

# **第 I 部分 系 统**



## I -0 系统引言

此推荐|国际标准的系统部分强调将一个或更多的音频、视频或其他的基本数据流合成单个或多个数据流，以适应于存储和传送。按照本说明中的句法和语法进行系统编码，可以在一个很宽的恢复和接收条件下进行同步译码。

系统编码可有两种方法：传送流和程序流，分别适用于不同的应用。本推荐|国际标准中定义的传送流和程序流提供了必要的和充分的编码语法，保证同步译码和显示音频、视频信息，同时确保译码缓冲区既不上溢也不下溢。根据译码和显示时使用的编码音频视频数据，以及数据本身的传送情况，信息在编码中使用不同的时间标签。两种流定义都是面向分组的多路复用流。

单个的音频和视频原始数据流的基本多路复用方法如图 I -0-1 所示，按照 ITU-T Rec. H. 262|ISO/IEC 13818-2 和 ISO/IEC 13818-3 标准对视频和音频信号进行编码压缩后的原始流被组合形成 PES 分组。当 PES 分组形成的时候，使用该分组所需的与传送流及程序流无关的信息是可被加入的。当 PES 分组加上系统级信息形成传送流或程序流时信息则是不必要加入的。本系统标准包括图中虚线右侧的处理。

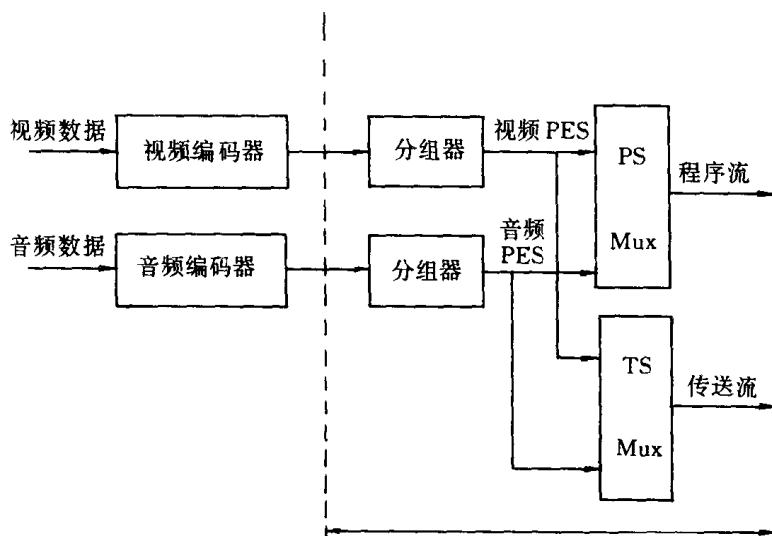


图 I -0-1 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 简化的总体框图

程序流和协议 ISO/IEC 11172 的系统层相似，它是将一个或多个具有相同时间基点的数据流的 PES 分组合为单个流。

那些原始数据流可以是一个程序的一些独立的非多路复用的数据流，也可以来自不同的程序流，每个程序流对应一个原始流，并具有相同的时间基点，在这种情况下，各个流中的 SCR 域值应保持一致。

和单程序流一样,所有的原始数据流可以被同步译码。

程序流是针对错误相对较少的环境设计的,适用于像交互式多媒体这样一些涉及软件处理系统信息的应用。程序流分组是可变的而且相对较长。

传送流将有多个独立时间基点的多道程序合成一个单独的数据流,其中属于同一道程序的各个原始数据流的 PES 分组具有相同的时间基点。传送流是针对那些很容易发生错误的环境而设计的,譬如在容易丢失或高噪音的媒体中存储和传送。传送流分组长度为 188 字节。

程序流和传送流针对不同的应用而设计,它们的定义并不严格地遵守分层模式。从一种形式转换到另一种形式是可能的、合理的,然而并不存在子集或超集的关系。尤其是,从传送流中抽取一道程序的内容并产生有效的程序流是可能的,只要通过普通的 PES 分组格式变换就行了。但并不是程序流需要的所有域值都可从传送流中直接获得,有一些必须经过推导,在多层模式中,传送流可能跨过几层范围,为的是更有效地、轻易地实现高带宽的应用。

本系统描述所规定的句法和语义规则不同于仅仅用于编码的系统层规则,它不包括压缩层视频、音频编码说明,相反,此语义规则全部用于数据流的合并。

本系统规范没有详细规定编码器或解码器的结构或实现,也没有规定多路复用器和多路解调器的结构或实现,然而比特流的性质对编码器、解码器、多路复用器和多路解调器提出了功能和性能上的要求。例如编码器必须满足最小时钟容许偏差。尽管有这样或那样的要求,它们的设计和实现仍具有相当大的自由度。

## I -0. 1 传 流

传送流是根据 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-2 和 ISO/IEC 13818-3 协议而定义的一种数据流,其目的是为了在有可能发生严重错误的环境下进行一道或多道程序的编码数据的传送和存储。这种错误表现为比特值错误或分组丢失。

传送流的速率可以是变化的或固定不变的。在任何情况下组成的原始流可以变化或固定,在这些情况下的语法和语义限制都是相同的。传送流速率由程序参考时钟(PCR)字段的位置和数值所决定,通常对于每个程序都有自己的 PCR 字段。

然而,当传送流有多道程序时,要构造一个所有的比特率都在变化的传送流是很困难的,因为对每道程序来说传送流速率是分段常数,而且当有复合程序时,那些被检测的拐点不会在同一地方。

传送流可以用很多方法产生有效的数据流:从原始的编码的数据流,从程序流,从其他本身包含一个或多个程序的传送流都可以构成含有一道或多道程序的传送流。

传送流的设计方法使它可能以最小的努力完成下面的一些操作:

- (1) 从传送流中的一道程序恢复被编码的数据,解码并显示解码结果,如图 I -0-2。
- (2) 从传送流中的一道程序抽取分组,并生成仅含此道程序的新的传送流,如图 I -0-3。
- (3) 从多个传送流中提取一道或多道程序的分组,并生成新的传送流。

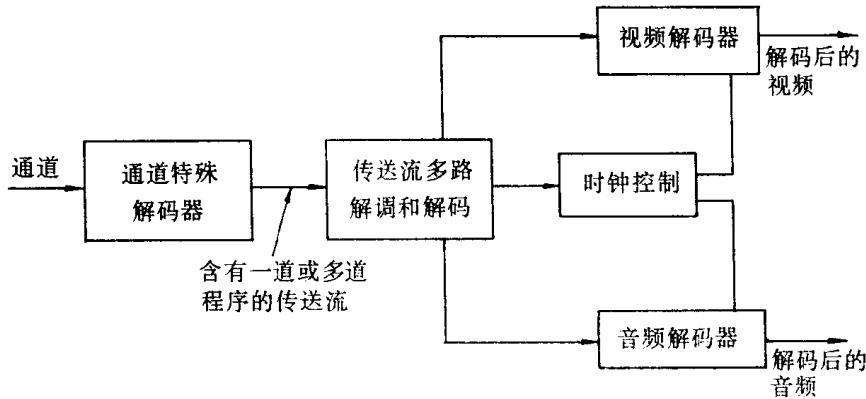


图 I -0-2 典型的传送多路解调和解码的示例

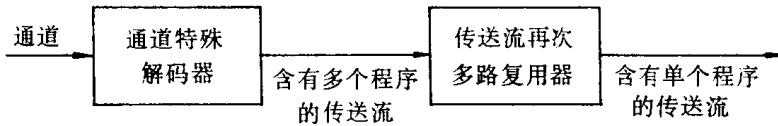


图 I -0-3 典型的传送多路复用示例

(4) 从传送流中提取一道程序生成含有此道程序的程序流,如图 I -0-4。

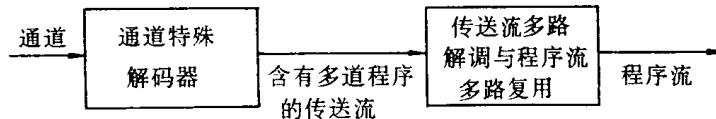


图 I -0-4 典型的传送流到程序流的转换

(5) 把一个程序流转换为一个传送流以适于通过容易出错的环境。然后,恢复成一个有效的,有时完全相同的程序流。

图 I -0-2 和图 I -0-3 表示以传送流为输入的原型的多路解调和解码系统。图 I -0-2 是第一种情况,传送流进行直接多路解调和解码。传送流由两层构成:系统层和压缩层。传送流解码器的输入流有一个包含压缩层的系统层,视频和音频解码器的输入流只含有压缩层。

接收传送流的原型解码器执行的操作或针对整个传送流(“多路复用宽操作”),或针对某个单独的原始流(“特定数据流操作”)。传送流的系统层可分为两个子层,一个相应于多路复用宽操作(传送流分组层),一个相应于特定数据流操作(PES 分组层)。

传送流的模型解码器,包括音频和视频,也在图 I -0-2 中示出,说明解码器的功能。它的结构并不是唯一的:一些系统解码器的功能,例如解码器的时钟控制,可以同样很好地分布于原始流解码器和通道特殊解码器中,但这个图对于讨论来说也是有用的。同样,由通道特殊译码器所发现的错误传送给某个音频和视频解码器的方法是多种多样的,这

种通信路径并未在图中示出。此模型解码器的设计并没有指出任何传送流解码器设计的正式要求。实际上非音频、视频的数据也是允许的,但并未表明。

图 I -0-3 表明了第二种情况,含有多道程序的传送流被转化为仅含有单道程序的传送流。这种情况下,重新多路复用操作将包括程序参考时钟(PCR)时间标志的修正,以计算时间标志到达的变化。

图 I -0-4 表明了另一种情况,一个含有多道程序的传送流首先被多路解调,接着又被转换成一个程序流。

图 I -0-3 和图 I -0-4 表明在不同类型和构成的传送流之间转换的可能性和合理性。在传送流和程序流中的特殊字段的语法定义有利于如图所示的转换,同时并不要求某个特定的多路解调器或解码器实现所有这些功能。

## I -0.2 程序流

程序流是一种数据流定义,用来传送和保存一道程序的编码数据或其它数据。它主要是针对那些不容易发生错误的环境,同时也针对系统编码的处理为主要关注对象,特别是软件处理的环境。

程序流可有固定的或变化的速率,在每一种情况下,组成的原始数据流可以有固定的或变化的速率。语法和语义限制对每种情况来说都是相同的。程序流速率由系统参考时钟(SCR)和 mux\_rate 字段的位置和数值所决定。

一个音频/视频程序流解码系统的原型如图 I -0-5 所示。它的结构不是唯一的——包括解码时钟控制在内的系统解码器功能可以通过平均分布在原始数据流解码器和通道特殊解码器来完成——但这对讨论来说仍是有用的。这个原型的解码器设计并未对程序解码器的设计提出任何正式的要求,事实上,非音频/视频数据也是允许的,但并没有画出。

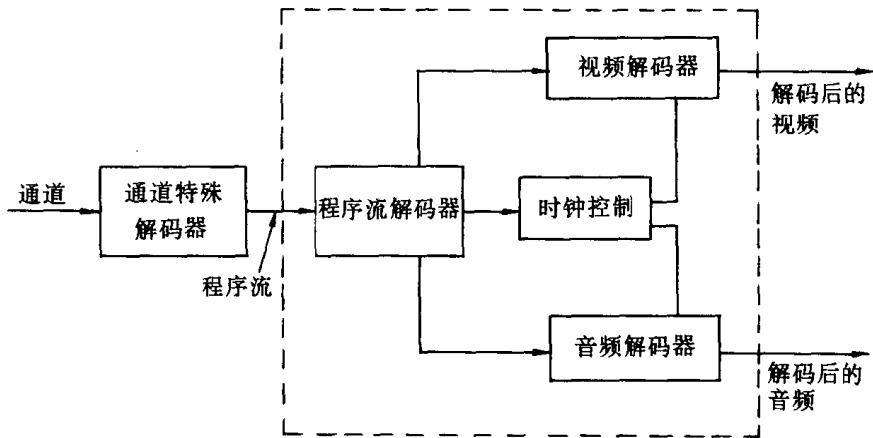


图 I -0-5 程序流原型解码器

图 I -0-5 中所示的典型程序流解码器由系统、视频和音频解码器组成,它们分别对应

于本推荐|国际标准的第 I , II , III 部分。在本解码器中,一道或多道音频、视频流的多路复用编码表示,假设以某种特殊通道格式在一些通道上被存储或被传送,这种特殊通道格式不受本推荐|国际标准的制约,也不属于原型解码器的通道特殊解码部分。

原型的解码器以程序流为输入,并依靠程序流解码器从输入流中提取时间信息。程序流解码器对输入流进行多路解调,产生作为音频、视频解码器输入的原始数据流,最后结果就是解码后的视频、音频信号。

在此图中并未画出的设计有:时间信息在程序流解码器、音频视频解码器和通道特殊解码器中的流动。音频视频解码器相互之间以及和通道之间通过这个时间信息保持同步。

程序流由两层构成:系统层和压缩层。程序流解码器的输入流有一个包含压缩层的系统层。音频、视频解码器的输入流只有一个压缩层。

原型解码器的操作可以用于整个程序流(“多路复用宽操作”),或者用于单个的原始数据流(“特定数据流操作”)。传送流系统层又可分为两个子层,一个相应于多路复用宽操作(组合层),另一个相应于特殊流操作(PES 分组层)。

### I -0. 3 传送流和程序流之间的转换

通过 PES 分组,传送流和程序流之间的转换是可能的和合理的,只要满足本推荐|国际标准在第 I -2. 4. 1 节和 I -2. 5. 1 节提出的关于传送流和程序流的规格化要求就可以了。在某些条件限制下,PES 分组可以直接从一个多路复用的比特流的有效负载映射为另一个多路复用比特流的有效负载。有可能通过识别一道程序中 PES 分组的正确顺序来支持上面的映射转换。

转换所需的其它一些信息,例如原始流之间的关系都可以从两种流的表格和头部信息获得。所有流中的这些信息在转换前后都必须有效和正确。

### I -0. 4 分组的原始数据流

传送流和程序流都是从 PES 分组按第 I -2. 4. 3. 6 节所定义的语法规则生成的。传送流和程序流的转换需使用 PES 分组,有些情况下 PES 分组执行转换时,甚至不需要改变。PES 分组的大小比传送流分组大得多。

一个原始流的具有相同流号的连续 PES 分组可以用来构造 PES 流。当 PES 分组用来构造 PES 流时,它们需包括原始流参考时钟(ESCR)字段和原始流速率(ES\_Rate)字段,同时还需满足第 I -2. 4. 3. 7 节所定义的限制。PES 流数据应该保持它们在原始流中的原始顺序。PES 流中不包含一些传送流和程序流所包含的一些必需的系统信息:例如组头、系统头、程序流映射、程序流目录、程序映射表以及传送流分组语法等信息。

PES 流是为了实现本标准而作的一种逻辑结构,然而它并不是为了进行交换和相互操作而定义的。那些要求数据流只含一个原始流的应用可以使用仅含一个原始流的传送流或程序流,它们含有足够的系统信息。那些只含有单道原始流的传送流或程序流,可以

在相同的时间基点上构造,以此运载一道完整的音频、视频程序。

## I -0.5 时间模式

系统、视频和音频都有一个时间模式,它的端到端延迟——从信号进入编码器到信号从解码器输出——是一个常数,它包括下面一些延时:编码、编码缓冲、多路复用、传送或存储、多路解调、解码缓冲、解码和显示。作为此时间模式的一部分所有视频图象和音频采样仅被显示一次,除非有意被编成反码,而且编码器和解码器具有相同的图象间隔时间或音频采样频率。系统流编码包含的时间信息可以保证实现具有恒定端到端延迟常数的系统。解码器的实现可能并不严格地遵守这种模式,但至少也要保证可以接受的性能。时序包含在本标准的规格化说明中,它必须和所有有效的比特流保持一致,而不管它们的产生方式。

所有的时序被定义为共同的系统时钟,被作为系统时序时钟。在程序流中这个时钟与视频或音频采样时钟之间有一个严格的特定比率,或者稍有区别但仍足以提供精确的端到端时序和时钟恢复。

在传送流中此系统时钟频率与视频或音频采样时钟之间任何时候都必须有严格的比率关系,这样限制是为了简化解码时采样率的恢复。

## I -0.6 条件访问

对于在程序流和传送流中的编码程序条件访问的编码和量化是由系统数据流定义来支持的。条件访问机制并没有详细规定,但流定义的设计使得实用的条件访问系统成为可能,有一些特定的句法元素提供了特别的支持。

## I -0.7 多路复用宽操作

多路复用宽操作包括从通道提取数据的协调、时钟的调整以及缓冲区的管理。这些任务是密切相关的,如果通道的数据发送率是可以控制的,那么数据发送将被调整到使解码缓冲区既不上溢也不下溢。但如果数据率是不可控制的,则原始流解码器必须根据从通道接收到的数据来调整时序以避免上溢或下溢。

程序流是由一些组构成的,它们的首部保证完成以上的任务。组首部规定了每个字节经通道到达程序流解码器的预定时间,这个目标到达时间表作为时钟修正和缓冲区管理的参考。解码器并不一定要严格遵守这个时间表,但必须补偿这种偏离。

同样,传送流是由一些传送流分组构成的,它们的首部规定了通道数据到达传送流解码器的预定时间,这个时间表提供与程序流中相同的功能。

另一个附加的多路复用宽操作是为传送流或程序流的解码建立资源,解码器利用程序流的第一分组传来的参数协助其完成这个任务,例如流的最大数据率和同时的视频通道的最高数目。传送流同样包括全局有效信息。

每个传送流和程序流都含有识别组成每一道程序的原始流的相互关系的信息,包括在实现多层视频编码时音频通道中的语音及其视频数据流之间的关系。

## I -0. 8 单独的流操作(PES 分组层)

基本的特殊流操作包括:① 多路解调;② 多原始流的同步显示。

### I -0. 8. 1 多路分配

在编码的时候,程序流由原始流多路复用而成,而传送流则由原始流、程序流或其它传送流的内容多路复用而成。原始流除包括音频、视频数据流外还有私用流、保留流、填充流。数据流被暂时分成分组包,而且保持串行顺序。一个 PES 分组包包含只有一个原始流的编码数据。

程序流的分组长度在本说明第 I -2. 5. 1 节和第 I -2. 5. 2 节中的限制下可以固定也可以变化,而传送流的分组长度为 188 字节。PES 分组长度可固定,也可变化,但在大多数应用中相对较长。

在解码的时候多路分配就是从多路复用的程序流和传送流中重建原始数据流的,此时需利用程序流分组首部的 Stream-id 字段和传送流中的分组 ID 码。

### I -0. 8. 2 同步

多原始流的同步是通过程序流和传送流中的显示时间标签(PTS)来完成的。时间标签一般以 90kHz 为单位,但系统参考时钟(SCR)、程序参考时钟(PCR)和可选的基本流参考时钟(ESCR)可以达到 27 MHz。N 个原始流解码的同步是通过把待解码的数据流调整到一个相同的主时间基点而不是调整一个解码流去匹配另一个。主时间基点可以是 N 个解码器时钟的一个时钟、数据源时钟或外部时钟。

传送流(可能有多道程序)中每一道程序都有自己的时钟基点。不同程序的时间基点可以不同。

因为 PTS 用于单个的原始流的解码,因此在传送流和程序流的 PES 分组层中都需要。随着编码器在捕获时记录时间标签,随着时间标签和编码数据传到解码器,随着解码器使用时间标签来安排显示时间表,端到端的同步得以实现。

只有一个通道的解码器的同步实现是依靠程序流中的 SCR,而在传送流中则是靠相应的 PCR。SCR 和 PCR 是将比特流本身的时间标签,它可从同一道程序的视频和音频 PTS 值所用的共同时间基点推出。由于每道程序有自己的时间基点,所以含有许多道程序的传送流的每一道程序有独立的 PCR 字段,当然,有些情况下也可以共享 PCR 字段。在 I -2. 4. 4 节的程序特殊信息(PSI)中有关于识别一道程序所带的 PCR 方法。一道程序有一个而且仅有一个 PCR 时间基点与之关联。

### I -0. 8. 3 与压缩层的关系

在某种意义上说 PES 分组层是独立于压缩层之外的,但并不绝对。考虑到 PES 分组

的有效负载不需像本推荐|国际标准的第Ⅰ、Ⅲ部分所规定的那样以压缩层开始码字开头,PES分组层是独立的。例如,视频的开始码字可出现在PES分组有效负载的任何位置,同时可被PES分组首部所分开。然而,在PES分组首部编码的时间标签是用来决定压缩层结构的显示时间(即显示单位)。

## I -0. 9 系统参考解码器

ISO/IEC 13818 的第 I 部分使用了“系统目标解码器”(STD),对传送流称作“传送流系统目标解码器”(T-STD)见 I -2. 4. 2 节,而对程序流就称作“程序流系统目标解码器”(P-STD)见第 I -2. 5. 2 节,以此提供时序和缓冲关系的形式。因为 STD 的参数是依据 ITU-T Rec. H. 222. 0 的字段(比如,缓冲区大小)而来的,每一个原始流决定了有自己的参数的 STD。编码器必须保证以正常速度向前播放时产生的比特流速度与 STD 相适应。实际的解码器可以假设数据流按它的 STD 正确工作;而必须设法补偿其设计上与 STD 有差异的作法。

## I -0. 10 应用

本文件定义的数据流有很宽的应用范围,应用开发者可以简单地选择最合适的数据流。

现代的数据通信网络可以支持 ITU-T Rec. H. 222. 0 | ISO/IEC 13818 的视频以及 ISO/IEC 13818 的音频通信,这要求一个实时传送协议。程序流可能适于这样的网络传送。

程序流也适合于 CD-ROM 上的多媒体应用。程序流的软件处理是便利的。

传送流更适用于易发生错误的环境,比如在长距离网络上或通过无线广播系统传送压缩的比特流。

许多应用要求在各种数字存储媒体(DSM)上保存和恢复 ITU-T Rec. H. 222. 0 | ISO/IEC 13818 比特流。为了便于对这种媒体的控制,本推荐|国际标准的附录 A 介绍了一种数字存储媒体命令和控制(DSM CC)协议。

## I -1 系统概述

### I -1.1 范围

ITU-T Rec. H. 222.0 | ISO/IEC 13818 的这一部分描述编码的系统层,主要是为了支持将本推荐|国际标准的第Ⅱ、Ⅲ部分中视频、音频编码算法结合起来而开发的。此系统层支持五项基本功能:①解码时多压缩流的同步;②将多个压缩流交织成单个的数据流;③解码开始时缓冲区的初始化;④连续的缓冲区管理;⑤时间识别。

ITU-T Rec. H. 222.0 | ISO/IEC 13818 的多路复用比特流可以是传送流也可以是程序流。这两种流都是由 PES 分组和一些包含其它必需信息的分组构成的。它们都支持一道程序的视频和音频压缩流在共同时间基点上的多路复用。传送流还支持多道程序的视频和音频压缩流在不同的时间基点上的多路复用。程序流通常更适合于那些错误较少的环境,特别支持程序信息的软件处理;而传送流则更适合于错误较多的环境。

一个 ITU-T Rec. H. 222.0 | ISO/IEC 13818 多路复用比特流,不论是传送流还是程序流,都由两层组成:外层为系统层,内层为压缩层。系统层为在一个系统中使用一个或多个压缩的数据流提供必要的服务,本说明的视频和音频部分定义了视频、音频数据的压缩编码层,本说明没有定义其它类型的数据编码,但系统层支持符合第 I -2.7 节中限制的数据类型。

### I -1.2 规格化参考

下面的推荐和国际标准包含通过参考本文的条款,构成了本推荐|国际标准的条款。本书出版时,那些指定的版本都是有效的。所有的推荐和国际标准都有可能被修改,在此推荐|国际标准为基础的协议上签字的团体应致力于把此标准尽可能多的应用于下面所提到的最新版的推荐和国际标准。IEC 和 ISO 的成员保存有当前有效的推荐|国际标准的登记,ITU 的电信标准局保留有最新有效的 ITU-T 推荐标准。

### I -1.3 推荐相似的国际标准

ITU-T Rec. H. 262 | ISO/IEC 13818-2: 1994 Information technology-Coding of moving pictures and associated audio-Part 2: video.

### I -1.4 其他的参考资料

[1] ISO/IEC 11172-1:1993 Information technology-Coding of moving pictures and

- associated audio for digital storage media at up to about 1.5 Mbit/s-Part 1: Systems
- [2] ISO/IEC 11172-2;1993 Information technology-Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-Part2: Video
- [3] ISO/IEC 11172-3;1993 Information technology-Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-Part3: Audio
- [4] ISO/IEC 13818-3:1994 Information technology-Coding of moving pictures and associated audio-Part3 Audio
- [5] CCIR Recommendation 601-2 Encoding parameters of digital television for studios
- [6] CCIR Report 624-4 Characteristics of systems for monochrome and colour television
- [7] CCIR Recommendation 648 Recording of audio signals
- [8] CCIR Report 955-2 Sound broadcasting by satellite for portable and mobile receivers, including Annex IV Summary description of Advanced Digital System II
- [9] CCITT Recommendation J. 17 Pre-emphasis used on Sound-Programme Circuits
- [10] IEEE Standard 1180-1990 Standard Specification for the Implementations of 8 by 8 Inverse Discrete Cosine Transform
- [11] IEC Publication 908;198, CD Digital Audio System
- [12] ISO/CD 13522; 1993 Information Technology-Coded Representation of Multi-media and Hypermedia Information (MHEG) Objects
- [13] ISO/CD 639-2, Terminology-Code for Presentation of Names of Languages- Part 2 Alpha-3 Code
- [14] ISO 8859-1, Information Processing-8 bit Single-byte Coded Graphic Character Sets-Part 1;Latin Alphabet No. 1

## I -2 技术要素

### I -2. 1 定义

本推荐|国际标准使用了下面这些定义,如果只是针对某一部分而言的,则以括号标明。

- (1) **访问单元[系统]**: 一个显示单元的编码表示。对压缩的音频数据,访问单元就是音频访问单元,而对压缩的视频数据,访问单元就是图象的编码表示(视频访问单元)。
- (2) **比特率**: 压缩的比特流从通道进入解码器的速率。
- (3) **字节对准**: 如果从流的第一个比特开始,某个比特的位置是 8 的倍数,则编码比特流中的这个比特是字节对准的。
- (4) **通道**: 一个用来存储和传送 ITU-T Rec. H. 222. 0 | ISO/IEC 13818 流的数字媒体。
- (5) **编码表示**: 以编码形式表示的数据元素。
- (6) **压缩**: 减少一个数据项表示所用的比特数。
- (7) **恒定比特率**: 从压缩比特流的开始到结束,其比特率都不变的操作。
- (8) **受限系统参数流;CSPS[系统]**: 受第 I 部分 I -2. 7. 9 节限制的传送流或程序流。
- (9) **CRC**: 用来证实数据正确性的循环冗余校验码。
- (10) **数据元素**: 一个数据项在编码前和解码后的表示。
- (11) **解码流**: 从压缩的比特流经解码重新生成的流。
- (12) **解码器**: 进行解码操作的器件。
- (13) **解码(处理)**: 本推荐|国际标准所定义的读入编码的比特流与输出解码后的图象或音频采样的处理。
- (14) **解码时间标签;DTS[系统]**: PES 分组头部的一个字段,用来指明一个访问单元在系统目标解码器被解码的时间。
- (15) **数字存储媒体;DSM**: 用于数字存储或传送的设备或系统。
- (16) **权利控制消息;ECM**: 是私用的条件访问信息,用来指明控制字以及其它可能的特殊流加密和/或控制参数。
- (17) **权利管理消息;EMM**: 是私用的条件访问信息,用来指明某个解码器的优先级和服务。它可以针对单个或一组解码器。
- (18) **编辑**: 从一个或多个压缩的比特流中生成一个新的压缩的比特流。为适于编辑,比特流必须满足本推荐|国际标准所提出的要求。
- (19) **原始流;ES[系统]**: 对编码的视频、音频或其他数据的比特流的通称。
- (20) **原始流参考时钟;ESCR[系统]**: 一个 PES 流中的时间标签,用来获得解码器的时序。

- (21) **编码器**: 进行编码处理的器件。
- (22) **编码(处理)**: 本推荐|国际标准并未详细说明的一种处理,它读入图象或音频采样的数据流以产生一种在本推荐|国际标准中定义的有效的编码比特流。
- (23) **熵编码**: 对信号的数字表示进行可变长度的无损失编码,以减少数据冗余度。
- (24) **事件**: 一个事件就是一些原始数据流的集合,它们有着共同的时间基点和相关的起始时间、结束时间。
- (25) **快放[视频]**: 以比实时快的速度和正常顺序显示一组或一组中的部分图象。
- (26) **禁止**: “禁止”这个词用在定义编码比特流时,表明某个数值不允许被使用。通常用来避免与开始码字的竞争。
- (27) **ITU-T Rec. H. 222. 0|ISO/IEC 13818(多路复用)流[系统]**: 以本推荐|国际标准第 I 部分定义的方式由零个或多个原始流所合成的一个比特流。
- (28) **层[视频和系统]**: 本推荐|国际标准第 I 、II 部分关于视频和系统说明中的数据层次中的一个级别。
- (29) **组[系统]**: 组是由组的首部以及尾随的零个或多个分组构成。它是本推荐|国际标准在 I -2. 5. 3. 3 节中有关系统编码语法中的一个层次。
- (30) **分组[系统]**: 分组由一个分组首部及尾随的来自一个原始数据流的连续字节所构成。它是本推荐|国际标准在 I -2. 4. 3. 6 中有关系统编码语法中的一个层次。
- (31) **分组数据[系统]**: 出现在分组中的来自一个原始流的连续的字节数据。
- (32) **分组标号;PID[系统]**: 一个唯一的整数值,用来联系 I -2. 4. 3 节定义的含有单道或多道程序的传送中的一道程序的原始数据流。
- (33) **填充[音频]**: 一种调整音频帧的平均长度的方法。根据相应 PCM 编码的采样要求,有条件地在音频帧中添加数据。
- (34) **有效负载**: 有效负载指分组中跟随在首部字节之后的字节。例如传送流分组的有效负载包括 PES\_packet\_header 和它的 PES\_packet\_data\_bytes,或者包括 Pointer-field 和 PSI 部分,或者包括私有数据。但一个 PES-packet-payload 仅包括 PES\_packet-data-bytes,而传送流的首部和调整字段就不是有效负载。
- (35) **PES[系统]**: 分组原始流的简写。
- (36) **PES 分组[系统]**: 用来传递原始流数据的一种数据结构。由 PES 分组首部以及其后的 PES 分组有效负载构成。在 I -2. 4. 3. 6 节中有详细描述。
- (37) **PES 流[系统]**: PES 流由 PES 分组构成,这些分组的有效负载全部来自单个原始流的数据,而且具有相同的 stream\_id。PES 流有特殊的语义限制。
- (38) **显示时间标签;PTS[系统]**: PES 分组首部中的一个字段,用来指明一个显示单元在系统目标解码器中出现的时间。
- (39) **显示单元;PU[系统]**: 解码后的音频访问单元或解码后的图象。
- (40) **程序[系统]**: 程序是一组具有相同时间基点并可能同步显示的原始流的集合。
- (41) **程序参考时钟;PCR[系统]**: 传送流中的一个时间标签,用来获得解码器时序。
- (42) **程序特殊信息;PSI[系统]**: PSI 包括 I -2. 4. 4 节中所描述的规格数据,这是对传送流进行多路解调和成功的再生成程序所必须的。某些 PSI 情况下,非必要的网络信息

表是由用户自己定义的。

(43) **随机访问**: 在任意点开始读入和解码编码比特流的操作。

(44) **保留**: “保留”这个词在定义编码的比特流时使用, 表明某个数值可能在将来 ISO 定义扩展时使用。在本推荐 | 国际标准中若非特别申明, 所有保留的位将被置‘1’。

(45) **源数据流**: 在压缩编码之前的单个非多路复用流。

(46) **开始码字[系统]**: 在编码比特流中唯一的 32 位码字, 用在编码语法中识别一些层。在表 I -2-18 中开始码字由 24 位(0x0000001)的前缀和 8 位的 stream\_id 组成。

(47) **STD 输入缓冲区[系统]**: 系统目标解码器中一个先进先出缓冲区, 用来在解码之前存储来自原始流的压缩数据。

(48) **静止画面**: 一个编码的静止画面由一个仅包含一幅编码图象的视频序列组成。此图象没有相关的 PTS 和后续图象的显示时间, 即使有, 至少也比本静止画面晚两个以上图象周期。

(49) **系统首部[系统]**: 系统首部是在本推荐 | 国际标准 I -2.5.3.5 节所定义的一个数据结构, 用来加载 ITU-T Rec. H. 222.0 | ISO/IEC 13818 多路复用流的系统特征的摘要信息。

(50) **系统参考时钟;SCR[系统]**: 程序流的时间标签, 用来获得解码器的时序。

(51) **系统目标解码器;STD[系统]**: 一种解码处理的虚拟参考模式, 用来描述 ITU-T Rec. H. 222.0 | ISO/IEC 13818 多路复用比特流的语义。

(52) **时间标签[系统]**: 一个术语, 用来指明某个特殊动作的时间, 例如字节的到达或显示单元的出现。

(53) **传送流分组首部[系统]**: 一个用来传达传送流有效负载信息的数据结构。

(54) **可变比特率**: 在压缩的比特流的解码过程中, 比特率随时间而变化的一种操作。

## I -2.2 符号与缩写

本推荐 | 国际标准中所使用的操作符与 C 语言中使用的类似。然而, 整数除法的截断与舍入有特殊的定义。位操作符是以整数的两位补码表示为基础而定义的。计数和循环通常从 0 开始。

### I -2.2.1 算术操作符

+	加法
-	减法(双目操作)或者求反(单目操作)
++	加 1
--	减 1
* 或 ×	乘法
^	乘方
/	整数除法(结果取整)。例如: $7/4$ 和 $-7/-4$ 取整结果为 1, $-7/4$ 和 $7/-4$

取整结果为-1。

// 整数除法(结果四舍五入)。例如:  $3//2=2$ ,  $-3//2=-2$

DIV 整数除法(结果向负向取整)

% 取模。仅对正数有效

$$\text{sign}() \quad \text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$$

NINT() 取最接近的整数操作符。返回与实数值最相近的整数值

sin 正弦

cos 余弦

exp 指数

$\sqrt{\quad}$  平方根

$\log_{10}$  以 10 为底的对数

$\log_e$  自然对数

## I -2. 2. 2 逻辑操作符

$\|$  逻辑或

$\&\&$  逻辑与

! 逻辑非

## I -2. 2. 3 关系操作符

$>$  大于

$\geqslant$  大于等于

$<$  小于

$\leqslant$  小于等于

$= =$  等于

$!=$  不等于

$\max[\dots]$  取参数表中的最大值

$\min[\dots]$  取参数表中的最小值

## I -2. 2. 4 位操作符

$\&$  与

$|$  或

$>>$  右移, 符号位扩展

$<<$  左移, 补零

## I -2. 2. 5 赋值

$=$  赋值操作符

## I -2. 2. 6 助记符

下述的助记符用来描述编码的比特流中所使用的不同的数据类型。

bslbf	位串,左边的为第一个。这个“左边”就是指在本推荐 国际标准中位串的书写顺序。位串就是单引号内的‘0’,‘1’串,例如‘1 000 0001’。位串中的空格只是为了便于阅读,没有其它特别含义
ch	通道
gr	在音频层Ⅱ中3*32的子带采样的颗粒,在音频层Ⅲ中18*32的子带采样的颗粒
main_data	比特流的main_data部分包含幅度因子、哈夫曼编码数据以及附加的信息
main_data_beg	为帧提供main_data在比特流中的开始位置,它等于前一帧中main_data的最后位置加1,可从前一帧的main_data_end值计算而得
part2_length	幅度因子在main_data中所占的比特数
rpchof	余项的多项式系数,高位在前
sb	子带
scfsi	幅度因子的选择信息
switch_point_l	使用的窗口开关点的比例因子带数(长块比例因子带)
switch_point_s	使用的窗口开关点的比例因子带数(短块比例因子带)
tcimsbf	双符号位补码,符号位在前
uimsbf	无符号位整位,高位在前
vlclbf	长度可变码,左边在前。所谓“左边”是指长度可变码的书写顺序
window	在block_type==2时,实际的时间空隙数,0≤window≤2 多字节字的顺序是高位在前。

## I -2. 2. 7 常数

π	3. 14159265359
e	2. 71828182845

## I -2. 3 描述比特流语法的方法

第I-2.4.1节和第I-2.5.1节描述了用解码器恢复比特流。比特流的数据项都以黑体字表示。由它的名字、比特数,数据类型和传送顺序的助记符组成。

在比特流中一个解码的数据元素引起的操作由其数值以及前面解码的数据元素所决定。在语法的语义描述时说明了数据元素的解码和解码时所用到的状态变量的定义。下面的结构用来表示数据元素以一般类型出现的条件。

注意:本语法使用C语言的惯例,当一个变量或表达式取非零值时,条件为真。

while (condition) {           如果条件为真,则数据流中出现下面一组数据元素,重

<b>data_element</b>	复,直到条件为假
...	
}	
do {	数据元素至少出现一次。数据元素重复出现直到条件为假
<b>data_element</b>	
...	
while (condition)	
if (condition) {	如果条件为真,则第一组数据元素出现在数据流中
<b>data_element</b>	
...	
}	
else {	如果条件为假,第二组数据元素出现在数据流中
<b>data_element</b>	
...	
}	
for (i=0;i<n;i++) {	这组数据元素重复出现 n 次。数据元素组的条件结构为循环控制变量 i 的值。初值为 0,以后每出现一次数据元素加 1
<b>data_element</b>	
...	
}	

注意: 数据元素中可能嵌套条件结构。约定,当只跟着一个数据元素时,{}省略。

<b>data_element[ ]</b>	data_element[ ]是一个数据数组。数据元素的个数由上下文指出
<b>data_element[n]</b>	data_element[n]是数据数组中第 n+1 个元素
<b>data_element[m][n]</b>	data_element[m][n]是二维数据数组中第 m+1, n+1 个元素
<b>data_element[l][m][n]</b>	data_element[l][m][n]是三维数据数组中第 l+1, m+1, n+1 个元素
<b>data_element[m..n]</b>	data_element 中比特 m 与比特 n 之间的比特

当语法以过程的形式表示时,不可假定图 I -2-6 或图 I -2-7 实现了一个令人满意的解码过程。尤其是,它们需要一个没有错误的输入比特流。实际上的解码器为了正确地开始译码,必须有一种方法寻找开始码字和同步字(传送流中),而且在解码时要能正确地开始并发现错误、丢失或插入。发现这些情况的方法以及之后采取的措施不在本标准之中。

### **bytealigned 函数定义**

如当前位置在一个字节的边缘,也就是在比特流中下一个位是一个字节的第一个位,函数 bytealigned()返回 1,否则,返回 0。

### **nextbits()函数定义**

nextbits()函数允许将一个比特串与待解码的比特流中紧接着的比特串进行比较。

### **next\_start\_code()函数定义**

next\_start\_code()函数跳过比特流中所填充的0位和0字节,而定位于下一个开始码字。

表 1-2-1 下一个开始码字

语 法	位数	缩写
<pre>next_start_code() {     while (! bytealigned())         zero_bit     while(nextbits() != '0000 0000 0000 0000 0000 0001')         zero_byte }</pre>	1 8	'0' '00000000'

本函数检查当前位置是否字节对准,如果不是对准的,说明有填充的‘0’,在开始字出现之前可能有任何多个0字节。因此开始码字通常是字节对准的,而且在其之前填充了一些0位。

## I - 2. 4 传送流比特流要求

### I - 2. 4. 1 传送流的编码结构和参数

传送流的编码层允许将一道或多道程序合成为单个流。程序是一些有着相同时间基点的原始流的集合。从每个原始流来的数据被多路复用在一起,并带有足够的信息,允许一道程序的所有原始流能同步显示。

#### ITU-T Rec. H. 222. 0 | ISO/IEC 13818 多路复用传送流

传送流由一道或多道程序组成,每道程序由一个或多个原始流和一些其它的流多路复用在一起。每个原始流包含访问单元,也就是显示单元的编码表示。视频原始流的显示单元就是一幅图象,相应的访问单元包括此图象的所有编码数据。如果此图象是一组图象的第一幅(GOP),则访问单元也包含以 group\_start\_code 开始的一些前缀数据。group\_start\_code 在本推荐|国际标准的第 I 部分定义。如果此图象是队列首部之后的第一幅编码图象,则访问单元也包括了 sequence\_header\_code 开始的队列首部,sequence\_header\_code 在本推荐|国际标准的第 II 部分定义。而在队列尾部包含最后一幅编码图象的访问单元中含有 sequence\_end\_code(本推荐|国际标准第 I 部分定义了 sequence\_end\_code)。一个音频原始流的显示单元相当于一个音频帧采样的一些采样的集合。相应的音频访问单元从音频帧的同步字的第一个字节开始(ISO/IEC 13818-3 定义了音频帧)。

原始流数据加载在 PES 分组中。PES 分组由 PES 分组首部以及其后的分组数据组成。PES 分组是插在传送流分组中的,每个 PES 分组首部的第一个字节就是传送流分组有效负载的第一个字节。

PES 分组首部以 32 位的开始码字起头,此码识别该分组所属的流。PES 分组首部可能含有解码和/或显示时间标签(DTS 和 PTS),此标签针对于分组中第一个访问单元。

PES 分组首部也可含有其他的可选字段,而 PES 分组数据字段则包含来自一个原始流的长度可变的连续字节串。

传送流分组以 4 个字节的前缀开始,其中包括表 I -2-3 所定义的 13 比特的分组 ID (PID)、PID 通过程序特殊信息(PSI)表来识别传送流分组中所带的数据,一个 PID 值的传送流分组只带有来自一个原始流的数据。

PSI 表也在传送流中,有以下四种 PSI 表:

- 程序相关表
- 程序映射表
- 网络信息表
- 条件访问表

这些表中包含进行多路解调和显示程序的必要的和足够的信息。表 I -2-27 中的程序映射表就可以指明哪些 PID 值,也即哪些原始流组成一道程序。这个表也为每道程序指明带有 PCR 字段的传送流分组的 PID。

传送流分组可能是空分组。空分组是用来填充传送流的,它们可能在重新多路复用时被插入或删除。因此,传送到解码器的空分组的有效负载是不确定的。

本标准没有详细规定用作条件访问系统部分的编码数据。但本标准提供了一种机制,使程序服务提供者能够传送和识别待解码的数据,提供本标准所规定的数据。这种支持由传送流分组结构和表 I -2-27 中的条件访问表来提供。

#### I -2. 4. 2 传送流系统目标解码器(T-STD)

第 I -2. 4. 3 节中传送流的语义和第 I -2. 7 节中对这些语义的限制,要求严格定义字节到达、解码事件以及它们发生的时间。在本国际标准中借助传送流系统目标解码器这个虚拟的解码器来完成这些定义。

T-STD 是一个概念上的模型,用来精确地定义这些术语,并可用来规范在构造和检验传送流中的解码处理,T-STD 就是为了这些目的而定义的。T-STD 的结构和时序描述都不会妨碍其它不同结构和时序的解码器连续同步地显示传送流。

下面的注释用来描述传送流系统目标解码器,其中一部分在图 I -2-6 中画出。

- i, i', i'' 传送流中字节的序号。第一个字节的序号为 0  
j 原始流中访问单元的序号  
k, k', k'' 原始流中显示单元的序号  
n 原始流的序号  
p 传送流中传送流分组的序号  
t(i) 指示传送流中第 i 个字节进入系统目标解码器的时间,以秒为单位  
PCR(i) 在 PCR 字段中的编码时间,以 27MHz 的系统时钟为单位计算。其中 i 是 PCR\_base 字段最后一个字节的序号  
A<sub>n</sub>(j) 是原始流 n 中第 j 个访问单元。注意,访问单元按解码顺序排序  
td<sub>n</sub>(j) 原始流 n 中第 j 个访问单元在系统目标解码器中的解码时间,以秒为单位  
P<sub>n</sub>(k) 原始流 n 中第 k 个显示单元

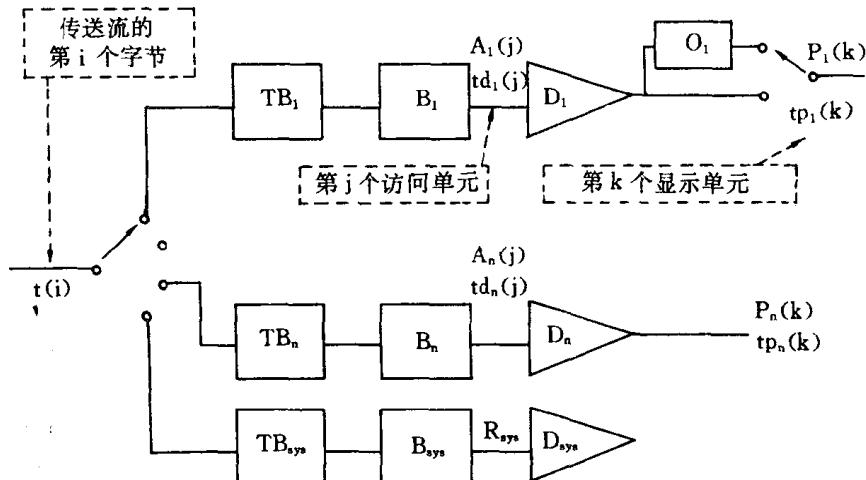


图 I - 2-6 传送流系统目标解码器的表示法

- $tp_n(k)$  原始流  $n$  中第  $k$  个显示单元在系统目标解码器中的显示时间,以秒为单位
- $t$  是以秒为单位的时间
- $F_n(t)$  在时间  $t$  原始流  $n$  在系统目标解码器的输入缓冲区填充的字节数
- $B_n$  在系统目标解码器中属于原始流  $n$  的主缓冲区
- $BS_n$  在系统目标解码器中属于原始流  $n$  的主缓冲区的大小,以字节为单位
- $B_{sys}$  在系统目标解码器中属于正在解码的程序的系统信息的主缓冲区
- $BS_{sys}$  在系统目标解码器中属于正在解码的程序的系统信息时主缓冲区的大小,1536 个字节。
- $TB_{sys}$  在系统目标解码器中属于正在解码的程序的系统信息的输入缓冲区
- $TBS_{sys}$  在系统目标解码器中属于正在解码的程序的系统信息的输入缓冲区的大小,512 个字节
- $TB_n$  在系统目标解码器中属于原始流  $n$  的传送缓冲区
- $TBS_n$  在系统目标解码器中属于原始流  $n$  的传送缓冲区大小,512 个字节
- $D_{sys}$  为程序流  $n$  的系统信息解码的解码器
- $D_n$  为原始流  $n$  解码的解码器
- $O_n$  将视频原始流  $n$  重排的缓冲器
- $R_{sys}$  数据从  $B_{sys}$  中移出的速率
- $R_{x_n}$  数据从  $TB_n$  中移出的速率
- $R_{x_{sys}}$  数据从  $TB_{sys}$  中移出的速率: 1Mbps

#### I - 2. 4. 2. 1 系统时钟频率

T-STD 中所用时序信息由本推荐|国际标准所定义的几个数据字段传送,可参考 I - 2. 4. 3. 4 节和 I - 2. 4. 3. 6 节。这信息以程序系统时钟的采样值的形式编码于 PCR 字段。实际的解码器可以根据这些值以及它们到达的时间重建时钟。下面是对解码器收到

的 PCR 字段值形式的系统时钟频率的一些限制。

系统时钟频率以 Hz 为单位并需要满足以下条件；

$$27\ 000\ 000 - 540 \leq \text{系统时钟频率} \leq 27\ 000\ 000 + 540$$

$$\text{系统时钟频率的时间变化率} \leq 75 \times 10^{-3} \text{Hz/s}$$

程序的系统时钟频率有可能比所要求的更精确，这种精确度上的提高可以通过 I - 2.6.20 节的系统时钟描述符传送给解码器。

“系统时钟频率”在此推荐 | 国际标准中用来表示适合此限制的时钟频率，为了讨论方便，在 PCR, PTS 或 DTS 方程中得到的时间值是精确到  $(300 \times 2^{33} / \text{系统时钟频率})$  秒的整数倍。这是因为 PCR 时序信息是将系统时钟频率的三百分之一编成 33 位码并加上 9 位余数，而 PTS 和 DTS 就是将系统时钟频率的三百分之一编成 33 位码。

#### I - 2.4.2.2 传送流系统目标解码器的输入

输入到传送流系统目标解码器(T-STD)的就是传送流。传送可能包括独立时间基点上的多道程序，然而，T-STD 在一个时间只为一道程序解码。在 T-STD 模式中，所有的时间信息都针对那道程序的时间基点。

数据从传送流进入 T-STD 的速率是一个分段常数。第 i 个字节在时间  $t(i)$  进入，这个字节进入 T-STD 的时间可以通过对输入流的程序参考时钟(PCR)字段解码而恢复，此字段被编码在传送流分组的调整字段。PCR 在方程 I - 2-3 中分两部分编码：一个以系统时钟频率的 1/300 为单位，称为 PCR\_base(方程 I - 2-1)，另一个以系统时钟频率为单位，称为 PCR\_ext(方程 I - 2-2)。编码在 PCR(i) 中的数值代表了  $t(i)$ ，i 指包含 PCR\_base 字段的最后一个位的字节。

具体规定如下：

$$\text{PCR\_base}(i) = ((\text{系统时钟频率} \times t(i)) \text{D I V } 300) \% 2^{33} \quad (\text{I - 2-1})$$

$$\text{PCR\_ext}(i) = ((\text{系统时钟频率} \times t(i)) \text{D I V } 1) \% 300 \quad (\text{I - 2-2})$$

$$\text{PCR}(i) = \text{PCR\_base}(i) \times 300 + \text{PCR\_ext}(i) \quad (\text{I - 2-3})$$

所有其它输入字节到达的时间  $t(i)$ ，如方程 I - 2-4 所示，可以由  $\text{PCR}(i'')$  和此数据到达时的传送速率计算而得。传送速率可以由在传送流中同一程序的包含最后一位的两个连续的 PCR 字段字节之间的字节数除以这两个 PCR 编码时间值之间的差值来确定。

$$t(i) = \frac{\text{PCR}(i'')}{\text{系统时钟频率}} + \frac{i - i''}{\text{传送速率}(i)} \quad (\text{I - 2-4})$$

其中：

i 程序中正在被解码的字节序号；

i'' 包含程序中最近被解码的 PCR\_base 的最后一位的字节的序号；

$\text{PCR}(i'')$  编码在 PCR\_base 和 PCR\_ext 字段中的时间，以系统时钟为单位。

传送速率 transport\_rate 由下式决定：

$$\text{transport\_rate}(i) = \frac{((i' - i'') \times \text{系统时钟频率})}{\text{PCR}(i') - \text{PCR}(i'')} \quad (\text{I - 2-5})$$

其中：

i' 程序中即将被译码的 PCR\_base 的最后一个字节的序号。

注意:  $i'' < i \leq i'$

PCR 容忍度是接收到的 PCR 的最大允许误差,这种误差可以由初始 PCR 值和在重新多路复用过程中对 PCR 的修改引起的。它并不包括由于网络抖动引起的分组到达时间错误。PCR 的容忍度限制了一份传送流的存储拷贝的 PCR 错误,PCR 容忍度为±500ns。

#### I -2. 4. 2. 3 缓冲

包含来自原始流 n 的数据的整个传送流分组,由 PID 指明,被送入属于流 n 的缓冲区,TB<sub>n</sub>。这就包括传送流分组的复制,其中有一些没有有效负载的分组。第 i 个字节从系统目标解码器进入 TB<sub>n</sub> 是瞬间完成的,因此第 i 个字节进入大小为 TBS<sub>n</sub> 的流 n 的缓冲区的时间也为 t(i)。

所有进入 TB<sub>n</sub> 缓冲区的数据以速率 R<sub>x</sub> 移出。其中属于 PES 分组内容的字节被移入主缓冲区 B<sub>n</sub>,其它的字节不移入,并可能用来控制整个系统。复制的传送流分组不被送入 B<sub>n</sub>。

缓冲区 TB<sub>n</sub> 将按以下方法排空:如果 TB<sub>n</sub> 中没有数据,则 R<sub>x</sub> 应为零,否则:

$$\text{对视频} \quad R_x = 1.2 \times R_{\max}[\text{p.1}]$$

其中: R<sub>max</sub>[p.1]由本推荐|国际标准第 I 部分表 I -8.12 中框架和级别所决定,表 I -8.12

规定了在一个特定框架和级别上每个原始视频流的上限。

$$\text{对音频} \quad R_x = 2 \times 10^6 \text{ bit/s}$$

$$\text{对系统数据} \quad R_x = 1 \times 10^6 \text{ bit/s}$$

R<sub>x</sub> 以系统时钟频率为基础计算。

针对解码过程所选的程序,包含系统信息的完整传送流分组以传送流速率进入系统传送缓冲区,TB<sub>sys</sub>。这类分组有 PID 值为 0 或 1 的传送流分组和由程序关联表识别的具有 program\_map\_PID 值的传送流分组,由 NIT PID 所标志的网络信息表(NIT)数据不送入 TB<sub>sys</sub>。

传送缓冲区的大小固定为 512 个字节。

主缓冲区的大小从 BS<sub>1</sub> 到 BS<sub>n</sub> 如下定义:

$$\text{对视频} \quad BS_n = BS_{\text{mux}} + BS_{\text{dec}} + BS_{\text{oh}}$$

其中:  $BS_{\text{mux}} = 4_{\text{ms}} \times R_{\max}[\text{p.1}]$ ;  $BS_{\text{dec}} = VBV_{\max}[\text{p.1}]$

$R_{\max}[\text{p.1}]$  和  $VBV_{\max}[\text{p.1}]$  分别为最大视频比特流和最大 VBV 大小。对应于本推荐|国际标准第 I 部分表 I -8-12 和表 I -8-13 所定义的每种框架、级别和层次。

$$BS_{\text{oh}} = (1/750)s \times R_{\max}[\text{p.1}]$$

BS<sub>n</sub> 中的  $BS_{\text{mux}} = 4_{\text{ms}} \times R_{\max}[\text{p.1}]$  部分用于多路复用的缓冲,剩下的  $BS_{\text{dec}}$ ,  $BS_{\text{oh}}$  也可用于初始化多路复用。

在 I -2. 5. 2. 4 节中定义的 P-STD 将 PES 分组开销占有的缓冲区直接限制于 PES 流。有可能,但没有必要从 PES 流构造传送流。

$$\text{对音频:} \quad BS_n = BS_{\text{mux}} + BS_{\text{dec}} + BS_{\text{oh}} = 3584 \text{ 字节}$$

访问单元缓冲区 BS<sub>dec</sub> 和 PES 分组总的缓冲区 BS<sub>oh</sub> 需满足

$$BS_{\text{dec}} + BS_{\text{oh}} \leq 2848 \text{ 字节}$$

3584 字节的缓冲区中的 736 个字节用于多路复用的缓冲,剩下的 2848 个字节,包括访问单元缓冲区( $BS_{dec}$ )、PES 分组头上的缓冲区( $BS_{oh}$ )和附加的多路复用缓冲区( $BS_{mux}$ )。

系统数据的主缓冲区  $B_{sys}$  的大小为 1536 个字节。

对每个原始流缓冲区来说,在输入缓冲区  $A_n(j)$  中时间最久的访问单元的所有数据在解码时间  $td_n(j)$  将立即被移出。解码时间  $td_n(j)$  在 DTS 和 PTS 字段规定。紧接着访问单元  $j$  之后没有 DTS 和 PTS 字段的访问单元的解码时间  $td_n(j+1), td_n(j+2), \dots$  可由原始流中的信息推导而得,请参考附录 I -C 和附录 II -C 或 ISO/IEC 11172 的附录 C,也可以参考 I -2.7.6 节。在 ITU-T Rec. H. 262 | ISO/IEC 13818-2 视频访问单元中为此目的而定义的数据在 ITU-T Rec. H. 262 | ISO/IEC 13818-2 附录 II -C 的 II -C.5 节也有定义。在 ISO/IEC 13818-3 音频流中,访问单元由音频帧组成。那些在访问单元之前被立即存储或包含在访问单元数据之中的 PES 分组首部和访问单元同时被移去。一旦访问单元被移去,它就同时被解码为一个显示单元。

$TB_n, TB_{sys}, B_n$  缓冲区决不会溢出。 $B_n$  缓冲区只是在一幅略过的图象产生并处于低延迟模式时才发生下溢, $B_{sys}$  缓冲区决不会溢出。

### 系统数据

只要  $B_{sys}$  缓冲区中至少有一个字节,系统数据就以速率  $R_{sys}$  从主缓冲区  $B_{sys}$  中移出。

$$R_{sys} = \max[80\text{ kbps}, \text{transport\_rate}(i)/500] \quad (\text{I -2-6})$$

在高的传送率情况下  $R_{sys}$  的增加使得程序联系表的数据速率的增加成为可能。 $R_{sys}$  是以系统时钟频率为基础的。

### 低延迟

当视频队列扩展中的低延迟标志被置为‘1’时(ITU-T Rec. H. 262 | ISO/IEC 13818-2 II -6.2.2.3 节),VBV 缓冲区可能下溢。当 T-STD 的原始流缓冲区  $B_n$  在时间  $td_n(j)$  被检查时,访问单元的数据可能不都在  $B_n$  中。当这种情况发生时,每隔两个字段时间间隔就要检查一次缓冲区,直到全部访问单元都在缓冲区为止。这时整个访问单元和相关的系统数据,如果有的话,应立即从  $B_n$  中移出, $B_n$  缓冲区不能溢出。

在低延迟模式下,VBV 缓冲区的下溢可以无限制地连续发生。在上一段与比特流中编码的任意 DTS 或 PTS 值相一致时的最早时刻上,T-STD 解码器将从  $B_n$  缓冲区中退出访问单元的数据。请注意,根据 DTS 和 PTS 的说明,直到 VBV 缓冲区的下溢停止、比特流中发现 PTS 或 DTS 时为止,解码器可能不会重建正确的解码和显示时间。

### 特技模式

当含有 B 型视频访问单元的开始的 PES 分组首部中的 DSM\_trick\_mode 标志(见 I -2.4.3.6 节)被置为‘1’,而且 trick\_mode\_control 字段被置为‘001’(慢动作)或‘010’(静止帧)时,直到最后一次图象的字段被解码和显示时,B 图象的访问单元才从视频数据缓冲区  $B_n$  中移去。(在慢动作,慢倒和 field\_id\_Cntr1 时)显示字段和图象的重复在 I -2.4.3.7 节定义。访问单元在由 field\_rep\_Cntr1 值指示的时间从  $B_n$  中移出。

当含有视频访问单元的开始的 PES 分组首部中的 DSM\_trick\_mode 标志被置为‘1’时,PES 分组中第一个视频访问单元被从缓冲区中移走,则 trick\_mode 状态为真。

trick 模式保持为真,直到 T-STD 接收到的 PES 分组首部的 DSM\_trick\_mode 标志被置为 1 而且 PES 分组首部后的第一个视频访问单元从缓冲区中移走为止。当特技模式为真时,B<sub>n</sub> 缓冲区有可能下溢,其它对正常流的限制在特技模式为真时仍有效。

#### I -2. 4. 2. 4 解码

在 B<sub>1</sub> 到 B<sub>n</sub> 缓冲区的原始流可以同时在 D<sub>1</sub> 到 D<sub>n</sub> 解码器中进行解码,而在被输出 T-STD 显示之前可能在重排缓冲区 O<sub>1</sub> 到 O<sub>n</sub> 被延迟。重排缓冲区仅当一些访问单元不是以显示顺序出现的视频原始流时使用。在显示之前,这些显示单元需要重整,尤其是,如果 P<sub>n</sub>(k) 是一幅 I-图像或 P-图象,并处于一幅或多幅 B-图象之前,在显示之前它必须在重排缓冲区 O<sub>n</sub> 中延迟。任何先前存储在 O<sub>n</sub> 中的图象在当前图象能被存储之前即显示。P<sub>n</sub>(k) 应被延迟到下一幅 I-图像或 P-图象被解码之后。当它被存在重排缓冲区时,其后的 B-图象被解码和显示。

显示单元显示给观众的时间为 tp<sub>n</sub>(k)。对那不需要重排延迟的显示单元来说,tp<sub>n</sub>(k) 等于 td<sub>n</sub>(j),因为访问单元是立即被解码的,例如 B-帧就是这种情况。对那些有延时的显示单元来说,tp<sub>n</sub>(k) 与 td<sub>n</sub>(j) 的区别在于 P<sub>n</sub>(k) 在重排缓冲区中的延时是一幅图象周期的倍数。应当注意为原始视频流的开始提供足够的重排延时以满足整个流的要求。例如,一个流开始只含有 I- 和 P- 图象,但后来又含有 B- 图象,那么在流的开始就应包含重排延迟。

本推荐 | 国际标准的第 II 部分详细介绍了视频图象的重排。

#### I -2. 4. 2. 5 显示

一个解码系统的功能就是从压缩的数据流中重建显示单元,并在正确的显示时间同步显示。虽然实际的音频和视频显示设备通常有有限的和不同的延迟,以及一些由后续处理或输出功能引起的附加延迟,但对于系统目标解码器模型,这些延迟都为零。

在图 I -2-6 中的 T-STD,视频显示单元的延迟就在它的显示时间 tp<sub>n</sub>(k) 立即发生。

当解码器提供第一个采样的同时,T-STD 的音频显示单元在它的显示时间 tp<sub>n</sub>(k) 开始输出,余下的采样序列则按音频采样频率按顺序显示。

### I -2. 4. 3 传送流语法和语义的说明

下面的语法描述一个流的字节。

#### I -2. 4. 3. 1 传送流

表 I -2-2 传 送 流

语 法	位数	缩写
MPEG_transport_stream() { do { transport_packet() } while(nextbits() == sync_byte) }		

### I -2. 4. 3. 2 传送流分组层

表 I -2-3 ITU-T Rec. H. 222. 0|ISO/IEC 13818 传送流分组

语 法	位数	缩写
transport_packet() {		
sync_byte	8	bslbf
transport_error_indicator	1	bslbf
payload_unit_start_indicator	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
transport_scrambling_control	2	bslbf
adaptation_field_control	2	bslbf
continuity_counter	4	uimsbf
if(adaptation_field_control=='10'    adaptation_field_control=='11') {		
adaptation_field()		
}		
if(adaptation_field_control=='01'    adaptation_field_control=='11') {		
for(i=0;i<N;i++) {		
data_byte	8	bslbf
}		
}		
}		

### I -2. 4. 3. 3 传送流分组层中的字段语义定义

**sync\_byte** 值为‘0100 0111’(0×47)的固定 8 位字段, sync\_byte 与像 PID 那样的按规律产生的字段的值的选择应避免竞争的发生。

**transport\_error\_indicator** 1 比特标志位。当置为 1 时表明在相关的传送分组中至少有一个不可纠正的错误位。此标志应由传送层之外的实体置‘1’。当被置‘1’后, 在错误被纠正之前不能重置为‘0’。

**payload\_unit\_start\_indicator** 1 比特标志位, 用来指示传送流分组带有 PES 分组或 PSI 数据时的情况。

当传送流分组的有效负载带有 PES 分组数据时, payload\_unit\_start\_indicator 具有以下特性: ‘1’表明传送流分组的有效负载将以 PES 分组的第一个字节开始; ‘0’表明在传送流的开始不是 PES 分组。当 payload\_unit\_start\_indicator 置为‘1’时, 在所有传送流分组中有且仅有一个以 PES 分组开始。这也可用于 stream\_type 6 的私有流(表 I -2-35)。

当传送流分组的有效负载带有 PSI 数据时, payload\_unit\_start\_indicator 具有以下性质: 如果传送流分组带有一个 PSI 部分的第一个字节, payload\_unit\_start\_indicator 值应被置‘1’, 表明传送流分组的第一个字节带有 pointer\_field。如果传送流分组不带有一个 PSI 部分的第一个字节, payload\_unit\_start\_indicator 值应被置‘0’, 表明在有效负载中没有 pointer\_field。请参考 I -2. 4. 4. 1 和 I -2. 4. 4. 2 节。以上也可用于 stream\_

type5 的私有流(表 I -2-35)。

空分组的 payload\_unit\_start\_indicator 应置‘0’。

对于只带有私用数据的传送流分组在本推荐|国际标准中对此位的含义没有定义。

**transport\_priority** 1 位指示器。当被置为‘1’时表明相关的分组比其它具有相同 PID 但此位没有被置‘1’的分组有更高优先级。传送机制可根据此位确定在一个原始流中数据的优先级。根据不同的应用,此优先级字段的编码可以不考虑 PID,或者仅在一个 PID 内编码。此字段可能被通道特殊编码器或解码器修改。

**PID** 13 位字段,指示存储于分组有效负载中数据的类型,PID 值 0x0000 为程序关联表(表 I -2-25)保留,而 0x0001 为条件访问表(表 I -2-27)保留。PID 值 0x0000—0x000F 保留,0x1FFF 为空分组保留。

**transport\_scrambling\_control** 2 位字段,用来指示传送流分组有效负载的加密模式。传送流分组首部包括调整字段,不应被加密。在空分组时 transport\_scrambling\_control 字段的值应置为‘00’。加密控制值如表 I -2-4 所示。

表 I -2-4 加密控制值

取 值	描 述
00	未加密
01	用户定义
10	用户定义
11	用户定义

**adaptation\_field\_control** 2 位字段。用于指示本传送流分组首部是否跟随有调整字段和/或有效负载。调整字段控制值如表 I -2-5 所示。

表 I -2-5 调整字段控制值

值	描 述
00	为 ISO/IEC 未来使用保留
01	无调整字段,仅含有效负载
10	仅含调整字段,无有效负载
11	调整字段后为有效负载

ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 解码器应丢弃 adaptation\_field\_control 字段为‘00’的传送流分组。空分组此字段应为‘01’。

**continuity\_counter** 4 位字段。随着每一个具有相同 PID 的传送流分组而增加,当它达到最大值后又回复到 0。如果 adaptation\_field\_control 字段为‘00’或‘10’,则 continuity\_counter 不应增加。

在传送流中,复制的分组可能被做为两个连续的具有相同 PID 的传送流分组送出。复制的分组和原分组有着相同的 continuity\_counter 值,而 adaption\_field\_control 字段

应为‘01’或‘11’。除去程序参考时钟字段有效值(如果有的话)将被重新编码外,在复制的分组中,原分组中每一个字节都被拷贝。

在一个特定的传送流中那些具有相同 PID 的分组的 continuity\_counter 是连续的,或者与前一个有相同 PID 的分组相差 1 时,或者在遇到“不增加”条件(adaptation\_field\_control 为‘00’或‘10’,或上面所描述的复制分组)时,在 discontinuity\_indicator 为‘1’时(I-2.4.3.4 节),continuity\_counter 将不连续。空分组的此字段无定义。

**data\_byte** 数据字节是来自 PES 分组、PSI 部分以及不在这些结构中的私有数据的连续字节,空分组的 PID 为 0x1FFF。data\_byte 可以赋为任何值。

#### I -2. 4. 3. 4 调整字段(adaptation\_field)

传送流调整字段见表 I -2-6。

表 I -2-6 传送流调整字段

语 法	位数	缩写
adaptation_field()		
adaptation_field_length	8	uimsbf
if(adaptation_field_length>0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if(PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf
reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if(OPCR_flag == '1') {		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if(splicing_point_flag == '1') {		
splice_countdown	8	tcimsbf
}		
if(transport_private_data_flag == '1') {		
transport_private_data_length	8	uimsbf
for(i=0;i<transport_private_data_length;i++) {		
private_data_byte	8	bslbf
}		

续表

语 法	位数	缩写
}		
if(adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf
reserved	5	bslbf
if(ltw_flag == '1') {		
ltw_valid_flag	1	bslbf
ltw_offset	15	uimsbf
}		
if(piecewise_rate_flag == '1') {		
reserved	2	bslbf
piecewise_rate	22	uimsbf
}		
if(seamless_splice_flag == '1') {		
splice_type	4	bslbf
DTS_next_au[32..30]	3	bslbf
marker_bit	1	bslbf
DTS_next_au[29..15]	15	bslbf
marker_bit	1	bslbf
DTS_next_au[14..0]	15	bslbf
marker_bit	1	bslbf
}		
for(i=0;i<N;i++) {		
reserved	8	bslbf
}		
}		
for(i=0;i<N;i++) {		
stuffing_byte	8	bslbf
}		
}		

#### 1 -2. 4. 3. 5 调整字段的语义定义

**adaptation\_field\_length** 8 位字段, 用以指明紧接 adaptation\_field\_length 之后的调整字段中的字节数。0 值用来在传送流分组中插入一个单个填充字节。当 adaptation\_field\_control 为 '11' 时, adaptation\_field\_length 应在 0 到 182 之间; 若 adaptation\_field\_control 为 '10', 则 adaptation\_field\_length 应为 183。对带有 PES 分组的传送流来说, 如果 PES 分组的数据不足以填充整个传送流分组的有效负载字节, 则需插入填充字节。填

充可以通过定义比实际数据元素总长更长的调整字段来完成,这样在调整字段之后剩下的有效负载字节恰好与可获得的 PES 分组的数据配套。调整字段中的额外空间中填入填充字节。

这是带有 PES 分组的传送流分组所允许的唯一填充方法,那些带 PSI 信息的传送流分组的填充方法则在 I -2. 4. 4 节中描述。

**discontinuity\_indicator** 1 位字段,当置为‘1’时表明当前传送流分组的不连续状态为真。对于一个带有 PCR\_PID(I -2. 4. 4. 9 节)的传送流分组,若不连续状态为真,则具有相同 PID 的传送流分组的下一个 PCR 代表相应程序的新的系统时钟频率的采样,这也可能在出现不连续指示的同一个分组中发生。

当不连续状态为真时,含有新时间基点的第一个 PCR 的传送流分组中 discontinuity\_indicator 应置‘1’。那些在具有新时间基点 PCR 的分组之前的、有相同 PCR\_PID 的传送流分组中的 discontinuity\_indicator 也应置‘1’,一旦 discontinuity\_indicator 被置为‘1’,直到具有新时间基点的第一个 PCR 的传送流分组为止,所有具有相同的 PCR\_PIO 的传送流分组的此字段都应置‘1’。

在任何传送流分组中,若不连续状态为真,则本分组中 continuity\_counter 对于前一个具有相同 PID 的传送流分组来说可能不连续,continuity\_counter 的不连续点产生在传送流分组中不连续状态为真的情况下,而不连续状态是针对前一个具有相同 PID 的传送流分组而说的。continuity\_counter 的不连续点的产生最有可能发生在从不连续状态的开始到其结束这段时间内。尤其是,对某个具有特殊 PID 的传送流分组,若其 discontinuity\_indicator 为‘1’,且其 PID 不等于 PCR\_PID,则具有相同 PID 的下一个传送流分组的 discontinuity\_indicator 应置为‘1’,而第三个分组的 discontinuity\_indicator 不需置‘1’。

当不连续状态为真时,如果具有相同的 PID 的两个连续的传送流分组具有相同的 continuity\_counter,并且 adaptation\_field\_control 值为‘01’或‘11’,则解码器将丢弃第二个分组。传送流不应该通过有可能引起有效数据丢失的丢弃分组的方式来构造(如,丢失 PES 组负载,PSI 部分和私用部分)。

在 continuity\_counter 的不连续点之后,具有相同 PID 的传送流分组中,原始流的第一个字节应是一个原始流访问点的第一个字节(见 I -2. 4. 3. 4 节中的 random\_access\_indicator)。

在一个带有 PSI 信息的传送流分组中 continuity\_counter 的不连续点出现之后,PSI 部分中 version\_number 的单个不连续可能发生。在这样一个 continuity\_counter 的不连续点出现时,相应程序的 TS\_program\_map\_sections 的版本中 section\_length 等于 13,这样,其中不含有 program\_descriptor,而且也没有描述的原始流。在此之后每一个受影响的程序都应有一个 version\_number 加 1 的 TS\_program\_map\_section 版本,且包含完整的程序定义。以上指明了 PSI 数据中的版本改变。

当 discontinuity\_indicator 置‘0’时,不连续状态为‘假’。

一个 PCR 不连续点在具有新时间基点的 PCR 分组的第一个字节到达 T-STD 输入的时刻被及时定义。

在 PES 分组首部内针对新时间基点的 PTS 和 DTS 应称为‘新’时间标签;同样,PES

分组首部针对旧时间基点的 PTS 和 DTS 应称为‘旧’时间标签。下面这些要求对于在 PCR 不连续点附近的 PES 分组头部中时间标签的编码有效。

- (1) ‘旧’PTS 和 DTS 值不应在 PCR 不连续点之后到达 T-STD 输入。
- (2) ‘新’PTS 和 DTS 不应在 PCR 不连续点之前到达 T-STD 输入。

对于每个不连续状态，正常的 T-STD 缓冲区限制继续有效。

**random\_access\_indicator** 1 位字段，当被置为‘1’时，表明下一个有相同 PID 的 PES 分组应含有 PTS 字段和一个原始流访问点。如果 payload\_unit\_start\_indicator 为‘1’，则下一个 PES 分组的首字节就在此 random\_access\_indicator 为‘1’的传送流分组中出现，否则下一个 PES 分组的首字节就在下一个具有相同 PID 且 payload\_unit\_start\_indicator 为‘1’的传送流分组中。由于上述的原因，原始流访问点有如下定义：

视频：视频序列首部的第一个字节。

音频：音频帧的第一个字节。

当一个 PID 与 PSI 中 PCR\_PID 相等的传送流分组中的 random\_access\_indicator 置为‘1’时，本分组的调整字段至少包括 program\_clock\_reference\_base 和 program\_clock\_reference\_extension 字段。如果被置为‘0’，则不表明一个随机访问点是否产生。对私有数据此标志位无定义。

**elementary\_stream\_priority\_indicator** 1 位字段，在具有相同 PID 的分组中，用以指明此传送流分组有效负载中原始流数据的优先级。‘1’表明此有效负载比其它传送流分组的有效负载优先级高。在视频情况下仅当有效负载中含有一个或多个内部编码组块字节时，此字段才置为‘1’。‘0’表示此有效负载与其它此位不为‘1’的有效负载具有相同优先级。

**PCR\_flag** 1 位标志，‘1’表示调整字段包括一个 PCR 字段；‘0’表明此调整字段不含任何 PCR 字段。

**OPCR\_flag** 1 位标志，‘1’表示调整字段包括一个 OPCR 字段；‘0’表明此调整字段不含任何 OPCR 字段。

**splicing\_point\_flag** 1 位标志，当被置为‘1’时，表示 splice\_countdown 字段应出现在相应的调整字段中，规定拼接点的出现。‘0’表明调整字段中没有 splice\_countdown 字段。

**transport\_private\_data\_flag** 1 位标志，当被置为‘1’时，表明调整字段中含有一个或多个私用数据字节；‘0’表示调整字段中不含任何私用数据字节。

**adaptation\_field\_extension\_flag** 1 位标志，当被置‘1’时，表明存在调整字段扩展；‘0’表明调整字段没有扩展。

**program\_clock\_reference\_base; program\_clock\_reference\_extension** 程序参考时钟(PCR)在方程 I -2-3 中定义，是一个 42 位码字，由两部分组成：一部分以系统参考时钟的 1/300(90kHz)为单位，称为 program\_clock\_reference\_base(方程 I -2-1)，33 位字段；另一部分称为 program\_clock\_reference\_extension(方程 I -2-2)，以系统参考时钟(27MHz)为单位的 9 位字段，它的出现以 PCR\_flag 为标志。PCR 指示了 program\_clock\_reference\_base 的最后一个字节预定到达系统目标解码器输入的时间。

`program_clock_reference_extension` 字段表示了在一个 90kHz 的时钟周期开始之后, 27MHz 时钟的周期数。此字段的值在 0 到 299 之间。当其值为 299 时, `program_clock_reference_extension` 将转回到 0, 同时, `program_clock_reference_base` 字段的最低位加 1。对于那些带有视频或音频原始流的传送流分组, 如果在调整字段中有 PCR 字段, 则 PCR 字段必须对传送流分组中的原始流有效, 请参考 I -2.7.3 节中编码频率对 PCR 的要求。

**original\_program\_clock\_reference\_base ; original\_program\_clock\_reference\_extension** 可选的源程序参考时钟(OPCR)是由两部分组成的 42 位码, 这两部分包括基础和扩展, 分别相应于 PCR 中的两部分。OPCR 的出现由 OPCR\_flag 来标志。OPCR 字段仅在有 PCR 字段的传送流分组中出现。由单道程序或多道程序组成的传送流中都可以有 OPCR。

OPCR 帮助从一个传送流中重建一个只含单道程序的传送流。当重建原来单道程序的传送流时, OPCR 可以拷贝到 PCR 字段中。由此产生的 PCR 值仅当原单道程序的传送流完整正确重建时才有效。这样至少要求包括那些在原传送流中出现的任何 PSI 和私用数据分组, 也许还需要其他的私用安排, 这也意味着 OPCR 必须是原单程序传送流相关的 PCR 的拷贝。

OPCR 具有以下表示:

$$\text{OPCR\_base}(i) = ((\text{系统参考时钟} \times t(i)) \text{DIV} 300) \% 2^{33} \quad (\text{I -2-7})$$

$$\text{OPCR\_base}(i) = ((\text{系统参考时钟} \times t(i)) \text{DIV} 1) \% 300 \quad (\text{I -2-8})$$

$$\text{OPCR}(i) = \text{OPCR\_base}(i) \times 300 + \text{OPCR\_ext}(i) \quad (\text{I -2-9})$$

解码器将忽略 OPCR 字段。任务多路复用器或解码器都不应修改 OPCR 字段。

**splice\_countdown** 8 位字段, 代表一个可正可负的数值, 正值代表在遇到一个拼接点之前, 在相关传送流分组之后具有相同 PID 的传送流分组的剩余数目, 复制的传送流分组和只含有调整字段的传送流分组除外。拼接点的出现是在含有 splice\_countdown 为 0 的传送流分组从 STD 模式的缓冲区 TB<sub>n</sub> 中送出之后, 在 splice\_countdown 为 0 的传送流分组中, 传送流分组有效负载的最后一个字节将是一访问单元的最后一个字节。在此之后的具有相同 PID 的传送流分组可能含有相同类型的不同原始流的数据。

具有相同 PID 的下一个传送流分组的有效负载将从 PES 分组的第一个字节开始, 此 PES 分组有效负载的第一个字节将是一个具有相同 PID 的传送流分组所包含的原始流的一个访问点的第一个字节。因此前一个访问单元恰好与分组边缘对准或是通过‘填充’而达到对准。在拼接点之后, countdown 字段也将出现。当 splice\_countdown 是一个负值 (-n) 时, 表明相应的传送流分组是拼接点之后第 n 个分组。

**transport\_private\_data\_length** 8 位字段, 表明紧接 transport\_private\_data\_length 之后私用数据的字节数。私用数据的字节数不应使私用数据延伸到调整字段之外。

**private\_data\_byte** 8 位字段, 将由 ISU/IEC 定义。

**adaptation\_field\_extension\_length** 8 位字段, 表示跟在此字段之后扩展调整字段的数据长度。

**ltw\_flag(legal\_time\_window\_flag)** 1位字段,当被置‘1’时代表 ltw\_offset 字段的存在。

**piecewise\_rate\_flag** 1位字段,置‘1’表示 piecewise\_rate 字段的存在。

**seamless\_splice\_flag** 1位字段,当满足以下条件时将被置‘1’。

(1) 在 splice\_countdown 为 0 的分组中,结束的一个访问单元 AU<sub>a</sub> 的最后一个字节,在 BS<sub>dec</sub> 缓冲区中停留了 splice\_decoding\_delay-P 时间,其中 splice\_decoding\_delay 可从表 I-2-7 到表 I-2-16 中推出,‘P’是访问单元 AU<sub>b</sub> 显示的持续时间,AU<sub>b</sub> 是 AU<sub>a</sub> 在解码时正在显示的访问单元。

(2) 如果解码器的输入在一个拼接点变为具有等于 max\_splice\_rate(从表 I-2-7 到表 I-2-16) 的恒定速率的流,并持续长度为 splice\_decoding\_delay 的时间时,对于视频,BS<sub>dec</sub> 等于最大的相应框架和级别的 VBV 的大小(本推荐|国际标准第Ⅱ部分表Ⅱ-8-13);而对于音频,BS<sub>dec</sub> 等于音频解码缓冲区大小则这样的 T-STD 不应该发生溢出。

一旦此标志置为‘1’,则在遇到拼接点之前具有相同 PID 且 splicing\_point\_flag 为‘1’的所有分组,此位都为‘1’。如果 splicing\_point\_flag 为‘0’则此字段无定义。

**ltw\_valid\_flag(legal\_time\_window\_valid\_flag)** 1位字段,为‘1’时表示 ltw\_offset 有效,如果 splicing\_point\_flag 为‘0’则此字段无定义。

**ltw\_offset(legaltime window\_offset)** 15位字段,仅当如上面描述的 ltw\_valid\_flag 为‘1’时此字段才有定义。当有效时,ltw\_offset 是以 90kHz 为单位,是从包含此域的分组的第一个字节在 t<sub>s</sub> 进行替换。t<sub>s</sub> 满足以下条件:如果此传送流分组的第一个字节在间隙 [t<sub>s</sub>, t<sub>s</sub> + BS<sub>max</sub>/R<sub>max</sub>[p.1]] 之中到达 T-STD, 主缓冲区 B<sub>n</sub> 不会溢出也不会下溢,除了本推荐|国际标准第Ⅱ部分Ⅱ-6.2.2.3 节允许的 B<sub>vbv</sub> 下溢。

**piecewise\_rate** 22位字段,一个正整数,规定具有此 PID 的传送分组的比特率,这些分组以含有此字段的分组开始,到具有相同 PID 的分组结束,且此分组位于具有同样 PID 且再次包含该字段的分组之前。

**piecewise\_rate** 以 50 字节/s 为单位。‘0’值被禁止。

**splice\_type** 4位字段,用于从表 I-2-7 到表 I-2-16 中获得 splice\_decoding\_delay 和 max\_splice\_rate 值。

表 I-2-7 拼接参数表 1

“简单”框架“主”级别,“主”框架“主”级别,SNR 框架“主”级别(双层),“空域”框架“高-1440”级别(基层),“高”框架“主”级别(中+基层)

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	120	15.0
0001	150	12.0
0010	225	8.0
0011	250	7.2
0100-1111	保留	保留

表 I -2-8 拼接参数表 2

“主”框架“低”级别,SNR 框架“低”级别(双层),“高”框架“主”级别(基层)视频

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	115	4.0
0001	155	3.0
0010	230	2.0
0011	250	1.8
0100-1111	保留	保留

