

## 3.63

网络抽象层单元流 NAL unit stream

NAL单元流

一个网络抽象层单元序列。

## 3.64

显示顺序 display order

显示解码图像的顺序。

## 3.65

序列参数集 sequence parameter set

一种语法结构，包含适用于零个或多个编码视频序列的语法元素。

## 3.66

样本 sample

构成图像的基本元素。

## 3.67

样本宽高比 weight height ratio

一帧图像中亮度样本列间的水平距离与行间的垂直距离之比。表示为 $h:v$ ，其中 $h$ 是水平宽度， $v$ 是垂直高度。

## 3.68

样值 sample value

样本的幅值。

## 3.69

游程 run

在解码过程中若干连续的相同数据元素。一方面指在块扫描中一个非0系数前值为0的系数的个数；另一方面指跳过的宏块的数目。

## 3.70

预测 prediction

预测过程的具体实现。

## 3.71

预测过程 prediction process

使用预测器对当前解码样值或者数据元素进行估计。

## 3.72

预测值 prediction value

在后续样值或数据元素的解码过程中，使用的先前解码的样值或数据元素的组合。

## 3.73

语法结构 syntax structure

比特流中按规定顺序排列的零个或多个语法元素的集合。

## 3.74

语法元素 syntax element

比特流中的数据单元解析后的结果。

## 3.75

源 source

描述编码前视频素材或其某些属性的术语。

### 3.76

原始字节序列净载 raw byte sequence payload

RBSP

一种语法结构，包含封装于一个网络抽象层单元的整数个字节。一个RBSP可以是空的，也可以是一个包含语法元素的比特串加上一个等于‘1’的比特和零个或更多等于‘0’的比特。

### 3.77

运动矢量 motion vector

用于帧间预测的二维矢量，其值为从当前块坐标到参考块坐标的偏移。

### 3.78

直流系数 DC coefficient

DC系数

二维变换域上索引号全为0的变换系数。

### 3.79

帧 frame

视频信号空间信息的表示，由一个亮度样本矩阵（Y）和两个色度样本矩阵（Cb和Cr）构成。

### 3.80

帧间编码 inter coding

使用帧间预测对宏块或图像进行编码。

### 3.81

帧间预测 inter prediction

使用先前解码图像生成当前图像样本预测值的过程。

### 3.82

帧内编码 intra coding

使用帧内预测对宏块或图像进行编码。

### 3.83

帧内解码图像 intra decoded picture

I帧

只使用帧内预测解码的图像。

### 3.84

帧内预测 intra prediction

在相同解码图像中使用先前解码的样值生成当前样本预测值的过程。

### 3.85

字节 byte

8位的比特串。

### 3.86

字节对齐 byte alignment

从比特流的第一个比特开始，如果某比特的位置是8的整数倍，则该比特是字节对齐的。

### 3.87

字节流 byte stream

包含起始码前缀和网络抽象层单元的网络抽象层单元流的封装。

## 4 缩略语

CBR: 恒定比特率 (Constant Bit Rate)  
 CIF: 通用中间格式 (Common Intermediate Format )  
 HRD: 虚拟参考解码器 (Hypothetical Reference Decoder)  
 IDR: 即时解码刷新 (Instantaneous Decoding Refresh)  
 LSB: 最低有效位 (Least Significant Bit)  
 MB: 宏块 (Macroblock)  
 MSB: 最高有效位 (Most Significant Bit)  
 NAL: 网络抽象层 (Network Abstraction Layer)  
 QCIF: 四分之一通用中间格式 (Quarter Common Intermediate Format)  
 RBSP: 原始字节序列净载 (Raw Byte Sequence Payload)  
 SEI: 辅助增强信息 (Supplemental Enhancement Information)  
 VBR: 可变比特率 (Variable Bit Rate)  
 VLC: 变长编码 (Variable Length Coding)

## 5 约定

GB/T XXXXX的本部分中使用的数学运算符和优先级与C语言使用的类似。但对整型除法和算术移位操作进行了特定定义。除特别说明外，约定编号和计数从0开始。

### 5.1 算术运算符

算术运算符定义如下：

+	加法运算。
-	减法运算（二元运算符）或取反（一元前缀运算符）。
×	乘法运算。
$a^b$	幂运算，表示 $a$ 的 $b$ 次幂。也可表示上标。
/	整除运算，沿向0的取值方向截断。例如， $7/4$ 和 $-7/-4$ 截断至1， $-7/4$ 和 $7/-4$ 截断至-1。
÷	除法运算，不做截断或四舍五入。
$\frac{a}{b}$	除法运算，不做截断或四舍五入。
$\sum_{i=a}^b f(i)$	自变量 $i$ 取由 $a$ 到 $b$ （含 $b$ ）的所有整数值时，函数 $f(i)$ 的累加和。
$a \% b$	模运算， $a$ 除以 $b$ 的余数，其中 $a$ 与 $b$ 都是正整数。

### 5.2 逻辑运算符

逻辑运算符定义如下：

$a \ \&\& \ b$	$a$ 和 $b$ 之间的与逻辑运算
$a \    \ b$	$a$ 和 $b$ 之间的或逻辑运算
!	逻辑非运算

### 5.3 关系运算符

关系运算符定义如下：

>	大于
>=	大于或等于
<	小于
<=	小于或等于
==	等于
!=	不等于

## 5.4 位运算符

位运算符定义如下：

&            与运算

|            或运算

~            取反运算

$a \gg b$         将 $a$ 以2的补码整数表示的形式向右移 $b$ 位。仅当 $b$ 取正数时定义此运算。

$a \ll b$         将 $a$ 以2的补码整数表示的形式向左移 $b$ 位。仅当 $b$ 取正数时定义此运算。

## 5.5 赋值

赋值运算定义如下：

=            赋值运算符

++           递增， $x++$ 相当于 $x = x + 1$ 。当用于数组下标时，在递增运算前先求变量值。

—           递减， $x--$ 相当于 $x = x - 1$ 。当用于数组下标时，在递减运算前先求变量值。

+=           自加指定值，例如 $x += 3$ 相当于 $x = x + 3$ ， $x += (-3)$ 相当于 $x = x + (-3)$ 。

-=           自减指定值，例如 $x -= 3$ 相当于 $x = x - 3$ ， $x -= (-3)$ 相当于 $x = x - (-3)$ 。

## 5.6 数学函数

数学函数定义如下：

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases} \quad (1)$$

$$\text{Ceil}(x) \quad \text{取不小于} x \text{的最小整数} \quad (2)$$

$$\text{Clip1}(x) = \text{Clip3}(0, 255, x) \quad (3)$$

$$\text{Clip3}(a, b, c) = \begin{cases} a & ; \quad c < a \\ b & ; \quad c > b \\ c & ; \quad \text{其它} \end{cases} \quad (4)$$

$$\text{Floor}(x) \quad \text{取不大于} x \text{的最大整数} \quad (5)$$

$$\text{Log}_2(x) \quad \text{取以2为底的} x \text{的对数} \quad (6)$$

$$\text{Log}_{10}(x) \quad \text{取以10为底的} x \text{的对数} \quad (7)$$

$$\text{Median}(x, y, z) = x + y + z - \text{Min}(x, \text{Min}(y, z)) - \text{Max}(x, \text{Max}(y, z)) \quad (8)$$

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases} \quad (9)$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases} \quad (10)$$

$$\text{Round}(x) = \text{Sign}(x) \times \text{Floor}(\text{Abs}(x) + 0.5) \quad (11)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x \geq 0 \\ -1 & ; \quad x < 0 \end{cases} \quad (12)$$

## 5.7 比特流语法、解析过程和解码过程的描述

### 5.7.1 描述方法

比特流语法描述方法类似C语言。比特流的语法元素使用粗体字表示，每个语法元素通过名字（由下划线分割的英文字母组成，所有字母都是小写）、语法和语义来描述。语法表和正文中语法元素的值用常规字体表示。

某些情况下，可在语法表中应用从语法元素导出的其它变量值，这样的变量在语法表或正文中用不带下划线的小写字母和大写字母混合命名。大写字母开头的变量用于解码当前以及相关的语法结构，也可用于解码后续的语法结构。小写字母开头的变量只在它们所在的小节内使用。

语法元素值的助记符和变量值的助记符与它们的值之间的关系在正文中说明。在某些情况下，二者等同使用。助记符由一个或多个使用下划线分隔的字母组表示，每个字母组以大写字母开始，也可包括多个大写字母。

比特串的长度是4的整数倍时，可使用十六进制符号表示。十六进制的前缀是‘0x’，例如‘0x1a’表示比特串‘0001 1010’。

条件语句中0表示FALSE，非0表示TRUE。

语法表描述了所有符合本标准的比特流语法的超集，附加的语法限制在相关小节中说明。

下面给出了描述语法的伪代码例子。当语法元素出现时，表示从比特流中读一个数据单元。

	描述符
/*语句是一个语法元素的描述符，或者说明语法元素的存在、类型和数值，下面给出两个例子。*/	
<b>syntax_element</b>	ue(v)
conditioning statement	
/*花括号括起来的语句组是复合语句，在功能上视作单个语句。*/	
{	
statement	
statement	
...	
}	
/*“while”语句测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	
while ( condition )	
Statement	
/*“do ... while”语句先执行循环体一次，然后测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	
Do	
Statement	
while ( condition )	
/*“if ... else”语句首先测试condition，如果为TRUE，则执行primary语句，否则执行alternative语句。如果alternative语句不需要执行，结构的“else”部分和相关的alternative语句可忽略。*/	
if ( condition )	
primary statement	
else	
alternative statement	



/* “for” 语句首先执行initial语句，然后测试condition，如果conditon为TRUE， 则重复执行primary语句和subsequent语句直到condition不为TRUE。*/	
for ( initial statement; condition; subsequent statement )	
primary statement	

解析过程和解码过程用文字和类似C语言的伪代码描述。

5.7.2 函数

以下函数用于语法描述。假定解码器中存在一个比特流指针，这个指针指向比特流中要读取的下一个比特的位置。函数由函数名及左右圆括号内的参数构成。函数也可没有参数。

**byte\_aligned( )**

如果比特流的当前位置是字节对齐的，返回TRUE，否则返回FALSE。

**byte\_aligned\_next\_bits(n)**

如果比特流当前位置不是字节对齐的，返回比特流当前位置的下一个字节开始的n个比特，MSB在前，不改变比特流指针；如果比特流当前位置是字节对齐的，返回比特流随后的n个比特，MSB在前，不改变比特流指针。如果剩余的比特少于n，则返回0。

**more\_data\_in\_byte\_stream ( )**

用于字节流NAL单元语法结构，如果字节流中有更多数据，返回TRUE，否则返回FALSE。

**more\_rbsp\_data( )**

如果一个Rbsp中有更多的数据位于rbsp\_trailing\_bits()之前，返回TRUE，否则返回FALSE。

**next\_bits(n)**

返回比特流的随后n个比特，MSB在前，不改变比特流指针。如果剩余的比特少于n，则返回0。

**read\_bits(n)**

返回比特流的随后n个比特，MSB在前，同时比特流指针前移n个比特。如果n等于0，则返回0，比特流指针不前移。

函数也用于解析过程和解码过程的描述。

5.7.3 描述符

以下描述符表示不同语法元素的解析过程。

**b(8)**

一个任意取值的字节。解析过程由函数read\_bits(8)的返回值规定。

**ce4(v)**

变长编码的语法元素。解析过程在8.3中定义。

**f(n)**

取特定值的连续n个比特。解析过程由函数read\_bits(n)的返回值规定。

**i(n)**

n位整数。在语法表中，如果n是‘v’，其比特数由其它语法元素值确定。解析过程由函数read\_bits(n)的返回值规定，该返回值用高位在前的2的补码表示。

**me(v)**

用指数哥伦布码编码的语法元素。解析过程在8.2中定义。

**se(v)**

有符号整数语法元素，用指数哥伦布码编码。解析过程在8.2中定义。

**u(n)**

n位无符号整数。在语法表中，如果n是‘v’，其比特数由其它语法元素值确定。解析过程由函数read\_bits(n)的返回值规定，该返回值用高位在前的二进制表示。

**ue(v)**

无符号整数语法元素，用指数哥伦布码编码。解析过程在8.2中定义。

5.7.4 保留和禁止

本部分定义的比特流语法中,某些语法元素的值被标注为‘保留’(reserved)或‘禁止’(forbidden)。  
‘保留’定义了一些特定语法元素值用于将来对本部分的扩展。这些值不应出现在符合本部分的比特流中。  
‘禁止’定义了一些特定语法元素值,这些值不应出现在符合本部分的比特流中。  
‘标记’(marker\_bit)指该比特的值应为‘1’。  
比特流中的‘保留位’(reserved\_bits)表明保留了一些语法单元用于将来对本部分的扩展,解码处理应忽略这些比特。

6 编码比特流的结构

本章说明编码后比特流的结构及其层次关系和处理顺序。  
6.1 比特流结构

符合本部分的比特流为网络抽象层单元流或字节流。一个网络抽象层单元流包含一系列按解码顺序排列的网络抽象层单元序列。一个字节流由网络抽象层单元流组成,网络抽象层单元流包含一系列按解码顺序排列的网络抽象层单元序列,每个网络抽象层单元前包含起始码前缀。

6.2 编码视频序列

编码视频序列是比特流的最高层语法结构。编码视频序列由一个IDR图像开始,后面跟着零个或多个非IDR图像,直到但不包括下一个IDR图像或直到比特流结尾。编码图像在比特流中按比特流顺序排列,比特流顺序应与解码顺序相同。

6.3 图像

一幅图像是一帧。帧由三个样本矩阵构成,包括一个亮度样本矩阵(Y)和两个色度样本矩阵(Cb和Cr)。样本矩阵元素的值为整数。Y,Cb和Cr三个分量与原始的(模拟)红、绿和蓝色信号之间的关系,包括原始信号的色度和转移特性等可在比特流中定义,这些信息不影响解码过程。  
解码器的输出是一系列帧,两帧之间存在着一个帧时间间隔。  
图像的解码处理包括解析过程和解码过程。

6.3.1 图像格式

本部分支持4:2:0格式的图像,Cb和Cr矩阵水平和垂直方向的尺寸都只有Y矩阵的一半。  
亮度和色度样本位置见图1,图中“○”代表亮度样本,“×”代表色度样本。

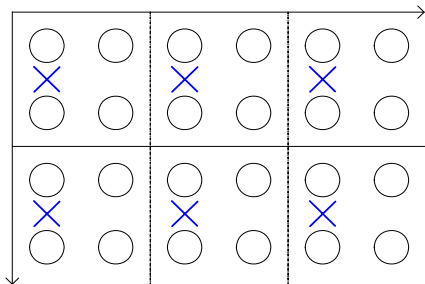


图1 亮度和色度样本位置

6.3.2 图像类型

本部分定义了两种解码图像:  
——帧内解码图像(I 帧);  
——前向帧间解码图像(P 帧)。

6.3.3 参考图像

P帧最多可参考前向的两帧。运动矢量所指的参考像素可超出参考图像的边界,在这种情况下对超出参考图像边界的整数样本应使用距离该整数参考样本所指位置最近的图像内的整数样本进行边界扩

展。对亮度样本矩阵，参考块的像素在水平和垂直方向均不应超出参考图像边界外16个像素。对色度样本矩阵，参考块的像素在水平和垂直方向均不应超出参考图像边界外8个像素。

6.4 条带

条带是按光栅扫描顺序连续的若干宏块，条带内的宏块不应重叠，条带之间也不应重叠。除环路滤波外，条带内宏块的解码处理不应使用本图像其它条带的数据，见图2。

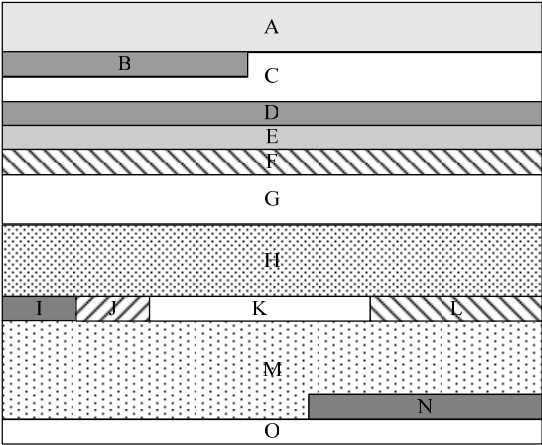


图2 条带结构

6.5 宏块

图像划分为宏块，宏块左上角的点不应超出图像边界。

宏块的划分见图3，这种划分用于运动补偿。图3中矩形里的数字表示宏块划分后运动矢量和参考索引在码流中的顺序。

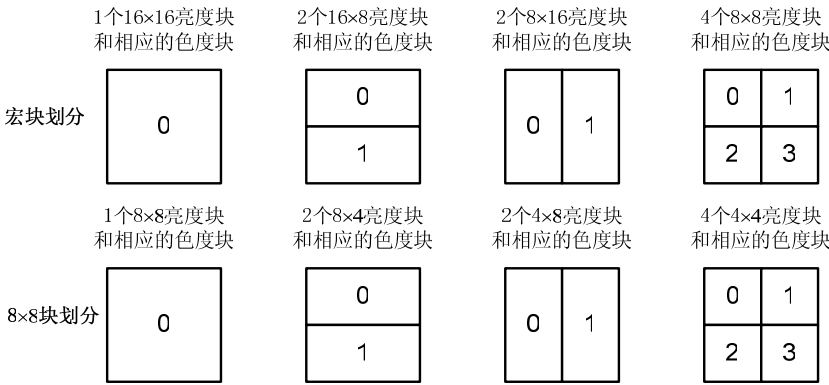


图3 宏块的划分

6.6 8×8 块

一个宏块包括4个8×8亮度块（Y）和2个8×8色度块（1个Cb，1个Cr），见图4，图中数字为宏块中8×8块的顺序号。

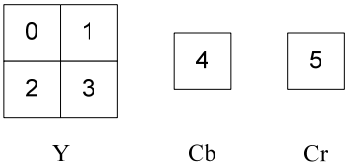


图4 宏块划分为8×8块

宏块中的各个8×8块在比特流中出现的顺序由图4中的数字规定。

6.7 4×4 块

一个宏块包括16个4×4亮度块（Y）和8个4×4色度块（4个Cb，4个Cr），见图5，图中数字为宏块中4×4块的顺序号。

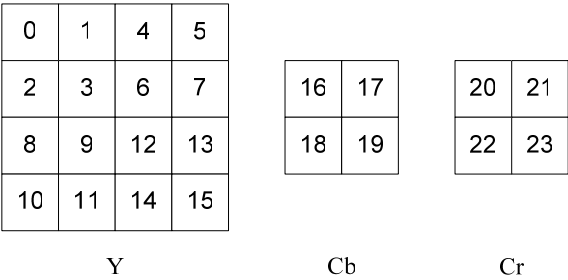


图5 宏块划分为4×4块

宏块中的各个4×4块在比特流中出现的顺序由图5中的数字规定。

7 比特流的语法和语义

7.1 语法描述

7.1.1 NAL 单元定义

如下：

nal_unit( NumBytesInNALunit ) {	描述符
<b>forbidden_zero_bit</b>	f(1)
<b>nal_ref_idc</b>	u(2)
<b>nal_unit_type</b>	u(5)
for ( i = 0; i < NumBytesInNALunit-1; i++ ) {	
<b>rbsp_byte[i]</b>	b(8)
}	
}	