表 I -2-9 拼接参数表 3

“主”框架“高-1440”级别,“空域”框架“高-1440”级别(所有层),“高”框架“高-1440”级别(中+基层)视频

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	120	60.0
0001	160	45.0
0010	240	30.0
0011	250	28.5
0100-1111	保留	保留

表 I -2-10 拼接参数表 4

“主”框架“高”级别,“高”框架“高-1440”级别(所有层),“高”框架“高”级别(中+基层)视频

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	120	80.0
0001	160	60.0
0010	240	40.0
0011	250	38.0
0100-1111	保留	保留

表 I -2-11 拼接参数表 5

SNR 框架“低”级别(基层)视频

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	115	3.0
0001	175	2.0
0010	250	1.4
0011-1111	保留	保留

表 I -2-12 拼接参数表 6

SNR 框架“主”级别(基层)视频

splice-type	splice-decoding-delay(msec)	max-splice-rate(Mbit/s)
0000	115	10.0
0001	145	8.0
0010	235	5.0
0011	250	4.7
0100-1111	保留	保留

表 I -2-13 拼接参数表 7

“空域”框架“高-1440”级别(中+基层)视频

splice-type	splice-decoding-delay(msec)	max-splice-rate(Mbit/s)
0000	120	40.0
0001	160	30.0
0010	240	20.0
0011	250	19.0
0100-1111	保留	保留

表 I -2-14 拼接参数表 8

“高”框架“主”级别(所有层),“高”框架“高-1440”级别(基层)视频

splice-type	splice-decoding-delay(msec)	max-splice-rate(Mbit/s)
0000	120	20.0
0001	160	15.0
0010	240	10.0
0011	250	9.5
0100-1111	保留	保留

表 I -2-15 拼接参数表 9

“高”框架“高”级别(所有层)视频

splice-type	splice-decoding-delay(msec)	max-splice-rate(Mbit/s)
0000	120	25.0
0001	165	18.0
0010	250	12.0
0011-1111	保留	保留

表 I -2-16 拼接参数表 10

“高”框架“高”级别(基层)视频

splice_type	splice_decoding_delay(msec)	max_splice_rate(Mbit/s)
0000	120	100.0
0001	160	75.0
0010	240	50.0
0011	250	48.0
0100-1111	保留	保留

DTS\_next\_au(decoding\_time\_stamp\_next\_access\_unit) 33 位字段,由三部分组成,指明一个非拼接的或无缝拼接流的下一个访问单元的 DTS 值。此值在拼接之前由系统时钟推出。

**stuffing\_byte** 可由编码器插入的 8 位固定码字‘1111 1111’,由解码器丢弃。

#### I -2-4.3.6 PES 分组

表 I -2-17 PES 分组

语 法	位数	缩写
PES_packet()		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if(stream_id != program_stream_map && stream_id != padding_stream && stream_id != private_stream_2 && stream_id != ECM && stream_id != EMM && stream_id != program_stream_directory) {		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf

续表

语 法	位数	缩写
<b>PES_header_data_length</b>	<b>8</b>	<b>uimsbf</b>
if(PTS_DTS_flags == '10') {		
'0010'	<b>4</b>	<b>bslbf</b>
PTS[32..30]	<b>3</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
PTS[29..15]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
PTS[14..0]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
}		
if(PTS_DTS_flags == '11') {		
'0011'	<b>4</b>	<b>bslbf</b>
PTS[32..30]	<b>3</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
PTS[29..15]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
PTS[14..0]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
'0001'	<b>4</b>	<b>bslbf</b>
DTS[32..30]	<b>3</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
DTS[29..15]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
DTS[14..0]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
}		
if(ESCR_flag == '1') {		
reserved	<b>2</b>	<b>bslbf</b>
ESCR_base[32..30]	<b>3</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
ESCR_base[29..15]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
ESCR_base[14..0]	<b>15</b>	<b>bslbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
ESCR_extension	<b>9</b>	<b>uimsbf</b>
marker_bit	<b>1</b>	<b>bslbf</b>
}		
if(ES_rate_flag == '1') {		

续表

语 法	位数	缩写
<b>marker_bit</b>	1	<b>bslbf</b>
<b>ES_rate</b>	22	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
}		
if(DSM_trick_mode_flag == '1') {		
<b>trick_mode_control</b>	3	<b>uimsbf</b>
if(trick_mode_control == '000') {		
<b>field_id</b>	2	<b>bslbf</b>
<b>intra_slice_refresh</b>	1	<b>bslbf</b>
<b>frequency_truncation</b>	2	<b>bslbf</b>
}		
else if(trick_mode_control == '001') {		
<b>field_rep_cntrl</b>	5	<b>uimsbf</b>
}		
else if(trick_mode_control == '010') {		
<b>field_id</b>	2	<b>uimsbf</b>
<b>reserved</b>	3	<b>bslbf</b>
}		
else if(trick_mode_control == '011') {		
<b>field_id</b>	2	<b>bslbf</b>
<b>intra_slice_refresh</b>	1	<b>bslbf</b>
<b>frequency_truncation</b>	2	<b>bslbf</b>
else if(trick_mode_control == '100') {		
<b>field_rep_cntrl</b>	5	<b>uimsbf</b>
}		
else		
<b>reserved</b>	5	<b>bslbf</b>
}		
if(additional_copy_info_flag == '1') {		
<b>marker_bit</b>	1	<b>bslbf</b>
<b>additional_copy_info</b>	7	<b>bslbf</b>
}		
if(PES_CRC_flag == '1') {		
<b>previous_PES_packet_CRC</b>	16	<b>bslbf</b>
}		
if(PES_extension_flag == '1') {		
<b>PES_private_data_flag</b>	1	<b>bslbf</b>
<b>pack_header_field_flag</b>	1	<b>bslbf</b>

续表

语 法	位数	缩写
<b>program_packet_sequence_counter_flag</b>	1	bslbf
<b>P-STD_buffer_flag</b>	1	bslbf
<b>reserved</b>	3	bslbf
<b>PES_extension_flag_2</b>	1	bslbf
if(PES_private_data_flag == '1') {		
<b>PES_private_data</b>	128	bslbf
}		
if(pack_header_field_flag == '1') {		
<b>pack_field_length</b>	8	uimsbf
<b>pack_header()</b>		
}		
if(program_packet_sequence_counter_flag == '1') {		
<b>marker_bit</b>	1	bslbf
<b>program_packet_sequence_counter</b>	7	uimsbf
<b>marker_bit</b>	1	bslbf
<b>MPEG1_MPEG2_identifier</b>	1	bslbf
<b>original_stuff_length</b>	6	uimsbf
}		
if(P-STD_buffer_flag == '1') {		
'01'	2	bslbf
<b>P-STD_buffer_scale</b>	1	bslbf
<b>P-STD_buffer_size</b>	13	uimsbf
}		
if(PES_extension_flag_2 == '1') {		
<b>marker_bit</b>	1	bslbf
<b>PES_extension_field_length</b>	7	uimsbf
for(i=0;i<PES_extension_field_length;i++) {		
<b>reserved</b>	8	bslbf
}		
}		
for(i=0;i<N1;i++) {		
<b>stuffing_byte</b>	8	bslbf
}		
for(i=0;i<N2;i++) {		
<b>PES_packet_data_byte</b>	8	bslbf
}		
}		

续表

语 法	位数	缩写
<pre> else if(stream_id == program_stream_map        stream_id == private_stream_2        stream_id == ECM        stream_id == EMM        stream_id == program_stream_directory) {     for(i=0;i&lt;PES_packet_length;i++) {         PES_packet_data_byte     } }  else if(stream_id == padding_stream) {     for(i=0;i&lt;PES_packet_length;i++) {         padding_byte     } } } </pre>	8	bslbf

#### I -2. 4. 3. 7 PES 分组字段的语义定义

**packet\_start\_code\_prefix** 24 位码,与其后的 stream\_id 一起构成标示一个分组开始的分组开始码字,packet\_start\_code\_prefix 等于 ‘0000 0000 0000 0000 0000 0001’ (0x0000001)

**stream\_id** 规定原始流的类型和数目,仅可从 ‘1011 1100’ 到 ‘1111 1111’ 之间取值。此字段的定义见表 I -2-18。

表 I -2-18 stream\_id 的取值

stream_id	注释	流的编码
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H. 262 ISO/IEC 13818-2 or ISO/IEC 11172-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream

续表

stream_id	注释	流的编码
1111 0010	5	DSM_CC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 xxxx		<b>reserved data stream-number xxxx</b> <b>'0100'-'1110' are reserved.</b>
1111 1111	4	<b>program_stream_directory</b>

标号 x 表示流的类型与此位取‘0’或‘1’无关。具体的流号则由 x 所取值决定。

注释：

- ① program\_stream\_map 类型的 PES 分组在第 I -2.5.4.1 节中有单独的语法说明。
- ② private\_stream\_1 类型的 PES 分组与 ISO/IEC\_13552\_stream 具有相同的 PES 分组语法, 和 ITU-T Rec. H. 222. 0|ISO/IEC 13818-2 视频和 ISO/IEC 13818-3 音频流一样。
- ③ private\_stream\_2, ECM\_stream, EMM\_stream 类型的 PEC 分组与 private\_stream\_1 的类似,除了语法中没有定义 PES\_packet\_length 字段。
- ④ program\_stream\_directory 类型的分组在 I -2.5.4.2 节有单独的语法说明。
- ⑤ DSM\_CC\_stream 类型的 PES 分组在附录 I -A 中有单独的语法说明。

**PES\_packet\_length** 16 位字段,说明在此字段最后一个字节之后 PES 分组的字节数。‘0’值表明 PES 分组的长度既没有说明也没有限制,这种情况只有在 PES 分组的有效负载是传送流分组中的视频原始流时才允许。

**PES\_scrambling\_control** 2 位字段,表示 PES 分组有效负载的加密模式,PES 分组首部,包括可能出现的可选字段,都不能被加密。PES 加密控制值如表 I -2-19 所示。

表 I -2-19 PES 加密控制值

取 值	描 述
00	未加密
01	用户定义
10	用户定义
11	用户定义

**PES\_priority** 1 位字段,指示 PES 分组有效负载的优先级。‘1’表示此 PES 分组的有效负载比其它此字段为‘0’的 PES 分组有效负载有更高的优先级。多路复用器可以使用此字段决定原始流中数据的优先级。此字段不应被传送机制修改。

**data\_alignment\_indicator** 1 位标志,当被置‘1’时表明此 PES 分组头部之后紧跟在表 I -2-46 中 data\_stream\_alignment\_descriptor 所定义的访问单元数据类型。若为‘0’则没有定义是否对准。

**copyright** 如此位为‘1’，则表明相应 PES 分组的有效负载中数据是有版权的。如果含有相关 PES 分组的原始流或相关的程序中出现了 PSI\_copyright\_descriptor(I-2.6.24 节)，则含有具有版权内容的 PES 分组的此字段需置‘1’，如果 copyright\_descriptor 没有出现，则此字段可为‘0’或‘1’。当置‘0’时表示此内容没有版权。

**original\_or\_copy** 若此位置‘1’，则相关 PES 分组的有效负载是原始数据；若为‘0’则是备份。

**PTS\_DTS\_flags** 2 位标志，若为‘10’，则 PES 分组首部有 PTS 字段；若为‘11’，则 PES 分组首部有 PTS 和 DTS 字段；若为‘00’则 PTS 和 DTS 都不在 PES 分组首部出现；‘01’值被禁止。

**ESCR\_flag** 1 位标志，当置‘1’时表示 ESCR 的 base 和 extension 字段都在 PES 分组首部出现；当置‘0’表示无 ESCR 字段。

**ES\_rate\_flag** 1 位标志，当置‘1’时表示 PES 分组首部有 ES\_rate 字段；若置‘0’表示无 ES\_rate 字段。

**DSM\_trick\_mode\_flag** 1 位标志，当置‘1’时表示有 8 位的 trick\_mode\_field；若置‘0’表示不出现此字段。

**additional\_copy\_info\_flag** 1 位标志，当置‘1’时表示有 additional\_copy\_info 字段；若置‘0’表示此字段不出现。

**PES\_CRC\_flag** 1 位标志，置‘1’时表示在 PES 分组中有 CRC 字段；若置‘0’则此字段不出现。

**PES\_header\_data\_length** 8 位字段，表示在 PES 分组首部中可选字段和填充字节所占有总字节数。可选字段的出现由 PES\_header\_data\_length 字段之前那个字节指示。

**marker\_bit** 1 位字段，值为‘1’。

**PTS(presentation\_time\_stamp)** PTS 是 33 位数字，编码成三个独立字段，表示此分组中第一个访问单元在系统目标解码器中的预定显示时间，PTS 的值为 27MHz 系统时钟的周期数除以 300，可以有 I-2.4.2 节中说明的容忍度。利用上述条款的注释，在 presentation\_time\_stamp 中的编码值为：

$$PTS(k) = ((\text{系统时钟频率} \times tp_n(k)) \text{DIV } 300) \% 2^{33} \quad (I-2-10)$$

其中： $tp_n(k)$  是显示单元  $P_n(k)$  的显示时间。 $P_n(k)$  是相应于此分组数据中开始的第一个访问单元的显示单元。如果访问单元的第一个字节包含在分组数据中则称访问单元在分组中开始（请参考 I-2.4.1 节中对音频和视频访问单元第一个字节的定义）。

显示时间和解码时间有如下关系：

对音频显示单元(PUs)、低延迟序列中的视频 PUs、B-图象来说，显示时间  $tp_n(k)$  应等于解码时间  $td_n(k)$ 。

对非低延迟序列中的 I/P 图象，并当访问单元(AUs) $k$  和  $k'$  之间没有解码不连续时，则显示时间  $tp_n(k)$  应等于解码时间  $td_n(k')$ ， $k'$  是  $k$  的下一个传送的 I/P 图象（请参考 I-2.7.6 节）。如果有解码不连续或遇上流结束， $tp_n(k)$  和  $td_n(k)$  之间的差值和假设原始流没有不连续和没有结束的一样。

注意：低延迟序列是 low-flag-delay 标志被置位的视频序列（见 ITU-T Rec. H.262 | ISO / IEC 13818-2 中的 I -6.2.3 节）。

如果音频中有滤波，则此系统模型假设滤波不引进延迟，因此，在编码中 PTS 所指的采样和解码中 PTS 所指的采样相同。在分级编码情况下，相应的 PTS 应和相应的访问单元在所有分级层次中保持一致。请参考 I -2.7.5 节。

**DTS (decoding\_time\_stamp)** 33 位数字，被编成三个独立字段，表示在分组中开始的第一个访问单元预计在系统目标解码器中的解码时间。DTS 的值为 27MHz 的系统时钟的周期数除以 300，在 I -2.4.2.1 节中有对容忍度的说明。由 I -2.4.2 节中的注释，编码在 decoding\_time\_stamp 中的值为：

$$DTS(j) = ((\text{系统时钟频率} \times td_n(j)) \text{ DIV } 300) \% 2^{33} \quad (I -2-11)$$

其中： $td_n(j)$  是访问单元  $A_n(j)$  的解码时间。 $A_n(j)$  是第一个在此分组数据中开始的访问单元。如果一个访问单元的第一个字节包含在此分组数据中，则称一个访问单元在分组中开始（请参考 I -2.4.2 中关于音频视频访问单元第一个字节的定义）。

在分级编码中，相关的 DTS 必须与相应的访问单元在所有分级层次中保持一致。请参考 I -2.7.6 节。

**ESCR (elementary\_stream\_clock\_reference)** 可选的 42 位字段，由 33 位的 ESCR\_base 字段和 9 位的 ESCR\_ext 字段组成。当其出现在 PES 流中，表示 ESCR\_base 的最后一个字在系统目标解码器中的预定出现时间。ESCR\_base 的值是 90kHz 的系统时钟的周期数，可以有 I -2.4.2.1 说明的容忍度。

特别地：

$$ESCR\_base(i) = ((\text{系统时钟频率} * t(i)) \text{ DIV } 300) \% 2^{33} \quad (I -2-12)$$

$$ESCR\_ext(i) = ((\text{系统时钟频率} * t(i)) \text{ DIV } 1) \% 300 \quad (I -2-13)$$

$$ESCR(i) = ESCR\_base(i) \times 300 + ESCR\_ext(i) \quad (I -2-14)$$

ESCR\_extension 字段表示在一个 90kHz 的时钟周期开始后 27MHz 时钟的周期数，此字段的值在 0 到 299 之间。当其值为 299 时，ESCR\_ext 将回复到 0，同时，ESCR\_base 字段会加 1。请参考 I -2.7.4 节。

**ES\_rate (elementary\_stream\_rate)** 正整数，在传送流情况下表明系统目标解码器从 PES 分组中接收字节的速率。直到遇到一个新的 ES\_rate 字段之前，对本 PES 分组之后那些属于相同 PES 流的分组来说 ES\_rate 都是有效的。ES\_rate 的值以 50 字节/s 为单位。0 值被禁止。在本推荐 | 国际标准的 I -2.5.2 节中，ES\_rate 的值定义 PES 流中字节到达 P-STD 输入的时间。各 PES 分组编码在 ES\_rate 字段中的数值可能不同。

**trick\_mode\_control** 3 位字段，用来表示哪种 trick\_mode 被应用于相应的视频流。注意当 trick\_mode 状态停止时，流中可能出现一个不连续点。trick 模式控制值如表 I -2-20 所示。

表 I -2-20 trick 模式控制值

取 值	描 述
'000'	快进
'001'	慢进
'010'	静止
'011'	快倒
'100'	慢倒
'101'-'111'	保留

在 trick\_mode 情况下,非正常的解码和显示速度可能引起在视频原始流数据中定义某些字段的值发生错误。同样,对 slice 结构的语义限制可能有效,除了以下这些视频语法元素:

```
bit_rate
vbv_delay
repeat_first_field
v_axis_positive
field_sequence
subcarrier
burst_amplitude
subcarrier_phase
```

解码器在 trick\_mode 中,不能依据这些字段所编码的值。

**fast\_forward** trick\_mode\_control 字段的值为 '000',表示一个快速向前的视频流。intra\_slice\_refresh 位置 '1' 表示有可能丢失宏块,那些宏块是解码器可用先前解码图象的相同位置的宏块来代替的宏块。表 I -2-21 中定义的 field\_id 指出哪些扫描场应被显示。frequency\_truncation 字段表明一个受限系数集将被包括在内。此字段值的含义在表 I -2-22 中。

**slow\_motion**—在 trick\_mode\_control 字段中的 '001' 值表示一个慢动作的视频流。field\_rep\_cntrl 字段值表明一个隔行扫描图象的每个扫描场显示的次数,或一个逐行扫描图象的显示次数,在隔行扫描的图象中,首先是顶部场显示 field\_rep\_cntrl 时间,然后底部场显示相同的时间。

**freeze\_frame** trick\_mode\_control 字段为 '010',表示一个静止帧视频流。field\_id 字段在表 I -2-21 中定义。为“Field\_ID 字段控制值”。field\_id 字段通常指含有此字段的 PES 分组中第一个在其中开始的视频访问单元。如果 PES 分组中有效负载字节数为 0,则 field\_id 字段指的是在此之前最近的视频访问单元。

**fast\_reverse** trick\_mode\_control 字段为 '011',表示一个快倒播放的视频流。intra\_slice\_refresh 字段可置为 '1',表示可能丢失的宏块,那些宏块为解码器可以用先前

解码图象的相同位置的宏块来代替的宏块,表 I -2-21 定义的 field\_id 字段表示将被显示的扫描场,frequency\_truncation 字段表示一个可能包含的受限系数集。此字段值的含义示在表 I -2-22“系数选择值”中。

**slow\_reverse trick\_mode\_control** 字段为 ‘100’ , 表示一个慢倒播放的视频流。field\_rep\_cntrl 字段值表示隔行扫描图象的每一个扫描场显示的次数,或逐行扫描图象的显示次数。在隔行扫描中首先是底部场显示 field\_rep\_cntrl 次,然后首部场显示相同的次数。

**field\_id** 2 位字段,表示在特技方式中哪些场将被显示。以表 I -2-21 中的方式编码。

表 I -2-21 Field\_id 字段控制值

取 值	描 述
'00'	仅从顶部字段显示
'01'	仅从底部字段显示
'10'	显示整个帧
'11'	保留

**intra\_slice\_refresh** 1 位标志,置‘1’表示在此 PES 分组视频数据的编码间隙中可能丢失宏块;‘0’表示不发生此情况。本推荐|国际标准的第 I 部分有更详细的说明,解码器将用先前解码图象的相同位置的宏块代替丢失的宏块。

**frequency\_truncation** 2 位字段,表示可能用于此 PES 分组视频数据编码的受限系数集。其值定义在表 I -2-22 中。

表 I -2-22 系数选择植

值	描 述
'00'	仅 DC 系数非零
'01'	通常仅前三个系数非零
'10'	通常仅前六个系数非零
'11'	所有系数可为非零

**field\_rep\_cntrl** 5 位字段,表示隔行扫描图象的每一个扫描场显示的次数,或者逐行扫描图象的扫描次数。它是 trick\_mode\_control 字段关于在隔行扫描中是首部场还是底部场首先被显示的一个函数。‘0’值被禁止。

**additional\_copy\_info** 7 位字段,包含和版权有关的私用数据。

**previous\_PES\_packet\_CRC** 16 位字段,包含 CRC 值,使得在以多项式  $x^{16} + x^{12} + x^5 + 1$  处理前一个 PES 分组的数据后,不包括 PES 分组首部;使得与附录 I -B 中解码器相似的 16 个寄存器输出为 0。

CRC 主要用于网络系统检测,比如孤立源的偶然错误。原始流的解码器并不使用它,

它仅计算数据字节,因为 PES 分组首部数据能在传送过程中被改变。

**PES\_private\_data\_flag** 1 位标志,置‘1’表示 PES 分组首部中含有私有数据;置‘0’则表示 PES 分组首部无私有数据。

**pack\_header\_field\_flag** 1 位标志,置‘1’表示一个 ISO/IEC 11172 组首部或程序流组首部存储在此 PES 分组首部。如果此字段在一个程序流的 PES 分组中,则必须置‘0’。在传送流中置‘0’表示 PES 首部没有组首部。

**program\_packet\_sequence\_counter\_flag** 1 位字段,置‘1’表示 PES 分组中有 program\_packet\_sequence\_counter 和 original\_stuff\_length 字段。置‘0’表示 PES 分组首部没有这些字段。

**P-STD\_buffer\_flag** 1 位字段,置‘1’时表示 PES 分组首部中有 P-STD\_buffer\_scale 和 P-STD\_buffer\_size 字段;置‘0’表示 PES。首部中没有这些字段。

**PES\_private\_data** 16 位字段,包含私用数据。与其前后的字段相连,此数据不能与 packet\_start\_code\_prefix(0x000001)发生竞争。

**PES\_extension\_flag-2** 1 位字数,当置‘1’时表示出现 PES\_extension\_field;置‘0’表示 PES\_extension\_field\_length 及其相关字段都不出现。

**pack\_field\_length** 8 位字段,指示 pack\_header\_field() 的字节长度。

**program\_packet\_sequence\_counter** 7 位字段,可选的计数器,随着每一个连续的程序流或 ISO/IEC 11172 系统流的 PES 分组或只含单道程序的传送流中的 PES 分组而增加,提供与 continuity\_counter 相似的功能,使得可以恢复一个程序流或 ISO/IEC 11172 系统流的原始 PES 分组序列。此计数器在达到最大值后恢复到 0。不应发生 PES 分组的重复,因此,在此程序多道复用的两个连续 PES 分组不会有相同的 program\_packet\_sequence\_counter 值。

**MPEG1\_MPEG2\_identifier** 1 位标置,置‘1’表示此 PES 分组中带有 ISO/IEC 11172 系统流的信息;置‘0’时表示此 PES 分组带有程序流的信息。

**original\_stuff\_length** 6 位字段,规定在原始的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 PES 分组首部或原始 ISO/IEC 11172-1 分组首部中填充字节的数目。

**P-STD\_buffer\_scale** [程序流才有] 1 位字段,表示用来解释其后的 P-STD\_buffer\_size 字段的幅度因子。如果前面的 stream\_id 表明是一个音频流,则 P-STD\_buffer\_scale 应为‘0’值;如果前面的 stream\_id 表明是一个视频流,则 P-STD\_buffer\_scale 应为‘1’。对其他的流类型,此值可能为‘1’或‘0’。

**P-STD\_buffer\_size** [程序流才有] 13 位无符号整数,定义 P-STD 中输入缓冲区 BS<sub>n</sub> 的大小。如果 P-STD\_buffer\_scale 为‘0’,则 P-STD\_buffer\_size 以 128 个字节为单位来衡量缓冲区大小;如果 P-STD\_buffer\_scale 为‘1’,则 P-STD\_buffer\_size 以 1024 个字节为单位。因此:

```
if(P-STD_buffer_scale==0)
    BSn=P-STD_buffer_size×128;
else
    BSn=P-STD_buffer_size×1024;
```

$$BS_n = P\text{-STD\_buffer\_size} \times 1024$$

( I -2-16)

一旦 P-STD\_buffer\_size 字段被 ITU-T Rec. H. 222.0 | ISO/IEC 13818 系统目标解码器接收, 则 P-STD\_buffer\_size 中的编码值立即有效。请参考 I -2.7.7 节。

**PES\_extension\_field\_length** 7 位字段, 规定在此字段之后 PES 扩展字域的字节长度。

**stuffing\_byte** 固定的 8 位数值 ‘1111 1111’, 由编码器插入以满足通道的要求, 由解码器丢去。在一个 PES 分组首部不能有超过 32 个填充字节。

**PES\_packet\_data\_byte** PES\_packet\_data\_byte 是来自分组的 stream\_id 或 PID 标志的原始流的连续的数据字节, 原始流的字节顺序不能改变。PES\_packet\_data\_byte 的数目 N 由 PES\_packet\_length 字段决定。N 应等于 PES\_packet\_length 减去 PES\_packet\_length 字段的最后一个字节与第一个 PES\_packet\_data\_byte 之间的字节数。

在 private\_stream\_1 或 private\_stream\_2 中, PES\_packet\_data\_byte 字段的内容由用户定义, 并且将来也不会由 ISO 规定。

**padding\_byte** 固定的 8 位数值 ‘1111 1111’ 由解码器丢去。

#### I -2.4.4 程序特殊信息(PSI)

程序特殊信息包括所有使解码器能进行程序的多路解调的 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 规格化数据和私用数据。程序由一些原始流组成, 每个都有一个 PID 标注程序、原始流或其中部分都有可能被条件访问。然而, 程序特殊信息不可以加密。

在传送流中, 程序特殊信息, 被分成 4 类表结构如表 I -2-23 所示。当这些结构可能被认为是单个表格时, 实际上它将被分段插入传送流分组中。

表 I -2-23 程序特殊信息

结构名	流类型	保留的 PID #	描述
程序关联表	ITU-T Rec. H. 222.0   ISO/IEC 13818-1	0x00	关联程序号与程序映射表 PID
程序映射表	ITU-T Rec. H. 222.0   ISO/IEC 13818-1	赋值的	说明一个或多个程序组成部分的 PID 值
网络信息表	私用	赋值的	物理网络参数: FDM 频率、发送器号码等等
条件访问表	ITU-T Rec. H. 222.0   ISO/IEC 13818-1	0x01	建立一个或多个(私用)EMM 流与每个独立的 PID 值的联系

ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 中定义的 PSI 表可被分成一段或多段置于传送流分组中。一段就是一个语法结构, 用来将 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 中定义的 PSI 表映射到传送流分组中。

除了 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 中定义的 PSI 表以外, 还可以有私用数据表。私用信息在传送流分组中的运载方法是私自定义的。它可以按用于携带 ITU-T

Rec. H. 222. 0|ISO/IEC 13818-1 中定义的 PSI 表的方法构造,这样私用数据的映射语法和 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 中定义的 SPI 表中所使用的映射一样。为了这个目的定义了私用部分,如果带有私用数据的传送流分组和带有程序映射表的传送流分组具有相同的 PID 值,则 private\_section 语法和语义有效。private\_section 使得数据以最小结构传送。若不使用此结构,则传送流分组中私用数据的映射在 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 中无定义。

分段的长度可变。分段的开始由传送流分组有效负载中的 pointer\_field 指示,此字段的语法在表 I -2-24 中说明。

在一个传送流中,分组填充字节 0xFF 可能出现在一个分段的最末字节之后,则此传送流分组在此之后的字节都为填充字节 0xFF,这些字节将由解码器弃去。在这种情况下,具有相同 PID 值的下一个传送流分组的有效负载将以值为 0x00 的 pointer\_field 开始,表明下一个分段的立即开始。

所有的传送流将含有一个或几个 PID 值为 0x0000 的传送流分组。这些传送流分组一起包含了此传送流中所有程序的一个清单,current\_next\_indicator 为‘1’的最新传送的表将对传送流中的当前数据有效,传送流中程序的任何改变都将在 PID 值为 0x0000 的传送流分组中的被修改的程序关联表的修改版本中描述。这些分段都应使用 table\_id 值 0x00。只有具有此 table\_id 值的分段才可以出现在 PID 值为 0x0000 的传送流分组中。

只要一个或几个传送流中的原始流被加密,与这些加密流相关的 PID 值为 0x0001 的传送流分组将被发送,其 CA\_section 中含有 CA\_descriptor 字段。这些传送流分组一起构成一个完整的条件访问表版本。Current\_next\_indicator 为‘1’的最近传送的此表版本将对传送流中当前数据有效。任何使得现有表格无效或不完整的加密中的变化应在修改的条件访问表格中描述。这些分段中 table\_id 值应为 0x01,只有具有此 table\_id 值的分段才能出现在 PID 值为 0x0001 的传送流分组中。

所有的传送流中将包含一个或多个传送流分组,其 PID 值在程序关联表中被标志,就好像传送流分组中应含有 TS 程序映射分段一样。在程序关联表中的每一道程序应在一个 TS 程序映射分段中描述,每道程序应在传送流本身中充分描述。current\_next\_indicator 为‘1’的最近传送的 TS\_program\_map\_section 版本应对传送流中的当前数据有效。传送流中任何程序定义的任何变化都应在具有与此程序 map\_PID 相同 PID 的传送流分组中的相应程序映表分段的修改版本中描述。所有带有一给定程序映射分段的传送流分组应具有相同 PID 值。在一道程序的连续存在期间,包括所有与之相关的文件,program\_map\_PID 不应发生改变。

table\_id 值为 0x02 的分段中应含有程序映射表信息。这样的分段可能由具有不同 PID 值的传送流分组运载。

网络信息表是可选的,其内容是私有的。如果存在,则它由具有同样 PID 值的传送流分组携带,称为 network\_PID, network\_PID 值由用户定义,并且当出现时,应在程序关联表中定义,program\_number 为保留的 0x0000,如果网络信息表存在,则应采取一个或多个私用分段的形式。

ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 中定义的 PSI 表格的一个分段的最大字节

数为 1K(1024 字节),一个私用分段的最大字节数为 4K(4096 字节)。

无论是 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 或私用的 PSI 数据都没有对出现开始码字、同步码字或其它比特模式的限制。

#### I -2. 4. 4. 1 指针

pointer\_field 的语法定义如表 I -2-24。

表 I -2-24 程序特殊信息指针

语 法	位数	缩写
pointer_field	8	uimsbf

#### I -2. 4. 4. 2 指针语法中字段的语义定义

**pointer\_field** 8 位字段,其值为在此字段之后到传送流分组有效负载的第一个分段的第一个字节之间的字节数(因此 pointer\_field 中的一个 0x00 值表明分段在 pointer\_field 之后立即开始)。当一给定传送流分组中至少有一个分段开始时,则 payload\_unit\_start\_indicator 应为‘1’,并且传送流分组有效负载的第一个字节应包含此指针。如果在一给定传送流分组中没有一个分段开始,则 payload\_unit\_start\_indicator 为‘0’,此分组的有效负载不需发送此指针。

#### I -2. 4. 4. 3 程序关联表

程序关联表提供一 program\_number 和包含此程序定义的传送流分组的 PID 值之间的对应之处。program\_number 是与一道程序相应的数字标号。

program\_number ‘0x0000’是为规定 network\_PID 而保留的。它识别带有网络信息表的传送流分组。

整个表按表 I -2-25 表示的语法分成一个或多个分段。

表 I -2-25 程序关联分段

语法	位数	缩写
program_association_section()		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<N;i++) {		

续表

语法	位数	缩写
<b>program-number</b>	<b>16</b>	<b>uimsbf</b>
<b>reserved</b>	<b>3</b>	<b>bslbf</b>
if(program-number == '0') {		
<b>network-PID</b>	<b>13</b>	<b>uimsbf</b>
}		
else {		
<b>program-map-PID</b>	<b>13</b>	<b>uimsbf</b>
}		
<b>CRC-32</b>	<b>32</b>	<b>rpchof</b>
}		

#### I -2. 4. 4. 4 Table\_id 赋值

table\_id 字段标明传送流 PSI 分段的内容,如表 I -2-26 所示。

表 I -2-26 table\_id 的赋值

值	描述
0x00	program_association_section
0x01	条件 access_section
0x02	program_map_section
0x03~0x3F	ITU-T Rec. H. 222. 0 ISO/IEC 13818 保留
0x40~0xFE	用户私用
0xFF	禁止

#### I -2. 4. 4. 5 程序关联表分段中各字段语义定义

table\_id 8 位字段,如表 I -2-26 所示,置为 0x00。

section\_syntax\_indicator 1 位字段,应置‘1’。

section\_length 12 位字段,头两位应为‘00’,它表明在 section\_length 字段之后此分段的字节数,包括 CRC。

transport\_stream\_id 16 位字段,用于在一个网络中从其它的多路复用中识别此传送流,其值由用户定义。

version\_number 5 位字段,整个程序关联表的版本号。版本号将随着每次程序关联表定义的改变而加 1,当达到 31 后,其值回复到 0。当 current\_next\_indicator 为‘1’时,version\_number 就为当前可用程序关联表的版本号。当 current\_next\_indicator 为‘0’时,version\_number 就是下一个可用程序关联表的版本号。

current\_next\_indicator 1 位指示,当置‘1’时表示所发送的程序关联表当前有效。

若置‘0’则表示发送的程序关联表不可用,下一个表才有效。

**section\_number** 8位字段,为此分段的号码。程序关联表第一分段的 section\_number 应为 0x00,它将随程序关联表的每一个附加分段而增加 1。

**last\_section\_number** 8位字段,表示全部程序关联表的最后一分段的号码(也就是最高 section\_number 的分段的号码)。

**program\_number** 16位字段,描述 program\_map\_PID 所适用的程序,如果此字段为 0x0000,则下一个参考 PID 将是网络 PID,其它情况下此字段的值由用户定义。在程序关联表的一个版本中,此字段不能取某单值多于一次。例如 program\_number 可以用于像广播通道这样的设计中。

**network\_PID** 13位字段,指定含有网络信息表的传送流分组的 PID。network\_PID 字段值由用户定义,但不能用那些为其它目的而保留的值。network\_PID 为可选项。

**program\_map\_PID** 13位字段,规定了传送流分组的 PID,这个分组应包含适用于 program\_number 所指明的程序的 program\_map\_section。一个 program\_number 不应有多个 program\_map\_PID 赋值。program\_map\_PID 的值由用户定义,但不应取为其它目的而保留的值。

**CRC\_32** 32位字段,包含在处理完整个程序关联表分段后使附录 I -B 中定义的解码器寄存器输出为 0 的 CRC 值。

#### I -2. 4. 4. 6 条件访问表

条件访问(CA)表提供一个或多个 CA 系统,它们的 EMM 流和一些特殊的参数之间的联系请参考第 I -2. 6. 16 节中 descriptor() 字段的定义。

在插入传送流分组之前,此表格将按表 I -2-27 所示的语法分成几段。

表 I -2-27 条件访问分段

语 法	位数	缩写
CA_section() table_id section_syntax_indicator '0' reserved section_length reserved version_number current_next_indicator section_number last_section_number for(i=0;i<N;i++) { descriptor() } CRC_32	8 1 1 2 12 18 5 1 8 8 32	uimsbf bslbf bslbf bslbf uimsbf bslbf uimsbf bslbf uimsbf uimsbf rpchof

#### I -2. 4. 4. 7 条件访问分段中各字段的语义定义

**table\_id** 8 位字段,总应取值‘0x01’,见表 I -2-26。

**section\_syntax\_indicator** 1 位字段,置‘1’。

**section\_length** 12 位字段,头两个位应为‘00’,表示在此字段之后立即开始的分段的字节数,包括 CRC。

**version\_number** 5 位字段,为整个条件访问表的版本号,当 CA 表所带的信息每一次发生改变时,都使版本号加 1,当其达到 31 时,回复到 0,若 current\_next\_indicator 为‘1’,则 version\_number 为当前可用条件访问表的版本号;若 current\_next\_indicator 为‘0’,则 version\_number 为下一个可用条件访问表的版本号。

**current\_next\_indicator** 1 位指示,置‘1’表示所发送的条件访问表是当前可用的;置‘0’则表示发送的条件访问表当前不可用,而下一个表将为有效。

**section\_number** 8 位字段,表示此分段的号码。条件访问表的第一个分段的 section\_number 为 0x00,其后将随条件访问表的每一个附加分段而增加 1。

**last\_section\_number** 8 位字段,规定条件访问表的最后一个分段的号码(也就是有最高 section\_number 的分段的号码)。

**CRC\_32** 32 位字段,在处理整个条件访问分段之后使得附录 I -B 定义的解码器中的寄存器输出为 0 的 CRC 值。

#### I -2. 4. 4. 8 程序映射表

程序映射表提供程序号码与组成它们的原始流之间的映射,这种映射的一个简单例子称为“程序定义”,程序映射表是一个传送流中所有程序定义的集合。此表将在分组中传送,其 PID 值是私自选择的。如果需要的话,可以使用多个 PID 值。在被插入到传送流分组之前,此表格将按表 I -2-28 所述语法分成一个或多个分段。

表 I -2-28 传送流程序映射分段

语 法	位数	缩写
TS_program_map_section()		
<b>table_id</b>	8	uimsbf
<b>section_syntax_indicator</b>	1	bslbf
'0'	1	bslbf
<b>reserved</b>	2	bslbf
<b>section_length</b>	12	uimsbf
<b>program_number</b>	16	uimsbf
<b>reserved</b>	2	bslbf
<b>version_number</b>	5	uimsbf
<b>current_next_indicator</b>	1	bslbf
<b>section_number</b>	8	uimsbf
<b>last_section_number</b>	8	uimsbf
<b>reserved</b>	3	bslbf

续表

语 法	位数	缩写
<b>PCR-PID</b>	13	uimsbf
<b>reserved</b>	4	bslbf
<b>program_info_length</b>	12	uimsbf
for(i=0;i<N;i++) { descriptor() }  for(i=0;i<N1;i++) { <b>stream_type</b> <b>reserved</b> <b>elementary_PID</b> <b>reserved</b> <b>ES_info_length</b> for(i=0;i<N2;i++) { descriptor() } }	8 3 13 4 12	uimsbf bslbf uimsnf bslbf uimsbf
<b>CRC_32</b>	32	rpchof
}		

在本推荐|国际标准 I -2.6 节中定义了 descriptor() 字段。

#### I -2.4.4.9 传送流程序映射分段中各字段语义定义

**table\_id** 8 位字段, 如表 I -2-26 所示, 在 TS\_program\_map\_section 中总是为 0x02。

**section\_syntax\_indicator** 1 位字段, 值应为 ‘1’。

**section\_length** 12 位字段, 头两位应为 ‘00’。规定此字段之后此分段的字节数, 包括 CRC。

**program\_number** 16 位字段, 规定此 program\_map\_PID 所对应的程序, 一道程序定义只能在一个 TS\_program\_map\_section 中, 这暗示一道程序的定义长度不应超过 1016 个字节, 附录 I -C 中有关于如何处理长度不足的情况的信息。program\_number 可用于像广播通道这样的设计。通过描述一道程序的不同原始流, 可以将不同来源(例如连续的事件)的数据联结成一个连续的使用同一 program\_number 的流的集合。附录 I -C 中有应用的例子。

**version\_number** 5 位字段, 表示 TS\_program\_map\_section 的版本号。版本号将随着此分段中信息的每一次改变而增加 1。当达到 31 后, 又回复到 0。版本号对应于一个单个程序的定义, 因此也就对应于单个分段。当 current\_next\_indicator 为 ‘1’ 时, version\_number 就是当前可用 TS\_program\_map\_section 的版本号; 若 current\_next\_indicator

为‘0’，则 version\_number 为下一个可用 TS\_program\_map\_section 的版本号。

**current\_next\_indicator** 1 位字段，当置‘1’表示所发送的 TS\_program\_map\_section 是当前可用的；当置‘0’表示所发送的 TS\_program\_map\_section 不可用，下一个 TS\_program\_map\_section 将为有效。

**section\_number** 8 位字段，值总为 0x00。

**last\_section\_number** 8 位字段，值总为 0x00。

**PCR\_PID** 13 位字段。表示包括由 program\_number 指定的程序中包含有效 PCR 字段的传送流分组的 PID，如果一私用流的程序定义无 PCR 与之相连，则此字段应为 0x1FFF。

**program\_info\_length** 12 位字段，头两位应为‘00’，规定在此字段之后 descriptor 的字节数。

**stream\_type** 8 位字段，规定有由 elementary\_PID 指定的 PID 值的分组中原始流或有效负载的类型 stream\_type 的值在表 I -2-35 中规定。

**elementary\_PID** 13 位字段，指定带有相关原始流或有效负载的传送流分段的 PID。

**ES\_info\_length** 12 位字段，头两位应为‘00’，规定在此字段之后相关原始流 descriptor 的字节数。

**CRC\_32** 32 位字段，在处理完整个传送流程序映射分段之后，使得附录 I -B 中解码器寄存器输出为零的 CRC 值。

#### I -2. 4. 4. 10 私用分段的语法

当私用数据在一具有在程序关联表中设计成为一个程序映射表 PID 的 PID 值的传送流分组中发送时，private\_section 的使用是必须的。此分段使得数据能以最小结构发送，以便 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 解码器能分析此数据流。此分段有两种使用方法：如果 section\_syntax\_indicator 为‘1’，则应使用所有表通用的整体结构；若为‘0’则仅有‘table\_id’到‘private\_section\_length’的字段符合通用的结构语法和语义。余下的私用分段将采用用户定义的任何形式，如表 I -2-29 所示。附录 I -C 中有关于此语法扩充使用的例子。

一个私用表可由几个 private\_section 构成，都有相同的 table\_id。见表 I -2-29。

表 I -2-29 私用分段

语 法	位数	缩写
private_section()		
<b>table_id</b>	<b>8</b>	<b>uimsbf</b>
<b>section_syntax_indicator</b>	<b>1</b>	<b>bslbf</b>
<b>private_indicator</b>	<b>1</b>	<b>bslbf</b>
<b>reserved</b>	<b>2</b>	<b>bslbf</b>
<b>private_section_length</b>	<b>12</b>	<b>uimsbf</b>

续表

语 法	位数	缩写
if(section_syntax_indicator == '0') {		
for(i=0;i<N;i++) {		
<b>private_data_byte</b>	<b>8</b>	<b>bslbf</b>
}		
}		
else {		
<b>table_id_extension</b>	<b>16</b>	<b>uimsbf</b>
<b>reserved</b>	<b>2</b>	<b>bslbf</b>
<b>version_number</b>	<b>5</b>	<b>uimsbf</b>
<b>current_next_indicator</b>	<b>1</b>	<b>bslbf</b>
<b>section_number</b>	<b>8</b>	<b>uimsbf</b>
<b>last_section_number</b>	<b>8</b>	<b>uimsbf</b>
for(i=0;i<private_section_length-9;i++) {		
<b>private_data_byte</b>	<b>8</b>	<b>bslbf</b>
}		
<b>CRC_32</b>	<b>32</b>	<b>rpchof</b>
}		
}		

#### I - 2. 4. 4. 11 私用分段中各字段的语义定义

**table\_id** 8位字段,其值识别此分段所属的私用表。一些值为保留(表 I-2-26)。

**section\_syntax\_indicator** 1 位指示,当置‘1’表示此私用分段采取一般的分段语法(在 private\_section\_length 字段之后的部分);置‘0’表示在 private\_section\_length 字段之后即为 private\_data\_byte。

**private\_indicator** 1位由用户定义的标志,ITU-T|ISO/IEC将来也不会规定它。

**private\_section\_length** 12 位字段, 规定在私用部分中, 紧跟在此字段之后直到 private\_section 结尾的剩余字节数。

**private\_data\_byte** 此字段由用户定义,将来 ITU-T | ISO/IEC 也不会规定。

**table\_id\_extension** 16位字段，其使用和取值由用户定义。

**version\_number** 5 位字段, private\_section 的版本号。version\_number 将随着 private\_section 中所带信息的每一次改变而加 1, 当到达 31 后, 又回复到 0。当 current\_next\_indicator 为 ‘0’ , 则 version\_number 为下一个具有相同 table\_id 和 section\_number 的可用 private\_section 的版本号。

**current\_next\_indicator** 1位字段,置‘1’表示所发送的 private\_section 是当前可用的。当此字段为‘1’时,version\_number 是当前可用 private\_section 的版本号;若此位为

‘0’，则所发送的 private\_section 不可用，下一个具有相同 section\_number 和 table\_id 的 private\_section 将为有效。

**section\_number** 8 位字段，表示 private\_section 的号码。私用表中第一分段的 section\_number 应为 0x00。section\_number 将随私用表中每一附加的分段而增 1。

**last\_section\_number** 8 位字段，规定此分段所属私用表的最后一分段的号码（即具有最高 section\_number 的分段的号码）。

**CRC\_32** 32 位字段，在处理完整个私用分段之后，使附录 I-B 解码器的寄存器输出为零的 CRC 值。

## I -2. 5 程序流比特流要求

### I -2. 5. 1 程序流的编码结构和参数

程序流编码层允许一个或多个原始流被合并成一单独的流。每个原始流的数据被多路复用和编码，包括那些允许原始流同步重新显示的信息。

程序流由一道程序中的一个或多个原始流多路复用而组成。每个原始流由访问单元（即显示单元的编码表示）组成，视频原始流的显示单元是一幅图象，相应的访问单元包括此图象的所有编码数据。在包含一组图象的第一幅编码图象的访问单元中，也包括此组图象的所有前缀数据，如本推荐|国际标准第Ⅱ部分中定义，并以 group\_start\_code 开头。包含一序列首部之后第一幅编码图象的访问单元，如本推荐|国际标准第Ⅱ部分所定义的那样，也包括此序列首部，sequence\_end\_code 字段则在含有一序列中的最后一幅编码图象的访问单元中（第Ⅱ部分中有 sequence\_end\_code 的定义），音频原始流的显示单元则是相应于一音频帧中采样的一采样集。（ISO/IEC 13818-3 中有音频帧的定义。）

原始流中的数据存储在 PES 分组中。PES 分组由 PES 分组首部及其后的分组数据组成。

PES 分组首部以一 32 位开始码字开始，此码字同时标明此分组数据所属流（见表 I-2-18）。PES 分组首部中可能包含针对在此分组中开始的第一个访问单元的解码和/或显示时间标签（DTS 和 PTS），PES 分组首部中也包含许多用来打开一系列可选字段的标志，分组数据包含来自一原始流的可变数目的连续字节。

在程序流中，PES 分组被组织成组。组以一组头开始，并跟有零个或多个 PES 分组。组头以一个 32 位开始码字开始，用来存储时序和比特率信息。

程序流以一可选择重复的系统首部开始，此系统首部带有此流中定义的系统参数的总结信息。

此标准未规定可作为条件访问系统部分使用的编码数据，然而本标准提供的机制使得程序服务提供者能为解码处理传送和识别这种数据，同时也能正确地参考由此标准所规定的数据。

### I -2. 5. 2 程序流系统目标解码器

在 I -2. 5. 4 节中说明的程序流语义和对这些语义的限制，要求对解码事件以及这些

事件发生的时间严格定义,所需要的定义在本推荐|国际标准中通过一个称为程序流系统目标解码器(P-STD)的虚拟解码器来说明。

P-STD 是一个概念上的模型,用来精确定义这些术语,并且在构造程序流的期间建立解码过程的模型,P-STD 仅为此目的而定义。P-STD 的结构和时序都不妨碍具有不同结构或时序表的解码器进行不间断的同步程序流显示。

下面的注释用来描述程序流系统目标解码器,并在图 2-7 中有部分说明。

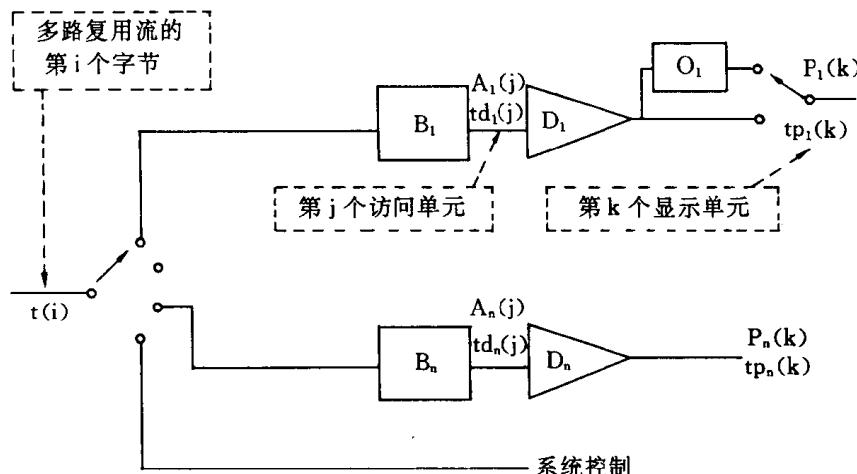


图 I -2-7 程序流系统目标解码器的表示法

$i, i'$	程序流中字节的序号,第一个字节序号为 0。
$j$	在原始流中访问单元的序号。
$k, k', k''$	为原始流中显示单元的序号。
$n$	为原始流的序号。
$t(i)$	表示程序流中第 $i$ 个字节进入系统目标解码器的时间(以秒为单位)。 $t(0)$ 值是一个任意常数。
$SCR(i)$	是编码在 SCR 字段中以 27MHz 系统时钟为单位的时间。其中 $i$ 为 SCR_base 字段的最后一个字节的序号。
$A_n(j)$	原始流 $n$ 中第 $j$ 个访问单元。注意,访问单元以解码顺序编号。
$td_n(j)$	以秒为单位,在原始流 $n$ 中第 $j$ 个访问单元系统目标解码器中的解码时间。
$P_n(k)$	原始流 $n$ 中第 $k$ 个显示单元。
$tp_n(k)$	以秒为单位的原始流 $n$ 中第 $k$ 个显示单元在系统目标解码器中的显示时间。
$t$	以秒为单位的时间。
$F_n(t)$	在时间 $t$ 系统目标解码器的输入缓冲区被原始流 $n$ 填充的字节数。
$B_n$	系统目标解码器中原始流 $n$ 的输入缓冲区。
$BS_n$	系统目标解码器中针对原始流 $n$ 的输入缓冲区的大小,以字节为单位。

- D<sub>n</sub> 原始流 n 的解码器。  
O<sub>n</sub> 视频原始流 n 的重排缓冲区。

### I -2. 5. 2. 1 系统时钟频率

P-STD 中的时序信息由本国际参考标准中定义的几个数据字段传送。这些字段的定义在 I -2. 5. 3. 3 节和 I -2. 4. 3. 6 节中,时序信息编码成系统时钟的采样值。

此系统时钟频率值以赫(Hz)为单位,且满足以下限制:

$$27\,000\,000 - 540 \leq \text{系统时钟频率} \leq 27,\,000,\,000 + 540$$

$$\text{系统时钟频率的时间变化率} \leq 75 \times 10^{-3} \text{Hz/s}.$$

“系统时钟频率”用在本标准中的几处,指满足这些要求的时钟频率。为了注释的方便,在有 SCR,PTS 或 DTS 出现的方程中,时间值都是秒( $300 \times 2^{33}$ /系统时钟频率)的整数倍。这是因为 SCR 时序信息是将系统频率的 1/300 编成 33 位码,再加上 9 位的余数,而 PTS,DTS 就是将系统时钟频率的 1/300 编成 33 位码。

### I -2. 5. 2. 2 程序流系统目标解码器的输入

从程序流来的数据进入系统目标解码器。第 i 个字节在 t(i) 时进入。通过对已编码在组头中的输入系统参考时钟(SCR)字段进行解码,此字节进入系统目标解码器的时间可从输入流恢复。在方程 I -2-19 中定义的 SCR 被编码成两部分:一部分以系统时钟频率的 1/300 为单位,称为 system\_clock\_reference\_base(方程 I -2-17),另一部分称 system\_clock\_reference\_ext(方程 I -2-18),以系统时钟频率为单位。以下这些字段被编码的值记作 SCR\_base(i) 和 SCR\_ext(i),SCR(i) 字段中编码的值为时间 t(i),i 指含有 SCR\_base 字段的最后一个位的字节。特别地:

$$\text{SCR\_base}(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{DIV } 300) \% 2^{33} \quad (\text{I -2-17})$$

$$\text{SCR\_ext}(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{DIV } 1) \% 300 \quad (\text{I -2-18})$$

$$\text{SCR}(i) = \text{SCR\_base}(i) \times 300 + \text{SCR\_ext}(i) \quad (\text{I -2-19})$$

方程 I -2-20 中给出的输入到达时间 t(i),对所有其它字节来说都必须由 SCR(i) 及数据到达率来构造,其中,每一组中的数据到达率是组头中 program\_mux\_rate 字段所表示的值。

$$t(i) = \frac{\text{SCR}(i')}{\text{system\_clock\_frequency}} + \frac{i - i'}{\text{program\_mux\_rate} \times 50} \quad (\text{I -2-20})$$

其中:

i' 包含组头中 system\_clock\_reference\_base 字段的最后一位的字节的序号。

i 组中任何一个字节的序号,包括组头中的字节。

SCR(i') 编码在 SCR\_base 和 SCR\_ext 字段以系统时钟为单位的时间。

program\_mux\_rate 在第 I -2. 5. 3. 3 节中定义的一字段。

在传送完一组的最后一个字节后,可能有一时间间隔,没有字节传送到 P-STD 的输入。

### I -2. 5. 2. 3 缓冲

PES 分组中原始流 n 的数据被送入流 n 的输入缓冲区 B<sub>n</sub>。从系统目标解码器将字节 i 送入 B<sub>n</sub> 是瞬时完成的,因此字节 i 进入大小为 BS<sub>n</sub> 的流 n 的缓冲区时间为 t(i)。

像 SCR, DTS, PTS 和 packet\_length 字段这样出现在程序流的组头,系统首部或 PES 分组首部的字节,将不送入任何缓冲区,但可用于控制此系统。

输入缓冲区的大小 BS<sub>1</sub> 到 BS<sub>n</sub> 由语法中方程 I -2-15 和方程 I -2-16 的 P-STD 缓冲区大小的参数决定。

在解码时间 td<sub>n</sub>(j)时,所有已在输入缓冲区的访问单元(A<sub>n</sub>(j))的数据应立即被移出。对于 ITU-T Rec. H. 262|ISO/IEC 13818-2 视频流的解码时间 td<sub>n</sub>(j)可以从 ITU-T Rec. H. 262|ISO/IEC 13818-2 中附录 I -C 的 I -C. 9 到 I -C. 11 节中推出。确认为属于 ITU-T Rec. H. 262|ISO/IEC 13818-2 的视频访问单元的数据在 ITU-T Rec. H. 262|ISO/IEC 13818-2 的附录 I -C 的 I -C. 5 中定义。ISO/IEC 13818-3 的音频流访问单元由音频帧组成,解码时间由 ISO/IEC 13818-3 定义。一旦访问单元从缓冲区移出,则马上被解码成显示单元。

当视频序列扩展的 low-delay 标志为‘1’(见 ITU-T Rec. H. 262|ISO/IEC 13818-2, I -6. 2. 2. 3 节)时,VBV 缓冲区可能下溢。当 P-STD 原始流缓冲区 B<sub>n</sub> 在 td<sub>n</sub>(j)时刻被检查时,访问单元的数据可能没有全部出现在缓冲区 B<sub>n</sub> 中。如果发生这种情况,缓冲区应每隔两个字段周期检查一次,直到整个访问单元出现在缓冲区中。在此时整个访问单元以及一些可能的相关系统数据(如果有的话)应立即从 B<sub>n</sub> 缓冲区中移出,缓冲区 B<sub>n</sub> 的溢出不应发生。

VBV 缓冲区的下溢允许无限地连续发生,P-STD 解码器应尽早地将访问单元数据从缓冲区 B<sub>n</sub> 中移出,与上一段的时间和编码在比特流中的 DTS 或 PTS 值一致。注意,解码器可能不能重建由 DTS 和 PTS 指示的正确解码与显示时间,直到 VBV 缓冲区的下溢状态结束和在比特流中又出现 PTS 或 DTS。

### I -2. 5. 2. 4 PES 流

构造一个由连续的含有同一原始流数据的 PES 分组所组成的数据流是可能的,这种流称 PES 流,除了原始系统参考时钟(ESCR)用于 SCR 处外,PES 流的 PES-STD 模式和程序流是一致的。而多路解调器仅将数据送到一个原始流缓冲区。

PES-STD 模式中缓冲区大小 BS<sub>n</sub> 定义如下:

对视频,BS<sub>n</sub>=VBV<sub>max</sub>[p. l.] + BS<sub>oh</sub>。

BS<sub>oh</sub>=(1/750)s \* R<sub>max</sub>[p. l.],其中 VBV<sub>max</sub>[p. l.]为 VBV 大小的最大值,而 R<sub>max</sub>[p. l.]为每个框架、级别、层次的最大比特率,如本推荐|国际标准中第 II 部分表 II -8. 12 和表 II -8. 13 中所定义。BS<sub>oh</sub>被指定为 PES 分组首部的开销。

对音频,BS<sub>n</sub>=2848 字节。

### I -2. 5. 2. 5 解码与显示

解码和显示在程序流系统目标解码器中的定义与其在传送流系统目标解码器中的定义相同,请参考 I -2. 4. 2 节。

### I -2. 5. 3 程序流语法和语义的说明

下面的语法描述一个流的字节。

#### I -2. 5. 3. 1 程序流

表 I -2-30 程序流

语 法	位数	缩 写
MPEG2_program_stream() { do { pack() } while(nextbits() == pack_start_code) <b>MPEG_program_end_code</b> }	32	bslbf

#### I -2. 5. 3. 2 程序流各字段的语法定义

**MPEG\_program\_end\_code** 为位串‘0000 0000 0000 0000 0000 0001 1011 1001’(0x000001B9)，它结束程序流。

#### I -2. 5. 3. 3 程序流的组层

表 I -2-31 程序流组

语 法	位数	缩 写
pack() { pack_header() while(nextbits() == packet_start_code_prefix) { PES_packet() } }		

表 I -2-32 程序流组头部

语 法	位数	缩 写
pack_header() { <b>pack_start_code</b> '01' system_clock_reference_base[32..30] marker_bit system_clock_reference_base[29..15] marker_bit system_clock_reference_base[14..00]	32 2 3 1 15 1 15	bslbf bslbf bslbf bslbf bslbf bslbf bslbf

续表

语 法	位数	缩 写
<b>marker_bit</b>	1	<b>bslbf</b>
<b>system_clock_reference_extension</b>	9	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>program_mux_rate</b>	22	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>reserved</b>	5	<b>bslbf</b>
<b>pack_stuffing_length</b>	3	<b>uimsbf</b>
for( <i>i</i> =0; <i>i</i> < <b>pack_stuffing_length</b> ; <i>i</i> ++) { <b>stuffing_byte</b> } if(nextbits() == <b>system_header_start_code</b> ) { <b>system_header</b> () }	8	<b>bslbf</b>
}		

#### I -2. 5. 3. 4 程序流组中各字段的语义定义

**pack\_start\_code** 位串‘0000 0000 0000 0000 0000 0001 1011 1010’(0x000001BA), 标志组的开始。

**system\_clock\_reference\_base ; system\_clock\_reference\_extension** 定义在方程 I -2-19 中的 **system\_clock\_reference**(SCR) 是一 42 位字段, 被编码成两部分: 一部分以系统时钟频率的 1/300(90kHz)为单位, 称为 **system\_clock\_reference\_base**(方程 I -2-17), 为 33 位字段; 另一部分称 **system\_clock\_reference\_extension**(方程 I -2-18), 以系统时钟频率(27MHz)为单位的 9 位字段。SCR 表示在程序目标解码器输入 **system\_clock\_reference\_base** 包括最后一位的字节预计到达的时间。

**system\_clock\_reference\_extension** 字段表示在一个 90kHz 时钟周期开始后 27MHz 时钟的周期数。此字段的值在 0 到 299 之间, 当其值达到 299 后, **system\_clock\_reference\_extension** 将回复到 0, 与此同时 **system\_clock\_reference\_base** 字段的最低位 lsb 将加 1。

**marker\_bit** 1 位字段, 其值为‘1’。

**program\_mux\_rate** 22 位整数, 表示 P-STD 接收此字段所在组的程序流的速率。**program\_mux\_rate** 值以每秒 50 字节为单位。禁止 0 值。在本推荐 | 国际标准中 I -2. 5. 2 节 **program\_mux\_rate** 的值用来定义进入 P-STD 输入的字节的到达时间, 在 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 程序多路复用流中组和组可有不同的 **program\_mux\_rate** 值。

**pack\_stuffing\_length** 3 位整数, 规定在此字段之后填充字节的数目。

**stuffing\_byte** 8 位固定字节‘1111 1111’, 可被编码器插入以满足通道的要求。被解码器丢弃。在每个组头不能有多于 7 个填充字节出现。

### I -2. 5. 3. 5 系统首部

表 I -2-33 程序流系统首部

语 法	位数	缩 写
system_header()		
<b>system_header_start_code</b>	32	bslbf
<b>header_length</b>	16	uimsbf
<b>marker_bit</b>	1	bslbf
<b>rate_bound</b>	22	uimsbf
<b>marker_bit</b>	1	bslbf
<b>audio_bound</b>	6	uimsbf
<b>fixed_flag</b>	1	bslbf
<b>CSPS_flag</b>	1	bslbf
<b>system_audio_lock_flag</b>	1	bslbf
<b>system_video_lock_flag</b>	1	bslbf
<b>marker_bit</b>	1	bslbf
<b>video_bound</b>	5	uimsbf
<b>reserved_byte</b>	8	bslbf
while(nextbits() == '1') {		
<b>stream_id</b>	8	uimsbf
'11'	2	bslbf
<b>P-STD_buffer_bound_scale</b>	1	bslbf
<b>P-STD_buffer_size_bound</b>	13	uimsbf
}		
}		

### I -2. 5. 3. 6 系统首部各字段的语义定义

**system\_header\_start\_code** 位串 ‘0000 0000 0000 0000 0001 1011 1011’ (0x000001BB), 标志一系统首部的开始。

**header\_length** 16 位字段, 表示在此字段之后系统首部的字节数。注意, 未来此推荐国际标准的扩展可能扩充此系统首部。

**rate\_bound** 22 位字段, 为一大于等于程序流中所有组的 program\_mux\_rate 字段最大值的整数。可以被一解码器用来断言其是否可以对整个流进行解码。

**audio\_bound** 6 位字段。在 0 到 32 之间的整数, 大于或等于同时进行解码处理的程序流中音频流的最大数目。为了本章的目的, 如果 STD 缓冲器非空或被解码的访问单元

在 P-STD 模型中正被显示，则 ISO/IEC 13818-3 音频流的解码处理为激活状态。

**fixed\_flag** 1 位标志，置‘1’表示为固定比特率操作；若置‘0’表示可变比特率操作。

**CSPS\_flag** 1 位字段，若其值为‘1’表示程序流满足本推荐|国际标准 I -2.7.9 节定义的限制。

**system\_audio\_lock\_flag** 1 位字段，表示在音频采样率和 STD 中 system\_clock\_frequency 之间有一特定的常数比例关系。I -2.5.2 节中定义了 system\_clock\_frequency，而音频采样率在 ISO/IEC 13818-3 中说明。如果对所有程序流的音频基本流的所有显示单元来说，system\_clock\_frequency 和实际音频采样率的比例 SCASR 为常数，且等于在音频流中指示的正常采样率如下表中所规定的值时，则 system\_audio\_lock\_flag 只能置‘1’。

$$SCASR = \frac{\text{system\_clock\_frequency}}{\text{audio\_sample\_rate\_in\_the\_P-STD}} \quad (I -2-21)$$

注释，上式  $\frac{X}{Y}$  表示实数除法。

正常音频采样频率/kHz	16	32	22.05	44.1	24	48
SCASR	$\frac{27\ 000\ 000}{16\ 000}$	$\frac{27\ 000\ 000}{32\ 000}$	$\frac{27\ 000\ 000}{22\ 050}$	$\frac{27\ 000\ 000}{44\ 100}$	$\frac{27\ 000\ 000}{24\ 000}$	$\frac{27\ 000\ 000}{48\ 000}$

**system\_video\_lock\_flag** 1 位字段，表示在视频图象速率和系统目标解码器中的 system\_clock\_frequency 之间有特定的恒定比例关系，system\_clock\_frequency 在第 I -2.5.2 中定义，视频图象速率在本推荐|国际标准的第 I 部分中定义。如果对 ITU-T Rec. H. 222.0|ISO/IEC 13818-1 的程序中所有视频原始流的所有显示单元来说，其 system\_clock\_frequency 和实际视频图象速率的比值 SCPR 为常数，并等于视频流正常图象速率情况下、下表所指示的值时，system\_video\_lock\_flag 只能置‘1’。

$$SCPR = \frac{\text{system\_clock\_frequency}}{\text{picture\_rate\_in\_the\_P\_STD}} \quad (I -2-22)$$

正常的图象速率/Hz	23.976	24	25	29.97	30	50	59.94	60
SCPR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

SCRP 的值是严格的，正常速率是每秒 23.976, 29.97, 或 59.94 幅图象，实际的视频图象速率与之稍有差别。

**video\_bound** 5 位整数，取值从 0 到 16，大于等于在解码处理中同时激活的程序流的 ITU-T Rec. H. 262|ISO/IEC 13818-2 流的最大数目。为了本章的目的，如果 P-STD 缓冲区非空，或被解码的访问单元正在 P-STD 模型中出现，或重排缓冲区非空，则 ITU-T Rec. H. 262|ISO/IEC 13818-2 视频流的解码操作被激活。

**reserved\_byte** 由 ISO/IEC 为将来使用而保留的字节，除非 ITU-T|ISO/IEC 特别说明，其值为‘1111 1111’。

**stream\_id** 8 位字段，表示其后 P-STD\_buffer\_bound\_scale 和 P-STD\_buffer\_

`size_bound` 字段所指流的编码原始流的号码。如果 `stream_id` 等于 ‘1011 1000’，则其后的 `P-STD_buffer_bound_scale` 和 `P-STD_buffer_size_bound` 对应于程序流中所有音频流，如果 `stream_id` 等于 ‘1011 1001’，则其后的 `P-STD_buffer_bound_scale` 和 `P-STD_buffer_size_bound` 字段对应于程序流中所有视频流。如果 `stream_id` 取其它值，则应大于等于 ‘1011 1100’，且按表 I -2-18 解释为原始流的号码和流编码的参考。

程序流中每一个原始流的 `P-STD_buffer_bound_scale` 和 `P-STD_buffer_size_bound` 都应在每个系统首部通过这种机制严格的规范一次。

**P-STD\_buffer\_bound\_scale** 1 位字段，表示用于解释随后的 `P-STD_buffer_size_bound` 字段的幅度因子。如果先前 `stream_id` 表示一个音频流，则 `P-STD_buffer_bound_scale` 应取值 ‘0’。如果先前 `stream_id` 指示一个视频流，则 `P-STD_buffer_bound_scale` 应为 ‘1’。对所有其它流类型，`P-STD_buffer_bound_scale` 可为 ‘0’ 或 ‘1’。

**P-STD\_buffer\_size\_bound** 13 位无符号整数，大于等于程序流所有分组中流 n 的 `P-STD` 输入缓冲区大小  $BS_n$  的最大值。若 `P-STD_buffer_bound_scale` 为 ‘0’，则 `P-STD_buffer_size_bound` 以 128 字节为单位，若 `P-STD_buffer_bound_scale` 值为 ‘1’，则 `P-STD_buffer_size_bound` 以 1024 字节为单位，由此

```
if(P-STD_buffer_bound_scale == 0)
    BSn≤P-STD_buffer_size_bound×128;
else
    BSn≤P-STD_buffer_size_bound×1024;
```

#### I -2. 5. 3. 7 程序流分组层

程序流的分组层在 I -2. 4. 3. 6 节中由 PES 分组层定义。

#### I -2. 5. 3. 8 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 组头字段

如 I -0. 3 节所描述，传送流和程序流之间的转换靠 PES 分组进行是可能的，ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 `pack_header` 字段若出现在 PES 分组中，则包含了 `pack_header` 数据，该数据可能含有 `system_header` 数据。在从传送流到程序流的转换中，ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 组头字段从来自同一程序的 PID 的所有 PES 分组中除去，并被放在与数据流相关的 PES 分组首部的前面。`pack_header_field_flag` 字段应置为 ‘0’，表示 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 `pack_header` 字段不再出现在 PES 分组头部。结果是一程序流。

当从一程序流转换为传送流时，`pack_header` 数据，包括可能有的 `system_header` 数据，被从程序流中移出而置于 I -2. 4. 3. 6 节中定义的 PES 分组首部中。`pack_header_field_flag` 应被置为 ‘1’，表示 ITU-T Rec. H. 222. 0 | ISO/IEC 13818 组头字段出现在 PES 分组首部中。结果是可用于单道或多道程序传输的多路复用器的 PES 分组流。

#### I -2. 5. 4 程序流映射

程序流映射提供了关于程序流中原始流以及它们之间相互关系的描述。当在传送流中传送时，其结构不能被修改。当 `stream_id` 值为 0xBC 时，PSM 作为一个 PES 分组

出现。

本推荐|国际标准 I -2. 6 节有 descriptor()字段的定义。

#### I -2. 5. 4. 1 程序流映射的语法

表 I -2-34 程序流映射

语 法	位数	缩写
program_stream_map() {		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimsbf
program_stream_map_length	16	uimsbf
current_next_indicator	1	bslbf
reserved	2	bslbf
program_stream_map_version	5	uimsbf
reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimsbf
for(i=0;i<N;i++) {		
descriptor()		
}		
elementary_stream_map_length	16	uimsbf
for(i=0;i<N1;i++) {		
stream_type	8	uimsbf
elementary_stream_id	8	uimsbf
elementary_stream_info_length	16	uimsbf
for(i=0;i<N2;i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

#### I -2. 5. 4. 2 程序流映射中各字段的语义定义

**packet\_start\_code\_prefix** 24 位码,与其后的 map\_stream\_id 一起组成一分组开始码字,标志一个分组的开始。其值为位串‘0000 0000 0000 0000 0000 0001’[0x000001(16 进制)]。

**map\_stream\_id** 8 位字段,其值为 0xBC(16 进制)。

**program\_stream\_map\_length** 16 位字段,表示在此字段之后程序流映射的字节总数。此字段最大值为 1024(1K)。

**current\_next\_indicator** 1 位字段,置‘1’时表示所发送的程序流映射是当前可用的;置‘0’则表示所发送的程序流映射不可用,下一个将变为有效。

**program\_stream\_map\_version** 5 位字段,为整个程序流映射的版本号。此版本号将随着程序流映射定义的每一次改变而加 1。当其值达到 31 后,又回复到 0。若 current\_next\_indicator 为‘1’,则 program\_stream\_map\_version 为当前可用程序流映射的版本号;若 current\_next\_indicator 为‘0’,则 program\_stream\_map\_version 为下一个可用程序流映射的版本号。

**program\_stream\_info\_length** 16 位字段,表示在此字段之后 descriptor 的总长度。

**marker\_bit** 1 位字段,值为‘1’。

**elementary\_stream\_map\_length** 16 位字段,规定在此程序流映射中所有的原始流信息的总字节数。

**stream\_type** 8 位字段,按表 I -2-35 决定原始流的类型。

表 I -2-35 流类型取值

值	描述
0x00	ITU-T ISO/IEC 保留
0x01	ISO/IEC 11172 视频
0x02	ITU-T Rec. H. 262 ISO/IEC 13818-2 视频
0x03	ISO/IEC 11172 音频
0x04	ISO/IEC 13818-3 音频
0x05	ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 私用分段
0x06	含有私用数据的 ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 PEC 分组
0x07	ISO/IEC 13522 MHEG
0x08	ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 DSM CC
0x09	ITU-T Rec. H. 222. 0 ISO/IEC 13818-1/11172-1
0x0A—0x7F	ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 保留
0x80—0xFF	用户私用

**elementary\_stream\_id** 8 位字段,表示此原始流所在 PES 分组的 PES 分组首部中 stream\_id 字段的值。

**elementary\_stream\_info\_length** 16 位字段,表示在此字段之后描述子的字节数。

**CRC\_32** 32 位字段,在处理完整个程序流映射后,使得附录 I -B 中解码器的寄存器输出为零的 CRC 值。

## I -2. 5. 5 程序流目录

整个流的目录由 directory\_stream\_id 所指示的 PES 分组中所有目录数据组成,含有程序流目录的 PES 分组语法如下定义。

如本推荐|国际标准第Ⅰ部分和 ISO/IEC 11172 中所定义,视频流中的 I -图象可能

需要目录条目作参考。如果在一目录条目中被参考的 I - 图象之前为一无插入图象首部的序列首部，则目录条目应参考此序列首部第一个字节。如果在一目录条目中注释的 I - 图象之前为一组图象首部，它们没有插入图象首部且没有紧接在前面序列的首部，则此目录条目应参考此组图象首部的第一个字节。其它目录条目参考的图象应参考图象首部的第一个字节。

注意：我们建议对紧跟在序列首部之后的 I - 图象应在目录结构中注释，这样目录在每一个解码器可能完全重新启动的点都有一条目。

ISO/IEC 13818-3 中定义的音频流目录注释应为音频帧的同步字。

注意：建议被注释的访问单元之间不应有超过半秒以上的间隙。

访问单元应以和它们出现在比特流中的顺序相同的顺序包括在目录中。

#### I - 2.5.5.1 程序流目录的 PES 分组语法

表 I - 2-36 程序流目录的 PES 分组语法

语 法	位数	缩写
directory_PES_packet()		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
number_of_access_units	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
for(i=0;i<number_of_access_units;i++) {		
packet_stream_id	8	uimshf
PES_header_position_offset_sign	1	tcimsbf
PES_header_position_offset[43..30]	14	uimsbf
marker_bit	1	bslbf
PES_header_position_offset[29..15]	15	uimsbf

续表

语 法	位数	缩写
<b>marker_bit</b>	1	<b>bslbf</b>
<b>PES_header_position_offset[14..0]</b>	15	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>reference_offset</b>	16	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>reserved</b>	3	<b>bslbf</b>
<b>PTS[32..30]</b>	3	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>PTS[29..15]</b>	15	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>PTS[14..0]</b>	15	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>bytes_to_read[22..8]</b>	15	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>bytes_to_read[7..0]</b>	8	<b>uimsbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>intra_coded_indicator</b>	1	<b>bslbf</b>
<b>coding_parameters_indicator</b>	2	<b>bslbf</b>
<b>reserved</b>	4	<b>bslbf</b>
}		
}		

### I -2. 5. 5. 2 程序流目录中各字段的语义定义

**packet\_start\_code\_prefix** 24 位码字,与其后的 stream\_id 一起构成一分组开始码字,标志一分组的开始。packet\_start\_code\_prefix 为位串 ‘0000 0000 0000 0000 0000 0001’ [0x0000001(16 进制)]

**stream\_id** 8 位字段,值总为 0xFF (16 进制)

**PES\_packet\_length** 16 位字段,表示在此字段之后的程序流目录中的字节总数。

**number\_of\_access\_units** 15 位字段,在此目录 PES 分组中被注释的访问单元数。

**prev\_directory\_offset** 45 位无符号整数,表示包含程序流目录信息的前一个 PES 分组的开始码字的第一个字节的字节偏移地址。此偏移地址是相对于含有此 prev\_directory\_offset 字段的 PES 分组开始码字的第一个字节来说的。‘0’为保留值,表示没有前一个程序流目录信息。

**next\_directory\_offset** 45 位无符号数,表示含有程序流目录信息的下一个 PES 分组开始码字的第一个字节的字节偏移地址。此偏移地址是相对于含有此 next\_directory\_

**offset** 字段的 PES 分组开始码字的第一个字节来说的。‘0’值保留,表示没有下一个程序流目录信息。

**packet\_stream\_id** 8 位字段,是含有此目录条目所参考的访问单元的原始流的 stream\_id。

**PES\_header\_position\_offset\_sign** 1 位字段,为随后要描述的 PES\_header\_position\_offset 字段的算术符号。‘0’值表示 PES\_header\_position\_offset 为正偏移;‘1’值表示 PES\_header\_position\_offset 为负偏移。

**PES\_header\_position\_offset** 44 位无符号整数,表示含有被参考的访问单元的 PES 分组第一个字节的字节偏移地址。此偏移地址是相对于含有此 PES\_header\_position\_offset 字段的 PES 分组开始码字的第一个字节来说的。‘0’值保留,表示无访问单元被参考。

**reference\_offset** 16 位无符号整数,表示被参考的访问单元第一个字节的位置,以字节为单位,相对于含有被参考的访问单元第一个字节的 PES 分组首字节来说。

**PTS(presentation\_time\_stamp)** 33 位字段,表示被参考的访问单元的 PTS。PTS 字段的编码语义在 I -2. 4. 3. 6 节中描述。

**bytes\_to\_read** 23 位无符号整数,表示在 reference\_offset 所指字节之后的程序流中完全解码访问单元所需字节数。此数包括多路复用在系统层中的所有字节,其中包括一些包含其它流的信息。

**intra\_coded\_indicator** 1 位标志,置‘1’时表示被参考的访问单元不是预测编码。这是独立于其它解码参考、并为解码访问单元所需的。例如,对视频‘I’帧,此字段应编码为‘1’,而对‘P’,‘B’帧应为‘0’。对所有含有非 ITU-T Rec. H. 262|ISO/IEC 13818-2 视频流数据的 PES 分组,此字段值保留。

表 I -2-37 内部编码的指示器

取 值	含 义
0	非内部
1	内部

**coding\_parameters\_indicator** 2 位字段,指示解码被参考的访问单元所需编码参数的位置。例如,此字段可用来决定视频帧方程矩阵的位置。

表 I -2-38 编码参数指示器

值	含 义
00	所有编码参数被设置为默认值
01	所有编码参数都在此访问单元中设置,其中至少有一个非默认值
10	部分编码参数在此访问单元中设置
11	无编码参数在此访问单元中设置

## I -2. 6 程序流和原始流描述子(descriptor)

程序流和原始流描述子用来扩展程序和原始流定义的结构,所有的流描述子的格式都以 8 位标签值开始,此标签值之后为 8 位描述子长度和数据字段。

表 I -2-39 提供了 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 中定义的;ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 中保留的;用户可用的描述子标签值。TS 或 PS 列中的‘X’表示此描述子对传送流或程序流可用。注意,描述子中各字段的意义与其在什么流中使用有关。这在下面的描述子语义中规定。

### I -2. 6. 1 流描述子中各字段的语义定义

下面用于描述子的语义在 I -2. 6. 2 节到 I -2. 6. 29 节中定义。

表 I -2-39 流描述子

descriptor-tag	TS	PS	Identification
0	n/a	n/a	保留
1	n/a	n/a	保留
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	system_clock_descriptor
12	X	X	multiplex_buffer_utilization_descriptor
13	X	X	copyright_descriptor
14	X		最大比特率
15	X	X	私有数据指示器描述子
16~63	n/a	n/a	ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 保留
64~255	n/a	n/a	用户私用

**descriptor-tag** 8 位字段,标志每一个描述子。它的含义在表 I -2-39 中给出。有一些值有规定的含义,有一些值为将来的 ITU-T|ISO/IEC 使用而保留,余下的值可由用户定义。

**descriptor-length** 8 位字段,规定在此字段之后的描述子的字节数。

### I -2. 6. 2 视频流描述子

视频流描述子提供基本信息,用以识别在 ITU-T Rec. H. 262|ISO/IEC 13818-2 或

ISO/IEC 11172-2 中描述的视频原始流的编码参数。

表 I -2-40 视频流描述子

语 法	位数	缩写
video_stream_descriptor()		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
multiple_frame_rate_flag	1	bslbf
frame_rate_code	4	uimsbf
MPEG_2_flag	1	bslbf
constrained_parameter_flag	1	bslbf
still_picture_flag	1	bslbf
if(MPEG_2_flag == 1) {		
profile_and_level_indication	8	uimsbf
chroma_format	2	uimsbf
frame_rate_extension_flag	1	bslbf
reserved	5	bslbf
}		
}		

### I -2. 6. 3 视频流描述子中各字段的语义定义

**multiple\_frame\_rate\_flag** 1 位字段, 置‘1’时表示多种帧速率可能出现在视频流中; 当置‘0’时表示仅出现一种帧速率。

**frame\_rate\_code** 4 位字段, 在 ITU-T Rec. H. 262 | ISO/IEC 13818-2 I -6. 3. 3 节中定义。除在 multiple\_frame\_rate\_flag 为‘1’时所指示的一特殊帧速率外, 也允许在视频流中出现其它一些帧速率, 如表 I -2-41 所示。

表 I -2-41 帧速率码

编码为	也包括
23.976	
24.0	23.976
25.0	
29.97	23.976
30.0	23.976 24.0 29.97
50.0	25.0
59.94	23.976 29.97
60.0	23.976 24.0 29.97 30.0 59.94

**MPEG\_2\_flag** 1 位字段, 置‘1’时表示视频流中含有 ITU-T Rec. H. 262 | ISO/IEC 13818-2 数据, 若置‘0’表示视频流中含有 ISO/IEC 11172-2 数据。

**constrained\_parameter\_flag** 1 位字段,当 MPEG\_2\_flag 为‘0’时,和 ISO/IEC 11172-2 视频数据中 constrained\_parameter\_flag 字段有相同值;若 MPEG\_2\_flag 为‘1’,constrained\_parameter\_flag 置‘0’。

**still\_picture\_flag** 1 位字段,置‘1’时表示视频流中仅含静止图象;若置‘0’时视频流可含动画或静止图象数据。

**profile\_and\_level\_indication** 8 位字段,和视频流中 profile\_and\_level\_indication 字段相同。

**chroma\_format** 2 位字段,和 ITU-T Rec. H. 262|ISO/IEC 13818-2 视频流中 chroma\_format 字段相同。

**frame\_rate\_extension\_flag** 1 位标志,置‘1’时表示 frame\_rate\_extension\_n 和 frame\_rate\_extension\_d 字段中的任一个或两者在 ITU-T Rec. H. 262|ISO/IEC 13818-2 视频流中为非零。

#### I -2. 6. 4 音频流描述子

音频流描述子提供识别在 ISO/IEC 13818-3 或 ISO/IEC 11172-3 中描述的音频原始流的编码版本的基本信息,如表 I -2-42 所示。

表 I -2-42 音频流描述子

语 法	位数	缩写
audio_stream_descriptor() {		
<b>descriptor_tag</b>	8	uimsbf
<b>descriptor_length</b>	8	uimsbf
<b>free_format_flag</b>	1	bslbf
<b>ID</b>	1	bslbf
<b>layer</b>	2	bslbf
<b>reserved</b>	4	bslbf
}		

#### I -2. 6. 5 音频流描述子中各字段的语义定义

**free\_format\_flag** 1 位字段,置‘1’表示音频流数据中 bitrate\_index 为‘0000’;若置‘0’表示 bitrate\_index 非‘0000’。

**ID** 1 位字段,设置成与音频流中 ID 字段相同的值。

**layer** 2 位字段,设置成与音频流中 layer 字段相同的值。

#### I -2. 6. 6 体系描述子

体系描述子提供识别在 ITU-T Rec. H. 262|ISO/IEC 13818-2 和 ISO/IEC 13818-3 中描述的含有多路复用在多个流中的分级编码视频和音频数据的原始流的信息,如表 I -

2-43 所示。

表 I -2-43 体系描述子

语 法	位数	缩写
hierarchy_descriptor()		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	4	bslbf
hierarchy_type	4	uimsbf
reserved	2	bslbf
hierarchy_layer_index	6	uimsbf
reserved	2	bslbf
hierarchy_embedded_layer	6	uimsbf
reserved	2	bslbf
hierarchy_priority	6	uimsbf
}		

### I -2. 6. 7 体系描述子中各字段的语义定义

**hierarchy\_type** 相关的体系层次和它的体系嵌入层次之间的体系关系在表 I -2-44 中定义。

表 I -2-44 体系描述子字段值

值	描 述
0	保留
1	ITU-T Rec. H. 262 ISO/IEC 13818-2 空域分层
2	ITU-T Rec. H. 262 ISO/IEC 13818-2 SNR 分层
3	ITU-T Rec. H. 262 ISO/IEC 13818-2 时域分层
4	ITU-T Rec. H. 262 ISO/IEC 13818-2 数据分割
5	ISO/IEC 13818-3 扩展比特流
6—14	保留
15	基层

**hierarchy\_layer\_index** 6 位字段, 定义了在编码层次体系表中相关原始流的唯一序号。在单个程序定义中序号应唯一。

**hierarchy\_embedded\_layer\_index** 6 位字段, 定义了在解码和此体系描述子相关的原始流之前需被访问的原始流的分级体系表序号, 若 hierarchy\_type 值为 15(基本层)则此字段无定义。

**hierarchy\_priority** 6 位字段, 表示传送分级体系中相关原始流的优先级。最重要的原始流(基本层)由 hierarchy\_priority 的最低值定义。

### I -2. 6. 8 注册描述子

注册描述子提供了一种唯一地、无歧义地识别私用数据格式的方法。

表 I -2-45 注册描述子

语 法	位数	缩写
registration_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
format_identifier	32	uimsbf
for(i=0;i<N;i++) {		
additional_identification_info	8	bslbf
}		
}		

### I -2. 6. 9 注册描述子中各字段的语义定义

**format\_identifier** 32 位值, 从 SC29 设计的注册许可中获得。

**additional\_identification\_info** 此字节的含义, 如果有, 由 format\_identifier 的代理人所定义, 且一旦定义就不能改变。

### I -2. 6. 10 数据流对准描述子

数据流对准描述子描述在相关原始流中对准的类型。如果 PES 分组首部的 data\_alignment\_indicator 为‘1’, 它应表明在此描述子中说明的对准类型。

表 I -2-46 数据流对准描述子

语 法	位数	缩写
data_stream_alignment_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
alignment_type	8	uimsbf
}		

### I -2. 6. 11 数据流对准描述子中各字段的语义定义

**alignment\_type** 表 I -2-47 的数据流对准描述子表描述了当 PES 分组首部中 da-

ta\_alignment\_indicator 为‘1’时的视频对准类型。

表 I -2-47 视频流对准取值

对准类型	描述
00	保留
01	切片、图片、GOP 或 SEQ
02	图片、GOP 或 SEQ
03	GOP 或 SEQ
04	SEQ
05—FF	保留

表 I -2-48 的数据流对准描述子表描述了当 PES 分组首部中 data\_alignment\_indicator 为‘1’时的音频对准类型。

表 I -2-48 音频流对准值

对准类型	描述
00	保留
01	音频帧
02—FF	保留

## I -2.6.12 目标背景栅格描述子

有的视频流解码时可能不需要全部占用显示区域(如显示器)。目标背景栅格描述子和视频窗口描述子的联合使用使得此视频窗口可在指定位置显示。目标背景栅格描述子用来描述投射到显示区域的单位象素的栅格，视频窗口描述子用来描述将被显示的视频显示单元第一个象素在栅格上的位置。

表 I -2-49 目标背景栅格描述子

语 法	位数	缩写
target_background_grid_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
horizontal_size	14	uimsbf
vertical_size	14	uimsbf
pel_aspect_ratio	4	uimsbf
}		

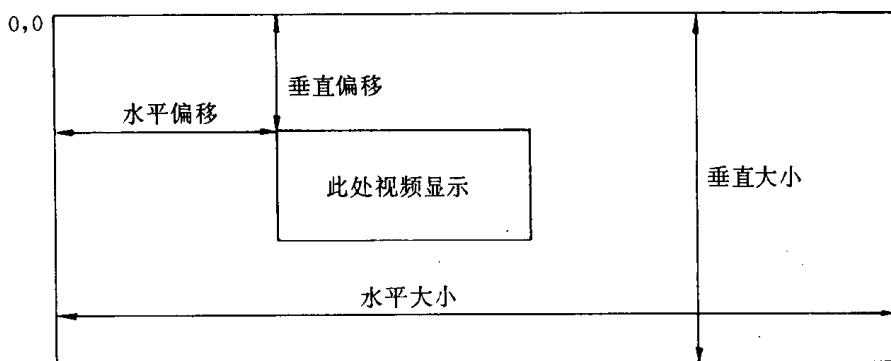


图 I -2-8 目标背景栅格描述子显示区域

### I -2. 6. 13 目标背景栅格描述子中各字段的语义定义

**horizontal\_size** 目标背景栅格的水平大小以象素为单位。

**vertical\_size** 目标背景栅格的垂直大小,以象素为单位。

**pel\_aspect\_ratio** 目标背景栅格值的象素纵横比如视频序列首部中定义。见本推荐国际标准的第 I 部分和 ISO 11172 的第二部分。

### I -2. 6. 14 视频窗口描述子

视频窗口描述子用来描述相关原始流窗口的特征,其值参考相同流的目标背景栅格描述子。见 I -2. 6. 12 节的目标背景栅格描述子。

表 I -2-50 视频窗口描述子

语 法	位数	缩写
video_window_descriptor() {     descriptor_tag     descriptor_length     horizontal_offset     vertical_offset     window_priority }	8 8 14 14 4	uimsbf uimsbf uimsbf uimsbf uimsbf

### I -2. 6. 15 视频窗口描述子中各字段的语义定义

**horizontal\_offset** 此值表示当前视频窗口左上象素在目标背景栅格描述子中定义的用于显示的目标背景栅格上的水平位置。

**vertical\_offset** 此值表示当前视频窗口的左上象素在目标背景栅格描述子中定义的用于显示的目标背景栅格上的垂直位置。

**window\_priority** 此值表示窗口怎样重叠在一起。0 值为最低优先级,15 为最高优先级。例如,优先级为 15 的窗口总是可见的。

### I -2. 6. 16 条件访问描述子

条件访问描述子用来描述宽系统条件访问管理信息(如 EMM)和原始特殊流信息(如 ECM),它可以被用在 TS 程序映射分段和程序流映射中。如果任一原始流被加密,含有此原始流的程序中必须有 CA 描述子。如果任何宽系统条件访问管理信息存在于传送流中,在相应的映射分段中必须有 CA 描述子。

当 CA 描述子出现在 TS\_program\_map\_section(table\_id=0x02) 中,CA\_PID 指向含有和访问控制信息(如 ECM)相关的程序的分组。它作为扩展的程序信息可用于整个程序。同样,它作为扩展的 ES 信息可用于相关原始流。规则同样适用于私用数据。

当 CA 描述子出现在 CA\_section 中(table\_id=0x01),CA\_PID 指向含有 system-wide 和/或访问控制管理信息(如 EMM)的分组。

含有条件访问信息的传送流分组的内容是私用的。

表 I -2-51 条件访问描述子

语 法	位数	缩写
CA_descriptor() { descriptor_tag descriptor_length CA_system_ID reserved CA_PID for(i=0;i<N;i++) { private_data_byte } }	8 8 16 3 13 8	uimsbf uimsbf uimsbf bslbf uimsbf uimsbf

### I -2. 6. 17 条件访问描述子中各字段的语义定义

**CA\_system\_ID** 16 位字段,表示适用于相关 ECM 和/或 EMM 流的 CA 系统类型。其编码是私自定义的,在 ITU-T|ISO/IEC 中无规定。

**CA\_PID** 13 位字段,表示传送流分组的 PID,此分组中含有由相应 CA\_system\_ID 指明的 CA 系统的 ECM 或 EMM 信息,由 CA\_PID 指明的分组的内容(ECM 或 EMM)由 CA\_PID 所在的上下文决定,即传送流中的程序映射表或 CA 表或程序流中的 stream\_id 字段。

### I -2. 6. 18 ISO 639 语言描述子

语言描述子用来描述相关原始流的语言。

表 I -2-52 ISO 639 语言描述子

语 法	位数	缩写
ISO_639_language_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++) {		
ISO_639_language_code	24	bslbf
}		
audio_type	8	bslbf
}		

### I -2. 6. 19 ISO 639 语言描述子中各字段的语义定义

**ISO\_639\_language\_code** 识别相应原始流使用的语言。ISO\_639\_language\_code 含有 ISO 639 第二部分中规定的 3 字符码。每个字符按 ISO 8859-1 编成 8 位码字并按顺序插入 24 位字段。在多种语言的音频流情况下,descriptor\_length 值是在相关音频流中带有的语言数的 3 倍多 1。ISO\_639\_language\_code 序列应反映音频流的内容。

**audio\_type** 8 位字段,按表 I -2-53 规定音频类型。

表 I -2-53 音频类型值

值	描 述
0x00	保留
0x01	清晰
0x02	听力削弱
0x03	visual_impaired_commentary
0x40-0xFF	保留

**clean effects** 此字段表示被参考的原始流中无语言。

**hearing impaired** 此字段表示被参考的原始流是为受损的听力准备的。

**visual impaired\_commentary** 此字段表示被参考的原始流是为视力受损的观众准备的。

### I -2. 6. 20 系统时钟描述子

此描述子传递有关用于产生时间标签的系统时钟的信息。

如果使用了外部参考时钟,external\_clock\_reference\_indicator 被置位。如果可用的话,解码器可选用此外部参考。

如果系统时钟比所要求的 20ppm 精确度更精确,时钟的精确度可通过将其编码于

`clock_accuracy` 字段中来传递,时钟频率精确度为:

$$\text{clock\_accuracy\_integer} \times 10^{\text{clock\_accuracy\_exponent}} \text{ ppm} \quad (\text{I - 2-23})$$

若 `clock_accuracy_integer=0`,则系统时钟精确度为 20 ppm.

当描述子的两部分都被使用时,时钟精确度附属于外部参考时钟。

表 I - 2-54 系统时钟描述子

语 法	位数	缩写
<code>system_clock_descriptor()</code> {		
<code>descriptor_tag</code>	8	<code>uimsbf</code>
<code>descriptor_length</code>	8	<code>uimsbf</code>
<code>external_clock_reference_indicator</code>	1	<code>bslbf</code>
<code>reserved</code>	1	<code>bslbf</code>
<code>clock_accuracy_integer</code>	6	<code>uimsbf</code>
<code>clock_accuracy_exponent</code>	3	<code>uimsbf</code>
<code>reserved</code>	5	<code>bslbf</code>
}		

### I - 2. 6. 21 系统时钟描述子中各字段的语义定义

**external\_clock\_reference\_indicator** 1 位指示,当置‘1’时表示系统时钟已从可在解码器中获得的外部参考频率中获得。

**clock\_accuracy\_integer** 6 位整数,与 `clock_accuracy_exponent` 一起,表示系统时钟的部分频率精确度。

**clock\_accuracy\_exponent** 3 位整数,与 `clock_accuracy_integer` 一起,表示系统时钟的部分频率精确度。

### I - 2. 6. 22 多路复用缓冲区使用描述子

表 I - 2-55 多路复用缓冲区使用描述子

语 法	位数	缩写
<code>multiplex_buffer_utilization_descriptor()</code>		
<code>descriptor_tag</code>	8	<code>uimsbf</code>
<code>descriptor_length</code>	8	<code>uimsbf</code>
<code>mdv_valid_flag</code>	1	<code>bslbf</code>
<code>multiplex_delay_variation</code>	15	<code>uimsbf</code>
<code>multiplex_strategy</code>	3	<code>uimsbf</code>
<code>reserved</code>	5	<code>bslbf</code>
}		

## I -2. 6. 23 多路复用缓冲区使用描述子中各字段的语义定义

**mdv\_valid\_flag** 值为‘1’表示 multiplex\_delay\_variation 字段有效。

**multiplex\_delay\_variation** 15 位字段,仅当 mdv\_valid\_flag 为‘1’时其值才有定义。当有效时 multiplex\_delay\_variation 以(27MHz/300)时钟周期为单位。这是流中所有分组的多个延迟的最大峰到峰变化。

**multiplex\_strategy** 3 位字段,指明了为解码器中多路解调器缓冲区 BS<sub>mux</sub>的目标使用而采取的多路复用器策略。

<b>multiplex_strategy</b>	策 略
0	未规定
1	<b>early</b>
2	<b>late</b>
3	<b>middle</b>
4—7	保留

**early** 多路复用器目标应尽早发送字节,只要 B<sub>n</sub> 缓冲区的大小 BS<sub>n</sub> 允许。因此 BS<sub>n</sub> 中 BS<sub>mux</sub>部分几乎完全使用。multiplex\_delay\_variation 如果有效,表示相对于尽早发送目标的字节发送的最大延迟。

**late** 多路复用器目标尽可能迟地发送字节。因此 BS<sub>n</sub> 的 BS<sub>mux</sub>部分将极少使用。multiplex\_delay\_variation 字段如果有效,表示在目标迟发送时间之前一字节被发送的最大提前时间量。

**middle** 多路复用器目标将在‘early’和‘late’之间的速度发送字节。multiplex\_delay\_variation 字段如果有效,表示对目标发送中间时间来说,字节发送时间的变动。

## I -2. 6. 24 版权描述子

版权描述子提供了标志音像作品的一种方法,版权描述子用于程序或程序中的原始流。

表 I -2-56 版权描述子

语 法	位数	缩写
copyright_descriptor() { <b>descriptor_tag</b> <b>descriptor_length</b> <b>copyright_identifier</b> for(i=0;i<N;i++) { <b>additional_copyright_info</b> } }	8 8 32 8	<b>uimsbf</b> <b>uimsbf</b> <b>uimsbf</b> <b>bslbf</b>

## I -2. 6. 25 版权描述子中各字段的语义定义

**copyright\_identifier** 32 位值,从 SC29 设计的注册权威中获得。

**additional\_copyright\_info** 这些字节若有含义,则由 **copyright\_identifier** 的代理人所定义,一旦定义,不能改变。

## I -2. 6. 26 最大比特率描述子

表 I -2-57 最大比特率描述子

语 法	位数	缩写
maximum_bitrate_descriptor() { <b>descriptor_tag</b> <b>descriptor_length</b> <b>reserved</b> <b>maximum_bitrate</b> }	8 8 2 22	uimsbf uimsbf bslbf uimsbf

## I -2. 6. 27 最大比特率描述子中各字段的语义定义

**maximum\_bitrate** 最大比特率为编码于 **maximum\_bitrate** 字段中的 22 位正整数。此整数表示最大比特率的上限,其中包括在此原始流或程序中将遇到的传送附加首部。最大比特率以每秒 50 字节表示,**maximum\_bitrate\_descriptor** 被包括在程度映射表(PMT)中,它作为扩展的程序信息可用于整个程序,它作为扩展的原始流信息可用于相关的原始流。

## I -2. 6. 28 私用数据指示器描述子

表 I -2-58 私用数据指示器描述子

语 法	位数	缩写
private_data_indicator_descriptor() { <b>descriptor_tag</b> <b>descriptor_length</b> <b>private_data_indicator</b> }	8 8 32	uimsbf uimsbf uimsbf

## I -2. 6. 29 私用数据指示器描述子中各字段的语义定义

**descriptor\_length** 应取值为 4。

**private\_data\_indicator** 取值为私用的,在 ITU-T | ISO/IEC 中无定义。

## I -2. 7 多路复用流语义的限制

### I -2. 7. 1 缓冲区管理

I -2. 5. 2 节中描述的程序流和 I -2. 4. 3. 6 节中定义的 PES 分组的构造以及  $t(i)$  的选择应使得程序系统目标解码器的大小为  $BS_1$  到  $BS_n$  的输入缓冲区既不上溢也不下溢,也就是

$$0 \leq F_n(t) \leq BS_n \quad \text{对所有 } t \text{ 和 } n$$

且在  $t=t(0)$  的前一瞬间  $F_n(t)=0$ 。 $F_n(t)$  是 P-STD 缓冲区  $B_n$  在某一瞬间填充的字节数。

此条件的一个例外是当“跳过的图象”发生时,P-STD 缓冲区  $B_n$  可能发生下溢,见 I -2. 5. 2. 3 节。

I -2. 4. 2 节描述的传送流的构造以及  $t(i)$  的选择应使传送系统目标解码器大小为  $BS_1$  到  $BS_n$  的输入缓冲区既不上溢也不下溢,同时在传送系统目标解码器中传送缓冲区  $TB_1$  到  $TB_n$  不上溢,也就是

$$0 \leq F_n(t) \leq BS_n \quad \text{对所有的 } t \text{ 和 } n$$

$$0 \leq F_{Tn}(t) \leq TBS_n \quad \text{对所有的 } t \text{ 和 } n$$

在  $t=t(0)$  之前瞬间  $F_n(t)=0$ 。

$F_n(t)$  是 T-STD 的主缓冲区  $B_n$  在某一瞬间填充的字节数, $F_{Tn}(t)$  是 T-STD 的传送缓冲区  $TB_n$  在某一瞬间填充的字节数。

此条件的一个例外是当“跳过的图象”发生时,缓冲区  $B_n$  可能发生下溢。见 I -2. 4. 2 节和 I -2. 5. 2 节。

系统主缓冲区  $B_{sys}$  可能下溢, $TB_{sys}$  不应上溢。

对所有程序流和传送流来说,除了对静止图象的视频数据外,由系统目标输入缓冲引起的延迟应小于等于一秒钟。

输入缓冲延时指的是一字节进入输入缓冲区的时间和其被解码时间的差值。

特别地:

$$td_n(j) - t(i) \leq 1s$$

对包含在访问单元 j 中的所有字节。

### I -2. 7. 2 系统参考时钟的编码频率

程序流的结构应使得在连续组中含有 system\_clock\_reference 字段最后 1 位的字节间的间隔小于等于 0.7s,也就是

$$|t(i) - t(i')| \leq 0.7s$$

其中  $i, i'$  为含有连续 system\_clock\_reference 字段最后 1 位的字节。

### I -2. 7. 3 程序参考时钟的编码频率

传送流的构造应使得含有每一程序的 PCR\_PID 的传送流分组中连续出现的 pro-

gram\_clock\_reference 字段最后一位的字节间间隔小于等于 0.1s,也就是

$$|t(i) - t(i')| \leq 0.1s$$

其中 i, i' 为具有同一程序的 PCR\_PID 传送流分组中含有连续的 program\_clock\_reference 字段最后一位的字节。

#### I -2. 7. 4 原始流系统参考时钟的编码频率

程序流和传送流的结构应使得当含有某给定原始流的 PES 分组中编入原始流参考时钟字段时,则含有连续的原始流参考时钟字段的最后一位的字节间时间间隔小于等于 0.7s。在 PES 流中要求 ESCR 的编码具有相同的间隔,因此

$$|t(i) - t(i')| \leq 0.7s$$

其中 i, i' 为含有连续的原始流参考时钟字段最后位的字节。

注意:原始流参考时钟字段的编码是可选的,它们并非必须被编码,但一旦被编码,必须有此限制。

#### I -2. 7. 5 显示时间标签的编码频率

程序流和传送流的结构应使得对每一个原始视频或音频流来说,编码的显示时间标签的最大差值为 0.7s,也就是

$$|tP_n(k) - tP_n(k'')| \leq 0.7s$$

对所有满足以下条件的 n, k, k'':

- ①  $P_n(k), P_n(k'')$ , 为编码显示时间标签所对应的显示单元;
- ② k 和 k'' 的选择是为了满足:不存在显示单元  $P_n(k')$ , 它带有显示时间标签且  $k < k' < k''$ ;
- ③ 在原始流 n 的  $P_n(k)$  和  $P_n(k'')$  之间不存在解码不连续。

#### I -2. 7. 6 时间标签的条件编码

对程序流或传送流中每一个原始流,显示时间标签(PTS)应被编码在原始流的第一个访问单元开始于其中的 PES 分组中。为了本节的目的,如果一幅图象的开始码字的第一个字节出现在 PES 分组数据中,则一个视频访问单元开始于一个 PES 分组(见本推荐 | 国际标准第 I 部分和 ISO 11172 的第二部分)。如果一音频帧同步字的第一个字节出现在 PES 分组数据中,则一个音频访问单元开始于一个 PES 分组(见 ISO/IEC 13818-3 和 ISO/IEC 11172-3)。

如果此访问单元的解码时间  $td_n(j)$  比系统时钟频率的容忍度给出的最大允许值还大,则一解码不连续存在于原始流 n 的访问单元  $A_n(j)$  的开始,对视频来说,这仅在一视频序列的开端允许发生。如果传送流或程序流中任何原始视频或音频流中存在解码不连续,则 DTS 或 PTS 的编码应针对于解码不连续之后的第一个  $A_n$ 。有以下两个例外:

- ① 当 low\_delay = '1' 时;
- ② 在 trick 模式操作中。

除以下情况外,PTS 可出现在任何 PES 分组首部。如果无访问单元开始于 PES 分组

数据，则 PTS 不应出现在 PES 分组首部。如果 PTS 出现在 PES 分组首部，它针对和开始于 PES 分组数据的第一个访问单元对应的显示单元。

解码时间标签(DTS)出现在 PES 分组首部，仅当下面两个条件满足时：

- ① PTS 出现在 PES 分组首部；
- ② 解码时间不同于显示时间。

### I -2. 7. 7 PES 分组首部中 P-STD\_buffer\_size 的编码频率

在程序流中，P-STD\_buffer\_scale 和 P-STD\_buffer\_size 字段应出现在每个原始流的第一个 PES 分组中，而且在此值的每次改变时再出现。也可以出现在其它任何 PES 分组中。

### I -2. 7. 8 程序流中系统首部的编码

在程序流中，系统首部可出现在任何组的组头之后。程序流的第一个组中应有系统首部。程序流中所有系统首部中的码值应相同。

### I -2. 7. 9 系统参数受限的程序流

如果一程序流满足本节中所说明的限制，就被称为“系统参数受限流”(CSPS)。程序流并没有受限于 CSPS 所规定的条件，CSPS 可由 I -2. 5. 3. 5 节中的系统首部中定义的 CSPS\_flag 来标志。CSPS 为所有可能程序流的一个子集。

#### 分组速率

在 CSPS 中，如果编码在 program\_mux\_rate 字段中的值小于或等于 4 500 000 比特/s，则分组到达 P-STD 输入的最大速率为每秒 300 分组。对于更高的比特率，CSPS 分组速率受限于 program\_mux\_rate 字段中编码值的一种线性关系。

特别地，对于程序流中所有分组 p

$$NP \leq (t_m(i') - t_m(i)) \times 300 \times \max\left[1, \frac{R_{\max}}{4.5 \times 10^6}\right] \quad (I - 2-24)$$

其中

$$R_{\max} = 8 \times 50 \times \text{rate\_bound}\left(\frac{\text{bit}}{\text{s}}\right) \quad (I - 2-25)$$

式中

NP 为相邻的 pack\_start\_codes 之间或在表 I -2-30 和 I -2. 5. 3. 2 节语义中定义的最后的 pack\_start\_code 和 MPEG\_program\_end\_code 之间的 pack\_start\_code\_prefix 和 system\_header\_start\_code 的数目；

$t_m(i)$  以秒为单位，编码于组 p 的 system\_clock\_reference 中的时间；

$t_m(i')$  以秒为单位，编码于组 p 之后第  $p+1$  组 system\_clock\_reference 的时间或当为程序流的最后一组时，含有 MPEG\_program\_end\_code 最后一位的字节到达时间。

#### 解码器缓冲区大小

在 CSPS 中系统目标解码器中的每个输入缓冲区的最大尺寸是受限的，对视频原始

流和音频原始流有着不同的限制。

CSPS 中视频原始流的限制如下：

$BS_n$  大小等于本推荐|国际标准第 I 部分中说明的视频缓冲校验器的大小加上附加缓冲区大小  $BS_{add}$ 。 $BS_{add}$  规定为：

$$BS_{add} \leq MAX [6 \times 1024, R_{v_{max}} \times 0.001] \text{ 字节}$$

CSPS 中音频原始流的限制如下：

$$BS_n \leq 4096 \text{ 字节}$$

$R_{v_{max}}$  为峰值视频比特率。

### I -2. 7. 10 传送流

锁定于传送流中的采样率。

在传送流中,音频采样速率和系统目标解码器的系统时钟频率之间有特定的常数比例关系,而且视频图象速率和系统时钟频率之间也有特定的比例关系。I -2. 4. 2 节中定义了系统时钟频率。视频图象速率在本推荐|国际标准第 I 部分或 ISO 11172 的第二部分中说明。音频采样率在 ISO/IEC 13818-3 或 ISO/IEC 11172-3 中定义。对传送流的所有音频原始流中的所有显示单元来说,系统时钟频率和实际音频采样率的比值,SCASR,为常数且等于在音频流中正常采样率情况下下表所指示的值:

$$SCASR = \frac{\text{system\_clock\_frequency}}{\text{audio\_sample\_rate\_in\_the\_T-STD}} \quad (I -2-26)$$

注释  $\frac{X}{Y}$  意味着实数除法。

正常的音频采样频率/kHz	16	32	22.05	44.1	24	48
SCASR	$\frac{27\ 000\ 000}{16\ 000}$	$\frac{27\ 000\ 000}{32\ 000}$	$\frac{27\ 000\ 000}{22\ 050}$	$\frac{27\ 000\ 000}{44\ 100}$	$\frac{27\ 000\ 000}{24\ 000}$	$\frac{27\ 000\ 000}{48\ 000}$

对传送流中所有视频原始流的所有显示单元来说,系统时钟频率和实际视频图象速率之间的比值,SCPR,为常数且等于在视频流中正常图象速率情况下下表所指示的值。

$$SCPR = \frac{\text{system\_clock\_frequency}}{\text{picture\_rate\_in\_the\_T-STD}} \quad (I -2-27)$$

正常图象速率/Hz	23.976	24	25	29.97	30	50	59.94	60
SCPR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

SCPR 的值是精确的,当正常的速率为每秒 23.976, 29.97 或 59.94 帧图象时,实际的图象速率与之略有差别。

## I -2. 8 与 ISO/IEC 11172 的兼容性

ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 的程序流定义与 ISO/IEC 11172-1 向前兼容。ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 定义的程序流解码器同时支持 ISO/IEC 11172-1 的解码。

## 附录 I -A 数字存储媒体的命令和控制[DSM CC]

(此附录为本推荐|国际标准的必需部分)

### I -A. 0 简 介

DSM CC 协议是一个特殊的应用协议,用来提供在数字存储媒体上管理 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流所需的基本控制函数和操作。此 DSM CC 是在网络/OS 层之上,应用层之下的一低层次协议。

DSM CC 在以下意义上是透明的:

- (1) 它独立于所使用的 DSM;
- (2) 它与 DSM 是本地的或远程的无关;
- (3) 它独立于 DSM CC 所相关的网络协议。

它独立于作用于 DSM 的各种各样的操作系统。

#### I -A. 0. 1 目的

许多 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 DSM 控制命令的应用要求访问通过一系列数字存储媒体存储于本地或远程的 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流。不同的 DSM 有着自己特殊的控制命令,为了访问不同的 DSM 上的 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流,用户必须懂得各种特殊 DSM 控制命令集,这为 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 或 ISO 11172-1 应用系统的界面设计带来困难。为克服此困难,独立于所使用的特殊的 DSM 的一套通用 DSM 控制命令集是必要的。

#### I -A. 0. 2 未来应用

除由当前 DSM 控制命令支持的即时应用外,基于 DSM 控制命令扩展的未来的应用可包括以下几种:

##### 点播电视

根据顾客所要求的视频程序由各种通信通道提供,顾客可从视频服务器中得到的程序表上选择一视频程序,应用于旅店、有线电视、教育机构、医院等部门。

##### 交互式视频服务

在这些应用中,用户通过频繁的反馈控制所存储的视频和音频的操作。这些服务包括视频基础上的游戏,用户控制的视频旅游,电子购物等等。

##### 视频网络

多种应用可能希望通过某种计算机网络交换所存储的音频和视频数据,用户可以通过接到他们终端上的视频网络使用 AV 信息,如电子出版等多媒体应用就是这类应用的

例子。

### I -A. 0. 3 优点

通过规定独立于 DSM 的 DSM 控制命令,高级用户能执行 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 解码而不需详细了解所使用的特殊 DSM 的细节操作。

DSM 控制命令的编码向高级用户保证 ISO/IEC 13818-1 比特流能独立于 DSM 和用户界面而以相同语义显示和存储,它们是控制 DSM 操作的基本命令。

### I -A. 0. 4 基本功能

#### I -A. 0. 4. 1 流选择

DSM CC 提供了选择执行后续操作的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流的方法,这些操作中包括一新比特流的产生。这些功能的参数包括:

- ① ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流的索引(此索引和对应应用中有意义的名字之间的映射关系不属当前 DSM CC 的范围);
- ② 模式(恢复/存储)。

#### I -A. 0. 4. 2 恢复

DSM CC 提供以下功能:

- ① 显示某确认的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流;
- ② 在给定显示时间显示;
- ③ 设定播放速度(正常或快速);
- ④ 设置播放时间片(直到某一特定显示时间,向前播放的比特流结束或倒放的开始或一停止命令的发出);
- ⑤ 设置方向(向前或向后);
- ⑥ 暂停;
- ⑦ 重新开始;
- ⑧ 改变比特流中的访问点;
- ⑨ 停止。

#### I -A. 0. 4. 3 存储

DSM CC 提供以下功能:

- ① 在一特定时间片段中存储有效比特流;
- ② 停止存储。

DSM CC 提供了在以 DSM 为基础的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 应用所要求的有用但受限的功能子集。可以乐观地期望随着以后的扩展,可增加重要的附加功能。

## I -A. 1 基本元素

### I -A. 1. 1 范围

此工作的范围就是发展一个国际标准来规定用来控制存有 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流的数字存储媒体的有效命令集。这些命令能独立于特殊的 DSM 而对数字存储媒体进行通常的远程控制，并可作用于任何存储在 DSM 上的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流。

### I -A. 1. 2 DSM CC 应用的概况

当前的 DSM CC 语法和语义覆盖了 DSM 应用的单个用户。用户系统可以恢复一个 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流，也可以(可选)生成一 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流。DSM 命令及应答信号发送所用的控制通道是在波段通道之外的，如图 I -A-1 所示如果波段通道之外的通道无法获得，则可以通过将 DSM CC 命令和应答信号插入 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流中来完成。

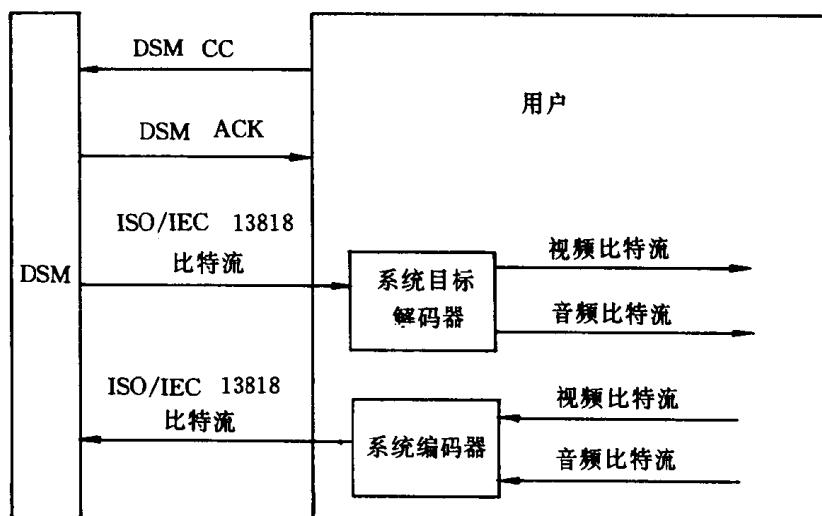


图 I -A-1 DSM CC 应用图例

### I -A. 1. 3 DSM CC 命令和应答的传送

根据在 I -A. 2. 2 到 I -A. 2. 9 节中定义的语法和语义定义，DSM CC 被编码在 DSM CC 比特流中，DSM CC 比特流可以作为一独立的比特流传送，也可以在一个 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 系统比特流中传送。

当 DSM CC 比特流被作为一单独流模式传送时，它和系统比特流及解码处理的关系如图 I -A-2 所示。此时，DSM CC 比特流未嵌入系统比特流中。此传送模式可用于 DSM 被直接连到 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 解码器中的应用中，也可以用于 DSM CC 比特流可由其它类型网络多路复用器控制和传送的应用中。

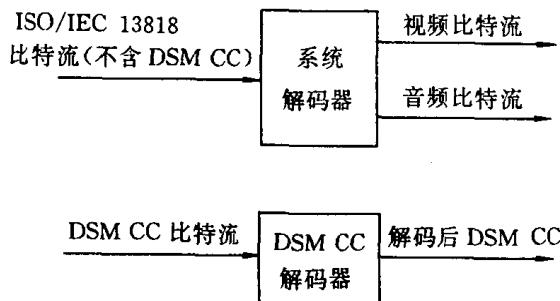


图 I - A - 2 DSM CC 做为一单独比特流被解码

在某些应用中,希望在 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 系统比特流中传送 DSM CC,这样 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 系统比特流的一些特性同样可应用于 DSM CC 比特流,此时,DSM CC 比特流被系统多路复用器并入系统比特流中。

在下面的过程中,DSM CC 比特流由系统编码器编码。首先,按本推荐 | 国际标准中 I - 2. 4. 3. 6 节中描述的语法,DSM CC 比特流被分为分组元素流(PES),接着 PES 按传送媒体的要求被多路复用进一程序流(PS)或传送流(TS),解码过程是编码的逆过程,如图 I - A - 3 中系统解码器的方框图所示。

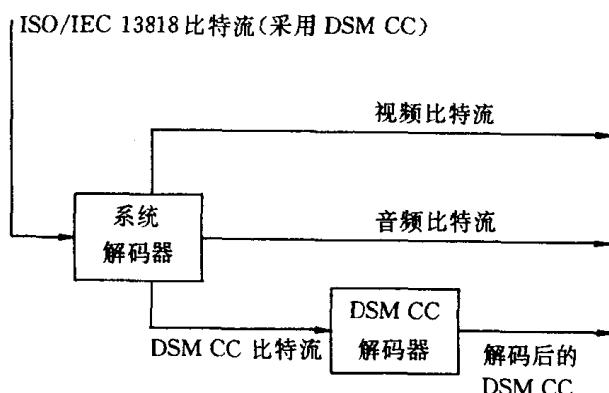


图 I - A - 3 DSM CC 做为系统比特流的一部分被解码

在图 I - A - 3 中,系统解码器的输出是视频流、音频流和/或 DSM CC 比特流,DSM CC 比特流可由其等于二进制码字‘1111 0010’的 stream\_id 来识别,在本国际标准的系统部分的表 I - 2 - 18 中定义,一旦 DSM CC 比特流被识别,后续的比特流将被直接送到 DSM CC 解码器,直到下一个非 DSM CC stream\_id 被检查出为止。

## I - A. 2 技术元素

### I - A. 2. 1 定义

**DSM CC** 数字存储媒体控制命令,由本推荐 | 国际标准规定,用来控制含有 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流的本地或远程数字存储媒体。

**DSM ACK** 从 DSM CC 命令接收器到命令发出者的应答信号。

**MPEG 比特流** 一个 ISO/IEC 11172-1 系统流、程序流或传送流。

**DSM CC 服务器** 一个本地或远程系统;用来存储和/或恢复 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流。

**随机访问点** ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流中的一点,具有以下性质:比特流中至少有一原始流,完全包含在比特流中的下一个访问单元‘N’可以无需参考以前的访问单元而解码,同时对比特流中每个原始流来说具有相同或迟一些的显示时间的所有访问单元全被包含在后续的比特流中,并可不访问随机访问点之前的任何信息而被系统目标解码器完全解码。存储在 DSM 中的比特流可能有某些随机访问点,DSM 的输出中可能有一些由 DSM 对存储内容处理而生成的附加随机访问点(例如,存储量化矩阵使在需要时随时能产生序列首部)。随机访问点都有一相关的 PTS,即实际的或暗示的访问单元‘N’的 PTS。

**当前操作的 PTS 值** 是在由当前选出的 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流中的 DSM 所提供的最近的访问单元之前,与最近的随机访问点相关的实际的或暗示的 PTS。如果此 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流未提供访问单元,则 DSM 不能在当前流中提供随机访问,而当前操作的 PTS 值就是 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流中第一个随机访问点。

**DSM CC 比特流** 满足第 I -A. 2. 2 节中语法的比特序列。

## I -A. 2. 2 DSM CC 语法规范

- 每个 DSM 控制命令应以 DSM\_start\_code 开始;
- 实际的控制命令或应答信息应紧跟 DSM\_start\_code 之后;
- 根据所收到的特殊命令,DSM 控制流接收器在所要求的操作开始或完成之后应提供一应答信息流;
- DSM 在任何时候都有义务提供一个合法的 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 系统比特流。这可能包括处理本国际标准系统部分定义的 trick 模式位。

表 I -A-1 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 DSM CC

语 法	位数	缩 写
DSM_CC() { DSM_CC_start_code command_id If(command_id == '01') { control() } else if(command_id == '02' { ack() } }	32 8	bslbf uimsbf

### I -A. 2. 3 DSM CC 语法规范中各字段的语义

**DSM\_start\_code** 32 位串 ‘0000 0000 0000 0000 0000 0001 1111 0010’  
(0x000001F2h)。标志了 DSM CC 比特流的开始。

**command\_id** 8 位无符号整数,识别一比特流是控制命令还是应答信息。见表 I -A-2 中的定义。

表 I -A-2 command\_id 字段的取值

值	command_id
0x00	禁止
0x01	控制
0x02	确认
0x03—0xFF	保留

### I -A. 2. 4 控制层

在 DSM CC 控制中设置标志位的限制:

- 在每一个 DSM 控制命令中,用于选择,播放和存储的标志位中最多有一个置‘1’,若无一个被置位则此命令被忽略;
- 在每一个恢复命令中,pause\_mode、resume\_mode、stop\_mode、play\_flag 和 jump\_flag 等标志位中最多有一个被置位,如无一个被置位,则此命令被忽略;
- 在每一个存储命令中,recode\_flag 和 stop\_mode 中最多有一个被挑选,若无一被挑选,则此命令被忽略。

表 I -A-3 DSM CC 控制

语 法	位数	缩 写
control(){		
<b>select_flag</b>	1	bslbf
<b>retrieval_flag</b>	1	bslbf
<b>storage_flag</b>	1	bslbf
<b>reserved</b>	12	bslbf
<b>marker_bit</b>	1	bslbf
<b>If(select_flag == '1')</b> {		
<b>bitstream_id[31..17]</b>	15	bslbf
<b>marker_bit</b>	1	bslbf
<b>bitstream_id[16..2]</b>	15	bslbf

续表

语 法	位数	缩 写
<b>marker_bit</b>	<b>1</b>	<b>bslbf</b>
<b>bitstream_id[1..0]</b>	<b>2</b>	<b>bslbf</b>
<b>select_mode</b>	<b>5</b>	<b>bslbf</b>
<b>marker_bit</b>	<b>1</b>	<b>bslbf</b>
}		
if(retrieve_flag == '1') {		
<b>jump_flag</b>	<b>1</b>	<b>bslbf</b>
<b>play_flag</b>	<b>1</b>	<b>bslbf</b>
<b>pause_mode</b>	<b>1</b>	<b>bslbf</b>
<b>resume_mode</b>	<b>1</b>	<b>bslbf</b>
<b>stop_mode</b>	<b>1</b>	<b>bslbf</b>
<b>reserved</b>	<b>10</b>	<b>bslbf</b>
<b>market_bit</b>	<b>1</b>	<b>bslbf</b>
if(jump_flag == '1') {		
<b>reserved</b>	<b>7</b>	<b>bslbf</b>
<b>direction_indicator</b>	<b>1</b>	<b>bslbf</b>
<b>time_code()</b>		
}		
if(play_flag == '1') {		
<b>speed_mode</b>	<b>1</b>	<b>bslbf</b>
<b>direction_indicator</b>	<b>1</b>	<b>bslbf</b>
<b>reserved</b>	<b>6</b>	<b>bslbf</b>
<b>time_code()</b>		
}		
}		
if(storage_flag == '1') {		
<b>reserved</b>	<b>6</b>	<b>bslbf</b>
<b>record_flag</b>	<b>1</b>	<b>bslbf</b>
<b>stop_mode</b>	<b>1</b>	<b>bslbf</b>
if(record_flag == '') {		
<b>time_code()</b>		
}		
}		
}		

## I -A. 2. 5 控制层中各字段的语义

**marker\_bit** 1位标志,总置‘1’以避免开始码字的竞争。

**reserved\_bits** 特定数目的比特,为 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 DSM 控制命令的将来扩展而保留。在所有 DSM CC 语法中应置‘0’。

**select\_flag** 1位标志,当置‘1’时,规定一比特流选择操作。

**retrieval\_flag** 1位标志,置‘1’时表示一特殊的恢复(播放)动作将发生。此操作从当前操作 PTS 值处开始。

**storage\_flag** 1位标志,当置‘1’时表示一存储操作将被执行。

**bitstream\_ID** 32位串,由语法中定义的三个字段合成一无符号整数来表示将被选中的 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流。将存储在 DSM 中的 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流的名字与由 bitstream\_ID 所代表的一系列数字建立唯一的映射关系是 DSM 服务器的责任。

**select\_mode** 5位无符号整数,表示要求哪一种模式的比特流操作。表 I -A-4 规定了定义的模式。

表 I -A-4 选择模式的取值

码	模 式
0	禁止
1	存储
2	恢复
3—31	保留

**storage** 下面的语法元素是比特流存储命令。

**retrieval** 下面的语法元素是比特流恢复命令。

**jump\_flag** 1位标志,表示将显示指针跳转到一个新的访问单元。新的 PTS 由对应于当前操作 PTS 值的相关 time\_code 所规定。此动作仅当此 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流处于“停止”模式时才有效。

**play\_flag** 1位标志,表示将一比特流显示一段时间。当此标志为‘1’时,速度、方向、显示时间为比特流中附加参数。显示从当前操作 PTS 值开始。

**pause\_mode** 1位码字,用来停止播放动作并将播放指针保持在当前操作 PTS 值。

**resume\_mode** 1位码字,规定从当前操作 PTS 值继续播放动作。仅当此比特流处于暂停(“pause”)状态时,重新开始(resume)才有意义,比特流将被置为正常速度的向前显示状态。

**stop\_mode** 1位码字,表示停止比特流传送。

**direction\_indicator** 1位码字,表示播放的方向,置‘1’代表向前显示,否则为向后显示。

**speed\_mode** 1位码字,规定速度标尺。若置‘1’表示为正常显示速度;若置‘0’表示

为快速显示(快放或快倒)速度。

**record\_flag** 1 位标志,规定从高级用户到 DSM 录制比特流的要求,可以是一段定时间内,也可直到接到“停止”命令时(不管哪个先到)。

## I -A. 2. 6 应答层

DSM CC 控制中设置标志位的限制:

在每个 DSM 询问流中,选择、恢复、存储和错误等询问只能选择一个(置为‘1’)。

表 I -A-5 DSM CC 确认

语 法	位数	缩写
ack()		
<b>select_ack</b>	1	<b>bslbf</b>
<b>retrieval_ack</b>	1	<b>bslbf</b>
<b>storage_ack</b>	1	<b>bslbf</b>
<b>error_ack</b>	1	<b>bslbf</b>
<b>reserved</b>	10	<b>bslbf</b>
<b>marker_bit</b>	1	<b>bslbf</b>
<b>cmd_status</b>	1	<b>bslbf</b>
If(cmd_status == '1' && (retrieval_ack == '1'    storage_ack == '1')) { time_code() } }		

## I -A. 2. 7 应答层中各字段的语义

**select\_ack** 1 位码字,置‘1’表示此 ack()命令是应答一个选择命令。

**playback\_ack** 1 位码字,置‘1’时表示此 ack()命令是应答一个播放命令。

**storage\_ack** 1 位码字,若置‘1’表示此 ack()命令是对存储命令的应答。

**error\_code** 1 位码字,置‘1’表示发现了 DSM 错误。目前已定义的错误有正在检索流时的 EOF(向前显示时遇到文件尾或倒放时遇到文件头)和正在存储流时的 Disk Full(磁盘已满)。若此位被设置,则 cmd\_status 无定义。无论怎样,当前流仍是被选择的。

**cmd\_status** 1 位标志,为 DSM 控制命令发出者提供 DSM 命令接收器的反应。若被置‘1’,说明命令被接收,否则命令被拒收。根据所接收到的命令的类型,有以下语义:

如果 select\_ack 和 cmd\_status 皆被置‘1’,表明 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流被选且服务器已准备好提供所选的操作模式。当前操作的 PTS 值被置为最近被选的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 比特流的第一个随机访问点。如果 cmd\_status 置‘0’,此操作失败,无比特流被选择。

如果 retrieval\_ack 和 cmd\_status 皆为‘1’，表示检索操作已根据所有的检索命令而初始化，当前操作的 PTS 点的位置由后续 time\_code 报告。

根据 infinite\_time\_flag != 1 的 play\_flag 命令，将发送第二个应答信息，这将确认显示操作已经通过达到 play\_flag 命令定义的时间而结束。

如果在一检索应答中 cmd\_status 置‘0’，表示此操作失败。可能的原因有：无效的 bitstream\_ID，跳到文件尾之后，或者以正常速度倒放之类不被支持的功能。

如果 storage\_ack 被置位，表示已因 record\_flag 命令而开始存储操作或因 stop\_mode 命令而停止存储。最后一被存储的访问单元的 PTS 由后续 time\_code 报告。

如果因到达 storage\_flag 命令所定义的时间片而结束了记录操作，另一个应答信息将发出，在纪录之后的当前操作的 PTS 值将被报告。

如果在一存储应答中 cmd\_status 置‘0’，此操作失败。导致失败的可能原因有：无效的 bitstream\_ID 或 DSM 无法存储数据。

### I -A. 2. 8 时间码(time code)

时间码的限制：

- 在观察到某一存取单元实际的或隐含的 PTS 值之后，由 time\_code 给定的时间间隔以前的操作将停止。因此，PTS 减去在操作开始时的当前操作 PTS 值再模  $2^{33}$  后将超过时间间隔；
- 在观察到某一存取单元实际或隐含的 PTS 值之后，由 time\_code 给定的时间间隔以后的操作将停止。因此，操作开始时的当前操作 PTS 值减去那个 PTS 值再模  $2^{33}$  后将超过时间间隔；
- 对所有控制层的命令来说，time\_code 被规定为和当前操作 PTS 值相关的一时间片；
- 对所有应答层的命令来说，time\_code 由当前操作的 PTS 值规定。

表 I -A-6 时间码

语 法	位数	缩写
time_code()		
reserved	7	bslbf
infinite_time_flag	1	bslbf
if(infinite_time_flag == '0') {		
reserved	4	bslbf
PTS[32..30]	3	bslbf
marker	1	bslbf
PTS[29..15]	15	bslbf
marker_bit	1	bslbf
PTS[14..0]	15	bslbf
marker_bit	1	bslbf
}		
}		

## I -A. 2.9 时间码中各字段的语义

**infinite\_time\_flag** 1位码字,当置‘1’时表示一无限时间区域,在一些应用中当某操作的时间周期无法事先定义时此标志置‘1’。

**PTS[32..0]** 比特流访问单元的显示时间标签,根据不同功能,此字段可以为一绝对值或在 90 kHz 系统时钟循环中的相对时间延迟。

## 附录 I - B CRC 解码器模型

(此附录是构成本推荐国际标准不可缺少的部分)

## I -B. 0 CRC 解码器模型

32位CRC解码器模型如图1-B-1所示。

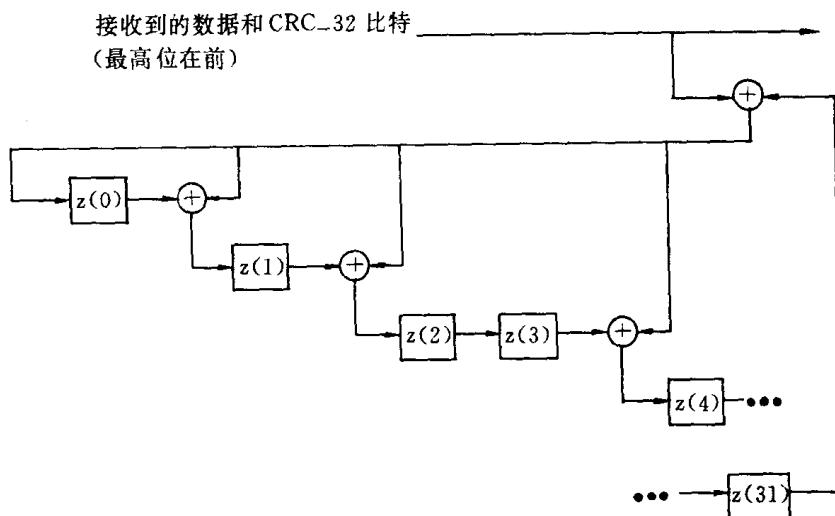


图 I-B-1 32 比特 CRC 解码器模型

32位CRC解码器在位层次操作,由14个加法器‘+’和32个延迟器件 $z(i)$ 组成。CRC解码器的输入被加到 $z(B1)$ 的输出,其结果作为 $z(0)$ 的输入,以及其余每个加法器的输入之一。这些加法器的另一个输入是 $z(i)$ 的输出,同时它们的输出与 $z(i+1)$ 的输入相连,其中 $i=0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22, 25$ ,见上图。

以下为 CRC 的计算多项式：

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

CRC 解码器的输入接收字节。每个字节一次一位地移入 CRC 解码器，最高位首先移入。例如字节 0x01(开始码前缀的最后一字节)，首先是 7 个‘0’进入 CRC 解码器，接着为一个‘1’。在 CRC 处理某一部分数据之前，每个延迟器件  $z(i)$  的输出初值为‘1’。在初始化之后，此部分的每一个字节被输入 CRC 解码器，包括 4 个 CRC-32 字节。在将 CRC-32 最后一字节最后一位移入解码器后，也就是和  $z(31)$  输出相加进入  $z(0)$  后，所有延迟器件的输出  $z(i)$  被读出。如果没有错误，所有的  $z(i)$  输出应为 0。在 CRC 编码器中 CRC-32 字段以保证相同值被编码。

## 附录 I-C 程序特殊信息

(此附录不是构成本推荐 | 国际标准必需部分)

### I-C. 0 传送流中程序特殊信息的说明

I-2.4.4 节中有关于程序特殊信息的正规语法、语义和正文,任何情况下都要求服从 I-2.4.4 节中的限制。此附录提供如何使用 PSI 功能的解释信息,并讨论实际应用的例子。

### I-C. 1 简介

为了多路解调和显示程序的目的,ITU-T Rec. H.222.0 | ISO/IEC 13818-1 提供了描述传送流分组内容的方法。编码规则通过程序特殊信息(PSI)来完成这项功能。本附录讨论 PSI 的使用。

PSI 可认为分属于四个表:

- ① 程序关联表(PAT)
- ② TS 程序映射表(PMT)
- ③ 网络信息表(NIT)
- ④ 条件访问表(CAT)

PAT,PMT,CAT 中的内容由本标准规定,NIT 为一私用表,但带有此表的传送流分组 PID 值在 PAT 中说明,它仍须满足本标准定义的分段结构。

### I-C. 2 功能机制

上面所列出的表都是概念上的,并不需要在解码器中以某种特殊形式重新生成。虽然这些结构可以被认为是单纯的表,但它们在被插入到传送流分组之前可能被分成好几部分。这种操作的语法支持是通过允许将表分为几段,并提供和传送流分组有效负载之间正规的映射方法来实现的。同时还有以相似的格式携带私有数据的方法。解码器中相同的基本处理同时用于 PSI 数据和私有数据,这有利于降低费用。关于 PSI 在传送流中位置的建议,见附录 I-D。

每一分段由下列元素组合唯一识别:

- (1) table\_id

8 位的 table\_id 标志分段所属的表。

table\_id 为 0x00 的分段属于程序关联表。

table\_id 为 0x01 的分段属于条件访问表。

table\_id 为 0x02 的分段属于 TS 程序映射表。

table\_id 的其它值可由用户分配用于私用目的。

可以用 table\_id 字段建立一滤波器来识别一新的分段是否属于我们感兴趣的表。

#### (2) table\_id\_extension

此 16 位字段存在于一分段的增长版本中。在程序相关表中用它来识别流的 transport\_stream\_id——实际上用户定义的标签,能使一个传送流在网络或交叉网络中与别的流相区别。在条件访问表中此字段目前没有意义,因此被标上“保留”字样,表示应编码为 0xFFFF。但在 ISO 本标准的下一版本时可能被定义。在 TS 程序映射分段中此字段包含 program\_number,以此识别所指分段中数据所属程序。table\_id\_extension 字段在某些情况下还可作为滤波器的结点。

#### (3) section\_number

section\_number 字段使得解码器能将一特定表的各分段按原来顺序重新合在一起。在此标准中不是必须按号码顺序传送各分段,但最好是这样,除非表中某些分段的传送比其它的更频繁,例如对于随机访问。

#### (4) version\_number

当 PSI 中描述的传送流特征发生改变时(例如增加新的程序,一给定程序的不同原始流组成),因为最近传送的标有“current”的分段必须有效,带有改变后的信息的新的 PSI 数据必须发送。解码器应该能识别最近收到的分段是否和它们已处理/存储的分段相同(这种情况下,新的分段可被丢弃)或不同。不同时,表示了结构的改变。它是通过发送一与以前的分段具有相同的 table\_id,table\_id\_extension 和 section\_number 的分段来实现的,但和以前不同的是 version\_number 为下一个值。

#### (5) current\_next\_indicator

搞清楚在比特流中的哪一点 PSI 有效是很重要的。每个分段可能被记为当前(current)有效,或下一刻(next)有效。这就使得可以在变化之前传送未来的结构,给解码器为变化而作准备的机会,然而不是必须传送分段的下一个版本,但如果已经传送,则必须为此分段的下一个正确版本。

## I-C. 3 从分段映射到传送流分组

分段被直接映射为传送流分组,也就是说没有先映射为 PES 分组。分段不一定在传送流分组的开端处开始(也许它们可能这样),因为传送流分组有效负载中第一个分段的开始点由 pointer\_field 指出。pointer\_fileld 的出现由 PSI 分组中 payload\_unit\_stat\_indicator 置‘1’表示(在非 PSI 分段中,此标志表示一 PES 分组在传送流分组中开始)。pointer\_field 指向传送流分组中第一个分段的开始,在传送流分组中决不会有第二个 pointer\_field,由于语法不允许在传送流分组的分段之间有空隙,因此其它任何分段的开始点可由计算第一个和以后的分段长度而获得。

在某一 PID 值的传送流分组中,一个分段必须在别的分段开始之前结束,这一点很

重要,否则不能识别数据属于哪一个分段首部。如果一分段在传送流分组之前结束,但此时并不适合开始另一分段,必须使用某种填充机制填满空间,填充就是在分组的剩余字节中填入 0xFF,一旦 0xFF 出现在分段的尾部,则传送流分段剩余字节必须填以‘0xFF’,使得解码器可以丢弃传送流分组剩余部分。填充也可以使用一般的 adaptation\_field 机制。

## I-C. 4 重复率和随机访问

在随机访问成为一值得考虑的问题的系统中,即使结构没有变化,也最好将 PSI 分段重复传送几次。因为在通常情况下,解码器需要 PSI 数据来识别传送流的内容以开始解码。ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 对 PSI 分段的重复率没有要求。虽然重复分段的增加有利于随机访问,但同时会引起 PSI 数据使用的比特率增加。如果程序映射为静态或半静态的,它们可以被存在解码器中以增快数据访问,而不是等待重传。对于所需的存储量和通道获取时间的影响,这两者之间的取舍由解码器制造者决定。

## I-C. 5 程序是什么

在本标准中程序的概念有精确定义,(I-2. 1 节)。“程序[系统]:具有共同时间基点的原始流的集合”,此概念的含义为:一道程序就是由具有共同 PCR 时钟的所有原始流组成,这样可以有效地在传送流中产生一虚拟通道。

注意,此定义和通常用在广播中的定义不同,在那里组成“程序”的原始流不仅仅有相同的时间基点,并且有相同的开始和结束时间。一系列的“广播程序”(在本附录中称为事件)可以用相同的 program\_number 顺序传送而产生传统的“TV 广播通道”(有时候称作服务)。

事件描述可以在私用分段中传送。

标志一程序的 program\_number 仅在传送流中有意义。program\_number 为 16 位无符号整数,这样允许 65535 个独立的程序共存于一传送流中(program\_number 值 0 为识别 NIT 而保留)。当解码器需处理几个传送流时(例如在电缆网络中),为了成功地对一道程序多路解调,解码器必须使用 transport\_stream\_id(为找到正确的多路复用器)和 program\_number(为找到多路复用器中正确的程序)。

传送流映射可通过可选的网络信息表完成。注意,网络信息表可存储在解码器中的非动态存储器中,以减少通道访问时间。在这种情况下,它的传送频率只要能及时满足支持解码器初始化设置操作就行了。NIT 的内容是私用的,但至少有最小分段结构。

## I-C. 6 program\_number 的分配

并不是在所有情况下将所有具有共同参考时钟的原始流组成一道程序都能很方便进行,有可能使用具有一共同 PCR 集的一多服务传送流。虽然广播者可能愿意将传送流在逻辑上分为几道程序,但它们的 PCR\_PID(参考时钟的位置)是相同的,这种将原始流分

为伪独立程序的方法有几种用途,下面为两个例子:

(1) 对应于不同市场的多语言传送 一视频流可能伴有不同语言的多个音频流,每个音频流最好带有一 ISO\_639\_language\_descriptor,以便进行选择正确的程序和音频。定义几个具有不同的 program\_number 的程序,其中所有的程序针对相同的视频流和 PCR\_PID,但有不同的音频 PID。然而将视频流和所有的音频流列为同一程序也是合理和可行的,只要不超过 1024 字节的分段大小限制。

(2) 很大的程序的定义 分段长度的最大限制为 1024 字节(包括分段首部和 CRC-32),这意味着单个程序的定义不能超过此长度。对大多数情况来说,即使每个原始流有几个描述子,此大小也足够了。然而人们必须处理可能超过此限制的高比特率系统的情况。这时通常可采用分离流的方法,使它们不必列在一块,一些原始流可以在多个程序中被参考,另一些原始流仅在一程序中被参考。

## I-C. 7 在一典型系统中 PSI 的使用

一通信系统,特别是在广播应用中,可由几个传送流组成。四个 PSI 数据结构中的每一个都可以出现在系统中的每一个传送流里。必须有一个完整的程序相关表用来列出传送流中所有程序,必须有一完整的 TS 程序映射表包含传送流中所有程序的定义。如果任何流被加密,则必须有一条件访问表列出相关 EMM 流(授权管理信息)。NIT 的出现为完全可选。

PSI 表通过以上描述的分段结构被映射为传送流分组,每个分组的首部有一 table\_id 字段,使得 PSI 表的分段和私用分段中的私用数据可以混合于具有相同 PID 值的传送流分组中,甚至还可以为同一传送流分组。然而具有相同 PID 的分组中,一个完整的分段必须在下一分段开始之前被传送。这种情况只可能发生在标有包含 TS 程序映射表或 NIT 的分组中,因为私用数据不能映射到 PAT 或 CAT 分组中。

要求所有的 PAT 分段被映射到 PID=0 的传送流分组,而 CA 分段被映射到 PID=1 的分组。PMT 分段可映射到用户选择 PID 值的分组中,作为每个程序关联表中的程序的 PMT\_PID 而列出。如果 NIT 存在的话,同样带有 NIT 的传送流分组 PID 也是用户选择的,但必须由 PAT 中“program\_number==0”的条目所指定。

任何 CA 参数流的内容都是私用的,但 EMM $\neq$ 0,ECM 必须和传送流分组一起传送以完成此标准。

私用数据表的发送可使用私用分段语法,这种表可用于在广播环境中描述一服务、一即将发生的事件、广播时间表和相关信息。

## I-C. 8 PSI 结构的关系

图 1-C-1 表示了四个 PSI 结构和传送流的关系的实例。可能有其它实例,但此图仅表示基本联系。

每个 PSI 表将在下面各段中描述。

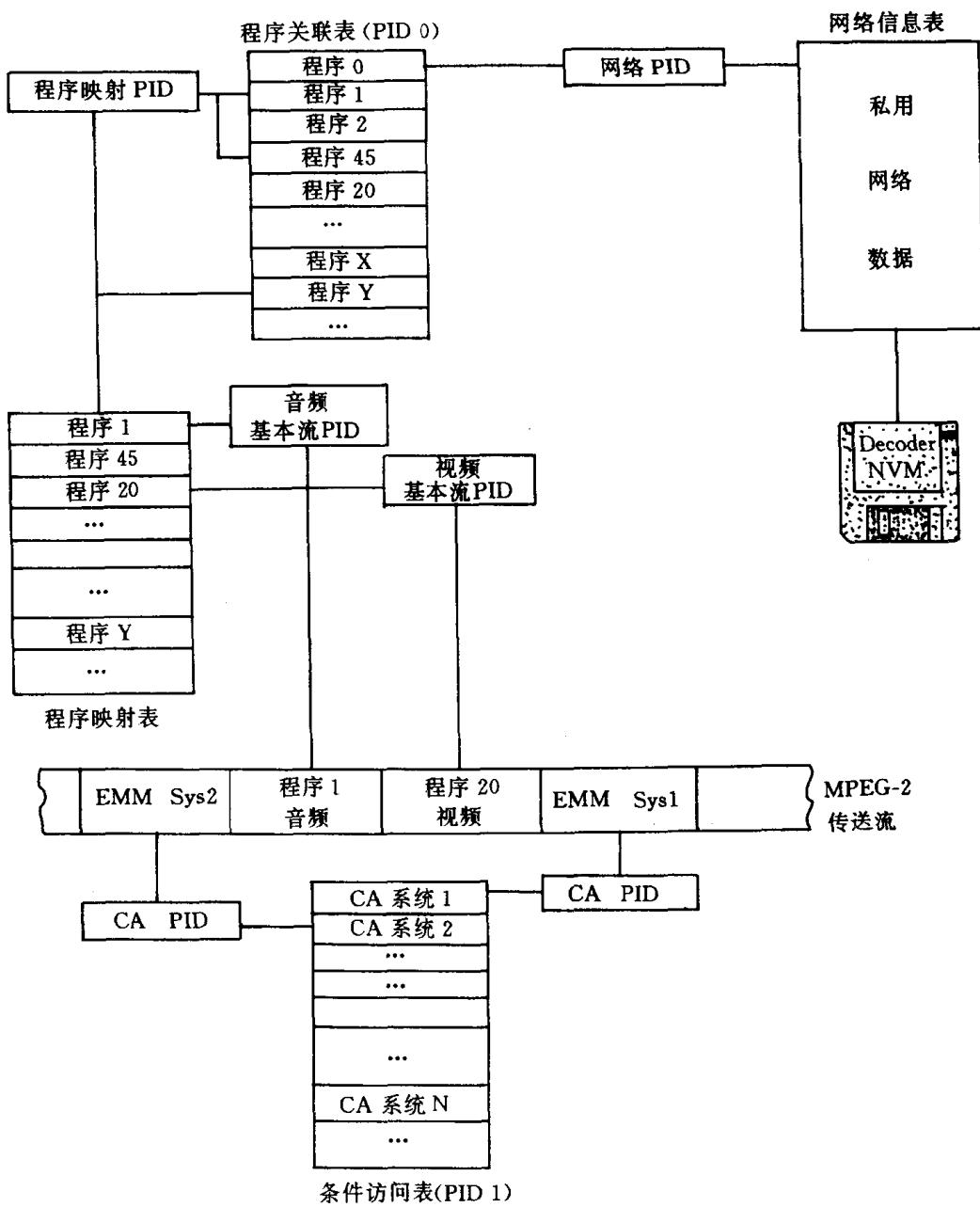


图 I-C-1 程序和网络映射关系

### I-C. 8.1 程序关联表

每个传送流必须有一完整有效的程序关联表，程序关联表给出了 `program_number` 和带有此程序定义的传送流分组 PID(`PMT_PID`)之间的对应。在映射为一传送流分组之前，PAT 可能被分成 255 个分段，每个分段含有整个 PAT 的一部分。这种分法，预计在错误条件下将使数据丢失最小化，也就是分组丢失或位错误可定位于更小的 PAT 分段，这样就允许其它分段被接收和正确解码。如果整个 PAT 信息置于一个分段中，则像 table-

*id* 中位的改变这样的错误,将导致整个 PAT 的丢失,然而,分段的长度只要不超 1024 字节的限制就行了。

程序 0 保留,用于规定网络 PID。为指向带有网络信息表的传送流分组的指针。

程序关联表经常是未经加密而直接传送的。

## I-C. 8. 2 程序映射表

程序映射表提供一 *program-number* 和组成此程序的原始流之间的映射。此表可出现在含有一个或多个私自选择的 PID 值的传送流分组中,在传送流分组中针对不同程序的 TS PMT 分段可能具有相同的 PID 值。

ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 要求程序识别至少有这些标志:程序号、PCR PID、原始流类型以及原始流 PID。程序或原始流的附加信息可用 *descriptor()* 结构传递。见 I-C. 9. 6 节。

私用数据也可在标志带有 TS 程序映射表分段的传送流分组中发送。利用私用分段完成。在私用分段中由应用决定 *version-number* 和 *current-next-indicator* 字段的值是针对于单个分段还是针对于作为一更大私用表部分的多个分段。

注意 1:带有程序映射表的传送流分组非加密传送。

注意 2:可以用 TS 程序映射分段中的私用描述子传送事件信息。

## I-C. 8. 3 条件访问表

条件访问表(CA)给出一个或多个 CA 之间的关系,并带有 EMM 流和所有特殊参数。

注意:带有 EMM 和 CA 参数的传送流分组的(私用)内容,如果出现的话,将被加密。

## I-C. 8. 4 网络信息表

NIT 的内容为私用,未由本标准规定。通常包含用户选择的服务和 *transport-stream-id*、通道频率、卫星脉冲发射机数、调制特性等的映射。

## I-C. 8. 5 Private-section()

*Private-section()* 可以两种基本形式出现:短版本(包含到 *section-length* 为止,并包含此字段)或长版本(出现到 *last-section-number* 为止的所有字段,并在私用数据字节之后有 CRC-32 字段)。

*Private-section()* 可以出现在 PID 标为 PMT\_PID 的传送流分组中,或出现在仅含有 *Private-section()* 的分组中,其中包括 PID 分配给 NIT 的那些分组。如果带有 *private-section()* 的 PID 的那些传送流分组被标志为带有 *private-section* 的 PID(*stream-type* 赋值 0x05),则在此 PID 值的传送流分组中仅能出现 *private-section*。这些分段可以为短的或长的类型。

## I-C. 8.6 描述子

本标准中定义了几个正规的描述子,也可以定义更多的私用描述子。所有的描述子都有一共同格式:{标志、长度、数据},任何私自定义的描述子也必须遵守此格式。私用描述子的数据部分是私自定义的。

当一个描述子(CA\_descriptor())出现在 TS PMT 分段中时,它被用来表示和原始流相关的 ECM 数据的位置(传送分组的 PID 值);若出现在 CA 分段中,则针对 EMM。

为了扩展可用的 Private\_descriptor 的数目,可使用以下机制:一私用描述子标志可私自定义成一成分描述子的结构。进一步详细规定将此私用描述子的私用数据字节的第一部分定义为子描述子(subdescriptor)。结构如表 I-C-1,I-C-2。

表 I-C-1 Composite\_descriptor

语法	位数	标志符
Composite_descriptor(){		
descriptor_tag(privatey defined)	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i < N;i++) {		
sub_descriptor()		
}		
}		

表 I-C-2 Sub\_descriptor

语法	位数	标志符
sub_descriptor(){		
sub_descriptor_tag	8	uimsbf
sub_descriptor_length	8	uimsbf
for(i=0;i < N;i++) {		
private_data_byte	8	uimsbf
}		
}		

## I-C. 9 带宽效用和信号获取时间

ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 比特流的任何实现都必须对 PSI 信息提出合理的带宽要求,而在随机访问为讨论重点的应用中,必须改善快速信号获取。本段分析这些问题并给出一些广播应用实例。

传送流的以分组为基础的特性,允许在多路复用数据中插入具有良好粒度的 PSI 信息。这提供了 PSI 结构和传送的充分灵活性。

一实际解码器的信号获取时间取决于许多因素,包括:FDM 音调回旋时间、多路解调

时间、序列首部和 I-帧产生速率以及加密关键字恢复和处理。

本段参考 I-2.4.4.4、I-2.4.4.9 节中 PSI 语法对比特率和信号获取时间的影响。假设条件访问表不需随每次程序的改变而动态接收，私用 EMM 流也有此假设，因为这些流中不含有用于原始流加密的快速变化 ECM 成分。

同样，在下面的讨论中 ECM 的获取和处理时间被忽略了。

表 I-C-3 给出了在一定传送流条件范围中带宽的使用值。此表的行为单个传送流中所含程序数，列为传送流中 PSI 信息的传送频率。

此频率对 PSI 结构来说是决定信号获取时间组成的关键。

两个带宽使用表（表 I-C-3, I-C-4）都假设仅提供最小的程序映射信息。这意味着 PID 值和流类型不需附加描述子来提供。实例中所有的程序由两个原始流组成，程序关联为 2 字节长，而最小程序映射为 26 字节长，同时还有附加的版本号、分段长度等，这相当于中等到最大长度的分段（几百字节到 1024 字节）中使用的整个 PSI 比特率的 1%—3%，这里将忽略之。

上面的假设允许 46 个程序关联映射到一程序关联表传送分组中（若不出现调整字段）。同样，7 个 TS 程序映射分段装入一单个传送分组。需注意对于简单的“减少/增加”，有可能使每一 PMT-PID 仅传送一个 TS 程序映射分段，然而这样将引起不愿看到的 PSI 使用比特率的增加。

表 I-C-3 程序关联表带宽使用值(bps)

PA 表信息的频率/ $s^{-1}$	传送流的程序数				
	1	5	10	32	128
1	1504	1504	1504	1504	4512
10	15040	15040	15040	15040	45120
25	37600	37600	37600	37600	112800
50	75200	75200	75200	75200	225600
100	150400	150400	150400	150400	451200

注释：因为 46 个程序关联分段装入一传送分组，表中数字直到最后一列才改变。

表 I-C-4 程序映射表带宽使用值(bps)

PM 表信息的频率/ $s^{-1}$	传送流中的程序数				
	1	5	10	32	128
1	1504	1504	3008	7520	28576
10	15040	15040	30080	75200	285760
25	37600	37600	75200	188000	714400
50	75200	75200	150400	376000	1428800
100	150400	150400	300800	601600	2857600

两个 PSI 表中取频率为  $25\text{ s}^{-1}$  时,产生信号获取时间的最坏情况,大约为 80ms。这种情况仅当所需的 PAT 数据“刚好错过”而当 PAT 被获取和解码后,所需 PMT 数据又“刚好错过”时发生。最坏情况获取时间的加倍是由 PAT 结构间接引起的附加级缺陷,此效应可通过协调相关的 PAT 和 PMT 分组的传送而减少。大概此方法的“减少/增加”重新多路复用操作的好处能提供补偿。

PSI 频率为  $25\text{ s}^{-1}$  时,可构造以下实例(所有实例都留下足够空间允许多种数据链、FEC,CA 和日常前缀):

#### **6MHz CATV 通道:**

5 个 5. 2Mbps 程序:	26. 5Mbps(包含传输开销)
整个 PSI 带宽:	75. 2kbps
CA 带宽:	500kbps
整个 ITU-T Rec. H. 2220. 0 ISO/IEC 13818-1 传输带宽:	27. 1Mbps
PSI 开销:0. 28%	

#### **OC-3 光纤通道(155Mbps):**

32 个 3. 9Mbps 程序:	127. 5Mbps(包含传输开销)
整个 PSI 带宽:	225. 6kbps
CA 带宽:	500kbps
整个 ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 传输带宽:	128. 2Mbps
PSI 开销:0. 18%	

#### **C-带卫星脉冲转发机:**

128 个 256kbps 音频程序:	33. 5Mbps(包含传输开销)
整个 PSI 带宽:	826. 4kbps
CA 带宽:	500kbps
整个 ITU-T Rec. H. 222. 0 ISO/IEC 13818-1 传输带宽:	34. 7Mbps
PSI 开销:2. 4%(如果每个程序仅用一个 PID,则实际会更低)	

正如所预料的,前缀在低速率服务中的比重增加,因为传送流中可能有更多的服务。然而,前缀并非在所有情况下都为多余,更高的 PSI 数据传送速率(高于  $25\text{ s}^{-1}$ )可用于减少通道获取时间的影响而仅适当地增加比特率要求。

## 附录 I-D ITU-T H. 222.0 | ISO/IEC 13818-1 系统时序模型和应用的内在要求

(此附录不是本推荐|国际标准的必需部分)

### I-D. 0 简介

ITU-T Rec. H. 262 | ISO/IEC 13818-2 系统说明包括一特殊的时序模型,用于采样、编码、编码缓冲、传送、接收、解码缓冲、解码、视频与音频联合数字显示。此模型直接体现在 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 数据流的语法说明和语义要求中。当一解码系统接收到一正确发送的允许比特流,且其时序模型与用来指导实现解码器的模型一致,则将产生正确同步的高质量的音频和视频输出,虽然没有什么标准的要求,解码器的实现方法足以产生高质量的显示输出。在某些应用中,数据不以正确的时序送入解码器中,也有可能产生所需的显示输出,但这种能力通常不能保证。此附录信息详细描述 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 系统时序模型,并给出一些适合特殊应用的解码器系统的实现建议。

#### I-D. 0.1 时序模型

ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 系统包含了一时序模型,其中所有的数字化图象和音频采样进入编码器后,经一恒定的端到端延迟后在解码器的输出分别显示。采样率,包括视频图象速率和音频采样速率在编码器和解码器中严格相等,此时序模型如图 I-D-1 所示。

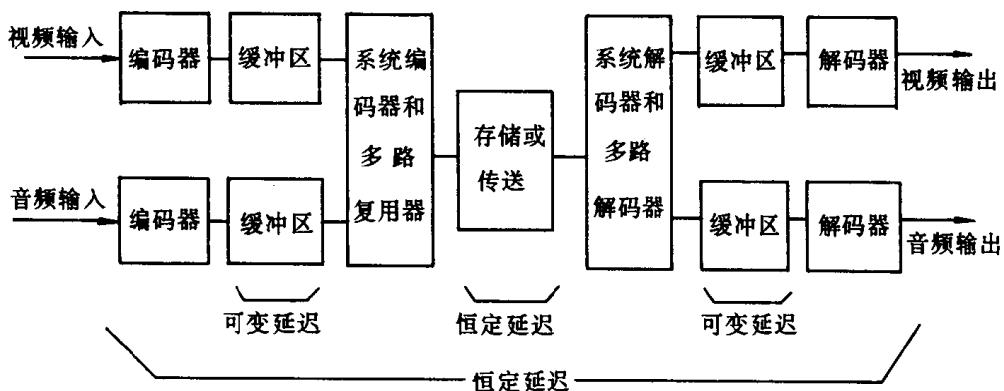


图 I-D-1 恒定延迟模型

如图中所表示的,在此模型中从编码器的输入到解码器的输出或显示之间的延时恒

定,虽然通过每个编码器、解码器缓冲区的延时是可变的。不仅仅是通过一原始流通道的缓冲区的延时可变,就是在音频和视频通道中单个的缓冲区延迟也是可变的,因此表示音频和视频信息的编码比特在相应流中的相对位置不能表示同步信息。编码的音频和视频的相关位置仅由系统目标解码器(STD)模型决定,因此解码器的缓冲区必须正确地操作。由此代表将被同步显示的声音和图象的编码音频视频可最大以一秒钟片段分离在编码比特流中,此一秒钟由 STD 模型允许的最大缓冲区延时所决定。

编码器中音频和视频采样率相互差别很大,它们之间是否有严格、固定的关系,取决于联合流为程序流还是传送流,取决于程序流中是否设置了 System\_audio\_locked 和 System\_video\_locked 标志。一组音频采样的持续时间(一音频显示单元)通常和一视频图像的持续时间不同。

编码器中有单一的共同系统时钟,此时钟用来产生指示音频和视频的正确显示和解码时序的时间标签,同时可用来指示在抽样间隙中系统时钟本身的瞬时值。指示音频和视频显示时间的时间标签称为显示时间标签(PTS);而用来指示解码时间的称解码时间标签(DTS);指示系统时钟值的称系统参考时钟(SCR)(在程序流中)或程序参考时钟(PCR)(在传送流中)。正是由于解码器中共同系统时钟的出现,使得时间标签由其产生。解码器中时钟的重新生成、时间标签的正确使用,为解码器中操作的正确同步提供便利。

编码器的实现不一定严格遵守此模型,然而由实际的编码器、程序系统、网络和一个或多个多路复用器所产生的数据流必须严格遵守此模型(数据的发送根据应用可有所偏差)。因此此附录中“编码器系统时钟”一词用来表示此模型中描述的实际共同系统时钟或可能被实现的等效功能。

因为通过整个系统的端到端延迟为常数,所以音频和视频显示将精确同步。系统比特流的构造使得当它被依此模型实现的解码器解码时,适宜大小的解码器缓冲区保证不发生上溢或下溢,除一些特殊的有意的下溢情况。

为了使解码器系统产生精确的延迟以保证整个端到端延迟为常数,解码器的系统时钟必须使其操作频率和绝对瞬时值与编码器的相应值匹配。传送解码器系统时钟所必须的信息编码于 SCR 或 PCR 中。下面解释此功能。

实现和此时序模型一致的解码器,它们以严格的恒定速率显示音频采样和视频图象(除一些有意的特殊编码外),它们的缓冲区和此模型一样操作。这种解码器在本附录中被称为精确时序解码器或产生精确时序输出的解码器。然而,在这种实现中音频和视频显示的同步并不一定精确,而且解码器缓冲区的动作可能不符合参考解码器模型。避免解码器缓冲区的上溢是非常重要的,因为上溢引起的数据丢失将对解码处理结果产生重大影响。本附录基本上包含这种精确时序解码器的操作以及实现这些解码器时可用的一些选项。

## I-D. 0. 2 音频和视频显示同步

在 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 系统数据的编码中有关系到视频图象和音频采样块的显示及解码的时钟标签。图象和块被称为“显示单元”,简写为 PU。那些代表 PU 的编码比特以及包含在 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 比特流中的比特集被称为“访问单元”,简写为 AU。音频访问单元简写为 AAU,视频访问单元简写为

VAU。在 ISO/IEC 13818-3 音频中“音频帧”这一术语根据上下文和 AAU 或 APU 有相同的含义。VPU 是一幅图象,VAU 则为一编码图象。

一些,但并不一定是全部的 AAU 和 VAU 与其 PTS 相关联。PTS 表示由与其 PTS 相应的 AU 解码而得的 PU 被显示于用户的时间。音频 PTS 和视频 PTS 都是被称为系统时间时钟(STC)的共同时间时钟的采样。根据包含在数据流中正确的音频和视频 PTS 值,根据在以共同 STC 形式表现的 PTS 所指示的时间被显示的音频和视频 PU,解码器系统实现了音频和视频显示的精确同步。STC 不是本国际标准的标准部分,在标准中相应的信息通过 System\_Clock\_Frequency 传送,但 STC 是用来解释此时序模型的重要且方便的元素,而且实现包含某种形式 STC 的编码器和解码器通常是可行的。

因为音频和视频 PU 通常具有非常不同的和本质上不相关的持续时间,所以需要 PTS 来传送音频和视频之间精确的相对时序,例如,有 1152 个样本的音频 PU 的采样率为每秒 44.100 采样,持续时间为 26.12ms;而帧速率为 29.97Hz 的视频 PU 的持续时间约为 33.76ms。通常 APU 和 VPU 的时域边界即使重合,也是极少的。分离的音频和视频 PTS 为指示音频和视频 PU 的精确时域关系提供信息,而不要求任何音频和视频 PU 的持续时间和间隔时间之间的特殊关系。