7.1.2 序列参数集定义

如下：

seq_parameter_set_rbsp( ) {	描述符
profile_idc	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
delta_time_picture_distance_1	u(16)
num_ref_frames	ue(v)
horizontal_size_minus1	ue(v)
vertical_size_minus1	ue(v)
aspect_ratio	u(4)
frame_cropping_flag	u(1)
if ( frame_cropping_flag == '1' ) {	
frame_crop_left_offset	ue(v)
frame_crop_right_offset	ue(v)
frame_crop_top_offset	ue(v)
frame_crop_bottom_offset	ue(v)
}	
hrd_parameters_present_flag	u(1)
if ( hrd_parameters_present_flag == '1' )	
hrd_parameters( )	
rbsp_trailing_bits( )	
}	

7.1.3 图像参数集定义

如下：

pic_parameter_set_rbsp( ) {	描述符
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
fixed_picture_qp_flag	u(1)
picture_reference_flag	u(1)
if ( ! picture_reference_flag )	
sliding_window_size_flag	u(1)
skip_mode_flag	u(1)
loop_filter_disable_flag	u(1)
if ( ! loop_filter_disable_flag )	
loop_filter_parameter_flag	u(1)
constrained_intra_pred_flag	u(1)
if ( more_rbsp_data( ) )	
half_pixel_mv_flag	u(1)
rbsp_trailing_bits( )	
}	

7.1.4 辅助增强信息和消息定义

7.1.4.1 辅助增强信息定义

如下：

sei_rbsp() {	描述符
do {	
sei_message( )	
} while( more_rbsp_data( ) )	
}	

#### 7.1.4.2 辅助增强信息消息定义

如下：

sei_message() {	描述符
payloadType = 0	
while ( next_bits(8) == 0xFF ) {	
<b>ff_byte</b> /* equal to 0xFF */	f(8)
payloadType += 255	
}	
<b>last_payload_type_byte</b>	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while ( next_bits(8) == 0xFF ) {	
<b>ff_byte</b> /* equal to 0xFF */	f(8)
payloadSize += 255	
}	
<b>last_payload_size_byte</b>	u(8)
payloadSize += last_payload_size_byte	
sei_payload( payloadType, payloadSize )	
}	

#### 7.1.5 随机访问点指示定义

如下：

random_access_point_indicator_rbsp() {	描述符
}	

#### 7.1.6 图像头定义

如下：

picture_header_rbsp() {	描述符
<b>picture_coding_type</b>	u(2)
<b>picture_distance</b>	u(8)
<b>pic_parameter_set_id</b>	ue(v)
if (nal_unit_type == 2 )	
<b>picture_distance_gap_minus1</b>	ue(v)
<b>frame_num</b>	u(5)
<b>picture_qp</b>	u(6)

if ( loop_filter_parameter_flag != 0 ) {	
<b>alpha_ci_offset</b>	se(v)
<b>cp_offset</b>	se(v)
<b>loop_filter_qp_offset</b>	se(v)
}	
if ( sliding_window_size_flag )	
<b>sliding_window_size_minus1</b>	u(1)
rbsp_trailing_bits( )	
}	

### 7.1.7 条带层定义

如下：

slice_layer_rbsp( ) {	描述符
slice_header( )	
slice_data( )	
rbsp_trailing_bits( )	
}	

#### 7.1.7.1 条带头定义

如下：

slice_header( ) {	描述符
<b>first_mb_in_slice</b>	ue(v)
if ( fixed_picture_qp_flag == '0' ) {	
<b>fixed_slice_qp_flag</b>	u(1)
<b>slice_qp_delta</b>	se(v)
}	
if ( ! loop_filter_disable_flag )	
<b>disable_loop_filter_slice_flag</b>	u(1)
}	

#### 7.1.7.2 条带数据定义

如下：

slice_data( ) {	描述符
do {	
if ( PictureType != 0 ) {	
if ( skip_mode_flag == '1' )	
<b>mb_skip_run</b>	ue(v)
}	
if ( MbIndex < MbWidth × MbHeight && more_rbsp_data( ) )	
macroblock( )	
} while ( more_rbsp_data( ) )	
}	

### 7.1.8 宏块定义

如下：

macroblock( ) {	描述符
<b>mb_type</b>	ue(v)   u(1)

if ( MbType != 'P_Skip' ) {	
if ( MbType == 'I_4x4' ) {	
for ( i=0; i<16; i++ ) {	
<b>pred_mode_flag</b>	u(1)
if ( pred_mode_flag )	
<b>intra_luma_pred_mode</b>	u(3)
}	
<b>intra_chroma_pred_mode</b>	ue(v)
}	
else if (MbType == 'I_Direct' ) {	
<b>intra_chroma_pred_mode</b>	ue(v)
}	
else {	
if ( IsRef0anIdrPic == 0 && picture_reference_flag == 0 ) {	
for ( i = 0; i < MvNum; i++ )	
<b>mb_reference_index</b>	u(1)
}	
if ( MbType != 'P_8x8' ) {	
for ( i = 0; i < MvNum; i++ ) {	
<b>mv_diff_x</b>	se(v)
<b>mv_diff_y</b>	se(v)
}	
}	
else {	
for ( i = 0; i < 4; i++ )	
<b>sub_mb_type</b>	ue(v)
for ( i = 0; i < 4; i++ ) {	
for ( j = 0; j < SubMvNum[ sub_mb_type[ i ] ]; j++ ) {	
<b>mv_diff_x</b>	se(v)
<b>mv_diff_y</b>	se(v)
}	
}	
}	
}	
if ( ! (MbTypeIndex >= 5) )	
<b>cbp</b>	me(v)
for ( i = 0; i < 6; i++ ) {	
if ( MbCBP & ( 1 << i ) )	
<b>cbp_4x4</b>	u(4)
}	
if ( MbCBP > 0 && ! fixed_slice_qp_flag )	
<b>mb_qp_delta</b>	se(v)
for ( i = 0; i < 6; i++ )	



block( i )	
}	
}	

7.1.9 块定义

如下：

block( i ) {	描述符
if ( i < 6 && ( MbCBP & ( 1 << i ) ) ) {	
for ( j=0; j<4; j++ ) {	
if ( cbp_4x4 & ( 1 << j ) ) {	
do {	
<b>trans_coefficient_4</b>	ce4(v)
if ( trans_coefficient_4 >= 71 )	
<b>escape_run</b>	ce4(v)
} while ( trans_coefficient_4 != "EOB" && ! LastCoefficient )	
}	
}	
}	
}	

7.1.10 RBSP 结尾定义

如下：

rbsp_trailing_bits() {	描述符
<b>rbsp_stop_one_bit</b> /* equal to 1 */	f(1)
while( ! byte_aligned() )	
<b>rbsp_alignment_zero_bit</b> /* equal to 0 */	f(1)
}	

7.2 语义描述

7.2.1 NAL 单元

NumBytesInNALunit规定该NAL单元的大小，以字节为单位。这个值在解码时需要用到。如何得到该值不在本标准规定的范围之内。

注：NumBytesInNALunit的值通常由系统层给出。

禁止位 forbidden\_zero\_bit

值应为 ‘0’ 。

NAL参考标号 nal\_ref\_idc

2位无符号整数。值不等于0说明该NAL单元包含的数据为序列参数集、图像参数集或参考帧数据；值等于0说明该NAL单元包含数据为非参考帧数据。

对于序列参数集或图像参数集NAL单元，nal\_ref\_idc不应为0。对于某一帧，如果某一NAL单元的nal\_ref\_idc为0，则该帧所有NAL单元的nal\_ref\_idc均应为0。IDR图像的NAL单元的nal\_ref\_idc不应为0。nal\_unit\_type等于6的NAL单元的nal\_ref\_idc应为0。nal\_unit\_type等于2、7的NAL单元的nal\_ref\_idc不应为0。

NAL单元类型 nal\_unit\_type

5位无符号整数。规定NAL单元中RBSP数据结构类型，见表1。

如果一个编码图像中的图像头所在的NAL单元的nal\_unit\_type等于2，则该图像称为IDR图像。

表 1 NAL 单元类型

nal_unit_type	NAL单元和Rbsp语法结构所含内容
0	未规定
1	非IDR图像的图像头picture_header_rbsp( )
2	IDR图像的图像头picture_header_rbsp( )
3	条带slice_layer_rbsp( )
4	序列参数集seq_parameter_set_rbsp( )
5	图像参数集pic_parameter_set_rbsp( )
6	辅助增强信息sei_rbsp( )
7	随机访问点指示random_access_point_indicator_rbsp( )
8~23	保留
24~31	未规定

Rbsp字节      rbsp\_byte

任意取值的字节。

7.2.1.1 NAL 单元的顺序以及对应到编码图像、访问单元和编码图像序列

本条描述对比特流中NAL单元排列顺序的限制。任何符合这些限制的NAL单元的顺序叫做NAL单元的解码顺序。在一个NAL单元内部，解码顺序由相应语法表规定。符合本部分的解码器应能按照解码顺序接收NAL单元和NAL单元的语法元素。

7.2.1.2 图像参数集、序列参数集的顺序和激活

一个图像参数集包含可以被一个或多个编码图像的NAL单元引用的参数。解码过程开始时，所有图像参数集都处于非激活状态。在解码过程中的任一时刻，最多只有一个图像参数集处于激活状态。当一个图像参数集被激活时，此前处于激活状态的图像参数集变为非激活状态。

当一个处于非激活状态的图像参数集被一个编码图像的图像头引用时，该图像参数集被激活。图像参数集被激活前，解码过程应能得到该图像参数集。

任何picture\_parameter\_set\_id值等于激活图像参数集的picture\_parameter\_set\_id值的图像参数集，其内容应与激活图像参数集的内容相同，除非该图像参数集在一帧编码图像的最后一个条带NAL单元之后并且在另一帧编码图像第一个图像头NAL单元之前。

一个序列参数集包含的参数被一个或多个图像参数集或一个或多个包含HRD缓冲区参数SEI消息的NAL单元引用。解码过程开始时，所有序列参数集都处于非激活状态。在解码过程中的任一时刻，最多只有一个序列参数集处于激活状态。当一个序列参数集被激活时，此前处于激活状态的序列参数集变为非激活状态。

如果发生以下两种情况之一，一个处于非激活状态的序列参数集被激活：

- 一个图像参数集被激活并且该图像参数集引用了这个序列参数集；
- 该序列参数集被一个包含 HRD 缓冲区参数 SEI 消息的 NAL 单元引用。

激活前解码过程应能得到该序列参数集。一个激活的序列参数集在整个编码图像序列内均应处于激活状态。

任何seq\_parameter\_set\_id值等于激活序列参数集的seq\_parameter\_set\_id值的序列参数集，其内容应与激活序列参数集的内容相同，除非该序列参数集在一个编码图像序列的最后一个访问单元之后并且在另一个编码图像序列第一个图像头NAL单元和第一个包含HRD缓冲区参数SEI消息的NAL单元之前。

7.2.1.3 访问单元的顺序以及对应到编码图像序列

符合本部分的码流包含一个或多个编码图像序列。

一个编码图像序列包含一个或多个访问单元。编码图像序列的第一个访问单元应为IDR访问单元，所有其它访问单元均应是IDR访问单元。

如果一个SEI NAL单元包含的数据对一个以上的访问单元有效，该SEI NAL单元应包含于其中第一个访问单元（解码顺序）。

7.2.1.4 NAL 单元和编码图像的顺序以及对应到访问单元

- 一个访问单元包含一个编码图像，除图像头NAL单元和条带NAL单元外，还可以包含其它NAL单元。
- 比特流中第一个访问单元应开始于比特流中第一个NAL单元。一个访问单元最后一个条带NAL单元后的第一个NAL单元为另一个访问单元的开始。
- 访问单元内的编码图像的图像头NAL单元、条带NAL单元和其它NAL单元的顺序应满足以下限制：
- 如果存在，随机访问点指示 NAL 单元应为访问单元的第一个 NAL 单元。
  - 任何 SEI NAL 单元应在编码图像之前。
  - 包含 HRD 缓冲区参数 SEI 消息的 NAL 单元，该 HRD 缓冲区参数 SEI 消息应为该访问单元中第一个 SEI NAL 单元的第一个 SEI 消息净载。
  - nal\_unit\_type 值等于 0，8~31 的 NAL 单元不宜出现在一个编码图像第一个图像头 NAL 单元之前。
  - 条带 NAL 单元应按 first\_mb\_in\_slice 的值从小到大的顺序排列。

7.2.2 序列参数集

**档次标号**     **profile\_idc**  
8位无符号整数。表示比特流的档次。

**级别标号**     **level\_idc**  
8位无符号整数。表示比特流的级别。  
档次和级别见附录B。

**序列参数集标号**   **seq\_parameter\_set\_id**  
序列参数集的标号。解析过程见8.2。seq\_parameter\_set\_id的取值范围是0~15。

**图像间距时间差**   **delta\_time\_picture\_distance\_1**  
16位无符号整数，表示当图像间距picture\_distance之差为1时对应的的时间差值，为1/90000s。

**参考帧数**       **num\_ref\_frames**  
指定一个编码图像序列中解码图像时需要用到的最大参考帧数，取值范围是0~2。解析过程见8.2。

**水平尺寸**       **horizontal\_size\_minus1**  
规定图像的水平尺寸。解析过程见8.2。  
MbWidth按以下方法计算：

$$\text{MbWidth} = \text{horizontal\_size\_minus1} + 1$$

horizontal\_size\_minus1的单位是宏块数。

**垂直尺寸**       **vertical\_size\_minus1**  
规定图像的垂直尺寸。解析过程见8.2。  
MbHeight按以下方法计算：

$$\text{MbHeight} = \text{vertical\_size\_minus1} + 1$$

vertical\_size\_minus1的单位是宏块数。

**宽高比**   **aspect\_ratio**  
4位无符号整数。规定重建图像的样本宽高比（SAR）或显示宽高比（DAR），见表2。

表 2 宽高比

aspect_ratio	样本宽高比（SAR）	显示宽高比（DAR）
0	禁止	禁止
1	1.0	-

表 2（续） 宽高比

aspect_ratio	样本宽高比（SAR）	显示宽高比（DAR）
2	-	4 ÷ 3
3	-	16 ÷ 9
4	-	2.21 ÷ 1
5~15	-	保留

帧裁剪标志 `frame_cropping_flag`

标志。值为‘1’表示比特流中存在帧裁剪参数；值为‘0’表示比特流中不存在帧裁剪参数。

帧裁剪参数 `frame_crop_left_offset`、`frame_crop_right_offset`、`frame_crop_top_offset`、`frame_crop_bottom_offset`

指定位于一个矩形内的图像样本，至少应有一个图像样本包含于该矩形内。解析过程见8.2。

对于亮度矩阵，该矩形的左上角坐标为(x1, y1)，右下角坐标为(x2, y2)，则：

$$x1 = 2 \times \text{frame\_crop\_left\_offset}$$
$$y1 = 2 \times \text{frame\_crop\_top\_offset}$$
$$x2 = \text{MbWidth} \times 16 - (2 \times \text{frame\_crop\_right\_offset} + 1)$$
$$y2 = \text{MbHeight} \times 16 - (2 \times \text{frame\_crop\_bottom\_offset} + 1)$$

对于色度矩阵，该矩形的左上角坐标为(x1/2, y1/2)，右下角坐标为(x2/2, y2/2)。

`frame_crop_left_offset`的取值范围是0~(8 × MbWidth - (frame\_crop\_right\_offset + 1))，`frame_crop_top_offset`的取值范围是0~(8 × MbHeight - (frame\_crop\_bottom\_offset + 1))。如果`frame_cropping_flag`等于‘0’，则`frame_crop_left_offset`、`frame_crop_right_offset`、`frame_crop_top_offset`和`frame_crop_bottom_offset`的值均为0。

HRD参数标志 `hrd_parameters_present_flag`

标志。值为‘1’表示比特流中存在HRD参数；值为‘0’表示比特流中不存在HRD参数。

7.2.3 图像参数集

图像参数集标号 `pic_parameter_set_id`

图像参数集的标号。解析过程见8.2。`pic_parameter_set_id`的取值范围是0~127。

序列参数集标号 `seq_parameter_set_id`

序列参数集的标号。解析过程见8.2。`seq_parameter_set_id`的取值范围是0~15。

固定图像量化参数 `fixed_picture_qp_flag`

标志。值为‘1’说明在该帧图像内量化参数不变；‘0’说明在该帧图像内量化参数可变。

图像参考标志 `picture_reference_flag`

标志。与`IsRef0anIdrPic`一起决定当前编码图像内所有帧间编码宏块是否都使用缺省参考图像。`picture_reference_flag`的值为‘1’或`IsRef0anIdrPic`的值为‘1’表示所有帧间编码宏块都使用缺省参考图像；否则表示由每个宏块在宏块层自行确定参考图像。当`picture_reference_flag`的值为‘0’时，`num_ref_frames`的值应等于2。确定参考图像及`IsRef0anIdrPic`的值的方法见9.3.3。

滑动窗口标志 `sliding_window_size_flag`

标志。值为‘1’表示图像头中包含`sliding_window_size_minus1`；值为‘0’表示图像头中不包含`sliding_window_size_minus1`。如果`sliding_window_size_flag`不存在，其值为‘0’。

宏块跳过模式标志 `skip_mode_flag`

标志。值为‘1’表示宏块跳过模式使用游程编码；值为‘0’表示宏块跳过模式的编码由`mb_type`决定，见9.3.1。

环路滤波禁用标志 `loop_filter_disable_flag`

标志。值为‘1’表示不应使用环路滤波；值为‘0’表示应使用环路滤波。

环路滤波参数标志      **loop\_filter\_parameter\_flag**

标志。值为‘1’表示图像头中包含alpha\_ci\_offset、cp\_offset和loop\_filter\_qp\_offset；值为‘0’表示图像头中没有alpha\_ci\_offset、beta\_offset和loop\_filter\_qp\_offset。如果loop\_filter\_parameter\_flag不存在，其值为‘0’。

受限帧内预测标志    **constrained\_intra\_pred\_flag**

值为‘0’表示样本帧内预测允许使用任何相邻宏块的信息；值为‘1’表示样本帧内预测只使用相邻帧内编码宏块的信息。

半像素运动矢量标志   **half\_pixel\_mv\_flag**

标志。值为‘0’表示当前帧运动矢量精度为1/4像素；值为‘1’表示当前帧运动矢量精度为1/2像素。如果比特流中没有half\_pixel\_mv\_flag，其值为‘1’。

7.2.4    辅助增强信息和消息

7.2.4.1    辅助增强信息

辅助增强信息（Supplemental Enhancement Inforamtion, SEI）在解码、显示或其它过程中起辅助性作用。

7.2.4.2    辅助增强信息消息（SEI 消息）

一个SEI NAL单元包含一个或多个SEI消息。每个SEI消息包含该SEI消息的净载类型payloadType和净载大小payloadSize。SEI净载在附录C中规定。payloadSize的单位为字节。

FF字节    **ff\_byte**

值为0xFF的字节。仅当payloadType或payloadSize大于254时才会出现。

净载类型的最后字节    **last\_payload\_type\_byte**

一个SEI消息类型的最后一个字节。

净载大小的最后字节    **last\_payload\_size\_byte**

一个SEI消息大小的最后一个字节。

7.2.5    随机访问点指示

一个随机访问点指示NAL单元所属的访问单元中的编码图像为IDR图像或该访问单元中包含逐随机访问SEI消息。一个访问单元中至多只能包含一个随机访问点指示NAL单元。如果存在，随机访问点指示NAL单元应为一个访问单元的第一个NAL单元（解码顺序），并且由此以后所有访问单元的解码不需要该随机访问点指示NAL单元之前的序列参数集NAL单元、图像参数集NAL单元或辅助增强信息NAL单元。

7.2.6    图像头

每帧编码图像的第一个NAL单元（解码顺序）应为图像头NAL单元。一帧编码图像中所有条带NAL单元前的图像头（如果存在）的内容均应相同。

图像编码方式      **picture\_coding\_type**

2位无符号整数。规定图像的编码方式，见表3。

表 3    图像编码方式

picture_coding_type	编码方式
0	帧内编码（I）
1	前向预测编码（P）
2	保留
3	保留

图像间距      **picture\_distance**

8位无符号整数。picture\_distance等于前一帧（显示顺序）的picture\_distance加1，再加上当前帧和前一帧之间被跳过的图像帧数（被跳过的帧数应小于32），最后模256。IDR图像的picture\_distance应为0。

#### 图像参数集标号 **pic\_parameter\_set\_id**

图像参数集的标号。解析过程见8.2。pic\_parameter\_set\_id的取值范围是0~127。

#### 图像间距间隔 **picture\_distance\_gap\_minus1**

表示IDR图像与前一图像（输出顺序）之间的图像间距（picture\_distance）间隔，取值范围是0~31。上述两帧图像的picture\_distance间隔等于picture\_distance\_gap\_minus1的值加1。解析过程见8.2。

#### 帧标号 **frame\_num**

5位无符号整数。frame\_num等于前一参考帧（解码顺序）的frame\_num加1模32。IDR图像的frame\_num应为0。

#### 图像量化参数 **picture\_qp**

6位无符号整数。给出图像的量化参数。量化参数取值范围是0~63。

#### $\alpha$ 偏移 **alpha\_ci\_offset**

当前图像环路滤波 $\alpha$ 的偏移，alpha\_ci\_offset取值范围是-8~8，环路滤波参数AlphaCIOffset等于alpha\_ci\_offset。如果比特流中没有alpha\_ci\_offset，AlphaCIOffset等于0。解析过程见8.2。

#### CP偏移 **cp\_offset**

当前图像环路滤波CP的偏移，cp\_offset取值范围是-16~16，环路滤波参数CP0ffset等于cp\_offset。如果比特流中没有cp\_offset，CP0ffset等于0。解析过程见8.2。

#### 跳过宏块环路滤波QP门限的偏移 **loop\_filter\_qp\_offset**

当前图像内跳过宏块环路滤波QP门限的偏移，loop\_filter\_qp\_offset取值范围是-40~23，ThresholdQP等于40加loop\_filter\_qp\_offset。如果比特流中没有loop\_filter\_qp\_offset，ThresholdQP等于40。解析过程见8.2。

#### 滑动窗口尺寸 **sliding\_window\_size\_minus1**

1位无符号整数，指定参考图像滑动窗口的尺寸。参考图像选择见9.3.3。如果sliding\_window\_size\_minus1不存在，其值为1。滑动窗口尺寸SSW为：

$$SSW = \text{sliding\_window\_size\_minus1} + 1$$

### 7.2.7 条带层

条带层包含一个条带头和条带数据。

#### 7.2.7.1 条带头

##### 条带起始宏块 **first\_mb\_in\_slice**

给出条带的第一个宏块在图像中按光栅顺序的位置，以宏块为单位，取值范围是0~(MbWidth × MbHeight - 1)。解析过程见8.2。在同一帧编码图像中，每个条带的first\_mb\_in\_slice的值应按从小到大的顺序排列。在同一帧编码图像中，first\_mb\_in\_slice的值相同的条带只应出现一次。

##### 固定条带量化参数 **fixed\_slice\_qp\_flag**

标志。值为‘1’说明在该条带内量化参数不变；‘0’说明在该条带内量化参数可变。如果fixed\_slice\_qp\_flag不存在，其值为‘1’。

##### 条带量化参数增量 **slice\_qp\_delta**

给出当前条带的初始量化参数相对于图像量化参数picture\_qp的增量，取值范围是-32~31。解析过程见8.2。

##### 条带环路滤波禁止标志 **disable\_loop\_filter\_slice\_flag**

值为‘1’表示当前条带边界不进行环路滤波；值为‘0’表示当前条带边界进行环路滤波。

#### 7.2.7.2 条带数据

**跳过宏块计数     mb\_skip\_run**

跳过宏块计数。解析过程见8.2，解码过程见9.3.1。

**7.2.8 宏块****宏块类型     mb\_type**

宏块的类型，其语义由图像类型和skip\_mode\_flag决定。解析过程见8.2，解码过程见9.3.1。

**预测模式标志     pred\_mode\_flag**

标志。值为‘0’表示根据预测模式的预测值确定帧内亮度预测模式；值为‘1’表示根据intra\_luma\_pred\_mode确定帧内亮度预测模式。

**帧内亮度预测模式     intra\_luma\_pred\_mode**

3位无符号整数。用于确定亮度块的帧内预测模式，解码过程见9.3.2。

**帧内色度预测模式     intra\_chroma\_pred\_mode**

用于确定宏块中序号为4、5的两个色度块（参见图6-4）的帧内预测模式，解析过程见8.2，解码过程见9.3.2。

**宏块参考索引     mb\_reference\_index**

1位无符号整数。先解码全部前向参考索引，然后解码全部后向参考索引，解码过程见9.3.3。

**运动矢量水平分量差值     mv\_diff\_x****运动矢量垂直分量差值     mv\_diff\_y**

运动矢量差值，单位为1/2样本或1/4样本，由half\_pel\_mv\_flag的值决定，见7.2.2和7.2.3。先解码全部前向运动矢量，然后解码全部后向运动矢量，解码过程见9.3.4。mv\_diff\_x和mv\_diff\_y的取值范围是-1024~1023。

**子宏块类型     sub\_mb\_type**

子宏块的类型，其语义由图像类型和skip\_mode\_flag决定。解析过程见8.2，解码过程见9.3.1。

**宏块编码模板     cbp**

cbp确定宏块中序号为0到5的亮度块和色度块是否包含编码数据。由cbp解析得到一个6位无符号整数MbCBP，解析过程见8.2。MbCBP的解码过程见9.3.5。

**宏块量化参数增量     mb\_qp\_delta**

给出当前宏块的量化参数相对预测量化参数的增量，取值范围是-32~31。

**4×4块编码模板     cbp\_4x4**

4位无符号整数。cbp\_4x4确定8×8块中序号为0到3的4×4块是否包含编码数据。cbp\_4x4的第*n*位为‘0’表示序号为*n*的4×4块没有非零系数，等于‘1’表示该4×4块至少有一个非零系数（序号号见6.5）。cbp\_4x4的值不应为0。

**7.2.9 块****变换系数     trans\_coefficient\_4**

用于确定游程长度和量化非零系数值的联合索引，或用于确定量化非零系数值的索引，解析过程见8.3，解码过程见9.4.1。

**转逸游程     escape\_run**

当trans\_coefficient\_4不能确定游程长度和量化非零系数值的联合索引时，escape\_run用于确定转逸系数的游程长度，解析过程见8.3，解码过程见9.4.1。

**7.2.10 RBSP 结尾****RBSP停止比特     rbsp\_stop\_one\_bit**

一个等于‘1’的比特。

**RBSP对齐比特     rbsp\_alignment\_zero\_bit**

一个等于‘0’的比特。

**8 解析过程**

本章定义语法元素的解析过程。语法元素描述符ue(v)，se(v)和me(v)的解析过程见8.2；ce4(v)的解析过程见8.3。

8.1 k阶指数哥伦布码

解析k阶指数哥伦布码时，首先从比特流的当前位置开始寻找第一个非零比特，并将找到的零比特个数记为leadingZeroBits，然后根据leadingZeroBits计算CodeNum。用伪代码描述如下：

```
leadingZeroBits = -1;
for ( b = 0; ! b; leadingZeroBits++ )
    b = read_bits(1)
CodeNum = 2leadingZeroBits + k - 2k + read_bits(leadingZeroBits + k)
```

表4给出了0阶和1阶指数哥伦布码的结构。指数哥伦布码的比特串分为‘前缀’和‘后缀’两部分。前缀由leadingZeroBits个连续的‘0’和一个‘1’构成。后缀由leadingZeroBits + k个比特构成，即表4中的x<sub>i</sub>串，x<sub>i</sub>的值为‘0’或‘1’。

表 4 k阶指数哥伦布码表

阶数	码字结构	CodeNum取值范围
k = 0	1	0
	0 1 x <sub>0</sub>	1~2
	0 0 1 x <sub>1</sub> x <sub>0</sub>	3~6
	0 0 0 1 x <sub>2</sub> x <sub>1</sub> x <sub>0</sub>	7~14
	.....	.....
k = 1	1 x <sub>0</sub>	0~1
	0 1 x <sub>1</sub> x <sub>0</sub>	2~5
	0 0 1 x <sub>2</sub> x <sub>1</sub> x <sub>0</sub>	6~13
	0 0 0 1 x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub>	14~29
	.....	.....

8.2 ue(v)，se(v)和me(v)

ue(v)，se(v)和me(v)描述的语法元素使用0阶指数哥伦布码，其解析过程为：

- ue(v)：语法元素的值等于 CodeNum。
- se(v)：根据表 5 给出的有符号指数哥伦布码的映射关系求语法元素的值。
- me(v)：根据表 6 求 MbCBP（见 9.3.1 和 9.3.5）。

表 5 se(v) 与 CodeNum 的映射关系

CodeNum	语法元素值
0	0
1	1
2	- 1
3	2
4	- 2
5	3
6	- 3
k	(- 1) <sup>k+1</sup> ×Ceil( k÷2 )



表 6 MbCBP 与 CodeNum 的映射关系

CodeNum	MbCBP xxxxxx (543210)	
	宏块帧内编码模式	宏块帧间编码模式
0	63	0
1	31	4
2	15	8
3	47	1
4	0	2
5	61	15
6	55	12
7	59	5
8	51	10
9	62	3
10	14	16
11	12	13
12	13	14
13	7	32
14	60	7
15	11	11
16	29	9
17	3	6
18	16	63
19	30	31
20	23	47
21	53	48
22	10	20
23	27	17
24	32	24
25	28	18
26	8	21
27	1	36
28	48	33
29	4	40
30	2	34

表 6（续） MbCBP 与 CodeNum 的映射关系

CodeNum	MbCBP xxxxxx (543210)	26
32	宏块帧内编码模式	宏块帧间编码模式
33	58	19
34	46	28
35	21	42
36	39	29
37	43	23
38	26	27
39	49	30
40	45	44
41	17	53
42	44	46
43	35	62
44	6	35
45	9	58
46	52	61
47	24	45
48	50	39
49	56	43
50	18	55
51	20	59
52	42	49
53	37	60
54	25	52
55	36	50
56	33	56
57	57	51
58	22	25
59	54	22
60	40	38
61	34	41
62	38	57
63	41	54

ce4(v) 描述的语法元素采用0阶或1阶指数哥伦布码进行解析，其阶数由下面的规则决定：

——本部分定义了 18 个与 ce4(v) 有关的变长码表，即 VLC\_Intra0, VLC\_Intra1, VLC\_Intra2, VLC\_Intra3, VLC\_Intra4, VLC\_Intra5, VLC\_Intra6, VLC\_Inter0, VLC\_Inter1, VLC\_Inter2, VLC\_Inter3, VLC\_Inter4, VLC\_Inter5, VLC\_Inter6, VLC\_Chroma0, VLC\_Chroma1, VLC\_Chroma2, VLC\_Chroma3，见附录 A。不同的码表决定了 ce4(v) 所用的指数哥伦布码的阶数。其中 VLC\_Intra0, VLC\_Intra2, VLC\_Intra3, VLC\_Intra5, VLC\_Inter0, VLC\_Chroma0 采用 1 阶指数哥伦布码，其它的码表采用 0 阶指数哥伦布码。

——escape\_run 采用 0 阶指数哥伦布码进行解析。

上述 18 个码表的选择过程见 9.4.1。

ce4(v) 描述的语法元素解析过程如下：

——语法元素 trans\_coefficient\_4 等于 CodeNum。

——语法元素 escape\_run 等于 CodeNum。

## 9 解码过程

本章定义解码过程。

### 9.1 图像头解码

图像头解码过程如下：

如果 picture\_code\_type 等于 0，则表示图像是 I 帧，PictureType 等于 0。

如果 picture\_code\_type 等于 1，则表示图像是 P 帧，PictureType 等于 1。

当前图像的宏块索引 MbIndex 初始化为 0。

### 9.2 条带解码

条带解码过程如下：

宏块索引 MbIndex 等于 first\_mb\_in\_slice。

如果 fixed\_picture\_qp\_flag 等于 ‘1’，条带的初始量化参数 PreviousQP 等于 picture\_qp；如果 fixed\_picture\_qp\_flag 等于 ‘0’，条带的初始量化参数 PreviousQP 等于 slice\_qp\_delta 与 picture\_qp 之和。PreviousQP 的取值范围是 0~63，picture\_qp 和 slice\_qp\_delta 的取值应保证 PreviousQP 的值不超过其取值范围。

如果 mb\_skip\_run 存在，由 mb\_skip\_run 解析得到 SkipMbCount，否则 SkipMbCount 等于 0。若 SkipMbCount 不等于 0，从 MbIndex 至 MbIndex+SkipMbCount-1 的所有宏块的 MbType 设为 P\_Skip 并进行处理。处理完成后，MbIndex 应加上 SkipMbCount。

### 9.3 宏块解码

解码索引值为 MbIndex 的宏块，完成后 MbIndex 加 1。

#### 9.3.1 宏块类型

宏块类型 MbType 和宏块子类型 MbPartType 的解码过程如下：

——如果当前图像是 I 帧，MbTypeIndex 等于 0，MvNum 等于 0。

1) 如果 mb\_type 的值为 ‘1’，当前宏块是普通帧内预测模式，MbType 等于 ‘I\_4x4’。

2) 如果 mb\_type 的值为 ‘0’，当前宏块是直接帧内预测模式，MbType 等于 ‘I\_Direct’。

——如果当前图像是 P 帧

如果 skip\_mode\_flag 的值为 ‘1’，MbTypeIndex 等于 mb\_type 加 1；否则 MbTypeIndex 等于 mb\_type。如果 MbTypeIndex 大于等于 5，则 mb\_type 中包含的 CBP 信息 CBPCodeNum 等于 MbTypeIndex 减 5，然后令 MbTypeIndex 等于 5。MbType 和 MvNum 的值见表 7。

表 7 P 宏块类型

MbTypeIndex	MbType	MvNum	MbPredMode
0	P_Skip	0	前向
1	P_16x16	1	前向
2	P_16x8	2	前向
3	P_8x16	2	前向
4	P_8x8	4-16	前向
5	I_4x4	0	无

表7定义的MbType中，有Skip字样的称为跳过模式。表7中的MvNum表示宏块在比特流中的运动矢量数。如果MbType为‘P\_8x8’，则宏块在比特流中的运动矢量数由SubMvNum决定。  
子宏块类型SubMbType和SubMvNum的值见表8。

表 8 子宏块类型

sub_mb_type	SubMbType	SubMvNum
0	Sub_8x8	1
1	Sub_8x4	2
2	Sub_4x8	2
3	Sub_4x4	4

表8中的SubMvNum表示子宏块在比特流中的运动矢量数。

9.3.2 帧内预测模式

当前宏块的每一个4x4亮度块采用以下方法确定其预测模式：

——计算当前块预测模式的预测值

- 1) 如果左边块‘存在’并且是帧内预测块，则将左边块的帧内预测模式赋值给 intraPredModeA；否则 intraPredModeA 等于-1。
- 2) 如果上边块‘存在’并且是帧内预测块，则将上边块的帧内预测模式赋值给 intraPredModeB；否则 intraPredModeB 等于-1。
- 3) predIntraPredMode 根据 intraPredModeA 和 intraPredModeB 由表 9 得到。