PTS 字段的值在系统目标解码器(STD)中定义。其中 STD 是对所有系统比特流的一基本标准限制。STD 是一理想解码器的数学模型,精确描述了所有输入、输出解码器缓冲区的比特的动作。而对比特流的基本语义限制就是 STD 缓冲区决不能上溢或下溢,除某些特定情况下发生的下溢外,在 STD 模型中的虚拟解码器通常和数据源严格同步,而且音频和视频的解码和显示也是严格同步的。在保持严格性和一致性的同时,STD 在解码器的物理实现上有所简化,为的是使其规范清楚明白,并有利于多种解码器实现的应用。尤其是在 STD 模型中,在解码器中对比特流的每一个操作都是瞬时完成的,除了比特在解码器缓冲中所花的时间外。在一实际的解码器系统中单个的音频和视频解码器并非瞬时操作,它们的延时在实现设计中必须予以考虑。例如,如果视频图象的解码严格按每隔  $1/P$  时间显示一图象进行,其中  $P$  为图象速率,且压缩的视频数据到达解码器的比特率为  $R$ ,则和每幅图象相关的比特完全移出的时间比 PTS 和 DTS 字段所指示的时间晚  $1/P$ ,因此视频解码器缓冲区必须比 STD 模型所规定的大  $R/P$ 。视频显示相对于 STD 同样也有延迟,PTS 需相应进行处理。既然视频被延迟,音频解码和显示也要延迟近似的时间,以提供正确的同步。解码器实现音频和视频的解码、显示延迟可通过当解码器使用 PTS 时加一常数。

STD 和实际的解码器实现的另一个差别是在 STD 模型中有一个明显的假设:最后的音频和视频瞬时输出到用户而无更多的延迟。实际情况并不是这样,特别在阴极管显示时,在设计时必须考虑附加的延迟。对编码器的要求就是所编码的音频、视频数据在被 STD 解码时能实现正确同步。音频、视频的输入和抽样延迟,例如视频摄像机的光电转换时间,必须考虑在内。

在 STD 模型中假定了正确的同步,以此假设为基础,时间标签和缓冲区行为被作为比特流有效的一个条件来检测。当然在物理解码器中精确的同步并非自动实现,主要是由于突然启动和显示时时序抖动。解码器的精确时序就是解码器设计的一个目标。解码器

时序的不精确将影响解码器缓冲区的动作。此问题将在本附录的稍后部分详细讨论。

STD 除 PTS 字段外还包括解码时间标签(DTS)。DTS 指一 AU 从解码器缓冲区提出而在解码器模型中解码的时间。因为音频和视频原始流解码器在 STD 中为瞬时操作，解码时间和显示时间在多数情况下相同，唯一的例外是视频图象的编码比特流需重排，例如在 non\_low\_delay 视频序列中的 I 和 P 图象情况下。当有重排时，视频解码器中的临时延迟缓冲区用于存放相应的解码后的 I 图象或 P 图象，直到被显示。当解码和显示时间在 STD 中相同时，即所有的 AAU、B 图象 VAU、low\_delay 视频序列的 I、P 图象 VAU 等，因为 DTS 和 PTS 有相同值，所以不被解码。当两者不同时，如果有一个被编码，另一个一定被编码。对所有仅 PTS 被编码的 AU 来说，此字段应被解释为 PTS 和 DTS。

因为并非每个 AAU 和 VAU 都需要 PTS 和 DTS，解码器可对那些未编码的进行插值。在原始音频和视频流中，PTS 的间隔不能超过 700ms。此时间间隔是以显示时间而并非这些字段被传送和接收的时间来衡量的。这样就和这些字段的值有相同的上下文。在按本国际标准的规范部分所定义的系统、音频和视频时钟被锁定的数据流中，跟在一个 DTS 或 PTS 被直接编码的 AU 之后的每一个 AU 的有效解码时间为前一个解码时间加上一基于 STC 的恒定差值。例如，以 29.97Hz 编码的视频中每一图象在视频和系统时钟被锁定时与前一图象在时间上相差 STC 中 90kHz 部分的 3003 个周期。在连续 AU 的解码中存在同样的时间关系，虽然解码器中重排延迟影响了解码器中 AU 和显示 PU 之间的关系。当数据的编码中视频或音频时钟未锁定于系统时钟，则连续解码的 AU 之间的差别可用上面指示的值来估计。但这些时间差别并不完全是由于编码器中图象速率、音频采样率和系统时钟频率的关系不精确而引起的。

需要注意 PTS 和 DTS 字段本身并未表示在开始时或任何时候解码器缓冲区的正确充满，同样它们也不表示在解码开始之前接收到最初的比特时应有时间延迟。这个信息的重新获得是由 PTS 和 DTS 的功能及正确时钟恢复联合实现的，将在下面讨论。在 STD 模型中，也就是在按此模型而构造的解码器中，在假设数据的发送和此时序模型一致时，解码器缓冲区的动作完全由 SCR(或 PCR)值、它们被接收的时间、PTS 和 DTS 值所决定。此信息说明了编码数据在解码器缓冲区中所用时间。在编码数据缓冲区中数据量没有明确说明，也没有必要，因为时序已经详细说明了。还需注意若不正确使用时间标签，数据缓冲区的填充将以解码器不可预料的方式随时间发生相当大的变化。

为了使音频和视频 PTS 正确参考共同的 STC，解码器系统中必须有一正确的共同时钟。此问题在下一部分介绍。

### I-D. 0. 3 解码器中系统时间时钟的恢复

在 ITU-T Rec. H. 222. 0|ISO/IEC 13818-1 系统数据流中，除了 PTS 和 DTS 字段外还有参考时钟时间标签。这些参考为系统时间时钟的采样，可用于解码器和编码器。它们一部分的分辨力为每秒 27 000 000，在传送流中出现间隔为 100ms，在程序流中出现间隔为 700ms。这样，它们可以在解码器中用来实现时钟重建控制循环，而保证对所有应用有足够的精确度。

在程序流中，参考时钟字段称为系统参考时钟或 SCR。在传送流中，参考时钟字段称

为程序参考时钟或 PCR。通常 SCR 和 PCR 定义可认为是相同的，虽然有些区别。本节的余下部分使用 SCR 一词，除特别说明之外，那些阐述也适用于 PCR。传送流中的 PCR 为一道程序提供参考时钟，其中程序为具有共同时钟基点的原始流集合，将被同步解码和显示。传送流中可有多道程序，每一道都有独立的时间基点和各自的 PCR 集合。

当 SCR 被解码器接收时，它表示 STC 的正确值。因为 SCR 占有多个数据字节，且系统数据流被定义为字节流，所以 SCR 到达解码器被定义为 SCR\_base 的最后一个字节被解码器接收的时刻。另一种情况 SCR 可解释为 SCR 字段应到达解码器的时间，假设 STC 已知为正确的，使用哪一种解释取决于应用系统的结构。在数据源可由解码器控制的应用中，例如本地 DSM，解码器可能有自主的 STC 频率，因此 STC 无需恢复。然而在许多重要的应用中，此假设不能成立，考虑这种情况：一个数据流被同时送到多个解码器。如果每个解码器都有自己自主的 STC 和独立的时钟频率，SCR 不能保证在正确的时间到达所有解码器；通常一个解码器要求发 SCR 比源发送的速度快一些，而另一个要求慢一些。对未确定时间长度的数据接收来说这种差别不能由有限大小的数据缓冲区来弥补。因此以下主要讨论 STC 必须服从于接收到的 SCR(或 PCR)时序的情况。

在一正确构造和发送的 ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 数据流中，每个 SCR 到达解码器的精确时间由 SCR 值指示，在此上下文中，“时间”就是 STC 的正确值。在概念上，STC 值就是在 SCR 被存储或发送时编码器的 STC 值。然而编码可能不是实时操作或数据流从原始编码后已经被修改，而且通常编码器和数据源的实现可有多种方法，因此编码器的 STC 可能是一理论上的数值。

如果解码器的时钟频率和编码器的时钟频率严格匹配，那么视频和音频的解码和显示将自动和编码器保持相同的速率，而端到端的延迟将为常数。当编码器和解码器时钟匹配时，任何正确的 SCR 值可用来设置解码器 STC 的瞬时值，而且此后不需更多的调整，解码器的 STC 就将与编码器相匹配。此条件保持为“真”直到发生时序的不连续，比如说程序流的结尾或传送流中不连续标记出现。

实践中解码器的自主系统时钟频率不和编码器的系统时钟匹配，后者被抽样并以 SCR 形式表示。解码器的 STC 可利用接收到的 SCR 值而服从于编码器。使解码器的时钟服从于接收的数据流的典型方法是通过锁相环(PLL)，基本 PLL 的变形或其他的方法适用于不同的应用要求。

一种在解码器中恢复 STC 的直接型 PLL 图示描述如图 I-D-2。

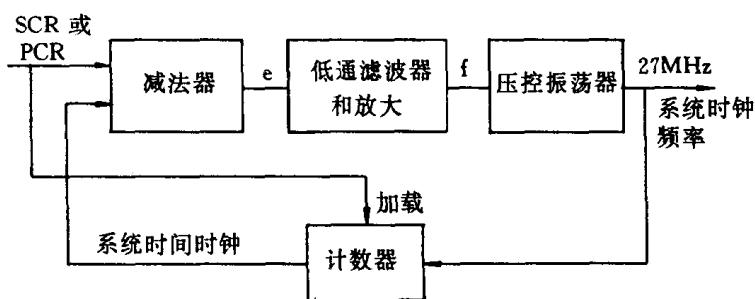


图 I-D-2 使用 PLL 的 STC 恢复

此图表示了一经典的 PLL,除了参考和反馈项为数字(STC 和 SCR 或 PCR 值)而不是像边缘这种信号事件。

当要求一个新的时间基点时,即一新程序,STC 被设置为编码于 SCR 中的当前值。通常第一个 SCR 被直接装入 STC 计数器,其后 PLL 为一闭环操作。也许可用本方法的变形,例如考虑到 SCR 值可能发生颤抖或错误。

PLL 的闭环操作如下所述:当每一个 SCR(或 PCR)到达解码器时,此值和 STC 的当前值比较,差值为一数字,一部分以 90kHz 为单位,另一部分以此频率的 300 倍,27MHz 为单位,此差值被转化为一维线性空间,通常以 27MHz 为单位,并被称为‘e’,表示环的误差。 $e$  值序列被输入低通滤波器并经放大以适应应用的要求。此阶段的输出为控制信号‘f’,用来控制压控振荡器(VCO)的瞬时频率,VCO 的输出为频率在 27MHz 左右的振荡信号,此信号做为解码器中的系统时钟频率。27MHz 时钟被输入到计数器中产生当前 STC 值,它由通过除 300 而得的 27MHz 扩展值和在 33 位计数器中计数而得的 90kHz 基本值两部分组成。STC 输出中 33 位 90kHz 部分用于和 PTS 和 DTS 比较,整个 STC 也被反馈输入到减法器。

在连续的 SCR(700ms)或 PCR(100ms)之间的最大间隔限制允许设计和构造稳定的 PLL。PLL 的带宽上限由此间隔决定,正像下面将表示的,在许多应用中 PLL 要求一很低的带宽,因此此限制对解码器的设计和动作无重大影响。

如果自主态或起始的 VCO 的频率足够接近正确的编码器系统时钟频率,在 PLL 达到所谓锁定状态之前,一旦 STC 正确初始化,解码器就能令人满意地操作。对一给定的解码器其 STC 频率和编码于 SCR 字段中的频率有一定限度内的差异,且此限度在解码器应用的要求之内,如果没有 PLL,则编码器和解码器 STC 频率之间的不匹配将导致解码器缓冲区填充逐渐地但不可避免地增加或减少,这样在有限解码器缓冲区大小时,最后将发生上溢或下溢。因此在解码器 STC 频率被锁定之前,允许的时间长短由允许的附加解码器缓冲区大小和延迟决定。

如果解码器接收到的 SCR 值和时序代表了编码器中恒定频率 STC 的瞬时正确采样,则误差项  $e$  在环达到锁定状态后将变为一基本恒定值。正确 SCR 值的条件或者等价于数据从编码器到解码器的恒定存储和传送延迟,或者等价于当此延迟不是常数时,SCR 值被校正以反映延迟的变化来达到恒定延迟的存储和传送的效果。随着  $e$  转化为一常数,在环被锁定后 VCO 频率的瞬时值变化基本为零,VCO 此时称为有很小的抖动或频率变化。在环被锁定的过程中,VCO 频率的变化率,也就是频率抖动率能由低通滤波器和增益层的设计而严格控制。通常 VCO 旋转率按照解码器缓冲区的大小和延时的限制而设计,以满足应用要求。

#### I-D. 0. 4 SCR 和 PCR 抖动

如果网络或传送流多路复用器使数据从编码器或存储系统到解码器的传送延迟发生变化,则这种变化可能引起 SCR(或 PCR)值和当它们被实际接收时应有的值之间有差值,这就是所谓 SCR 或 PCR 抖动。例如,如果传送一 SCR 的延迟比同一程序中其它类似字段经历的延迟大,则 SCR 迟到了;同样,如果延迟比程序中其它参考时钟字段少,则此

字段早到了。

解码器输入的时间抖动通过 SCR 的值和它们被接收的时间联合反映。假设一时钟恢复结构如图 1-D-2 所示,则时间抖动由误差项  $e$  表示,非零的  $e$  值为  $f$  变化的原因,而导致 27MHz 系统时钟的变化。根据特殊的应用要求,恢复时钟的频率变化在解码器系统中可能可以接受,也可能不能接受。例如在产生复合视频输出的精确时序解码器中,所恢复的时钟频率通常用于产生复合视频采样时钟和彩色 sub\_carrier; sub\_carrier 频率稳定性的适用规范只能允许系统时钟频率有极缓慢的调整。在解码器输入有大量的 SCR 或 PCR 抖动,且 STC 的频率抖动率有严格限制的应用中,合理的附加解码器缓冲区的大小和延迟的限制可能不承认某些操作。

SCR 或 PCR 抖动的出现可能是由于例如网络传输等因素引起的,其中综合了分组或个体的多路复用或分组通过网络的可变延迟等因素。可变延迟是由于排队延迟或在媒体共享系统中可变的网络访问时间引起的。

传送流或程序流的多路复用或再次复用改变了数据分组的次序和相对时域位置,因此也改变了 SCR 或 PCR 的这些性质。SCR 时域位置的改变引起原来正确的 SCR 值变为错误,因为通常它们的值不反映其本身在一恒定延迟的网络中被发送的时间。同样,具有正确 SCR 或 PCR 值的程序流或传送流的发送网络在数据流上增加了一可变延迟,却没有校正 SCR 或 PCR 值,这种效果又是一种 SCR 或 PCR 抖动,随时影响着解码器的设计和操作。网络对解码器收到的 SCR 或 PCR 所施加的抖动在最坏情况下的数目取决于在本国际标准之外的因素,包括在每个网络开关中实现的序列深度和串联作用于数据流之上的网络开关或再次多路复用操作的总数。

对于传送流来说,在再次多路复用操作中,PCR 的校正是必须的,可从一个或多个传送流中产生一新的传送流。此校正通过在 PCR 上加一校正项实现,此校正项如此计算:

$$\Delta \text{PCR} = \text{del}_{\text{act}} - \text{del}_{\text{const}}$$

其中  $\text{del}_{\text{act}}$  为 PCR 经历的实际延迟, $\text{del}_{\text{const}}$  为此程序中所有 PCR 都使用的一个常数。 $\text{del}_{\text{const}}$  使用值取决于原始编码器/多路复用器所使用的策略。例如,此策略为尽早划分分组,以允许其后的传送链对它的延迟。表 I-D-1 给出三种不同多路复用策略适宜的  $\text{del}_{\text{const}}$  值。

表 I-D-1 再次多路复用策略

策略	$\text{del}_{\text{const}}$
早	$\text{del}_{\text{min}}$
晚	$\text{del}_{\text{max}}$
中	$\text{del}_{\text{avg}}$

当设计一系统时,需要私用协议决定编码器/多路复用器使用何种策略,因为这将影响执行附加再次多路复用的能力。

本标准未规定多路复用抖动的限制,然而,在一性能良好的系统中 4ms 可为抖动的最大数目。

在包含再次多路复用的系统中必须特别注意保证传送流中信息的一致性,特别是对于 PSI 和不连续点。PSI 表的改变必须被插入传送流中以保证后续再次多路复用不会将这些信息移动太远而变为错误。例如在某些情况下,在此改变影响的数据的 4ms 之内不应发送新的 PMT 分段。

同样,编码器/多路复用器应避免在一不连续点周围±4ms 的窗口内插入 PTS 或 DTS。

#### I-D. 0.5 出现网络抖动时的时钟恢复

在接收的参考时钟的时间标签有很多抖动的应用中,解码器的设计有几种可选方案。到底怎么设计解码器主要依赖于所要求的解码器输出信号的特性以及输入数据和抖动的特性。

各种解码器应用对所恢复系统时钟的精确度和稳定性有不同要求,且所要求的稳定性和精确性程度可认为处于一条轴上。此轴的一个极端可认为是那些重建的系统时钟直接用于合成复合视频彩色 sub-carrier 的应用。此要求通常存在于那些精确时序类型的视频中,如上所述每幅编码图象严格显示一次;也用于符合可用规范的复合视频的输出中。在这种情况下,彩色 sub-carrier、象素时钟、图象速率等都有精确说明的速率,而且和系统时钟有确定的关系。复合视频 sub-carrier 必须有足够的精确度和稳定性来保证普通的电视接收机彩色 sub-carrier PLL 能锁定 sub-carrier,且使用恢复的 sub-carrier 进行解调彩色信号不出现可觉察的人为造成的色度相位。一些应用中,要求使用系统时钟产生完全满足 NTSC, PAL 或 SECAM 规定的 sub-carrier。这些规定通常比用于普通电视机的更严格。例如 NTSC 的 SMPTE 规定要求 sub-carrier 3ppm 的精确度,最大短期抖动为每水平线时间 1ns,最大长期漂移为每秒 0.1Hz。

在恢复的系统时钟不用于产生彩色 sub-carrier 的应用中,它也可能用来产生视频象素时钟和音频采样时钟。这些时钟有它们自己的稳定性要求,取决于关于接收显示监视器的假设和在解码器输出的可接受的音频频率漂移或“失真和抖动”。

在每幅图象和每个音频采样不是仅播放一次的应用中,例如允许“跳过”图象和音频采样时,系统时钟可以有相对宽松的精确度和稳定性要求。这种解码器没有精确的音频、视频显示同步,最后的音频和视频显示质量与精确的时序解码器的质量不同。

对恢复的系统时钟的精确度和稳定性的要求的选择取决于不同的应用。下面集中讨论上面提到的最严格的要求:系统时钟用于产生彩色 sub-carrier。

#### I-D. 0.6 用于产生彩色 sub-carrier 的系统时钟

产生 sub-carrier 的要求和必须接受的最大数目网络抖动决定了解码器的设计要求。同样,如果系统时钟性能要求和解码器设计能力已知,则可决定最大容许网络抖动。虽然描述这样的要求已超出本国际标准的范围,为了简化问题的描述和说明代表性的设计方法,说明设计的数字被一一指出。

对于图 I-D-2 所示的时钟恢复 PLL 电路,所恢复的系统时钟必须满足相对于正常情况时以 ppm 为单位的最坏频率偏移要求和以 ppm/s 为单位的最坏频率抖动率。峰到峰

的非校正网络时序抖动值以毫秒为单位。在这种 PLL 中网络时序抖动以误差项  $e$  的形式在图中出现,又因为 PLL 相当于输入抖动的一个低通滤波器,当输入的 PCR 时序出现最大幅度的脉冲函数时,对 27MHz 的输入频率有最大影响,这时  $e$  值为等于峰到峰抖动的最大幅度。低通滤波器(LPF)输出  $f$  的最大改变率,在输入  $e$  为最大值时,直接决定 27MHz 输出的最大频率抖动率。可以规定具有任何最大  $e$  值和最大  $f$  改变率的 LPF。然而由于 LPF 的增益或截止频率减小,PLL 锁定 SCR 或 PCR 的频率所需的时间增加。长时间恒定的 PLL 实现可使用数字 LPF 技术,也可使用模拟滤波器技术。在数字 LPF 实现时,当频率项  $f$  为到模拟信号 VCO 的输入,需通过一数模转换器,在计算输出频率的最大抖动率时需考虑其脉冲的大小。

为了保证  $e$  值集中于一趋于零的值,PLL 的开环增益必须很大,这可以通过 PLL 中低通滤波器的积分函数实现。

对于给定的精确度要求,有可能构造一 PLL 使其初始化操作频率满足精确度要求。此时在 PLL 锁定之前,27MHz 的初始化频率已足够满足所述输出频率要求的精确度。如果不是因为解码器缓冲区最终将上溢或下溢的事实,此初始化时钟频率对长期操作已足够了。然而,从解码器开始接收和译码数据,到系统时钟锁定接收到的 SCR 或 PCR 所代表的时间和时钟频率为止,数据到达缓冲区的速率和其被移出的速率不一样,或同样地,解码器移出访问单元的时间和系统目标解码器(STD)模型中的不一样。根据相对于编码器时钟频率所恢复系统时钟的轨迹,解码器缓冲区相对于 STD 的缓冲区将连续地变得越来越多或越来越少。依赖于初始 VCO 频率和编码器系统时钟频率,解码器缓冲区填充增加或减少。假设不知道这种关系,解码器需要附加的数据缓冲来应付各种情况。为应付初始化 VCO 频率大于编码器时钟频率而分配了附加的缓冲,解码器的构造必须使所有解码操作至少延迟相当于附加缓冲的时间以防止缓冲区下溢,如果初始化 VCO 频率的精确度不足以达到前面所述的精确度要求,则 PLL 必须在解码开始之前达到锁定状态,对于此时 PLL 的动作及合适的附加缓冲区数和静态延迟另有考虑。

产生图 I-D-2 PLL 中误差项  $e$  的脉冲函数的输入时序抖动脉冲将产生输出频率项  $f$ ,必须保证其乘上 VCO 增益后,最大改变速率小于规定的频率抖动率。VCO 的增益指相对于控制输入改变的输出频率改变。对 PLL 中 LPF 的一附加限制是当环被锁定时, $e$  的静态值必须受限以限制必须实现的附加缓冲和静态解码延迟。当 LPF 有很大的 DC 增益时,此项减小。

和图 I-D-2 有所不同的时钟恢复电路是可行的。例如,它可能实现一数字控制振荡器(NCO)的控制环,而不是 VCO,在 NCO 中使用一固定频率的振荡器,通过从输出的正常周期事件中插入或删除时钟周期以调整解码和显示时序。这种实现方法在和复合视频一起使用时可能有困难,可能引起 sub\_carrier 有问题的相位漂移或水平、垂直扫描时序的抖动。一种可能的方法为在保持彩色 sub\_carrier 相位的同时,在垂直空白的开始处调整水平扫描周期。

总的来说,根据要求所规定的值,在保持所需解码器缓冲区大小和增加的解码延迟的同时,构造能重构具有足够精确度和稳定性的系统时钟的解码器可能是可行的,也可能不可行。

## I-D. 0.7 视频和音频分量的重构

如果解码器输出产生视频分量,时间精确度和稳定性的要求通常比复合视频的要求低。通常,频率容许值就为显示偏转电路系统所能接受的值,而稳定性容忍值由避免显示中可察觉图象错位的需要决定。

前面描述的原理依然有效,只是所提出的要求通常更容易满足。

音频采样率的重建也遵循这种原理,然而稳定性要求由可接受的长、短期抽样率变化决定。使用上一节中的 PLL 方法,短期偏移可以很小,而长期的频率变化就表示为可觉察的音高的变化。同样,一旦此变化的界限设定,就可决定设计的要求。

## I-D. 0.8 帧的掠过

在某些不要求精确解码器时序的应用中,解码器的系统时钟并不需调整其操作频率以适应所接收 SCR(或 PCR)代表的频率;在解码器 STC 服从于所接收数据的同时,它可能有一自由的 27MHz 时钟。这种情况下必须修改 STC 值以匹配所接收的 SCR。根据 SCR 的接收情况修改 STC 将引起 STC 值的不连续,这些不连续的最大幅度取决于解码器 27MHz 时钟和所接收 SCR 代表的编码器 27MHz 时钟之间的差别,也取决于接收到的连续 SCR 或 PCR 之间的间隔。既然解码器的 27MHz 时钟频率未锁定于接收的数据的时钟频率,它就不能在保证下面所述的精确时序假设的同时,又被用来产生视频或音频采样时钟。有关假设为:每个视频和音频显示单元严格显示一次,以及在精确的音频和视频同步下,在解码器和编码器保持同样的显示图象及音频输出。有多种可能性来实现这种解码和显示系统。

在一类实现中图象和音频采样的解码时间由解码器 STC 指定,而它们的显示时间和当地产生的抽样时钟稍有不同。根据解码器的抽样时钟和编码器系统时钟的关系,图象和音频采样偶尔会显示多次或根本不显示,在音频中这就称为“帧的平移”或“采样平移”。这种机制可能导致能感觉到的不自然。由于在这些时间单元中图象和音频演示单元被重复或删除,音频视频的同步通常是不精确的。根据特殊的实现要求,解码器中经常需要附加的缓冲区来存储编码数据或解码的显示数据。解码操作可能在显示之前即完成,而不完全是在由解码器 STC 指示的时间来完成,或者解码后的显示单元为延迟的和可能重复的显示而存储。如果解码操作在显示时间进行,需要一种机制来支持删除图象和音频采样的显示而不影响预测编码数据的解码。

## I-D. 0.9 网络抖动的平滑

在某些应用中有可能在网络和解码器之间引入一种机制以减小网络引起的抖动程度,这种方法是否可行取决于所接收到数据流的类型和所预计的抖动类型。

所有传送流和程序流在它们的语法中都表明了流输入解码器的速率,这些指示的速率都不精确,不能用来精确重建数据流时序,然而却可作为平滑机制的有用部分。

例如传送流可能来自于一个网络,这样数据的传送便是突发性的,有可能缓冲所接收的数据,并以近乎恒定的速率送给解码器以保证缓冲区接近半充满。

然而,速率可变流并不以恒定速率发送,这时平滑缓冲区不会总是半充满的。通过缓冲区的恒定平均延迟要求缓冲区的填充随着数据率而改变。数据从缓冲区中提取并输入到解码器的速率可近似使用数据流中的速率信息。在传送流中预期的速率由 PCR 字段的值及它们之间的传送流字节的数目决定。在程序流中预期的速率由 Program\_mux\_rate 明确规定,虽然在此标准规定中,在 SCR 处速率可能下降为零,例如当数据以指定速率发送时,如果 SCR 比预期时间早到就有可能发生这种情况。

对于可变速率流,平滑缓冲区的正确填充随时间而改变,且不由速率信息严格规定,在一种可选方法中,SCR 和 PCR 用来衡量数据进入缓冲区的时间以及控制离开缓冲区的时间。可以设计一控制循环来提供通过缓冲区的恒定平均延迟,其设计和图 I-D-2 中的控制循环相似。在解码器之间插入一个平滑机制所获得的效果也可通过串联多个时钟恢复 PLL 获得;串联 PLL 中联合的低通滤波器有利于从接收时序中剔除抖动。

## 附录 I-E 数据传输应用

(此附录不是本推荐|国际标准的必需部分)

### I-E. 0 几点考虑

- ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 传送多路复用不仅仅用来传送音频和视频,也可用来传送数据。
- 数据原始流不具有播放应用中视频和音频流所表现的连续性。
- 虽然已可以识别一数据 PES 分组的开始,通常不能通过下一个数据 PES 分组的开始来识别此数据 PES 分组的结束,因为一个(或更多的)带有数据 PES 分组的传送分组将丢失。

### I-E. 1 建议

一合适的解决方法是紧接一相关 PES 分组立即传送其后的 PES 分组,当没有 PES 分组可发送时,可用无有效负载的 PES 分组代替。

表 I-E-1 中为一此类 PES 分组的例子。

表 I-E-1 PES 分组首部示例

PES 分组头域	值
packet_start_code_prefix	0x000001
stream_id	assigned
PES_packet_length	0x0003
'10'	'10'
PES_scrambling_control	'00'
PES_priority	'0'
data_alignment_indicator	'0'
copyright	'0'
original_or_copy	'0'
PTS_DTS_flags	'00'
ESCR_flag	'0'
ES_rate_flag	'0'
DSM_trick_mode_flag	'0'
additonal_copy_info_flag	'0'
PES_CRC_flag	'0'
PES_extension_flag	'0'
PES_header_data_length	0x00

## 附录 I-F ITU-T H.222.0 | ISO/IEC 13818-1 的语法图

(此附录不是本推荐|国际标准的必需部分)

### I-F.0 简介

本标准的这部分是非规范的附录,提供传送流和程序流的语法图解。此部分决不能代替前面的规范部分。

为了产生清晰的图例,并非所有字段都被完全描述或表示,保留字段可被忽略或以阴影表示。字段长度由比特数表示。

#### I-F.0.1 传送流语法

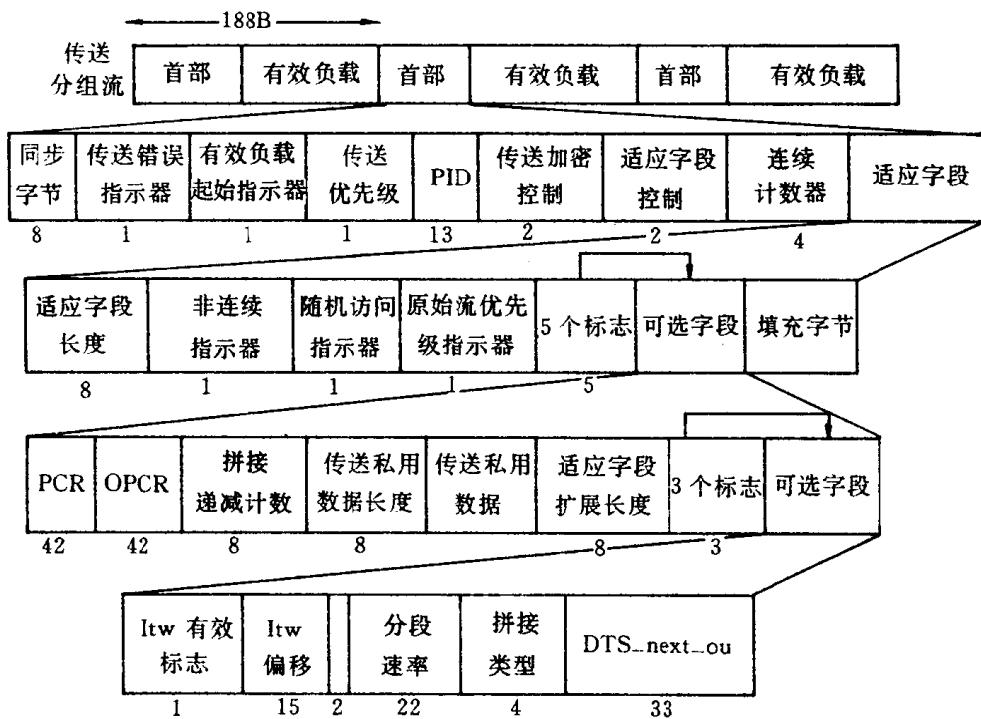


图 I-F-1 传送流语法图

## I-F. 0.2 PES 分组

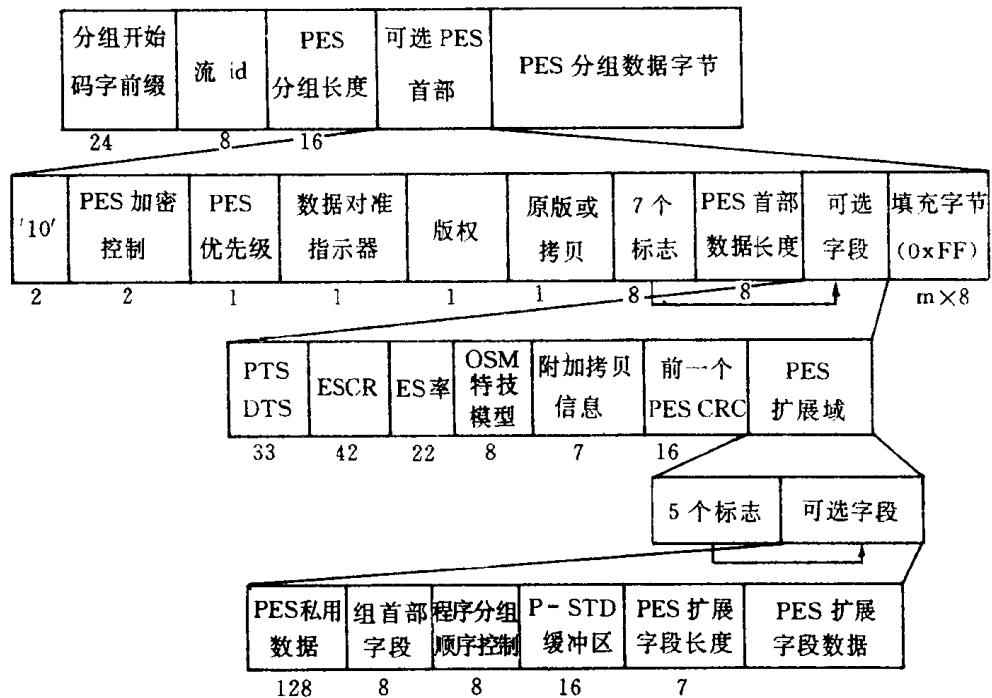


图 I-F-2 PES 分组语法图

## I-F. 0.3 程序关联分段

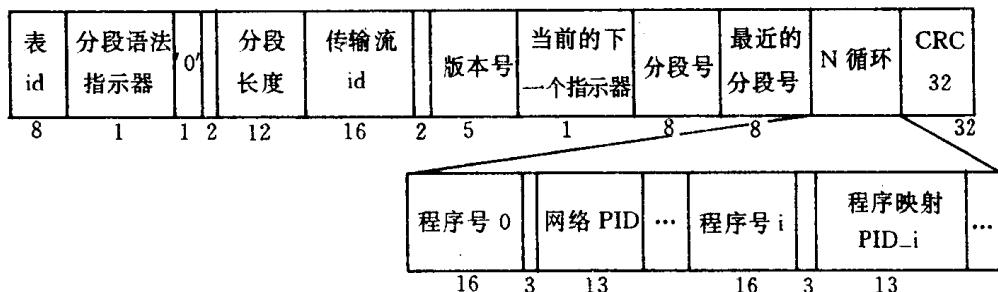


图 I-F-3 程序关联分段图

## I-F. 0.4 CA 分段

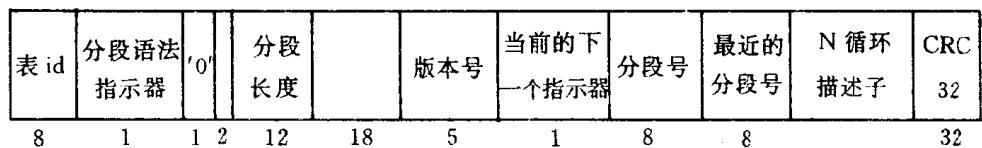


图 I-F-4 条件访问分段图

### I-F. 0.5 TS 程序映射分段

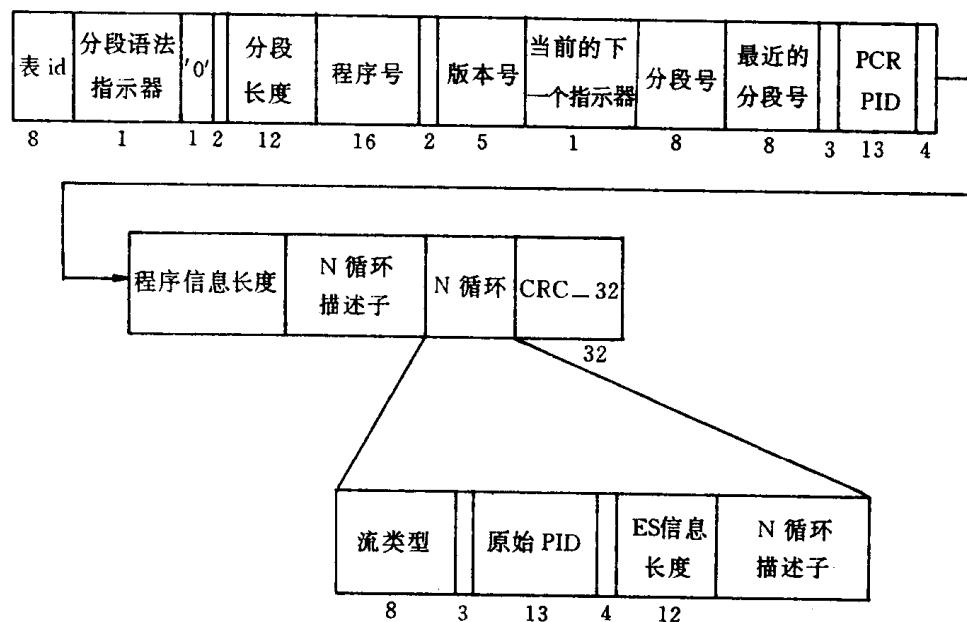


图 I-F-5 TS 程序映射图

### I-F. 0.6 私用分段

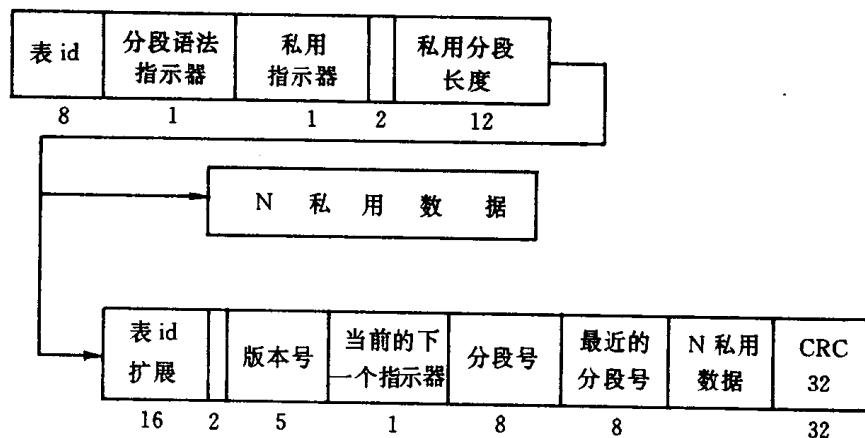


图 I-F-6 私用分段图

## I-F.0.7 程序流

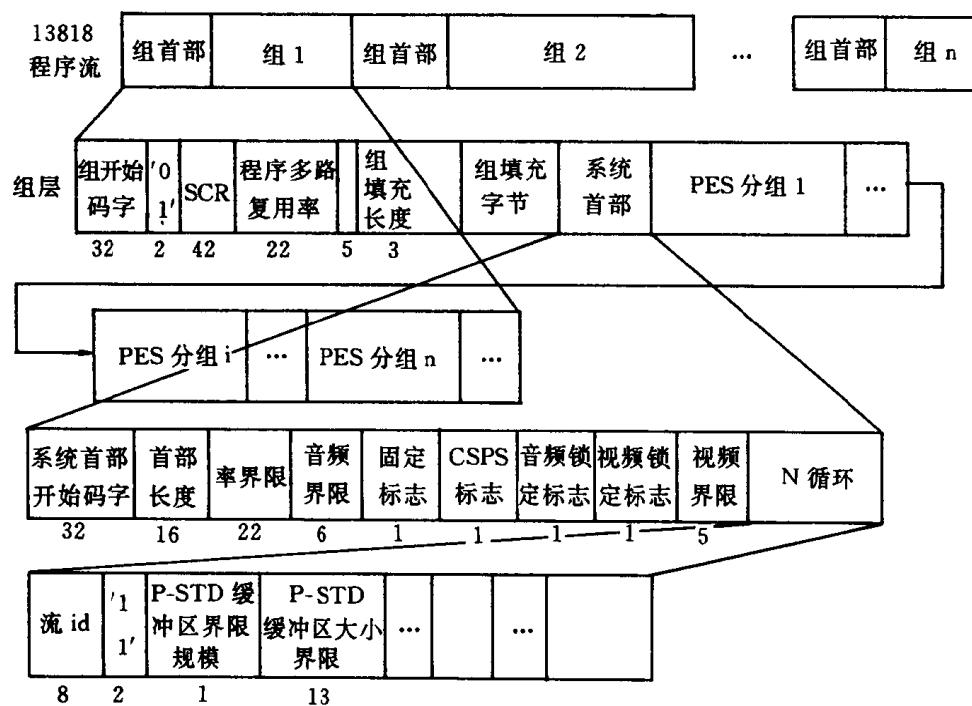


图 I-F-7 程序流图

## I-F.0.8 程序流映射

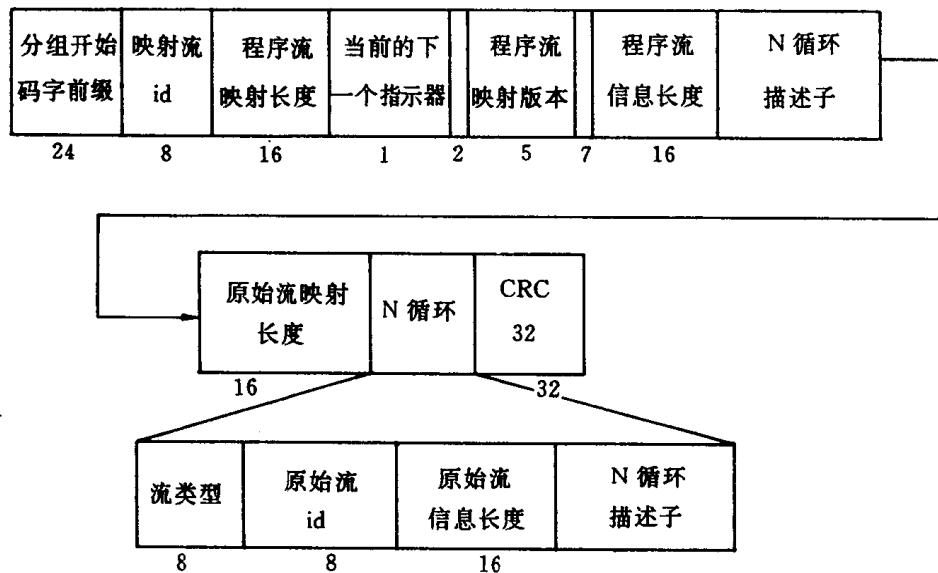


图 I-F-8 程序流映射图

## 附录 I-G 普通信息

(本附录不是本推荐|国际标准的必需部分)

### I-G. 0 普通信息

#### I-G. 0. 1 同步字节竞争

建议在选择 PID 值时应避免规则的同步字节竞争。这种竞争有可能在 PID 字段或 PID 字段和相邻的标志设置组合中潜在发生。建议在最多 4 个连续传送分组中分组首部的同一位置允许同步字节竞争发生。

#### I-G. 0. 2 忽略的图象状态和解码处理

假设被显示的序列中只含有 I 和 P 帧。下一幅被解码的图象标为 picture\_next, 当前被显示的图象标为 picture\_current。因为视频编码器有可能忽略图象, 当需要把它们移出并瞬时解码和显示时, 有可能并非所有 picture\_next 的比特都在 STD 缓冲区中。当这种情况发生时, 在 STD 中 picture\_next 的所有比特被瞬时移去, 并再次显示 picture\_current。当下一幅图象的显示时间到达时, 如果相应于 picture\_next 的余下的比特正在 STD 缓冲区中, 则这些比特被移出, picture\_next 被显示; 如果 picture\_next 的所有比特都不在 STD 缓冲区中, 仍出现上面的移出比特操作而重复显示 picture\_current, 此操作重复进行直到 picture\_next 可以显示。注意, 如果比特流中 PTS 位于 picture\_next 之前, 则错误发生且必须被忽略。

无论何时上面描述的忽略图象的情形发生, 编码器需在 picture\_next 之后, 在图象被解码之前插入一 PTS。这就使解码器能立即证实已正确显示所接收的图象序列。

#### I-G. 0. 3 PID 值的选择

鼓励应用中尽可能多地使用小的 PID 值和组值。

#### I-G. 0. 4 PES 开始码字竞争

具有 packet\_start\_code\_prefix(0x000001)值的三个连续字节和第四个字节连接时有可能在流中不应该的地方和 PES\_packet\_header 发生竞争。

因此开始码字竞争不可能在视频原始流中发生, 只可能发生在音频和数据原始流中, 也有可能在 PES\_packet\_header 和 PES\_packet 有效负载的交接处发生, 即使 PES\_packet 分组有效负载为视频信号。

## 附录 I-H 私用数据

(本附录不是本推荐|国际标准的必需部分)

### I-H.0 私用数据

私用数据在本国际标准中指不按 ITU-T |ISO/IEC 规定标准编码的用户数据。此数据内容在将来不会也不必要在 ITU-T Rec. H. 222.0 |ISO/IEC 13818-1 中规定。

私用数据可在 ITU-T Rec. H. 222.0 |ISO/IEC 13818-1 语法中的下述位置中携带。

#### 1. 传送流分组表 I-2-3

`transport_packet()`语法的数据字节中可带有私用数据。这种格式携带的私用数据被称为流类型表 I-2-35 中的用户私用。它允许带有私用数据的传送流分组中也可带有 `adaptation_field()`。

#### 2. 传送流调整字段表 I-2-6

在 `adaptation_field()` 中出现任何可选 `private_data_bytes` 的出现由 `transport_private_data_flag` 指示。`private_data_bytes` 的数目固定由 `adaptation_field_length` 字段的语义限制, 其中 `adaptation_field_length` 不能超过 183 字节。

#### 3. PES 分组表 I-2-17

在 PES 分组中传送私用数据有两种可能性。第一种可能性是在 `PES_packet_header` 中 16 字节的可选 `PES_private_data`, 此字段的出现由 `PES_private_data_flag` 指示。`PES_private_data_flag` 的出现由 `PES_extension_flag` 指示。当将这些字节(如果出现的话)和其相邻字段一起考虑时, 它们不会和 `packet_start_code_prefix` 发生竞争。

第二种可能性是在 `PES_packet_data_byte` 字段中。这被称为流类型表 I-2-35 里 PES 分组中的私用数据。这些私用数据可分为两类:`private_stream_1` 指满足 `PES_packet()` 语法的 PES 分组中的私用数据, 其中到 `PES_header_data_length` 为止(并包括它)但并不受它限制的所有字段都出现;`private_stream_2` 指 PES 分组中的私用数据, 其中仅有头三个字段, 后面跟着带有私用数据的 `PES_packet_data_bytes`。

注意, 程序流和传送流中都有 PES 分组, 因此程序流和传送流中都有 `private_stream_1` 和 `private_stream_2`。

#### 4. 描述子

描述子存在于程序流和传送流中。用户可以定义一定范围内的私用描述子。这些描述子将以 `descriptor-tag` 和 `descriptor-length` 字段开始。对于私用描述子, `descriptor-tag` 的取值在 64—255 之间, 在表 I-2-39 中定义。这些描述子可置于 `program_stream_map()` (表 I-2-23), `CA_section()`(表 I-2-27), `TS_program_map_section()`(表 I-2-28)和任何私用 `section()`(表 I-2-27)中。

特别地, `private_data_bytes` 也出现在 `CA_descriptor()` 中。

## 5. 私用分段

私用分段表 I-2-29 又提供了两种形式的携带私用数据的方法。这种类型的原始流可凭流类型表识别为 PSI 分段中的 `private_data`。一种类型的 `private_section()` 仅包括头五个定义的字段, 后面跟着私用数据。这种结构时, `section_syntax_indicator` 应置 ‘1’, 且到 `last_section_number` 为止(并包括它)的所有语法字段都出现, 其后跟着 `private_data_bytes` 并以 CRC-32 结尾。

## 附录 I-I 对 ITU-T Rec. H. 222.0 | ISO/IEC 13818 已提出专利申请的公司名单

(此附录不是本推荐|国际标准的必需部分)

### I-I.0 对 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 已提出专利申请的公司

用户需注意这种可能性——对于在 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 所规定的一些过程来说,与此国际推荐标准一致,可能需使用涉及某些有专利权的发明。

在出版此部分 ITU-T Rec. H. 222.0 | ISO/IEC 13818-1 时,没有注意考虑此声明的有效性或这里有关的任何专利权的有效性。然而,本附录中列出的每一个公司都已向 ITU-TF 提交了一份声明,表示愿意在坚持申请这些专利的合理公平的限制和条件基础上发放专利许可证,有关这些专利的材料可从下述组织获得。

表 I-I-1 总结了所收到的正式的专利声明并指明了其对应本标准的部分。名单中包括了提出非正式声明的组织。但若无‘X’,则表示尚未收到这些组织的正式专利声明。

表 I-I-1 提交专利申明的公司清单

公 司	V	A	S
AT&T	X	X	X
BBC Research Department			
Bellcore	X		
Belgian Science Policy Office	X	X	X
BOSCH	X	X	X
CCETT			
CSELT	X		
David Sarnoff Research Center	X	X	X
Deutsche Thomson-Brandt GmbH	X	X	X
France Telecom CNET			
Fraunhofer Gesellschaft		X	X
GC Technology Corporation	X	X	X
General Instruments			
Goldstar			

续表

公 司	V	A	S
Hitachi, Ltd.			
International Business Machines Corporation	X	X	X
IRT		X	
KDD	X		
Massachusetts Institute of Technology	X	X	X
Matsushita Electric Industrial Co. ,Ltd.	X	X	X
Mitsubishi Electric Corporation			
National Transcommunications Limited			
NEC Corporation		X	
Nippon Hoso Kyokai	X		
Nippon Telegraph and Telephone	X		
Nokia Research Center	X		
Norwegian Telecom Research	X		
Philips Consumer Electronics	X	X	X
OKI			
Qualcomm Incorporated	X		
Royal PTT Nederland N. V. ,PTT Research(NL)	X	X	X
Samsung Electronics			
Scientific Atlanta	X	X	X
Siemens AG	X		
Sharp Corporation			
Sony Corporation			
Texas Instruments			
Thomson Consumer Electronics			
Toshiba Corporation	X		
TV/Com	X	X	X
Victor Company of Japan Limited			

## **附录 I-J 系统一致和实时接口**

(此附录不是本推荐|国际标准的必需部分)

### **I-J. 0 系统一致和实时接口**

ITU-T Rec. H. 222. 0 | ISO/IEC 13818-1 程序流和传送流的一致性在本推荐|国际标准的这一部分中以规范性说明规定。这些说明中包括系统目标解码器(T-STD 和 P-STD)，用来规定当流输入一理想化解码器时解码器的动作。此模型和相关的证明不包括有关流的实时发送操作信息，除了传送流和程序流代表的系统时钟频率的精确度。所有的传送流和程序流都必须遵从此说明。

还有一关于输入解码器的传送流和程序流的实时接口说明，此说明允许 MPEG 解码器和转接器到网络、通道或存储媒介间的接口标准化。虽然没有必要将所有的 MPEG 解码器都实现这个接口，但包含此类接口的实现都必须遵守此说明。此说明包括将传送流和程序流发送到解码器的实时动作，以保证解码器中编码数据缓冲区不发生上溢或下溢，并保证解码器能执行应用所要求的时钟恢复。

MPEG 实时接口规定了相对于流中编码于 PCR 和 SCR 字段的理想字节发送时间表的最大偏移。

## 附录 I-K “抖动引入”网络到 MPEG-2 解码器的接口

(此附录不是本推荐 | 国际标准的必需部分)

### I-K. 1 简介

在本附录中系统流既指 MPEG-2 传送流又可指 MPEG-2 程序流。当使用 STD 一词时,可以理解为对程序流的 P-STD 和对传送流的 T-STD。

系统流的预期字节发送时间表可通过分析流而推导。系统流如果能被 STD——理想解码器的数学模型——解码,就称为合格的。如果一合格流通过一“抖动引入”网络而传送,字节的实际发送时间表和预期发送时间表有很大的不同。这种情况下此系统流可能不能由理想解码器解码,因为抖动将引起缓冲区的上溢和下溢,且难以恢复时间基点。这种“抖动引入”网络的重要例子为 ATM。

此附录的目的就是对有关在“抖动引入”网络上发送系统流的问题提供指导和考察。发送系统流的网络特定一致模型有可能针对于包括 ATM 在内的一些网络而开发。STD 加上实时接口定义就可完全定义这种网络,开发网络一致模型的框架在 I-K. 2 节中出现。

I-K. 3 节中讨论了网络编码的三个例子,它们使得构造抖动平滑网络转接器成为可能。在第一个例子中假设了一恒定比特率系统流并使用 FIFO 进行抖动平滑。在第二个例子中网络调整层包括利于抖动平滑的时间标签。在最后一个例子中设想一普通的网络时钟可以端到端地获得,并被用来实现抖动平滑。

I-K. 4 节中提供了两个可以容忍网络引入抖动的解码器的实现。在第一个例子中,在网络输出和 MPEG-2 解码器之间插入一抖动平滑网络转接器。此 MPEG-2 解码器假定为遵守实时 MPEG-2 接口说明,此接口要求 MPEG-2 解码器比理想化的 STD 解码器有更大的抖动容忍度。网络转接器处理输入抖动比特流并输出具有与实时规范一致的字节传送机制的系统流。例一在第 I-K. 4. 1 节中讨论。因为网络转接器需要两步处理,这对一些应用来说太昂贵了。因此在第二个例子中去抖动和 MPEG-2 解码功能被合并。去抖动设备的中间处理被绕过,在时钟恢复时只需要一步就可以了。执行集成化去抖动和解码操作的解码器在本附录中被称为集成化网络特殊解码器,或简称集成解码器。集成解码器在 I-4. 2 节已讨论。

为了建造一网络转接器或集成解码器,必须设定峰到峰网络抖动的最大值。为了提高交互操作性,对每种相关网络类型的峰到峰抖动限制都必须进行规定。

## I-K. 2 网络一致模型

一种通过抖动引入网络传送系统流的模型如图 I-K-1 所示。

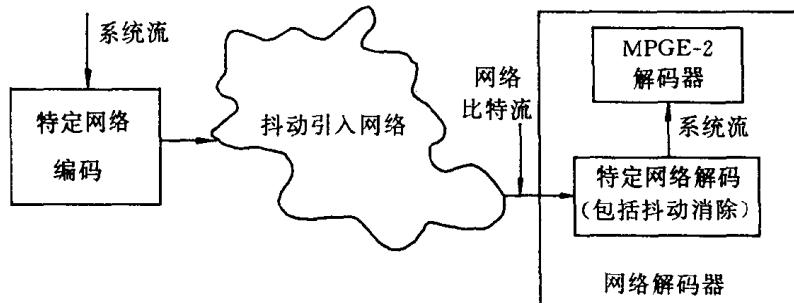


图 I-K-1 在一抖动引入网络中传送系统流

系统流被输入到一特定网络编码设备中,被转化成网络特殊格式。支持从网络输出中除去抖动的信息属于此格式的一部分。网络解码器由一特定网络解码器和 MPEG-2 解码器构成。设想 MPEG-2 遵守实时接口说明,可以和 STD 有着一样的体系结构,只是带有更大的缓冲区以提供抖动容忍,特定网络解码器移去由特定网络编码器所加的非 MPEG-2 数据并除去网络输出中的抖动。特定网络解码器的输出为遵守实时说明的系统流。

网络目标解码器(NTD)可基于以上结构定义。合格的网络流就是能被 NTD 解码的流。网络解码器是合格的,如果它能对任何可被 NTD 解码的网络比特流进行解码。实际的网络解码器不一定有 NTD 的体系结构。

## I-K. 3 抖动平滑的网络说明

对于具有恒定比特率的系统流来说,抖动平滑可通过 FIFO 实现;网络调整层中不需要可支持抖动消除的附加数据。在网络编码中所增加的字节被移去后,系统流数据被放入一 FIFO。一 PLL 根据缓冲区填充情况调整输出速率以保持缓冲区合适的半充满。在此例中抖动平滑的效果取决于 FIFO 的大小和 PLL 的特性。

图 I-K-2 表示实现抖动平滑的第二种方法,在此例中设想一支持网络调整层的时间标签。使用这种方法可以对所有恒定比特率和可变比特率的系统流进行抖动消除。

设想网络转接器设计用来补偿 J 秒的峰到峰抖动,预期的字节发送时间表从一时钟 (TC) 的参考时钟抽样(CR)而重建。CR 和 TC 分别对应于 PCR 和 STC。网络数据分组 (NDP) 编码器将每一个系统流分组转化为网络数据分组 (NDP)。网络数据分组包含一 带有 CR 值的字段,TC 的当前值在 NDP 离开 NDP 编码器时被插入。网络传输分组 (NXP) 函数将 NDP 封装于网络传输分组中,在通过网络传输后,当 NDP 进入 NDP 解码器时 CR 被 NDP 解码器抽取出来。CR 可通过使用如 PLL 来重建 TC。当延迟的 TC(TCd)等于第一个 MPEG-2 分组 CR 时,第一个 MPEG-2 分组被从抖动消除缓冲区中移出,以后

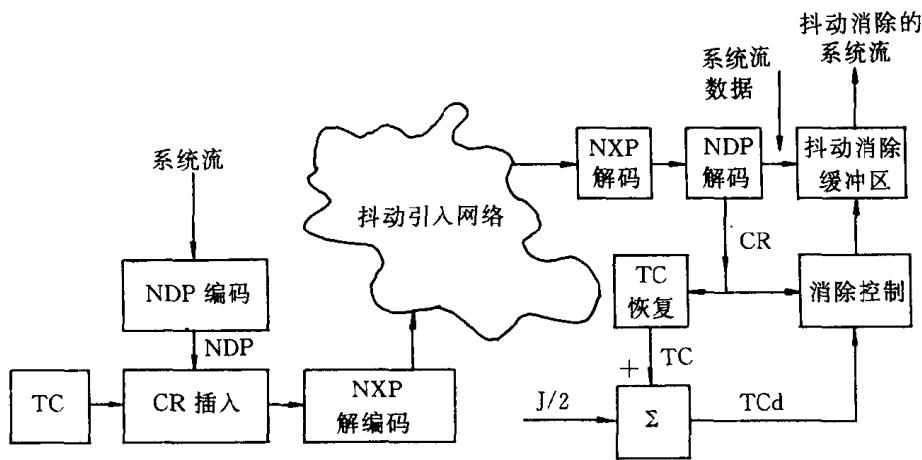


图 I-K-2 使用网络层时间标签进行抖动平滑

的 MPEG-2 分组在它们的 CR 值等于 TCd 时被移出。

当忽略 TC 时钟恢复循环的速度和 TC 的整个这类实现时的细节时,抖动消除缓冲区大小仅取决于将被平滑的最大峰到峰抖动和系统流的最大传送速率。抖动消除缓冲区大小  $B_{dj}$  由下式决定:

$$B_{dj} = JR_{\max}$$

其中  $R_{\max}$  为以每秒比特数为单位的系统流最大数据速率。当通过网络的分组经历正常的延迟时,缓冲区为半满,当经历  $J/2$  秒的延迟时,缓冲区为空,当经历  $-J/2$  秒的延迟(超前)时,缓冲区为满。

作为最后一个例子,在某种情况下一共同的网络时钟可以端到端地获得。且可以将系统时钟频率锁定于一共同时钟。网络转接器可用 FIFO 进行抖动平滑。转接器使用 PCR 或 SCR 来重建原始字节发送时间表。

## I-K. 4 解码器实现示例

### I-K. 4. 1 跟有 MPEG-2 解码器的网络转接器

在此实现中一符合网络一致性说明的网络转接器被联接于一符合实时接口说明的 MPEG-2 解码器。

### I-K. 4. 2 集成解码器

在 I-K. 4. 1 节中给出的例子要求两步处理:第一步是进行网络输出的抖动消除;第二步通过处理 PCR 或 SCR 恢复 STC,为 STD 解码器所需。本节中给出的解码器将抖动消除和解码功能集成于单个系统中。STC 的时钟恢复直接使用抖动的 PCR 或 SCR 值。此例中使用 MPEG-2 传送流。

图 I-K-3 表示了集成解码器的操作。输入解码器的网络分组流和图 I-K-2 中的一样。

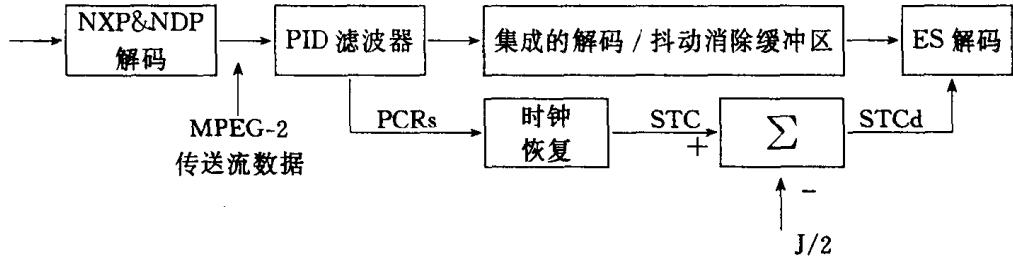


图 1-K-3 集成的抖动消除和 MPEG-2 解码

输入的网络分组被 NXP 和 NDP 的解码函数转化为 MPEG-2 传送流数据。抖动的 MPEG-2 传送流接着被滤波以抽取具有所需的 PID 的分组。在本情况中, 被解码的 PID 中也带有 PCR, PCR 值被送入 PLL 以恢复 STC。被选中 PID 的整个分组被置于集成缓冲区中。从 STC 中减去一正值  $J/2$  得到延迟的 STC, STCd。同样,  $J$  为“网络适应”解码器所容许的峰到峰抖动。引入延迟以保证在访问单元的 PTS/DTS 等于 STCd 的当前值时, 此访问单元的所有数据都已到达缓冲区。

忽略 STC 时钟恢复循环的速度和 STC 整合这些实现细节时, 有

$$\begin{aligned} B_{\text{size}} &= B_{\text{dec}} + B_{\text{mux}} + B_{\text{OH}} + 512 + B_J \\ &= B_n + 512 + B_J \end{aligned}$$

其中  $B_J = R_{\max}J$ ,  $R_{\max}$  为数据输入 PID 滤波器的最大速率。依据不同实现, 集成存储器可像在传送 STD 那样分为两部分。



## **第Ⅱ部分 视 频**



## II -0 视频引言

### II -0.1 目的

数字存储媒体、电视广播以及通信等应用领域对于运动图象及其伴音的通用编码方法的要求日益增大,本规范的这一部分即应这种要求而产生。它的用处在于可以使运动视频数据作为一种计算机可处理的数据形式,并且可以存储在各种存储媒体上,可以在现存或未来的网络上发送、接收,并且可以在现存或未来的广播信道上传播。

### II -0.2 应用

本规范涉及的应用领域(但并非仅限于此),如下所列:

BSS	卫星转播服务(家庭应用)
CATV	光纤网、铜轴电缆等上的有线电视转播
CDAD	有线数字音频转播
DAB	数字音频广播(地面和卫星广播)
DTTB	数字地面电视广播
EC	电子影院
ENG	电子化新闻采集(包括 SNG——卫星新闻采集)
FSS	固定卫星服务
HTT	家庭电视剧场
IPC	人际通信(电视会议,视频电话等)
ISM	交互式存储媒体(光盘等)
MMM	多媒体邮件
NCA	新闻和快讯
NDB	网络数据库服务
RVS	遥控监视
SSM	连续存储媒体(数字式的 VTR 等)

### II -0.3 框架和级别

本规范拟成为一个通用的应用于较广范围的比特率、分辨率、质量和服务的标准。除了其它应用,本规范至少还应涉及数字存储媒体,电视广播和通信。在创建本规范的过程中,考虑到了各个典型领域的应用要求,制定了必要的算法元素,并将其综合为一个单一的语法体系,因此,本规范可以使得各种应用系统中的数据交换变得很容易。

考虑到实现这个规范的完整语法体系的实效,我们通过“框架”和“级别”的方式来约定有限数目的语法子集。有关的术语将在本规范的第3章中正式定义。

框架是本规范定义的完整比特流语法的一个子集。在给定某框架所规定的语法范围后,比特流参数的各种取值仍可使编码和解码过程有很大变化。例如可以指定帧大小(近似)为 $2^{14}$ 个象素点宽及 $2^{14}$ 个象素点高。目前实现一个能处理任意大小的帧的解码器仍是不实用且不经济的。

为了解决这个问题,在每个框架中又定义了“级别”。级别是一个对比特流各参数进行限定的集合。这些限定可能是一些简单的数字上的约束,也可能是以几个参数的算术组合的形式提出的(例如帧宽与帧高及帧速的乘积的约束)。

与本规范相一致的比特流使用一种普通的语法。为了实现一个完整语法的子集,比特流中包括了指示现存或即将出现的句法元素的标志。这样,规定语法上的限制(即定义一个框架)就可以通过规定那些指示现存的或即将出现的句法元素的标志来实现。

## II -0.4 分级和不可分级语法

完整的语法可分为两类。一类是不可分级语法,是ISO/IEC11172-2定义的语法的一个超集。不可分级语法的主要特点是它有对隔行扫描的视频信号的额外压缩方法。第二类是可分级语法,主要特点是可以从分块的比特流中分割重构出有用的视频信号,这可将整个比特流分成两个或更多的层次来实现,从一个独立的基本层开始并逐渐加入一些附加的层。基本层可以用不可分级语法,或在某些情况下参考ISO/IEC 11172-2的语法。

### II -0.4.1 不分级语法简介

不分级语法中定义的编码表示方法在保持良好的图象质量的同时可以实现很高的压缩率。由于编码过程中并不保持采样值的精确值,因此这种算法并不是无损的。在特定的比特率情况下要获得很好的图象质量就要求有很高的压缩比,而这仅靠帧内编码是达不到的。高质量的图象和高压缩比与随机访问编码的比特流之间应达到一种平衡,而对各种技术的选择正是基于这一要求。

有许多种技术被用来实现高压缩比。算法中首先用到了基于块的运动补偿来减少时域冗余性。运动补偿既应用于从过去帧预测当前帧,也用于从过去帧和未来帧进行插值预测。图象中的每个 $16 \times 16$ 的象素块都定义有运动向量。差值信号,即预测误差,进一步用离散余弦变换(DCT)来压缩,以求在其进行丢弃某些不重要的信息的不可逆量化过程之前去除其空间相关性。最后,运动向量和预测误差的DCT信息合起来采用可变长码来进行编码。

#### II -0.4.1.1 时域处理

由于随机访问和高效压缩这两个要求之间的冲突,引入了三种主要的图象类型的定义。内部编码图(I-图)的编码不需参考别的图,这种类型的图提供了对编码图象数据序列的访问点,使解码可由此点开始,但这种类型的图仅采用了最普通的压缩方法。预测编码的图(P-图)采用更有效的编码方法,它使用运动补偿,通过过去的I-图或P-图来预测,一

般也可作为后面预测的参考图。双向预测编码的图(B-图)实现了最高程度的压缩,但需要使用过去的和未来的参考图来进行运动补偿。双向预测编码的图不能再作为别的预测的参考图(除非在空域分级增强层中将结果图作为参考)。序列中这三种类型的图的组织是非常灵活的,由编码器根据实际情况来选择其安排。图 II -0-1 说明了这三种图的关系。

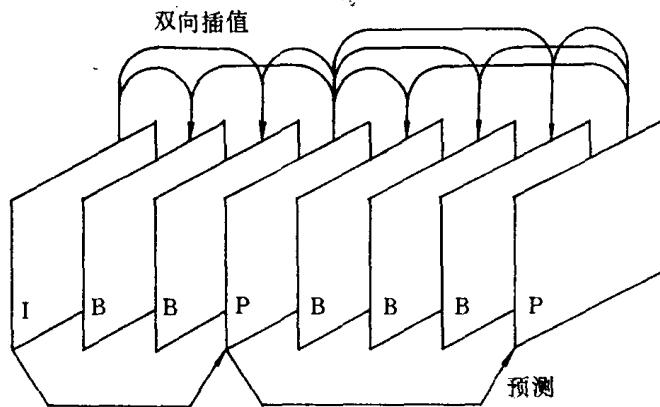


图 II -0-1 时域图象结构

#### II -0. 4. 1. 2 隔行扫描视频的编码

隔行扫描视频信号的每一帧均由两个场组成,用场周期来划分。本标准既允许一个帧被编码成一幅图,也允许两个场被编码为两幅图。帧编码或场编码的选择可以帧为单位来进行。帧编码主要用于包含很多细节,但只有很少的运动信息的图象。场编码(第二场的编码可由第一场来预测)更适用于有快速运动的图象。

#### II -0. 4. 1. 3 运动描述-宏块

就像在 ISO/IEC 11172-2 中一样,选择  $16 \times 16$  的宏块作为运动补偿的单元是将其提供的运动信息对编码的好处和描述它所需的开销进行折衷的结果。每个宏块的预测均可采用许多种不同的时域预测方法中的一种来实现。例如,对于帧编码,从以前的参考帧所作的预测既可以是基于帧的,也可以是基于场的。取决于宏块的类型,运动向量信息和其它附带信息与压缩过的预测误差在每个宏块中是一起进行编码的。运动向量采用对前面最后一个编码的运动向量的差分编码来表示,并使用可变长码。运动向量可以使用的最长的码的长度是可编程的(基于一幅一幅的图象),这样,在保持系统的正常运行的同时,最苛刻的应用要求也可被满足。

应由编码器负责计算正确的运动向量,本标准对于如何计算不加阐述。

#### II -0. 4. 1. 4 空域冗余的减少

原始图象和预测差值均存在着很高的空间冗余性。本标准采用基于块的 DCT,并使用基于视觉的加权量化和行程码。经过预测或插值的运动补偿之后的图象再被分为  $8 \times 8$  的块,这些块被变换到 DCT 域之后进行加权的量化。量化后的系数有许多变为零,因此可以使用二维的行程码和可变长码来对剩余的系数进行有效的编码。

#### II -0. 4. 1. 5 色度格式

除了支持 ISO/IEC 11172-2 中所支持的  $4 : 2 : 0$  格式,本规范还支持  $4 : 2 : 2$  和

4 : 4 : 4 的色度格式。

## II -0.4.2 分级的扩展

本规范中的分级工具被设计用来支持那些单层视频所不能支持的应用系统。在那些众所周知的应用领域中,我们主要涉及视频通信、异步传输模式网络(ATM)上的视频、视频标准的交互工作、具有多路的空域、时域以及质量分辨率的视频服务级别、具有嵌入式 TV 的 HDTV 以及允许向时域高分辨率迁移的 HDTV 系统等。虽然对于分级视频来讲,一个简单的解决方法是基于多路独立编码的视频重构存储转发,也就是联播技术,但一种更为有效的方法是视频分级编码。在这种编码中,分配给已给定的视频重构的带宽可以被部分地复用在下一个重构视频中。在分级编码中,假定给定一个已编码的比特流,具有不同复杂性的解码器可以解码并显示出适当的重构视频。一个分级视频解码器与单层解码器相比增加了复杂性。对应着几种具有不同复杂性的相对独立的应用,本标准提供了几种不同形式的分级方法。提供的基本的分级工具有:数据分割、SNR 分级、空域分级和时域分级。此外,(本标准)也支持这些基本分级工具的组合,并作为一种混合分级提供给我们。在基本分级方法中,定义了低层和增强层这两个层次,而在混合分级中支持三个以上的层次。表 I -0-1 到表 II -0-3 提供了一些不同的分级方法的应用实例。

表 II -0-1 SNR 分级的应用

低 层	增 强 层	应 用
ITU-R-601	与低层具有同样的分辨率和格式	对于标准 TV 提供两种质量服务
高清晰度	与低层具有同样的分辨率和格式	对于 HDTV 提供两种质量服务
4 : 2 : 0 高清晰度	4 : 2 : 2 色度联播	视频产生/分配

表 II -0-2 空域分级的应用

基	增 强	应 用
逐行(30Hz)	逐行(30Hz)	CIF/SCIF 兼容或分级
隔行(30Hz)	隔行(30Hz)	HDTV/SDTV 分级
逐行(30Hz)	隔行(30Hz)	ISO/IEC 11172-2/与本规范兼容
隔行(30Hz)	逐行(60Hz)	向 HR 逐行 HDTV 迁移

表 II -0-3 时域分级的应用

基	增 强	更 高	应 用
逐行(30Hz)	逐行(30Hz)	逐行(60Hz)	向 HR 逐行 HDTV 迁移
隔行(30Hz)	隔行(30Hz)	逐行(60Hz)	向 HR 逐行 HDTV 迁移

### II -0.4.2.1 空域的分级扩展

空域分级是我们打算应用于涉及到电讯、视频标准交替工作、视频数据库浏览、

HDTV 和 TV 交替工作等视频应用中的一个工具,即我们需要一个至少有两个空间分辨率层次的视频系统。空域分级包括从一个视频源产生两个空间分辨率的视频层,这样,低层自己编码来提供基本的空间分辨率;增强层利用这个空域插值的低层来生成输入视频源的全分辨率。低层和增强层均可使用本标准中的编码方法,或者低层使用 ISO/IEC 11172-2 中的方法而增强层使用本标准中的方法。后一种情况中,通过促进各种编码标准的交替工作,我们可以获得更长远的益处。此外,空域分级提供了对于每层中用到的视频格式的选择的灵活性。空域分级的另一个好处是它对于传送错误的复原能力,这是由于对于更重要的低层数据我们可以使用出错率低的信道,而对于相比之下要求不太严格的增强层我们可使用较差的信道。

#### **II -0. 4. 2. 2 SNR 分级扩展**

SNR 分级是我们打算用在涉及到电讯、多质量视频服务、TV 和 HDTV 等这些视频应用中的一个工具,即我们需要一个至少有两个视频质量层的视频系统。SNR 分级包括从一个视频源产生两个具有相同的空间分辨率和不同的视频质量的视频层,这样,低层自己编码来提供基本视频质量,而增强层被编码来增强低层。当把增强层加回到低层时,它将重新产生输入视频的一个高质量重构。低层和增强层均可使用本标准(中的编码方法),或者低层使用 ISO/IEC 11172-2 标准而增强层使用本标准。SNR 分级的另一个优点是它对于传送错误所提供的很高程度的复原能力,这是基于对更重要的低层数据可以使用出错率低的信道,而对于要求不太严格的增强层可以使用出错率较高的信道。

#### **II -0. 4. 2. 3 时域分级扩展**

时域分级应用于从电信到 HDTV 的不同视频应用中,对它们来说,从低时域分辨率系统到高时域分辨率系统的转变是必要的。在多数情况下,低时域分辨率视频系统可能是现存的系统或者是廉价的早期生成系统,它们有着逐渐引入高级系统的能力。时域分级包括将视频帧分割成层,这样,低层自我编码以提供基本时域率,增强层通过低层进行时域预测编码,最后通过对这些层次的解码和时域多路复合来产生视频源的全时域分辨率。低时域分辨率的系统可以仅通过对低层的解码来提供基本时域分辨率,而未来的更高级的系统可以在保持与早期系统的交互工作的前提下,对两个层次一起解码。时域分级的另一个好处是它对于传送错误的很高的复原能力,这是基于对于更重要的低层数据我们可以使用出错率低的信道,而对于相比之下要求不太严格的增强层使用较差的信道。

#### **II -0. 4. 2. 4 数据分割扩展**

在 ATM 网络、陆上广播、磁性介质等应用中,我们可以有两个通道以供视频流存储和发送,在这里我们就要用到数据分割。在这两个通道间比特流被分割,这样,比特流中的重要部分(例如头、运动向量、DC 系数)可以在错误率低的通道中传送,而不太重要的部分(例如 DCT 系数)可以在较差的信道中传送。由此,信道错误造成的损失被减到最小,因为对于比特流的重要部分我们采取了更好的保护。对于不具有处理数据分割比特流能力的解码器,任何一个通道中的数据都不能被解码。

## II -1 范 围

本推荐 | 国际标准详述了数字存储媒体和数字视频通信中的图象信息的编码描述和解码过程。本标准支持固定比特率传送、可变比特率传送、随机访问、信道跨越、分级解码、比特流编辑以及一些特殊功能，如：快进播放、快退播放、慢动作、暂停和画面静止。本推荐 | 国际标准与 ISO/IEC 11172-2 向前兼容并与 EDTV, HDTV, SDTV 格式向上或向下兼容。

本推荐 | 国际标准主要应用于数字存储媒体、视频广播和通信。存储媒体可以直接与解码器相连，或者通过总线、局域网、电信通信线路等通信手段与其相连。

## II -2 标准参照

本文档通过参考下面的ITU-T建议和标准中的条款而形成自己的推荐|国际标准条款。在正式出版时,指明的版本是有效的。所有的建议和标准都会被修改,因此我们希望所有基于本推荐|国际标准而达成协议的团体应考虑使用下面指明的标准的最新版本。IEC 和 ISO 的成员保存有当前有效的国际标准的注册登记。TSB(电信标准化局)保存有当前有效的ITU-T的建议的一览表。

- Recommendations and reports of the CCIR, 1990  
XVIIth Plenary Assembly, Dusseldorf, 1990      Volume XI-Part 1  
Broadcasting Service(Television) Rec. 601-2“Encoding parameters of digital television for studios”
- CCIR Volume X and XI Part 3 Recommendation 648: Recording of audio signals
- CCIR Volume X and XI Part 3 Report 955-2: Sound broadcasting by satellite for portable and mobile receivers, including Annex IV Summary description of advanced digital system II
- ISO/IEC 11172(1993)“Information technology——Coding of moving picture and associated audio for digital storage media at up to about 1.5 Mbit/s”
- IEEE Standard Specifications for the Implementations of 8 by 8 Inverse Discrete Cosine Transform, IEEE Std 1180-1990, December 6, 1990
- IEC Publication 908;198, “CD Digital Audio System”
- IEC Standard Publication 461 Second edition, 1986, “Time and control code for video tape recorders”
- ITU-T Recommendation H. 261 (Formerly CCITT Recommendation H. 261) “Codec for audiovisual services at px64 kbit/s” Geneva, 1990
- ISO/IEC 10918-1 | ITU-T Rec. T. 81(JPEG) “Digital compression and coding of continuous tone still images”

## II -3 定义

以下的定义用于本推荐|国际标准。

- (1) **AC 系数**:一维或两维频率分量不为零的 DCT 系数。
- (2) **B-场图**:一个场结构的 B-图。
- (3) **B-帧图**:一个帧结构的 B-图。
- (4) **B-图、双向预测编码图**:由过去的和/或未来的参考场或帧,而采用运动补偿预测编码的图。
- (5) **向后兼容**:如果为旧的编码标准所设计的解码器可以部分或全部地对根据新的编码标准而产生比特流进行解码,那么新的编码标准称为与旧的编码标准向后兼容。
- (6) **后向运动向量**:用于从较晚显示的参考帧或参考场进行运动补偿的运动向量。
- (7) **双向预测编码图;B-图**:见(4)。
- (8) **比特率**:编码的比特流从存储媒体传送到解码器的输入端口的速率。
- (9) **块**:一个 8 行 8 列的样点矩阵,或者 64 个 DCT 系数(源、量化的或反量化的)。
- (10) **底场**:组成一帧的两场中的一个。底场的每一行的空间位置紧接在其对应的顶场的行的下面。
- (11) **字节定位**:在一个编码的比特流中,如果每位的位置正好是从流的第一位算起的 8 个位的倍数,那么这个位称为字节定位的位。
- (12) **字节**:8 个位的序列。
- (13) **信道**:存储或传送按本标准所构成的比特流的数字媒体。
- (14) **色度格式**:定义一个宏块中的色度块的数目。
- (15) **色度联播**:分级的一种类型(SNR 分级的子集),增强层仅包含经过修正的 DC 系数和所有的 AC 系数。
- (16) **色度(分量)**:代表与比特流中定义的元色形式相关的两个色差信号之一的矩阵、块或单样点。用于色度信号的符号是 Cr 和 Cb。
- (17) **编码的 B-帧**:一个 B-帧图或一对 B-场图。
- (18) **编码的帧**:编码的帧是一个编码的 I-帧,或一个编码的 P-帧,或一个编码的 B-帧。
- (19) **编码的 I-帧**:一个 I 帧图或一对场图,第一场是一个 I-图,第二场是一个 I-图或一个 P 图。
- (20) **编码的 P-帧**:一个 P-帧图或一对 P-场图。
- (21) **编码的图**:编码的图由一个图象头,紧接着的可选扩展和接下来的图象数据组成。编码的图可能是一个帧图或一个场图。
- (22) **编码的视频比特流**:由本标准定义的一幅图或多幅图的序列的编码表示。
- (23) **编码的顺序**:图象的传送和解码顺序。这个顺序不一定要与显示顺序相同。

- (24) **编码表示**:以编码的形式表示的一个数据单元。
- (25) **编码参数**:刻画编码的视频比特流的用户定义参数集合。比特流由编码参数描绘特性。解码器由其可以解码的比特流来刻画特性。
- (26) **分量**:组成一幅图的三个矩阵(一个亮度和两个色度)中的一个矩阵或块或单样点。
- (27) **压缩**:表示某项数据的位的数目的减少。
- (28) **固定比特率的编码的视频**:具有一个固定的平均比特率的压缩视频比特流。
- (29) **固定比特率**:从对编码的比特流处理的开始到结束,比特率是一个常数。
- (30) **数据单元**:编码前或解码后的一个数据项表示。
- (31) **数据分割**:为了错误的复原而将一个比特流分成两个分离的比特流。解码前这两个比特流将合而为一。
- (32) **D-图**:只在 ISO/IEC 11172-2 中用到的一种图。
- (33) **DC 系数**:两维的频率均为零的 DCT 系数。
- (34) **DCT 系数**:一个特定的余弦函数基的幅值。
- (35) **解码器输入缓冲器**:视频缓冲检验器中指明的先进先出(FIFO)缓冲器。
- (36) **解码器**:一个解码过程的具体体现者。
- (37) **解码(过程)**:本标准中所定义的过程,读进输入的编码的比特流然后产生解过码的图象或音频样本。
- (38) **反量化**:在量化的 DCT 系数被解码之后以及被反余弦变换之前,对其进行的再次变化的过程。
- (39) **数字存储媒体;DSM**:一个数字的存储、转发设备或系统。
- (40) **离散余弦变换;DCT**:正的离散余弦变换或反离散余弦变换中之一。DCT 是可转置的、离散正交变换。反变换在本标准的附录 I-A 中定义。
- (41) **显示顺序**:解码后的图象显示的顺序。通常这与图象在编码器输入端的表示顺序一致。
- (42) **编辑**:一个或多个编码的比特流经处理而产生一个新的编码比特流的过程。编辑的比特流必须满足本标准中的定义和要求。
- (43) **编码器**:一个编码过程的具体体现者。
- (44) **编码(过程)**:读入一个输入图象流或音频样本流,从而产生一个如本标准所定义的正确编码流的过程,这个过程并未在本标准中定义。
- (45) **快进播放**:按显示顺序并以快于实时播放的速度来显示一个或部分图象序列的过程。
- (46) **快退播放**:按显示顺序的反序并以快于实时播放的速度来显示图象序列的过程。
- (47) **场**:对于隔行视频信号,“场”就是一帧中的相隔行的集合。所以一个隔行帧由两场组成,一个顶场和一个底场。
- (48) **场周期**:帧速率的二倍的倒数。
- (49) **场图;场结构图**:场结构图就是一个具有“顶场”或“底场”的图结构的编码的图。

(50) **标志**:一个变量,仅可取本标准中定义的两值中的一值。

(51) **禁止**:在定义编码流的章节中所使用的“禁止”这一术语指明某值是永远不可被使用的。这主要是用来防止与起始码的冲突。

(52) **强制刷新**:为确保编码器和解码器的反 DCT 过程间的不匹配错误不致于急剧增大,而对宏块不时地进行帧内编码的过程。

(53) **向前兼容**:如果按新的编码标准所设计的解码器可以对旧的编码标准的比特流进行解码,那么这个新的编码标准称为与那个旧的编码标准向前兼容。

(54) **前向运动向量**:用于从较早显示的参考帧或参考场进行运动补偿的运动向量。

(55) **帧**:一个帧中有一行行的视频信号的空域信息。对于逐行视频,这些行包含着从某即时时间开始并按后续行直到帧底部的所有样点。对于隔行视频,一帧由两场组成,一个顶场,一个底场。场之间相隔一个场周期而交替出现。

(56) **帧周期**:帧速率的倒数。

(57) **帧图;帧结构图**:帧结构图是用与帧相等的 picture\_structure 编码的图。

(58) **帧速率**:帧从解码器输出的速率。

(59) **未来参考帧(场)**:未来参考帧(场)是在显示顺序中比当前图迟些时候出现的帧(场)。

(60) **头**:编码比特流中的一个数据块,包含一些从属于后面的编码数据的编码表示数据单元。

(61) **混和分级**:混和分级是两种或更多的分级方法的组合。

(62) **隔行(交叉)**:一种传统的电视帧的性质,其中交替的行及时地表示出不同的景物。在隔行帧中,有一场要被先显示,这个场被称为第一场。第一场可以是顶场也可以是底场。

(63) **I-场图**:一个场结构的 I-图。

(64) **I-帧图**:一个帧结构的 I-图。

(65) **I-图;帧内编码图**:一个仅使用自身信息编码的图。

(66) **帧内编码**:仅使用自身信息的宏块或图的编码。

(67) **帧内编码图;I-图**:见(65)。

(68) **级别**:本规范的一个特定框架中的参数所取值的限定集合。一个框架可以包含一个或多个级别。

(69) **亮度(分量)**:与比特流中定义的元色相关,并描述信号的单色表示的一个矩阵、块或单样点。亮度所用符号是 Y。

(70) **宏块**:四个  $8 \times 8$  的亮度数据的块和两个(4 : 2 : 0 色度格式)或四个(4 : 2 : 2 色度格式)或 8 个(4 : 4 : 4 色度格式)的  $8 \times 8$  色度数据的块。 $8 \times 8$  色度数据是从一个  $16 \times 16$  的亮度分量中得来的。宏块有时被用来指数据样本,有时指样值和本规范中定义的语法的宏块头中的数据单元的编码表示。从上下文中可以明确其应用。

(71) **运动补偿**:为提高样本值的预测效率而使用运动向量。预测使用运动向量来提供相对于包含已解码样值的过去和/或未来的参考帧(场)的偏距,那些解码样值被用来形成预测误差信号。

(72) **运动估计**:在编码过程中估算运动向量。

(73) **运动向量**:运动补偿所用的一个二维向量,它提供了从当前图的原点到参考帧(场)的原点位置的偏距。

(74) **非帧内编码**:一个宏块或图的编码,同时使用自身信息和其它时间出现的宏块和图的信息。

(75) **P-场图**:一个场结构的 P-图。

(76) **P-帧图**:一个帧结构的 P-图。

(77) **P-图;预测编码图**:由过去的参考帧(场)使用运动补偿预测编码的图。

(78) **参数**:本规范的语法中定义的可在很大范围内取值的变量。只可取两值中一值的变量,是标志而不是参数。

(79) **过去参考帧(场)**:一个过去参考帧(场)是在显示顺序中早于当前图的参考帧(场)。

(80) **图**:源、编码的或重构的图象数据。一个源或重构图包含三个分别代表亮度和两个色度信号的 8 位数字的矩形的阵。对于逐行扫描视频,图与帧一致;而对于隔行扫描视频,一幅图指一帧,或与上下文有关的这帧的顶场或底场。

(81) **预测**:使用预测值来提供现在正在解码的样值或数据单元的估计。

(82) **预测编码图;P-图**:见(77)。

(83) **预测误差**:样本或数据单元的实际值与其预测值之间的差别。

(84) **预测值**:以前解码的样本值或数据单元的线性组合。

(85) **框架**:本标准中定义的语法的子集。

注意 在本标准中“框架”一词将按上面定义来使用。它不可和这个词的其它定义相混淆,尤其是它不具有 JTC1/SGFS 中的定义。

(86) **逐行**:电影帧的性质。帧中所有样点及时地表现同一景物。

(87) **量化矩阵**:反量化器使用的 64 个 8 位值的集合。

(88) **量化的 DCT 系数**:反量化前的 DCT 系数。量化的 DCT 系数的变长码表示作为压缩视频的比特流的一部分而传送。

(89) **量化器比例**:在比特流中编码,由解码过程用来反量化变比的比例因子。

(90) **随机访问**:在一个任意点开始读入并解码已编码的比特流的过程。

(91) **重构帧**:一个重构帧由三个 8 位数据的矩形的阵组成,一个代表亮度信号,另两个代表色度信号。重构帧通过编码帧解码而获得。

(92) **重构图**:重构图通过对编码图的解码获得。重构图是一个重构帧(当对一个帧图解码时),或是重构帧的一个场(当对一个场图解码时)。如果编码的图是一个场图,那么重构图是重构帧的底场或顶场。

(93) **参考场**:参考场是重构帧的一个场。当 P-图和 B-图被解码时,参考场被用于向前和向后预测。值得注意的是,当 P-图的场被解码时,编码帧的第二个 P-图场的预测,使用这帧的第一个已重构的场作为参考场来进行预测。

(94) **参考帧**:参考帧是一个以 I-帧或 P-帧的形式编码的重构帧,当 P-图和 B-图被解码时,参考帧被用于前向和后向预测。

(95) **保留**:当“保留”这个术语用于定义编码的比特流的章节中时,表明它可能在将来的 ISO/IEC 定义的扩展中被用到。

(96) **样本高宽比**:(缩写为 SAR)。它代表了样本间的距离。它定义为(在本规范中)由帧中亮度样本竖直行距除以亮度样本水平样本间距。即它的单位为(米/行)÷(米/样本)。

(97) **分级**:分级是一个解码器对一个有序的比特流集合进行解码并生成一个重构序列的能力。在子集被解码的同时,有用的视频信号被输出。可被解码的最小子集称为基本层,是集合中的第一个比特流。集合中所有其它的比特流均被称为增强层。在对一个增强层寻址的时候,“低层”指的是增强层前面的比特流。

(98) **边际信息**:比特流中控制解码器所必需的信息。

(99) **跳过的宏块**:没有编码数据的宏块。

(100) **组块**:一系列宏块。

(101) **SNR 分级**:一种分级类型,其中增强层仅包含编码的低层 DCT 系数的修正值。

(102) **空域分级**:一种分级类型,增强层从低层抽取样本数据进行预测,并不使用运动向量。各种层可以有不同的帧尺寸,帧速率和色度格式。

(103) **起始码**[系统和视频]:嵌入在编码比特流中的 32 位单一码。它们被用于几种目的,包括辨别编码语法中的一些结构。

(104) **填充(位);填充(字节)**:可以插入到编码的比特流中的编码字,在解码过程中将被舍弃。它们用于增加数据流的比特率。

(105) **时域分级**:分级的一种类型,其中增强层从低层抽取数据样本,并利用运动向量进行预测。各层具有相同的帧尺寸和色度格式,但可能有不同的帧速率。

(106) **顶场**:组成一个帧的两个场之一。顶场的每行在空间上都位于对应的底场的行上面。

(107) **可变比特率**:在对一个编码比特流的解码过程中,比特率可随时间变化的一种操作。

(108) **变长码;VLC**:一个可逆的编码过程,对频度高的事件定为短码字,而对频度低的事件定长码字。

(109) **视频缓冲检验器;VBV**:概念上与编码器输出相连的假想解码器。它的目的是对编码器或编辑过程产生的数据速率的变化提供一个限制。

(110) **视频序列**:编码视频比特流中的最高层语法结构。它包含了一个编码帧序列。

(111) **z 型扫描顺序**:DCT 系数特定的从低频到高频的顺序排序。

## II -4 缩写和符号

本标准中使用的数学操作符与 C 程序语言中使用的类似。然而，整数除法中的取整和四舍五入有特别定义。计数和循环一般都从零开始。

### II -4.1 算术运算符

+	加法
-	减法(二元算符)或取反(一元算符)
++	增 1
--	减 1
×   * }	乘法
^	幂
/	零向取整的整数除法，例如， $7/4$ 和 $-7/-4$ 取整为 1，而 $-7/4$ 和 $7/-4$ 取整为 -1
//	四舍五入到最接近的整数的整数除法。整数的半值按远零方向四舍五入，除非特别定义。例如， $3//2$ 入到 2， $-3//2$ 入到 -2
DIV	负无穷方向取整的整数除法。例如， $3 \text{ DIV } 2$ 得 1， $-3 \text{ DIV } 2$ 得 -2
÷	用在数学等式中表示除法，不用取整或四舍五入
%	取模算符。仅为正数定义
Sign()	$\text{Sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$
Abs()	$\text{Abs}(x) = \begin{cases} x & x >= 0 \\ -x & x < 0 \end{cases}$

### II -4.2 逻辑算符

	逻辑或
&&	逻辑与
!	逻辑非

### II -4.3 关系算符

> 大于  
>= 大于等于  
< 小于  
<= 小于等于  
== 等于  
!= 不等于  
`max [,...]` 自变量表中的最大值  
`min [,...]` 自变量表中的最小值

### II -4.4 位 算 符

& 与  
| 或  
>> 带符号位右移  
<< 右边补零左移

### II -4.5 赋 值

= 赋值算符

### II -4.6 助 记 符

下面的助记符被定义用来描述编码比特流中的不同的数据类型。

- bslbf** 位串, 左位在前。这里“左”指位串在本标准中的书写顺序。位串是单引号内的 0 和 1 的串, 如 ‘1000 0001’。位串中的空白是为了阅读方便, 没有什么意义
- uimsbf** 无符号整数, 最高有效位在前
- simsbf** 带符号整数
- vlclbf** 变长码, 左位在前。这里“左”指 VLC 的书写顺序。多字节码字的字节顺序是最有效位在前

### II -4.7 常 量

$\pi$  3.1415925359...

e 2.71828182845...