表 9 predIntraPredMode 与 intraPredModeA、intraPredModeB 的关系

<div>intraPredModeB intraPredModeA</div>	-1	0	1	2	3	4	5	6	7	8
-1	8	8	8	8	8	8	8	8	8	8
0	8	0	0	2	0	0	0	2	0	2
1	8	2	1	2	2	2	2	2	2	2
2	8	2	2	2	2	2	2	2	2	2
3	8	2	1	2	3	4	5	2	7	2
4	8	4	4	2	4	4	4	6	4	4
5	8	5	5	2	5	5	5	6	5	5
6	8	6	6	6	6	6	6	6	6	6
7	8	7	7	2	7	7	7	6	7	7
8	8	0	1	2	3	4	5	6	7	8

对于普通帧内预测模式宏块：

- 如果 pred\_mode\_flag 的值为 ‘0’，则预测模式 IntraLumaPredMode 等于 predIntraPredMode。
- 如果 pred\_mode\_flag 的值为 ‘1’，并且 intra\_luma\_pred\_mode 小于 predIntraPredMode，则 IntraLumaPredMode 等于 intra\_luma\_pred\_mode；否则 IntraLumaPredMode 等于 intra\_luma\_pred\_mode 加 1。

对于直接帧内预测模式宏块：

- 预测模式 IntraLumaPredMode 等于 predIntraPredMode。

左边块或上边块 ‘存在’ 指该块应在图像内，并且该块应与当前块属于同一条带。

IntraLumaPredMode 的值与亮度块预测模式的关系见表10。

表 10 4×4 亮度块帧内预测模式

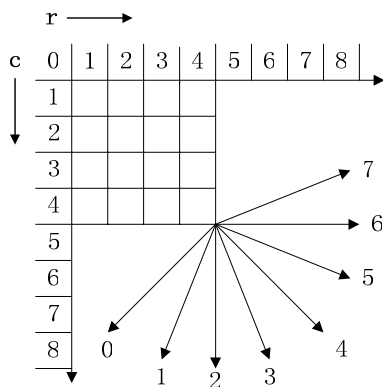
IntraLumaPredMode	名称
0	Intra_Luma_Down_Left
1	Intra_Luma_Vertical_Left
2	Intra_Luma_Vertical
3	Intra_Luma_Vertical_Right
4	Intra_Luma_Down_Right
5	Intra_Luma_Horizontal_Down
6	Intra_Luma_Horizontal
7	Intra_Luma_Horizontal_Up
8	Intra_Luma_DC

表10所示的4×4亮度块帧内预测模式见图6 a)。

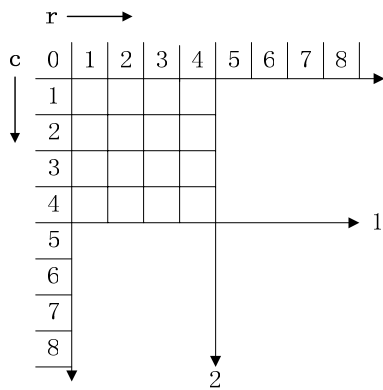
当前宏块的每一个8×8色度块中所有4×4色度块的预测模式都相同，其预测模式 IntraChromaPredMode 等于 **intra\_chroma\_pred\_mode**。IntraChromaPredMode 的值与色度块预测模式的关系见表11。表11所示的4×4色度块帧内预测模式参见图6 b)。

表 11 4×4 色度块帧内预测模式

IntraChromaPredMode	名称
0	Intra_Chroma_DC
1	Intra_Chroma_Horizontal
2	Intra_Chroma_Vertical



a) 4×4亮度块帧内预测模式



b) 4×4色度块帧内预测模式

图6 帧内预测模式

9.3.3 参考图像选择

每帧图像使用的参考图像数不应超过Max(num\_ref\_frames, 1)个, 它们应是nal\_ref\_idc不等于0的帧。

参考图像由参考帧号RefPicNum标明, IDR图像的RefPicNum值为0。对非IDR参考图像, 其帧标号为frame\_num, 令PrevFrameNum等于前一参考图像的帧标号, PrevRefPicNum等于前一参考图像的RefPicNum值, 则该参考图像的RefPicNum值为

```
if ( PrevFrameNum <= frame_num )
    RefPicNum = PrevRefPicNum + frame_num - PrevFrameNum
else
    RefPicNum = PrevRefPicNum + frame_num - PrevFrameNum + 32
```

RefPicNum的取值范围是0..(2<sup>32</sup>-1)。

参考索引值用来确定对当前图像进行解码处理所用的参考图像, 参考索引的取值范围是0..(Max(num\_ref\_frames, 1)-1)。参考索引值随着参考图像RefPicNum值的增加而减小。

解码当前帧后, 如果nal\_ref\_idc不等于0, 当前帧被标为‘用作参考’。如果当前帧为IDR图像, 则解码图像缓冲区中除当前帧外所有的参考图像都被标为‘不作参考’; 否则, 如果nal\_ref\_idc不等于0并且除当前帧外标为‘用作参考’的参考图像总数等于num\_ref\_frames, 进行以下操作: 如果num\_ref\_frames为1, 则非当前帧的标为‘用作参考’的参考图像被标为“不作参考”; 否则, 如果num\_ref\_frames为2并且SSW也为2, 解码图像缓冲区中除当前帧以外RefPicNum值较小的标为‘用作参考’

的参考图像被标为“不作参考”；否则，如果num\_ref\_frames为2并且SSW为1，除当前帧以外RefPicNum值较大的标为‘用作参考’的参考图像被标为“不作参考”。

参考索引值用来确定对当前图像进行解码处理所用的参考图像，参考索引的取值范围是0..(Max(num\_ref\_frames, 1)-1)。参考索引值随着标为‘用作参考’的参考图像RefPicNum值的减小而增加，增加量为1，标为‘用作参考’的参考图像中RefPicNum值最大的参考图像的参考索引值为0。

参考图像的参考索引值的标记方法见图7，此时num\_ref\_frames等于2。图7中的数字表示参考索引值，箭头所指的是参考图像。

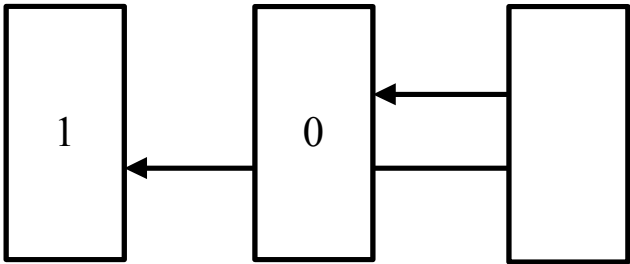


图7 参考索引标记方法

如果当前图像是P帧，num\_ref\_frames等于2，picture\_reference\_flag等于‘0’，IsRef0anIdrPic等于0，则参考索引值等于语法元素mb\_reference\_index的值。否则参考索引不出现在比特流中，缺省参考图像是图7中标记为0的图像。

如果图7中标记为0的‘用作参考’的图像是IDR图像，则IsRef0anIdrPic等于1；否则IsRef0anIdrPic等于0。

9.3.4 运动矢量

一个亮度块E和它的相邻4×4亮度块A、B、C和D之间的空间位置如图8所示。E的大小可以是16×16、16×8、8×16、8×8、8×4、4×8或者4×4。A是与E的左下角样本紧邻的块，B、D是与E的左上角样本紧邻的块，C是与E的右上角样本紧邻的块。

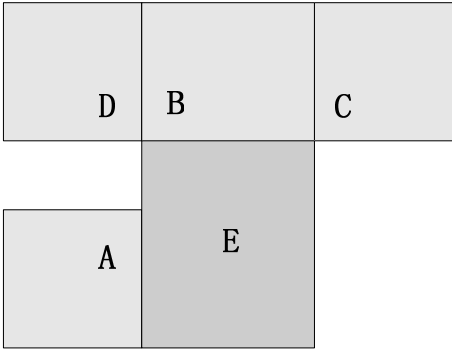


图8 亮度块E和相邻亮度块的空间位置关系

9.3.4.1 亮度运动矢量预测

图8中块A、B、C、D的原始运动矢量为mvA、mvB、mvC、mvD。

- 如果 A ‘不可用’ 或者采用帧内预测模式，mvA 为零矢量，A 的参考索引值为-1。
- 如果 B ‘不可用’ 或者采用帧内预测模式，mvB 为零矢量，B 的参考索引值为-1。
- 如果 D ‘不可用’ 或者采用帧内预测模式，mvD 为零矢量，D 的参考索引值为-1。
- 如果 C ‘不可用’ 或者采用帧内预测模式，那么 mvC 等于 mvD，C 的参考索引值等于 D 的参考索引值。

块‘不可用’指该块不存在，或者尚未解码；否则该块‘可用’。相邻块‘存在’指该块应在图像内，并且该块应与当前块属于同一条带；否则相邻块‘不存在’。

当前块E的运动矢量预测值MVEPred计算过程如下：

第一步，如果A、B、C三者中只有一个块的参考索引值与当前块E的参考索引值相同，那么MVEPred等于mvX（X为A、B或C）；否则进行第二步。

第二步，如果E所在宏块按16×8或8×16模式编码，计算过程如下（见图9）：

——8×16 模式：

- 1) E 为左块：如果 A 和 E 的参考索引值相同，MVEPred 等于 mvA；否则进行第三步。
- 2) E 为右块：如果 C 和 E 的参考索引值相同，MVEPred 等于 mvC；否则进行第三步。

——16×8 模式：

- 1) E 为上块：如果 B 和 E 的参考索引值相同，MVEPred 等于 mvB；否则进行第三步。
- 2) E 为下块：如果 A 和 E 的参考索引值相同，MVEPred 等于 mvA；否则进行第三步。

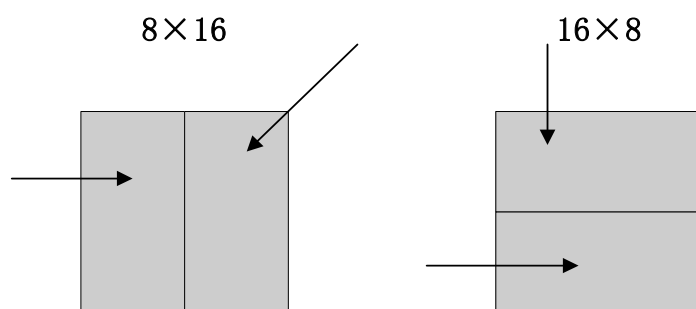


图9 8×16或16×8模式预测

第三步，定义距离 $\text{Dist}(\text{MV1}, \text{MV2}) = \text{Abs}(x1 - x2) + \text{Abs}(y1 - y2)$ ，其中运动矢量 $\text{MV1}=[x1, y1]$ ， $\text{MV2}=[x2, y2]$ 。定义VAB等于 $\text{Dist}(\text{mvA}, \text{mvB})$ ，VBC等于 $\text{Dist}(\text{mvB}, \text{mvC})$ ，VCA等于 $\text{Dist}(\text{mvC}, \text{mvA})$ 。MVEPred的计算如下：

- 计算 FMV 等于  $\text{Max}(VAB, VBC, VCA)$ ；
- 如果 FMV 和 VAB 相等，MVEPred 等于 mvC；
- 否则，如果 FMV 和 VBC 相等，MVEPred 等于 mvA；
- 否则，MVEPred 等于 mvB。

#### 9.3.4.2 亮度运动矢量解码

当前块E的运动矢量mvE等于MVEPred加上由mv\_diff\_x和mv\_diff\_y解码得到的运动矢量增量的和，其基本单位为1/4样本或1/2样本。

#### 9.3.5 宏块编码模板

如果当前宏块的cbp在比特流中，表6中CodeNum的值为cbp；否则，如果当前宏块的类型不等于P\_Skip，表6中CodeNum的值等于CBPCodeNum（见9.3.1）。由CodeNum得到6位无符号整数MbCBP（见8.2）。MbCBP表示宏块中顺序号为0到5的4个8×8亮度块和2个8×8色度块是否包含非零变换系数。MbCBP的第n位为‘0’表示顺序号为n的8×8块没有非零系数，等于‘1’表示该8×8块至少有一个非零系数（顺序号见6.5）。

#### 9.3.6 量化参数

如果当前宏块的mb\_qp\_delta在比特流中，当前宏块的量化参数CurrentQP等于PreviousQP加上mb\_qp\_delta；否则CurrentQP等于PreviousQP。CurrentQP的取值范围是0～63。picture\_qp、slice\_qp\_delta和mb\_qp\_delta的取值应保证CurrentQP的值不超过其取值范围。当前宏块解码后，PreviousQP的值等于CurrentQP的值。

### 9.4 块解码

#### 9.4.1 变长码解码



本条定义由语法元素生成量化系数值数组（Level数组）和量化系数游程数组（Run数组）的过程。Level数组包含非0量化系数的幅值，Run数组包含当前非0量化系数前的连续0的个数。生成量化系数数组所需语法元素的解析过程见8.3。

CurrentVLCTable的赋值按照以下规则进行：

——第1个解码量化系数的码表选择：

- 1) 帧内预测编码块的亮度系数，CurrentVLCTable = VLC\_Intra0，见表A.1。
- 2) 帧间预测编码块的亮度系数，CurrentVLCTable = VLC\_Inter0，见表A.8。
- 3) 色度系数，CurrentVLCTable = VLC\_Chroma0，见表A.15。
- 4) maxAbsLevel 等于0。
- 5) maxRun 等于-1。
- 6) absLevel 等于第1个解码量化系数值的绝对值。

——其他解码量化系数的码表选择：

- 1) 对于帧内预测编码块的亮度系数，如果 absLevel 大于 maxAbsLevel，或 maxAbsLevel 等于1，按如下方式进行码表切换：
  - ◆ 如果 absLevel 等于1，则：
    - a) 如果 Run 大于 maxRun，并且 CurrentVLCTable 不等于 VLC\_Intra2，那么：
      - i) 如果Run等于0、1、2或3，选择CurrentVLCTable = VLC\_Intra1，见表A.2。
      - ii) 如果Run大于3，选择CurrentVLCTable = VLC\_Intra2，见表A.3。
      - iii) maxRun等于Run。
    - b) 否则，不进行码表切换。
  - ◆ 如果 absLevel 等于2，选择 CurrentVLCTable = VLC\_Intra3，见表A.4。
  - ◆ 如果 absLevel 等于3，选择 CurrentVLCTable = VLC\_Intra4，见表A.5。
  - ◆ 如果 absLevel 等于4或5，选择 CurrentVLCTable = VLC\_Intra5，见表A.6。
  - ◆ 如果 absLevel 大于5，选择 CurrentVLCTable = VLC\_Intra6，见表A.7。
- 2) 对于帧间预测编码块的亮度系数，如果 absLevel 大于 maxAbsLevel，或者 maxAbsLevel 等于1，按如下方式进行码表切换：
  - ◆ 如果 absLevel 等于1，则
    - c) 如果 Run 大于 maxRun，并且 CurrentVLCTable 不等于 VLC\_Inter3，那么：
      - i) 如果Run等于0，1，2或3，选择CurrentVLCTable = VLC\_Inter1，见表A.9。
      - ii) 如果Run等于4，5或6，选择CurrentVLCTable = VLC\_Inter2，见表A.10。
      - iii) 如果Run大于6，选择CurrentVLCTable = VLC\_Inter3，见表A.11。
      - iv) maxRun等于Run。
    - d) 否则，不进行码表切换。
  - ◆ 如果 absLevel 等于2，选择 CurrentVLCTable = VLC\_Inter4，见表A.12。
  - ◆ 如果 absLevel 等于3或4，选择 CurrentVLCTable = VLC\_Inter5，见表A.13。
  - ◆ 如果 absLevel 大于4，选择 CurrentVLCTable = VLC\_Inter6，见表A.14。
- 3) 对于色度块的系数，如果 absLevel 大于 maxAbsLevel，按如下方式进行码表切换：
  - ◆ 如果 absLevel 等于1，选择 CurrentVLCTable = VLC\_Chroma1，见表A.16。
  - ◆ 如果 absLevel 等于2，选择 CurrentVLCTable = VLC\_Chroma2，见表A.17。
  - ◆ 如果 absLevel 大于2，选择 CurrentVLCTable = VLC\_Chroma3，见表A.18。
- 4) 如果 absLevel 大于 maxAbsLevel，maxAbsLevel 等于 absLevel。
- 5) absLevel 等于当前解码量化系数值的绝对值。

生成Run数组和Level数组的过程如下（数组地址和PreviousRunCnt初始值为0）：

——如果 trans\_coefficient\_4 小于 39：

- 1) 如果 CurrentVLCTable 中包含此 trans\_coefficient\_4 值的索引项，则以 trans\_coefficient\_4 为索引查 CurrentVLCVTable，得到量化系数和游程，把它们分别放入 Level 数组和 Run 数组；
- 2) 如果表中不包含此 trans\_coefficient\_4 值的索引项，则以 (trans\_coefficient\_4-1) 为索引查 CurrentVLCTable，得到量化系数和游程，并将量化系数符号取反后放入 Level 数组，将游程放入 Run 数组。

——如果 trans\_coefficient\_4 大于或等于 39 并且小于 71：

- 1) 如果 trans\_coefficient\_4 是奇数，将 (trans\_coefficient\_4-39)/2 放入 Run 数组，将  $-(1 + \text{LEVRUN}[\text{CurrentVLCTable}][(\text{trans\_coefficient\_4}-39)/2])$  放入 Level 数组；
- 2) 如果 trans\_coefficient\_4 是偶数，将 (trans\_coefficient\_4-39)/2 放入 Run 数组，将  $1 + \text{LEVRUN}[\text{CurrentVLCTable}][(\text{trans\_coefficient\_4}-39)/2]$  放入 Level 数组。

——如果 trans\_coefficient\_4 大于或等于 71：

- 1) 如果 trans\_coefficient\_4 是奇数，将 escape\_run 放入 Run 数组，将  $-(2 + (\text{trans\_coefficient\_4}-71)/2 + \text{LEVRUN}[\text{CurrentVLCTable}][\text{escape\_run}])$  放入 Level 数组；
- 2) 如果 trans\_coefficient\_4 是偶数，将 escape\_run 放入 Run 数组，将  $2 + (\text{trans\_coefficient\_4}-71)/2 + \text{LEVRUN}[\text{CurrentVLCTable}][\text{escape\_run}]$  放入 Level 数组。

——如果 trans\_coefficient 等于 EOB，结束块系数解码；否则数组地址加 1，PreviousRunCnt 加 Run 再加 1。如果得到的 PreviousRunCnt 等于 16，LastCoefficient 等于 1，结束块系数解码；否则 LastCoefficient 等于 0，解码下一个系数值和游程。

数组 LEVRUN[][] 的定义如下：

——对于帧内预测编码块的亮度系数，其值为：

```
LEVRUN_INTRA[7][16]=
{
    { 4, 3, 2, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0},
    { 6, 4, 3, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 6, 4, 4, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 8, 5, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {10, 5, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {12, 5, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {15, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
}
```

——对于帧间预测编码块的亮度系数，其值为：

```
LEVRUN_INTER[7][16]=
{
    { 4, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0},
    { 5, 3, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0},
    { 5, 3, 3, 2, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0},
    { 6, 4, 3, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 7, 4, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 9, 5, 3, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

```

    {13, 5, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
}
——对于色度块的系数，其值为：
LEVRUN_CHROMA[4][16]=
{
    { 6, 2, 2, 2, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0},
    { 7, 3, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0},
    { 9, 4, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {12, 6, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
}
```

9.4.2 逆扫描

由解码的Level数组和Run数组生成数组QuantCoeffArray（包含16个量化系数）的步骤如下：

——首先将 QuantCoeffArray 数组的所有元素初始化为 0；

——第二步将非 0 量化系数的值赋给 QuantCoeffArray 数组中相应的元素。定义 j 和 coeffNum，令 j 等于 9.4.1 中 Run 数组或 Level 数组中最后一个赋值元素的地址索引，coeffNum 等于-1，while (j >= 0)