## II -5 约定

### II -5.1 比特流语法的描述方法

在 II -6.2 节中,将对解码器要恢复的比特流进行描述。比特流中的每个数据项都有醒目的类型,它由名字、类型助记符和传送顺序来描述。

比特流中的一个解码的数据项引起的动作取决于那个数据项的值和前面已解码的数据项。数据项的解码和解码中用到的状态变量在 II -6.3 节中进行描述。下面的结构用来表示数据元素以一般类型出现时的情况:

while (condition){ <b>data_element</b> ... }	如果条件为真,那么下面的数据元素组紧接着出现在数据流中。这将一直重复到条件非真
do { <b>data_element</b> ... } while (condition)	数据元素至少总要出现一次
if (condition){ <b>data_element</b> ... } else { <b>data_element</b> ... }	如果条件为真,第一组数据元素紧接着出现在数据流中  如果条件不为真,第二组数据元素紧接着出现在数据流中
for(i=0;i<n;i++){ <b>data_element</b> ... }	这组数据元素出现 n 次。数据元素组中的条件结构取决于循环控制变量 i。i 第一次被置为 0,第二次被加到 1,如此进行下去
/* comment... */	注释可被完全删去,而丝毫不改变语法

这里的语法使用‘C-编码’的约定:具有非零值的一个变量或表达式对等于一个真值的条件。在许多情况下,条件中使用一个字符串。例如:

```
if (scalable_mode == "spatial scalability")...
```

在那样的情况下,字符串用来描述 II -6.3 节中的比特流元素的值。在本例中,我们看到“spatial scalability”在表 II -6-10 中有定义,由两位二进制数‘01’来表示。

如所述,数据单元的组中还可以包含嵌套的条件结构。为了简洁起见,在只有一个数据单元时,{}将被省略。

data\_element[n] data\_element[n]是数据的数组中的第 n+1 个元素。

data\_element[m][n] data\_element[m][n]是数据的二维数组中的第 m+1,n+1 个元素。

尽管这个语法以程序的方式表述,我们仍不能认为 I-6.2 节完整地描述了一个令人满意的解码过程。它只是特别定义了一个无错的输入比特流。实际的解码器还必须包含以下方法:为了正确开始解码而找到起始码,在解码时辨别错误、删除和插入。辨别这些情况的方法以及采取的动作并未被标准化。

## II -5.2 函数的定义

图象编码算法的几个工具函数按下述定义。

### II -5.2.1 bytealigned()函数的定义

函数 bytealigned()返回 1,如果现在的位置是在一个字节的边界,即当比特流中的下一位是一个字节中的第一位。否则函数返回 0。

### II -5.2.2 nextbits()函数的定义

函数 nextbits()将一个位串和比特流中待解码的下面的位来比较。

### II -5.2.3 next\_start\_code()函数的定义

next\_start\_code()函数去掉零位和零填充字节,并定位下一个起始符。

next_start_code()	位数	助记符
while(! bytealigned())		
zero_bit	1	“0”
while(nextbits() != '0000 0000 0000 0000 0000 0001')		
zero_byte	8	“0000 0000”
}		

此函数检查当前位置是否为字节边界,如果不是,说明有零填充位出现。直到起始码之前,下面可能出现任意数目的零字节。因此,起始码总是字节定位的,并且可以在前面填充任意数目的零位。

## **II -5. 3 保留、禁止和 marker\_bit**

术语“保留”和“禁止”用在编码比特流中的几个域的某些值的描述中。

术语“保留”指明这个值可能在将来的 ISO/IEC 定义的扩展中用到。

术语“禁止”指明了一个永远不会被用到的值(通常是为了防止出现与起始码相冲突的码)。

术语“marker\_bit”指明了一个不可为零值的点位的域。这些 marker 位在语法中的几处出现,来防止与起始码相冲突。

## **II -5. 4 算术精确性**

为了减小本规范的各种实现过程之间的差异,特别提出以下针对算术精确性的规则:

(1) 在算术精确性没有被特别强调时,如在 IDCT 的计算中,必须保证有足够的精度以防在最后的整数值上出现重大差错。

(2) 在用两个点给出值域时,如果使用方括号,表示包括边界点;如果使用“小于”(<)和“大于”(>)符,表示不包括边界点。如,[a.. b>表示从 a 到 b,包括 a 而排除 b。

## II -6 视频比特流的语法和语义

### II -6. 1 编码视频数据的结构

编码的视频数据由一称作层的比特流序列组成。如果仅有一层,那么这编码的视频数据称为不分级的视频比特流。如果有两层或更多层,那么这些编码的视频数据称为被划分等级。

第一层(有序序列中的)称为基本层,它总是可以被独立解码的。对基本层的解码过程的描述参看本规范的 II -7. 1 到 II -7. 6 节和 II -7. 12 节, II -7. 10 节中描述的数据分割情况除外。

其它的层被称为增强层,它只能由基本层开始,与所有的低层(在序列中的前面的层)一起解码。对划分等级的解码过程的描述参看本标准的 II -7. 7 到 II -7. 11 节。

对于多层的复合的描述参看 ITU-T Rec. xxx | ISO/IEC 13818-1。

等级划分中的基本层可与本规范一致或与其它标准(如 ISO/IEC 11172-2)一致。细节见 II -7. 7 到 II -7. 11。增强层与本规范一致。

在所有情况中,除了数据分割以外,基本层不包含 sequence\_scalable\_extension()。增强层总包含 sequence\_scalable\_extension()。

通常视频比特流可被看作语法的等级体系,其中各语法结构又包含一个或更多的从属结构。例如,结构“picture\_data()”包含一个或更多的语法结构“slice()”,而它又包含一个或更多的有序的结构“macroblock()”。

这个结构很像 ISO/IEC 11172-2 中所用到的。

#### II -6. 1. 1 视频序列

编码的比特流中的最高语法结构就是视频序列。

一个视频序列以一个序列头开始,后面可选地跟着一组图象的头和一个或更多的编码帧。编码帧在编码比特流中的顺序就是解码器处理它们的顺序,但并不一定就是显示顺序。视频序列以一个 sequence\_end\_code 终止。在一个视频序列的不同地方,某一特定的编码帧的前面可能会有一个重复的序列头或一组图象的头,或两者都有。(当一个重复的序列头和一组图象的头紧挨着位于某一特定的编码帧前面的时候,图象组头应跟在重复序列头的后面。)

##### II -6. 1. 1. 1 逐行和隔行序列

本规范对逐行和隔行序列的编码均可处理。

对于隔行序列来说,解码过程的输出由一系列以场周期按时间分开的重构场组成。一帧的两个场可以被分开编码(场图)。或者,两个场可以作为一个帧而共同编码(帧图)。帧图和场图均可被用在一个单视频序列中。

在逐行序列中,序列的每个图都是一个帧图。在解码过程的输出端,序列由一系列以帧周期按时间分开的重构帧组成。

#### II -6. 1. 1. 2 帧

一帧由三个整数矩阵组成:一个亮度矩阵(Y)和两个色度矩阵(Cb 和 Cr)。

这些 Y,Cb 和 Cr 分量以及红、绿、蓝( $E'_R$ , $E'_G$ , $E'_B$ )三元色信号(模拟)之间的关系;这些元色的色度和源帧的变换特性都可以在比特流中加以说明(或以一些其它方法说明)。这个信息不影响解码过程。

#### II -6. 1. 1. 3 场

代表一帧的三个整数矩阵的隔行样点组成场。

帧是一个顶场和一个底场的合并。顶场是包含三个矩阵的各自最顶行的场,底场则是另一个场。

#### II -6. 1. 1. 4 图

一个重构图通过对一个编码图的解码获得。一个编码图也就是由一个图象头、紧跟后面的可选扩展以及图数据组成。一个编码图可以是一个帧图或一个场图。一个重构图可以是一个重构帧(当对一个帧图解码时),或是一个重构帧的一个场(当对一个场图解码时)。

##### 1. 场图

场图使用时总是成对出现的(一个顶场和一个底场),共同构成一个编码帧。组成一个编码帧的两个场图在比特流中编码的顺序应是其在解码过程输出端出现的顺序。

当编码帧的第一幅图是一个 P-图时,第二幅图也将是一个 P-图。类似地,当编码帧的第一幅图是一个 B-图时,编码帧的第二幅图也将是一个 B-图。

当编码帧的第一幅图是 I-图时,该帧的第二幅图是 P-图或 I-图。

##### 2. 帧图

当使用帧图对隔行序列编码时,这帧的两个场应互相交叉,然后整个帧作为单一的帧图编码。

#### II -6. 1. 1. 5 图类型

有三种使用不同编码方法的图。

一个内部编码(I)的图是仅使用自身信息进行编码的图。

一个预测编码(P)的图是从过去的 I-图和 P-图使用运动补偿预测进行编码的图。

一个双向预测编码(B)的图是从过去的和/或将来的 I-图或 P-图使用运动补偿预测进行编码的图。

#### II -6. 1. 1. 6 序列头

视频序列头从一个 sequence\_header\_code 开始,后面跟着一系列数据单元。本规范中 sequence\_header()后面跟着 sequence\_extension(),它包含了 ISO/IEC 11172-2 所未使用过的一些参数。当出现 sequence\_extension()时,ISO/IEC 11172-2 中定义的语法和语义就不再适用了,而本规范可以适用。

在重复的序列头中,所有的数据单元将与第一个序列头具有同样的值,除了一个可以允许的例外:量化矩阵的定义(load\_intra\_quantiser\_matrix,load\_non\_intra\_quantiser

和可选的 intra\_quantiser\_matrix 以及 non\_intra\_quantiser\_matrix)。当每一次序列头出现在比特流中时,量化矩阵都可能被修改(注意,量化矩阵也可用 quant\_matrix\_extension()修改)。

所有跟在重复的 sequence\_header()后面的 sequence\_extension()中的数据单元都应具有与第一个 sequence\_extension()中相同的值。

如果在第一个 sequence\_header()后面出现了一个 sequence\_scalable\_extension(),那么下面所有的序列头都要跟着一个 sequence\_scalable\_extension(),它的所有数据单元具有与第一个 sequence\_scalable\_extension()相同的值。反之,如果在第一个序列头和第一个图象头之间 没有出现 sequence\_scalable\_extension(),那么比特流中就不会再出现 sequence\_scalable\_extension()了。

通过对序列头的数据单元进行重复而重复第一个序列头,从而使得对视频序列的随机访问成为可能。

在编码比特流中,重复序列头可以在 I-图或 P-图的前面,但不能在 B-图前面。如果一个隔行帧作为两个分开的场图进行编码,那么重复序列图不能出现在第二场图前面。

如果比特流经过编辑,使得任何重复序列头前的数据都被去掉(或者另一种情况,对那个序列头进行随机访问),那么得到的比特流是遵守本规范的合法比特流。如果结果比特流的第一个图是 P-图,那么它有可能包含非内部宏块。既然解码过程所需的参考图不能得到,那么重构帧就可能不会完全生成。而刷新这整个帧所需的时间由所使用的刷新技术决定。

#### II-6.1.1.7 I-图和图组头

I-图用来帮助对序列随机访问。需要用到随机访问的各种应用,如快进播放或快退播放可能使用 I-图更频繁些。

I-图也用在场景切换或其它用不上运动补偿的情况下。

图象组头是可选的,它用在一个编码 I-帧的紧前头,来向解码器指明在随机访问时,紧跟在编码 I-帧后面的第一个 B-图是否能被正确重构。实际上,如果不能得到前面的参考帧,那些 B-图就不能被正确重构,除非它们仅使用后向预测。对于这一点的更准确的说明在描述 closed\_gop 和 broken\_link 的章节中,图象组头中也包含着解码过程所不使用的一个时间编码信息。在编码比特流中,图象组头后面的第一个编码帧是 I-帧。

#### II-6.1.1.8 4:2:0 格式

在这种格式中,Cb 和 Cr 矩阵在水平和垂直方向都是 Y 矩阵尺寸的二分之一。Y 矩阵的行数和(每行)样本数都是偶数。

注意 当隔行帧作为场图来编码时,由这些场图重构的图的 Y 矩阵行数只有相应帧的 Y 矩阵行数的一半。所以一个整帧的全部行数应可被 4 整除。

亮度和色差样点的位置如图 II-6-1 所示。

为了进一步说明结构组织,图 II-6-2 和 II-6-3 示出隔行帧中样点(垂直)位置。图 II-6-4 示出逐行帧样点(垂直)位置。

在隔行帧的每场中,色差样点并不位于(垂直)亮度样点的正中,这样就使得不论一帧是由一幅单一帧图表示还是两幅场图表示,帧中色差样点的空间位置是相同的。

X	X	X	X	X	X	X
O	O	O	O	O	O	O
X	X	X	X	X	X	X
-	-	-	-	-	-	-
X	X	X	X	X	X	X
O	O	O	O	O	O	O
X	X	X	X	X	X	X
-	-	-	-	-	-	-
X	X	X	X	X	X	X
O	O	O	O	O	O	O
X	X	X	X	X	X	X

× 代表亮度样本

O 代表色差样本

图 I - 6-1 亮度和色差样本的位置。4 : 2 : 0 数据

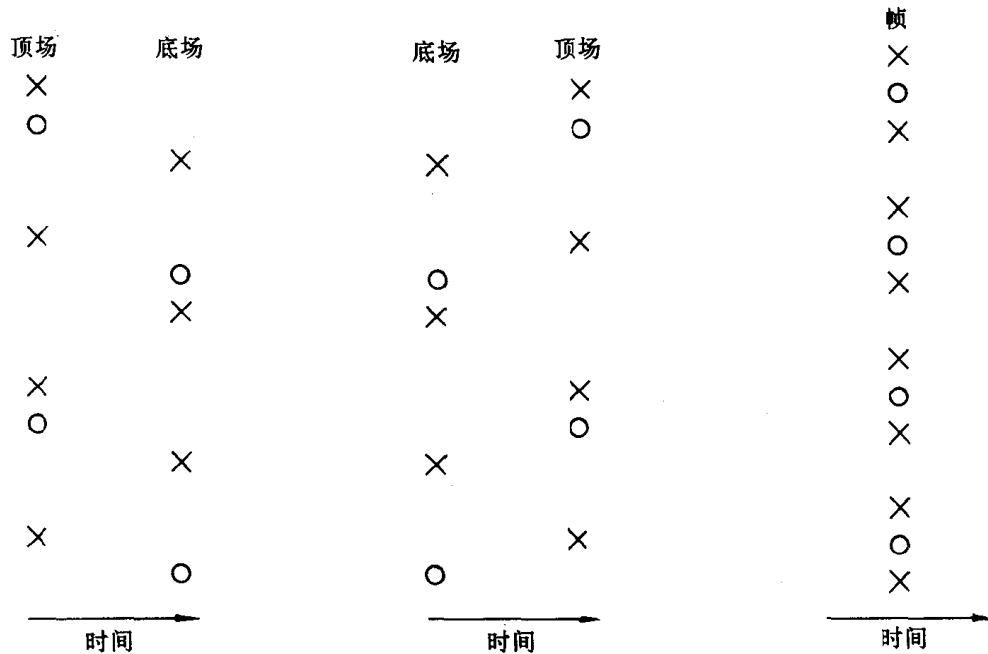


图 I - 6-2 top\_field\_first  
=1 时, 隔行帧样本的竖直  
和时域位置

图 I - 6-3 top\_field\_first  
=0 时, 隔行帧样本的竖直  
和时域位置

图 I - 6-4 逐行帧中样本的  
竖直和时域位置

#### I - 6. 1. 1. 9 4 : 2 : 2 格式

在这种格式中, Cb 和 Cr 矩阵在水平方向是 Y 矩阵尺寸的一半, 在竖直方向与 Y 矩阵具有相同尺寸。Y 矩阵(每行)有偶数样点。

注意 当隔行帧作为场图编码时, 由这些场图重构的图的 Y 矩阵行数只有相应帧的 Y 矩阵行数的一半。所以一个整帧的全部行数应可被 2 整除。

亮度和色差样点定位如图 II-6-5 所示。

为了阐明其组织结构,图 II-6-6 示出样点(竖直)位置,当帧分为两场时。

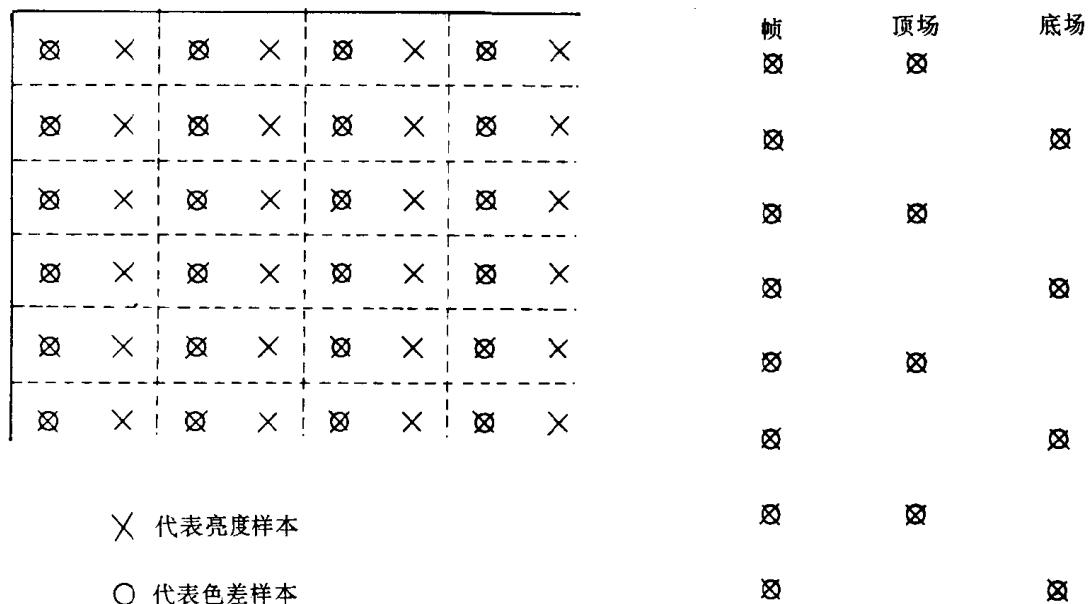


图 II-6-5 亮度和色差样本的位置。4:2:2 数据

图 II-6-6 4:2:2 和 4:4:4 数据  
样本的竖直位置

#### II-6.1.1.10 4:4:4 格式

在这种格式中,Cb 和 Cr 矩阵在水平和竖直方向都具有和 Y 矩阵相同的尺寸。

注意 当隔行帧作为场图编码时,由这些场图重构的图的 Y 矩阵行数只有相应帧的 Y 矩阵行数的一半。这样,整帧的 Y 矩阵全部行的数目应能被 2 整除。

亮度和色差样点定位如图 II-6-6 和 II-6-7 所示。

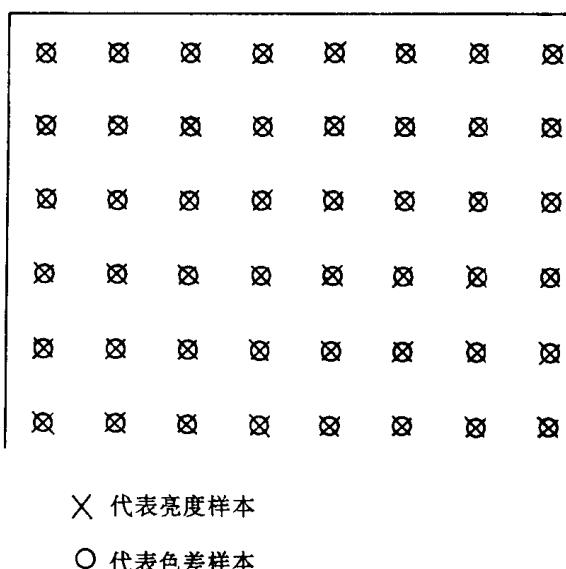


图 II-6-7 亮度和色差样本的位置。4:4:4 数据

### II -6. 1. 1. 11 帧重排

编码帧在编码比特流中的顺序就是解码器处理它们的顺序。如果序列中用到了B-图,编码帧在比特流中的顺序有时不同于解码过程输出的重构帧或场的顺序。

下面是从一个视频序列中的开始部分取出的图象的例子。在本例中,在顺序的P-帧间有两个编码B-帧,或者在顺序的I-帧和P-帧间有两个编码B-帧。帧‘1I’用来为帧‘4P’形成预测。帧‘4P’和‘1I’共同用来为帧‘2B’形成预测。由此编码序列中的编码帧的顺序应该是‘1I’,‘4P’,‘2B’,‘3B’。而解码器按‘1I’,‘2B’,‘3B’,‘4P’的顺序显示它们。

在编码器输入

1	2	3	4	5	6	7	8	9	10	11	12	13
I	B	B	P	B	B	P	B	B	I	B	B	P

在编码器输出、在编码比特流中,并在解码器的输入

1	4	2	3	7	5	6	10	8	9	13	11	12
I	P	B	B	P	B	B	I	B	B	P	B	B

在解码器输出

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

连续的编码B-帧的数目是可变的。在顺序的编码P-帧(或编码I-帧和编码P-帧之间)间可能不出现B-帧。每组连续的B帧在比特流中的顺序就是它们在解码器输出端显示的顺序。

如果序列中不包含编码的I-帧,则在序列的开始处,以及在序列中需要随机访问和错误复原的地方,都应特别小心注意。

一个序列可能不包含P-帧。

一个序列不能仅由编码B-帧组成。

### II -6. 1. 2 组块

一个组块是一系列任意数目的宏块。组块的第一个和最后一个宏块不能是跳过的宏块。一个组块至少要包含一个宏块。组块间不能重叠。图与图之间的组块位置可以改变。

组块的第一个和最后一个宏块应该在同一宏块水平行。

组块按它们将要被遇到的顺序出现在比特流中,从图象的左上角开始,按光栅扫描顺序从左到右,从上到下(本节图中以字母顺序表示)。

#### II -6. 1. 2. 1 一般的组块结构

在最一般的情况下,并不要求组块覆盖整幅图。图 II -6-8 示出了这种情况。没有被组块围住的区域不被编码,对于这种区域没有信息可供编码(在特定图中)。

如果组块不覆盖整幅图,如果这幅图随后被用来形成预测,那么预测只能由那些图上被组块围住的区域作出。保证这一点是解码器的任务。

本规范没有定义对于组块之间的区域解码器将采取什么动作。

### II -6. 1. 2. 2 受限制的组块结构

在某一限定框架的限定级别中, 将用到图 I -6-9 说明的受限制的组块结构。在这种情况下, 图象中每个宏块都将被包括在某个组块中。

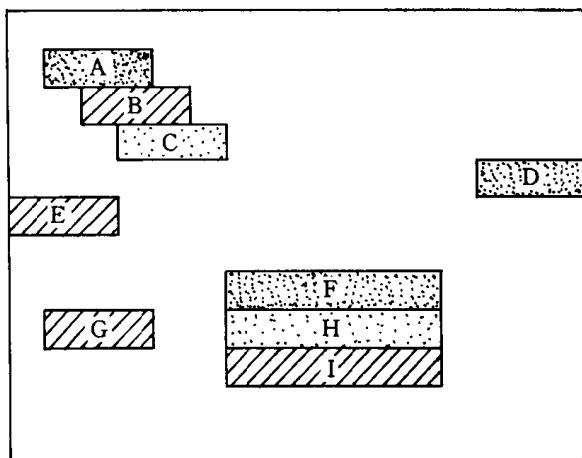


图 I -6-8 最一般的组块结构

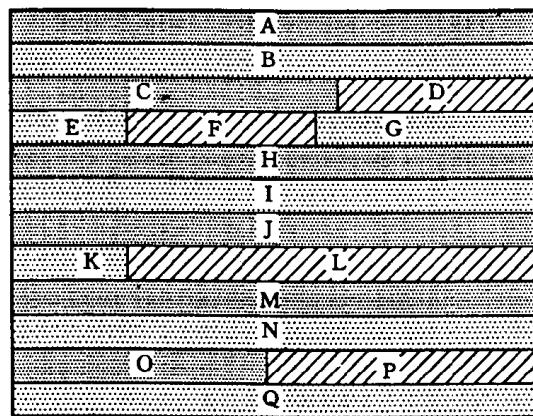


图 I -6-9 受限的组块结构

在定义框架的限定级别时要求组块结构遵守本节所述的限制, 这种情况可能用到“受限制的组块结构”这一术语。

### II -6. 1. 3 宏块

宏块包含一部分亮度分量和空间相关的色差分量。术语宏块既可以指源和解码的数据, 也可指相应的编码数据单元。对于一个跳过的宏块没有信息可供传递(见 I -7. 6. 6)。一个宏块可以有三种色差格式, 即  $4:2:0$ ,  $4:2:2$  和  $4:4:4$  格式。对于每种不同的色差格式宏块中的块的顺序会有不同, 阐明如下:

一个  $4:2:0$  宏块由六个块组成。这种结构包括四个 Y 块, 一个 Cb 块和一个 Cr 块。块顺序由图 I -6-10 描述。

一个  $4:2:2$  宏块由八个块组成。这种结构包括 4 个 Y 块, 两个 Cb 块和两个 Cr 块。块顺序由图 I -6-11 描述。

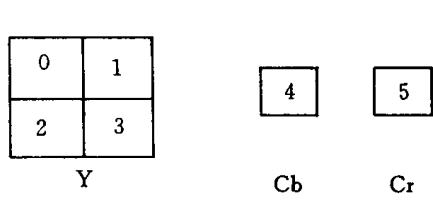


图 I -6-10  $4:2:0$  宏块结构

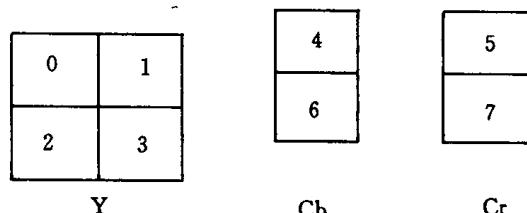


图 I -6-11  $4:2:2$  宏块结构

一个  $4:4:4$  的宏块由 12 个块组成。这种结构包括四个 Y 块, 四个 Cb 块和四个 Cr

块。块顺序如图 I -6-12 所描述。

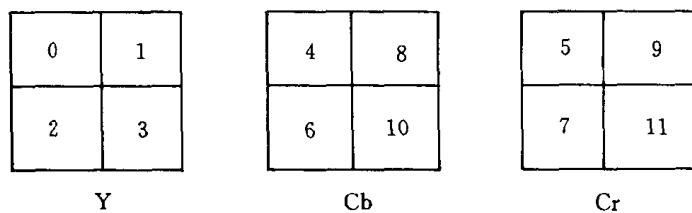


图 I -6-12 4:4:4 宏块结构

如果帧图中既会用到帧 DCT 编码,也会用到场 DCT 编码,则对于每种情况,宏块内部组织是不同的:

(1) 帧 DCT 编码中,每个块由两场的行交替组成,如图 I -6-13 所示。

(2) 场 DCT 编码中,每个块仅由两场中之一个场的行组成,如图 I -6-14 所示。

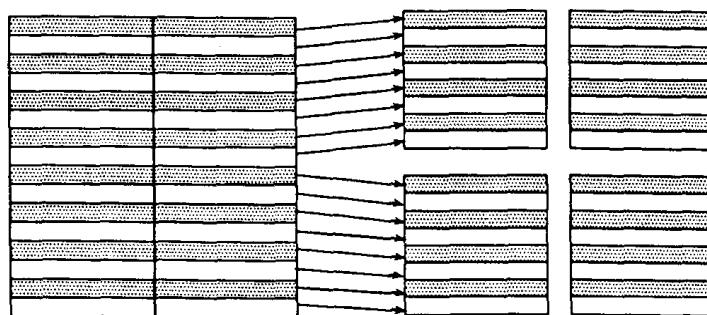


图 I -6-13 在帧 DCT 编码中的亮度宏块结构

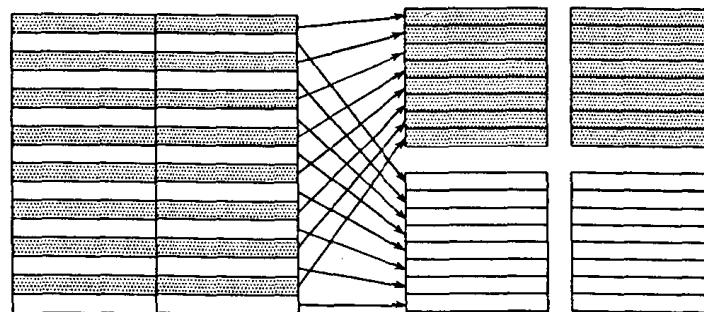


图 I -6-14 在场 DCT 编码中的亮度宏块结构

就色差块来说,它的结构取决于正在使用的色差格式。对于 4:2:2 和 4:4:4 格式来讲(在宏块的垂直方向有两个块),色差块与亮度块同等对待。然而,在 4:2:0 格式中,色差块将一直按帧结构进行组织,以满足 DCT 编码目的。然而值得注意的是,要做到对这些块基于场的预测,在一般情况下,要求必须能做到对 8×4 区域(半样点滤波以后)的预测。

就逐行帧而言,帧 DCT 编码总会按图 I-6-12 所示被用到。

在场图中,每个图仅包含从某一场来的行。在这种情况下,如图 I-6-13 所示,每个块都由从图中连续行中取出的行组成。

#### I -6. 1. 4 块

术语“块”既可以指源图象数据和重构数据,也可以指 DCT 系数或相应的编码数据单元。

当“块”指源图象数据或重构数据时,它指的是亮度分量或色差分量的正交部分。在块中有 8 行,(每行)8 个样点。

### I -6. 2 视频比特流语法

#### I -6. 2. 1 起始码

起始码是不会在视频流中另外出现的特定位模式。

每个起始码由一个起始码前缀和跟在后面的一个起始码值组成。起始码前缀由一串 23 位零值和跟着的一位 1 值组成。这样起始码前缀就是位串“0000 0000 0000 0000 0001”。

起始码值是一个八位整数,表明了起始码类型。大部分起始码类型只有一个起始码值,而 slice\_start\_code 由许多起始码值表示,在这种情况下,组块起始码值是 slice\_vertical\_position。

所有起始码都是字节定位的。这由在起始码前缀前面插入零值位来实现,这样起始码前缀的第一位就是一个字节的最高有效位。

表 I-6-1 定义了视频比特流中用到的起始码的一组码值。

表 I-6-1 起始码值

名 称	起始码值(十六进制)
picture_start_code	00
slice_start_code	01~AF
保留	B0
保留	B1
user_data_start_code	B2
sequence_header_code	B3
sequence_error_code	B4
extension_start_code	B5
保留	B6
sequence_end_code	B7
group_start_code	B8
系统起始码	B9~FF
注意,系统起始码定义于本规范的第一部分	

除了 sequence\_error\_code, 起始码的应用在下面的语法描述中都有定义。

sequence\_error\_code 已经被分配了用处, 为媒体接口来指明哪里检测到了未修正的错误。

## II -6. 2. 2 视频序列

video_sequence()	位数	助记符
next_start_code()		
sequence_header()		
if (nextbits() == extension_start_code){		
sequence_extension()		
do{		
extension_and_user_data(0)		
do {		
if (nextbits() == group_start_code){		
group_of_pictures_header()		
extension_and_user_data(1)		
}	.	.
picture_header()		
picture_coding_extension()		
extensions_and_user_data(2)		
picture_data()		
}while((nextbits() == picture_start_code)		
(nextbits() == group_start_code))		
if (nextbits() != sequence_end_code){		
sequence_header()		
sequence_extension()		
}		
} while(nextbits() != sequence_end_code)		
} else{		
/* ISO/IEC 11172-2 */		
}		
sequence_end_code	32	bslbf
}		

## II -6. 2. 2. 1 序列头

sequence_header()	位数	助记符
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantiser_matrix	1	
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8 * 64	uimsbf
load_non_intra_quantiser_matrix	1	
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8 * 64	uimsbf
next_start_code()		
}		

## II -6. 2. 2. 2 扩展和用户数据

extension_and_user_data(i){	位数	助记符
while((nextbits() == extension_start_code)		
(nextbits() == user_data_start_code)){		
if(i != 1)		
if(nextbits() == extension_start_code)		
extension_data(i)		
if(nextbits() == user_data_start_code)		
user_data()		
}		
}		

## 1. 扩展数据

extension_data(i){	位数	助记符
while(nextbits() == extension_start_code){		
<b>extension_start_code</b>	32	bslbf
if (i == 0){/* follows sequence_extension() */		
if (nextbits() == "Sequence Display Extension ID")		
sequence_display_extension()		
if (nextbits() == "Sequence Scalable Extension ID")		
sequence_scalable_extension()		
}		
/* Note: extension never follows a group_of_pictures_header() */		
if (i == 2){/* follows picture_coding_extension() */		
if (nextbits() == "Quant Matrix Extension ID")		
quant_matrix_extension()		
if (nextbits() == "Picture Pan Scan Extension ID")		
picture_display_extension()		
if (nextbits() == "Picture Spatial Scalable Extension ID")		
picture_spatial_scalable_extension()		
if (nextbits() == "Picture Temporal Scalable Ext. ID")		
picture_temporal_scalable_extension()		
}		
}		
}		

## 2. 用户数据

user_data(){	位数	助记符
<b>user_data_start_code</b>	32	bslbf
while(nextbits() != '0000 0000 0000 0000 0000 0001') {		
<b>user_data</b>	8	
}		
<b>next_start_code()</b>		
}		

### II -6. 2. 2. 3 序列扩展

sequence_extension(){	位数	助记符
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code()		
}		

### II -6. 2. 2. 4 序列显示扩展

sequence_display_extension(){	位数	助记符
extension_start_code_identifier	4	uimsbf
video_format	3	uimsbf
colour_description	1	uimsbf
if(colour_description){		
colour_primaries	8	uimsbf
transfer_characteristics	8	uimsbf
matrix_coefficients	8	uimsbf
}		
display_horizontal_size	14	uimsbf
marker_bit	1	bslbf
display_vertical_size	14	uimsbf
next_start_code()		
}		

## II -6. 2. 2. 5 序列分级扩展

sequence_scalable_extension()	位数	助记符
extension_start_code_identifier	4	uimsbf
scalable_mode	2	uimsbf
layer_id	4	uimsbf
if (scalable_mode == "spatial scalability") {		
lower_layer_prediction_horizontal_size	14	uimsbf
marker_bit	1	bslbf
lower_layer_prediction_vertical_size	14	uimsbf
horizontal_subsampling_factor_m	5	uimsbf
horizontal_subsampling_factor_n	5	uimsbf
vertical_subsampling_factor_m	5	uimsbf
vertical_subsampling_factor_n	5	uimsbf
}		
if (scalable_mode == "temporal scalability") {		
picture_mux_enable	1	uimsbf
if(picture_mux_enable)		
mux_to_progressive_sequence	1	uimsbf
picture_mux_order	3	uimsbf
picture_mux_factor	3	uimsbf
}		
next_start_code()		
}		

## II -6. 2. 2. 6 图组头

group_of_pictures_header()	位数	助记符
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code()		
}		

## II -6. 2. 3 图象头

picture_header{}{	位数	助记符
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type == 2    picture_coding_type == 3){		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if(picture_coding_type == 3){		
full_pel_backward_vector	1	
backward_f_code	3	uimsbf
}		
while(nextbits() == '1') {		
extra_bit_picture /* with the value "1" */	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture /* with the value "0" */	1	uimsbf
next_start_code()		
}		

### II -6. 2. 3. 1 图象编码扩展

picture_coding_extension(){	位数	助记符
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code[0][0] /* forward horizontal */	4	uimsbf
f_code[0][1] /* forward vertical */	4	uimsbf
f_code[1][0] /* backward horizontal */	4	uimsbf
f_code[1][1] /* backward vertical */	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf

续表

picture_coding_extension()	位数	助记符
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if (composite_display_flag){		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code()		
}		

### I-6.2.3.2 量化矩阵扩展

quant_matrix_extension()	位数	助记符
extension_start_code_identifier	4	uimsbf
load_intra_quantiser_matrix	1	uimsbf
if (load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8 * 64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if (load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8 * 64	uimsbf
load_chroma_intra_quantiser_matrix	1	uimsbf
if (load_chroma_intra_quantiser_matrix)		

续表

quant_matrix_extension()	位数	助记符
chroma_intra_quantiser_matrix[64]	8*64	uimsbf
load_chroma_non_intra_quantiser_matrix	1	uimsbf
if(load_chroma_non_intra_quantiser_matrix)		
chroma_non_intra_quantiser_matrix[64]	8*64	uimsbf
next_start_code()		
}		

### II -6. 2. 3. 3 图象显示扩展

picture_display_extension()	位数	助记符
extension_start_code_identifier	4	uimsbf
for(i=0;i<number_of_frame_centre_offsets;i++) {		
frame_centre_horizontal_offset	16	simsbf
marker_bit	1	bslbf
frame_centre_vertical_offset	16	simsbf
marker_bit	1	bslbf
}		
next_start_code()		
}		

### II -6. 2. 3. 4 图象时域分级扩展

picture_temporal_scalable_extension()	位数	助记符
extension_start_code_identifier	4	uimsbf
reference_select_code	2	uimsbf
forward_temporal_reference	10	uimsbf
marker_bit	1	bslbf
backward_temporal_reference	10	uimsbf
next_start_code()		
}		

## II -6. 2. 3. 5 图象空域扩展

picture_spatial_scalable_extension()	位数	助记符
<b>extension_start_code_identifier</b>	4	uimsbf
<b>lower_layer_temporal_reference</b>	10	uimsbf
<b>marker_bit</b>	1	bslbf
<b>lower_layer_horizontal_offset</b>	15	simsbf
<b>marker_bit</b>	1	bslbf
<b>lower_layer_vertical_offset</b>	15	simsbf
<b>spatial_temporal_weight_code_table_index</b>	2	uimsbf
<b>lower_layer_progressive_frame</b>	1	uimsbf
<b>lower_layer_deinterlaced_field_select</b>	1	uimsbf
<b>next_start_code()</b>		
}		

## II -6. 2. 3. 6 图象数据

picture_data()	位数	助记符
do {		
slice()		
}while(nextbits() == slice_start_code)		
next_start_code()		
}		

## II -6. 2. 4 组块

slice()	位数	助记符
<b>slice_start_code</b>	32	bslbf
if (vertical_size > 2800)		
<b>slice_vertical_position_extension</b>	3	uimsbf
if (<sequence_scalable_extension() is present in the bitstream>)		
if(scalable_mode == "data partitioning")		
<b>priority_breakpoint</b>	7	uimsbf
<b>quantiser_scale_code</b>	5	uimsbf
if (nextbits() == '1') {		
<b>intra_slice_flag</b>	1	bslbf

续表

位数	助记符
1	uimsbf
7	uimsbf
1	uimsbf
8	
1	uimsbf
do{	
macroblock()	
} while(nextbits() !='000 0000 0000 0000 0000 0000')	
next_start_code()	
}	

## II -6. 2. 5 宏块

位数	助记符
11	bslbf
1~11	vlclbf
5	uimsbf
1	bslbf

续表

macroblock()	位数	助记符
coded_block_pattern()		
for(i=0;i<block_count;i++){		
block(i)		
}		
}		

### II -6. 2. 5. 1 宏块模式

macroblock_modes()	位数	助记符
<b>macroblock_type</b>	1~9	vlclbf
if ((spatial_temporal_weight_code_flag==1)&&(spatial_temporal_weight_code_table_index !='00')){		
<b>spatial_temporal_weight_code</b>	2	uimsbf
}		
if (macroblock_motion_forward    macroblock_motion_backward){		
if (picture_structure=='frame'){              if (frame_pred_frame_dct==0) <b>frame_motion_type</b>	2	uimsbf
} else { <b>field_motion_type</b>	2	uimsbf
}		
}		
if (decode_dct_type){ <b>dct_type</b>	1	uimsbf
}		
}		

### II -6. 2. 5. 2 运动向量

motion_vectors(s)	位数	助记符
if(motion_vector_count==1){		
if((mv_format==field)&&(dmv !=1))		

续表

<code>motion_vectors(s){</code>	位数	助记符
<b>motion_vertical_field_select[0][s]</b>	1	uimsbf
<code>motion_vector(0,s)</code>		
<b>}else{</b>		
<b>motion_vertical_field_select[0][s]</b>	1	uimsbf
<code>motion_vector(0,s)</code>		
<b>motion_vertical_field_select[1][s]</b>	1	uimsbf
<code>motion_vector(1,s)</code>		
<b>}</b>		
<b>}</b>		

### 1. 运动向量

<code>motion_vector(r,s){</code>	位数	助记符
<b>motion_code[r][s][0]</b>	1~11	vlclbf
<code>if((f_code[s][0]! =1)&amp;&amp;(motion_code[r][s][0]! =0))</code>		
<b>motion_residual[r][s][0]</b>	1~8	uimsbf
<code>if(dmv==1)</code>		
<b>dmvector[0]</b>	1~2	vlclbf
<b>motion_code[r][s][1]</b>	1~11	vlclbf
<code>if((f_code[s][0]! =1)&amp;&amp;(motion_code[r][s][1]! =0))</code>		
<b>motion_residual[r][s][1]</b>	1~8	uimsbf
<code>if(dmv==1)</code>		
<b>dmvector[1]</b>	1~2	vlclbf
<b>}</b>		

### II -6. 2. 5. 3 编码块的模式

<code>coded_block_pattern(){</code>	位数	助记符
<b>coded_block_pattern_420</b>	3~9	vlclbf
<code>if(chroma_format==4 : 2 : 2)</code>		
<b>coded_block_pattern_1</b>	2	uimsbf
<code>if(chroma_format==4 : 4 : 4)</code>		
<b>coded_block_pattern_2</b>	6	uimsbf
<b>}</b>		

II -6. 2. 6 块

对于术语“第一个 DCT 系数”，“连续的 DCT 系数”和“块结尾”的详细描述在 I-7.2 节中。

**注意** 当使用数据分割时,本节对块层语法的描述是不恰当的。见 I-7.10 节。

## II - 6.3 视频比特流语义

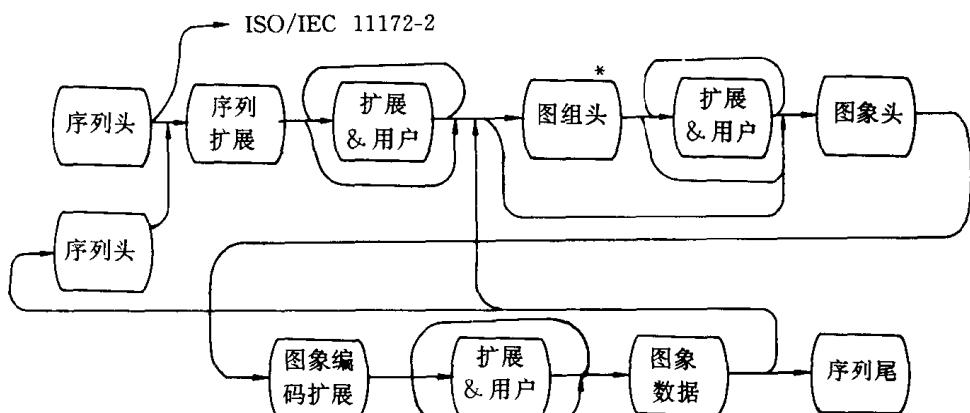
## II -6. 3. 1 更高层语法结构的语义规则

对于将更高层的语法元素组合在一起而生成合法比特流的方法，本节将详述其指导规则。下一小节将详述视频比特流中所有域的语义含义。

图 1-6-15 阐明视频比特流的高层结构。

下面的语义规则适用：

- 如果序列的第一个 sequence\_header() 后没有跟 sequence\_extension(), 那么此数



\* 在 GOP 后的第一幅图应为 I-图

图 II-6-15 高层比特流组织

据流将与 ISO/IEC 11172-2 一致,不在本规范中说明。

- 如果序列的第一个 sequence\_header()后面跟有 sequence\_extension(),那么所有接下来出现的 sequence\_header()后面都将紧跟着 sequence\_extension()。
- sequence\_extension 仅紧跟在 sequence\_header()后。
- 如果 sequence\_extension 在比特流中出现,那么每个 picture\_header()后面都将紧跟一个 picture\_coding\_extension()。
- picture\_coding\_extension()仅紧跟在 picture\_header()后。
- 跟在 group\_of\_picture\_header()后面的第一个编码图应该是 I-图。

除了 sequence\_extension()和 picture\_coding\_extension()以外,还定义了一些扩展。在带扩展的语法中的不同地方,允许的扩展集是不同的。表 II-6-2 为每种扩展都定义了一个四位 extension\_start\_code\_identifier。

在比特流中允许扩展的每一处,都可能包含限定的允许集中任意数目的扩展。然而每种类型的扩展最多只能出现一次。

如果在本规范中,解码器遇到带有“保留”扩展辨别符的扩展,则应丢弃所有后续数据,直到遇到下一个起始符。这一要求保证了将来与本规范兼容的扩展的定义。

表 II-6-2 extension\_start\_code\_identifier 码

extension_start_code_identifier	名 称
0000	保留
0001	序列扩展 ID
0010	序列显示扩展 ID
0011	量化矩阵扩展 ID
0100	保留

续表

extension_start_code_identifier	名 称
0101	序列分级扩展 ID
0110	保留
0111	图象显示扩展 ID
1000	图象编码扩展 ID
1001	图象空域分级扩展 ID
1010	图象时域分级扩展 ID
1011	保留
1100	保留
...	...
1111	保留

### II -6. 3. 2 视频序列

**sequence\_end\_code** sequence\_end\_code 是十六进制位串 000001B7。它结束一个视频序列。

### II -6. 3. 3 序列头

**sequence\_header\_code** sequence\_header\_code 是十六进制位串 000001B3。它识别一个序列头的起始。

**horizontal\_size\_value** 这个字形成了 horizontal\_size 的低 12 位有效位。

**vertical\_size\_value** 这个字形成了 vertical\_size 的低 12 位有效位。

**horizontal\_size** horizontal\_size 是一个 14 位无符号整数, 低 12 位有效位由 horizontal\_size\_value 定义, 高两位有效位在 horizontal\_size\_extension 中定义。horizontal\_size 是用样点数表示的图象亮度分量可显示部分的宽度, 用宏块数表示的图象亮度分量的宽度, mb\_width, 是  $(\text{horizontal\_size} + 15) / 16$ 。编码图中显示部分是左定位的。

为了防止对起始码的竞争, horizontal\_size\_value 不能是零。这使得 horizontal\_size 的值不能被 4096 整除。

**vertical\_size** vertical\_size 是一个 14 位无符号整数, 低 12 位有效位在 vertical\_size\_value 中定义, 高两位有效位在 vertical\_size\_extension 中定义。vertical\_size\_value 是用行数表示的帧的亮度分量可显示部分的高度。

如果 progressive\_sequence 是“1”, 以宏块数表示的图象亮度分量的高度, mb\_height, 是  $(\text{vertical\_size} + 15) / 16$ 。

如果 progressive\_sequence 是“0”, 以宏块数表示的图象亮度分量的高度, mb\_height, 是  $2 * ((\text{vertical\_size} + 31) / 32)$ 。以宏块表示的场图亮度分量的高度, mb\_height,

是((vertical\_size+31)/32)。

编码图中可显示部分是顶定位的。

为了防止对起始码的竞争,vertical\_size\_value 不能为零。这使得 vertical\_size 的值不能为 4096 的倍数。

**aspect\_ratio\_information** 这是表 II-6-3 中定义的一个四位整数。

aspect\_ratio\_information 或者规定了重构帧的“样点高宽比”(SAR)是 1.0(样本平方);或者给出了“显示高宽比”(DAR)。

- 如果 sequence\_display\_extension() 没有出现,那么它表明整个重构帧将要映射到整个显示活动区。样本高宽比可以按如下计算:

$$SAR = DAR \times \frac{\text{horizontal\_size}}{\text{vertical\_size}}$$

注意 在这种情况下,horizontal\_size 和 vertical\_size 由源图象的 SAR 和选定的 DAR 所限制。

- 如果 sequence\_display\_extension() 出现,那么样本高宽比按如下计算:

$$SAR = DAR \times \frac{\text{display_horizontal\_size}}{\text{display_vertical\_size}}$$

表 II-6-3 aspect\_ratio\_information

aspect_ratio_information	样本高宽比	DAR
0000	禁止	禁止
0001	1.0(样本平方)	—
0010	—	3÷4
0011	—	9÷16
0100	—	1÷2.21
0101	—	保留
...	—	...
1111	—	保留

**frame\_rate\_code** 这是用来定义表 II-6-4 中所示的 frame\_rate\_value 的一个四位整数。frame\_rate 可以从 frame\_rate\_value,frame\_rate\_extension\_n 和 frame\_rate\_extension\_d 中求得,如下所示:

$$\begin{aligned} \text{frame\_rate} = & \text{frame\_rate\_value} * (\text{frame\_rate\_extension\_n} + 1) \\ & + (\text{frame\_rate\_extension\_d} + 1) \end{aligned}$$

当表 II-6-4 中直接存在一个“帧速率”项时,frame\_rate\_extension\_n 和 frame\_rate\_extension\_d 为零。

如果 progressive\_sequence 为“1”,则在解码过程输出端的两个连续帧之间的周期是 frame\_rate 的倒数。参见图 II-7-18。

如果 progressive\_sequence 为“0”,则在解码过程输出端的两个连续场之间的周期是

frame\_rate 的倒数的一半。参见图 II-7-20。

如果在 sequence\_scalable\_extension()中的 picture\_mux\_enable 是“1”,在时域分级的增强层中标志的 frame\_rate 是时域再复合操作后的帧率的组合。

表 II-6-4 frame\_rate\_value

frame_rate_code	frame_rate_value
0000	禁止
0001	$24\ 000 \div 1001(23.976)$
0010	24
0011	25
0100	$30\ 000 \div 1001(29.97)$
0101	30
0110	50
0111	$60\ 000 \div 1001(59.94)$
1000	60
...	保留
1111	保留

**bit\_rate\_value** bit\_rate 的低 18 位。

**bit\_rate** 这是一个 30 位整数。这个整数的低 18 位在 bit\_rate\_value 中,高 12 位在 bit\_rate\_extension 中。这个 30 位的整数以 400bit/s 为单位描述比特流的比特率,并向上取整。值不可为零。

如果是常量比特率的比特流,那么规定的比特率就是附录 I-C 中提到的 VBV 的操作实际速率。如果是可变比特率的比特流,那么在 ISO/IEC 13818-1 中规定的 STD 替换了 VBV,并且用这里规定的比特率来标定传输流 STD (ITU-T Rec. xxx | ISO/IEC 13818-1 中的 I-2.4.2) 或程序流 STD (ITU-T Rec. xxx | ISO/IEC 13818-1 的 I-2.4.5)。

如果比特流不是一个常量速率流,那么 vbv\_delay 域将为十六进制的 FFFF 值。

如果在比特率域给定了编码的值,比特流就可以产生,而视频编码和最坏情况下的多路抖动也不会引起 STD 缓冲区上溢或下溢。

**marker\_bit** 这是一个将要被置为“1”的位。这个位防止了对起始码的竞争。

**vbv\_buffer\_size\_value** vbv\_buffer\_size 的低 10 位。

**vbv\_buffer\_size** vbv\_buffer\_size 是一个 18 位的整数。整数的低 10 位在 vbv\_buffer\_size\_value 中,高 8 位在 vbv\_buffer\_size\_extension 中。这个整数定义了 VBV(视频缓冲区检验器,见附录 I-C)缓冲区的尺寸,来对序列进行解码。它定义如下:

$$B = 16 * 1024 * vbv_buffer_size$$

B 是对序列进行解码所需的按位表示的 VBV 缓冲区的最小尺寸(见附录 I-C)。

**constrained\_parameters\_flag** 这个标志(在 ISO/IEC 11172-2 中使用的)在本规范

中没有意义,将取零值。

**load\_intra\_quantiser\_matrix** 见第 II-6.3.7 节“Quant matrix extension”。

**intra\_quantiser\_matrix** 见第 II-6.3.7 节“Quant matrix extension”。

**load\_non\_intra\_quantiser\_matrix** 见第 II-6.3.7 节“Quant matrix extension”。

**non\_intra\_quantiser\_matrix** 见第 II-6.3.7 节“Quant matrix extension”。

## II -6.3.4 扩展和用户数据

**extension\_start\_code** extension\_start\_code 是十六进制位串 000001B5。它指明了除 ISO/IEC 11172-2 以外的扩展的开始。

### II -6.3.4.1 用户数据

**user\_data\_start\_code** user\_data\_start\_code 是十六进制位串 000001B2。它标明了用户数据的开始。用户数据一直延续到另一个起始码的出现。

**user\_data** user\_data 由用户为其特殊应用所定义。user\_data 中不能包含 23 个以上的零的位串。

## II -6.3.5 序列扩展

**extension\_start\_code\_identifier** 这是识别扩展的一个四位整数。见表 II-6-2。

**profile\_and\_level\_indication** 这是用来标志框架和级别的一个 8 位整数。这些位的意义在第 II-8 节中给出。

**注意** 在分级体系中,每层的比特流都可以将 profile\_and\_level\_indication 置成一个不同的值,如第 II-8 节所述。

**progressive\_sequence** 如果被置“1”,那么编码的视频序列仅含逐行的帧图。当 progressive\_sequence 被置“0”,编码的视频序列可以同时包含帧图和场图,而帧图可以是逐行的或隔行的帧。

**chroma\_format** 这是指明如表 II-6-5 所示的色差格式的一个两位整数。

表 II-6-5 chroma\_format 的含义

chroma_format	含    义
00	保留
01	4 : 2 : 0
10	4 : 2 : 2
11	4 : 4 : 4

**horizontal\_size\_extension** 这个字构成了 horizontal\_size 的最高两位有效位。

**vertical\_size\_extension** 这个字构成了 vertical\_size 的最高两位有效位。

**bit\_rate\_extension** 这个字构成了 bit\_rate 的高 12 位有效位。

**vbv\_buffer\_size\_extension** 这个字构成了 vbv\_buffer\_size 的高 8 位有效位。

**low\_delay** 当这个标志被置为“1”时,表示序列中不包含 B-图,并且帧重排延迟不

在 VBV 描述中出现,而且 VBV 下溢可能发生。

当被置为“0”,它表明序列可以包含 B-图,在 VBV 描述中不出现帧重排延迟,并且 VBV 缓冲区不会发生下溢。

这个标志在解码过程中不会用到,因此可被解码器忽略,但限定和验证 low\_delay 比特流的一致性是必要的。

**frame\_rate\_extension\_n** 见 frame\_rate\_code。

**frame\_rate\_extension\_d** 见 frame\_rate\_code。

### II -6.3.6 序列显示扩展

本规范未定义显示过程。在这个扩展中的信息不影响解码过程,并且为了与本规范一致,可以忽略这些信息。

**video\_format** 这是一个 3 位的整数,它指明了图象在按本规范编码之前的格式。它的含义在表 II -6-6 中定义。

表 II -6-6 video\_format 的含义

video_format	含 义
000	分量
001	PAL
010	NTSC
011	SECAM
100	MAC
101	未规定视频格式
110	保留
111	保留

**colour\_description** 是一个标志,如被置为“1”,表明比特流中将出现 colour\_primaries,transfer\_characteristics 和 matrix\_coefficients。

**colour\_primaries** 这个 8 位整数描述了源图象的元色色度坐标,如表 II -6-7 所定义。

表 II -6-7 元色

值	元 色
0	(禁止)
1	ITU-R 推荐 709(1990) 元色 x y 绿 0.300 0.600 蓝 0.150 0.060 红 0.640 0.330 白 D65 0.3127 0.3290

续表

值	元色
2	未规定的视频 未知图象特性
3	保留
4	ITU-R 推荐 624-4 系统 M 元色 x y 绿 0.21 0.71 蓝 0.14 0.08 红 0.67 0.33 白 C 0.310 0.316
5	ITU-R 推荐 624-4 系统 B,G 元色 x y 绿 0.29 0.60 蓝 0.15 0.06 红 0.64 0.33 白 D65 0.313 0.329
6	SMPTE 170M 元色 x y 绿 0.310 0.595 蓝 0.155 0.070 红 0.630 0.340 白 D65 0.3127 0.3290
7	SMPTE 240M(1987) 元色 x y 绿 0.310 0.595 蓝 0.155 0.070 红 0.630 0.340 白 D65 0.3127 0.3291
8~255	保留

如果 sequence\_display\_extension() 未在比特流中出现, 或 colour\_description 为零, 则假定色度与 colour\_primaries 为 1 时相对应。

**transfer\_characteristics** 这个 8 位整数描述了源图象的光电传输特性, 如表 II -6-8 定义。

表 II -6-8 传输特性

值	传输特性
0	(保留)
1	ITU-R 推荐 709(1990) $V = 1.099 L_c^{0.45} - 0.099$ $1 \geq L_c \geq 0.018$

续表

值	传输特性
	$V = 4.500 L_c$ $0.018 > L_c \geq 0$
2	未规定的视频 未知图象特性
3	保留
4	ITU-R 推荐 624-4 系统 M 假设显示格玛 2.2
5	ITU-R 推荐 624-4 系统 B,G 假设显示格玛 2.8
6	SMPTE 170M $V = 1.099 L_c^{0.45} - 0.099$ $1 \geq L_c \geq 0.018$ $V = 4.500 L_c$ $0.018 > L_c \geq 0$
7	SMPTE 240M(1987) $V = 1.1115 L_c^{0.45} - 0.1115$ $L_c \geq 0.0228$ $V = 4.0 L_c$ $0.0228 > L_c$
8	线性变换特性 i. e. $V = L_c$
9~255	保留

如果 sequence\_display\_extension() 没有在比特流中出现或 colour\_description 的值为零, 则传输特性被假定为与 transfer\_characteristics 值为 1 时相对应。

**matrix\_coefficients** 这个 8 位整数描述了从绿、蓝、红基色推导亮度和色差信号所用到的矩阵系数, 如表 II-6-9 所定义。

表 II-6-9 系数矩阵

值	矩 阵
0	(禁止)
1	ITU-R 推荐 709 (1990) $E'_Y = 0.7154E'_G + 0.0721E'_B + 0.2125E'_R$ $E'_PB = -0.386E'_G + 0.500E'_B - 0.115E'_R$ $E'_PR = -0.454E'_G - 0.046E'_B + 0.500E'_R$

续表

值	矩阵
2	未规定的视频 未知图象特性
3	保留
4	FCC $E'_Y = 0.59E'_G + 0.11E'_B + 0.30E'_R$ $E'_{PB} = -0.331E'_G + 0.500E'_B - 0.169E'_R$ $E'_{PR} = -0.421E'_G - 0.079E'_B + 0.500E'_R$
5	ITU-R 推荐 624-4 系统 B,G $E'_Y = 0.587E'_G + 0.114E'_B + 0.299E'_R$ $E'_{PB} = -0.331E'_G + 0.500E'_B - 0.169E'_R$ $E'_{PR} = -0.419E'_G - 0.081E'_B + 0.500E'_R$
6	SMPTE 170M $E'_Y = 0.587E'_G + 0.114E'_B + 0.299E'_R$ $E'_{PB} = -0.331E'_G + 0.500E'_B - 0.169E'_R$ $E'_{PR} = -0.419E'_G - 0.081E'_B + 0.500E'_R$
7	SMPTE 240M(1987) $E'_Y = 0.701E'_G + 0.087E'_B + 0.212E'_R$ $E'_{PB} = -0.384E'_G + 0.500E'_B - 0.116E'_R$ $E'_{PR} = -0.445E'_G - 0.055E'_B + 0.500E'_R$
8~255	保留

在这个表中：

$E'_Y$ 是值在 0 与 1 之间的模拟量

$E'_{PB}$ 和  $E'_{PR}$ 是值在 -0.5 与 0.5 之间的模拟量

$Y, Cb$  和  $Cr$  与  $E'_Y, E'_{PB}$  和  $E'_{PR}$  的关系由下面的公式给出：

$$Y = (219 * E'_Y) + 16$$

$$Cb = (224 * E'_{PB}) + 128$$

$$Cr = (224 * E'_{PR}) + 128$$

注意 由本规范给出的解码过程将输出样本值限制在 [0 : 255] 的范围。在解码过程的输出端, 可能出现超出上面公式所指范围的编码字, 尤其是可能出现编码字 0 和 255。

如果 sequence\_display\_extension 未在比特流中出现或者 colour\_description 为零, 则矩阵系数假定为与 matrix\_coefficients 为 1 时相对应。

**display\_horizontal\_size** 见 **display\_vertical\_size**。

**display\_vertical\_size** **display\_vertical\_size** 和 **display\_horizontal\_size** 共同定义了一个矩形, 可以被当作“预想显示”的活动区域。如果这个矩形比编码帧的尺寸小, 那么我们希望显示过程仅显示这个编码帧的一部分。反之, 如果显示矩形比编码帧的尺寸还大,

那么我们希望显示过程在显示设备的一部分上显示重构帧,而不是在整个显示设备上显示。

`display_horizontal_size` 应与 `horizontal_size` 具有相同的单位(编码帧的每行样本数)。

`display_vertical_size` 应与 `vertical_size` 具有相同的单位(编码帧的行数)。

`display_horizontal_size` 和 `display_vertical_size` 并不影响解码过程,但可能被并未在本规范中标准化的显示过程用到。

## II -6.3.7 序列分级扩展

有一个语义限制:如果在比特流中跟在 `sequence_extension()` 后面有 `sequence_scalable_extension()`,那么 `sequence_scalable_extension()` 将跟在每个 `sequence_extension()` 后面。这样,比特流或者是分级的,或者是不分级的。不可能在一个序列中将分级和非分级混在一起。

**scalable\_mode** `scalable_mode` 指明了在视频序列中用到的分级类型。如果在比特流中没有出现 `sequence_scalable_extension()`,那么那个序列就没有使用分级。`scalable_mode` 还指明了将要用到的 `macroblock_type` 表。然而在空域分级的情况下,如果对于一个给定的图没有 `picture_spatial_extension()`,那么那个图将以非分级方式解码。`scalable_mode` 的定义如表 II -6-10 所示。

表 II -6-10 `scalable_mode` 的定义

<code>scalable_mode</code>	含 义	<code>picture_spatial_scalable_extension()</code>	<code>macroblock_type</code> tables
<code>sequence_scalable_extension() not present</code>			B-2,B-3 and B-4
00	数据分割		B-2,B-3 and B-4
01	空域分级	存在	B-5,B-6 and B-7
		不存在	B-2,B-3 and B-4
10	SNR 分级		B-8
11	时域分级		B-2,B-3 and B-4

**layer\_id** 这是一个在分级体系中标明各个层次的整数。基本层的 `layer_id` 通常为 0。然而一个分级体系的基本层并不带有 `sequence_scalable_extension()`,因此也不带 `layer_id`,除了在数据分割的情况下。每个接下去的后续层都有一个 `layer_id`,比它的前一层(即这层为其增强层)多 1。

对于数据分割,其中零分区的 `layer_id` 应为零,一分区的 `layer_id` 应为 1。

**lower\_layer\_prediction\_horizontal\_size** 这是一个 14 位的整数,指明了用作预测的低层帧的水平尺寸。这个数应包含低层比特流中 `horizontal_size` 所包含的值(`horizontal_size_value` 和 `horizontal_size_extension`)。

**lower\_layer\_prediction\_vertical\_size** 这是一个 14 位的整数, 指明了用作预测的低层帧的竖直尺寸。这个数中应包含低层比特流中 vertical\_size 所包含的值(vertical\_size\_value 和 vertical\_size\_extension)。

**horizontal\_subsampling\_factor\_m** 如 II-7.7.2 中所定义, 这个值影响到空域分级的上采样过程。这个值不能为零。

**horizontal\_subsampling\_factor\_n** 如 II-7.7.2 中所定义, 这个值影响到空域分级的上采样过程。这个值不能为零。

**vertical\_subsampling\_factor\_m** 如 II-7.7.2 中所定义, 这个值影响到空域分级的上采样过程。这个值不能为零。

**vertical\_subsampling\_factor\_n** 如 II-7.7.2 中所定义, 这个值影响到空域分级的上采样过程。这个值不能为零。

**picture\_mux\_enable** 如果这个值被置为 1, 则 picture\_mux\_order 和 picture\_mux\_factor 被用于显示过程前的再次复合。

**mux\_to\_progressive\_sequence** 当这个标志被置为 1 时, 表明这两层相应的解码图象将被时域多路复合而生成一个逐行序列以供显示。当时域多路复合打算生成一个隔行序列时, 这个标志将为零。

**picture\_mux\_order** 它指明了在第一个基层图象之前的增强层图象数目。这样, 由于它包含了在编码器实现的多路分解的反过程信息, 它能对显示前的图象的再次多路复合有所帮助。

**picture\_mux\_factor** 它指明了在相继的基本层图象之间的增强层图象的数目, 从而保证产生正确的基层和附加层的再次多路复合, 以供显示。由于它包含了在编码器实现的多路分解的反过程信息, 它也能对显示前的图象的再次多路复合提供帮助。值“000”保留。

### II-6.3.8 图组头

**group\_start\_code** 这个 group\_start\_code 是十六进制位串 0000 01B8。它可以用来识别一组图的头的开始。

**time\_code** 这个 25 个位的字段包含如下内容: drop\_frame\_flag, time\_code\_hours, time\_code\_minutes, marker\_bit, time\_code\_seconds 和 time\_code\_pictures, 如表 II-6-11 所示。这些字段与 IEC 标准中“视频磁带录象机的时间和控制编码”所定义的字段相对应。(见附录 II-G 文献目录)这个码涉及 temporal\_reference 为零的图组头后第一幅图象。drop\_frame\_flag 可以被置为“0”或“1”。仅当帧速率为 29.97Hz 时它可以被置为“1”。如果它为“0”, 那么则假定每秒钟的图象数目取其为最接近的整数。例如, 29.97Hz 将被取整为 30Hz。如果它为“1”, 那么在每分钟的开始(除了第 0, 10, 20, 30, 40, 50 分钟), 图象数目 0 和 1 将从计数中忽略。

注意 time\_code 所带信息不在解码过程中起作用。

表 II -6-11 time\_code

time_code	值域	位数	缩写
drop_frame_flag		1	
time_code_hours	0~23	5	uimsbf
time_code_minutes	0~59	6	uimsbf
marker_bit	1	1	“1”
time_code_seconds	0~59	6	uimsbf
time_code_pictures	0~59	6	uimsbf

**closed\_gop** 这是一个一位的标志,它指出了紧跟在第一个编码的 I-帧(跟在图组头后)后面的 B-图所使用的预测的性质。

closed\_gop 置为“1”表明这些 B-图仅使用了后向预测编码。

这个位被提供用于在编码后进行的任何编辑过程。如果以前的图被编辑移走,broken\_link 则置为“1”,由此解码器可以避免显示这些紧跟在第一个 I-图(跟图组头后)的 B-图。然而如果 closed\_gop 位被置为“1”,则编辑器可以选择不将 broken\_link 位置“1”,因为这些 B-图可以被正确解码。

**broken\_link** 这是一个在编码过程中应被置为“0”的一位标志。如果它被置为“1”,则表明由于用于预测的参考帧不能得到(由于编辑),紧跟在第一个在图组头后的 I-帧后面的 B-图(如果有的话)可能不能被正确解码。

解码器可以使用这个位来避免显示不能被正确解码的帧。

### II -6. 3. 9 图象头

**picture\_start\_code** picture\_start\_code 是一个值为 00000100 的十六进制 32 位的串。

**temporal\_reference** temporal\_reference 是一个与每个输入图象均有关系的 10 位无符号整数。对于每个输入帧,它将进行增 1 模 1024 的运算。当一个帧按两场编码时,在两场中的图象头中的 temporal\_reference 具有相同值。

在一个组起始头后面的最早图象(按显示顺序)的 temporal\_refrence 将被清为零。

**picture\_coding\_type** picture\_coding\_type 指明了一个图象是帧内编码图(I),还是预测编码图(P)或双向预测编码图(B)。picture\_coding\_type 的含义在表 II -6-12 中定义。

注意 ISO/IEC 11172-2 中用到的仅含 DC 系数(D-图)的帧内编码图不被本规范所支持。

表 II -6-12 picture\_coding\_type

picture_coding_type	编码方法
000	禁 止
001	内部编码(I)

续表

picture_coding_type	编码方法
010	预测编码(P)
011	双向预测编码(B)
100	不应使用 (ISO/IEC 11172-2 中的 dc 内部编码)
101	保留
110	保留
111	保留

**vbv\_delay** **vbv\_delay** 是一个 16 位无符号整数。对于常量比特率的操作,它被用来在播放的开始时设置解码器缓冲区的初始分配,以防解码器缓冲区出现上溢或下溢。**vbv\_delay** 测量在 R 比特率下,从初始的空状态到当前的图象刚要被移出缓冲区之前,将 VBV 缓冲填充到适当程度所需的时间。

**vbv\_delay** 的值是在 VBV 接到图象起始码的最后一个字节之后所要等待的 90kHz 的系统时钟周期数。它可以以下的 VBV 的状态方程计算而得:

$$\text{vbv\_delay}_n = 90\ 000 * B_n^* / R$$

其中:

$$n > 0$$

$B_n^*$  = 在图象 n 刚被从缓冲区移走之前,而在位于图象 n 数据元素之前的任何图组头数据、序列头数据和 picture\_start\_code 被移走之后,以比特计数的对 VBV 的占有(情况)

R = 实际的比特率(即为完整精确度的,而不是由序列头给出的 bit\_rate 中的量化值)

对于非常量比特率的操作,**vbv\_delay** 具有十六进制的值 FFFF。贯穿整个序列的比特流既可是常量比特率的流,也可是可变比特率的流。

**full\_pel\_forward\_vector** 这个标志在 ISO/IEC 11172-2 中用到,而未在本规范中用到。它将为零值。

**forward\_f\_code** ISO/IEC 11172-2 中用到的这个参数并未在本规范中用到。它将为 7(全部相同)。

**full\_pel\_backward\_vector** 这个在 ISO/IEC 11172-2 中用到的标志并未在本规范中使用。它的值应为零。

**backward\_f\_code** 这个在 ISO/IEC 11172-2 中用到的标志并未在本规范中使用。它的值应为 7(全部相同)。

**extra\_bit\_picture** 指出下面将出现额外信息的一个位。如果 extra\_bit\_picture 被置为“1”,后面将跟着 extra\_information\_picture。如果它被置为“0”;,后面将没有数据。extra\_bit\_picture 将被置为“0”,值“1”将为 ITU-T | ISO/IEC 将来定义的扩展而保留。

**extra\_information\_picture** 保留。与本规范相一致的解码器在比特流中遇到 extra\_information\_picture 时,将对其忽略(即,从比特流中析出并舍弃)。与本规范相一致的比特流中将不包含这个语法元素。

### II -6. 3. 10 图象编码扩展

**f\_code[s][t]** 从 1 到 9 取值的无符号整数(如果不被用到,则值为 15)。禁止为零值。在一个 concealment\_motion\_vectors 为零的 I-图中(在这种情况下 f\_code[s][t]不在解码过程中被用到),f\_code[s][t]将取值 15(全部相同)。(标志 s 和 t 的含义参见表 I -7-7)

**intra\_dc\_precision** 这是表 I -6-13 中定义的一个 2 位整数。

如 I -7. 4. 1 中所说明,内部 DC 系数的反量化过程将被此参数修改。

**picture\_structure** 这是如表 I -6-14 所定义的一个 2 位整数。

表 I -6-13 内部 DC 精度

intra_dc_precision	精度/bit
00	8
01	9
10	10
11	11

表 I -6-14 picture\_structure 的含义

picture_structure	含义
00	保留
01	顶场
10	底场
11	帧图

当一个帧以两个场图的形式编码时,这两个场图必须具有同样的 picture\_coding\_type(图象编码类型),除非第一个编码场是 I-图,第二个编码场可为 I-图或 P-图。

一个帧的第一个编码场可以是一个顶场或一个底场,下一场必须是相对的另一种类型(场)。

当一个帧以两个场图的形式编码时,在每个场图中的下面语法单元可以被独立设置:

- f\_code[0][0], f\_code[0][1]
- f\_code[1][0], f\_code[1][1]
- intra\_dc\_precision,concealment\_motion\_vectors,q\_scale\_type
- intra\_vlc\_format, alternate\_scan

**top\_field\_first** 这个单元的含义由 picture\_structure,progressive\_sequence 和 repeat\_first\_field 决定。

如果 progressive\_sequence 等于零,这个标志指明重构帧的哪一场先被解码过程输出。

在一个场图中,top\_field\_first 有值“0”,由解码过程输出的唯一场为解码的场图。

在一个帧图中,如果 top\_field\_first 被置为“1”,则指明重构帧的顶场是解码过程输出的第一个场。top\_field\_first 被置为“0”则指明重构帧的底场是解码过程输出的第一场。

如果 progressive\_sequence 等于 1,则这个标志与 repeat\_first\_field 组合在一起,指明了重构帧被解码过程输出的次数(1 次,2 次或 3 次)。

如果 repeat\_first\_field 被置为 0，则 top\_field\_first 将被置为 0。由此，与这个重构帧相对应的解码过程的输出只包括一个逐行帧。

如果 top\_field\_first 置为 0 而 repeat\_first\_field 置为 1，则与这个重构帧相对应的解码过程的输出包括两个逐行帧。

如果 top\_field\_first 置为 1,repeat\_first\_field 也置为 1，则与这个重构帧相对应的解码过程的输出包括三个逐行帧。

**frame\_pred\_frame\_dct** 如果这个标志被置为“1”，则仅仅帧 DCT 和帧预测被用到。在一个场图中，这个标志应被置为“0”。如果 progressive\_frame 是“1”，则 frame\_pred\_frame\_dct 应为“1”。这个标志对比特流的语法有影响。

**concealment\_motion\_vectors** 这个标志值为“1”则指明运动向量是为内部宏块而编码的。

**q\_scale\_type** 这个标志影响 I-7.4.2.2 中所描述的反量化过程。

**intra\_vlc\_format** 这个标志影响 I-7.2.1 中所描述的变换系数数据的解码。

**alternate\_scan** 这个标志影响 I-7.3 中所描述的变换系数数据的解码。

**repeat\_first\_field** 这个标志仅在帧图中有用，在场图中，它将被置为 0，且不影响解码过程。

如果 progressive\_sequence 等于 0，且 progressive\_frame 也为 0，则 repeat\_first\_field 为 0，且与这个重构帧相对应的解码过程的输出由两场组成。

如果 progressive\_sequence 等于 0，而 progressive\_frame 等于 1：

如果这个标志置为 0，则与这个重构帧相对应的解码过程的输出由两场组成。第一场（由 top\_field\_first 指明是顶场还是底场）后面跟着第二场；

如果这个标志置为 1，则与这个重构帧相对应的解码过程的输出由三场组成。第一场（由 top\_field\_first 指明是顶场还是底场）后面跟着第二场，然后再重复第一场。

如果 progressive\_sequence 等于 1：

如果这个标志置为 0，则与这个重构帧相对应的解码过程的输出由一帧组成；

如果它被置为 1，则与这个重构帧相对应的解码过程的输出由两个或三个帧组成，这取决于 top\_field\_first 的值。

**chroma\_420\_type** 如果 chroma\_format 为“4：2：0”，则 chroma\_420\_type 与 progressive\_frame 相同；否则 chroma\_format 没有意义，将为零。这个标志的存在是由于历史的原因。

**progressive\_frame** 如果 progressive\_frame 被置为 0，则表明这个帧的两场为隔行场，在两场之间存在着场周期时间间隔（相对于空域样点）。在这种情况下下面的限制适用：

- repeat\_first\_field 将为零（两场期间）

如果 progressive\_frame 置为 1，则表明这两场（这帧的）实际上来自同一时刻。在这种情况下，一些对于比特流中其它参数和标志的限制适用：

- picture\_structure 将为“帧”
- frame\_pred\_frame\_dct 将为 1

当视频序列用作空域分级序的低层时，这个参数被使用。这里，它影响通过低层在增

强层中形成预测所用到的上采样过程。

**composite\_display\_flag** 这个标志置为 1 表示：如输入图象在被编码为与本规范相一致的比特流之前已经被编码为(模拟)全电视信号，则下面的字段有用。如果它被置为 0，那么这些参数不在比特流中出现。

这些信息与紧跟在扩展后面的图象有关，如果这幅图是帧图，这信息与帧的第一场有关。对于第二场，等价的信息也可被推知(无法在比特流中表示它)。

值得注意的是，repeat\_first\_field 将导致全电视信号场在颜色场序列之外被重复。我们建议不同时置参数 repeat\_first\_field 和 composite\_display\_flag。

**v\_axis** 一个 1 位整数，仅用于当比特流中出现按 PAL 制编码的信号时。此时 v\_axis 被置为正 1，否则置为 0。

**field\_sequence** 一个 3 位整数，定义了 PAL 制中使用的 8 场序列的场的数目，或者 NTSC 制中使用的四场序列的场数目，如表 I-6-15 中所示。

表 I-6-15 field\_sequence 的定义

场序列	帧	场
000	1	1
001	1	2
010	2	3
011	2	4
100	3	5
101	3	6
110	4	7
111	4	8

**sub\_carrier** 这是一个 1 位整数。如果为 0 则意味着 sub\_carrier/行的频率关系是正确的。为 1 则表示关系不正确。

**burst\_amplitude** 这是定义脉冲幅值的一个 7 位整数(仅对 PAL 和 NTSC 而言)。在省略 MSB 的情况下，sub\_carrier 脉冲的幅值被量化为 ITU-R 推荐 601 的一种亮度信号。

**sub\_carrier\_phase** 这是一个 8 位整数，根据 ITU-R 470 号推荐中定义的场起始来定义在场同步基准上的参考 sub\_carrier 的相位。见表 I-6-16。

表 I-6-16 Sub-carrier-phase 的定义

sub_carrier_phase	相位
0	( $[360^\circ \div 256] * 0$ )
1	( $[360^\circ \div 256] * 1$ )
...	...
255	( $[360^\circ \div 256] * 255$ )

## II -6. 3. 11 量化矩阵扩展

每个量化矩阵都有一套缺省值。当 sequence\_header\_code 被解码时,所有矩阵都将被复位成缺省值。用户定义矩阵可以被加载,这可以在 sequence\_header() 或 quant\_matrix\_extension() 中发生。

对于 4 : 2 : 0 数据,仅有两种矩阵被用到,一个针对内部块,一个针对非内部块。

对于 4 : 2 : 2 或 4 : 4 : 4 的数据,四种矩阵被用到。内部和非内部矩阵均为亮度块和色差块而提供。然而值得注意的是,有可能将同一用户定义矩阵同时加载到亮度和色差矩阵中。

对于内部块(亮度和色差)的默认值矩阵为:

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

对于非内部块(亮度和色差)的默认值矩阵为:

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

**load\_intra\_quantiser\_matrix** 这是一个 1 位的标志,如果跟有 intra\_quantiser\_matrix,则被置为“1”。如果它被置为“0”,则表明将要用到的值没有改变。注意,如果这个标志为零,则由于 sequence\_header\_code 将把矩阵复位成默认值,在 sequence\_header() 中将使用默认值。

**intra\_quantiser\_matrix** 这是 64 个 8 位无符号整数的表。按 I -7. 3. 1 中所描述的默认“之”字形扫描顺序编码的这些新值将代替原来的值。第一个值通常为 8。对于所有这些无符号 8 位整数,禁止为零值。对于 4 : 2 : 2 和 4 : 4 : 4 数据,亮度内部矩阵和色差内部矩阵都将使用到这些新值,然而色差内部矩阵可以接着装入一个不同的矩阵。

**load\_non\_intra\_quantiser\_matrix** 这是一个 1 位的标志,如果后面跟有 non\_intra\_quantiser\_matrix,则被置为“1”。如果它被置为“0”,则将要用到的值没有改变。注意如果这个标志为零,则由于 sequence\_header\_code 将要把矩阵复位成其缺省值,在 se-

quence\_header()中将使用缺省值。

**non\_intra\_quantiser\_matrix** 这是 64 个 8 位无符号整数的表,按 I-7.3.1 中所描述的缺省“之”字形扫描顺序编码的新值将取代原来的值。这些 8 位无符号整数不可为零值。对于 4:2:2 和 4:4:4 数据,亮度非内部矩阵和色差非内部矩阵将使用这些新值。然而色差非内部矩阵可以接下来装入一个不同的矩阵。

**load\_chroma\_intra\_quantiser\_matrix** 这是一个 1 位标志,如果后面跟着 chroma\_intra\_quantiser,则被置为“1”。如果它被置为“0”,则表明将要用的值没有变化。如果 chroma\_format 为“4:2:0”,则这个标志为“0”。

**chroma\_intra\_quantiser\_matrix** 这是 64 个 8 位无符号整数的表。按 I-7.3.1 中所描述的缺省“之”字形扫描顺序编码的新值将取代原来的值,第一个值通常为 8。所有 8 位整数均不可为零。

**load\_chroma\_non\_intra\_quantiser\_matrix** 这是一个 1 位标志,如果置为“1”,则表明后面跟有 chroma\_non\_intra\_quantiser\_matrix。如果为“0”,则将要用到的值中没有改变。如果 chroma\_format 为“4:2:0”,这个标志取值为“0”。

**chroma\_non\_intra\_quantiser\_matrix** 这是 64 个 8 位无符号整数的表。按 I-7.3.1 中所描述的缺省“之”字形扫描顺序编码的新值将取代原来的值。所有 8 位无符号整数均不可为零。

## II-6.3.12 图象显示扩展

本规范未定义显示过程。这个扩展中的信息不影响解码过程,可以被与本规范一致的解码器忽略。

图象显示扩展允许在 sequence\_display\_extension() 中定义的显示矩形的位置以一幅幅图象方式进行移动,这在全景扫描的实现中有所应用。

**frame\_centre\_horizontal\_offset** 这是一个 16 位有符号的整数,以 1/16 样点为单位给出水平位移。正值表明重构帧的中心位于显示矩形中心的右边。

**frame\_centre\_vertical\_offset** 这是一个 16 位有符号的整数,以 1/16 样点为单位给出竖直位移。正值表明重构帧的中心位于显示矩形中心的下方。

矩形区的维数在 sequence\_display\_extension() 中定义。编码图内区域的坐标在 picture\_display\_extension() 中定义。由于(在隔行序列情况下)一个编码图可以与一个、两个或三个解码场有关,picture\_display\_extension() 可以最多包括三个位移量。

picture\_display\_extension() 中的帧中心偏移量的数目按如下定义:

```
if((progressive_sequence==1) || (picture_structure=="field")){
    number_of_frame_centre_offsets=1
}else{
    if (repeat_first_field=="1")
        number_of_frame_centre_offsets=3
    else
```

```

    number_of_frame_centre_offsets = 2
}

```

除非前面的 sequence\_header 后跟有一个 sequence\_display\_extension(), 否则 picture\_display\_extension() 将不出现。

如果给定的图中没有 picture\_display\_extension(), 那么将使用最近解码帧的中心位移。注意, 丢失的帧中心位移量有相同值(即使是在 picture\_display\_extension() 已经出现, 包括两个或三个帧中心位移的情况下)。跟在 sequence\_header() 后面, 所有帧的中心位移量都将取零值, 直到有 picture\_display\_extension() 定义了非零值。

图 I - 6-16 说明了图象显示参数, 如所示, picture\_display\_extension() 中的帧中心位移将通过显示矩形的中心来规定重构帧的中心位置。

注意 显示矩形也有可能比重构帧大。

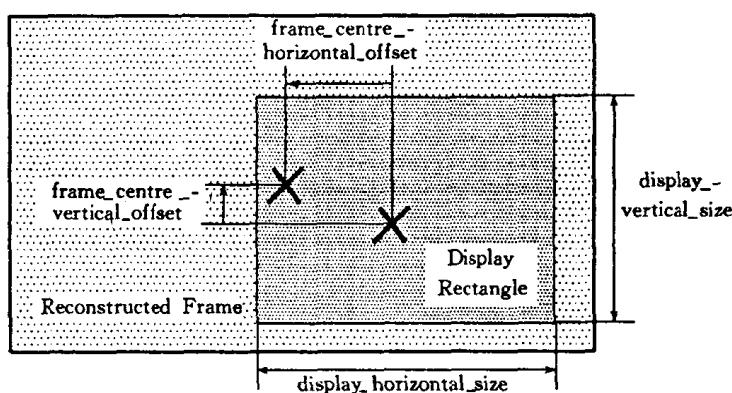


图 I - 6-16 帧中心偏移参数

注意 即使是在场图中, frame\_centre\_vertical\_offset 仍以帧行的 1/16 为单位表示帧中心位移(而不是场中的一行)。

#### I - 6. 3. 12. 1 全景扫描

帧中心位移可被用来实现全景扫描, 将定义一个在整个帧中漫游的矩形区域。

仅作举例, 这种方法可以用来在一个 9/16 格式的编码图中定义一个 3/4 高宽比的窗口, 这将允许解码器将一个针对增强型号电视的编码格式转化为可用于传统电视机的图象。这个 3/4 高宽比的区域将包含这个图象中“最感兴趣”的区域。

这个 3/4 的区域由 display\_horizontal\_size 和 display\_vertical\_size 定义。9/16 的帧尺寸由 horizontal\_size 和 vertical\_size 定义。

#### II - 6. 3. 13 图象时域分级扩展

**reference\_select\_code** 这是一个靠图象类型来识别用于预测的参考帧或参考场的 2 位编码。

**forward\_temporal\_reference** 这是一个无符号整数值, 指明了用于提供前向预测的低层帧的时域参考。如果低层用 10 位以上指明时域参考, 则对其低有效位进行编码。如

果低层用少于 10 位来指明时域参考，则对所有位进行编码，高有效位将被置为零。

**backward\_temporal\_reference** 这是一个无符号整数值，指明了用于提供后向预测的低层帧的时域参考。如果低层用 10 位以上指明时域参考，则对其低有效位进行编码，如果低层用少于 10 位来指明时域参考，则对所有位进行编码，高有效位将被置为零。

### II -6. 3. 14 图象空域分级扩展

**lower\_layer\_temporal\_reference** 指明用来提供预测的低层帧的时域参考的一个无符号整数。如果低层用 10 位以上指明时域参考，则对其低有效位进行编码。如果低层用少于 10 位指明时域参考，则对所有位进行编码，高有效位将置零。

**lower\_layer\_horizontal\_offset** 这个带符号整数规定了上采样低层帧相对于增强层图象的水平位移(左上角)，以增强层图象样本宽度为单位表示。如果色差格式为 4：2：0 或 4：2：2，则这个参数为偶数。

**lower\_layer\_vertical\_offset** 这个带符号整数规定了上采样低层帧相对于增强层图象的竖直位移(左上角)，以增强层图象样本高度为单位表示。如果色差格式为 4：2：0，则这个参数为偶数。

**spatial\_temporal\_weight\_code\_table\_index** 它指出应使用 I -7. 7 中定义的那个时空权重表。**spatial\_temporal\_weight\_code\_table\_index** 的允许值在表 I -7-21 中定义。

**lower\_layer\_progressive\_frame** 如果低层帧为隔行帧，则这个标志被置为 0；如为逐行帧，则置 1。这个标志在空域分级上采样过程中的应用在 I -7. 7 中定义。

**lower\_layer\_deinterlaced\_field\_select** 如 I -7. 7 中所述，它影响空域分级上采样过程。

### II -6. 3. 15 组块

**slice\_start\_code** **slice\_start\_code** 是一个 32 位的串。开始的 24 位为十六进制值 000001，后 8 位是 **slice\_vertical\_position**，在十六进制区间 01~AF 间取值。

**slice\_vertical\_position** 这由 **slice\_start\_code** 的后 8 位给出。它是一个无符号整数，以宏块数为单位给出第一个宏块在组块中的竖直位置。

在大型图中(帧的垂直尺寸大于 2800 行)，**slice\_vertical\_position** 被 **slice\_vertical\_position\_extension** 扩展。

宏块行可按以下计算：

```
if (vertical_size>2800)
    mb_row=(slice_vertical_position_extension<<7)+silce_vertical_position
            -1;
else
    mb_row=slice_vertical_position-1;
```

宏块第一行的 **slice\_vertical\_position** 为 1。有些组块可以有相同的 **slice\_vertical\_position**，这是由于组块可以在任何地方开始和结束。**slice\_vertical\_position** 的最大值为 175，除非在 **slice\_vertical\_position\_extension** 出现的情况下，此时 **slice\_vertical\_posi-**

tion 在[1 : 128]之间取值。

**priority\_breakpoint** 这个 7 位整数指明了比特流在语法中的分割点。允许值和语义解释在表 I - 7 - 30 中给出，在 1 分区中 priority\_breakpoint 取值零。

**quantiser\_scale\_code** 1~31 之间的一个无符号整数。解码器将一直使用这个值，直到在 slice() 或 macroblock() 中出现另一个 quantiser\_scale\_code。它不能为零值。

**intra\_slice\_flag** 这个标志置“1”表明比特流中出现 intra\_slice 和 reserved\_bits。

**intra\_slice** 如果组块中有任一宏块是非内部宏块，则这个值被置为“0”。如果所有宏块都是内部宏块，则 intra\_slice 可被置为“1”。intra\_slice 可以从比特流中被省略（通过对 intra\_slice\_flag 置“0”），此时认为它的值为零。

解码过程不使用 intra\_slice。intra\_slice 将被用来帮助 DSM 应用实现 FF/FR（见 II - D.12）。

**reserved\_bits** 这 7 个位都将为零。

**extra\_bit\_slice**——指明下面将要出现附加信息。如果 extra\_bit\_slice 被置为“1”，则后面将跟着 extra\_information\_slice。如果置为“0”，后面则没有数据。extra\_bit\_slice 将被置为“0”，“1”值为 ITU-T | ISO/IEC 将来可能定义的扩展而保留。

**extra\_information\_slice** 保留。与本规范相一致的解码器如果在比特流中遇到 extra\_information\_slice，则将其忽略。（即，从比特流中分析出然后舍弃）与本规范一致的比特流不包含这一语法单元。

### II - 6. 3. 16 宏块

注意 ISO/IEC-11172-2 所支持的“宏块填充”将不在本规范所定义的比特流中使用。

**macroblock\_escape** macroblock\_escape 是一个固定的位串“0000 0001 000”，当 macroblock\_address 和 previous\_macroblock\_address 的差值大于 33 时被使用。它使得 macroblock\_address\_increment 的值比后面的 macroblock\_escapes 和 macroblock\_address\_increment 码字解码所得的值大 33。

例如，如果在 macroblock\_address\_increment 之前有两个 macroblock\_escape 码字，则 macroblock\_address\_increment 所表示的值要加 66。

**macroblock\_address\_increment** 这是一个由附录 I - B 的表 I - B - 1 而来的变长码整数，指出了 macroblock\_address 和 previous\_macroblock\_address 间的差值。macroblock\_address\_increment 的最大值是 33。如果所需值比 33 大，则要用 macroblock\_escape 来编码。

macroblock\_address 是一个定义当前宏块绝对位置的变量。左上角的宏块的 macroblock\_address 为零。

previous\_macroblock\_address 是定义上一个不跳过宏块（跳过宏块的定义见 I - 7. 6. 6）绝对位置的变量，这不包括一个组块起始时的情况。在一个组块的起始，previous\_macroblock\_address 按如下复位：

`previous_macroblock_address = (mb_row * mb_width) - 1`

一个宏块在一幅图象中以宏块数为单位的水平空间位置(`mb_column`)可以通过`macroblock_address`按如下方法计算:

`mb_column = macroblock_address % mb_width`

其中`mb_width`是图象中一行内的宏块数目。

除了在一个组块的开始以外,如果从`macroblock_address_increment`和`macroblock_escape`(如果有)所恢复的`macroblock_address`的值与`previous_macroblock_address`的值之差超过1,则说明跳过了一些宏块。以下是一些规定:

- 在I-图中不得有跳过的宏块,除了当:当前图的`picture_header()`后跟有`picture_spatial_scalable`,或比特流中出现`sequence_scalable_extension()`且`scalable_mode`="SNR scalability"。
- 组块的第一个和最后一个宏块不可被跳过。
- 在B-图中,跳过的宏块不能紧跟在`macroblock_intra`为1的宏块后面。

#### I-6.3.16.1 宏块模式

**macroblock\_type** 这是一个按变长码编码的指示器,指明了与表I-B-2到表I-B-8有关的编码类型和宏块内容,由`picture_coding_type`和`scalable_mode`选定。

**macroblock\_quant** 这个值根据表I-B-2到I-B-8由`macroblock_type`中得出。置为“1”表示比特流中出现`quantiser_scale_code`。

**macroblock\_motion\_forward** 根据表I-B-2到表I-B-8,由`macroblock_type`求得。这个标志对比特流语法有影响,并被解码过程使用。

**macroblock\_motion\_backward** 根据表I-B-2到表I-B-8,由`macroblock_type`求得。这个标志影响比特流语法,并被解码过程使用。

**macroblock\_pattern** 根据表I-B-2~表I-B-8,由`macroblock_type`求得。这个值被置1,表明`coded_block_pattern()`在比特流中出现。

**macroblock\_intra** 根据表I-B-2~表I-B-8,由`macroblock_type`求得。这个标志对比特流语法有影响,并被解码过程使用。

**spatial\_temporal\_weight\_code\_flag** 由`macroblock_type`求得,指出`spatial_temporal_weight_code`是否在比特流中出现。

**spatial\_temporal\_weight\_code** 这是一个两位编码,就空域分级而言,它指明了怎样将空间预测和时间预测组合起来形成对宏块的预测。I-7.7中给出了形成空域分级预测的完整描述。

**frame\_motion\_type** 这个两位编码指明了宏块运动预测,定义于表I-6-17。

如果`frame_pred_frame_dct`等于1,则`frame_motion_type`从比特流中省略。在这种情况下,运动向量的编码和预测结构应好像`frame_motion_type`已指明了“基于帧的预测”一样而被实现。

就内部宏块而言(在一个帧图内),当`concealment_motion_vectors`等于1时,`frame_motion_type`不在比特流中出现。在这种情况下,运动向量解码和运动向量预测器的修改

应好像 frame\_motion\_type 已指明了“基于帧的预测”一样而被实现, 见 I -7.6.3.9。

表 II -6-17 frame\_motion\_type 的含义

编码	spatial_temporal_weight_class	预测类型	motion_vector_count	mv_format	dmv
00		保留			
01	0,1	基于场的预测	2	场	0
01	2,3	基于场的预测	1	场	0
10	0,1,2,3	基于帧的预测	1	帧	0
11	0,2,3	双基	1	场	1

**field\_motion\_type** 这个两位编码指明了宏块运动预测, 如表 II -6-18 中所定义。

表 II -6-18 field\_motion\_type 的含义

编码	spatial_temporal_weight_class	预测类型	motion_vector_count	mv_format	dmv
00		保留			
01	0,1	基于场的预测	1	场	0
10	0,1	16x8MC	2	场	0
11	0	双基	1	帧	1

就内部宏块(在一个场图内)而言, 当 concealment\_motion\_vectors 等于 1 时, 比特流中没有 field\_motion\_type。在这种情况下, 运动向量的解码和运动向量预测器的修改应好像 field\_motion\_type 已指明了“基于场的预测”一样而被实现, 见 I -7.6.3.9。

**decode\_dct\_type** 这个标志(从各种比特流元素中获得)决定了下面的语法单元 dct\_type 是否在比特流中出现。它按如下方法获得:

```
if((picture_structure == "frame")&&(frame_pred_frame_dct
    == 0)&&(macroblock_intra || macroblock_pattern)){
    decode_dct_type = 1;
} else
    decode_dct_type = 0;
```

**dct\_type** 这个标志指明了宏块是帧 DCT 编码的还是场 DCT 编码的。如果它被置为“1”, 则宏块是场 DCT 编码的。仅当 decode\_dct\_type 非零时, 比特流中才包含 dct\_type。

如果 decode\_dct\_type 为零, 则 dct\_type(在解码过程的剩余部分使用)将得到如表 II -6-19 所示。

表 II-6-19 如果比特流中没有 dct\_type 时, dct\_type 的取值

条 件	dct_type
picture_structure = "field"	由于在场图中没有帧/场的差异而不被使用
frame_pred_frame_dct = 1	0("frame")
! (macroblock_intra    macroblock_pattern)	未使用——宏块未被编码
宏块被跳过	未使用——宏块未被编码

### II-6.3.16.2 运动向量

如表 II-6-17 和表 II-6-18 中所指出的那样, motion\_vector\_count 由 field\_motion\_type 或 frame\_motion\_type 中获得。

如表 II-6-17 和表 II-6-18 中所指出的那样, mv\_format 由 field\_motion\_type 或 frame\_motion\_type 中获得。mv\_format 指出了运动向量是一个场运动向量还是一个帧运动向量。mv\_format 在运动向量语法和运动向量预测过程中使用。

如表 II-6-17 和表 II-6-18 所示, dmv 从 field\_motion\_type 或 frame\_motion\_type 中获得。

**motion\_vertical\_field\_select[r][s]** 这个标志指明了哪一个参考场将被用来形成预测。如果 motion\_vertical\_field\_select[r][s] 为零, 则用顶参考场, 如果为 1, 则用底参考场。(标号 r 和 s 的含义见表 II-7-7)

### II-6.3.16.3 运动向量

**motion\_code[r][s][t]** 这个变长码用于运动向量解码, 如 II-7.6.3.1 中所述。(标号 s,r,t 的含义见表 II-7-7)

**motion\_residual[r][s][t]** 这个整数用于运动向量解码, 如 II-7.6.3.1 中所述。(标号 r,s,t 的含义见表 II-7-7) 比特流中 motion\_residual[r][s][t] 的位数为 r\_size, 由 f\_code[s][t] 中获得, 如下所示:

$$r\_size = f\_code[s][t] - 1$$

注意 motion\_residual[0][s][t] 和 motion\_residual[1][s][t] 的位数由 f\_code[s][t] 指明。

**dmvector[t]** 这个变长码用于运动向量解码, 如 II-7.6.3.1 中所述。(标号 t 的含义见表 II-7-7)

### II-6.3.16.4 编码块模式

**coded\_block\_pattern\_420** 是一个变长码, 根据表 II-B-9, 来求得变量 cbp。

**coded\_block\_pattern\_1**

**coded\_block\_pattern\_2** 对于 4:4:4 和 4:2:2 的数据, 编码块模式由附加的两位或六位固定长度码所扩展, 即 coded\_block\_pattern\_1 或 coded\_block\_pattern\_2。然后, 用下面方法可由 cbp 得到 pattern\_code[i]:

```
for(i=0;i<12;i++)
    if(macroblock_intra != 0)
```

```

    pattern_code[i]=1;
else
    pattern_code[i]=0;
if (macroblock_intra==0){
    for(i=0;i<6;i++)
        if (cbp&(1<<(5-i)))pattern_code[i]=1;
    if (chroma_format=="4 : 2 : 2")
        for(i=6;i<8;i++)
            if (coded_block_pattern_1&(1<<(7-i)))pattern_code[i]=1;
    if (chroma_format=="4 : 4 : 4")
        for(i=6;i<12;i++)
            if(coded_block_pattern_2&(1<<(11-i)))pattern_code[i]=1;
}

```

如果 pattern\_code[i] 等于 1, i=0~(block\_count-1), 那么此宏块中包含图 I-6-8, I-6-9 和 I-6-10 中所定义的块号。

决定了宏块中块的数目的数字“block\_count”由色差格式获得,如表 I-6-20 所示。

**表 I-6-20 block\_count 作为 chroma\_format 的函数**

chroma_format	block_count
4 : 2 : 0	6
4 : 2 : 2	8
4 : 4 : 4	12

### I-6.3.17 块

第 7 章中将描述 block() 的语义。

## II - 7 视频解码过程

本章详述解码器将要执行的解码过程，实现从编码比特流中恢复图象数据。

除了反离散余弦变换(IDCT)，解码过程被定义为所有解码器都将产生同样数字结果。能够与这里所描述的解码过程产生同样结果的其它解码过程均被认为遵从本规范。

为了允许对其功能的不同实现方法，IDCT 以统计学方法进行定义。IDCT 规范由附录 II - A 给出。

在 II - 7.1 ~ II - 7.6 中，定义了最简单的解码过程，其中，分级特性没有被用到，II - 7.7 ~ II - 7.11 详述了使用分级扩展的解码过程。II - 7.12 定义了解码过程的输出。

图 II - 7 - 1 是未包括比例变换的视频解码过程的方框图。这个图为清楚起见进行了简化。

注意 贯穿本规范的二维数组以  $\text{name}[q][p]$  表示，其中  $q$  是竖直方向行数， $p$  是水平方向列数。

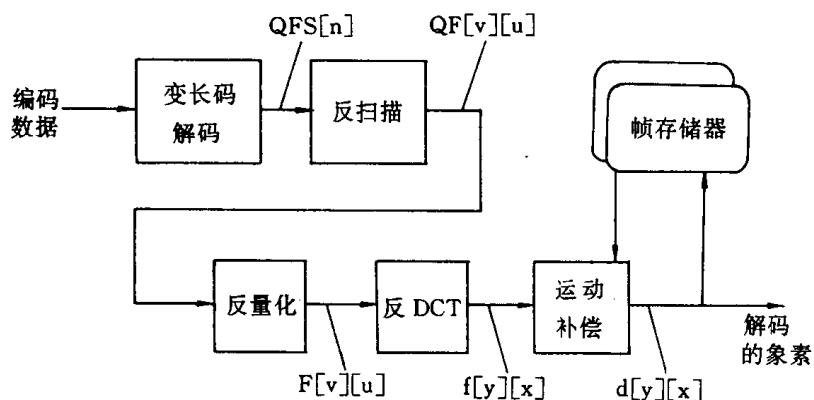


图 II - 7 - 1 简化的视频解码过程

### II - 7.1 高层语法结构

比特流中针对宏块和它以上的所有语法结构的各种参数和标志应按 II - 6 章中所指明的那样来解释。这些参数和标志中的大多数将对下面章节中描述的解码过程而产生影响。一旦一幅给定图象中的所有宏块均被处理过，则整幅图象将被重构。

重构场图将成对相伴而形成重构帧。（见 II - 6.3.10 中的“图结构 (picture-structure)”）

重构帧序列将按 II - 6.1.1.11 中所述进行重排序。

如果  $\text{progressive\_sequence} == 1$ ，则重构帧将以帧周期为时间间隔从解码过程输出，

如图 I-7-19 所示。

如果 progressive\_sequence == 0，则重构帧将被分成一个重构场的序列，在解码过程输出端以有规律的场周期为时间间隔而出现，如图 I-7-20 所示。如果一个帧图的 repeat\_first\_field == 1，则这个帧的第一场将在第二场后被重复。（见 I-6.3.10 中的“repeat\_first\_field”）

## II-7.2 变长码解码

I-7.2.1 将详述内部编码块中 DC 系数( $n=0$ )所使用的解码过程。(n 是在“之”字形扫描顺序中的系数索引)。I-7.2.2 说明所有其它系数的解码过程，包括 AC 系数( $n \neq 0$ )和非内部编码块中的 DC 系数。

用 cc 指代颜色分量。它与表 I-7-1 中所述块号相关。于是，对于 Y 分量，cc 为零，对 Cb 分量 cc 为 1，对于 Cr 分量 cc 为 2。

表 I-7-1 颜色分量索引 cc 的定义

块号	cc		
	4 : 2 : 0	4 : 2 : 2	4 : 4 : 4
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	1	1	1
5	2	2	2
6		1	1
7		2	2
8			1
9			2
10			1
11			2

### II-7.2.1 内部块中的 DC 系数

内部宏块中的块的 DC 系数按附录 I-B 的表 I-B-12 和 I-B-13 中所定义的变长码进行编码，并指明 dct\_dc\_size，如果 dct\_dc\_size 不等于零，则它后面要跟着一个固定长度码，以 dct\_dc\_size 为位数的 dc\_dct\_differential。差分值首先从编码数据中恢复，然后加到预测器上来恢复最终的解码系数。

如果 cc 为零，则 dct\_dc\_size 将用到表 I-B-12。如果 cc 非零，则 dct\_dc\_size 使用表 I-B-13。

有三个预测器，每个颜色分量有一个。每次对内部宏块中的块的 DC 系数进行解码时，预测器的值与差分值相加而恢复实际的系数。然后预测器再被置为刚刚解码的系数的

值。在下面所描述的不同时刻,预测器将被复位。复位值由表 II -7-2 中所规定的参数 intra\_dc\_precision 中获得。

表 II -7-2 intra\_dc\_precision 和预测器复位值的关系

intra_dc_precision	精确位	复位值
0	8	128
1	9	256
2	10	512
3	11	1024

在下面的各时刻预测器将被复位。

- 在一个组块的开始;
- 在对非内部宏块进行解码时;
- 当宏块被跳过时,即当 macroblock\_address\_increment>1 时。

预测器以 dc\_dct\_pred[cc] 表示。

QFS[0]通过 dc\_dct\_size 和 dc\_dct\_differential,按任意与下面等价的过程计算:

```

if (dc_dct_size == 0)  {
    dct_diff = 0;
} else {
    half_range = 2^(dc_dct_size - 1); 注意, ^ 表示乘方(而非 XOR)
    if (dc_dct_differential >= half_range)
        dct_diff = dc_dct_differential;
    else
        dct_diff = (dc_dct_differential + 1) - (2 * half_range);
}
QFS[0] = dc_dct_pred[cc] + dct_diff;
dc_dct_pred[cc] = QFS[0]

```

注意 dct\_diff 和 half\_range 是临时变量,不能在本规范其它地方使用。

比特流中有一个规定,QFS[0]在以下范围内取值:

$$0 \sim ((2^{(8+intra_dc_precision)}) - 1)$$

## II -7. 2. 2 其它系数

所有的系数,除了 DC 内部系数,都将使用表 I -B-14 和 I -B-15, I -B-16 来编码。

所有变长码首先都要使用表 I -B-14 或表 I -B-15 来解码,解码值指明了采取下面三种行动中的哪一种。

(1) 块结束。此时,块中再没有别的系数,块中剩余的系数(尚未得到解码值的)应被置为零。这由 I -6. 2. 6 的语法规范中的“End of block(块结尾)”指明。

(2) “正常”系数,其中 run 和 level 值的后面还跟有一个单一位 S,给出这个系数的符号。signed\_level 通过 level 和 S 计算得出,如下所示。run 个系数应被置为零,接下来的系数取值为 signed\_level:

```
if (s == 0)
    signed_level = level;
else
    signed_level = (-level);
```

(3) “换码”编码系数。其中 run 和 signed\_level 值为定长码编码,如 II-7.2.2.3 中所述。

#### II-7.2.2.1 表选择

表 II-7-3 指明了 DCT 系数的解码将用到哪个表。

表 II-7-3 DCT 系数 VLC 表的选择

intra_vlc_format	0	1
内部块 (macroblock_intra=1)	II-B-14	II-B-15
非内部块 (macroblock_intra=0)	II-B-14	II-B-14

#### II-7.2.2.2 非内部块的第一个系数

对一个非内部块(非内部宏块中的一个块)的第一个系数来说,表 II-B-14 按表下面的“注意 2”和“注意 3”所指出的那样被修改。

这个修改仅影响表示  $\text{run}=0, \text{level}=\pm 1$  的表项。由于不可能将块结束作为一个块的第一个系数编码(这种情况下块将为“非编码的”),所以不存在混淆的可能。

语法中使用这个修改表的地方由 II-6.2.6 的语法规范中“第一个 DCT 系数”指出。剩余的系数由“随后的 DCT 系数”指出。

注意 如果表 II-B-14 被一个内部块使用,则第一个系数将如 II-7.2.1 所述被编码。由于使用表 II-B-14 的第一个系数是块中的第二个系数,所以表 II-B-14 将不被修改。

#### II-7.2.2.3 换码编码

有许多 run 和 level 的组合,没有相应的变长码来表示它们。为了对这些从统计学上来看很少使用的组合进行编码,我们使用了换码编码。

表 II-B-16 定义了换码编码方法。换码 VLC 的后面跟着一个 6 位定长码给出了“run”,之后跟着一个 12 位定长码给出“signed\_level”的值。

注意 本规范中使用的换码编码方法与 ISO/IEC11172-2 中用到的不同。

#### II-7.2.2.4 小结

小结: II-7.2.2 节变长码解码过程与下面的过程等价。在过程的开始,对于非内部块 n 取“0”值,对于内部块 n 取“1”值。

```

eob_not_read=1;
while (eob_not_read)
{
    <decode VLC,decode Escape coded coefficient if required>
    if (<decode VLC indicates End of block>) {
        eob_not_read=0;
        while(n<64) {
            QFS[n]=0;
            n=n+1;
        }
    }else {
        for (m=0;m<run;m++) {
            QFS[n]=0;
            n=n+1;
        }
    }
    QFS[n]=signed_level
    n=n+1;
}
}

```

注意 eob\_not\_read 和 m 是临时变量,不能在本规范其它地方使用。

### II -7.3 反 扫 描

以 QFS[n] 表示在变长码解码器输出端的数据,n 在 0~63 的区间内。

这一节阐述将一维数据 QFS[n] 转换成以 QF[v][u] 表示的系数二维数组的方法。u 和 v 均在区间 0~7 中。

定义了两个扫描模板。将要用到的扫描将由图像头扩展中的 alternate\_scan 所决定。

对于 alternate-scan 为零的情况,图 II -7.2 定义了 scan[alternate\_scan][v][u]。如 alternate\_scan 为 1,图 II -7.3 定义 scan[alternate\_scan][v][u]。

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

图 II -7-2 scan[0][v][u] 的定义

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

图 I -7-3  $\text{scan}[1][v][u]$  的定义

反扫描应为任意与以下步骤相等价的过程：

```
for (v=0;v<8;v++)
    for (u=0;u<8;u++)
        QF[v][u]=QFS[scan[alternate_scan][v][u]]
```

### II -7.3.1 加载矩阵的反扫描

当量化矩阵被加载时,它们以一种扫描顺序在比特流中编码,被转化成反量化器中使用的二维矩阵,就像系数使用的方法一样。

对于矩阵加载,图 I -7-2(即零扫描)定义的扫描通常要被用到。

以  $W[w][u][v]$  表示反量化器中加权矩阵(见 I -7.4.2.1), $W[w][n]$  指明比特流中编码的矩阵。矩阵加载等价于下面的过程:

```
for (v=0;v<8;v++)
    for (u=0;u<8;u++)
        W[w][v][u]=W[w][scan[0][v][u]]
```

## II -7.4 反 量 化

系数的二维数组  $QF[v][u]$  被反量化而产生重构的 DCT 系数。这个过程的实质是以量化器步长为倍数的乘法运算,量化器步长大小以两种方法修改:在一个块内可以使用加权矩阵来修改步长大小,而使用比例因子修改步长可以节省位数(与用全新的加权矩阵相比)。

图 I -7-4 阐明了整个反量化过程。在适当的反量化算法之后,结果系数  $F''[v][u]$  被饱和而生成  $F'[v][u]$ ,之后执行一个解谐控制操作来给出最终的重构 DCT 系数  $F[v][u]$ 。

注意 本规范用来实现解谐控制的方法不同于 ISO/IEC 11172-2 中所使用的方法。

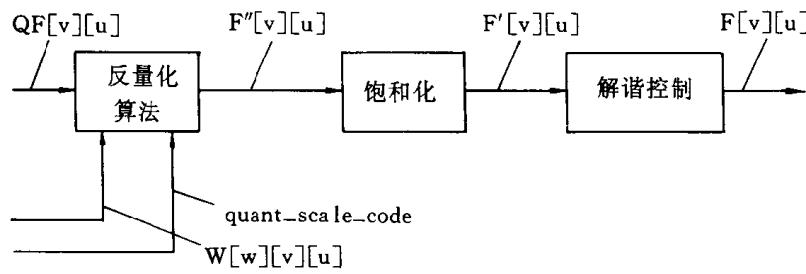


图 II -7-4 反量化过程

#### II -7.4.1 内部 DC 系数

内部编码块中的 DC 系数将以一种不同于所有其它系数的方法进行反量化。

在内部块中,  $F''[0][0]$  通过以一个常数因子与  $QF[0][0]$  相乘而得到, 这个常数因子是 intra\_dc\_mult, (一个既不会被加权矩阵修改, 也不会被比例因子修改的常数)。常数因子与在图象编码扩展中编码的参数 intra\_dc\_precision 有关。表 II -7-4 阐述了 intra\_dc\_precision 和 intra\_dc\_mult 间的关系。

表 II -7-4 intra\_dc\_precision 和 intra\_dc\_mult

intra_dc_precision	精度位数	intra_dc_mult
0	8	8
1	9	4
2	10	2
3	11	1

这样,  $F''[0][0] = \text{intra\_dc\_mult} \times QF[0][0]$

#### II -7.4.2 其它系数

除了内部块中的 DC 系数以外的其它系数都如本节所述, 被反量化。

##### II -7.4.2.1 加权矩阵

当使用 4 : 2 : 0 数据时, 要用到两个加权矩阵。一个用于内部宏块, 另一个用于非内部宏块。当使用 4 : 2 : 2 或 4 : 4 : 4 数据时, 用到四个矩阵, 允许亮度和色差数据使用不同矩阵。每个矩阵都有一套缺省值, 但可以通过加载用户定义的矩阵而被重写, 如 I -6.2.3.2 中解释的那样。

以  $W[w][v][u]$  表示加权矩阵, 其中  $w$  取值 0~3, 指明正在使用哪一个矩阵。表 II -7-5 总结了指导选择  $w$  的规则。

表 II -7-5 w 的选择

	4 : 2 : 0		4 : 2 : 2 和 4 : 4 : 4	
	亮度 (cc=0)	色差 (cc≠0)	亮度 (cc=0)	色差 (cc≠0)
内部块 (macroblock_intra=1)	0	0	0	2
非内部块 (macroblock_intra=0)	1	1	1	3

#### II -7.4.2.2 量化器比例因子

量化比例因子以固定长度码被编码,即 quantiser\_scale\_code。这个值指明了用于反量化算法中的适当的 quantiser\_scale。

q\_scale\_type(编码于图象编码扩展中)指明了 quantiser\_scale\_code 和 quantiser\_scale 间的两个映象哪一个适用。表 II -7-6 示出了 quantiser\_scale\_code 和 quantiser\_scale 间的两组映象。

表 II -7-6 quantiser\_scale 和 quantiser\_scale\_code 的关系

quantiser_scale_code	quantiser_scale[q_scale_type]	
	q_scale_type=0	q_scale_type=1
0	(禁用)	
1	2	1
2	4	2
3	6	3
4	8	4
5	10	5
6	12	6
7	14	7
8	16	8
9	18	10
10	20	12
11	22	14
12	24	16
13	26	18
14	28	20
15	30	22
16	32	24

续表

quantiser-scale-code	quantiser-scale[q-scale-type]	
	q-scale-type=0	q-scale-type=1
17	34	28
18	36	32
19	38	36
20	40	40
21	42	44
22	44	48
23	46	52
24	48	56
25	50	64
26	52	72
27	54	80
28	56	88
29	58	96
30	60	104
31	62	112

#### II -7.4.2.3 重构公式

下面的等式阐明了由  $QF[v][u]$  重构  $F''[v][u]$  的算法(针对除了内部 DC 系数以外的所有系数)。

$$F''[v][u] = ((2QF[v][u] + k) * W[w][v][u] * quantiser\_scale) / 32$$

其中：

$$k = \begin{cases} 0 & \text{内部块} \\ \text{Sign}(QF[v][u]) & \text{非内部块} \end{cases}$$

注意 上面的公式使用了 II -4.1 中定义的“/”运算符。

#### II -7.4.3 饱和化

从反量化算法中得到的结果系数被饱和化在  $[-2048, +2047]$  范围内。于是

$$F''[v][u] = \begin{cases} 2047 & F''[v][u] > 2047 \\ F''[u][v] & -2048 \leq F''[v][u] \leq 2047 \\ -2048 & F''[v][u] < -2048 \end{cases}$$

## II -7.4.4 解谐控制

解谐控制将由下面等价的任何过程实现。首先，块中所有的这些重构的、饱和化的系数  $F'[v][u]$  被求和。然后检测这个值为奇还是偶。如果和为偶，则要对系数  $F[7][7]$  进行调整修改。于是：

$$\begin{aligned} \text{sum} &= \sum_{v=0}^{v<8} \sum_{u=0}^{u<8} F'[v][u] \\ F[v][u] &= F'[v][u] \quad \text{对于所有除 } u=v=7 \text{ 外的 } u, v \\ F[7][7] &= \begin{cases} F'[7][7] & \text{sum 为奇} \\ F'[7][7]^{-1} & \text{如果 } F'[7][7] \text{ 为奇} \\ F'[7][7]^{-1} & \text{如果 } F'[7][7] \text{ 为偶} \end{cases} \end{aligned}$$

注意 1 以上对  $F[7][7]$  的修改可以通过对系数补码表示的最低有效位进行开关而简单实现。并且，既然我们仅对和(sum)的“奇”或“偶”感兴趣，我们可以用“异或”来计算“和”(sum)(仅对最低有效位)。

注意 2 警告。输入到 IDCT 的小非零值对于兼容 IDCT 可能产生零值输出。如果编码器中出现这种情况，则对于使用不同的兼容 IDCT 的解码器，其中的某些图象可能发生不匹配。编码器可以通过检查它自己的 IDCT 输出来避免这个问题。

## II -7.4.5 小结

作为小结，反量化过程即为任何与下面在数字结果上相等价的过程：

```

for(v=0;v<8;v++) {
    for (u=0;u<8;u++) {
        if ((u==0)&&(v==0)&&(macroblock_intra)) {
            F''[v][u]=intra_dc_mult * QF[v][u];
        } else {
            if (macroblock_intra) {
                F''[v][u]=(QF[v][u] * W[w][v][u] * quantiser_scale * 2)/32;
            }else{
                F''[v][u]=(((QF[v][u] * 2)+Sign(QF[v][u])) * W[w][v][u]
                           * quantiser_scale * 2)/32;
            }
        }
    }
    sum=0;
    for (v=0;v<8;v++) {
        for (u=0;u<8;u++) {
            if (F''[v][u]>2047) {

```

```

        F'[v][u]=2047;
    } else {
        if (F''[v][u]<-2048){
            F'[v][u]=-2048;
        } else {
            F'[v][u]=F''[v][u];
        }
    }
    sum=sum+F'[v][u];
    F[v][u]=F'[v][u];
}
}

if((sum&1)==0) {
    if ((F[7][7]&1)!=0){
        F[7][7]=F'[7][7]-1;
    } else {
        F[7][7]=F'[7][7]+1;
    }
}

```

## II -7.5 反 DCT

一旦 DCT 系数  $F[v][u]$  被重构,附录 II -A 中定义的反 DCT 变换将被用于获得反变换值  $f[y][x]$ 。这些值将被饱和为  $-256 \leq f[y][x] \leq 255$ , (对于所有  $x,y$ )。

### II -7.5.1 非编码的块和跳过的宏块

在一个未跳过的宏块中,如果对于宏块中一个给定块,  $pattern\_code[i]$  为 1, 则那个块的系数数据包含在比特流中。这将使用前面章节中阐述的方法解码。

然而,如果  $pattern\_code[i]$  为零,或者如果宏块被跳过,那么那个块不包含系数数据。这样的块的样本域系数  $f[y][x]$  都将取零值。

## II -7.6 运动补偿

运动补偿过程由以前的解码图象形成预测,与系数数据(从 IDCT 输出)组合起来恢复最终的解码样本。图 II -7-5 示出了这个过程的简化图解。

通常,对于每个块最多形成四个独立的预测,组合在一起构成最后的预测块  $p[y][x]$ 。

对于内部编码宏块,不形成预测,所以  $p[y][x]$  为零。当需要从  $f[y][x]$  中去掉负值时,仍需用到图 II - 7-5 所示的饱和化过程。内部编码的宏块可以带有被称为“隐藏运动向量”的运动向量,除此之外,在一般情况下,均不产生预测。当比特流错误妨碍了系数信息的解码时,才用到这种运动向量。解码器如何使用这种运动向量的方法并未被述及。对于这些运动向量的唯一要求是它们应符合运动向量的正确语法。对于这些运动向量的可能的使用方法描述见 II - 7.6.3.9。

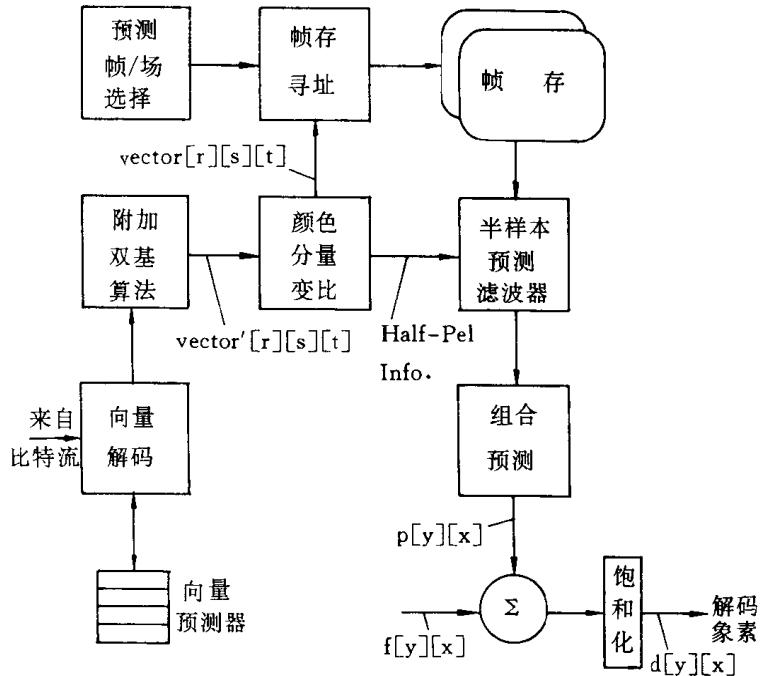


图 II - 7-5 简化的运动补偿过程

如果一个块未被编码,则可能是由于整个宏块是被跳过的,或者是由于这个未被编码的块没有系数数据。在这种情况下,  $f[y][x]$  为零,解码样本仅为预测值  $p[y][x]$ 。

## II - 7.6.1 预测方式

预测方式有两种主要分类:场预测和帧预测。

在场预测中,通过使用一个或多个以前的解码场中的数据,每个场都可以独立地产生预测。帧预测由一个或多个以前的解码帧形成帧的预测。我们必须理解:从中产生预测的这些场和帧本身是可以按场图或帧图解码的。

在一个场图内,所有预测都是场预测。而在帧图中,既可能用到场预测,也可能用到帧预测(在宏块基础上选择)。

除了场和帧预测的主要分类,还使用到两种特殊的预测方法:

- $16 \times 8$  运动补偿。其中,每个宏块使用两个运动向量,第一个运动向量用于上面的  $16 \times 8$  区域,第二个运动向量用于下面的  $16 \times 8$  区域。对于双向预测宏块,总共要用到四个运动向量,两个用于前向预测,两个用于后向预测。在本规范中, $16 \times 8$  运动补偿仅用于场图。

- 双基。其中，在比特流中，仅有一个运动向量（完整格式）和一个小差分向量被编码。对于场图，从这个信息中可以得到两个运动向量，它们用于分别从两个参考场（一个顶场，一个底场）中形成预测，之后被平均来形成最终的预测。对于帧图，两个场都重复这一过程，所以总共形成四个场预测。这种方式仅适用于在参考场（帧）与被预测场（帧）间没有B-图的P-图。

## II -7.6.2 预测场和帧的选择

采用哪些场和帧来形成预测将在本节中详述。

### I -7.6.2.1 场预测

在P图中，预测由最近的两个解码帧做出。图I-7-6所示为一种最简单的情况，用于编码帧的第一幅图的预测或帧图的场预测。在这些情况下，两个参考场是同一重构帧的组成部分。

**注意1** 参考场本身可能是由两个场图或一个单一帧图重构而成。

**注意2** 在对一个场图预测时，被预测的场既可为顶场也可为底场。

对一个编码帧的第二个场图进行预测更为复杂，因为要用到两个最近被解码的参考场，而最近的一个参考场要通过对这个编码帧的第一个场图解码而得。图I-7-7阐明了当此第二个场图为底场时的情形。图I-7-8阐明了当此第二个场图为顶场时的情形。

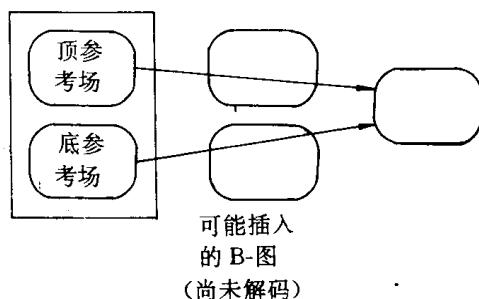


图 I -7-6 第一个场图的预测或帧图中的场预测

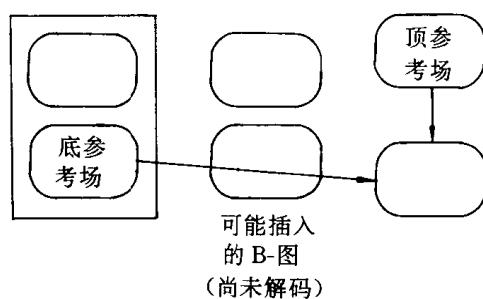


图 I -7-7 当第二个场图为底场时的预测

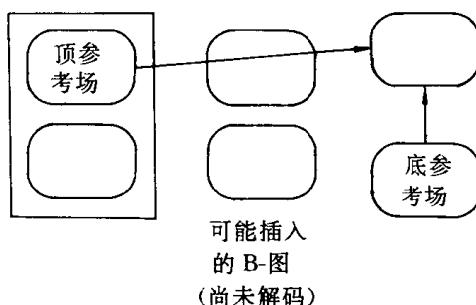


图 I -7-8 当第二个场图为顶场时的预测

**注意** 最早的参考场本身可由场图或帧图重构。

B-图中的场预测应由最近重构的两个参考帧的两场作出。图I-7-9阐述了这种情形。

**注意** 参考帧本身可能是由两个场图或一个单一的帧图重构。

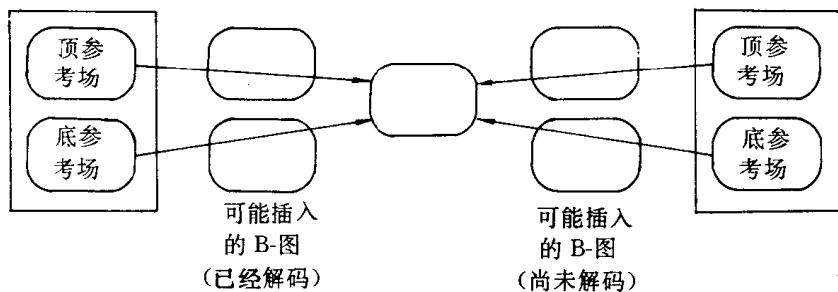


图 II -7-9 B 帧图或 B 场图的场预测

### II -7.6.2.2 帧预测

在 P-图中,从最近的重构参考帧作预测,如图 II -7-10 所述。

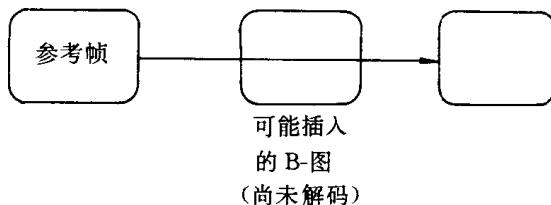


图 II -7-10 I-图和 P-图的帧预测

注意 参考帧本身作为两个场图和一个单一帧图编码。

类似地,B-图中的帧预测由两个最近的重构参考帧作出,如图 II -7-11 所述。

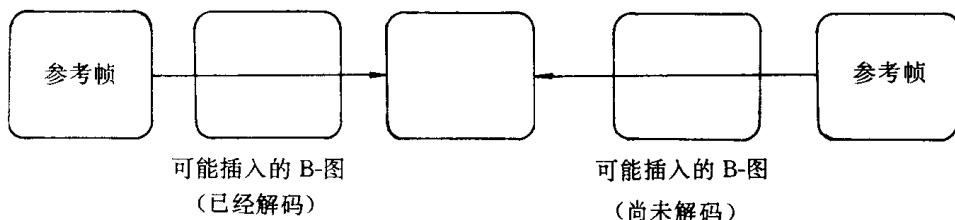


图 II -7-11 B-图的帧预测

注意 参考帧本身可以作为两场图或一个单一帧图编码。

### II -7.6.3 运动向量

为了减少表示运动向量所需的位数,它们根据前一运动向量进行差分编码。为了对运动向量进行解码,解码器将保持四个运动向量预测器(每个均有一个竖直分量,一个水平分量),以  $PMV[r][s][t]$  表示。对于每个预测,先得出一个运动向量  $vector'[r][s][t]$ 。然后根据样本结构( $4:2:0, 4:2:2$  或  $4:4:4$ )对其进行变化,为每个颜色分量给出一个运动向量  $vector[r][s][t]$ 。这个数组中维数的含义定义于表 II -7-7 中。

表 II -7-7 在 PMV[r][s][t],vector[r][s][t]和 vector'[r][s][t]中 r,s 和 t 标号的含义

	0	1
r	宏块中的第一个运动向量	宏块中的第二个运动向量
s	前向运动向量	后向运动向量
t	水平分量	竖直分量

### II -7. 6. 3. 1 运动向量解码

每个运动向量分量 vector'[r][s][t]都应按与下面等价的过程计算。注意运动向量预测器也应按这个过程修改。

```

r_size=f_code[s][t]-1
f=1<<r_size
high=(16*f)-1;
low=(( -16)*f);
range=(32*f);
if((f==1) || (motion_code[r][s][t]==0))
    delta=motion_code[r][s][t];
else {
    delta=((Abs(motion_code[r][s][t])-1)*f)+motion_residual[r][s][t]+
    1;
    if (motion_code[r][s][t]<0)
        delta=-delta;
}
prediction=PMV[r][s][t];
if ((mv_format == "field")&&(t == 1)&& (picture_structure == "Frame pic-
ture"))
    prediction=PMV[r][s][t]DIV 2;
vector'[r][s][t]=prediction+delta;
if (vector'[r][s][t]<low)
    vector'[r][s][t]=vector'[r][s][t]+range;
if (vector'[r][s][t]>high)
    vector'[r][s][t]=vector'[r][s][t]-range;
if ((mv_format == "field")&&(t == 1)&& (picture_structure == "Frame picture"))
    PMV[r][s][t]=vector'[r][s][t]*2;
else
    PMV[r][s][t]=vector'[r][s][t];

```

比特流中的参数应使重构差分运动向量 delta 在区间[low : high]中。此外，重构运动

向量  $\text{vector}'[r][s][t]$  和修改的运动向量预测器的值  $\text{PMV}[r][s][t]$  也应在  $[\text{low} : \text{high}]$  范围内。

$r\_size, f, \delta, \text{high}, \text{low}$  和  $\text{range}$  都是临时变量, 不在本规范其余地方出现。

$\text{motion\_code}, \text{motion\_residual}$  和  $\text{mv\_format}$  是从比特流中恢复的域。

$r, s, t$  指明了正在被处理的特定运动向量分量, 含义见表 I-7-7。

$\text{vector}'[r][s][t]$  是宏块亮度分量的最终重构运动向量。

### I-7.6.3.2 运动向量限制

在帧图中, 场运动向量的竖直分量应受限制, 以使它们仅覆盖与那些运动向量相关的  $f\_code$  所支持的范围的一半。这个限制保证了运动向量预测器总能具有对于后续帧运动向量的解码合适的值。表 I-7-8 总结了运动向量的大小, 它们可以作为  $f\_code$  的一个函数被编码。

表 I-7-8 作为  $f\_code[s][t]$  的函数的运动向量允许范围

$f\_code[s][t]$	帧图中场向量的水平分量 ( $t == 1$ )	其它情况
0	(禁用)	
1	$[-4 : +3.5]$	$[-8 : +7.5]$
2	$[-8 : -7.5]$	$[-16 : +15.5]$
3	$[-16 : +15.5]$	$[-32 : +31.5]$
4	$[-32 : +31.5]$	$[-64 : +63.5]$
5	$[-64 : +63.5]$	$[-128 : +127.5]$
6	$[-128 : +127.5]$	$[-256 : +255.5]$
7	$[-256 : +255.5]$	$[-512 : +511.5]$
8	$[-512 : +511.5]$	$[-1024 : +1023.5]$
9	$[-1024 : +1023.5]$	$[-2048 : +2047.5]$
10~15	(保留)	

### I-7.6.3.3 修改运动向量预测器

当宏块中出现的所有运动向量都使用前面章节所定义的过程解码后, 有时需要修改一些运动向量预测器。这是因为在一些预测方式中, 使用少于最大允许数目的运动向量。剩余的预测器必须保持“灵敏”的值, 以供后面使用。

运动向量预测器应按表 I-7-9 和 I-7-10 所示进行修改。对于跳过的宏块的运动向量预测器的修改规则在 I-7.6.6 中阐述。

注意 根据图类型来实现运动向量预测器的优化修改(复位)是可能的。例如, 在 P-图中, 后向预测向量的预测器不被用到, 也就无保留的必要。

表 II -7-9 帧图中运动向量预测器的修改

frame_motion_type	macroblock_motion_		macroblock_intra	要修改的预测器
	forward	backward		
基于帧的*	-	-	1	$PMV[1][0][1:0] = PMV[0][0][1:0]$ <sup>o</sup>
基于帧的	1	1	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$ $PMV[1][1][1:0] = PMV[0][1][1:0]$
基于帧的	1	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$
基于帧的	0	1	0	$PMV[1][1][1:0] = PMV[0][1][1:0]$
基于帧的*	0	0	0	$PMV[r][s][t] = 0$ <sup>s</sup>
基于场的	1	1	0	(none)
基于场的	1	0	0	(none)
基于场的	0	1	0	(none)
双基	1	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$

注意  $PMV[r][s][1:0] = PMV[u][v][1:0]$  表示：  
 $PMV[r][s][1] = PMV[u][v][1]$  且  $PMV[r][s][0] = PMV[u][v][0]$   
 ◇ 如果 concealment\_motion\_vectors 为零，则  $PMV[r][s][t]$  被置为零（对所有 r,s,t）。  
 ≈ frame\_motion\_type 不在比特流中出现，但假定为是基于帧的。  
 § (仅出现于 P-图中)  $PMV[r][s][t]$  被置为零（对所有 r,s,t），见 II -7. 6. 3. 4。

表 II -7-10 场图中运动向量预测器的修改

field_motion_type	macroblock_motion_		macroblock_intra	要修改的预测器
	forward	backward		
基于场的*	-	-	1	$PMV[1][0][1:0] = PMV[0][0][1:0]$ <sup>o</sup>
基于场的	1	1	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$ $PMV[1][1][1:0] = PMV[0][1][1:0]$
基于场的	1	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$
基于场的	0	1	0	$PMV[1][1][1:0] = PMV[0][1][1:0]$
基于场的*	0	0	0	$PMV[r][s][t] = 0$ <sup>s</sup>
16×8MC	1	1	0	(none)
16×8MC	1	0	0	(none)
16×8MC	0	1	0	(none)
双基	1	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$

注意  $PMV[r][s][1:0] = PMV[u][v][1:0]$  表示：  
 $PMV[r][s][1] = PMV[u][v][1]$  且  $PMV[r][s][0] = PMV[u][v][0]$   
 ◇ 如果 concealment\_motion\_vectors 为零，则  $PMV[r][s][t]$  被置为零（对所有 r,s,t）。  
 ≈ 比特流中不出现 field\_motion\_type，但假定为基于场的。  
 § (仅出现于 P-图)  $PMV[r][s][t]$  被置为零（对所有 r,s,t），见 II -7. 6. 3. 4。

#### II -7. 6. 3. 4 复位运动向量预测器

在下列情况中,所有的运动向量预测器都将被复位为零。

- 在每个组块的起始处。
- 当没有隐藏运动向量的内部宏块被解码时。
- 在 P-图中,当 macroblock\_motion\_forward 为零的非内部宏块被解码时。
- 在 P-图中,当一个宏块被跳过时。

#### II -7. 6. 3. 5 P-图中的预测

在 P-图中,如果 macroblock\_motion\_forward 为零且 macroblock\_intra 为零,则没有运动向量为这个宏块编码,然而必须形成预测。如果这是发生在一个 P-场图中,则下列规则适用:

- 预测方式应为“基于场”的。
- (场)运动向量应为零(0;0)。
- 运动向量预测器应被复位为零。
- 预测应通过与正被预测的具有相同奇偶性的场作出。

如果这发生于一个 P-帧图中,则下列规则适用:

- 预测方式应为“基于帧”的。
- (帧)运动向量应为零(0;0)。
- 运动向量预测器应被复位为零。

当一个 P-场图作为一个帧的第二个场被使用,而这个帧的第一个场是 I-场图时,需要一系列语义上的限制。这保证了预测仅由 I-场图作出。这些限制是:

- 没有 macroblock\_motion\_forward 为零且 macroblock\_intra 为零的宏块被编码。
- 不应使用双基预测。
- 如果场预测中的 motion\_vertical\_field\_select 指出与正被预测的场具有相同地位,则不能使用这个场预测。
- 不应有跳过的宏块。

#### II -7. 6. 3. 6 双基附加算法

在双基预测中,场运动向量(vector'[0][0][1 : 0])应按已描述过的过程解码。这描述子用于形成预测的运动向量,它是通过与当前被预测场具有相同奇偶性的参考场(或是一个帧图内的参考场)来形成预测的。“奇偶性”这个词用来区分两个场。顶场的“奇偶性”为零,底场的“奇偶性”为 1。

为了给具有相对奇偶性的场形成运动向量(vector'[r][0][1 : 0]),改变现存的运动向量的比例来反映两场间不同的时域距离。对竖直分量进行修改(来反映顶场与底场的行间竖直位移),然后加上一个小差分运动向量,这一过程如图 II -7-12 所示,是帧图的情形。

dmvector[0]是差分运动向量的水平分量,dmvector[1]是竖直分量。差分运动向量的这两个分量将直接使用表 II -B-11 解码,并仅取值 -1, 0, +1。

m[parity\_ref][parity\_pred]是预测场和参考场间的场距,如表 II -7-11 所示。“parity\_ref”是计算新运动向量的参考场的奇偶性。“parity\_pred”是将被预测的场的奇偶性。

e[parity\_ref][parity\_pred]是反映底场和顶场行间竖直位移所需的调节器,如表

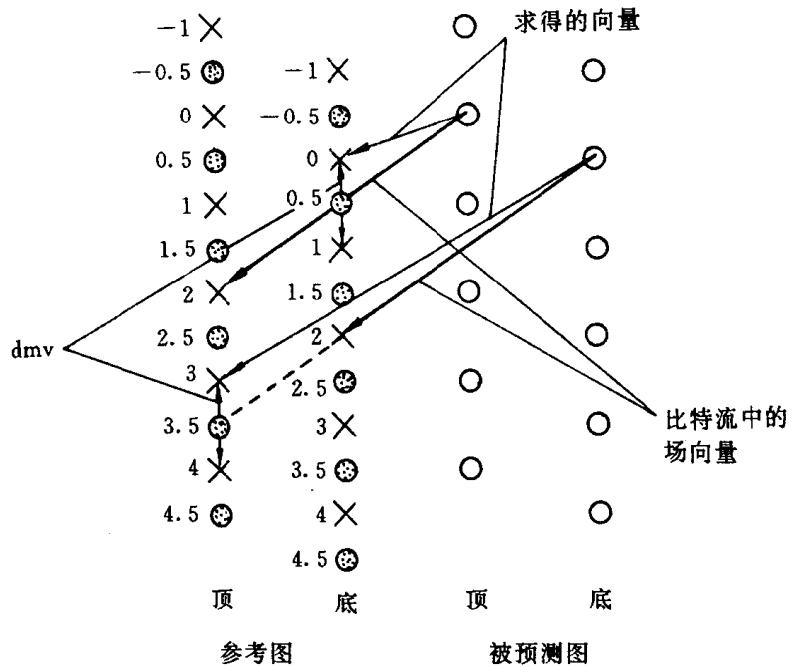


图 I - 7-12 双基预测的运动向量变化

I - 7-12 所定义。

表 II - 7-11  $m[\text{parity\_ref}][\text{parity\_pred}]$  的定义

picture_structure	top_field_first	$m[\text{parity\_ref}][\text{parity\_pred}]$	
		$m[1][0]$	$m[0][1]$
11 (帧)	1	1	3
11 (帧)	0	3	1
01 (顶场)	—	1	—
10 (底场)	—	—	1

表 II - 7-12  $e[\text{parity\_ref}][\text{parity\_pred}]$  的定义

parity_ref	parity_pred	$e[\text{parity\_ref}][\text{parity\_pred}]$
0	0	0
0	1	+1
1	0	-1
1	1	0

用于相对奇偶性预测的运动向量按下面计算：

```
vector'[r][0][0] = ((vector'[0][0][0] * m[parity_ref][parity_pred])//2) +
dmvector[0];
```

```
vector'[r][0][1] = ((vector'[0][0][1] * m[parity_ref][parity_pred]//2) + e[parity_ref][parity_pred] + dmvector[1]);
```

就场图而言,仅需一个这样的运动向量,这里  $r=2$ 。这样,(编码的)用于相同奇偶性的预测的运动向量是  $\text{vector}'[0][0][1 : 0]$ ,而用于相反奇偶性的预测的运动向量是  $\text{vector}'[2][0][1 : 0]$ 。

就帧图而言,需要两个这样的运动向量。两个场均使用编码运动向量( $\text{vector}'[0][0][1 : 0]$ ),用于相同奇偶性的预测。顶场用于相反奇偶性的预测的运动向量为  $\text{vector}'[2][0][1 : 0]$ ,底场用于相反奇偶性的预测的运动向量为  $\text{vector}'[3][0][1 : 0]$ 。

#### II - 7.6.3.7 色差分量的运动向量

前面章节所计算的运动向量指的是亮度分量,其中:

```
vector[r][s][t] = vector'[r][s][t] 对于所有 r,s,t
```

对于每个色差分量,运动向量比例系数变化如下:

4 : 2 : 0 运动向量的水平和竖直分量都除以 2:

```
vector[r][s][0] = vector'[r][s][0]/2;
```

```
vector[r][s][1] = vector'[r][s][1]/2;
```

4 : 2 : 2 运动向量水平分量除以 2,竖直分量不变:

```
vector[r][s][0] = vector'[r][s][0]/2;
```

```
vector[r][s][1] = vector'[r][s][1];
```

4 : 4 : 4 不修改运动向量:

```
vector[r][s][0] = vector'[r][s][0];
```

```
vector[r][s][1] = vector'[r][s][1];
```

#### II - 7.6.3.8 与预测有关的语义限制

作为对比特流的一个规定,对于一个解码器,我们仅需由参考场或参考帧中实际编码的组块中作出预测,这个规则甚至适用于跳过的宏块和 P-图中有零运动向量的宏块(如 II - 7.6.3.5 中所述)。

注意 如 II - 6.1.2 所述,一般来讲,并不需要组块覆盖整幅图。然而在许多规定框架的规定级别中,确实用到覆盖整幅图的“受限组块”。在这种情况下,语义规则可以更简单地阐述。“作为对比特流的限制,重构的运动向量不应指向编码图边界以外的样本”。

#### II - 7.6.3.9 隐藏的运动向量

隐藏的运动向量是内部宏块所带的运动向量,是为了当数据错误阻碍了系数解码时来掩盖错误。当(且仅当)`concealment_motion_vectors`(在 `picture_coding_extension()` 中)值为“1”时,在所有内部宏块中均会出现隐藏的运动向量。

在一般情况下,那些宏块不形成预测(像 `macroblock_intra=1` 时所作的那样)。本规范并未规定错误复原如何实现。然而作为一个建议,隐藏的运动向量应适合解码器使用,这个解码器通过形成预测来实现隐藏,就像下面这些控制预测结构的标志具有那些指明的值一样。

- 在一个场图中,`field_motion_type`=“基于场的预测”。
- 在一个帧图中,`frame_motion_type`=“基于帧的预测”。

注意 如果隐藏用于 I-图中,则解码器应对 P-图按同种方法形成预测。

隐藏运动向量是用于由于数据错误而导致信息丢失的情况。如果由于数据错误而需要进行错误复原,那么这个隐藏向量本身的丢失或被破坏也是很有可能的,由此在用到隐藏向量的宏块中对隐藏向量进行编码也就无必要了。因此,下面的语义规则是合适的。

- 对于除了那些最底一行的宏块以外的所有宏块,隐藏运动向量应适用于位于包含隐藏运动向量的宏块垂直下方的宏块。
- 当运动向量用于根据以前规则所识别的宏块中时,解码器必须认为这个运动向量可以指向参考帧或参考场中编码的组块以外的样本。
- 对于所有最底行的宏块,不使用重构的隐藏运动向量。因此运动向量(0;0)可以被用来减少不必要的开销。

#### II -7. 6. 4 形成预测

预测通过从参考帧或场读取预测样本而形成。一个给定样本通过读取参考场或帧中由运动向量指明的相应样本而被预测。

运动向量水平分量的正值表示预测是由位于当前被预测的样点右边的样点(在参考帧/场中)而作出的。

运动向量竖直分量的正值表示预测是由位于当前被预测的样点下方的样点(在参考帧/场中)而作出的。

所有运动向量被规定为具有半个样本的精确度。于是,如果运动向量的一个分量为奇数,则样本将从参考场/帧的实际样本的中间读取。这些“半样本”通过对实际样本的简单线性插值而计算得到。

对于基于场的预测,决定由两场中哪个场形成预测是必要的。对于双基的情况,我们规定运动向量是为两场而得出的,并且预测是由两场作出的。对于基于场的预测和 16×8MC,一个附加位 motion\_vertical\_field\_select 被编码来指明使用哪一场。

如果 motion\_vertical\_field\_select 为零,则预测由顶参考场得出。

如果 motion\_vertical\_field\_select 为 1,则预测由底参考场得出。

对于每个预测块,整数样本运动向量 int\_vec[t] 和半样本标志 half\_flag[t] 按以下方法形成:

```
for(t=0;t<2;t++) {  
    int_vec[t]=vector[r][s][t]DIV2;  
    if((vector[r][s][t]-(2 * int_vec[t]))!=0)  
        half_flag[t]=1;  
    else  
        half_flag[t]=0;  
}
```

对于预测块中的每个样本,样本的读取和半样本预测的应用如下:

```
if(! half_flag[0])&&(! half_flag[1])
```

```

pel_pred[y][x]=pel_ref[y+int_rec[1]][x+int_vec[0]],

if ((! half_flag[0])&&half_flag[1])
    pel_pred[y][x]=(pel_ref[y+int_vec[1]][x+int_vec[0]]+
                    pel_ref[y+int_vec[1]+1][x+int_vec[0]])//2;
if(half_flag[0]&&(! half_flag[1]))
    pel_pred[y][x]=(pel_ref[y+int_vec[1]][x+int_vec[0]]+
                    pel_ref[y+int_vec[1]+1][x+int_vec[0]+1])//2;
if(half_flag[0]&&half_flag[1])
    pel_pred[y][x]=(pel_ref[y+int_vec[1]][x+int_vec[0]]+
                    pel_ref[y+int_vec[1]+1][x+int_vec[0]+1]+
                    pel_ref[y+int_vec[1]+1][x+int_vec[0]])//4;

```

其中 pel\_pred[y][x]是正在形成的预测样点,pel\_ref[y][x]是参考场或帧中的样点。

### I -7.6.5 运动向量选择

表 I -7-13 示出了用于场图的预测方式,表 I -7-14 示出了用于帧图的预测。在每个表中,比特流中的运动向量以它们在比特流中出现的顺序列出。

表 I -7-13 场图中的预测和运动向量

field_motion_type	macroblock_motion--		macro-block--intra	运动向量	预测为准形成
	forward	backward			
基于场 <sup>#</sup>	-	-	1	vector'[0][0][1 : 0] <sup>o</sup>	无(运动向量 用于隐藏)
基于场	1	1	0	vector'[0][0][1 : 0] vector'[0][1][1 : 0]	全场,前向 全场,后向
基于场	1	0	0	vector'[0][0][1 : 0]	整场,前向
基于场	0	1	0	vector'[0][1][1 : 0]	整场,后向
基于场 <sup>#</sup>	0	0	0	vector'[0][0][1 : 0] <sup>* \$</sup>	整场,前向
16×8 MC	1	1	0	vector'[0][0][1 : 0] vector'[1][0][1 : 0] vector'[0][1][1 : 0] vector'[1][1][1 : 0]	上 16×8 场,前向 下 16×8 场,前向 上 16×8 场,后向 下 16×8 场,后向
16×8 MC	1	0	0	vector'[0][0][1 : 0] vector'[1][0][1 : 0]	上 16×8 场,前向 下 16×8 场,前向
16×8 MC	0	1	0	vector'[0][1][1 : 0] vector'[1][1][1 : 0]	上 16×8 场,后向 下 16×8 场,后向

续表

field-- motion-type	macroblock-motion--		macro-block-- intra	运动向量	预测为准形成
	forward	backward			
双基	1	0	0	vector'[0][0][1:0] vector'[2][0][1:0]*+	全场,来自同一奇偶性,前向 全场,来自相反奇偶性,前向
注意 运动向量以它们在比特流中出现的顺序列出。					
◇ 如果 concealment_motion_vectors 为 1, 则出现运动向量。					
† 在比特流中没有 field_motion_type, 但假定为基于场的。					
* 这些运动向量不在比特流中出现。					
+ 这些运动向量由 vectors'[0][0][1:0] 求出, 如 II-7.6.3.6 中所述					
§ 这个运动向量被置为(0,0), 如 II-7.6.3.5 所述。					

表 II-7-14 帧图中的预测和运动向量

frame-- motion-type	macroblock-motion--		macro-block-- intra	运动向量	预测为准形成
	forward	backward			
基于帧*	-	-	1	vector'[0][0][1:0]*	无(运动向量用于隐藏)
基于帧	1	1	0	vector'[0][0][1:0] vector'[0][1][1:0]	帧,前向 帧,后向
基于帧	1	0	0	vector'[0][0][1:0]	帧,前向
基于帧	0	1	0	vector'[0][1][1:0]	帧,后向
基于帧*	0	0	0	vector'[0][0][1:0]*	帧,前向
基于场	1	1	0	vector'[0][0][1:0] vector'[1][0][1:0] vector'[0][1][1:0] vector'[1][1][1:0]	顶场,前向 底场,前向 顶场,后向 底场,后向
基于场	1	0	0	vector'[0][0][1:0] vector'[1][0][1:0]	顶场,前向 底场,前向
基于场	0	1	0	vector'[0][1][1:0] vector'[1][1][1:0]	顶场,后向 底场,后向

续表

frame_motion_type	macroblock_motion--		macro-block-intra	运动向量	预测为准形成
	forward	backward			
双基	1	0	0	vector'[0][0][1 : 0] vector'[0][0][1 : 0] vector'[2][0][1 : 0]* <sup>†</sup> vector'[3][0][1 : 0]* <sup>†</sup>	顶场,来自同一奇偶性,前向 底场,来自同一奇偶性,前向 顶场,来自相反奇偶性,前向 底场,来自相反奇偶性,前向

注意 运动向量以它们在比特流中出现的顺序列出。

◇ 如果 concealment\_motion\_vectors 为 1, 则出现运动向量。  
 ≈ 在比特流中没有 frame\_motion\_type, 但假定为基于帧的。  
 \* 这些运动向量不在比特流中出现。  
 † 这些运动向量由 vector'[0][0][1 : 0] 中得出, 如 I-7.6.3.6 中所述。  
 § 这个运动向量被置为(0,0), 如 I-7.6.3.5 中所述。

## II-7.6.6 跳过的宏块

在跳过的宏块中(macroblock\_address\_increment 比 1 大的地方), 解码器既没有 DCT 系数信息, 也没有运动向量信息。解码器为这样的宏块形成可作为最终解码样本值的预测。

P-图和 B-图对于跳过宏块的处理是不同的。此外, 场图与帧图之间的处理也不同。

I-图中应没有跳过的宏块, 除了当:picture\_spatial\_scalable\_extension() 跟在当前图的 picture\_header() 之后; 或比特流中出现 sequence\_scalable\_extension() 且 scalable\_mode = “SNR 分级”。

### II-7.6.6.1 P 场图

- 预测应被作出, 好像 field\_motion\_type 为“基于场的”。
- 预测应从与当前被预测场具有相同奇偶性的场作出。
- 运动向量预测器被复位为零。
- 运动向量应为零。

### II-7.6.6.2 P 帧图

- 预测应被作出, 好像 field\_motion\_type 为“基于帧的”。
- 预测器应被复位为零。
- 运动向量应为零。

### II-7.6.6.3 B 场图

- 预测应被作出, 好像 field\_motion\_type 为“基于场的”。
- 预测应由与当前被预测的场具有相同奇偶性的场作出。
- 预测方向, 前向/后向/双向, 应与以前的宏块相同。
- 运动向量预测器不受影响。
- 运动向量由合适的运动向量预测器得出。

颜色分量的运动向量的变化应如 II -7.6.3.7 中所述实现。

#### II -7.6.6.4 B 帧图

- 预测应被作出,好像 field\_motion\_type 为“基于帧”的。
- 预测方向,前向/后向/双向,应与以前的宏块相同。
- 运动向量预测器不受影响。
- 运动向量直接由合适的运动向量预测器中得出。颜色分量的运动向量的变化应如 II -7.6.3.7 所述实现。

### II -7.6.7 组合预测

最后的步骤是将各种预测组合到一块,以形成最终的预测块。

将这些数据按场或帧组织到块中,使其可以直接加到解码系数上也是必要的。

变换数据是以场组织的还是以帧组织的由 dct\_type 规定。

#### II -7.6.7.1 简单帧预测

对于简单的帧预测来说,所需的下一步处理就是将 B-图中的前向和后向预测求平均。如果 pel\_pred\_forward[y][x] 是前向预测样本,且 pel\_pred\_backward[y][x] 是相应的后向预测,则最终的预测样本应如下形成:

```
pel_pred[y][x]=(pel_pred_forward[y][x]+pel_pred_backward[y][x])//2
```