```

{
    coeffNum += ( Run[j] + 1 )
    QuantCoeffArray[coeffNum] = Level[j]
    j--
}
```

——第三步通过逆块扫描将 QuantCoeffArray 数组映射为 QuantCoeffMatrix 数组。设 QuantCoeffArray 数组中某元素的地址为 k，在图 10 中找到值为 k 的单元对应的列号 i 和行号 j，然后将 QuantCoeffArray[k]赋给 QuantCoeffMatrix[i, j]。

	0	1	2	3	i
0	0	1	5	6	
1	2	4	7	12	
2	3	8	11	13	
3	9	10	14	15	
					j

图10 逆块扫描

9.5 反量化

9.5.1 确定量化参数

亮度量化参数和色度量化参数的取值范围是0~63。

如果当前块是亮度块，其量化参数QP等于CurrentQP。如果当前块是色度块，以CurrentQP为索引查表12得到色度块的QP。

表 12 色度块的 QP 与 CurrentQP 的映射关系

CurrentQP	<43	43	44	45	46	47	48	49	50	51	52
QP	CurrentQP	42	43	43	44	44	45	45	46	46	47
CurrentQP	53	54	55	56	57	58	59	60	61	62	63
QP	47	48	48	48	49	49	49	50	50	50	51

9.5.2 反量化

本条定义根据量化参数QP将二维量化系数矩阵QuantCoeffMatrix转换为二维变换系数矩阵CoeffMatrix的过程。其中，量化系数的取值范围是  $-2^{10} \sim (2^{10}-1)$ 。

二维变换系数矩阵CoeffMatrix的元素 $w_{ij}$ 由下式得到：

$$w_{ij} = (\text{QuantCoeffMatrix}[i, j] \times \text{DequantTable}(\text{QP}) + 2^{\text{ShiftTable}(\text{QP})-1}) \gg \text{ShiftTable}(\text{QP}) \quad i, j=0 \sim 4$$

QP与DequantTable和ShiftTable的关系见表13。

表 13 QP 与 DequantTable 和 ShiftTable 的关系

QP	0	1	2	3	4	5	6	7
DequantTable(QP)	32768	36061	38968	42495	46341	50535	55437	60424
ShiftTable(QP)	14	14	14	14	14	14	14	14
QP	8	9	10	11	12	13	14	15
DequantTable(QP)	32932	35734	38968	42495	46177	50535	55109	59933
ShiftTable(QP)	13	13	13	13	13	13	13	13
QP	16	17	18	19	20	21	22	23
DequantTable(QP)	65535	35734	38968	42577	46341	50617	55027	60097
ShiftTable(QP)	13	12	12	12	12	12	12	12
QP	24	25	26	27	28	29	30	31
DequantTable(QP)	32809	35734	38968	42454	46382	50576	55109	60056
ShiftTable(QP)	11	11	11	11	11	11	11	11
QP	32	33	34	35	36	37	38	39
DequantTable(QP)	65535	35734	38968	42495	46320	50515	55109	60076
ShiftTable(QP)	11	10	10	10	10	10	10	10
QP	40	41	42	43	44	45	46	47
DequantTable(QP)	65535	35744	38968	42495	46341	50535	55099	60087
ShiftTable(QP)	10	9	9	9	9	9	9	9
QP	48	49	50	51	52	53	54	55
DequantTable(QP)	65535	35734	38973	42500	46341	50535	55109	60097
ShiftTable(QP)	9	8	8	8	8	8	8	8
QP	56	57	58	59	60	61	62	63
DequantTable(QP)	32771	35734	38965	42497	46341	50535	55109	60099
ShiftTable(QP)	7	7	7	7	7	7	7	7

从符合本部分的比特流中解码得到的变换系数的取值范围应为  $-2^{11} \sim (2^{11}-1)$ 。

9.6 反变换

本条定义将4×4变换系数矩阵CoeffMatrix转换为4×4残差样值矩阵ResidueMatrix的过程，步骤如下：

——首先，对变换系数矩阵进行如下水平反变换：

$$H = \text{CoeffMatrix} \times T_4^T$$

其中， $T_4$ 是4×4反变换矩阵， $T_4^T$ 是 $T_4$ 的转置矩阵， $H$ 表示水平反变换后的中间结果。

$$T_4 = \begin{bmatrix} 2 & 3 & 2 & 1 \\ 2 & 1 & -2 & -3 \\ 2 & -1 & -2 & 3 \\ 2 & -3 & 2 & -1 \end{bmatrix}$$

——第二步，对矩阵 $H$ 进行如下垂直反变换：

$$H = T_4 \times H$$

其中， $H$ 表示反变换后的4×4矩阵。从符合本部分的比特流中解码得到的 $H$ 矩阵元素取值范围应为-215~(215-17)。

——第三步，残差样值矩阵ResidueMatrix的元素 $r_{ij}$ 计算如下：

$$r_{ij} = [h_{ij} + 2^4] \gg 5 \quad i, j = 0 \sim 4$$

其中 $h_{ij}$ 是 $H$ 矩阵的元素。

## 9.7 帧内预测

当前帧内预测块由其上边和左边的参考样本 $r[i]$  ( $i=0 \sim N$ ) 和 $c[i]$  ( $i=0 \sim N$ ) 来预测 ( $r$ 、 $c$ 可表示亮度或色度参考样本。对亮度参考样本， $N=8$ ；对色度参考样本， $N=4$ )，见图6，其中 $r[0]$ 等于 $c[0]$ 。参考样本的值为环路滤波前的重建样本值。本条定义帧内预测的具体方法，预测完成后得到一个 $N \times N$ 预测样本矩阵predMatrix。

### 9.7.1 参考样本的获得

设当前块所属的图像样本矩阵为 $I$ ， $I$ 可表示亮度或色度矩阵。对于亮度矩阵，如果某个图像样本所在的块‘不存在’，或此样本尚未解码，或constrained\_intra\_pred\_flag等于‘1’并且该块为帧间编码块，则此样本‘不可用’；反之此样本‘可用’。对于色度矩阵，如果某个图像样本所在的块‘不存在’，或constrained\_intra\_pred\_flag等于‘1’并且该块为帧间编码块，则此样本‘不可用’；反之此样本‘可用’。如果色度矩阵中某个图像样本所在的块‘存在’但此样本尚未解码，则参考样本值取为其对应的预测样本值，此时 $I$ 表示色度矩阵或色度预测矩阵。

设当前块左上角样本的坐标是( $x_0$ ,  $y_0$ )，其参考样本按以下规则获得：

- 如果坐标为( $x_0+i-1$ ,  $y_0-1$ ) ( $i=1 \sim 4$ )的样本‘可用’，则 $r[i]$  ( $i=1 \sim 4$ )等于 $I[x_0+i-1, y_0-1]$ ， $r[i]$  ( $i=5 \sim 8$ )等于 $r[4]$ ， $r[i]$  ( $i=1 \sim 8$ )‘可用’；否则 $r[i]$  ( $i=1 \sim 8$ )‘不可用’；
- 如果坐标为( $x_0-1$ ,  $y_0+i-1$ ) ( $i=1 \sim 4$ )的样本‘可用’，则 $c[i]$  ( $i=1 \sim 4$ )等于 $I[x_0-1, y_0+i-1]$ ， $c[i]$  ( $i=5 \sim 8$ )等于 $c[4]$ ， $c[i]$  ( $i=1 \sim 8$ )‘可用’；否则 $c[i]$  ( $i=1 \sim 8$ )‘不可用’；
- 如果坐标为( $x_0-1$ ,  $y_0-1$ )的样本‘可用’，则 $r[0]$ 等于 $I[x_0-1, y_0-1]$ ， $r[0]$ ‘可用’；否则， $r[0]$ 等于128， $r[0]$ ‘不可用’。

### 9.7.2 亮度块帧内预测

根据IntraLumaPredMode决定亮度块帧内预测方法。

——IntraLumaPredMode 等于 0 (Intra\_Luma\_Down\_Left 预测)

当 $r[i]$ 、 $c[i]$  ( $i=1 \sim 8$ ) 均‘可用’时，该模式才被使用，此时

$$\text{PredMatrix}[x, y] = (r[x+y+1] + r[x+y+3] + c[x+y+1] + c[x+y+3] + 2 \times (r[x+y+2] + c[x+y+2]) + 4) \gg 3 \quad (x, y=0 \sim 3 \text{ 且 } x, y \text{ 不同时等于 } 3)$$

$$\text{PredMatrix}[x, y] = (r[7] + r[8] + c[7] + c[8] + 2) \gg 2 \quad (x, y \text{ 都等于 } 3)$$

——IntraLumaPredMode 等于 1 (Intra\_Luma\_Vertical\_Left 预测)

当 $r[i]$  ( $i=1 \sim 8$ ) ‘可用’时，该模式才被使用，此时

- 1) 如果  $y$  等于 0 或 2,  $\text{predMatrix}[x, y] = (r[x+y/2+1] + r[x+y/2+2] + 1) \gg 1$  ( $x=0 \sim 3$ ); 否则
- 2)  $\text{predMatrix}[x, y] = (r[x+y/2+1] + 2 \times r[x+y/2+2] + r[x+y/2+3] + 2) \gg 2$  ( $x=0 \sim 3$ )

——IntraLumaPredMode 等于 2 (Intra\_Luma\_Vertical 预测)

当  $r[i]$  ( $i=1 \sim 4$ ) ‘可用’ 时, 该模式才被使用, 此时

$\text{predMatrix}[x, y] = r[x+1]$  ( $x, y=0 \sim 3$ )

——IntraLumaPredMode 等于 3 (Intra\_Luma\_Vertical\_Right 预测)

当  $r[i]$ 、 $c[i]$  ( $i=1 \sim 8$ ) 均 ‘可用’ 时, 该模式才被使用, 此时

令  $VR = 2 \times x - y$

- 1) 如果  $VR$  等于 0, 2, 4 或 6,  $\text{predMatrix}[x, y] = (r[x-y/2] + r[x-y/2+1] + 1) \gg 1$ ; 否则
- 2) 如果  $VR$  等于 1, 3 或 5,  $\text{predMatrix}[x, y] = (r[x-y/2-1] + 2 \times r[x-y/2] + r[x-y/2+1] + 2) \gg 2$ ; 否则
- 3) 如果  $VR$  等于 -1,  $\text{predMatrix}[x, y] = (c[1] + 2 \times c[0] + r[1] + 2) \gg 2$ ; 否则
- 4)  $\text{predMatrix}[x, y] = (c[y] + 2 \times c[y-1] + c[y-2] + 2) \gg 2$

——IntraLumaPredMode 等于 4 (Intra\_Luma\_Down\_Right 预测)

当  $r[i]$ 、 $c[i]$  ( $i=1 \sim 8$ ) 均 ‘可用’ 时, 该模式才被使用, 此时

- 1) 如果  $x$  等于  $y$ ,  $\text{predMatrix}[x, y] = (c[1] + 2 \times r[0] + r[1] + 2) \gg 2$  ( $x, y=0 \sim 3$ ); 否则
- 2) 如果  $x$  大于  $y$ ,  $\text{predMatrix}[x, y] = (r[x-y+1] + 2 \times r[x-y] + r[x-y-1] + 2) \gg 2$  ( $x, y=0 \sim 3$ ); 否则
- 3) 如果  $y$  大于  $x$ ,  $\text{predMatrix}[x, y] = (c[y-x+1] + 2 \times c[y-x] + c[y-x-1] + 2) \gg 2$  ( $x, y=0 \sim 3$ )

——IntraLumaPredMode 等于 5 (Intra\_Luma\_Horizontal\_Down 预测)

当  $r[i]$ 、 $c[i]$  ( $i=1 \sim 8$ ) 均 ‘可用’ 时, 该模式才被使用, 此时

令  $HD = 2 \times y - x$

- 1) 如果  $HD$  等于 0, 2, 4 或 6,  $\text{predMatrix}[x, y] = (c[y-x/2] + c[y-x/2+1] + 1) \gg 1$ ; 否则
- 2) 如果  $HD$  等于 1, 3 或 5,  $\text{predMatrix}[x, y] = (c[y-x/2-1] + 2 \times c[y-x/2] + c[y-x/2+1] + 2) \gg 2$ ; 否则
- 3) 如果  $HD$  等于 -1,  $\text{predMatrix}[x, y] = (r[1] + 2 \times r[0] + c[1] + 2) \gg 2$ ; 否则
- 4)  $\text{predMatrix}[x, y] = (r[x] + 2 \times r[x-1] + r[x-2] + 2) \gg 2$

——IntraLumaPredMode 等于 6 (Intra\_Luma\_Horizontal 预测)

当  $c[i]$  ( $i=1 \sim 4$ ) ‘可用’ 时, 该模式才被使用, 此时

$\text{predMatrix}[x, y] = c[y+1]$  ( $x, y=0 \sim 3$ )

——IntraLumaPredMode 等于 7 (Intra\_Luma\_Horizontal\_Up 预测)

当  $c[i]$  ( $i=1 \sim 8$ ) ‘可用’ 时, 该模式才被使用, 此时

令  $HU = x + 2 \times y$

- 1) 如果  $HU$  等于 0, 2 或 4,  $\text{predMatrix}[x, y] = (c[y+x/2+1] + c[y+x/2+2] + 1) \gg 1$  ( $x, y=0 \sim 3$ ); 否则

- 2) 如果 HU 等于 1 或 3,  $\text{predMatrix}[x, y] = (c[y+x/2+1] + 2 \times c[y+x/2+2] + c[y+x/2+3] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 3) 如果 HU 等于 5,  $\text{predMatrix}[x, y] = (c[3] + 3 \times c[4] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 4)  $\text{predMatrix}[x, y] = c[4]$ 。

——IntraLumaPredMode 等于 8 (Intra\_Luma\_DC 预测)

- 1) 如果  $r[i]$ 、 $c[i]$  ( $i=1 \sim 4$ ) 都‘可用’, 则  
 $\text{predMatrix}[x, y] = (r[1] + r[2] + r[3] + r[4] + c[1] + c[2] + c[3] + c[4] + 4) \gg 3$  ( $x, y = 0 \sim 3$ ) ;  
 否则
- 2) 如果  $r[i]$  ( $i=1 \sim 4$ ) ‘可用’, 则  
 $\text{predMatrix}[x, y] = (r[1] + r[2] + r[3] + r[4] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 3) 如果  $c[i]$  ( $i=1 \sim 4$ ) ‘可用’, 则  
 $\text{predMatrix}[x, y] = (c[1] + c[2] + c[3] + c[4] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 4)  $\text{predMatrix}[x, y] = 128$  ( $x, y = 0 \sim 3$ )。

### 9.7.3 色度块帧内预测

根据 IntraChromaPredMode 决定色度块帧内预测方法。

——IntraChromaPredMode 等于 0 (Intra\_Chroma\_DC 预测)

- 1) 如果  $r[i]$ 、 $c[i]$  ( $i=1 \sim 4$ ) 都‘可用’, 则  
 $\text{predMatrix}[x, y] = (r[1] + r[2] + r[3] + r[4] + c[1] + c[2] + c[3] + c[4] + 4) \gg 3$  ( $x, y = 0 \sim 3$ ) ;  
 否则
- 2) 如果  $r[i]$  ( $i=1 \sim 4$ ) ‘可用’, 则  
 $\text{predMatrix}[x, y] = (r[1] + r[2] + r[3] + r[4] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 3) 如果  $c[i]$  ( $i=1 \sim 4$ ) ‘可用’, 则  
 $\text{predMatrix}[x, y] = (c[1] + c[2] + c[3] + c[4] + 2) \gg 2$  ( $x, y = 0 \sim 3$ ) ; 否则
- 4)  $\text{predMatrix}[x, y] = 128$  ( $x, y = 0 \sim 3$ )。

——IntraChromaPredMode 等于 1 (Intra\_Chroma\_Horizontal 预测)

当  $c[i]$  ( $i=1 \sim 4$ ) ‘可用’时, 该模式才被使用, 此时

$\text{predMatrix}[x, y] = c[y+1]$  ( $x, y = 0 \sim 3$ )。

——IntraChromaPredMode 等于 2 (Intra\_Chroma\_Vertical 预测)

当  $r[i]$  ( $i=1 \sim 4$ ) ‘可用’时, 该模式才被使用, 此时

$\text{predMatrix}[x, y] = r[x+1]$  ( $x, y = 0 \sim 3$ )。

## 9.8 帧间预测

### 9.8.1 亮度运动矢量导出

如果当前宏块类型或者当前子块类型是跳过模式, 其运动矢量按照下面定义的方法导出, 否则将根据 9.3.4 解码得到的运动矢量, 按照宏块划分顺序 (在图 5 中定义) 分配给相应的子块。

——如果当前宏块类型为 ‘P\_Skip’ :

- 1) 当前块的前向参考图像为缺省参考图像, 即图 7 中标记为 0 的图像;
- 2) 如果当前宏块的上面宏块 B 或左面的宏块 A ‘不可用’,  $mvE$  等于零矢量;
- 3) 如果  $mvA$  为零矢量并且宏块 A 的参考索引值为 0, 或  $mvB$  为零矢量并且宏块 B 的参考索引值为 0, 则  $mvE$  等于零矢量;
- 4) 对于其它情况,  $mvE = MVEPred$ 。

### 9.8.2 参考样本的导出过程

首先对亮度进行1/2样本和1/4样本的插值，然后根据运动矢量得到相应的参考样本。

如果在插值过程中所参考的整数样本在参考图像外，应用该图像内距离参考样本最近的整数样本（边缘或角样本）代替，即运动矢量能指向参考图像外的样本。

#### 9.8.2.1 亮度样本插值过程

图11给出了参考图像整数样本、1/2样本和1/4样本的位置，其中用大写字母标记的为整数样本位置，用小写字母标记的为1/2和1/4样本位置。

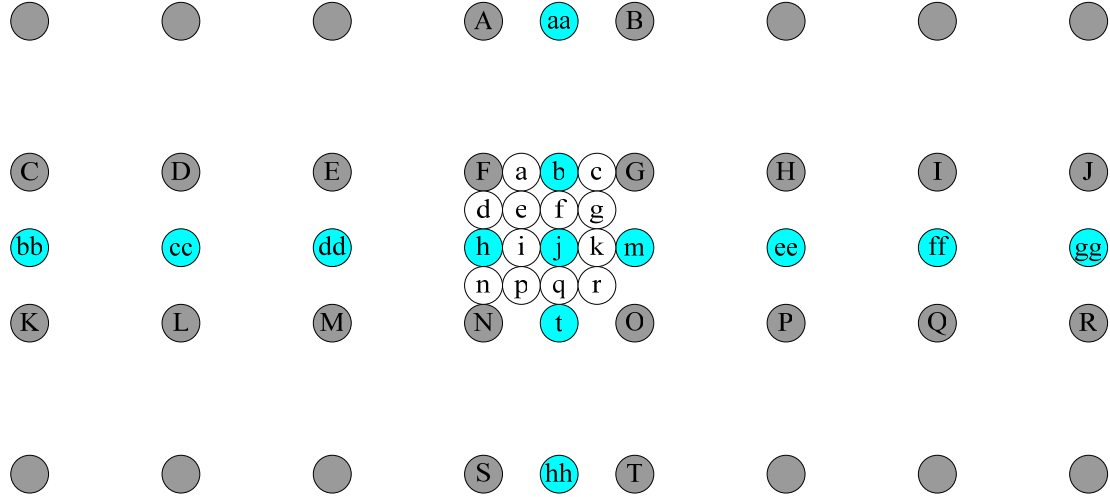


图11 整数样本、1/2样本和1/4样本的位置

1/2样本位置的预测值通过8抽头滤波器 $F_1(-1, 4, -12, 41, 41, -12, 4, -1)$ 和4抽头滤波器 $F_2(-1, 5, 5, -1)$ 计算得到。

1/2样本的计算过程如下：

- 1/2样本 $b$ ：首先用 $F_1$ 对水平方向上最近的8个整数样本滤波，得到中间值 $b' = (-C + 4D - 12E + 41F + 41G - 12H + 4I - J)$ ；最终的预测值 $b = \text{Clip1}((b' + 32) \gg 6)$ 。
- 1/2样本 $h$ ：首先用 $F_2$ 对垂直方向上最近的4个整数样本滤波，得到中间值 $h' = (-A + 5F + 5N - S)$ ；最终的预测值 $h = \text{Clip1}((h' + 4) \gg 3)$ 。
- 1/2样本 $j$ ：首先用 $F_2$ 在垂直方向上对最近的4个1/2样本最终预测值滤波，得到中间值 $j' = (-aa + 5b + 5t - hh)$ ，其中 $aa = \text{Clip1}((aa' + 32) \gg 6)$ ， $t = \text{Clip1}((t' + 32) \gg 6)$ ， $hh = \text{Clip1}((hh' + 32) \gg 6)$ 。标记为 $aa'$ ， $t'$ 和 $hh'$ 的1/2样本中间值可用 $F_1$ 对水平方向滤波得到（与 $b$ 的计算过程相同）。最终的预测值 $j = \text{Clip1}((j' + 4) \gg 3)$ 。

1/4样本位置的预测值通过线性插值计算得到。1/4样本的计算过程如下：

- 1/4样本 $a$ ： $a = (F + b + 1) \gg 1$ ，其中 $b$ 是相应位置1/2样本的最终预测值， $F$ 是整数样本值。
- 1/4样本 $c$ ： $c = (G + b + 1) \gg 1$ ，其中 $b$ 是相应位置1/2样本的最终预测值， $G$ 是整数样本值。
- 1/4样本 $d$ ： $d = (F + h + 1) \gg 1$ ，其中 $h$ 是相应位置1/2样本的最终预测值， $F$ 是整数样本值。
- 1/4样本 $n$ ： $n = (N + h + 1) \gg 1$ ，其中 $h$ 是相应位置1/2样本的最终预测值， $N$ 是整数样本值。
- 1/4样本 $i$ ： $i = (h + j + 1) \gg 1$ ，其中 $h$ 和 $j$ 是相应位置1/2样本的最终预测值。
- 1/4样本 $k$ ： $k = (m + j + 1) \gg 1$ ，其中 $m$ 和 $j$ 是相应位置1/2样本的最终预测值。
- 1/4样本 $f$ ： $f = (b + j + 1) \gg 1$ ，其中 $b$ 和 $j$ 是相应位置1/2样本的最终预测值。
- 1/4样本 $q$ ： $q = (t + j + 1) \gg 1$ ，其中 $t$ 和 $j$ 是相应位置1/2样本的最终预测值。
- 1/4样本 $e$ ， $g$ ， $p$ 和 $r$ ：

$$e = (F + j + 1) \gg 1$$

$$g = (G + j + 1) \gg 1$$

$$p = (N + j + 1) \gg 1$$



$r = ( 0 + j + 1 ) \gg 1$   
其中F, G, N和0是整数样本值,  $j$ 是相应位置1/2样本的最终预测值。

图8中当前亮度块E的左上角整像素位置为 $(x, y)$ , 预测样本矩阵的元素 $\text{predMatrix}[x, y]$ 根据表14赋值。

表 14 预测样本矩阵元素

xFracL	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
yFracL	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predMatrix [x, y]	F	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

表14中xFracL等于mvE的水平分量 $\text{mvE\_x} \& 3$ , yFracL等于mvE的垂直分量 $\text{mvE\_y} \& 3$ 。

9.8.2.2 色度样本插值过程

色度样本插值使用与对应亮度块的运动矢量mvE (mvE的水平分量为 $\text{mvE\_x}$ , 垂直分量为 $\text{mvE\_y}$ ) 对应的运动矢量mvC。mvC的水平分量为 $\text{mvC\_x}$ , 垂直分量为 $\text{mvC\_y}$ , mvC的基本单位为1/8样本,  $\text{mvC\_x}$ 的值等于 $\text{mvE\_x}$ ,  $\text{mvC\_y}$ 的值等于 $\text{mvE\_y}$ 。色度样本插值见图12, A, B, C, D是被插值样本周围的整数样本值,  $\text{dx}$ 与 $\text{dy}$ 分别是预测样本与A的水平和垂直距离,  $\text{dx}$ 等于 $\text{mvC\_x} \& 7$ ,  $\text{dy}$ 等于 $\text{mvC\_y} \& 7$ 。

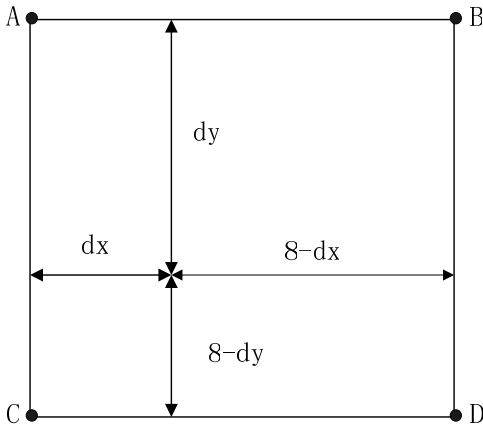


图12 1/8色度插值

预测样本矩阵的元素 $\text{predMatrix}[x, y]$ 根据下式计算：  
 $\text{predMatrix}[x, y] = \text{Clip1}(((8-\text{dx}) \times (8-\text{dy}) \times \text{A} + \text{dx} \times (8-\text{dy}) \times \text{B} + (8-\text{dx}) \times \text{dy} \times \text{C} + \text{dx} \times \text{dy} \times \text{D} + 32) \gg 6)$

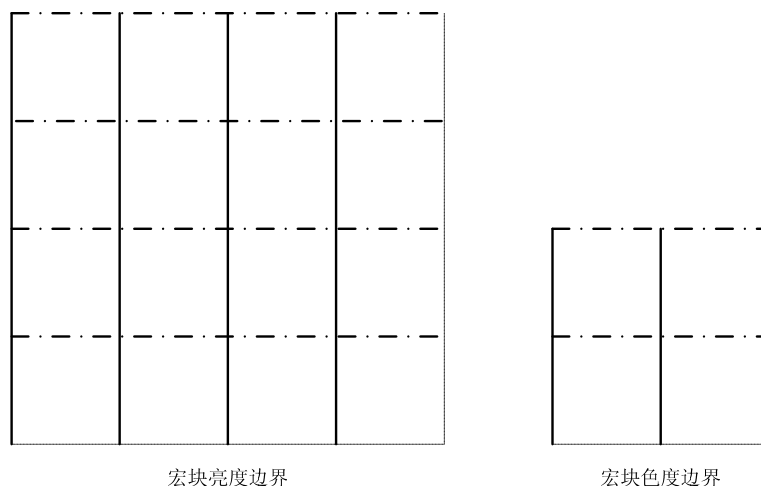
9.9 重建

重建系数矩阵RecMatrix计算如下：  
 $\text{RecMatrix}[x, y] = \text{Clip1}(\text{predMatrix}[x, y] + \text{ResidueMatrix}[x, y])$   
其中predMatrix是参考图像的预测样本矩阵。

9.10 环路滤波

除图像边界及 $\text{disable\_loop\_filter\_slice\_flag}$ 值为‘1’的条带的边界之外, 宏块的所有块边界都应进行滤波。此处宏块边界定义为宏块内部各个4x4块的边界, 以及当前宏块与相邻宏块的上边界和左边界。  
环路滤波以宏块为单位, 图像中每个宏块的滤波过程如下：

对亮度和色度分别做环路滤波，见图13，首先从左到右对垂直边界滤波，然后从上到下对水平边界滤波。当前宏块的环路滤波的输入为图像未进行滤波的样本值，当前宏块环路滤波会修改这些样本值。当前宏块垂直边界滤波过程中修改的样本值作为水平边界滤波过程的输入。



注：粗实线为垂直边界，点划线为水平边界。

图13 宏块中需要滤波的边界示意图

帧内预测使用环路滤波前的重建图像样本值。

#### 9.10.1 滤波模式的选择

根据宏块类型和宏块中亮度块的CurrentQP，按以下方法选择滤波模式：

首先，如果当前宏块是帧内编码宏块，则使用帧内宏块滤波模式。

其次，如果当前宏块不是跳过模式的帧间编码宏块，或者当前宏块的CurrentQP大于等于ThresholdQP，则选择帧间宏块滤波模式。

如果上述两个条件都不满足，则不对当前宏块滤波。

#### 9.10.2 块边界阈值的推导过程

图14表示块p和块q在水平或垂直边界两侧的4个样本点（边界用黑色粗线表示）。用P<sub>0</sub>、P<sub>1</sub>、Q<sub>0</sub>和Q<sub>1</sub>分别表示p<sub>0</sub>、p<sub>1</sub>、q<sub>0</sub>和q<sub>1</sub>滤波后的样本值。

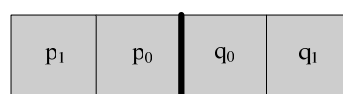


图14 4×4块水平或垂直边界样本

如果该宏块需要滤波，并且下式为真，则对边界样本滤波：

$$\text{Abs}(p_0 - q_0) < \alpha$$

其中 $\alpha$ 为块边界阈值，可以根据两个块的CurrentQP的平均值 $\text{CurrentQP}_{av}$ ，以及AlphaCIOffset计算查表索引IndexA。

CurrentQP<sub>av</sub>为：

$$\text{CurrentQP}_{av} = (\text{CurrentQP}_p + \text{CurrentQP}_q + 1) \gg 1$$

索引IndexA为：

$$\text{IndexA} = \text{Clip3}(0, 63, \text{CurrentQP}_{av} + \text{AlphaCIOffset})$$

根据索引IndexA与阈值 $\alpha$ 间的对应关系，由表15得到 $\alpha$ 值。

表 15 块边界阈值 $\alpha$ 与 IndexA 的关系

索引	$\alpha$	索引	$\alpha$	索引	$\alpha$	索引	$\alpha$
0	0	16	4	32	22	48	46
1	0	17	4	33	24	49	48
2	0	18	5	34	26	50	50
3	0	19	5	35	28	51	52
4	0	20	6	36	30	52	53
5	0	21	7	37	33	53	54
6	1	22	8	38	33	54	55
7	1	23	9	39	35	55	56
8	1	24	10	40	35	56	57
9	1	25	11	41	36	57	58
10	1	26	12	42	37	58	59
11	2	27	13	43	37	59	60
12	2	28	15	44	39	60	61
13	2	29	16	45	39	61	62
14	3	30	18	46	42	62	63
15	3	31	20	47	44	63	64

9.10.3 帧内宏块滤波模式的滤波过程

对亮度块或色度块边界两边的样本 $p_0$ 、 $p_1$ 、 $q_0$ 和 $q_1$ 进行滤波，滤波过程如下：

$delta = Clip3(-CI, CI, ((q_0 - p_0) \times 4 + (p_1 - q_1) + 4) \gg 3)$

$delta1 = delta \gg 1$

if ( $p_1 == p_0$ )

{

$P_0 = Clip1(p_0 + delta)$

$P_1 = Clip1(p_1 + delta1)$

}

else

$P_0 = Clip1(p_0 + delta1)$

if ( $q_1 == q_0$ )

{

$Q_0 = Clip1(q_0 - delta)$

$Q_1 = Clip1(q_1 - delta1)$

}

else

$Q_0 = Clip1(q_0 - delta1)$

上述滤波过程中， $CI$ 称为帧内宏块滤波裁剪参数。 $CI$ 与 $IndexCI$ 之间的关系见表16， $IndexCI$ 的定义如下：

$IndexCI = Clip3(0, 63, CurrentQP_{av})$

表 16 滤波裁减参数 CI 与 IndexCI 的关系

索引	CI	索引	CI	索引	CI	索引	CI
0	0	16	0	32	2	48	6
1	0	17	0	33	2	49	7
2	0	18	0	34	2	50	7
3	0	19	0	35	3	51	8
4	0	20	0	36	4	52	9
5	0	21	0	37	4	53	9
6	0	22	1	38	4	54	10
7	0	23	1	39	4	55	11
8	0	24	1	40	5	56	12
9	0	25	1	41	5	57	13
10	0	26	1	42	5	58	14
11	0	27	1	43	5	59	15
12	0	28	1	44	5	60	16
13	0	29	2	45	6	61	16
14	0	30	2	46	6	62	16
15	0	31	2	47	6	63	16

9.10.4 帧间宏块滤波模式的滤波过程

对亮度块或色度块边界两边的样本 $p_0$ 、 $p_1$ 、 $q_0$ 和 $q_1$ 进行滤波，滤波过程如下：

$$\mathit{delta} = \mathit{Clip3}(-\mathit{CP}, \mathit{CP}, ((q_0 - p_0) \times 4 + (p_1 - q_1) + 4) \gg 3)$$

$$\mathit{delta1} = \mathit{delta} \gg 1$$

$$\text{if } ((p_1 == p_0) \ \&\& \ (\mathit{Abs}(p_0 - q_0) < ((\alpha \gg 2) + 2)))$$

{

$$P_0 = \mathit{Clip1}(p_0 + \mathit{delta})$$

$$P_1 = \mathit{Clip1}(p_1 + \mathit{delta1})$$

}

else

$$P_0 = \mathit{Clip1}(p_0 + \mathit{delta1})$$

$$\text{if } ((q_1 == q_0) \ \&\& \ (\mathit{Abs}(p_0 - q_0) < ((\alpha \gg 2) + 2)))$$

{

$$Q_0 = \mathit{Clip1}(q_0 - \mathit{delta})$$

$$Q_1 = \mathit{Clip1}(q_1 - \mathit{delta1})$$

}

else

$$Q_0 = \mathit{Clip1}(q_0 - \mathit{delta1})$$

上述滤波过程中， $CP$ 称为帧间宏块滤波裁剪参数， $CP$ 与 $CI$ 的关系如下：

$$CP = \mathit{Max}(0, (CI \gg 1) + \mathit{CPoffset})$$

附 录 A  
(规范性附录)  
变长码表

本附录给出了变长码表。表A.1~A.18中Run表示量化系数游程，Level表示量化系数值，EOB栏及Level栏的各列数据表示语法元素trans\_coefficient\_4的值。解码时根据trans\_coefficient\_4可以查表得到Level和Run。表中EOB栏对应的数字表示代表EOB的trans\_coefficient\_4的值。

表 A.1 VLC\_Intra0（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	Level > 0				Run	Level > 0			
	1	2	3	4		1	2	3	4
0	0	8	16	30	5	10	34	-	-
1	2	18	36	-	6	14	-	-	-
2	4	20	-	-	7	22	-	-	-
3	6	26	-	-	8	24	-	-	-
4	12	32	-	-	9	28	-	-	-

表 A.2 VLC\_Intra1（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0					
		1	2	3	4	5	6
	0	-	-	-	-	-	-
0	-	1	5	9	15	27	31
1	-	3	13	23	35	-	-
2	-	7	19	33	-	-	-
3	-	11	25	-	-	-	-
4	-	17	37	-	-	-	-
5	-	21	-	-	-	-	-
6	-	29	-	-	-	-	-

表 A.3 VLC\_Intra2（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0					
		1	2	3	4	5	6
	0	-	-	-	-	-	-
0	-	1	3	9	15	23	33
1	-	5	11	19	31	-	-
2	-	7	17	27	35	-	-
3	-	13	25	-	-	-	-
4	-	21	37	-	-	-	-
5	-	29	-	-	-	-	-

表 A.4 VLC\_Intra3（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0							
		1	2	3	4	5	6	7	8
	0	—	—	—	—	—	—	—	—
0	—	1	3	5	9	13	23	25	31
1	—	7	11	17	27	37	—	—	—
2	—	15	19	33	—	—	—	—	—
3	—	21	35	—	—	—	—	—	—
4	—	29	—	—	—	—	—	—	—

表 A.5 VLC\_Intra4（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0									
		1	2	3	4	5	6	7	8	9	10
	0	—	—	—	—	—	—	—	—	—	—
0	—	1	3	5	7	11	15	17	23	31	35
1	—	9	13	19	27	33	—	—	—	—	—
2	—	21	25	37	—	—	—	—	—	—	—
3	—	29	—	—	—	—	—	—	—	—	—

表 A.6 VLC\_Intra5（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0											
		1	2	3	4	5	6	7	8	9	10	11	12
	0	—	—	—	—	—	—	—	—	—	—	—	—
0	—	1	3	5	7	9	11	13	19	21	25	31	35
1	—	15	17	23	29	33	—	—	—	—	—	—	—
2	—	27	37	—	—	—	—	—	—	—	—	—	—

表 A.7 VLC\_Intra6（用于解码帧内编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0	—	3	1	5	7	9	11	13	15	17	19	21	23	31	29	33
1	—	25	27	35	37	—	—	—	—	—	—	—	—	—	—	—

表 A.8 VLC\_Inter0（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	Level>0				Run	Level>0			
	1	2	3	4		1	2	3	4
0	0	10	22	36	4	8	34	—	—
1	2	20	—	—	5	12	30	—	—
2	4	24	—	—	6	14	—	—	—
3	6	26	—	—	7	16	—	—	—

表 A. 8（续） VLC\_Inter0（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	Level>0				Run	Level>0			
	1	2	3	4		1	2	3	4
8	18	-	-	-	10	28	-	-	-
9	32	-	-	-					

表 A. 9 VLC\_Inter1（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0				
		1	2	3	4	5
	0	-	-	-	-	-
0	-	1	7	15	27	37
1	-	3	17	33	-	-
2	-	5	21	-	-	-
3	-	9	29	-	-	-
4	-	11	35	-	-	-
5	-	13	-	-	-	-
6	-	19	-	-	-	-
7	-	23	-	-	-	-
8	-	25	-	-	-	-
9	-	31	-	-	-	-

表 A. 10 VLC\_Inter2（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0				
		1	2	3	4	5
	0	-	-	-	-	-
0	-	1	9	17	23	37
1	-	3	15	27	-	-
2	-	5	19	33	-	-
3	-	7	21	-	-	-
4	-	11	31	-	-	-
5	-	13	35	-	-	-
6	-	25	-	-	-	-
7	-	29	-	-	-	-

表 A. 11 VLC\_Inter3（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0					
		1	2	3	4	5	6
	0	-	-	-	-	-	-
0	-	1	7	11	21	27	35
1	-	3	13	25	37	-	-
2	-	5	17	29	-	-	-

表 A.11 (续) VLC\_Inter3 (用于解码帧间编码亮度块的游程和非零量化系数值的映射表)

Run	EOB	Level > 0					
		1	2	3	4	5	6
	—	—	—	—	—	—	—
3	—	9	19	—	—	—	—
4	—	15	31	—	—	—	—
5	—	23	33	—	—	—	—

表 A.12 VLC\_Inter4 (用于解码帧间编码亮度块的游程和非零量化系数值的映射表)

Run	EOB	Level > 0						
		1	2	3	4	5	6	7
	0	—	—	—	—	—	—	—
0	—	1	3	7	13	21	27	35
1	—	5	11	19	31	—	—	—
2	—	9	17	33	—	—	—	—
3	—	15	29	—	—	—	—	—
4	—	23	37	—	—	—	—	—
5	—	25	—	—	—	—	—	—

表 A.13 VLC\_Inter5 (用于解码帧间编码亮度块的游程和非零量化系数值的映射表)

Run	EOB	Level > 0								
		1	2	3	4	5	6	7	8	9
	0	—	—	—	—	—	—	—	—	—
0	—	1	3	5	9	11	17	21	29	33
1	—	7	13	19	25	35	—	—	—	—
2	—	15	23	31	—	—	—	—	—	—
3	—	27	—	—	—	—	—	—	—	—
4	—	37	—	—	—	—	—	—	—	—

表 A.14 VLC\_Inter6 (用于解码帧间编码亮度块的游程和非零量化系数值的映射表)

Run	EOB	Level > 0												
		1	2	3	4	5	6	7	8	9	10	11	12	13
	0	—	—	—	—	—	—	—	—	—	—	—	—	—
0	—	1	3	5	7	9	11	13	17	21	23	27	33	37
1	—	15	19	25	29	35	—	—	—	—	—	—	—	—
2	—	31	—	—	—	—	—	—	—	—	—	—	—	—

表 A.15 VLC\_Chroma0 (用于解码帧间编码亮度块的游程和非零量化系数值的映射表)

Run	Level > 0						Run	Level > 0					
	1	2	3	4	5	6		1	2	3	4	5	6
0	0	6	14	18	24	36	1	2	20	—	—	—	—



表 A. 15（续） VLC\_Chroma0（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	Level > 0						Run	Level > 0					
	1	2	3	4	5	6		1	2	3	4	5	6
2	4	28	—	—	—	—	6	16	—	—	—	—	—
3	8	26	—	—	—	—	7	22	—	—	—	—	—
4	10	—	—	—	—	—	8	30	—	—	—	—	—
5	12	34	—	—	—	—	9	32	—	—	—	—	—

表 A. 16 VLC\_Chroma1（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0						
		1	2	3	4	5	6	7
	0	—	—	—	—	—	—	—
0	—	1	7	11	19	23	29	33
1	—	3	13	25	—	—	—	—
2	—	5	21	—	—	—	—	—
3	—	9	27	—	—	—	—	—
4	—	15	—	—	—	—	—	—
5	—	17	—	—	—	—	—	—
6	—	31	—	—	—	—	—	—
7	—	35	—	—	—	—	—	—
8	—	37	—	—	—	—	—	—

表 A. 17 VLC\_Chroma2（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0								
		1	2	3	4	5	6	7	8	9
	0	—	—	—	—	—	—	—	—	—
0	—	1	3	7	9	15	19	25	29	35
1	—	5	11	17	27	—	—	—	—	—
2	—	13	21	37	—	—	—	—	—	—
3	—	23	33	—	—	—	—	—	—	—
4	—	31	—	—	—	—	—	—	—	—

表 A. 18 VLC\_Chroma3（用于解码帧间编码亮度块的游程和非零量化系数值的映射表）

Run	EOB	Level > 0											
		1	2	3	4	5	6	7	8	9	10	11	12
	0	—	—	—	—	—	—	—	—	—	—	—	—
0	—	1	3	5	7	11	13	17	23	25	29	31	33
1	—	9	15	19	27	35	37	—	—	—	—	—	—
2	—	21	—	—	—	—	—	—	—	—	—	—	—

附 录 B  
(规范性附录)  
档次和级别

档次和级别提供了一种定义GB/T XXXXX本部分的语法和语义的子集的手段。档次和级别对比特流进行了各种限制，同时也就规定了对某一特定比特流解码所需要的解码器能力。档次是GB/T XXXXX本部分规定的语法、语义及算法的子集。符合某个档次规定的解码器必须完全支持该档次定义的子集。级别是在某一档次下对语法元素和语法元素参数值的限定集合。在给定档次的情况下，不同级别往往意味着对解码器能力和存储器容量的不同要求。

本附录描述了不同档次和级别所对应的各种限制。所有未被限定的语法元素和参数可以取任何本部分所允许的值。如果一个比特流中不存在某个档次和级别所不允许的语法元素，并且其所含有的语法元素的值不超过此档次和级别所允许的范围，则认为此比特流在这个档次和级别上符合本部分。如果一个解码器能对符合某个档次和级别的所有比特流正确解码，则称此解码器在这个档次和级别上符合本部分。

profile\_idc和level\_idc定义了比特流的档次和级别。

B.1 档次

本部分定义的档次见表B.1。

表 B.1 档次

profile_idc	档次
0x00	禁止
0x10	基本档次 (Jiben profile)
其它	保留

对于一个给定的档次，不同的级别支持相同的语法子集。

B.1.1 基本档次

基本档次的比特流应满足以下条件：

- profile\_idc 的值应为 0x10。
- B.2.1 规定的级别限制。
- B.2.2.1 规定的级别限制。

符合基本档次某一级别的解码器应能正确解码所有符合同一档次的同一级别的比特流，并应能正确解码所有符合同一档次的更低级别的比特流。

B.2 级别

本部分定义的级别见表B.2。

表 B.2 级别

level_idc	级别
0	禁止
10	1.0
12	1.1

表 B.2（续） 级别

level_idc	级别
14	1.2
16	1.3
30	2.0
32	2.1
34	2.2
50	3.0
52	3.1
其它	保留

B.2.1 与档次无关的级别的限制

表B.3到表B.5给出了级别的参数限制。

表 B.3 级别的参数限制

参 数	级 别			
	1.0	1.1	1.2	1.3
每秒最大宏块个数	1,485	1,485	6,000	11,880
每帧最大宏块个数	99	99	396	396
宏块最大编码比特数	4,096	4,096	4,096	4,096
最大比特率（比特每秒）	64,000	128,000	384,000	768,000
最大解码图像缓冲区字节数	114,048	114,048	456,192	456,192
最大编码图像缓冲区比特数	175,000	350,000	1,000,000	2,000,000
最大水平运动矢量范围	[-2048, +2047.75]	[-2048, +2047.75]	[-2048, +2047.75]	[-2048, +2047.75]
最大垂直运动矢量范围	[-32, +31.75]	[-32, +31.75]	[-32, +31.75]	[-32, +31.75]
MaxSubMbRectSize（字节）	572	572	572	572
每帧图像最小压缩比	2	2	2	2

表 B.4 级别的参数限制

参 数	级 别		
	2.0	2.1	2.2
每秒最大宏块个数	11,880	19,800	20,250

表 B. 4（续） 级别的参数限制

参 数	级 别		
	2. 0	2. 1	2. 2
每帧最大宏块个数	396	792	1, 620
宏块最大编码比特数	4, 096	4, 096	4, 096
最大比特率（比特每秒）	2, 000, 000	4, 000, 000	4, 000, 000
最大解码图像缓冲区字节数	456, 192	912, 384	1, 555, 200
最大编码图像缓冲区比特数	2, 000, 000	4, 000, 000	4, 000, 000
最大水平运动矢量范围	[-2048, +2047. 75]	[-2048, +2047. 75]	[-2048, +2047. 75]
最大垂直运动矢量范围	[-32, +31. 75]	[-256, +255. 75]	[-256, +255. 75]
MaxSubMbRectSize（字节）	572	572	572
每帧图像最小压缩比	2	2	2

表 B. 5 级别的参数限制

参 数	级 别	
	3. 0	3. 1
每秒最大宏块个数	36, 000	40, 500
每帧最大宏块个数	1, 620	1, 620
宏块最大编码比特数	4, 096	4, 096
最大比特率（比特每秒）	6, 000, 000	8, 000, 000
最大解码图像缓冲区字节数	1, 555, 200	1, 555, 200
最大编码图像缓冲区比特数	6, 000, 000	8, 000, 000
最大水平运动矢量范围	[-2048, +2047. 75]	[-2048, +2047. 75]
最大垂直运动矢量范围	[-256, +255. 75]	[-256, +255. 75]
MaxSubMbRectSize（字节）	572	572
每帧图像最小压缩比	2	2

B. 2. 2 与档次有关的级别的限制

B. 2. 2. 1 基本档次

基本档次支持的级别包括：1.0、1.1、1.2、1.3、2.0、2.1、2.2、3.0和3.1。

B.2.3 级别与帧率的关系

对不同分辨率的图像，级别与最大帧率的关系见表B.6到表B.8。表B.6到表B.8仅供参考，并不是本部分的组成部分。

表 B.6 级别与最大帧率的关系

格式	亮度水平 样本数	亮度垂直 样本数	每帧宏块数	级 别			
				1.0	1.1	1.2	1.3
SQCIF	128	96	48	30.9	30.9	125.0	172.0
QCIF	176	144	99	15.0	15.0	60.6	120.0
QVGA	320	240	300	—	—	20.0	39.6
525 CIF	352	240	330	—	—	18.2	36.0
CIF	352	288	396	—	—	15.2	30.0

表 B.7 级别与最大帧率的关系

格式	亮度水平 样本数	亮度垂直 样本数	每帧宏块数	级 别		
				2.0	2.1	2.2
SQCIF	128	96	48	172.0	172.0	172.0
QCIF	176	144	99	120.0	172.0	172.0
QVGA	320	240	300	39.6	66.0	67.5
525 CIF	352	240	330	36.0	60.0	61.4
CIF	352	288	396	30.0	50.0	51.1
525 HHR	352	480	660	—	30.0	30.7
625 HHR	352	576	792	—	25.0	25.6
VGA	640	480	1,200	—	—	16.9
525 4SIF	704	480	1,320	—	—	15.3
525 SD	720	480	1,350	—	—	15.0
4CIF	704	576	1,584	—	—	12.8
625 SD	720	576	1,620	—	—	12.5

表 B. 8 级别与最大帧率的关系

格式	亮度水平 样本数	亮度垂直 样本数	每帧宏块数	级 别	
				3. 0	3. 1
SQCIF	128	96	48	172. 0	172. 0
QCIF	176	144	99	172. 0	172. 0
QVGA	320	240	300	120. 0	172. 0
525 CIF	352	240	330	109. 1	122. 7
CIF	352	288	396	90. 9	102. 3
525 HHR	352	480	660	54. 5	61. 4
625 HHR	352	576	792	45. 5	51. 1
VGA	640	480	1, 200	30. 0	33. 8
525 4SIF	704	480	1, 320	27. 3	30. 7
525 SD	720	480	1, 350	26. 7	30. 0
4CIF	704	576	1, 584	22. 7	25. 6
625 SD	720	576	1, 620	22. 2	25. 0

附 录 C  
(规范性附录)  
辅助增强信息

本附录规定辅助增强信息（SEI）净载的语法和语义。SEI消息在解码、显示或其它过程中起辅助作用。

C.1 SEI净载语法定义

如下：

sei_payload( payloadType, payloadSize ) {	描述符
if ( payloadType == 0 )	
user_data( payloadSize )	
else if ( payloadType == 1 )	
full_frame_freeze( payloadSize )	
else if ( payloadType == 2 )	
full_frame_freeze_release( payloadSize )	
else if ( payloadType == 3 )	
scalable_layer_profile_level( payloadSize )	
else if ( payloadType == 4 )	
hrd_buffering_parameters( payloadSize )	
else if ( payloadType == 5 )	
gradual_random_access( payloadSize )	
else	
reserved_sei_message( payloadSize )	
}	

C.1.1 用户数据SEI消息定义

如下：

user_data( payloadSize ) {	描述符
for ( i = 0; i < (payloadSize × 8) / 17; i++ ) {	
<b>user_data_payload_byte</b>	b(8)
<b>user_data_payload_byte</b>	b(8)
<b>marker_bit</b>	f(1)
}	
if ( ( ( payloadSize × 8 ) % 17 ) >= 8 )	
<b>user_data_payload_byte</b>	b(8)
rbsp_trailing_bits( )	
}	

C.1.2 全帧冻结SEI消息定义

如下：

full_frame_freeze( payloadSize ) {	描述符
rbsp_trailing_bits( )	
}	

C. 1. 3 全帧解冻SEI消息定义

如下：

full_frame_freeze_release( payloadSize ) {	描述符
rbsp_trailing_bits( )	
}	

C. 1. 4 可分级层档次和级别SEI消息定义

如下：

scalable_layer_profile_level( payloadSize ) {	描述符
<b>ref_pic_profile_idc</b>	b(8)
<b>ref_pic_level_idc</b>	b(8)
rbsp_trailing_bits( )	
}	

C. 1. 5 HRD缓冲区参数SEI消息定义

如下：



hrd_buffering_parameters( payloadSize ) {	描述符
<b>seq_parameter_set_id</b>	ue(v)
<b>cpb_dpb_cnt_minus1</b>	ue(v)
for ( HrdIdx = 0; HrdIdx <= cpb_dpb_cnt_minus1; HrdIdx++ ) {	
<b>cpb_underflow_allowable_flag</b> [HrdIdx]	u(1)
<b>bit_rate_value_minus1_lsb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
<b>bit_rate_value_minus1_msb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
<b>cpb_size_value_minus1_lsb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
<b>cpb_size_value_minus1_msb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
<b>dpb_size_value_minus1_lsb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
<b>dpb_size_value_minus1_msb</b> [HrdIdx]	ue(v)
<b>marker_bit</b>	f(1)
if ( initial_cpb_removal_delay_length > 16 ) {	
<b>initial_cpb_removal_delay_lsb</b> [HrdIdx]	u(16)
<b>marker_bit</b>	f(1)
<b>initial_cpb_removal_delay_msb</b> [HrdIdx]	u(v)
}	
else	
<b>initial_cpb_removal_delay</b> [HrdIdx]	u(v)
<b>marker_bit</b>	f(1)
if ( initial_dpb_output_delay_length > 16 ) {	
<b>initial_dpb_output_delay_lsb</b> [HrdIdx]	u(16)
<b>marker_bit</b>	f(1)
<b>initial_dpb_output_delay_msb</b> [HrdIdx]	u(v)
}	
else	
<b>initial_dpb_output_delay</b> [HrdIdx]	u(v)
<b>marker_bit</b>	f(1)
}	
rbsp_trailing_bits( )	
}	

### C. 1. 6 渐近随机访问SEI消息定义

如下：

gradual_random_access( payloadSize ) {	描述符
<b>recovery_frame_cnt</b>	ue(v)
rbsp_trailing_bits( )	
}	

### C. 1. 7 保留SEI消息定义

如下：

reserved_sei_message( payloadSize ) {	描述符
for ( i = 0; i < (payloadSize × 8) / 17; i++ ) {	
reserved_sei_message_payload_byte	b(8)
reserved_sei_message_payload_byte	b(8)
marker_bit	f(1)
}	
if ( ( ( payloadSize × 8 ) % 17 ) >= 8 )	
reserved_sei_message_payload_byte	b(8)
rbsp_trailing_bits( )	
}	

C. 2 SEI 净载语义

C. 2. 1 用户数据SEI消息

用户数据净载字节     **user\_data\_payload\_byte**

8位整数。用户数据的含义由用户自行定义。

C. 2. 2 全帧冻结SEI消息

表明前面显示的整帧保持不变，不被当前解码帧刷新，直到收到一个整帧解冻SEI消息或超时，以先发生的为准。超时的时间条件为5秒或5帧（输出顺序），以先发生的为准。如果超时发生，当前全帧冻结SEI消息失效。如果在超时的时间条件范围内收到另一个全帧冻结SEI消息，超时时间表从0重新计时。

C. 2. 3 全帧解冻SEI消息

表明恢复帧显示刷新，从当前解码帧的显示开始。任何随当前帧解码顺序之前的帧收到的全帧冻结SEI消息失效。

C. 2. 4 可分级层档次和级别SEI消息

如果存在，该SEI消息应出现在一个IDR图像前，所描述信息在出现下一个相同类型的SEI消息前一直有效。

档次标号     **ref\_pic\_profile\_idc**

8位无符号整数。表示由参考图像组成的比特流的档次。

级别标号     **ref\_pic\_level\_idc**

8位无符号整数。表示由参考图像组成的比特流的级别。

C. 2. 5 HRD缓冲区参数SEI消息

HRD缓冲参数SEI消息可以出现在任意访问单元中。如果一个编码视频序列引用的序列参数集的hrd\_parameters\_present\_flag的值为‘1’，则每一个IDR访问单元中和每一个带有逐渐解码刷新SEI消息的访问单元中都应出现一个HRD缓冲参数SEI消息。

序列参数集标号   **seq\_parameter\_set\_id**

序列参数集的标号，取值范围是0~15。解析过程见8. 2。

CPB和DPB数   **cpb\_dpb\_cnt\_minus1**

值加1指定比特流的HRD参数组的个数，取值范围是0~15。解析过程见8. 2。

允许下溢标志     **cpb\_underflow\_allowable\_flag**[HrdIdx]

值为‘1’表示CPB可下溢；值为‘0’表示CPB不可下溢。

比特率低位   **bit\_rate\_value\_minus1\_lsb**[HrdIdx]

比特率高位   **bit\_rate\_value\_minus1\_msb**[HrdIdx]

和 `bit_rate_scale` 一起指定第 `HrdIdx` 个 CPB 的最大输入比特率。  
`bit_rate_value_minus1_lsb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-2)$ ，`bit_rate_value_minus1_lsb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-2)$ 。`bitRateValueMinus1[HrdIdx]` 的值由下式确定：

$$\text{bitRateValueMinus1[HrdIdx]} = \text{bit\_rate\_value\_minus1\_lsb[HrdIdx]} + (\text{bit\_rate\_value\_minus1\_msb[HrdIdx]} \times 2^{16})$$

`bitRateValueMinus1[HrdIdx]` 的取值范围是  $0 \sim (2^{32}-2)$ 。对于任意大于 0 的 `HrdIdx`，`bitRateValueMinus1[HrdIdx]` 的值应大于 `bitRateValueMinus1[HrdIdx-1]`。比特率 `Bitrate[HrdIdx]` 的值（单位为比特每秒）由下式确定：

$$\text{BitRate[HrdIdx]} = (\text{bitRateValueMinus1[HrdIdx]} + 1) \times 2^{6 + \text{bit\_rate\_scale}}$$

**CPB大小低位** `cpb_size_value_minus1_lsb[HrdIdx]`

**CPB大小高位** `cpb_size_value_minus1_msb[HrdIdx]`

和 `cpb_size_scale` 一起指定第 `HrdIdx` 个 CPB 的大小。`cpb_size_value_minus1_lsb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-2)$ ，`cpb_size_value_minus1_msb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-1)$ 。`cpbSizeValue[HrdIdx]` 的值由下式确定：

$$\text{cpbSizeValueMinus1[HrdIdx]} = \text{cpb\_size\_value\_minus1\_lsb[HrdIdx]} + (\text{cpb\_size\_value\_minus1\_msb[HrdIdx]} \times 2^{16})$$

`cpbSizeValueMinus1[HrdIdx]` 的取值范围是  $0 \sim (2^{32}-2)$ 。对于任意大于 0 的 `HrdIdx`，`cpbSizeValueMinus1[HrdIdx]` 的值应小于等于 `cpbSizeValueMinus1[HrdIdx-1]`。CPB 大小 `CpbSize[HrdIdx]` 的值（单位为比特）由下式确定：

$$\text{CpbSize[HrdIdx]} = (\text{cpbSizeValueMinus1[HrdIdx]} + 1) \times 2^{4 + \text{cpb\_size\_scale}}$$

**DPB大小低位** `dpb_size_value_minus1_lsb[HrdIdx]`

**DPB大小高位** `dpb_size_value_minus1_msb[HrdIdx]`

和 `dpb_size_scale` 一起指定第 `HrdIdx` 个 DPB 的大小。`dpb_size_value_minus1_lsb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-2)$ ，`dpb_size_value_minus1_msb[HrdIdx]` 的取值范围是  $0 \sim (2^{16}-1)$ 。`dpbSizeValue[HrdIdx]` 的值由下式确定：

$$\text{dpbSizeValueMinus1[HrdIdx]} = \text{dpb\_size\_value\_minus1\_lsb[HrdIdx]} + (\text{dpb\_size\_value\_minus1\_msb[HrdIdx]} \times 2^{16})$$

`dpbSizeValueMinus1[HrdIdx]` 的取值范围是  $0 \sim (2^{32}-2)$ 。DPB 大小 `DpbSize[HrdIdx]` 的值（单位为字节）由下式确定：

$$\text{DpbSize[HrdIdx]} = (\text{dpbSizeValueMinus1[HrdIdx]} + 1) \times 2^{4 + \text{dpb\_size\_scale}}$$

至少应有一个 `HrdIdx` 值使 `BitRate[HrdIdx]` 小于等于 `MaxBR`（附录B中定义的最大比特率）且 `CpbSize[HrdIdx]` 小于等于 `MaxCPB`（附录B中定义的最大比编码缓冲区比特数），且 `DpbSize[HrdIdx]` 小于等于 `MaxDPB`（附录B中定义的最大解码缓冲区字节数）。

**初始CPB移出延时** `initial_cpb_removal_delay[HrdIdx]`

**初始CPB移出延时低位** `initial_cpb_removal_delay_lsb[HrdIdx]`

**初始CPB移出延时高位** `initial_cpb_removal_delay_msb[HrdIdx]`

指定第一个访问单元的第二个比特进入第 `HrdIdx` 个 CPB 到第一个访问单元的第二个比特移出第 `HrdIdx` 个 CPB 之间的延时 `InitCPBDelay[HrdIdx]`。`initial_cpb_removal_delay[HrdIdx]` 的编码长度为 `initial_cpb_removal_delay_length` 比特，`initial_cpb_removal_delay_msb[HrdIdx]` 的编码长度为 `(initial_cpb_removal_delay_length-16)` 比特。`InitCPBDealy[HrdIdx]` 的确定方法如下：

```
if (initial_cpb_removal_delay_length > 16)
    InitCpbDelay[HrdIdx] = initial_cpb_removal_delay_lsb[HrdIdx] +
        ( initial_cpb_removal_delay_msb[HrdIdx] × 216 )
else
```

$\text{InitCpbDelay}[\text{HrdIdx}] = \text{initial\_cpb\_removal\_delay}[\text{HrdIdx}]$

$\text{InitCpbDelay}[\text{HrdIdx}]$  的值不应为 0，并且不应超过  $(90000 \times (\text{CpbSize}[\text{HrdIdx}] \div \text{BitRate}[\text{HrdIdx}]))$ ，单位为 1/9000s。

初始DPB输出延时  $\text{initial\_dpb\_output\_delay}[\text{HrdIdx}]$

初始DPB输出延时低位  $\text{initial\_dpb\_output\_delay\_lsb}[\text{HrdIdx}]$

初始DPB输出延时高位  $\text{initial\_dpb\_output\_delay\_msb}[\text{HrdIdx}]$

指定第一个访问单元进入第HrdIdx个DPB到第一个访问单元从第HrdIdx个DPB输出之间的延时  $\text{InitDpbDelay}[\text{HrdIdx}]$ 。 $\text{initial\_dpb\_output\_delay}$ 的编码长度为 $\text{initial\_dpb\_output\_delay\_length}$ 比特， $\text{initial\_dpb\_output\_delay\_msb}$ 的编码长度为 $(\text{initial\_dpb\_output\_delay\_length}-16)$ 比特。

$\text{InitDpbDelay}[\text{HrdIdx}]$ 的确定方法如下：