对于 4 : 2 : 0, 4 : 2 : 2 和 4 : 4 : 4 格式的色差分量预测应有相应的 8×8, 8×16 和 16×16 的尺寸。

#### II -7.6.7.2 简单场预测

对于简单的场预测来说(即既非 16×8 也非双基),所需的下一步处理仅是将 B-图中的前向和后向预测求平均。这将按上节中“帧预测”那样实现。

对于 4 : 2 : 0, 4 : 2 : 2 和 4 : 4 : 4 格式的色差分量的预测,应有相应的 8×4, 8×8 和 16×8 的尺寸。

#### II -7.6.7.3 双基

在双基方式中,对每个场都形成两个预测,对于 B-图中的后向和前向预测采取相似的方法。如果 pel\_pred\_same\_parity[y][x] 是由相同奇偶性场中得来的预测样本而 pel\_pred\_opposite\_parity[y][x] 是由相反奇偶性场中得来的样本,那么最终预测样本按如下形成:

```
pel_pred[y][x]=(pel_pred_same_parity[y][x]+pel_pred_opposite_parity[y][x])//2;
```

对于帧图中的双基预测,对于 4 : 2 : 0, 4 : 2 : 2 和 4 : 4 : 4 格式的色差分量的预测应有相应的 8×4, 8×8 和 16×8 的尺寸。

对于场图中的双基预测,对于 4 : 2 : 0, 4 : 2 : 2 和 4 : 4 : 4 格式的色差分量的预测应有相应的 8×8, 8×16 和 16×16 的尺寸。

### II -7.6.8 预测和系数数据求和

预测块形成了,并按与变换数据块所用的场/帧结构相匹配的形式重组为预测样点

$p[y][x]$ 的块。

变换数据  $f[y][x]$  加到预测数据上, 饱和化后形成最终解码样本  $d[y][x]$ , 如下所示:

```
for(y=0;y<8;y++) {
    for(x=0;x<8;x++) {
        d[y][x]=f[y][x]+p[y][x];
        if(d[y][x]<0)d[y][x]=0;
        if(d[y][x]>255) d[y][x]=255;
    }
}
```

## II - 7.7 空域分级

这一节阐述空域分级扩展所需的附加解码过程。

低层和增强层都将使用“受限组块结构”(组块间没有空隙)。

图 II - 7-13 是带空域分级的解码过程图示。为清楚起见, 此图已被简化。

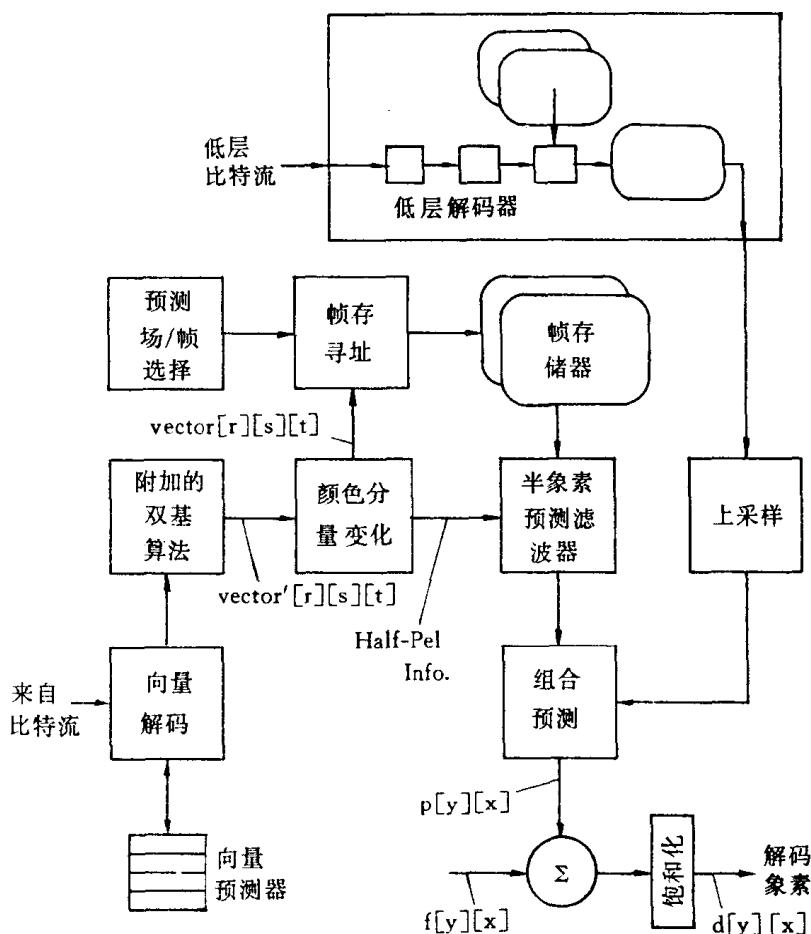


图 II - 7-13 空域分级的简化的运动补偿过程

## II -7.7.1 更高层的语法结构

一般来讲,空域分级体系的基层能适合任何编码标准,包括 ITU-T Rec. H. 261, ISO/IEC 11172-2 和 ITU-T Rec. xxxx | ISO/IEC 13818-2。然而要注意,在本规范中,我们仅考虑基层与本规范或 ISO/IEC 11172-2 相一致的空域分级体系的解码能力。

如果低层和增强层均是隔行的,则由于层间的“松散耦合”,增强层中仅需要一个语法限制,在这种情况下,picture\_structure 具有与低层中用于预测的参考帧相同的值。如何分辨这个参考帧,参见 II -7.7.3.1。

## II -7.7.2 增强层中的预测

运动补偿的“时域”预测由以前增强层中的解码图作出,如 II -7.6 所述。此外,采用低层解码帧的上采样方案,形成“空域”预测。如 II -7.7.3 所述,可以单独选择或将这些预测组合起来形成实际的预测。

通常,对每个宏块最多可形成四个独立的预测,它们组合在一起形成最终的预测宏块  $p[y][x]$ 。

如果一个宏块未被编码,既可能是因为整个宏块是被跳过的,也可能是因为这个宏块中没有系数数据。在这种情况下, $f[y][x]$  为零,解码的样本即为预测  $p[y][x]$ 。

## II -7.7.3 空域预测的构成

形成空域预测需要分辨正确的参考帧,并对空域再采样过程进行定义,这由下面的章节完成。

再采样过程是为一个整帧而定义的,然而,对于宏块的解码,仅需要上采样帧中与这个宏块相应的  $16 \times 16$  区域。

### II -7.7.3.1 参考帧的选择

空域预测通过由 lower\_layer\_temporal\_reference 指出的低层的解码帧作出。然而,如果基层和增强层比特流嵌入到一个 ITU-T Rec. xxxx | ISO/IEC 13818-1 多路系统,则 PES 头中的解码时间标签 DTS(decoding time stamps)给出的时间信息将使这个信息无效。

注意 如果 group\_of\_pictures() 经常出现在低层比特流中,那么低层中的时域参考可能发生混淆(因为在 group\_of\_pictures\_header() 之后 temporal\_reference 被复位)。

作出空域预测的图应该是如 DTS 值所指出的重合的或最近被解码的低层图。此外,还应注意,仅当低层中作出预测的帧在显示时间上与增强层中的被预测帧重叠(或相近),空域分级才会有效地使用。

### II -7.7.3.2 再采样过程

空域预测通过对低层帧再采样使其与增强层具有相同的坐标单位来作出,即使低层的隔行帧实际上是作为一个场结构帧被编码,这个坐标单位也以帧坐标的形式定义。

这个再采样过程如图 II -7-14 所示。

空域预测仅由增强层中的完全处于顶端采样低层重构帧内部的宏块作出。

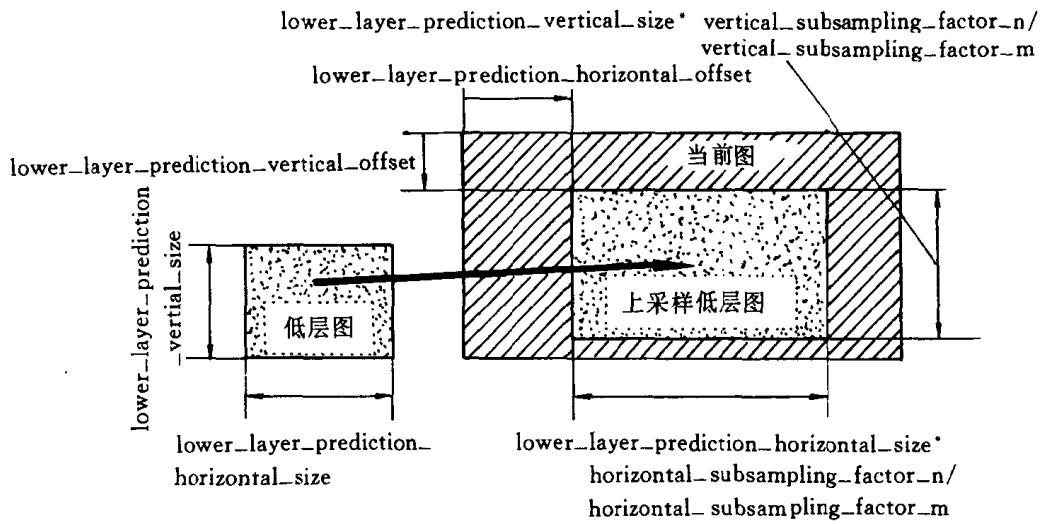


图 I - 7-14 通过对低层图插值得到的“空域”预测结构

顶端采样过程取决于由 `lower_layer_progressive_frame` 指明的低层重构帧是隔行还是逐行的, 和由 `progressive_frame` 指明的增强层帧是隔行还是逐行的。

当 `lower_layer_progressive_frame` 为“1”, 低层重构帧(改名为 `input_prog_field`)按 I - 7. 7. 3. 5 所述被垂直再采样。如果 `progressive_frame` 为“1”, 则结果帧被认为是逐行帧, 反之认为是隔行帧。结果帧按 I - 7. 7. 3. 6 所述被水平再采样。`lower_layer_deinterlaced_field_select` 应取值“1”。

当 `lower_layer_progressive_frame` 为“0”, 且 `progressive_frame` 为“0”, 每个低层重构场将按 I - 7. 7. 3. 4 所述而被反隔行, 以产生一个逐行场(`input_prog_field`)。参考 `lower_layer_vertical_offset`, 这个场按 I - 7. 7. 3. 5 所述被垂直再采样。结果场按 I - 7. 7. 3. 6 所述被水平再采样。最后, 再次参照 `lower_layer_vertical_offset`, 结果场被子采样, 来产生一个隔行场。`lower_layer_deinterlaced_field_select` 应取值“1”。

当 `lower_layer_progressive_frame` 为“0”且 `progressive_frame` 为“1”时, 每个低层重构场按 I - 7. 7. 3. 4 所述被反隔行, 以产生逐行场(`input_prog_field`)。仅需要其中一个场。如果 `lower_layer_deinterlaced_field_select` 为“0”, 则使用顶场, 否则使用底场。用到的这个场按 I - 7. 7. 3. 5 所述被竖直再采样。结果场按 I - 7. 7. 3. 6 中所述被水平再采样。

对于隔行帧, 如果当前帧(隐含低层)以场图编码, 则在这个场中作反隔行处理, 如 I - 7. 7. 3. 5 所述。

如果增强层中的亮度分量在那个维度中被子采样, 来匹配两层间的亮度样本, 则低层的位移应被限制为偶数值。

顶端采样过程总结为表 I - 7-15。

表 II-7-15 顶端采样过程

lower_layer_deinterlaced_field_select	lower_layer_progressive_frame	progressive_frame	应用反隔行过程	预测所用实体
0	0	1	yes	顶场
1	0	1	yes	底场
1	1	1	no	帧
1	1	0	no	帧
1	0	0	yes	两场

## II-7.7.3.3 颜色分量处理

根据不同的亮度和色差分量的采样坐标单位, II-7.7.3.4 和 II-7.7.3.6 中使用的一些变量对于亮度和色差再采样取不同的值。此外, 我们允许低层和增强层的色差格式彼此不同。

表 II-7-16 定义了用于 II-7.7.3.4 到 7.7.3.6 的变量的值。

表 II-7-16 用于 II-7.7.3.3 到 II-7.7.3.5 的局部变量

变量	亮度过程的值	色差过程的值
ll_h_size	lower_layer_prediction_horizontal_size	lower_layer_prediction_horizontal_size /chroma_ratio_horizontal
ll_v_size	lower_layer_prediction_vertical_size	lower_layer_prediction_vertical_size /chroma_ratio_vertical
ll_h_offset	lower_layer_horizontal_offset	lower_layer_horizontal_offset /chroma_ratio_horizontal
ll_v_offset	lower_layer_vertical_offset	lower_layer_vertical_offset /chroma_ratio_vertical
h_subs_m	horizontal_subsampling_factor_m	horizontal_subsampling_factor_m * format_ratio_horizontal
h_subs_n	horizontal_subsampling_factor_n	horizontal_subsampling_factor_n
v_subs_m	vertical_subsampling_factor_m	vertical_subsampling_factor_m * format_ratio_vertical
v_subs_n	vertical_subsampling_factor_n	vertical_subsampling_factor_n

表 II-7-17 和 II-7-18 给出了附加的定义。

表 II-7-17 亮度子采样比

色差格式 低层	chroma_ratio_	chroma_ratio
	horizontal	vertical
4 : 2 : 0	2	2
4 : 2 : 2	2	1
4 : 4 : 4	1	1

表 II-7-18 色差格式比

色差格式 低层	色差格式 增强层	format_ratio_	format_ratio_
		horizontal	vertical
4 : 2 : 0	4 : 2 : 0	1	1
4 : 2 : 0	4 : 2 : 2	1	2
4 : 2 : 0	4 : 4 : 4	2	2
4 : 2 : 2	4 : 2 : 2	1	1
4 : 2 : 2	4 : 4 : 4	2	1
4 : 4 : 4	4 : 4 : 4	1	1

#### II-7.7.3.4 反隔行

首先,每个低层场被填充上“零”来形成逐行坐标,具有与低层场率相同的帧率,并且和低层帧有相同的行数和每行样点数。表 II-7-19 规定了下面要用到的滤波器。如果 picture\_structure == “Frame\_Picture”,则亮度分量使用相关双场口径滤波器滤波,否则使用单场口径滤波器滤波。色差分量使用单场口径滤波器滤波。

这个表的“时域”和“垂直”两列指明了样本的相应的空域和时域坐标,另两列中定义的滤波器分支用于这些样本。通过将这些倍增的系数加在一起形成中间和。

表 II-7-19 反隔行滤波器

时域	垂直	双场口径		单场口径
		首场滤波器	次场滤波器	滤波器(双场)
-1	-2	0	-1	0
-1	0	0	2	0
-1	2	0	-1	0
0	-1	8	8	8
0	0	16	16	16
0	1	8	8	8
1	-2	-1	0	0
1	0	2	0	0
1	+2	-1	0	0

然后滤波器(和)的输出按下面公式改变比例:

$$\text{prog\_field}[y][x] = \text{sum} // 16$$

并且被饱和化到[0 : 255]的范围。

滤波器口径可以延伸到编码图尺寸之外。此时,这些活动图象之外的行的样本应取该场内最近的相邻样本的值(下面或上面),如下面所述。

对所有样本[y][x]:

```

if (y<0&&(y&1==1))
    y=1
if (y<0&&(y&1==0))
    y=0
if (y>=ll_v_size &&((y-ll_v_size)&1==1))
    y=ll_v_size-1
if(y)==ll_v_size &&
((y-ll_v_size)&1==0))
    y=ll_v_size-2

```

#### II -7.7.3.5 竖直再采样

被垂直再采样的帧(input\_prog\_field)被再采样为增强层的垂直采样坐标格式,其中使用了按下面公式进行的样本间的线性插值。mid\_field 为结果场。

$$\begin{aligned} \text{mid\_field}[y_h+ll_v\_offset][x] = & (16-\text{phase}) * \text{input\_prog\_field} \\ & [y1][x] + \text{phase} * \text{input\_prog\_field} \\ & [y2][x] \end{aligned}$$

其中  $y_h+ll_v\_offset=mid\_field$  中的输出样本坐标

$$\begin{aligned} y1 &= (y_h * v\_subs\_m) / v\_subs\_n \\ y2 &= y1 + 1 \quad \text{如果 } y1 < ll_v_size - 1 \\ &= y1 \quad \text{其它} \\ \text{phase} &= (16 * ((y_h * v\_subs\_m) \% v\_subs\_n)) // v\_subs\_n \end{aligned}$$

位于上采样所需的低层图之外的样本通过低层图的边界扩展而得到。

注意 对于 phase 的计算是基于假定在增强层中  $y_h=0$  处样本位置与低层中第一个样本位置一致。如果 chroma\_format == 4 : 2 : 0, 则被公认为是对色差分量的一个近似。

#### II -7.7.3.6 水平再采样

被水平再采样的帧 mid\_field 按增强层水平采样格式被再采样,其中使用了按下面公式进行的样本间的线性插值。output\_field 是结果场。

$$\text{output\_field}[y][x_h+ll_h\_offset] = ((16-\text{phase}) * \text{mid\_field}[y][x1] + \text{phase} * \text{mid\_field}[y][x2]) // 256$$

其中  $x_h+ll_h\_offset=output\_field$  中的输出样点坐标

$$\begin{aligned} x1 &= (x_h * h\_subs\_m) / h\_subs\_n \\ x2 &= \begin{cases} x1 + 1 & \text{如果 } x1 < ll_h\_size - 1 \\ x1 & \text{其它} \end{cases} \\ \text{phase} &= (16 * ((x_h * h\_subs\_m) \% h\_subs\_n)) // h\_subs\_n \end{aligned}$$

位于上采样所需的低层图以外的样本通过低层图的边界扩展获得。

## II -7.7.4 空域和时域预测的选择和组合

空域和时域预测可以被选择和组合来形成实际的预测。macroblock\_type(表 I -B-5, I -B-6 和 I -B-7)通过 spatial\_temporal\_weight\_class(可取值 0,1,2,3,4)指明, 预测是仅时域的, 仅空域的, 还是时域和空域预测的加权组合。spatial\_temporal\_weight\_class 的完整描述在 II -7.7.5 中给出。

在内部图中, 如果 spatial\_temporal\_weight\_class 为 0, 则实现通常的内部编码, 否则进行仅空域预测。在预测插值图中, 如果 spatial\_temporal\_weight\_class 为 0, 则预测为仅时域的; 如果 spatial\_temporal\_weight\_class 为 4, 则预测为仅空域的, 否则使用一个或一对预测加权来组合空域和时域预测。

可能的 spatial\_temporal\_weights 由图象分级扩展中选定的加权表给出。根据当前低层是逐行还是隔行, 最多可有四个不同的加权表以供使用, 在 macroblock\_modes() 中, 一个两位的编码 spatial\_temporal\_weight\_code, 被用来描述每个场或帧的预测, 如表 II -7-20 所示。在这个表中, spatial\_integer\_weight 指明了可以和双基预测一起使用的那些 spatial\_temporal\_weight\_codes(见表 II -7-22, II -7-23)。

**表 II -7-20 针对 spatial\_temporal\_weight\_code\_table\_index 和 spatial\_temporal\_weight\_codes 的 spatial\_temporal\_weights 和 spatial\_temporal\_weight\_classes**

spatial_temporal_weight_code_table_index	spatial_temporal_weight_code	spatial_temporal_weight(s)	spatial_temporal_weight class	spatial_temporal_integer_weight
00*	—	(0.5)	1	0
01	00	(0;1)	3	1
	01	(0;0.5)	1	0
	10	(0.5;1)	3	0
	11	(0.5;0.5)	1	0
10	00	(1;0)	2	1
	01	(0.5;0)	1	0
	10	(1;0.5)	2	0
	11	(0.5;0.5)	1	0
11	00	(1;0)	2	1
	01	(1;0.5)	2	0
	10	(0.5;1)	3	0
	11	(0.5;0.5)	1	0

\* 由于 spatial\_temporal\_weight\_code\_table\_index = 0, 所以不传输 spatial\_temporal\_weight\_code。

注意 仅空域的预测(`weight_class==4`)由 `macroblock_type` 的不同值标志出。

当 `spatial_temporal_weight` 组合以(a;b)形式给出时,“a”给出了由空域预测得出的针对顶场的预测所占比例,而“b”给出了由空域预测得出的针对底场的预测所占的比例。

当 `spatial_temporal_weight` 以(a)的形式给出时,“a”给出这幅图的预测所占比例,这幅图由空域预测得出。

对预测器的精确计算方法如下:

如果用 `pel_pred_temp[y][x]` 表示时域预测(在增强层中形成)。如 I-7.6 中对 `pel_pred[y][x]` 的定义,用 `pel_pred_lower[y][x]` 来表示从低层形成的预测,那么;如果 `spatial_temporal_weight` 为“零”,则没有预测由低层中作出。因此

$$\text{pel\_pred}[y][x] = \text{pel\_pred\_temp}[y][x];$$

如果 `spatial_temporal_weight` 为“1”,则没有预测由增强层中作出。因此

$$\text{pel\_pred}[y][x] = \text{pel\_pred\_lower}[y][x];$$

如果权为 0.5,则预测为时域和空域预测的平均。因此

$$\text{pel\_pred}[y][x] = (\text{pel\_pred\_temp}[y][x] + \text{pel\_pred\_lower}[y][x]) // 2$$

当 `chroma_format == 4:2:0`,且 `chroma_420_type == 0`,则色差按逐行处理,即,第一个加权用于顶场色差行,第二个加权用于底场色差行。

我们指定不同的加权码表用于下面的各种环境(所有其它允许值在括号中给出)。

表 II-7-21 `spatial_temporal_weight_code_table_index` 的指定(允许)值

低层格式	增强层格式	<code>spatial_temporal_weight_code_table_index</code>
逐行或隔行	逐行	00
逐行,与增强层顶场一致	隔行	10(00;01;11)
逐行,与增强层底场一致	隔行	01(00;10;11)
隔行( <code>picture_structure</code> =帧图)	隔行	00 或 11(01;10)
隔行( <code>picture_structure!</code> =帧图)	隔行	00

## II-7.7.5 修改运动向量预测器和运动向量选择

在用到场预测的帧图中,存在着其中某个场使用仅空域预测的可能性。此时,这个场的比特流中不出现运动向量。如果两场都使用仅空域预测,则 `macroblock_type` 表示那个宏块所在的比特流中没有运动向量。

“类”也可以指出编码比特流中出现的运动向量的数目和运动向量怎样按表 I-7-22 和表 I-7-23 中所定义的进行修改。“类”以下面的方式定义:

类 0 指明仅时域预测。

类1指明没有场使用仅空域预测。

类2指明顶场是仅空域预测。

类3指明底场是仅空域预测。

类4指明仅空域预测。

表 II -7-22 场图中运动向量预测器的修改

frame_motion_type					
	macroblock_motion_forward				
	macroblock_motion_backward				
	macroblock_intra				
	spatial_temporal_weight_class				
	Predictors to update				
基于场*	-	-	1	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$ <sup>o</sup>
基于场	1	1	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$ $PMV[1][1][1:0] = PMV[0][1][1:0]$
基于场	1	0	0	0,1	$PMV[1][0][1:0] = PMV[0][0][1:0]$
	0	1	0	0,1	$PMV[1][1][1:0] = PMV[0][1][1:0]$
基于场*	0	0	0	0,1,4	$PMV[r][s][t] = 0$ <sup>s</sup>
16×8 MC	1	1	0	0	(none)
16×8 MC	1	0	0	0,1	(none)
16×8 MC	0	1	0	0,1	(none)
双基	1	0	0	0	$PMV[1][0][1:0] = PMV[0][0][1:0]$

注意  $PMV[r][s][1:0] = PMV[u][v][1:0]$  表示：  
 $PMV[r][s][1] = PMV[u][v][1]$  且  $PMV[r][s][0] = PMV[u][v][0]$   
◇ 如果 concealment\_motion\_vectors 为零，则  $PMV[r][s][t]$  被置为零（对所有 r,s 和 t）。  
≠ 比特流中不出现 field\_motion\_type，但假定为基于场的。  
§  $PMV[r][s][t]$  被置为零（对所有 r,s 和 t）。见 I -7. 6. 3. 4。

表 I -7-23 帧图中运动预测器的修改

frame_motion_type					
					macroblock_motion_forward
					macroblock_motion_backward
					macroblock_intra
					spatial_temporal_weight_class
					Predictors to update
基于帧*	-	-	1	0	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$ °
基于帧	1	1	0	0	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$ $PMV[1][1][1 : 0] = PMV[0][1][1 : 0]$
基于帧	1	0	0	0,1,2,3	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$
基于帧	0	1	0	0,1,2,3	$PMV[1][1][1 : 0] = PMV[0][1][1 : 0]$
基于帧*	0	0	0	0,1,2,3,4	$PMV[r][s][t] = 0$ §
基于场	1	1	0	0	(none)
基于场	1	0	0	0,1	(none)
基于场	1	0	0	2	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$
基于场	1	0	0	3	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$
基于场	0	1	0	0,1	(none)
基于场	0	1	0	2	$PMV[1][1][1 : 0] = PMV[0][1][1 : 0]$
基于场	0	1	0	3	$PMV[1][1][1 : 0] = PMV[0][1][1 : 0]$
双基®	1	0	0	0,2,3	$PMV[1][0][1 : 0] = PMV[0][0][1 : 0]$

注意  $PMV[r][s][1 : 0] = PMV[u][v][1 : 0]$  表示:  
 $PMV[r][s][1] = PMV[u][v][1]$  且  $PMV[r][s][0] = PMV[u][v][0]$

◇ 如果 concealment\_motion\_vectors 为零, 则  $PMV[r][s][t]$  置为零(所有 r,s,t)。  
 ± 比特流中不出现 frame\_motion\_type, 但假定为基于帧的。  
 §  $PMV[r][s][t]$  置为零(对所有 r,s,t)。见 I -7.6.3.4。  
 @ 当 spatial\_temporal\_integer\_weight = “0”时, 不能用双基。

### I -7.7.5.1 运动向量预测器复位

除了 I -7.6.3.4 中指明的那些情况, 运动向量预测器将在下列情况下被复位:

- 在 P-图中, 当宏块仅采取空域预测时  
(spatial\_temporal\_weight\_class == 4)
- 在一个 B-图中, 当宏块仅采取空域预测时  
(spatial\_temporal\_weight\_class == 4)

注意 如果在一个帧图中, 当用到 field-based 预测时, 且 spatial\_temporal\_weight\_class == 2, 则传送的向量用于底场(见表 I -7-25)。然而这个向量 [0][s][1 : 0] 由  $PMV[0][s][1 : 0]$  中预测出。然后  $PMV[1][s][1 : 0]$  按表 I -7-23 所示被修改。

表 II -7-24 场图中的预测和运动向量

field_motion_type						
macroblock_motion_forward						
macroblock_motion_backward						
macroblock_intra						
spatial_temporal_weight_class		Motion vector		形成预测的方式		
基于场*	-	-	1	0	vector'[0][0][1 : 0]◊	无(运动向量用于隐藏)
基于场	1	1	0	0	vector'[0][0][1 : 0]	整场,前向
					vector'[0][1][1 : 0]	整场,后向
基于场	1	0	0	0,1	vector'[0][0][1 : 0]	整场,前向
基于场	0	1	0	0,1	vector'[0][1][1 : 0]	整场,后向
基于场	0	0	0	0,1,4	vector'[0][0][1 : 0]*§	整场,前向
16×8 MC	1	1	0	0	vector'[0][0][1 : 0]	上 16×8 场,前向
					vector'[1][0][1 : 0]	下 16×8 场,前向
					vector'[0][1][1 : 0]	上 16×8 场,后向
					vector'[1][1][1 : 0]	下 16×8 场,后向
16×8 MC	1	0	0	0,1	vector'[0][0][1 : 0]	上 16×8 场,前向
					vector'[1][0][1 : 0]	下 16×8 场,前向
16×8 MC	0	1	0	0,1	vector'[0][1][1 : 0]	上 16×8 场,后向
					vector'[1][1][1 : 0]	下 16×8 场,后向
双基	1	0	0	0	vector'[0][0][1 : 0]	整场,相同奇偶性,前向
					vector'[2][0][1 : 0]*†	整场,相反奇偶性,前向

注意 运动向量以它们在比特流中出现的顺序列出。

- ◊ 如果 concealment\_motion\_vectors 为“-”,则出现这个运动向量。
- \* 比特流中没有 field\_motion\_type,但假定为基于场的。
- \* 这些运动向量不在比特流中出现。
- † 这些运动向量由 vector'[0][0][1 : 0] 中得到,如 II -7.6.3.6 所述。
- § 运动向量取值为(0,0),如 II -7.6.3.5 所述。

表 II -7-25 帧图中的预测和运动向量

frame_motion_type						
macroblock_motion_forward						
macroblock_motion_backward						
macroblock_intra						
spatial_temporal_weight_class				Motion vector	形成预测的方式	
基于帧 <sup>#</sup>	-	-	1	0	vector'[0][0][1 : 0] <sup>*</sup>	无(运动向量用于隐藏)
基于帧	1	1	0	0	vector'[0][0][1 : 0]	帧,前向
					vector'[0][1][1 : 0]	帧,后向
基于帧	1	0	0	0,1,2,3	vector'[0][0][1 : 0]	帧,前向
基于帧	0	1	0	0,1,2,3	vector'[0][1][1 : 0]	帧,后向
基于帧 <sup>#</sup>	0	0	0	0,1,2,3,4	vector'[0][0][1 : 0] <sup>*§</sup>	帧,前向
基于场	1	1	0	0	vector'[0][0][1 : 0]	顶场,前向
					vector'[1][0][1 : 0]	底场,前向
					vector'[0][1][1 : 0]	顶场,后向
					vector'[1][1][1 : 0]	底场,后向
基于场	1	0	0	0,1	vector'[0][0][1 : 0]	顶场,前向
					vector'[1][0][1 : 0]	底场,前向
基于场	1	0	0	2		顶场,空域
					vector'[0][0][1 : 0]	底场,前向
基于场	1	0	0	3	vector'[0][0][1 : 0]	顶场,前向
						底场,空域
基于场	1	1	0	0,1	vector'[0][1][1 : 0]	顶场,后向
					vector'[1][1][1 : 0]	底场,后向
基于场	0	1	0	2		顶场,空域
					vector'[0][1][1 : 0]	底场,后向
基于场	0	1	0	3	vector'[0][1][1 : 0]	顶场,后向
						底场,空域
双基 <sup>@</sup>	1	0	0	0,2,3	vector'[0][0][1 : 0]	顶场,相同奇偶性,前向
					vector'[0][0][1 : 0] <sup>*</sup>	底场,相同奇偶性,前向
					vector'[2][0][1 : 0] <sup>*+</sup>	顶场,相反奇偶性,前向
					vector'[3][0][1 : 0] <sup>*+</sup>	底场,相反奇偶性,前向

注意 运动向量以它们在比特流中出现的顺序列出。

- ◇ 如果 concealment\_motion\_vectors 为“-”, 则出现这个运动向量。
- ≠ 比特流中不出现 frame\_motion\_type, 但假定为基于帧的。
- \* 这些运动向量不在比特流中出现。
- † 这些运动向量由 vector'[0][0][1 : 0] 中得到, 如 II -7.6.3.6 所述。
- § 这个运动向量取值为(0,0), 如 II -7.6.3.5 所述。
- @ 当 spatial\_temporal\_integer\_weight = “0”时, 不能使用双基。

## II -7.7.6 跳过的宏块

在所有情况中,跳过的宏块仅是预测的结果,且所有的 DCT 系数都被认为是零。

如果存在 sequence\_scalable\_extension, 且 scalable\_mod = “空域分级”, 则除了 I-7.6.6 中给出的以外,下面的规则也适用。

在 I-图中,允许有跳过的宏块。它们被定义为仅空域预测。

在 P-图和 B-图中,跳过的宏块是仅时域预测的。

在 B-图中,跳过的宏块不会跟在一个仅空域预测的宏块之后。

## II -7.7.7 低层中的 VBV 缓冲区下溢

对于空域分级来说,低层中的 VBV 缓冲区下溢可能会导致麻烦。这是因为帧到底是在哪个缓冲区中被特殊的解码器重复可能不确定。

所以在低层比特流中,VBV 缓冲区下溢不应发生。

## II -7.8 SNR 分级

本节描述 SNR 分级扩展所需的附加解码过程。SNR 分级定义了对流中另一层中编码的 DCT 系数的改进方法。如图 II-7-15 所述,由两个比特流来的数据,在反量化过程后,通过相加而组合在一起。两个层的解码过程是彼此独立的,直到数据被组合起来为止。

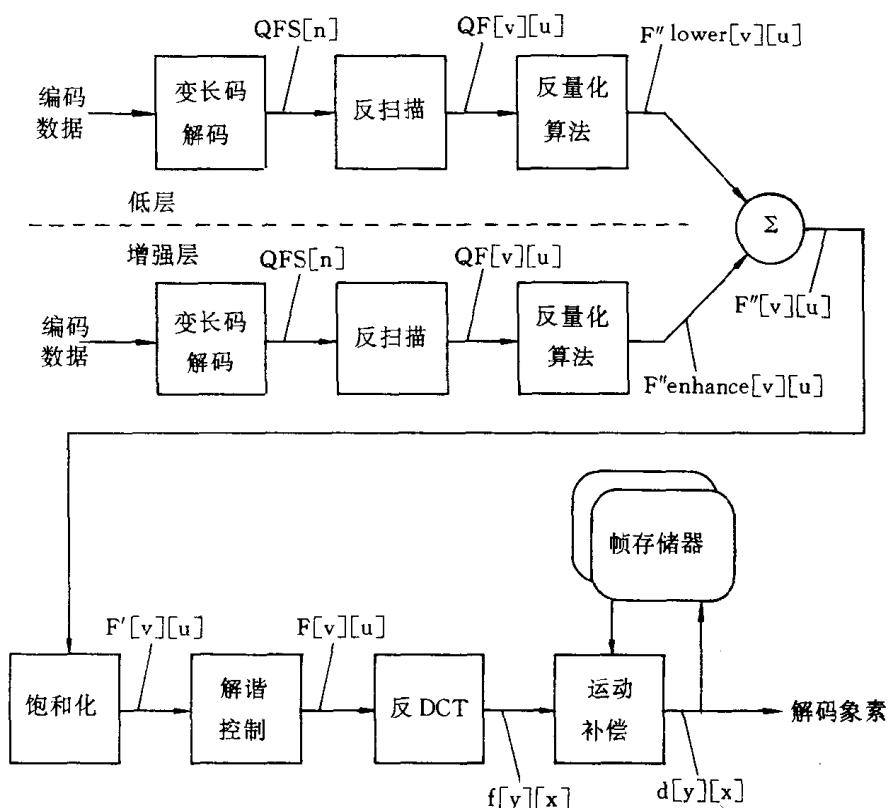


图 II-7-15 SNR 分级解码过程描述

I-7.8.1 规定了如何在流的多层集合中辨别这些比特流,可以按以下方法分类。

从首比特流中得到的“低层”,本身既可能是非分级的,也可能是需要用到空域或时域分级解码过程的。

从第二个比特流中得到的“增强层”,主要包含编码 DCT 系数和一些简单辅助操作。这个层的解码过程和两层的组合在本节中定义。

注意 所有与预测有关的信息仅包含在低层中。因此如果不同时对低层解码,是不可能重构增强层的。

此外,如 I-7.6, I-7.7 和 I-7.9 中所描述,对于组合的增强层和低层,图象的预测和重构仅与低层的各个解码步骤一致。

这一节中所描述的语义和解码过程包括一种“色差联播”的方法。在处理了增强层的数据以后,这种方法可以用来将低层中的 4:2:0 的信号增强成为 4:2:2 的信号。亮度数据还是像以前所述的那样处理,而从低层得到的色差信息(除了内部 DC 值,见 I-7.8.3.4)应被丢弃,并以从增强层中解码而具有更高色差分辨率的新的信息所代替。

在 SNR 分级中,两个层彼此有一种固有的紧密关系。作为一个要求,每层中的相关的图象都应在同一时间解码。

如果低层是与 ISO/IEC 11172-2(而不是本规范)相一致的,那么在解码时要用到两种不同的解谐控制方案,在编码器中要仔细考虑到这个问题。

## I-7.8.1 更高的语法结构

本节中的两个比特流层以从 sequence\_scalable\_extension 中解码的 layer\_id 来区分。

两种比特流应有连续的 layer\_id,增强层 layer\_id=id<sub>enhance</sub>,低层的 layer\_id=id<sub>enhance</sub>-1。

增强层的语法和语义分别定义于 I-6.2 和 I-6.3。

如果低层与 ISO/IEC 11172-2(而非本规范)相一致,则增强层和低层都将使用本规范所定义的“受限的组块结构”。

如下所述,语义限制应用于增强层的头和扩展中的几个值:

### 序列头

除了 bit\_rate, vbv\_buffer\_size, load\_intra\_quantiser\_matrix, intra\_quantiser\_matrix, load\_non\_intra\_quantiser\_matrix 和 non\_intra\_quantiser\_matrix 这些值以外,头应与低层中的头相一致。这些值都可以独立地选择,只有 load\_intra\_quantiser\_matrix 须为零。

### 序列扩展

除了 profile\_and\_level\_indication, chroma\_format, bit\_rate\_extension 和 vbv\_buffer\_size\_extension 这些值以外,这个扩展应与低层中的扩展一致。那些值可以独立地选择。

每个层中的不同的 chroma\_format 的值,都将导致 chroma\_simulcast 标志按表 I-7-26 设置。

增强层的 chroma\_format 应高于或等于低层的 chroma\_format。

表 II -7-26 chroma\_simulcast 标志

chroma_format (低层)	chroma_format (增强层)	chroma_simulcast
4 : 2 : 0	4 : 2 : 0	0
4 : 2 : 0	4 : 2 : 2	1
4 : 2 : 0	4 : 4 : 4	1
4 : 2 : 2	4 : 2 : 2	0
4 : 2 : 2	4 : 4 : 4	1
4 : 4 : 4	4 : 4 : 4	0

如果低层与 ISO/IEC 11172-2(而非本规范)相一致,则 sequence\_extension()不在低层中出现,且认为下面的值用于解码过程:

progressive_sequence	= 1
chroma_format	= "4 : 2 : 0"
horizontal_size_extension	= 0
vertical_size_extension	= 0
bit_rate_extension	= 0
vbv_buffer_size_extension	= 0
low_delay	= 0
frame_rate_extension_n	= 0
frame_rate_extension_d	= 0

增强层中的 sequence\_extension()应取上面所示的值。

### 序列显示扩展

由于增强层没有独立的显示过程,所以这个扩展不出现。

### 序列分级扩展

由于 scalable\_mode = “SNR scalability”,所以这个扩展不出现。

### GOP 头

这个头(如果存在)应与低层中的一样。

### 图象头

除了 vbv\_delay 的值以外,这个头应与低层中的头一样。这个值可以独立地选择。

### 图象编码扩展

除了 q\_scale\_type 和 alternate\_scan 的值以外,这个扩展应与低层中的一样。这两个值可以独立地选择。

如果 chroma\_simulcast 被设置,则 chroma\_420\_type 应置为“0”。否则它就应与低层中的值一样。

如果低层与 ISO/IEC 11172-2(而非本规范)相一致,则低层中不出现 picture\_coding\_extension(),且认为下面的值用于解码过程:

f_code[0][0]	= 低层中的 forward_f_code 或 15
f_code[0][1]	= 低层中的 forward_f_code 或 15
f_code[1][0]	= 低层中的 backward_f_code 或 15
f_code[1][1]	= 低层中的 backward_f_code 或 15
intra_dc_precision	= 0
picture_structure	= "Frame Picture"
top_field_first	= 0
frame_pred_frame_dct	= 1
concealment_motion_vectors	= 0
intra_vlc_format	= 0
repeat_first_field	= 0
chroma_420_type	= 1
progressive_frame	= 1
composite_display_flag	= 0

增强层中的 picture\_coding\_extension()应取上面所示的值。

对于低层,q\_scale\_type 和 alternate\_scan 假定为零。

注意 在增强层中,q\_scale\_type 和 alternate\_scan 可以被独立地设置。

### 量化矩阵扩展

这个扩展是可选的。语义定义于 I-6.3.11 中。

load\_intra\_quantiser\_matrix 和 load\_chroma\_intra\_quantiser\_matrix 都应为零。

注意 下面的解码过程中只会用到非内部矩阵。

### 全景扫描扩展

本扩展不出现。

注意 对于增强层,没有独立的显示过程。如果需要全景扫描功能,可以通过使用低层的全景扫描扩展提供的信息来实现。

### 组块头

组块应与低层中的一致。quantiser\_scale\_code 可以独立地设置,而不需考虑低层。

## I-7.8.2 宏块

下面,用“当前宏块”来表示正在被处理的宏块,用“低层的当前宏块”来表示与当前宏块具有相同的 macroblock\_address 的宏块。

宏块头信息的解码按 I-6.3.16 中的语义实现。

注意 由于增强层中的宏块仅包含低层当前宏的修改数据,所以在 scalable\_mod = = “SNR scalability”时所用到的表 I-B-8 将不设置 macroblock\_intra,macroblock\_motion\_forward 或 macroblock\_motion\_backward 这些标志。然而,低层中的当前宏块的相

应语法元素和标志与遵循 I -7.8.3.5 中描述的反 DCT 的低层和增强层联合解码过程有关。

#### I -7.8.2.1 dct\_type

如 I -6.3.16 中的语义所指出的,在低层和增强层的 macroblock\_modes() 中语法元素 *dct\_type* 可以不出现,也可以出现在两者中之一,或者在两者中均出现。

如果 *dct\_type* 在两层中的 macroblock\_modes() 中均出现,则它们具有相同的值。

#### I -7.8.2.2 跳过的宏块

在增强层中宏块可以被跳过,表示没有作系数增强。此时,解码过程恰如 I -7.6.6 中所述。

在低层中宏块也可被跳过,但仍可在增强层中编码。此时,将用到下面详述的解码过程。

### I -7.8.3 块

增强层块解码过程的第一部分与低层是互相独立的。

增强层块解码过程的第二部分必须与相应的低层块解码过程联合完成。

两个量化系数的集合  $F''_{lower}$  和  $F''_{enhance}$  加到一起形成  $F''$ (见图 I -7-15)。

$F''_{lower}$  由低层中得出,如 I -7.1~I -7.4.2.3 所定义。

$F''_{enhance}$  按下面章节所定义而得到。

结果信号  $F''$  被进一步处理,从饱和化开始,如 I -7.4.4 至 I -7.6(I -7.7, I -7.9) 中所定义。

#### I -7.8.3.1 变长码解码

在增强层块中,VLC 解码按 I -7.2,作为针对非内部块(如 macroblock\_intra = 0 指出)而实现。

#### I -7.8.3.2 反扫描

反扫描严格按 I -7.3 中所定义的实现。

#### I -7.8.3.3 反量化

在增强层块中,反量化按 I -7.4.2,作为针对非内部块而实现。

如果低层与 ISO/IEC 11172-2(而非本规范)相一致,则用来得到  $F''_{lower}[v][u]$ (见图 I -7-14)的“反量化算法”应包含 IDCT 解谐控制和饱和化,如 ISO/IEC 11172-2 中所规定。

#### I -7.8.3.4 两层系数的相加

来自各层的块的相应系数应被加在一起形成信号  $F''$ (见图 I -7-15)。

$F''[v][u] = F''_{lower}[v][u] + F''_{enhance}[v][u]$ , 对于所有  $v, u$

如果 chroma\_simulcast = 1, 则仅亮度块按上面所述处理。

对于色差块来说,用基层的 DC 系数来作增强层中相应块的 DC 系数的预测,而基层的 AC 系数被丢弃,增强层的 AC 系数按下面公式来形成图 I -7-14 中的信号  $F''$ :

$$F''[0][0] = F''_{lower}[0][0] + F''_{enhance}[0][0]$$

$$F''[v][u] = F''_{enhance}[v][u], \text{ 对于所有 } u, v, \text{ 除了 } u=v=0.$$

注意 色差联播块如非内部块那样被反量化,且使用色差非内部矩阵。

表 I -7-27 给出了色差块的索引,它的 DC 系数( $F''_{lower}[0][0]$ )将用来预测增强层的相应色差块的 DC 系数( $F''_{enhance}[0][0]$ )。

表 I -7-27 用于预测 DC 系数的块索引

chroma-format	块索引							
	4	5	6	7	8	9	10	11
基:4:2:0 上:4:2:2	4	5	4	5				
基:4:2:0 上:4:4:4	4	5	4	5	4	5	4	5
基:4:2:2 上:4:4:4	4	5	6	7	4	5	6	7

#### I -7.8.3.5 剩余的宏块解码步骤

在对两层的系数相加之后,由于现在仅剩有一个数据流  $F''[v][u]$  将被处理,则剩下的宏块解码步骤如 I -7.7.4~I -7.6(I -7.7, I -7.9, 如果用到)所描述。

在这个过程中,时/空预测信号  $P[y][x]$  根据由低层所知的宏块类型语法元素和当前宏块的标志而得出。

## I -7.9 时域分级

时域分级涉及到两层,一个低层,一个增强层。低层和增强层处理相同的空间分辨率。增强层增强低层的时域分辨率,并且如果同低层信号再次多路传输,则提供完整的时域速率,这就是增强层中指明的帧速率。增强层图象的解码过程与 I -7.1~I -7.6 中描述的一般解码过程相同。仅有的差异是 I -7.6.2 中所描述的“预测帧和场的选择”。

用于预测的参考图由 reference-select-code 选择,如表 I -7-28 中所述。在 P-图中,前向参考图可能是下面三个中之一:最近的增强层图,最近的低层帧或按显示顺序的下一个低层帧。注意,在后一种情况中,低层中用于预测的参考帧在时间上是落后的。

在 B-图中,前向参考可以是下面两个中之一:最近的增强层图或最近的(或时间上一致)低层帧,而后向参考可能是下面两个中之一:包括按显示顺序在时间上一致的图的最近低层图或按显示顺序的下一个低层帧。注意,在这种情况下,用于预测的低层中的参考帧在时间上是超前的。

后向预测不能由增强层中的图象作出。这避免了增强层中的帧重排。运动补偿过程使用低层的解码图和/或以前的来自增强层的时域预测来形成预测。

增强层可能包含 I-图,P-图或B-图,但增强层中的B-图表现得更像P-图,这是由于在增强层中解码的B-图可以用来预测后面的B-图或P-图。

当低层中最近的帧用作参考时,这包括在时间上与增强层中的帧或首场(对于场图)相一致的帧。P-图和B-图使用的预测参考分别如表 I -7-28 和表 I -7-29 所示。

低层和增强层应使用受限的组块。

表 I -7-28 P-图中选择的预测参考

reference_select_code	向前预测参考
00	最近的解码增强层图
01	按显示顺序的最近低层帧
10	按显示顺序的下一个低层帧
11	禁止

表 I -7-29 B-图中选择的预测参考

reference_select_code	前向预测参考	后向预测参考
00	禁止	禁止
01	最近的解码增强图	按显示顺序的最近的低层图
10	最近的解码增强图	按显示顺序的下一个低层图
11	按显示顺序的最近的低层图	按显示顺序的下一个低层图

图 I -7-16 示出了使用时域分级的对增强层的运动补偿过程的简化图。

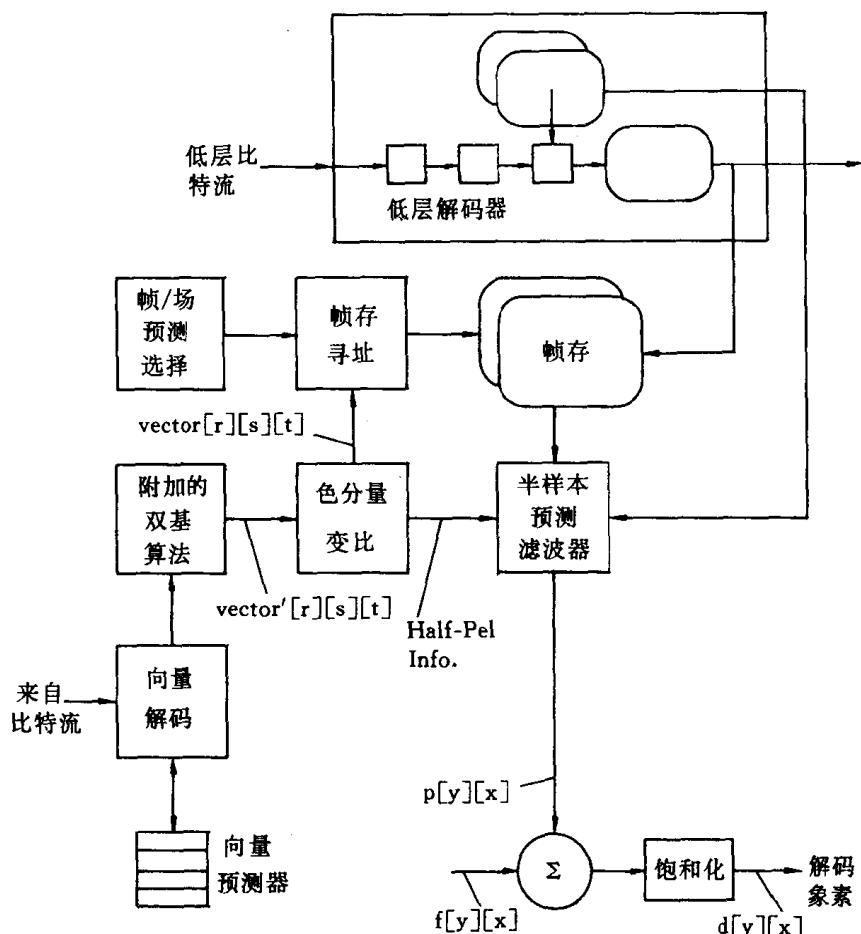


图 I -7-16 使用时域分级的对增强层的运动补偿过程的简化图

I-图不使用预测参考,为了指明这一点,I-图的 reference\_select\_code 应为‘11’。

根据 picture\_coding\_type,forward\_temporal\_reference 和 backward\_temporal\_reference 应取零值来表示它们未指出用作预测的参考。

## II -7.9.1 更高的语法结构

本节中的两个比特流层通过由 sequence\_scalable\_extension 中解码而得的 layer\_id 来区分。

两个比特流层应具有连续的 layer\_id,增强层的 layer\_id 为  $\text{id}_{\text{enhance}}$ ,低层的 layer\_id =  $\text{id}_{\text{enhance}} - 1$ 。

增强层的语法和语义分别定义于 II -6.2 节和 II -6.3 节。

如下所述,语义限制适用于增强层的头和扩展中的几个值。

低层应与本标准一致(而非 ISO/IEC 11172-2)。

### 序列头

除了 horizontal\_size\_value,vertical\_size\_value 和 aspect\_ratio\_information 以外,这个头中的值可能与低层中相应的头中的值不同。

### 序列扩展

除了 profile\_and\_level\_indication,bit\_rate\_extension,vbv\_buffer\_size\_extension,low\_delay,frame\_rate\_extension\_n 和 frame\_rate\_extension\_d 这几个值以外,这个扩展应与低层中的相一致。这些值可以独立地选择。值得注意的是,progressive\_sequence 指出了增强层帧的扫描格式,而不是多路传输以后的输出帧的扫描格式。后者由 mux\_to\_progressive\_sequence 指明(见“序列分级扩展”)。

### 序列显示扩展

由于对于增强层来说没有独立的显示过程,所以这个扩展不出现。

### 序列分级扩展

这个扩展在 scalable\_mod=“时域分级”时才出现。

当 progressive\_sequence = 0 且 mux\_to\_progressive\_sequence = 0 时,top\_field\_first 和 picture\_mux\_factor 可被选。

如 progressive\_sequence = 0 而 mux\_to\_progressive\_sequence = 1,则 top\_field\_first 应包含低层的 top\_field\_first 的补码值,但 picture\_mux\_factor 应为 1。

如 progressive\_sequence = 0,而 mux\_to\_progressive\_sequence = 1,则 top\_field\_first 应为 0,但 picture\_mux\_factor 能被选择。

progressive\_sequence=1 和 mux\_to\_progressive\_sequence=0 的组合不应出现。

### GOP 头

对于 GOP 头(如果存在的话)没有什么限制,和低层的 GOP 头一样。

### 图象头

对于图象头没有什么限制,如低层的图象头一样。

### 图象编码扩展

除了 top\_field\_first,concealment\_motion\_vectors 和 chroma\_420\_type 以及 pro-

gressive\_frame 这几个值外,这个扩展中的值与低层中的不同。top\_field\_first 应基于 progressive\_sequence 和 mux\_to\_progressive\_sequence(见上述序列分级扩展),concealment\_motion\_vectors 应为零。chroma\_420\_type 应和低层中的一样。progressive\_frame 总应与 progressive\_sequence 的值一样。

#### 图象时域分级扩展

每个图象中均出现这个扩展。

#### 量化矩阵扩展

增强层中可能出现这个扩展。

### II -7.9.2 对于时域预测的限定

虽然时域预测可以由 forward\_temporal\_reference 或 forward\_temporal\_reference 和 backward\_temporal\_reference 二者指出的解码图作出,但如果预测的形成是使用在时间上与被预测的增强层图非常接近的增强层和低层的解码图,则时域分级的效率是高的。对于比特流有一个要求:B-图和 P-图应从最近的或下一个图形成预测,如表 I -7-28 和表 I -7-29 所述。

如果 group\_of\_pictures\_header 在 lower\_layer 中出现得非常频繁,则由于时域参考可能不唯一而发生混淆(时域参考在每个 group\_of\_pictures\_header 被复位)。这种混淆应通过系统层时间信息的帮助来解决。

### II -7.10 数据分割

数据分割是将一个视频流分成两层的技术,称为分割。通过一个优先级断点来指明哪个语法元素被放在 0 区,也就是基本区(或称高优先区)。剩下的比特流放入 1 区(也称低优先区)。序列、GOP 和图象头均冗余拷贝于 1 区中,以实现错误复原。sequence\_end\_code 也冗余拷贝于 1 区中。冗余的头中所有域都应与原来的域一致。1 区中仅允许(且需要)扩展 sequence\_extension(),picture\_coding\_extension() 和 sequence\_scalable\_extension()。

注意 在两个分区中,还包括语法元素 extra\_bit\_slice。均遵循 I -6.2.4 中定义的 slice() 语法。

优先级断点的解释在表 II -7-30 中给出。

表 II -7-30 优先级断点值和相关的语义

优先级断点	包含于 0 区中的语法元素
0	这个值为 1 区保留。1 区中所有组块的优先级断点值都为 0
1	序列、GOP、图象和 slice()(直到其 extra_bit_slice)中的所有数据
2	包括上面的所有数据,再加上宏块语法元素,还包含 macroblock_address_increment
3	包括上面的所有数据,再加上宏块语法元素,直到 coded_block_pattern(),但不包含它
4~63	保留

续表

优先级断点	包含于 0 区中的语法元素
64	直到且包含 coded-block-pattern() 或 DC 系数(dct_dc_differential) 的所有语法元素, 和第一个 DCT 系数组(run,level), (或 EOB)
65	上面的所有语法元素, 加上两个 DCT 系数组(run,level)
...	
63+j	上面的所有语法元素, 加上 j 个 DCT 系数组(run,level)
...	
127	上面的所有语法元素, 加上 64 个 DCT 系数组(run,level)

VBV 的语义保持不变, 即, VBV 指的是两个分区的总和, 而不是单一的任一个。

比特流参数 bit\_rate(bit\_rate\_value 和 bit\_rate\_extension), vbv\_buffer\_size(vbv\_buffer\_size\_value 和 vbv\_buffer\_size\_extension) 和 vbv\_delay 在这两个分区中应取相同值。这些参数指出了由两个分区形成的整个比特流的特性。

解码过程按下列方法修改:

将 current\_partition 置为 0, 从包含 sequence\_scalable\_extension 的比特流(0 区)开始解码;

如果 current\_partition=0, 则检查比特流中的当前点是否优先级断点:

如果是, 将 current\_partition 置为 1, 下一项从 1 区解码获得;

否则, 继续从 0 区解码。从两个分区中移去序列、GOP 和图象头;

如果 current\_partition=1, 则检查优先级断点来判断下一个要被解码的项是否在 0 区中:

如果是, 将 current\_partition 置为 0。下一项从 0 区解码获得;

否则, 继续从 1 区解码。

图 II-7-17 中示出了一个例子, 其中优先级断点置于 64(一对(run,level))。

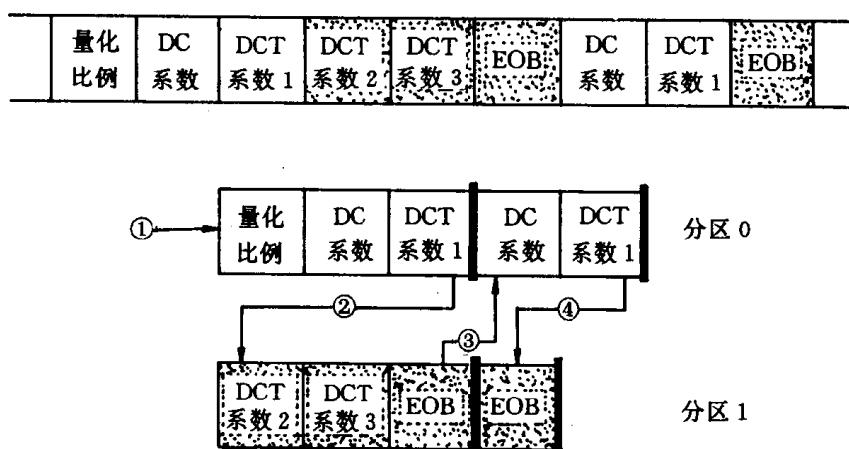


图 II-7-17 priority\_breakpoint 置为 64(一对(run,level)),  
带有两个分区的比特流的分割。两个分区如图所示,  
用箭头指出解码器需要如何在两分区间切换

## II -7.11 混合分级

混合分级是两种不同类型的分级的组合。可以被组合的分级是 SNR 分级,空域分级和时域分级。当两种类型的分级被组合在一起的时候,需对三个比特流进行解码,这些比特流所属层的命名如表 II -7-31 所示。

表 II -7-31 层的名字

layer_id	名 字	layer_id	名 字
0	基本层	2	增强层 2
1	增强层 1	...	...

对于在增强层 1 和增强层 2 之间的分级,增强层 1 是它的低层,增强层 2 是它的增强层。没有层次可以从阶梯体系中省略,例如,如果在增强层 1 和增强层 2 之间是 SNR 分级,则增强层 1 中的预测类型对增强层 1 和增强层 2 的组合解码过程也是有效的。

带有时域分级和空域分级的层的耦合关系比带有 SNR 分级的层的耦合关系要松散。因此,在这些分级中,在基层能被用在增强层中之前,首先应对其解码并向上转换。在 SNR 分级中,两层同时解码。解码顺序总结如下:

第一种情况:

基层

<空域或时域分级>

增强层 1

<SNR 分级>

增强层 2

首先对基层解码,然后同时对两个增强层解码。

第二种情况:

基层

<SNR 分级>

增强层 1

<时域或空域分级>

增强层 2

首先对基层和增强层 1 同时解码,然后对增强层 2 解码。

第三种情况:

基层

<空域或时域分级>

增强层 1

<空域或时域分级>

增强层 2

首先对基层解码,然后对增强层 1 解码,最后对增强层 2 解码。

## II - 7. 12 解码过程输出

这一部分描述了能对与本规范一致的比特流进行解码的解码过程理论模型的输出。

解码过程的输入是视频数据,由一个或多个层组成。这些视频层通常以系统流方式多路传输,而且包含时间信息。

解码过程的输出是一系列的场或帧,通常是显示过程的输入,解码过程输出场或帧的顺序通常称为显示顺序,可能与解码顺序不同(当使用 B-图时),显示过程负责执行将这些解码场或帧显示于显示设备上的操作。如果显示设备不能按比特流中指出的帧速率显示,则显示过程可以实现帧速率转化。本规范既未描述显示过程的理论模型,也未描述其操作。

由于显示过程可能需要一些语法元素,如 progressive\_frame,所以在这个解码过程的理论模型中,被解码过程解码的所有语法元素都由解码过程输出,并可由显示过程访问。

并且,解码过程所不使用的帧中,样本的位置(空间上和时间上的),显示过程也必须知道。

当逐行序列被解码时(progressive\_sequence 等于 1),重构帧的亮度和色差样本以逐行帧的形式由解码过程输出,并且输出速率即为帧速率。图 I - 7-18 阐明了当 chroma\_format 等于 4 : 2 : 0 时的情况。

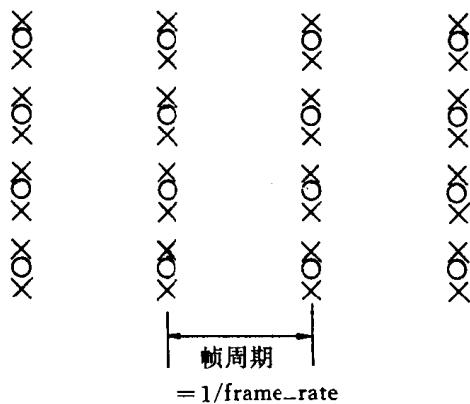


图 I - 7-18 progressive\_sequence == 1

如果 repeat\_first\_field 等于 0,则同样的重构帧输出一次;如果 repeat\_first\_field 等于 1,则连续输出两次或三次,这还要取决于 top\_field\_first 的值。图 I - 7-19 阐明了当 chroma\_format 为 4 : 2 : 0 且 repeat\_first\_field 等于 1 时的情况。

当对隔行序列(progressive\_sequence 等于 0)解码时,重构帧的亮度样本以隔行场的形式由解码过程输出,速率是帧速率的两倍。图 I - 7-20 阐明了这种情况。

作为对比特流的一个要求,在解码过程输出的场总应为交替的顶场和底场(注意一个

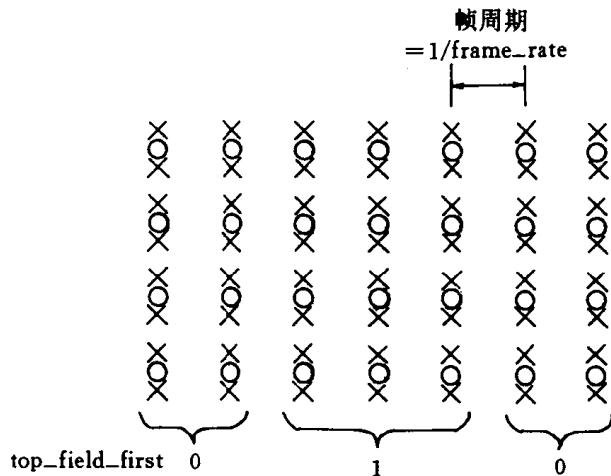


图 I -7-19 `progressive_sequence == 1,repeat_first_field == 1`

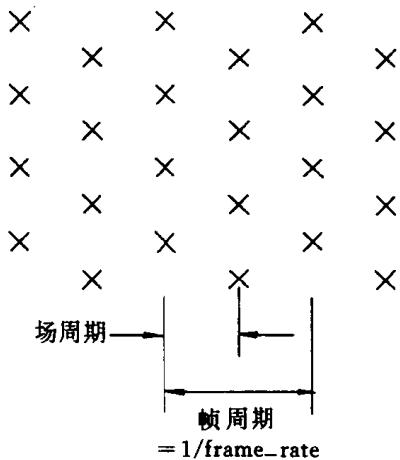


图 I -7-20 `progressive_sequence == 0`

序列的首场既可为顶场也可为底场)。

如果重构帧是隔行的(`progressive_frame` 等于 0), 则亮度和色差样本以两个连续场的形式由解码过程输出。解码过程输出的首场可为重构帧的底场或顶场, 这取决于 `top_field_first` 的值。

虽然逐行帧的所有样本及时表示同一事物, 但当序列是隔行时, 所有这些样本不是由解码过程在同一时间输出。

如果重构帧是逐行的(`progressive_frame` 等于 1), 则亮度样本以两个或三个连续场的形式由解码过程输出, 这取决于 `repeat_first_field` 的值。

注意 这些场都是源于比特流中同一逐行帧, 这一信息被传到显示过程。

重构的逐行帧的所有色差样本与亮度样本的首场同时由解码过程输出。这由图 I -7-21 和 I -7-22 阐明。

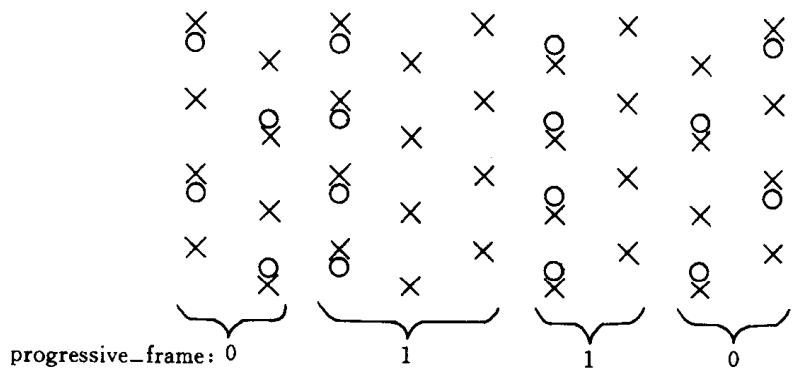


图 I-7-21 4 : 2 : 0 色差格式下 progressive\_sequence == 0

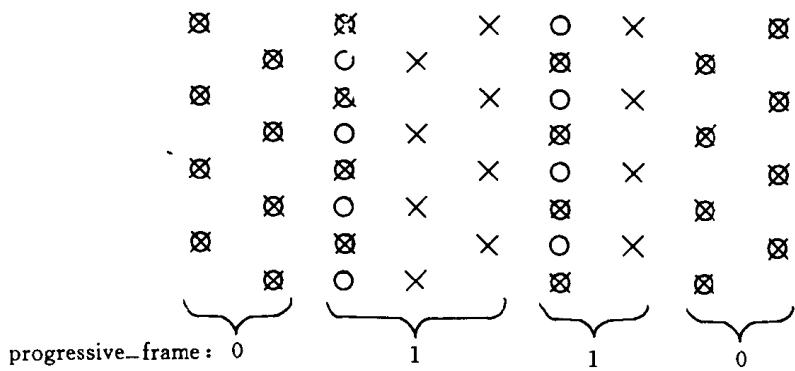


图 I-7-22 色差格式为 4 : 2 : 2 或 4 : 4 : 4, progressive\_sequence == 0

## II -8 框架和级别

注意 在本规范中,框架一词按下面定义使用。它不能与框架的其它定义相混淆,特别是不能与 JTC1/SGFS 的定义相混淆。

框架和级别提供了一种定义本规范的语法和语义的子集的手段,由此也就定义了针对某一特定比特流的解码器能力。框架是本规范所定义的整个比特流语法的一个限定的子集。级别是影响比特流中参数的限制集合。一致性检测将针对限定的框架、在限定的级别实现。

以框架和级别的形式定义一致性点的目的是使不同的应用之间交换比特流变得容易。我们鼓励本规范的实现者生成与那些限定的一致性范围相对应的解码器和比特流。分立定义的框架和级别是本规范的应用系统之间交换比特流的方法。

在本章中,描述了限定的框架和级别的约束部分。所有未被直接约束的语法元素和参数值可以取任何本规范所允许的可能的值。如果一个解码器能对某个框架某个级别所规定的语法元素的所有允许值正确解码,则称这个解码器与这个给定的框架在给定的级别相一致,如果一个比特流不超过允许值的允许范围且不包括不允许的语法元素,则认为这个比特流是与这个给定框架在给定级别上相一致。

应注意 II -5.4 节,它定义了描述数字范围的规范,这贯穿了所有对值和参数的描述。

在 sequence\_extension 中的 profile\_and\_level\_indication 指出了该比特流的框架和级别。这个参数中各个位的意义在表 II -8-1 中定义。

表 II -8-1 profile\_and\_level\_indication 中位的含义

位	场大小(位)	含义
[7 : 7]	1	换码位
[6 : 4]	3	框架识别
[3 : 0]	4	级别识别

表 II -8-2 规定了框架的识别码,表 II -8-3 规定了级别的识别码,当换码位等于零时,具有较大值的识别码的框架是具有较小值的识别码的框架的子集。类似地,如果换码位等于零,则具有较大值的识别码的级别是具有较小值的识别码的级别的子集。

表 II -8-2 框架识别

框架识别码	框 架
110~111	(保留)
101	简单
100	主

续表

框架识别码	框 架
011	SNR 分级
010	空域
001	高
000	(保留)

表 II -8-3 级别识别

级别识别码	级 别
1011~1111	(保留)
1010	低
1001	(保留)
1000	主
0111	(保留)
0110	高 1440
0101	(保留)
0100	高
0000~0011	(保留)

表 II -8-4 描述了换码位为 1 的框架和级别。对于这些框架和级别,没有由 profile\_and\_level\_indication 分配的暗含的等级体系,框架和级别无需为它者之子集。

表 II -8-4 换码 profile\_and\_level\_indication 识别

profile_and_level_indication	名 称
10000000 到 11111111	(保留)

应对附录 II -E 加以注意,它详细描述了 ISO/IEC 13818-2 中用于给定的框架和级别的部分。

## II -8. 1 ISO/IEC 11172-2 的兼容性

ISO/IEC 11172-2 中的约束参数的比特流对于简单框架、主框架、空域分级框架和高框架在所有级别都是可解码的,当产生了一个与 ISO/IEC 11172-2 约束参数编码相一致的比特流时,constrained\_parameters\_flag 应被置位。

此外,简单、主、空域分级、SNR 分级和高框架解码器应能对在这个解码器的级别约束之内的 ISO/IEC 11172-2 的 D-图比特流解码。

## II -8.2 限定的框架间的关系

简单、主、SNR 分级、空域分级和高框架间有等级关系。因此，“更高”的框架所支持的语法包含“较低”框架的所有语法元素(例如，对于一个给定的级别，一个主框架解码器应能够对一个与简单框架限定相一致的比特流解码)。对于一个给定的框架，对于不同的级别，支持同样的语法集。等级顺序在表 II -8-2 中给出。

框架间的语法差异在表 II -8-5 中给出。

表 II -8-5 框架的语法限制

语法元素	框 架				
	简单	主	SNR	空域	高
chroma_format	4 : 2 : 0	4 : 2 : 0	4 : 2 : 0	4 : 2 : 0	4 : 2 : 2 或 4 : 2 : 0
frame_rate_extension_n	0	0	0	0	0
frame_rate_extension_d	0	0	0	0	0
picture_coding_type	I,P	I,P,B	I,P,B	I,P,B	I,P,B
repeat_first_field	受限的		不受限的		
sequence_scalable_extension()	No	No	Yes	Yes	Yes
scalable_mode	—	—	SNR	SNR 或空域	SNR 或空域
picture_spatial_scalable_extension()	No	No	No	Yes	Yes
intra_dc_precision	8,9,10	8,9,10	8,9,10	8,9,10	8,9,10,11
Slice structure	受限的 见 II -6.1.3.2				

对于所有限定的框架，有一个对比特流的限制：针对宏块的所有数据应以少于表 II -8-6 中指出的位数来表示。然而，在每个水平行的任意两个宏块可以超越这一限制。

在这个意义上，我们认为宏块由 macroblock\_address\_increment(或 macroblock\_escape，如果有的话)的第一位开始，直到最后一个编码块的块尾标志的最后一一位(或者，如果没有编码块，则为 coded\_block\_pattern() 的最后一位)。在这些宏块之前(或之后)任何组块所需的位不计为宏块的一部分。

表 II -8-6 宏块中最多位数

chroma_format	最多位数
4 : 2 : 0	4608
4 : 2 : 2	6144
4 : 4 : 4	9216

在简单和主框架比特流中使用 repeat\_first\_field 要受到限制,如表 II -8-7 所述。

表 II -8-7 repeat\_first\_field 使用限制

frame_rate_code	frame_rate_value	progressive_sequence == 0	progressive_sequence == 1
repeat_first_field	repeat_first_field		
0000	禁止		
0001	24 000/1001(23.976)	0	0
0010	24	0	0
0011	25	0 或 1	0
0100	30 000/1001(29.97)	0 或 1	0
0101	30	0 或 1	0
0110	50	0 或 1	0
0111	60 000/1001(59.94)	0 或 1	0 或 1
1000	60	0 或 1	0 或 1
...	保留		
1111	保留		

高框架也是靠具有对亮度样本速率、最大比特率和 VBV 缓冲区大小的不同限制来区分的。参见表 II -8-11,表 II -8-12 和表 II -8-13。

符合简单框架主级别的解码器应能对“主”框架“低”级别的比特流解码。

### II -8.3 限定的级别间的关系

低,主,高-1440 和高级别间具有等级关系。因此,“高”级别的参数限制等于或超过“低”级别的参数限制(例如,对于一个给定的框架,主级别解码器应能对与低级别限制相一致的比特流进行解码)。等级顺序已在表 II -8-3 中给出。

对于各种级别的不同参数限制在表 II -8-8 中给出。

表 I -8-8 级别的参数限制

语法元素	级 别			
	低	主	高-1440	高
f_code[0][0](前向水平)	[1 : 7]	[1 : 8]	[1 : 9]	[1 : 9]
f_code[0][1](前向竖直)	[1 : 4]	[1 : 5]	[1 : 5]	[1 : 5]
f_code[1][0]* (后向水平)	[1 : 7]	[1 : 8]	[1 : 9]	[1 : 9]
f_code[1][1]* (后向竖直)	[1 : 4]	[1 : 5]	[1 : 5]	[1 : 5]
垂直向量范围帧图†	[-64 : 63.5]	[-128 : 127.5]	[-128 : 127.5]	[-128 : 127.5]
垂直向量范围场图†	[-32 : 31.5]	[-64 : 63.5]	[-64 : 63.5]	[-64 : 63.5]
frame_rate_code	[1 : 5]	[1 : 5]	[1 : 8]	[1 : 8]
样本密度	见表 I -8-10			
亮度样本速率	见表 I -8-11			
最大比特率	见表 I -8-12			
缓存大小	见表 I -8-13			

\* 对于不包括 B-图的“简单”框架比特流, backward\_horizontal\_f\_code 和 backward\_vertical\_f\_code 应被置为 15(不使用)。

† 这个限制适用于最终的重构运动向量。对于双基运动向量, 在变比之前、变比之后且加入小差分运动向量之后应用这个限制。

## II -8.4 分 级 的 层

SNR 分级框架, 空域分级框架和高框架可以使用一个以上的比特流来对图象编码。这些不同的比特流代表了编码的层, 它们组合起来可以产生比从单独的每一层获得的图象更高质量的图象(见附录 I -D)。对于一个给定框架的最多层数如表 I -8-9 所述。分级的层按表 I -7-29 取名。当使用最多允许层数编码时, 这些框架/级别组合的语法和参数限制在表 I -8-10, I -8-11, I -8-12 和表 I -8-13 中给出。如果层数少于最大允许值, 从表 I -8-15 到表 I -8-17 可以得到合适的参考。

应该注意, SNR 框架比特流的基层总能被相同级别的主框架解码器解码。反之, 主框架比特流应能被相同级别的 SNR 框架解码器解码。

表 II-8-9 在“SNR”、“空域”和“高”框架中分级层的上限

级别	哪些层的最大数	框 架		
		SNR	空域	高
高	所有层(base+enh.)			3
	空域增强层			1
	SNR 增强层			1
高-1440	所有层(base+enh.)		3	3
	空域增强层		1	1
	SNR 增强层		1	1
主	所有层(base+enh.)	2		3
	空域增强层	0		1
	SNR 增强层	1		1
低	所有层(base+enh.)	2		
	空域增强层	0		
	SNR 增强层	1		

#### II -8.4.1 允许的层组合

在 SNR 框架、空域框架和高框架中,有许多可能的层的组合。为了使交互操作可以最大限度地利用,仅允许全部组合的一个子集,并且对某些组合来讲,参数限制比表 I -8-10,表 I -8-11 和表 I -8-13 所指出的还要严格,这些附加的限制是为了保证低层比特流解码可由一个限定的较低框架/级别来实现。

下面的表是一个可允许的组合的总结,并遵循下面的规则。

- SNR 框架最多有两层;空域和高框架最多有 3 层(见表 I -8-9)。
- 在三层组合中,仅允许有一个 SNR 和一个空域分级,可为 SNR/空域或空域/SNR 顺序。
- 将 4 : 2 : 2 色差格式加到一个 4 : 2 : 0 的低层上,被认为是一个 SNR 分级。
- 如果低层为 4 : 2 : 2,则不允许有 4 : 2 : 0 的层。(见 II -7.8.1)

框架	分 级 方 式			最简单的 基层解码器的 框架/级别 (级别 ref. 顶层)*
	基层	增强层 1	增强层 2	
SNR	4 : 2 : 0	—	—	MP@same level
SNR	4 : 2 : 0	SNR	—	MP@same level
空域	4 : 2 : 0	—	—	MP@same level
空域	4 : 2 : 0	SNR,4 : 2 : 0	—	MP@same level
空域	4 : 2 : 0	空域,4 : 2 : 0	—	MP@(level-1)
空域	4 : 2 : 0	SNR,4 : 2 : 0	空域,4 : 2 : 0	MP@(level-1)
空域	4 : 2 : 0	空域,4 : 2 : 0	SNR,4 : 2 : 0	MP@(level-1)

续表

		分 级 方 式			最简单的 基层解码器的 框架/级别 (级别 ref. 顶层)*
框架	基层	增强层 1	增强层 2		
高	4 : 2 : 0	—	—	HP@same level HP@same level	
	4 : 2 : 2	—	—		
高	4 : 2 : 0	SNR, 4 : 2 : 0	—	HP@same level HP@same level	
	4 : 2 : 0	SNR, 4 : 2 : 2	—		
高	4 : 2 : 2	SNR, 4 : 2 : 2	—	HP@same level HP@(level-1)	
	4 : 2 : 0	空域, 4 : 2 : 0	—		
高	4 : 2 : 2	空域, 4 : 2 : 2	—	HP@same level HP@(level-1)	
	4 : 2 : 0	SNR, 4 : 2 : 0	空域, 4 : 2 : 0		
高	4 : 2 : 0	SNR, 4 : 2 : 2	空域, 4 : 2 : 2	HP@same level HP@(level-1)	
	4 : 2 : 2	SNR, 4 : 2 : 2	空域, 4 : 2 : 2		
高	4 : 2 : 0	空域, 4 : 2 : 0	SNR, 4 : 2 : 0	HP@same level HP@(level-1)	
	4 : 2 : 0	空域, 4 : 2 : 0	SNR, 4 : 2 : 2		
高	4 : 2 : 2	空域, 4 : 2 : 2	SNR, 4 : 2 : 2	HP@same level HP@(level-1)	
	4 : 2 : 2	空域, 4 : 2 : 2	SNR, 4 : 2 : 2		

\* 在假定比特流可以包含除了分级以外给出的框架所允许的任何语法的前提下,描述了最简单的可对基层解码的解码器。注意,对于高框架、主级别、空域分级的比特流,‘HP@(级别-1)’变为‘MP(级别-1)’。

## II -8.5 限定的框架、级别和层的参数值

表 II -8-10 采样密度上限

级别	空域分辨率层		框 架				
			简单	主	SNR	空域	高
高	增强	样本/行 行/帧 帧/s		1920 1152 60			1920 1152 60
	低	样本/行 行/帧 帧/s					960 576 30
高- 1440	增强	样本/行 行/帧 帧/s		1440 1152 60		1440 1152 60	1440 1152 60
	低	样本/行 行/帧 帧/s				720 576 30	720 576 30

续表

级别	空域分辨率层		框 架				
			简单	主	SNR	空域	高
主	增强	样本/行	720	720	720		720
		行/帧	576	576	576		576
	低	帧/s	30	30	30		30
		样本/行	—	—	—		352
低	增强	行/帧	—	—	—		288
		帧/s	—	—	—		30
	低	样本/行	—	—	—		—
		行/帧	—	—	—		—
		帧/s	—	—	—		—

注意 对于单一层的或 SNR 分级的编码，“增强层”所规定的限制仍适用。

这个表所参考的语法元素如下：

样本数/行： **horizontal\_size\_value**

行数/帧： **vertical\_size\_value**

帧数/s： **frame\_rate\_value**

表 II-8-11 亮度样本速率的上限(样本/s)

级别	空域分辨率层	框 架				
		简单	主	SNR	空域	高
高	增强	—	62 668 800	—	—	62 668 800 (4 : 2 : 2) 83 558 400 (4 : 2 : 0)
	低	—	—	—	—	14 745 600 (4 : 2 : 2) 19 660 800 (4 : 2 : 0)
高-1440	增强	—	47 001 600	—	47 001 600	47 001 600 (4 : 2 : 2) 62 668 800 (4 : 2 : 0)
	低	—	—	—	10 368 000	11 059 200 (4 : 2 : 2) 14 745 600 (4 : 2 : 0)
主	增强	10 368 000	10 368 000	10 368 000	—	11 059 200 (4 : 2 : 2) 14 745 600 (4 : 2 : 0)
	低	—	—	—	—	3 041 280 (4 : 2 : 0)
低	增强	—	3 041 280	3 041 280	—	—
	低	—	—	—	—	—

注意 对于单一层或 SNR 分级编码，“增强层”所规定的限制仍适用。

亮度样本速率 P 定义如下：

$$P = \text{horizontal\_size\_value} \times \text{vertical\_size\_value} \times \text{frame\_rate\_value}$$

表 II-8-12 比特率的上限(Mbit/s)

级别	框 架				
	简单	主	SNR	空域	高
高		80			100 所有层 80 中十基层 25 基层
高-1440		60		60 所有层 40 中十基层 15 基层	80 所有层 60 中十基层 20 基层
主	15	15	15 双层 10 基层		20 所有层 15 中十基层 4 基层
低		4	4 双层 3 基层		

注意 1 这个表定义了对于固定比特率操作的最大编码数据率, 和对于可变比特率操作的最大的基本流率 Res(max), 由 bit\_rate 指出(见 II-6.3.3)。也可参考 ISO/IEC 13818-1 的 I-2.4.2。

注意 2 这个表定义了对于给定层以下(包括给定层)的所有层的允许数据率。对于多层编码应用, 在各层之间按比例分配的数据率仅受给定层所允许的最大数据率限制, 如本表所述。

注意 3 1M bit = 1 000 000 bit

表 II-8-13 VBV 缓冲区大小的要求(bit)

级别	层	框 架				
		简单	主	SNR	空域	高
高	(增) Enh. 2					12 222 464
	(增) Enh. 1					9 781 248
	(基) Base		9 781 248			3 047 424
高-1440	(增) Enh. 2				7 340 032	9 781 248
	(增) Enh. 1				4 882 432	7 340 032
	(基) Base		7 340 032		1 835 008	2 441 216
主	(增) Enh. 2			1 835 008		2 441 216
	(增) Enh. 1			1 212 416		1 835 008
	(基) Base	1 835 008	1 835 008			475 136
低	(增) Enh. 2			475 136		
	(增) Enh. 1			360 448		
	(基) Base		475 136			