```
if (initial_dpb_output_delay_length > 16)
    InitDpbDelay[HrdIdx] = initial_dpb_output_delay_lsb[HrdIdx] +
                          ( initial_dpb_output_delay_msb[HrdIdx] × 216 )
else
```

```
    InitDpbDelay[HrdIdx] = initial_dpb_output_delay[HrdIdx]
```

$\text{InitDpbDelay}[\text{HrdIdx}]$  的单位为 1/9000s。

注：一个图像从DPB输出并不一定表示该图像的数据从DPB中移出或清除。

### C.2.6 逐渐随机访问SEI消息

恢复帧标号  $\text{recovery\_frame\_cnt}$

指定对应含有本SEI消息的图像的恢复图像的帧标号，取值范围为0~31。解析过程见8.2。如果解码过程从带有逐渐随机访问SEI消息的编码图像开始，从帧标号 $\text{frame\_num}$ 值为当前帧标号 $\text{frame\_num}$ 值与 $\text{recovery\_frame\_cnt}$ 值之和模32的帧开始，所有帧都应能被正确解码。

### C.2.7 保留SEI消息

包含AVS用于将来后向兼容目的的保留数据。除非AVS在将来的标准中有所规定，符合本标准的编码器不应发送保留SEI消息，符合本标准的解码器收到保留SEI消息时应丢弃该消息而不影响解码过程。

保留SEI消息净载字节  $\text{reserved\_sei\_message\_payload\_byte}$

8位整数。AVS保留作将来的应用。

附 录 D  
(规范性附录)  
字节流格式

本附录定义字节流的语法、语义及字节流单元的解码过程。

D.1 字节流的语法和语义

D.1.1 字节流NAL单元语法定义

如下：

byte_stream_nal_unit( NumBytesInNALunit ) {	描述符
if ( more_data_in_byte_stream() ) {	
<b>start_code_prefix</b> /* equal to 0x000001 */	f(24)
nal_unit( NumBytesInNALunit )	
}	
}	

D.1.2 字节流NAL单元语义

起始码前缀 **start\_code\_prefix**

比特串 0x000001。

D.2 字节流的解码过程

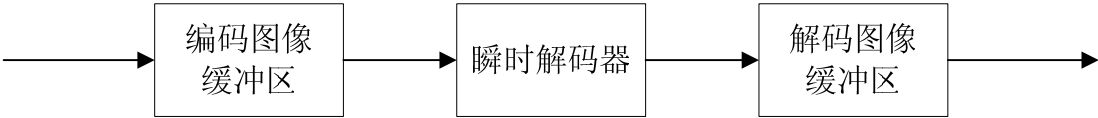
本章定义字节流的解码过程。字节流解码后得到NAL单元序列。

在解码过程的开始，解码器初始化字节流中的当前位置为字节流的开始位置。然后，解码器重复以下步骤直到字节流结束或者字节流中的最后一个NAL单元被解码。

- 如果字节流中当前位置开始接着的三个字节的序列的值为 0x000001，则这三个字节被提取出来并丢弃，同时把字节流中的当前位置置为紧跟这三个字节后的字节的位置。
- 变量 NumBytesInNALunit 等于当前位置和下一个起始码前缀（在字节对齐位置，值为 0x000001，长度为三个字节的序列）之间的字节数（包括起始码前缀前一个字节）。
- 从比特流中提取 NumBytesInNALunit 个字节，即 nal\_unit( NumBytesInNALunit )，然后交给 NAL 单元解码过程解码。同时，字节流中的当前位置向前移 NumBytesInNALunit 个字节。

附 录 E  
(规范性附录)  
虚拟参考解码器

本附录定义了虚拟参考解码器（以下简称HRD）。每一组HRD参数确定一个HRD操作模式。HRD包含一个编码图像缓冲区（CPB），一个瞬时解码器和一个解码图像缓冲区（DPB），见图E. 1。



图E. 1 虚拟参考解码器模型

对于任一个HRD操作模式，访问单元的数据按照第D. 3章定义的方式进入/移出编码图像缓冲区CPB和解码图像缓冲区DPB。符合的本部分的比特流和解码器应分别满足第E. 4章和第E. 5章规定的要求。

本附录中所有的运算都是实数运算，不存在舍入误差，例如HRD缓冲区中的比特数可不是整数。

E. 1 HRD参数语法

hrd_parameters() {	描述符
initial_cpb_removal_delay_length	u(5)
initial_dpb_output_delay_length	u(5)
bit_rate_scale	u(4)
cpb_size_scale	u(4)
dpb_size_scale	u(4)
}	

E. 2 HRD参数语义

初始CPB移出延时长度 initial\_cpb\_removal\_delay\_length

5位无符号整数。指定initial\_cpb\_removal\_delay[HrdIdx]的长度，单位为比特。如果该语法元素不存在，其值为24。initial\_cpb\_removal\_delay\_length不应为0。

初始DPB输出延时长度 initial\_dpb\_output\_delay\_length

5位无符号整数。指定initial\_dpb\_output\_delay[HrdIdx]的长度，单位为比特。如果该语法元素不存在，其值为24。initial\_dpb\_output\_delay\_length不应为0。

比特率数值范围 bit\_rate\_scale

4位无符号整数。和bit\_rate\_value\_minus1[HrdIdx]一起指定第HrdIdx个CPB的最大输入比特率。如果该语法元素不存在，其值为4。

CPB大小数值范围 cpb\_size\_scale

4位无符号整数。和cpb\_size\_value\_minus1[HrdIdx]一起指定第HrdIdx个CPB的大小。如果该语法元素不存在，其值为6。

DPB大小数值范围 dpb\_size\_scale

4位无符号整数。和dpb\_size\_value\_minus1[HrdIdx]一起指定第HrdIdx个DPB的大小。如果该语法元素不存在，其值为6。

E. 3 HRD操作

HRD可以在任一HRD缓冲参数SEI消息初始化。

- a) HRD 初始化后, CPB 和 DPB 的充满度为 0。
- b) 访问单元的比特以速率  $\text{BitRate}[\text{HrdIdx}]$  流入 CPB。
- c) 当第一个比特进入 CPB 时, 解码计时器开始计时, 初始值为  $0 - \text{InitCpbDelay}[\text{HrdIdx}]$ 。解码计时器的值小于 0 时, 没有数据从 CPB 中移出。
- d) 当解码计时器的值等于 0 时, 第一个访问单元从 CPB 中移出。其它访问单元从 CPB 中移出时, 解码计时器的值等于该访问单元的相对输出时间。  
注: 相对输出时间可以从系统层得到, 也可以从比特流中得到。
- e) 一个访问单元从 CPB 中移出后, 相应解码图像立即进入解码图像缓冲区 DPB。当第一帧解码图像进入 DPB 时, 输出计时器开始计时, 初始值为  $0 - \text{InitDpbDelay}[\text{HrdIdx}]$ 。输出计时器的值小于 0 时, 解码数据不从 DPB 中移出。
- f) 当输出计时器的值等于一个解码图像的相对输出时间时, 该解码图像立即从 DPB 中输出。如果该解码图像是非参考图像, 或者该解码图像是参考图像且被标为‘不作参考’, 该解码图像的数据同时从 DPB 中移出。从 DPB 中输出一个解码图像时, 所有被标为‘不作参考’且相对输出时间值小于输出计时器值的解码图像从 DPB 中移出。

相对输出时间 (ROT) 的推导方法如下:

——对于第一帧图像,

$$\text{ROT}(0) = 0$$

——对于其它非 IDR 图像,

$$\text{ROT}(n) = \text{ROT}(n-1) + (\text{picture\_distance}(n) - \text{picture\_distance}(n-1) + 256) \% 256 \\ \times \text{delta\_time\_picture\_distance\_1}$$

其中  $\text{ROT}(n)$  表示第  $n$  帧图像的相对输出时间,  $\text{picture\_distance}(n)$  表示第  $n$  帧图像的图像间距值。

——对于其它 IDR 图像,

$$\text{ROT}(n) = \text{ROT}(n-1) + (\text{picture\_distance\_gap\_minus1} + 1) \\ \times \text{delta\_time\_picture\_distance\_1}$$

其中  $\text{ROT}(n)$  表示该 IDR 图像的相对输出时间,  $\text{ROT}(n-1)$  表示前帧图像 (输出顺序) 的相对输出时间。

#### E.4 对符合本部分的比特流的要求

符合本部分的比特流应满足以下要求。

当采用上述 HRD 的任一操作模式对该比特流进行操作时,

——CPB 和 DPB 均不应上溢。如果  $\text{cpb\_underflow\_allowable\_flag}[\text{HrdIdx}]$  等于 ‘0’, 则 CPB 不应下溢; 否则, CPB 可下溢。

注: 在低延时模式下, 如果有编码比特数相对很大的图像存在, 可能需要允许 CPB 下溢。

——每一解码图像进入解码图像缓冲区 DPB 的时间应小于或等于其相对输出时间。

#### E.5 对符合本部分的解码器的要求

符合本部分某一档次和规格的解码器应能正确解码所有符合相应档次和规格的比特流。