注意 1 缓冲区大小与允许的比特率成比例, 下取整到与  $16 \times 1024$  bit 的倍数最接近的值。变比的参照值为“主”框架、“主”级别的缓冲区大小。

注意 2 这个表定义了给定层以下(包括给定层)解码所需的整个解码器缓冲区大小。对于多层编码应用, 各层的缓冲区申请存储空间的大小仅受表中指出的一个给定层的最大允许值的限制。

注意 3 与这个表相关的语法元素是 vbv\_buffer\_size(见 II-6.3.3)。

表 II-8-14 到表 II-8-18 给出了比特流每层的参数限制, 和能对每层完整解码的解码器的最小框架/级别。每个表都描述了在框架/级别矩阵中的单一一致点的限制。

表 II-8-14 分层框架的详细规范

级别	框 架		
	SNR	空域	高
高	—	—	见附录 I-E
高-1440	—	表 II-8-17	见附录 I-E
主	表 II-8-16	—	见附录 I-E
低	表 II-8-15	—	—

注意 对高框架的完整规范尚未完成。附录 I-E 中的表是为了引导的目的。

在下面的表中,采用了如下标记:

<框架缩写>@<级别缩写>

缩写在表 II-8-14a 中定义。

表 II-8-14a 框架和级别名称缩写

框 架	<框架缩写>	级 别	<级别缩写>
简单	SP	低	LL
主	MP	主	ML
SNR 分级	SNR	高-1440	H-14
空域分级	Spatial	高	HP
高	HP		

表 II-8-15 “SNR”框架@“低”级别

层 #	layer_id	分级方式	最大样本密度	最大亮度样本速率	最大总比特率/ $10^6$	最大总 VBV 缓冲区	最小解码器
1	0	—	MP@LL	MP@LL	4	475 136	MP@LL
2	0	—	MP@LL	MP@LL	3	360 448	MP@LL
2	1	SNR	MP@LL	MP@LL	4	475 136	SNR@LL

表 II-8-16 “SNR”框架@“主”级别

层 #	layer_id	分级方式	最大样本密度	最大亮度样本速率	最大总比特率/ $10^6$	最大总 VBV 缓冲区	最小解码器
1	0	—	MP@ML	MP@ML	15	1 835 008	MP@ML
2	0	—	MP@ML	MP@ML	10	1 212 416	MP@ML
2	1	SNR	MP@ML	MP@ML	15	1 835 008	SNR@ML

表 II -8-17 “空域”框架@“高-1440”级别

层 #	layer_id	分级方式	最大样本密度	最大亮度样本速率	最大总比特率/ $10^6$	最大总 VBV 缓冲区	最小解码器
1	0	—	MP@H-14	MP@H-14	60	7 340 032	MP@H-14
2	0	—	MP@H-14	MP@H-14	40	4 882 432	MP@H-14
2	1	SNR	MP@H-14	MP@H-14	60	7 340 032	Spatial@H-14
2	0	—	MP@ML	MP@ML	15	1 835 008	MP@ML
2	1	空域	MP@H-14	MP@H-14	60	7 340 032	Spatial@H-14
3	0	—	MP@ML	MP@ML	10	1 212 416	MP@ML
3	1	SNR	MP@ML	MP@ML	15	1 835 008	SNR@ML
3	2	空域	MP@H-14	MP@H-14	60	7 340 032	Spatial@H-14
3	0	—	MP@ML	MP@ML	15	1 835 008	MP@ML
3	1	空域	MP@H-14	MP@H-14	40	4 882 432	Spatial@H-14
3	2	SNR	MP@H-14	MP@H-14	60	7 340 032	Spatial@H-14

表 II -8-18 不同框架级别间的向前兼容

比特流	解码器											
	HP @ HL	HP @ H-14	HP @ ML	空域 @ H-14	SNR @ ML	SNR @ LL	MP @ HL	MP @ H-14	MP @ ML	MP @ LL	SP @ ML	
HP@HL	X											
HP@H-14	X	X										
HP@ML	X	X	X									
Spatial@H-14	X	X		X								
基层	X	X	X	X			X	X				
SNR@ML	X	X	X	X	X							
基层	X	X	X	X	X		X	X	X			
SNR@LL	X	X	X	X	X	X						
基层	X	X	X	X	X	X	X	X	X	X		
MP@HL	X						X					
MP@H-14	X	X		X			X	X				
MP@ML	X	X	X	X	X		X	X	X			
MP@LL	X	X	X	X	X	X	X	X	X	X	X*	
SP@ML	X	X	X	X	X		X	X	X		X	
ISO/IEC 11172	X	X	X	X	X	X	X	X	X	X	X	

X 表示这个解码器可以对这个比特流解码。  
\* 注意, SP@ML 解码器应能对 MP@LL 比特流解码。

注意 对于遵循等级结构的框架和级别, 我们建议比特流的每一层都应包含能够对该比特流的那一层成功解码的“最简单”解码器的 profile\_and\_level\_indication。如果 profile\_and\_level\_indication 的逸出位为 0, 则这将为 profile\_and\_level\_indication 的最大可能有效值。

## 附录 II -A 离散余弦变换

(这个附录是本推荐|国际标准必备的一部分)

$N \times N$  的二维 DCT 定义如下：

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

其中  $u, v, x, y = 0, 1, 2, \dots, N-1$ ；  
 $x, y$  是采样域的空间坐标；  
 $u, v$  是变换域的坐标。

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u, v = 0 \\ 1 & \text{其它} \end{cases}$$

反 DCT(IDCT) 定义如下：

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

正变换的输入和反变换的输出以 9 个比特位表示。DCT 系数以 12 个比特位表示。  
DCT 系数的变化范围是  $[-2048 : +2047]$ 。

$N \times N$  的反离散余弦变换与 IEEE 的  $8 \times 8$  反离散余弦变换标准规范 Std 1180-1990, 12 月 6 日, 1990 年相一致。注意, Std 1180-1990 2.3 节“关于 IDCT 不匹配误差的规定”需要规定定期的内部图编码来控制累积不匹配误差。这个标准所需的最大刷新周期是 132 幅图, 与按 ITU-T H.261 建议针对可视电话的 1180-1990 中指出的一致。(见附录 G)

## 附录 II -B 可变长度码表

(本附录为本建议|国际标准必备的一部分)

### II -B. 1 宏块寻址

表 II -B-1 macroblock\_address\_increment 的变长码

macroblock_address_increment VLC 码	增加值	macroblock_address_increment VLC 码	增加值
1	1	0000 0101 01	18
011	2	0000 0101 00	19
010	3	0000 0100 11	20
0011	4	0000 0100 10	21
0010	5	0000 0100 011	22
0001 1	6	0000 0100 010	23
0001 0	7	0000 0100 001	24
0000 111	8	0000 0100 000	25
0000 110	9	0000 0011 111	26
0000 1011	10	0000 0011 110	27
0000 1010	11	0000 0011 101	28
0000 1001	12	0000 0011 100	29
0000 1000	13	0000 0011 011	30
0000 0111	14	0000 0011 010	31
0000 0110	15	0000 0011 001	32
0000 0101 11	16	0000 0011 000	33
0000 0101 10	17	0000 0001 000	macroblock_escape

注意 在 ISO/IEC 11172-2 中的“宏块填充”入口在本规范中无效。

## II -B. 2 宏块类型

宏块属性可根据这些表的宏类型 VLC 决定。

**表 II -B-2 I-图中 macroblock-type 的变长码**

macroblock-type VLC 码						
macroblock-quant						
macroblock-motion-forward						
macroblock-motion-backward						
macroblock-pattern						
macroblock-intra						
spatial-temporal-weight-code-flag						
permitted spatial-temporal-weight-classes						
描述						
1	0	0	0	0	1	0
						帧内
01	1	0	0	0	1	0
						帧内量化
						0
						0

**表 II -B-3 P-图中 macroblock-type 的变长码**

macroblock-type VLC 码						
macroblock-quant						
macroblock-motion-forward						
macroblock-motion-backward						
macroblock-pattern						
macroblock-intra						
spatial-temporal-weight-code-flag						
permitted spatial-temporal-weight-classes						
描述						
1	0	1	0	1	0	0
						MC, 编码
01	0	0	0	1	0	0
						非 MC, 编码
001	0	1	0	0	0	0
						MC, 没有编码
0001 1	0	0	0	0	1	0
						帧内
0001 0	1	1	0	1	0	0
						MC, 编码量化
0000 1	1	0	0	1	0	0
						非 MC, 编码量化
0000 01	1	0	0	0	1	0
						帧内量化
						0
						0

表 II -B-4 B-图中 macroblock-type 的变长码

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
10	0	1	1	0	0	0	帧间预测, 没编码	0
11	0	1	1	1	0	0	帧间预测, 编码	0
010	0	0	1	0	0	0	后向预测, 没编码	0
011	0	0	1	1	0	0	后向预测, 编码	0
0010	0	1	0	0	0	0	前向预测, 没编码	0
0011	0	1	0	1	0	0	前向预测, 编码	0
0001 1	0	0	0	0	1	0	帧内	0
0001 0	1	1	1	1	0	0	帧间预测, 编码, 量化	0
0000 11	1	1	0	1	0	0	前向预测, 编码, 量化	0
0000 10	1	0	1	1	0	0	后向预测, 编码, 量化	0
0000 01	1	0	0	0	1	0	帧内量化	0

表 II -B-5 具有空域分级的 I-图中的 macroblock-type 的变长码

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
1	0	0	0	1	0	0	编码, 兼容	4
01	1	0	0	1	0	0	编码, 兼容量化	4
0011	0	0	0	0	1	0	帧内	0
0010	1	0	0	0	1	0	帧内量化	0
0001	0	0	0	0	0	0	非编码, 兼容	4

表 II -B-6 具有空域分级的 P-图中 macroblock-type 的变长码

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
10	0	1	0	1	0	0	MC, 编码	0
011	0	1	0	1	0	1	MC, 编码, 兼容	1,2,3
0000 100	0	0	0	1	0	0	非 MC, 编码	0
0001 11	0	0	0	1	0	1	非 MC, 编码, 兼容	1,2,3
0010	0	1	0	0	0	0	MC, 没编码	0
0000 111	0	0	0	0	1	0	帧内	0
0011	0	1	0	0	0	1	MC, 没编码, 兼容	1,2,3
010	1	1	0	1	0	0	MC, 编码, 量化	0
0001 00	1	0	0	1	0	0	非 MC, 编码, 量化	0
0000 110	1	0	0	0	1	0	帧内, 量化	0
11	1	1	0	1	0	1	MC, 编码, 兼容, 量化	1,2,3
0001 01	1	0	0	1	0	1	非 MC, 编码, 兼容, 量化	1,2,3
0001 10	0	0	0	0	0	1	非 MC, 没编码, 兼容	1,2,3
0000 101	0	0	0	1	0	0	编码, 兼容	4
0000 010	1	0	0	1	0	0	编码, 兼容, 量化	4
0000 011	0	0	0	0	0	0	非编码, 兼容	4

表 II -B-7 具有空域分级的 B-图中 macroblock-type 的变长码

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
10	0	1	1	0	0	0	帧间预测, 没编码	0
11	0	1	1	1	0	0	帧间预测, 编码	0
010	0	0	1	0	0	0	后向没编码	0
011	0	0	1	1	0	0	后向编码	0
0010	0	1	0	0	0	0	前向没编码	0
0011	0	1	0	1	0	0	前向编码	0

续表

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
0001 10	0	0	1	0	0	1	后向, 没编码, 兼容	1,2,3
0001 11	0	0	1	1	0	1	后向, 编码, 兼容	1,2,3
0001 00	0	1	0	0	0	1	前向, 非编码, 兼容	1,2,3
0001 01	0	1	0	1	0	1	前向, 编码, 兼容	1,2,3
0000 110	0	0	0	0	1	0	帧内	0
0000 111	1	1	1	1	0	0	帧间预测, 编码, 量化	0
0000 100	1	1	0	1	0	0	前向, 编码, 量化	0
0000 101	1	0	1	1	0	0	后向, 编码, 量化	0
0000 0100	1	0	0	0	1	0	帧内, 量化	0
0000 0101	1	1	0	1	0	1	前向, 编码, 兼容, 量化	1,2,3
0000 0110 0	1	0	1	1	0	1	后向, 编码, 兼容, 量化	1,2,3
0000 0111 0	0	0	0	0	0	0	非编码, 兼容	4
0000 0110 1	1	0	0	1	0	0	编码, 兼容, 量化	4
0000 0111 1	0	0	0	1	0	0	编码, 兼容	4

表 II-B-8 具有 SNR 分级的 I-图, P-图, B-图中的 macroblock-type 的变长码

macroblock-type VLC 码								
macroblock-quant								
macroblock-motion-forward								
macroblock-motion-backward								
macroblock-pattern								
macroblock-intra								
spatial-temporal-weight-code-flag								
permitted spatial-temporal-weight-classes								
描述								
1	0	0	0	1	0	0	编码	0
01	1	0	0	1	0	0	编码, 量化	0
001	0	0	0	0	0	0	非编码	0

注意 由于在 I, P 和 B-图中对宏块的处理一致, 所以各种类型的图没有差别。由于“跳过的宏块”不允许在组块的头和尾出现, 所以“未编码”类型是需要的。

## II -B. 3 宏块模式

表 II -B-9 coded\_block\_pattern 的变长码

coded_block_pattern VLC 码	cbp	coded_block_pattern VLC 码	cbp
111	60	0001 1100	35
1101	4	0001 1011	13
1100	8	0001 1010	49
1011	16	0001 1001	21
1010	32	0001 1000	41
1001 1	12	0001 0111	14
1001 0	48	0001 0110	50
1000 1	20	0001 0101	22
1000 0	40	0001 0100	42
0111 1	28	0001 0011	15
0111 0	44	0001 0010	51
0110 1	52	0001 0001	23
0110 0	56	0001 0000	43
0101 1	1	0000 1111	25
0101 0	61	0000 1110	37
0100 1	2	0000 1101	26
0100 0	62	0000 1100	38
0011 11	24	0000 1011	29
0011 10	36	0000 1010	45
0011 01	3	0000 1001	53
0011 00	63	0000 1000	57
0010 111	5	0000 0111	30
0010 110	9	0000 0110	46
0010 101	17	0000 0101	54
0010 100	33	0000 0100	58
0010 011	6	0000 0011 1	31
0010 010	10	0000 0011 0	47
0010 001	18	0000 0010 1	55
0010 000	34	0000 0010 0	59
0001 1111	7	0000 0001 1	27
0001 1110	11	0000 0001 0	39
0001 1101	19	0000 0000 1	0(NOTE)

注意 这个表不能和 4:2:0 的色差结构一同使用。

## II -B. 4 运动向量

表 I -B-10 motion\_code 的变长码

变长码	motion_code[r][s][t]	变长码	motion_code[r][s][t]
0000 0011 001	-16	010	1
0000 0011 011	-15	0010	2
0000 0011 101	-14	0001 0	3
0000 0011 111	-13	0000 110	4
0000 0100 001	-12	0000 1010	5
0000 0100 011	-11	0000 1000	6
0000 0100 11	-10	0000 0110	7
0000 0101 01	-9	0000 0101 10	8
0000 0101 11	-8	0000 0101 00	9
0000 0111	-7	0000 0100 10	10
0000 1001	-6	0000 0100 010	11
0000 1011	-5	0000 0100 000	12
0000 111	-4	0000 0011 110	13
0001 1	-3	0000 0011 100	14
0011	-2	0000 0011 010	15
011	-1	0000 0011 000	16
1	0		

表 I -B-11 dmvector[t]的变长码

编码	值
11	-1
0	0
10	1

## II -B. 5 DCT 系数

表 I -B-12 dct\_dc\_size\_luminance 的变长码

变长码	dct_dc_size_luminance
100	0
00	1
01	2
101	3
110	4
1110	5
1111 0	6
1111 10	7
1111 110	8
1111 1110	9
1111 1111 0	10
1111 1111 1	11

表 I -B-13 dct\_dc\_size\_chrominance 的变长码

变长码	dct_dc_size_chrominance
00	0
01	1
10	2
110	3
1110	4
1111 0	5
1111 10	6
1111 110	7
1111 1110	8
1111 1111 0	9
1111 1111 10	10
1111 1111 11	11

表 II-B-14 DCT 系数表 0

变长码(注释 1)	行 程	层 次
10	块结束	
1 s(注释 2)	0	1
11 s(注释 3)	0	1
011 s	1	1
0100 s	0	2
0101 s	2	1
0010 1 s	0	3
0011 1 s	3	1
0011 0 s	4	1
0001 10 s	1	2
0001 11 s	5	1
0001 01 s	6	1
0001 00 s	7	1
0000 110 s	0	4
0000 100 s	2	2
0000 111 s	8	1
0000 101 s	9	1
0000 01	换码	
0010 0110 s	0	5
0010 0001 s	0	6
0010 0101 s	1	3
0010 0100 s	3	2
0010 0111 s	10	1
0010 0011 s	11	1
0010 0010 s	12	1
0010 0000 s	13	1
0000 0010 10 s	0	7
0000 0011 00 s	1	4
0000 0010 11 s	2	3
0000 0011 11 s	4	2
0000 0010 01 s	5	2
0000 0011 10 s	14	1
0000 0011 01 s	15	1
0000 0010 00 s	16	1
0000 0001 1101 s	0	8
0000 0001 1000 s	0	9
0000 0001 0011 s	0	10
0000 0001 0000 s	0	11

注释 1——最后一位‘s’表示层的符号，‘0’为正，‘1’为负。  
注释 2——对于块中的第一个(DC)系数不能使用这个码。  
注释 3——对于所有其它系数均不能使用这个码。

续表

变长码(注释1)	行 程	层 次
0000 0001 1011 s	1	5
0000 0001 0100 s	2	4
0000 0001 1100 s	3	3
0000 0001 0010 s	4	3
0000 0001 1110 s	6	2
0000 0001 0101 s	7	2
0000 0001 0001 s	8	2
0000 0001 1111 s	17	1
0000 0001 1010 s	18	1
0000 0001 1001 s	19	1
0000 0001 0111 s	20	1
0000 0001 0110 s	21	1
0000 0000 1101 0 s	0	12
0000 0000 1100 1 s	0	13
0000 0000 1100 0 s	0	14
0000 0000 1011 1 s	0	15
0000 0000 1011 0 s	1	6
0000 0000 1010 1 s	1	7
0000 0000 1010 0 s	2	5
0000 0000 1001 1 s	3	4
0000 0000 1001 0 s	5	3
0000 0000 1000 1 s	9	2
0000 0000 1000 0 s	10	2
0000 0000 1111 1 s	22	1
0000 0000 1111 0 s	23	1
0000 0000 1110 1 s	24	1
0000 0000 1110 0 s	25	1
0000 0000 1101 1 s	26	1
0000 0000 0111 11 s	0	16
0000 0000 0111 10 s	0	17
0000 0000 0111 01 s	0	18
0000 0000 0111 00 s	0	19
0000 0000 0110 11 s	0	20
0000 0000 0110 10 s	0	21
0000 0000 0110 01 s	0	22
0000 0000 0110 00 s	0	23
0000 0000 0101 11 s	0	24
0000 0000 0101 10 s	0	25
0000 0000 0101 01 s	0	26
0000 0000 0101 00 s	0	27

注释——最后一位‘s’表示层的符号,‘0’为正,‘1’为负。

续表

变长码(注释1)	行 程	层 次
0000 0000 0100 11 s	0	28
0000 0000 0100 10 s	0	29
0000 0000 0100 01 s	0	30
0000 0000 0100 00 s	0	31
0000 0000 0011 000 s	0	32
0000 0000 0010 111 s	0	33
0000 0000 0010 110 s	0	34
0000 0000 0010 101 s	0	35
0000 0000 0010 100 s	0	36
0000 0000 0010 011 s	0	37
0000 0000 0010 010 s	0	38
0000 0000 0010 001 s	0	39
0000 0000 0010 000 s	0	40
0000 0000 0011 111 s	1	8
0000 0000 0011 110 s	1	9
0000 0000 0011 101 s	1	10
0000 0000 0011 100 s	1	11
0000 0000 0011 011 s	1	12
0000 0000 0011 010 s	1	13
0000 0000 0011 001 s	1	14
0000 0000 0001 0011 s	1	15
0000 0000 0001 0010 s	1	16
0000 0000 0001 0001 s	1	17
0000 0000 0001 0000 s	1	18
0000 0000 0001 0100 s	6	3
0000 0000 0001 1010 s	11	2
0000 0000 0001 1001 s	12	2
0000 0000 0001 1000 s	13	2
0000 0000 0001 0111 s	14	2
0000 0000 0001 0110 s	15	2
0000 0000 0001 0101 s	16	2
0000 0000 0001 1111 s	27	1
0000 0000 0001 1110 s	28	1
0000 0000 0001 1101 s	29	1
0000 0000 0001 1100 s	30	1
0000 0000 0001 1011 s	31	1

注释——最后一位‘s’表示层的符号,‘0’为正,‘1’为负。

表 I-B-15 DCT 系数表 1

变长码(注释)	行 程	层 次
0110	块结尾	
10 s	0	1
010 s	1	1
110 s	0	2
0010 1 s	2	1
0111 s	0	3
0011 1 s	3	1
0001 10 s	4	1
0011 0 s	1	2
0001 11 s	5	1
0000 110 s	6	1
0000 100 s	7	1
1110 0 s	0	4
0000 111 s	2	2
0000 101 s	8	1
1111 000 s	9	1
0000 01	换码	
1110 1 s	0	5
0001 01 s	0	6
1111 001 s	1	3
0010 0110 s	3	2
1111 010 s	10	1
0010 0001 s	11	1
0010 0101 s	12	1
0010 0100 s	13	1
0001 00 s	0	7
0010 0111 s	1	4
1111 1100 s	2	3
1111 1101 s	4	2
0000 0010 0 s	5	2
0000 0010 1 s	14	1
0000 0011 1 s	15	1
0000 0011 01 s	16	1
1111 011 s	0	8
1111 100 s	0	9
0010 0011 s	0	10
0010 0010 s	0	11
0010 0000 s	1	5
0000 0011 00 s	2	4
0000 0001 1100 s	3	3

注释——最后一位‘s’表示层的符号,‘0’为正,‘1’为负

续表

变长码(注释)	行 程	层 次
0000 0001 0010 s	4	3
0000 0001 1110 s	6	2
0000 0001 0101 s	7	2
0000 0001 0001 s	8	2
0000 0001 1111 s	17	1
0000 0001 1010 s	18	1
0000 0001 1001 s	19	1
0000 0001 0111 s	20	1
0000 0001 0110 s	21	1
1111 1010 s	0	12
1111 1011 s	0	13
1111 1110 s	0	14
1111 1111 s	0	15
0000 0000 1011 0 s	1	6
0000 0000 1010 1 s	1	7
0000 0000 1010 0 s	2	5
0000 0000 1001 1 s	3	4
0000 0000 1001 0 s	5	3
0000 0000 1000 1 s	9	2
0000 0000 1000 0 s	10	2
0000 0000 1111 1 s	22	1
0000 0000 1111 0 s	23	1
0000 0000 1110 1 s	24	1
0000 0000 1110 0 s	25	1
0000 0000 1101 1 s	26	1
0000 0000 0111 11 s	0	16
0000 0000 0111 10 s	0	17
0000 0000 0111 01 s	0	18
0000 0000 0111 00 s	0	19
0000 0000 0110 11 s	0	20
0000 0000 0110 10 s	0	21
0000 0000 0110 01 s	0	22
0000 0000 0110 00 s	0	23
0000 0000 0101 11 s	0	24
0000 0000 0101 10 s	0	25
0000 0000 0101 01 s	0	26
0000 0000 0101 00 s	0	27
0000 0000 0100 11 s	0	28
0000 0000 0100 10 s	0	29
0000 0000 0100 01 s	0	30

注释——最后一位‘s’表示层的符号,‘0’为正,‘1’为负

续表

变长码(注释)	行 程	层 次
0000 0000 0100 00 s	0	31
0000 0000 0011 000 s	0	32
0000 0000 0010 111 s	0	33
0000 0000 0010 110 s	0	34
0000 0000 0010 101 s	0	35
0000 0000 0010 100 s	0	36
0000 0000 0010 011 s	0	37
0000 0000 0010 010 s	0	38
0000 0000 0010 001 s	0	39
0000 0000 0010 000 s	0	40
0000 0000 0011 111 s	1	8
0000 0000 0011 110 s	1	9
0000 0000 0011 101 s	1	10
0000 0000 0011 100 s	1	11
0000 0000 0011 011 s	1	12
0000 0000 0011 010 s	1	13
0000 0000 0011 001 s	1	14
0000 0000 0001 0011 s	1	15
0000 0000 0001 0010 s	1	16
0000 0000 0001 0001 s	1	17
0000 0000 0001 0000 s	1	18
0000 0000 0001 0100 s	6	3
0000 0000 0001 1010 s	11	2
0000 0000 0001 1001 s	12	2
0000 0000 0001 1000 s	13	2
0000 0000 0001 0111 s	14	2
0000 0000 0001 0110 s	15	2
0000 0000 0001 0101 s	16	2
0000 0000 0001 1111 s	27	1
0000 0000 0001 1110 s	28	1
0000 0000 0001 1101 s	29	1
0000 0000 0001 1100 s	30	1
0000 0000 0001 1011 s	31	1

注释——最后一位‘s’表示层的符号,‘0’为正,‘1’为负。

表 I -B-16 跟在“换码”码后面的行程和层次的编码

定长码	行 程	定长码	signed_level
0000 00	0	1000 0000 0001	-2047
0000 01	1	1000 0000 0010	-2046
0000 10	2	...	...
...	...	1111 1111 1111	-1
...	...	0000 0000 0000	禁止
...	...	0000 0000 0001	+1
...	...	...	...
1111 11	63	0111 1111 1111	+2047

## 附录 II -C 视频缓冲检验器

(这个附录是本推荐|国际标准必备的一部分)

常数速率的编码视频比特流应满足本附录所定义的视频缓冲检验器(VBV)所加的约束。在变化的比特率的处理中,本附录以 ISO/IEC 13818-1 中定义的 STD 模型替换。如果 ISO/IEC 13818-1 中定义的系统编码层存在,则以 STD 模型代替 VBV 模型。

VBV 是一个假设的解码器,在概念上与编码器的输出相连接。编码的数据按正在使用的常数比特率被放置于缓冲区中。编码的数据按下面所述的方法从缓冲区内移出。与本规范相一致的比特流应不会导致 VBV 溢出。如果 low\_delay 等于 0,比特流不应导致 VBV 缓冲区下溢。如 low\_delay 等于 1,如 I-C.7 和 I-C.8 所述,VBV 缓冲区有可能下溢。

附录 I -C 中的所有计算都以实际值完成,因此不会传播取整误差。例如,VBV 缓冲区中位的数目不必为整数。

1. VBV 和视频编码器具有相同的时钟频率,相同的帧率,实行同步操作。
2. VBV 有一个大小为 B 的输入缓冲区,B 是在序列头和序列扩展(如果有的话)中编码的 vbv\_buffer\_size。
3. VBV 初始化为空。在将输入缓冲区中填入序列的第一幅图的起始码以前的所有数据和起始码本身之后,输入缓冲区在图象头中的 vbv\_delay 域所规定的时间内从比特流中填入数据。
4. 从这个时间开始,VBV 缓冲区在附录 C 中 9~12 定义的后续时间被检测。5~88 定义了在 VBV 每次被检测时所采取的动作。
5. 这一段定义了针对所有视频流的一个要求。在移去任何图象数据之前和恰好在这个图象的数据被移去之后,当 VBV 缓冲区被检测时,缓冲区中位的数目应在 0 和 B 之间,其中 B 是 VBV\_buffer\_size 所指出的 VBV 的大小。本附录,图象数据包含 picture\_data(),各种头,在其之前的用户数据和填充位,如果在 VBV 缓冲区被检测时出现了紧跟在 picture\_data()之后的跟踪填充位和字节,则它们也被包括在其中。对于视频序列中的第一个编码图象,刚好位于 sequence\_header()之前的任何零填充位或字节也包括在图象数据中。

为适应这个要求,下面的不等式必须满足:

$$d_{n+1} > B_n + (t_{n+2} - t_n) * R - B$$

这个不等式采用实际值计算(参考图 I -C-1)。

其中  $d_n$  是第 n 个编码图的图象数据,以位表示;

$B_n$  是在第 n 个编码图刚好从缓冲区中被移走之后,缓冲区的占用情况(以位计算);

R 是以 bit/s 表示的比特率。VBV 模型中的编码器应使用全精度的比特率,而不是

以 bit\_rate 和 bit\_rate\_extension 域编码的取整值；  
 $t_n$  是第  $n$  个编码图从 VBV 缓冲区中被移走的时刻(以秒表示,采用全精度)。

```
sequence_pheader( ), sequence_extension, extension_and_user_data(0),
group_of_picture_header( ) and extension_and_user_data(1)
```

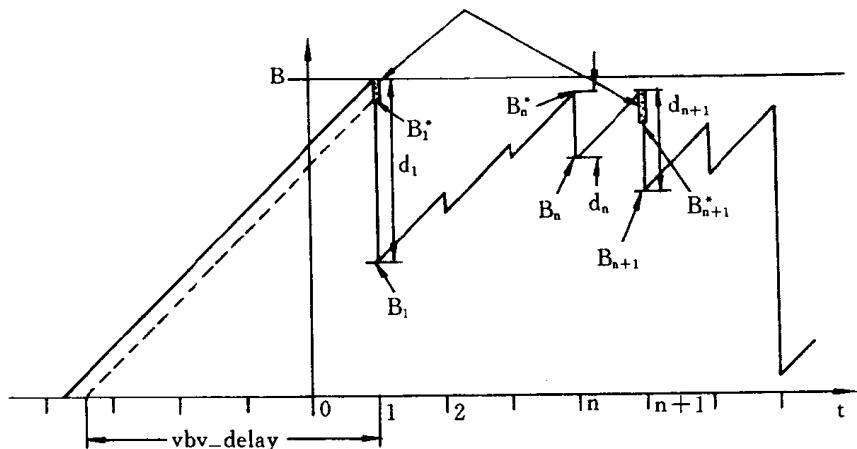


图 I-C-1 VBV 缓冲区占有情况

6. 这一段定义了当 low\_delay 标志等于 0 时,对视频比特流的要求。

在任何位被移走之前,每次 VBV 缓冲区被检测时,在缓冲区中时间最久的图象的所有数据都应在缓冲区中。

为了满足这个要求,必须满足下面的不等式:

$$d_n \leq B_n^*$$

其中  $d_n$  是第  $n$  个编码图的图象数据,以位表示;

$B_n^*$  是第  $n$  个编码图被从缓冲区中移走之前缓冲区的占用情况(以位表示)。

当 low\_delay 标志等于 0 时,VBV 缓冲区下溢(图象中所有数据都不在时发生)不应发生。

7. 仅当 low\_delay 标志为 1 时,才用到这一段。这一段描述了当 VBV 缓冲区被检测,而它未包含一个完整的图象时,即发生了下溢,VBV 所应采取的动作。

这是当下面的不等式被满足时的情况:

$$d_n > B_n^*$$

其中  $d_n$  是以位表示的第  $n$  个编码图的图象数据;

$B_n^*$  是第  $n$  个编码图从缓冲区中移走之前缓冲区的占有情况(以位表示)。

上述的过程可用于:

VBV 缓冲区以两个场周期的时间间隔被重复检测,直到完整的图象数据出现于 VBV 输入缓冲区中。在这种情况下,缓冲区中的位数必须少于  $B$ 。在那时,VBV 的正常操作恢复,如 5 所述。

当这一段起作用时,不影响 temporal\_reference 的值。如,当没有 B 帧图象时,不管 VBV 缓冲区是否会下溢,对每一帧 temporal\_reference 都加 1。

## 8. 这一段只是非正式的。

当低延迟的应用系统偶尔必须传送大的图象,如在场景切换时,VBV 缓冲区下溢可能会发生。

对这样的比特流解码可能会导致与解码器相联系的显示过程重复以前的解码场或帧,直到 VBV 的正常操作可以恢复。这个过程有时被称为出现了“跳过的图”。注意,这种情形在正常情况下不会出现,除非偶然,并且仅在低延迟的比特流中出现(即,当 low\_delay 等于 1 时)。

9. 这一段定义了在 progressive\_sequence 等于 1 且 low\_delay 等于 0 时,对 VBV 缓冲区的连续检测的时间间隔。此时,帧重排延迟一般存在,而且可能出现 B 图。

VBV 输入缓冲区的两次连续检测之间的时间间隔  $t_{n+1} - t_n$  是 T 的倍数,T 为帧速率的倒数。

如果第 n 幅图是 repeat\_first\_field 等于 0 的 B-图,则  $t_{n+1} - t_n$  等于 T。

如果第 n 幅图是 repeat\_first\_field 等于 1,top\_field\_first 等于 0 的 B-图,则  $t_{n+1} - t_n$  等于  $2 * T$ 。如果第 n 幅图是 repeat\_first\_field 等于 1,top\_field\_first 等于 1 的 B-图,则  $t_{n+1} - t_n$  等于  $3 * T$ 。如果第 n 幅图是 P-图或 I-图,且如果以前的 P-图或 I-图的 repeat-first-field 等于 0,则  $t_{n+1} - t_n$  等于 T。

如果第 n 幅图是 P-图或 I-图,且如果以前的 P-图或 I-图的 repeat-first-field 等于 1,且 top\_field\_first 等于 0,则  $t_{n+1} - t_n$  等于  $2 * T$ 。

如果第 n 幅图是 P-图或 I-图,且如果以前的 P-图或 I-图的 repeat-first-field 等于 1,top\_field\_first 等于 1,则  $t_{n+1} - t_n$  等于  $3 * T$ 。

如果由于以前的 P-图或 I-图并不存在(在序列的开始时可能发生)而不能由上面任一部分决定  $t_{n+1} - t_n$ ,那么在下面的限制下,时间间隔可为任意值:

移去某帧(或某帧的第一场)和移去下一帧的时间间隔可能为 T,2 \* T 或 3 \* T 中的任意值。`vbv_delay` 的值可以被用来决定使用哪个值。

10. 这一段定义了在 progressive\_sequence 等于 1 且 low\_delay 等于 1 时,VBV 缓冲区的连续检测之间的时间间隔。此时,这个序列不包含 B-图,且没有帧重排延迟。

VBV 输入缓冲区的两个连续检测之间的时间间隔  $t_{n+1} - t_n$  是 T 的倍数,T 为帧速率的倒数。如果第 n 幅图是 P-图或 I-图,且 repeat-first-field 等于 0,则  $t_{n+1} - t_n$  等于 T。

如果第 n 幅图是 P-图或 I-图,repeat-first-field 等于 1 且 top\_field\_first 等于 0,则  $t_{n+1} - t_n$  等于  $2 * T$ 。

如果第 n 幅图是 P-图或 I-图,repeat-first-field 等于 0 且 top\_field\_first 等于 1,则  $t_{n+1} - t_n$  等于  $3 * T$ 。

11. 这一段定义了当 progressive\_sequence 等于 0 且 low\_delay 等于 0 时,对 VBV 缓冲区的两次检测的时间间隔。此时,通常存在帧重排延迟,并且可能出现 B 图。

对 VBV 输入缓冲区的两次连续检测的时间间隔  $t_{n+1} - t_n$  是 T 的倍数,T 是帧速率的两倍的倒数。

如果第 n 幅图是按帧结构编码的 repeat\_first\_field 为 0 的 B-帧,则  $t_{n+1} - t_n$  等于  $2 * T$ 。

如果第  $n$  幅图是按帧结构编码的 repeat-first-field 等于 1 的 B-帧，则  $t_{n+1} - t_n$  等于  $3 * T$ 。

如果第  $n$  幅图是场结构的 B-图(B-场图)，则  $t_{n+1} - t_n$  等于  $T$ 。

如果第  $n$  幅图是帧结构编码的 P-帧或 I-帧，且以前的编码 P-帧或 I-帧的 repeat-first-field 等于 1，则  $t_{n+1} - t_n$  等于  $2 * T$ 。

如果第  $n$  幅图是帧结构编码的 P-帧或 I-帧，且以前的编码 P-帧或 I-帧的 repeat-first-field 等于 1，则  $t_{n+1} - t_n$  等于  $3 * T$ 。

如果第  $n$  幅图是场结构编码的 P-帧或 I-帧的首场，则  $t_{n+1} - t_n$  等于  $T$ 。

如果第  $n$  幅图是按场结构编码的 P-帧或 I-帧的第二场，且以前的编码 P-帧或 I-帧正使用场结构或其 repeat first field 等于 0，则  $t_{n+1} - t_n$  等于  $(2 * T - T)$ 。

如果第  $n$  幅图是按场结构编码的 P-帧或 I-帧的第二场，且以前的编码 P-帧或 I-帧正使用帧结构或其 repeat\_first\_field 为 1，则  $t_{n+1} - t_n$  等于  $(3 * T - T)$ 。

如果由于不存在以前的编码 P-帧或 I-帧(比如在序列的起始处)，而使  $t_{n+1} - t_n$  不能由上面的各部分得出，则时间间隔在下面的这些限制的前提下可取任意值：

移去一帧(或某帧的首场)和下一帧之间的时间间隔可在  $2 * T$  或  $3 * T$  中任选一个，`vbv_delay` 的值可被用来决定选哪一个。

图 II-C-2 示出了在只有帧-图时的 VBV 的一种简单情况。帧  $P_0, B_1$  和  $B_4$  的显示时间为 3 个场周期。

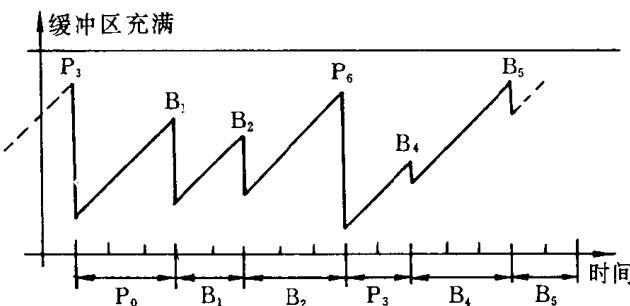


图 II-C-2 只有 P,B 图时 VBV 的一种简单情况

12. 这一节定义了当 progressive\_sequence 等于 0 且 low\_delay 等于 1 时对 VBV 缓冲区两次检测的时间间隔。此时，序列不包含 B-图，且没有帧重排延迟。

对 VBV 缓冲区的连续两次检测的时间间隔  $t_{n+1} - t_n$  为  $T$  的整数倍， $T$  是帧速率的两倍的倒数。

如果第  $n$  幅图是帧结构编码的 P-帧或 I-帧，且 repeat\_first\_field 等于 0，则  $t_{n+1} - t_n$  等于  $2 * T$ ，如果第  $n$  幅图是帧结构编码的 P-帧或 I-帧，且 repeat\_first\_field 等于 1，则  $t_{n+1} - t_n$  等于  $3 * T$ 。

如果第  $n$  幅图是按场结构编码的 P-帧或 I-帧，则  $t_{n+1} - t_n$  等于  $T$ 。

图 II-C-3 示出了只有帧图时的 VBV 的一种简单情况。帧  $I_0, P_2$  和  $P_4$  的 repeat\_first\_field 均等于 1。

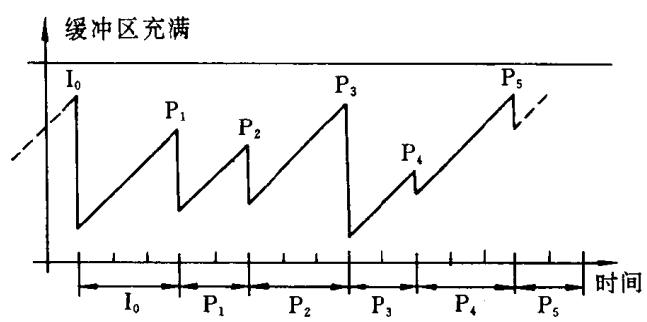


图 II - C - 3 只有 I, P 图时 VBV 的一种简单情况

## 附录 II -D 算法所支持的性能

(本附录不作为本推荐|国际标准必备的一部分)

### II -D. 1 概述

本规范包含下面简述的性能：

- (1) 能表示不同的采样格式(即 4：2：0, 4：2：2 和 4：4：4)。
- (2) 隔行和逐行的扫描格式的视频信号均可被编码。
- (3) 解码器应能以 3：2 下拉方式用帧速率近似为 30fps 的视频来表示帧速率近似为 24fps 的影片。
- (4) 能在一个大屏面内用可移动的全景扫描窗口来选择显示区。
- (5) 可以使用在很大范围内变化的各种质量的图象。
- (6) 支持常量和变量比特率通道。
- (7) 可获得针对“面对面”交互系统的低延迟模式。
- (8) 可获得随机访问(针对 DSM, 信道获取, 信道跳换)。
- (9) 能对 ISO/IEC 11172-2 比特流解码。
- (10) 能产生针对不同程度复杂性(硬件)的解码器的比特流。
- (11) 支持对编码视频的编辑。
- (12) 能实现对录制的比特流的快进和快速回放。
- (13) 编码的比特流能对错误复原。

### II -D. 2 视频格式

#### II -D. 2. 1 采样格式和颜色

本规范的视频编码支持逐行和隔行视频, 以在序列扩展编码中传输的 progressive-sequence 标志来指出使用哪种格式。

允许的屏面大小为: 水平和竖直方向亮度样本在 1 到  $(2^{14}-1)$  之间。视频以可选元色的亮度/色差空间表示。色差可以 4：2：0 采样(在水平方向和竖直方向均为亮度样本的一半), 或以 4：2：2 采样(仅在水平方向为一半)。此外, 实际应用中的特定样本高宽比和图象高宽比可灵活变化。在序列扩展编码中包含 chroma\_format 参数。

样本高宽比信息通过 sequence\_display\_extension() 中的 aspect\_ratio\_information 和(可选的)display\_horizontal\_size 提供。与 ITU-R Rec. 601 一致的信号采样的合适值之例在表 I -D-1 中给出。

表 I -D-1 显示尺寸大小值举例

信号格式	display_horizontal_size	display_vertical_size
525-行	711	483
625-行	702	575

为了将来可能用到,本规范实现了支持 4 : 4 : 4 色差格式的方法。然而,目前在任何框架中均未对此支持。

## II -D. 2. 2 影片定时

当对一个逐行图象序列进行编码时,解码器能实现 3 : 2 下拉。每幅编码的影片图象可以独立地规定其显示时间为两个或三个视频场周期,所以“非常规”的 3 : 2 下拉源素材可以逐行视频形式传输。两个标志:top\_field\_first 和 repeat\_first\_field 随图象编码扩展一同传输,足以描述必要的显示时间。

## II -D. 2. 3 显示格式控制

显示过程将一系列数字化的帧(对逐行视频来讲)或场(对隔行视频来讲)转化为输出的视频。它不是本标准的正式部分。本规范的视频语法确实传递用于重构视频的某些显示参数。可选的信息(在序列显示扩展中)规定了色度、显示基色、光电传输特性(例如,γ 值)以及 RGB 到亮度/色差的转换矩阵。

此外,编码屏面内的显示窗口可以被定义为全景扫描,或者,编码屏面可以被定义为一个大面积显示设备上的一个窗口。对于全景扫描来讲,表示一幅更大的图象的显示区域的窗口的位置可以在一场一场的基础上定义,这在 I -6. 3. 12 中描述的图象显示扩展中规定。全景扫描窗口的一个典型的应用就是在 16 : 9 的视频序列中定义“重要的”4 : 3 高宽比的矩形。类似地,对于大的显示区上的小的编码图象,可以规定显示区的大小和显示区上的窗口位置。

## II -D. 2. 4 全电视信号的透明编码

对于像组合和分离这样的较高质量的应用,全电视信号源传输之前由 PAL/NTSC 解码的和传输之后再编码为 PAL/NTSC 的,均需要对载波振幅和场参考信号(以及 PAL 的 V-轴开关)进行精确重构。

输入格式可在序列头中用 video\_format 位指出。可能的信号源格式为:PAL, NTSC, SECAM 和 MAC。可以使用载波参数来重构载波信号:V\_axis, field\_sequence, sub\_carrier, burst\_amplitude 和 sub\_carrier\_phase, 通过置图象头中的 composite\_display\_flag 来对这些参数使能。

## II -D. 3 图象质量

图象的高质量与使用的比特率有关。与某一特定框架中的确定级别有关的足够高的

比特率限制可以提供高质量的图象质量。高质量的色差频带可通过使用 4：2：2 色差格式来实现。

通过加载和使用附带一个“量化比例编码”的量化矩阵可以实现几乎无损的编码。

此外，具有灵活可变的比特率的分级编码为质量分级和降低质量但保持可靠性提供了可能。例如，对于代表低分辨率图象的比特流的子集进行解码，就可以实现在造价低的接收器上以相应质量进行解码；而对整个比特流进行解码则可获得完整的高质量图象。

此外，可以通过使用低帧速率（通过编码前的预处理或跳过由 temporal\_reference 指出的帧）和低的空间分辨率来适应基于低比特率的操作。

## II -D. 4 数据率控制

在很大范围内可选的单位时间传输的比特数可以通过两种方法控制，并均受到本规范的支持。有一个 bit\_rate 描述符与序列头编码一同传输。

对于常数比特率(CBR)编码，在信道上单位时间传输的位数是一个常数。由于对不同内容的图象，编码输出数据率通常是变化的，所以应用缓冲等方法来调整速率常数。在 CBR 中，图象质量根据其内容而变化。

另一种模式是可变比特率(VBR)编码，对它来讲，在一些限制之下，信道上的单位时间传送位数是变化的。VBR 是打算用来提供固定质量编码的。VBR 的一个应用模型是在 B-ISDN 信道上服从“参数控制应用”(UPC)的近似固定质量编码。

## II -D. 5 低延迟模式

低延迟的编码和解码模式适用于像可视化技术、视频会议、监控这样的实时视频通信。对于本规范的低延迟模式操作，可以实现少于 150 毫秒延迟的全部编码和解码。通过对序列头编码中的 low\_delay\_flag 置位可以定义低延迟比特流。

可以通过产生不包含 B-图的比特流来使全部的编码和解码过程保持低延迟。这避免了帧重排延迟。通过使用针对 P-帧的双基预测，仍能保证图象的高质量。

低延迟需要在编码和解码时有较少的缓冲区占用时间。编码器应避免大型的编码图象。通过对每帧的一个或多个组块，而非对内部帧进行修正，可以适用这一要求。

对于低延迟操作，如果超过了每帧所需的比特数，则编码器能够跳过一个或多个帧。解码器通过 VBV 缓冲区或 STD 缓冲区的状态来得知进行了这种跳过，即，解码器缓冲区的下溢指明了编码器跳过了一些图。

## II -D. 6 随机访问/信道跳换

本规范的语法支持随机访问和信道跳换。通过将合适的随机访问点编入比特流，可在不产生严重图象质量失真的情况下，获得足够的随机访问和信道跳换功能。

对于存储媒体上的视频信号,随机访问是一个基本特性。它要求任何图象均可在一段有限时间内被访问并被解码。这意味着在比特流中存在着随机访问点——那是一段可识别的信息,并且可以不依赖于其它数据段而被独立解码。在本规范中,随机访问点由 sequence\_header() 提供,并且后面跟有帧内信息(不需访问前面已解码的图而可以独立解码的图象数据)。每秒钟内有两段随机访问点的时间间隔将不会对图象质量产生严重影响。

在诸如广播这种传输应用中,信道跳换具有类似的情形。一旦选定了一个新的信道且解码器已获得了这个选定信道的比特流,则需要用上面描述的方法发现下一个数据入口,即随机访问点,并开始对新的段进行解码。

## II -D. 7 分 级

本规范的语法支持比特流分级。为了适应本规范所设想的应用所需的各种功能,开发了一些比特流分级工具:

- SNR 分级 主要是为了满足降低性能,但保证系统可靠性的需要。
- 色度同时联播 以高色差质量应用为目的。
- 数据分割 主要以 ATM 网中的单元损失恢复为目的。
- 时域分级 对于使用高时域分辨率逐行视频格式服务的交互工作,这是一种合适的方法。也适于在信道出现故障时,在适度降低系统性能的同时,很好地保证系统的可靠性。
- 空域分级 使适于视频服务交互工作应用系统的多分辨率编码技术成为可能。它也能提供在较低层上,实现与现存编码标准,即 ISO/IEC 11172-2 兼容的编码方法。

### II -D. 7. 1 SNR 分级在单一空域分辨率中的应用

SNR 分级的主要目标是为双层服务的传输提供一种方法,这两个层提供了相同的图象分辨率,但不同的质量级别,例如,具有两个不同质量级别的服务的传输有望在将来的 TV 广播应用中起到作用,特别是当大型显示接收器需要非常好的图象质量的时候。序列被编码为两个比特流,一个称为低层比特流,一个称为增强层比特流。低层比特流可以不需要增强层比特流而单独解码。在 3~4 Mbit/s 的比特率时,低层能提供与现在 NTSC/PAL/SECAM 等价的图象质量。这样,同时使用低层和增强层比特流,增强的解码器就可以在整个 7~12 Mbit/s 的比特率下传送接近于演播级质量水平的图象。

#### I -D. 7. 1. 1 附加特性

##### 1. 错误复原

如 I -D. 12 中所述,SNR 分级可以作为复原错误的一种方法。如果接收到具有不同错误率的比特流,当增强层被破坏时,我们只有求助于受到较好保护的低层,它是一个很好的替代。

##### 2. 色度联播

SNR 分级语法可以用于色度联播系统,它为具有同样的亮度分辨率但不同的色差采样格式(即低层中为 4 : 2 : 0,加上增强层和色差分量联播为 4 : 2 : 2)的服务的同时分配提供了一种手段。SNR 分级的增强层中包含一些亮度修正。4 : 2 : 2 色差以联播方式传送。只有色差 DC 由低层中预测。双层亮度和 4 : 2 : 2 色差的组合形成了高级别质量。

#### I-D. 7. 1. 2 SNR 分级编码过程

##### 1. 描述

在低层中,对于非分级情况,依据描述、自适应量化、缓冲器管理等编码过程是类似的,内部的或误差预测宏块进行 DCT 变换,然后使用一个初步的相当粗糙的量化器对这些系数进行量化。量化的系数再被 VLC 编码,并和一些必须的附带信息(macroblob-type,运动向量,coded\_block\_pattern())一起发送。

与此同时,来自低层的量化 DCT 系数被反量化。系数和反量化系数间的残余误差使用第二个较好的量化器进行再量化。结果的修正系数被 VLC 编码,然后和一些数量受限的附带信息(quantiser\_scale\_code,coded\_block\_pattern()...)一起形成附加的增强层。由于传输的信号具有差分属性,所以对增强层中的所有系数均使用非内部 VLC 表。

##### 2. 一些重要的评论

由于对于两层来说预测是相同的,所以我们建议在运动估测循环中使用修整过的图象(即由低层和增强层联合得到的图象)。这样,在编码器这一端的预测信号和低层解码器得到的用于预测的信号间有一些差异。从 P-图到 P-图,这个偏差不断积累,遇到 I-图时才被清零。然而,大约每隔 15 幅图便可发现这种偏差在图象中的微小影响。

由于增强层仅包含修整后的系数,所以所需的开销大大减少;有关宏块的大部分信息(宏块类型,运动向量...)包含在低层中,因此比特流的语法大大简化了:

(1) 宏块类型表仅需指明增强层中的 quantiser\_scale\_code 是否改变了,以及宏块是否为“非编码”的(对于组块的第一个和最后一个宏块),这些总共需要三个 VLC 码字。

(2) 如果值改变了,则需传输增强层中的 quantiser\_scale\_code。

(3) 对于所有的编码宏块传输 coded\_block\_pattern(),不在组块头或尾的所有“非编码”的宏块被“跳过”,这是由于开销信息可以从低层中推求出。

我们建议对于低层和增强层使用不同的加权矩阵。当第一次量化幅度比第二次量化要大时,可以获得一些更好的效果。然后为了避免颤动效果,我们建议不要对与交错运动有关的 DCT 系数进行太粗糙的量化。

#### I-D. 7. 2 使用 SNR 分级的多分辨率分级比特流

分辨率分级的目的是为了对适于在降低的空域分辨率下显示的基层视频进行解码。此外,我们需要实现一个能满足这一目标而具有较低程度复杂性的解码器。对于既不能也不愿显示由双层次支持的完整空域分辨率的接收器显示设备以及以软件解码为目的的应用,这种功能是有用处的。本节描述的方法使用 I-7 章中所述的 SNR 分级语法,在两个层次上传送视频信号。值得注意的是,本节中所建议的各种选择都不会改变具有最高分辨率的解码器的结构,它应仍与图 I-7-14 中所述的一致。在两个层次上产生的比特流与“高”框架兼容。然而,基层解码器的实现可能有所不同,它的实现复杂性被降低来适合软

件解码。

#### I -D. 7.2.1 解码器的实现

在较低的空域分辨率解码中,当对基层进行解码时,可能要使用较小尺寸的反 DCT 变化。在解码器 MC 循环中所需的帧存储器也将随之减少。

如果在编码器这边仅用一个 MC 循环来产生两个 SNR 分级层的比特流,则基层视频将会飘移。根据实际的应用,这个飘移可能是可被接受的,也可能不可接受。图象质量在很大程度上会依赖于用于解码器中的运动补偿的子采样精度。使用基层中为运动补偿而传输的那种全精度运动向量,并具有可与更高层相比的子采样精度,这也是可能的。通过使用先进的子采样插值滤波器[见附录 I -G 中的(12),(13)和(16)]可以实现将飘移最小化。

#### I -D. 7.2.2 编码器的实现

对基层 SNR 分级比特流进行剪裁,来满足变比分率的解码器的特殊要求是可能的。通过仅仅传输属于基层比特流中合适子集的恰当的 DCT 系数,可更简单地获得较小尺寸的 DCT。

最后,通过在编码器结构中加入多于一个的 MC 循环,可以在较低的分辨率比例上实现无飘移解码。编码器和解码器中使用同样的重构过程。

### I -D. 7.3 在数据分割中的比特率分配

当可获得具有不同错误率的两个通道时,数据分割可以将比特流分开,来实现更强的错误复原能力。通常,需要限制每一部分的比特率。这可以通过在编码器中适当地改变每个组块中的优先级断点来实现。

编码器可以使用针对于两个比特流的两个虚拟缓冲器,并且通过选择最接近每个信道的目标速率的优先级断点来实现速率反馈控制。

为了获得更高的错误复原能力,需要在帧与帧之间改变比特率。一般来讲,I-图在 0 区中比 P-图有更多的数据,并可从中获得好处,而 B-图则全部被放在 1 区中。

### I -D. 7.4 时域分级

由一个基层和一个增强层组成的双层时域分级编码结构在图 I -D-1 中示出。考虑以全时域速率输入到时域分解器的视频流,在我们的例子中,它被时域分解而形成两个视频序列,一个输入到基层编码器,另一个输入到增强层编码器。基层编码器是工作在半时域速率的无等级编码器。增强层编码器类似于一个“主”框架编码器,并且也工作在半时域速率,只在它使用于运动补偿预测的基层解码图象时有例外。在系统复合器中,基层和增强层的编码比特流被复合成单一比特流。系统分解器再将其分解为两路比特流,并将相应的比特流分别输入到基层和增强层解码器。基层解码器的输出可在半时域速率独自显示,或者在与增强层解码帧复合后以全时域速率显示。

下面的时域分级形式是被支持的,并以“更高层”表示:基层-增强层图象格式。

- (1) 逐行: 逐行-逐行时域分级。
- (2) 逐行: 隔行-隔行时域分级。

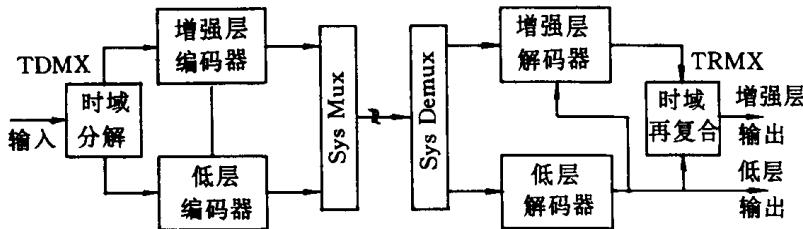


图 I-D-1 针对时域分级的双层编码解码结构

(3) 隔行：隔行-隔行时域分级。

#### I-D.7.4.1 逐行：逐行-逐行时域分级

假定输入的是逐行视频信号，如果需要在基层和增强层中对逐行格式视频信号进行编码，则“时域分解”的操作相对简化，包括将输入帧时域分解为两个逐行序列；“时域再复合”的处理是一个反过程，即，它对两个逐行序列进行再复合来产生全时域速率的逐行输出。见图 I-D-2。

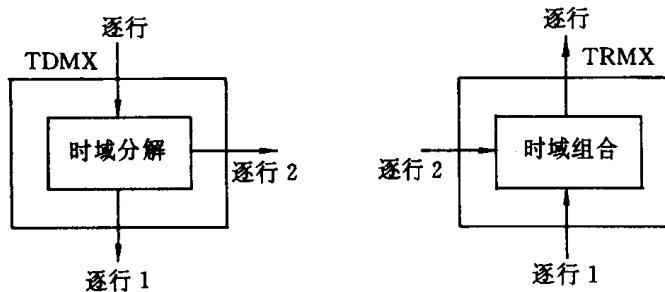


图 I-D-2 对于“逐行：逐行-逐行时域分级”的时域分解器和再复合器

#### I-D.7.4.2 逐行：隔行-隔行时域分级

再次假定输入为全时域速率的逐行视频信息，如果需要在基层中对隔行格式的视频信号进行编码，则“时域分解”的处理可能包括逐行到两个隔行序列的转化；这一过程包括从输入的逐行视频信号中抽取常规隔行序列和补足隔行序列。“时域再复合”是一个反过程，即，它实现两个隔行序列到一个逐行序列的转化，来产生全时域速率的逐行输出。图 I-D-3, I-D-4 和 I-D-5 示出了从逐行到两个隔行序列、从两个隔行序列到一个逐行序列的转化过程以及两个隔行到一个逐行序列的分解器和再复合器。

#### I-D.7.4.3 隔行：隔行-隔行时域分级

假定输入为隔行视频信号，如果需要在基层和增强层中对隔行格式的视频进行编码，则“时域分解”处理可能相对简化，包括将输入帧分解为两个隔行序列；“时域再复合”是一个反过程，即，它实现对两个隔行序列再复合，并生成全时域速率的隔行输出。分解和再复合过程类似于图 I-D-2 中所述。

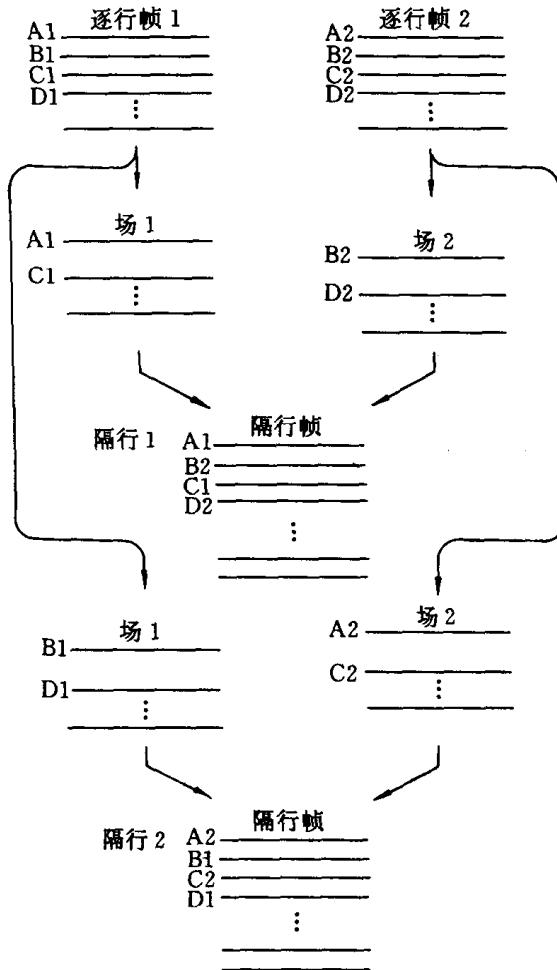


图 I -D-3 逐行一双隔行的转化

## II -D. 7.5 空域、SNR 和时域分级扩展的混合

本标准也允许通过使用各种分级方法的组合,来产生用于支持更多的应用需求的两个以上的视频层。一次取两种分级方法,可以直接产生三种组合结果。此外,在每种组合内,如果改变分级方法的应用顺序,也会产生各种特有的应用结果。在混合分级中,涉及到三个层次,用基层、增强层 1 和增强层 2 来表示。

### I -D. 7.5.1 空域和 SNR 的混合分级应用

#### (1) HDTV 和具有两种质量的标准 TV

基层提供基本质量的标准 TV 分辨率,增强层 1 通过使用 SNR 分级产生更高质量的标准 TV 分辨率,增强层 2 使用由空域分级编码的 HDTV 分辨率和格式,这与增强层 1 产生的高质量标准 TV 分辨率有关。

#### (2) 具有两种质量的标准 TV 和低清晰度电视/视频电话

基层提供视频电话/低清晰度质量,通过使用空域分级,增强层 1 提供具有基本质量的标准 TV 分辨率,增强层 2 使用 SNR 分级协助产生高质量的标准 TV。

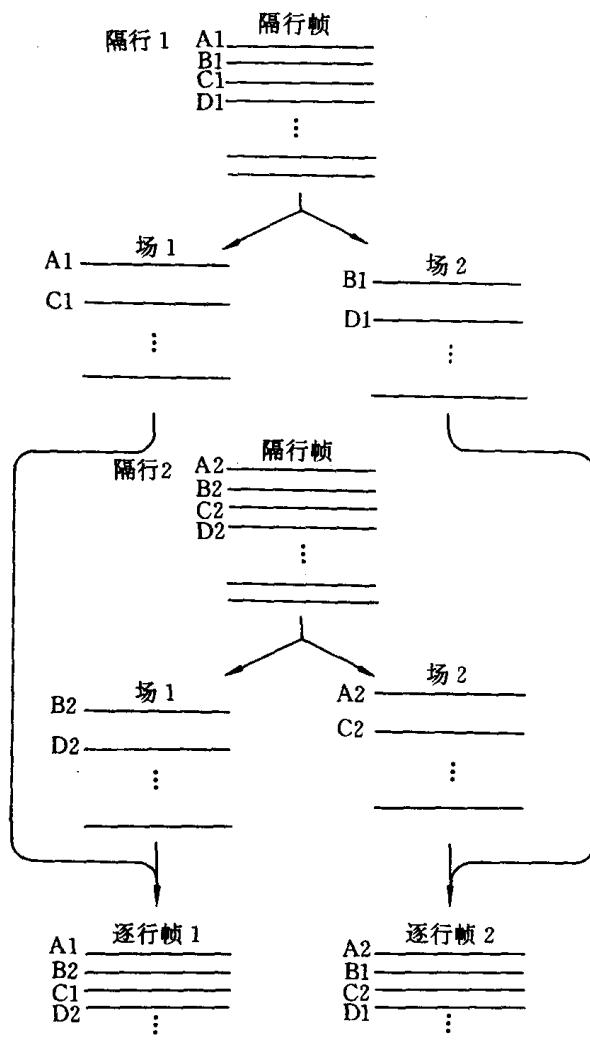


图 I-D-4 双隔行—逐行的转换

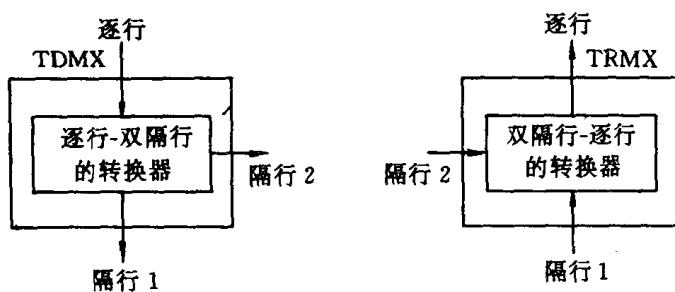


图 I-D-5 对于“逐行:隔行-隔行时域分级”的时域分解器和再复合器

### (3) 具有两种质量的 HDTV 和标准 TV

基层提供标准 TV 分辨率。通过使用空域分级，增强层 1 提供基本质量的 HDTV，增强层 2 使用 SNR 分级协助产生高质量的 HDTV。

## **II -D. 7. 5. 2 空域和时域混合分级的应用**

### **(1) 具有基本隔行 HDTV 和标准 TV 的高时域分辨率逐行 HDTV**

基层提供标准 TV 分辨率。通过使用空域分级,增强层 1 提供隔行格式的基本 HDTV,增强层 2 使用时域分级协助产生全时域分辨率的逐行 HDTV。

### **(2) 具有增强逐行 HDTV 和基本逐行 HDTV 的高分辨率逐行 HDTV**

基层以时域分辨率提供基本的逐行 HDTV 格式。通过使用时域分级,增强层 1 以全时域分辨率协助产生逐行 HDTV,增强层 2 通过使用空域分级来提供高空域分辨率的逐行 HDTV(以全时域分辨率)。

### **(3) 具有增强逐行 HDTV 和基本隔行 HDTV 的高分辨率逐行 HDTV**

基层提供基本的隔行 HDTV 格式,通过使用时域分级,增强层 1 以全时域分辨率协助产生逐行 HDTV,增强层 2 使用空域分级来提供高空域分辨率的逐行 HDTV(以全时域分辨率)。

## **II -D. 7. 5. 3 时域分级和 SNR 分级的混合应用**

### **(1) 具有两种质量基本逐行 HDTV 的增强逐行 HDTV**

基层以两种时域分辨率提供基本的逐行 HDTV。通过使用时域分级,增强层 1 以全时域速率但以基本质量协助产生逐行 HDTV,增强层 2 使用 SNR 分级以高质量协助产生逐行 HDTV(以全时域分辨率)。

### **(2) 具有两种质量基本隔行 HDTV 的增强逐行 HDTV**

基层提供基本质量的隔行 HDTV。通过使用 SNR 分级,增强层 1 以高质量协助产生隔行 HDTV,增强层 2 以全时域分辨率协助产生逐行 HDTV(高质量)。

## **II -D. 8 兼容性**

本标准支持各种分辨格式之间的兼容性,也支持和 ISO/IEC 11172-2(以及 ITU-T Rec. H. 261)的兼容性。

### **II -D. 8. 1 较高和较低的分辨率格式间的兼容**

本规范支持不同的视频分辨率格式间的兼容。对于空域和时域分辨率,通过使用空域分级和时域分级等方法来提供兼容。视频编码为两种分辨率的层次。仅能或仅愿显示较低分辨率视频的解码器接收低层比特流,并对其解码。完整分辨率的视频可以通过接收两个分辨率层并对其解码而得到重构。

### **II -D. 8. 2 和 ISO/IEC 11172-2(以及 ITU-T Rec H. 261)的兼容**

本规范的语法支持与 ISO/IEC 11172-2 的向前和向后兼容。由于本规范的语法是 ISO/IEC 11172-2 语法的超集,所以可以提供与 ISO/IEC 11172-2 的向前兼容。不包含 sequence\_extension 的 MPEG-2 比特流是向后兼容的。本规范提供的空域分级方法允许在较低分辨率即在基层中使用 ISO/IEC 11172-2 编码。这样,可以实现向后兼容。

视频语法中还包含了实现与 H. 261 兼容的工具,这在将来有可能用到,然而目前未

获得任何框架的支持。

同时联播可以作为提供与 H.261 和 ISO/IEC 11172-2 向后兼容的一种简单的可选方案。

## II -D. 9 本规范与 ISO/IEC 11172-2 的差异

本节列举了 MPEG-1 和 MPEG-2 视频流间的差别。

所有依据当前定义的框架和级别的 MPEG-2 视频解码器应能对 MPEG-1 比特流进行解码。

在许多情况下, MPEG-2 代表 MPEG-1 的一个超集。例如, MPEG-1 系数的“之字形”扫描是 MPEG-2 的两种系数扫描方式之一。然而, 在有些时候, MPEG-1 的语法元素(或语义)不能在 MPEG-2 中找到直接等价的内容。下面列出所有这些元素。

本文档可以帮助(标准的)实现者来区分出哪些是不能在 MPEG-2 中找到等价部分的 MPEG-1 视频语法(或语义), 因此为了保证 MPEG-1 的兼容性, 应特别对此引起重视。

在本节中, MPEG-1 指的就是 ISO/IEC 11172-2, 而 MPEG-2 指的是本规范。

### II -D. 9. 1 IDCT 解谐

MPEG-1 IDCT 解谐控制主要在于将反量化后仍为偶数的每个非零系数加 1(或减 1)。这作为反量化过程的一部分在 MPEG-1 的 2.4.4.1, 2.4.4.2 和 2.4.4.3 节中有所描述。

MPEG-2 IDCT 解谐控制在于如果反量化后所有系数的总和仍为偶, 则将系数 [7][7] 加 1(或减 1), 这在 MPEG-2 的 I-7.4.4 中有所描述。

### II -D. 9. 2 宏块填充

MPEG-1 在每个 macroblock\_address\_increment 之前, 可以插入任意多个 VLC 码“0000 0001 111”(macroblock\_stuffing)。解码器必须丢弃这个码。这在 MPEG-1 的 2.4.2.7 节中有所描述。

MPEG-2 这个 VLC 码被保留, 不在 MPEG-2 中使用。在 MPEG-2 中, 填充可以通过仅在起始码之前插入零字节来产生。这在 MPEG-2 的 I-5.2.3 节中有所描述。

### II -D. 9. 3 Run-level 换码语法

MPEG-1 不能用 VLC 编码的 Run-level 值通过后面跟有 14 位 FLC( $-127 \leq \text{level} \leq 127$ ) 或 22 位 FLC( $-255 \leq \text{level} \leq 255$ ) 的“换码”码“0000 01”来编码。这在 MPEG-1 的附录 B, B.5 中有所描述。

MPEG-2 不能用 VLC 编码的 Run-level 值通过后面跟有 18 位 FLC( $-2047 \leq \text{level} \leq 2047$ ) 的“换码”码“0000 01”来编码。这在 MPEG-2 的 I-7.2.2.3 中有所描述。

## **II -D. 9. 4 色差样本水平位置**

MPEG-1 色差样本的水平位置处于亮度样本的中间。这在 MPEG-1 的 2.4.1 中有所描述。

MPEG-2 色差样本与亮度样本处于水平同一位置。这在 MPEG-2 的 I-6.1.1.8 中有所描述。

## **II -D. 9. 5 组块**

MPEG-1 组块没有必要在宏块的同一水平行起始和结束。由此，在一个单一组块中就有可能包括一幅图的所有宏块。这在 MPEG-1 的 2.4.1 中有所描述。

MPEG-2 组块通常在宏块的同一水平行起始和结束。这在 MPEG-2 的 I-6.1.2 中有所描述。

## **II -D. 9. 6 D-图**

MPEG-1 对于 D-图(picture\_coding\_type=4)定义了一种特殊的语法。D-图像 I-图一样，只有内部 DC 系数，没有块结尾，有一个特殊的 end\_of\_macroblock 码“1”。

MPEG-2 不允许有 D-图(picture\_coding\_type=4)。这在 MPEG-2 的 I-6.3.9 中有所描述。

## **II -D. 9. 7 Full-pel 运动向量**

MPEG-1 语法元素 full\_pel\_forward\_vector 和 full\_pel\_backward\_vector 可被置为“1”。此时，编码的运动向量以全象素为单位。而不是以半象素为单位。运动向量的坐标在其用于预测之前必须乘以 2。这在 MPEG-1 的 2.4.4.2 和 2.4.4.3 中有所描述。

MPEG-2 语法元素 full\_pel\_forward\_vector 和 full\_pel\_backward\_vector 必须等于“0”。运动向量通常以半象素为单位进行编码。

## **II -D. 9. 8 高宽比信息**

MPEG-1 在序列头中编码的 4 位 pel\_aspect\_ratio 值描述了象素高宽比。这在 MPEG-1 的 2.4.3.2 中有所描述。

MPEG-2 在序列头中编码的 4 位 aspect\_ratio\_information 值规定了显示高宽比。象素高宽比可由显示高宽比，以及帧尺寸和显示尺寸求得。这在 MPEG-2 的 I-6.3.3 中有所描述。

## **II -D. 9. 9 forward\_f\_code 和 backward\_f\_code**

MPEG-1 用于对运动向量解码的 f\_code 值为 picture\_header() 中的 forward\_f\_code 和 backward\_f\_code。

MPEG-2 用于对运动向量解码的 f\_code 值为 picture\_coding\_extension() 中的 f\_code[s][t]。forward\_f\_code 和 backward\_f\_code 的值必须为“111”，且被忽略。这在

MPEG-2 的 I -6.3.9 中有所描述。

#### II -D. 9. 10 constrained\_parameter\_flag 和最大的 horizontal\_size

MPEG-1 当 constrained\_parameter\_flag 被置为“1”，这表明检测到了一些约束。那些约束之一是使 horizontal\_size $\leq$ 768。应该注意的是，一个受约束的 MPEG-1 视频比特流可以有水平尺寸最多为 768 个象素的图象。这在 MPEG-1 的 2.4.3.2 中有所描述。

MPEG-2 以框架和级别的方法代替了 constrained\_parameter\_flag 方法。然而，应该注意 MPEG-2 比特流中不应有水平尺寸大于 720 个象素的图象。这在 MPEG-2 的 I -8.2.3.1 中有所描述。

#### II -D. 9. 11 MPEG-2 语法转换为 MPEG-1 语法

通过使用 MPEG-1 中所没有的 MPEG-2 中的各种语法元素的特定值，可以生成与 MPEG-1 语法非常接近的 MPEG-2 比特流。

换句话说，当对以下参数如此赋值之后，MPEG-1 解码过程便与 MPEG-2 解码过程一致（除了以前提到的一些特定点外）：

```
progressive_sequence="1" (逐行序列)
chroma_format="01" (4 : 2 : 0)
frame_rate_extension_n=0 且 frame_rate_extension_d=0 (MPEG-1 帧速率)
intra_dc_precision="00" (8 位内部 DC 精度)
picture_structure="11" (帧图，由于 progressive_sequence="1")
frame_pred_frame_dct=1 (仅为基于帧的预测和帧 DCT)
concealment_motion_vectors="0" (无隐藏运动向量)
q_scale_type="0" (线性 quantiser_scale)
intra_vlc_format="0" (针对内部宏块的 MPEG-1 VLC 表)
alternate_scan="0" (MPEG-1“之字形”扫描)
repeat_first_field="0" (由于 progressive_sequence="1")
chroma_420_type="1" (色差为“基于帧的”，由于 progressive_sequence="1")
progressive_frame="1" (由于 progressive_sequence="1")
```

#### II -D. 10 复杂性

MPEG-2 标准支持高性能/高复杂性解码器和低性能/低复杂性解码器的组合。这通过 MPEG-2 的各种框架和级别的定义来提供，对于每个新的框架，均引入新的工具和功能的集合。这样，就可以通过降低 MPEG-2 编码方法实现复杂性来改变其性能。

此外，某些限制可以降低解码器的实现开销。

## II -D. 11 编辑编码的比特流

为了避免重编码的开销和性能上的代价,我们支持针对编码比特流的一些特殊处理。对编码的比特流进行编辑、连结,不需重编码,且不破坏解码图象序列也是可能的。

既要高压缩比,又要能容易地进行编辑,这中间有一些冲突。编码结构和语法不是以对任何图象均可方便地编辑为主要目的而设计的。然而还是有一些特性能使对编码数据的编辑成为可能。

对编码的 MPEG-2 比特流的编辑的实现应归功于编码视频比特流的语法等级体系。在这个等级体系中不同的级别具有其独特的起始码(即,视频序列、图象组等等)。视频编码可以在比特流中加入图内/组块内访问点。这使得不需对整个视频进行解码,而可以识别、访问、编辑比特流的某一部分成为可能。

## II -D. 12 特技方式

某些 DSM(数字存储媒体)提供诸如 FF/FR(快进/快退)这样的特技方式。MPEG-2 语法支持 ISO/IEC-11172-2 的所有特殊的访问、搜索和扫描方式。这一功能由视频比特流的语法等级体系所支持,它可以实现在视频序列内辨识出有关部分。能够提供比特流分级来限制访问比特率(即数据分割和一般的组块结构)的 MPEG-2 工具可以帮助实现以上功能。这一节给出了针对 DSM 提供的比特流的解码的一些原则。

解码器从 PES 包头中的一位标志(DSM\_trick\_mode\_flag)来获得信息。这个标志指出了比特流以特技方式由 DSM 重构,且从语法的观点来看,比特流是有效的,而从语义的观点来看,比特流是无效的。当这一位被置后,后面跟有一个 8 位的域(DSM\_trick\_modes)。DSM\_trick\_modes 的语义见 ISO/IEC 13818-1。

### II -D. 12. 1 解码器

当解码器对 DSM\_trick\_mode\_flag 为 1 的 PES 包进行解码时,我们建议解码器应:对比特流进行解码并依据 DSM\_trick\_modes 进行显示。

#### 前处理

当解码器遇到 DSM\_trick\_mode\_flag 为 1 的 PES 包,我们建议解码器应:  
从缓冲区中清除非特技方式的比特流。

#### 后处理

当解码器遇到 DSM\_trick\_mode\_flag 为 0 的 PES 包,我们建议解码器应:  
从缓冲区中清除特技方式的比特流。

#### 视频部分

如果解码器对 DSM\_trick\_mode\_flag 为 1 的 PES 包进行解码,则我们建议解码器应:

忽略 VBV\_delay 和 temporal\_reference 的值。

对一幅图进行解码并显示,且一直延续到下一幅图被解码。

处于特技方式的比特流在组块之间可能有一个间隙。当解码器遇到组块间的间隙时,我们建议其应:

对这个组块进行解码,并依据组块头中的组块竖直位置来显示。并用最后显示图象的同一位置的数据来填充间隙。

## II -D. 12. 2 编码器

我们建议编码器应:

用带内部宏块的短小组块编码。

通过内部图或内部组块进行短周期刷新编码。

### DSM

我们建议 DSM 应以特技方式向比特流提供完备的语法。

#### 前处理

我们建议 DSM 应:

在 picture\_header() 和更高的语法结构上完成“标准的”比特流。

#### 系统部分

我们建议 DSM 应:

将 PES 包头中的 DSM\_trick\_mode\_flag 置为 1。

根据特技方式置 DSM\_trick\_modes(8 位)。

#### 视频部分

我们建议 DSM 应:

向 sequence\_header() 中插入与标准比特流相同的参数。

向序列扩展头中插入与标准比特流相同的参数。

向图象头中插入与标准比特流相同的参数,除了 vbv\_delay 以外,将 vbv\_delay 置为 FFFF,来表明为变长比特率。

注意 在解码器中忽略 temporal\_reference 和 vbv\_delay,因此 DSM 不需将 temporal\_reference 和 vbv\_delay 置为当前值。

将由内部编码宏块组成的组块连结在一起。连结在一起的组块的竖直位置应以升序排列。

## II -D. 13 错误复原

大多数的数字存储媒体和通信通道都不是无错的。应使用适当的通道编码方法,但这超出了本规范的范围。不过,MPEG-2 语法支持与 ATM 网中的单元丢失和传输中的位错误(脱离和破裂)有关的错误复原方式。本规范中所定义的压缩方案中的组块结构使得解码器在遇到残留的数据错误后,可以恢复并再同步其解码过程。因此,压缩数据中的位错误所导致的解码图象中的错误将局限于一定区域内。解码器可以使用“隐藏”策略将这些错误掩盖起来。错误复原包括:与位错误率(BER)相称的适度(品质)降低和在丢失视频位

或数据包情况下的适度恢复。应该注意的是，所有项目可能都需要系统层的额外支持。

作为一个基于分组包的系统的例子，下面将阐述具有异步传输模式(ATM)的B-ISDN的某些细节。对于其它的用“向前修改错误编码”方法来保护各个数据包的系统，我们也可作出类似的阐述。

ATM 使用短小的、定长的分组包，称为“单元”。它由包含常规信息的 5 个字节的头和 48 个字节的用户有效负荷组成。ATM 上的错误主要是：一些单元可能丢失，以及某些单元的用户有效负荷可能包含位错误。依靠 AAL(ATM 适应层)功能，可以指出丢失的单元和包含位错误的单元。

表 I-D-2 基于简单的统计模型，对于一定范围的 CLR 和服务比特率值，总结了单元丢失的平均时间间隔，这可以反映出在 ATM 环境中单元丢失所造成的影响。(每个单元要为 AAL 功能准备一个字节，这样(每单元)就剩下了 376 位=47 字节的单元有效负荷)。然而，应该注意，这个表中的统计未考虑单元丢失突发和其它的更短期的时间统计。

表 I-D-2 对于一定范围的 CLR 和服务比特率的单元丢失平均间隔

	错误平均时间间隔			
	5Mbit/s	10Mbit/s	50Mbit/s	100Mbit/s
$10^{-2}$	7.52ms	3.76ms	0.752ms	0.376ms
$10^{-3}$	75.2ms	37.6ms	7.52ms	3.76ms
$10^{-4}$	752ms	376ms	75.2ms	37.6ms
$10^{-5}$	7.52s	3.76s	752ms	376ms
$10^{-6}$	1.25min	37.6s	7.52s	3.76s
$10^{-7}$	12.5min	6.27min	1.25min	37.6s
$10^{-8}$	2.09h	1.04h	12.5min	6.27min

对于孤立的位错误，能够计算出符合上面出错平均间隔的位错误率(BERs)。由公式  $BER = CLR / 376$  可以求出产生同样错误发生率的 BER。

以下这些用来减少单元丢失和其它错误的影响的技术可供参考，当出错时，可使用本规范中提供的各种工具和方法来解决问题。应该注意的是，对于其它尺寸大小的分组包(例如，LANs 或某些存储媒体)，或具有除单元丢失以外的其它类型的未修改错误的视频数据，都可用到这些技术。由于位错误所造成的影响不可能被预测，所以把已知的清除(记录)(已知存在于数据块中某处的未修改的位错误)作为丢失的数据块来处理可能是合适的。然而，这应成为一个解码器的任选项。以下的讨论主要应涉及“传输分组包”，以强调对各种存储转发系统的适应性。然而，某些特例应涉及“单元丢失率”(CLRs)，这是因为在准备本规范之时，对单元传输进行了最完整的规定。

错误复原技术总结为三种类型，包括当错误出现时对它的隐藏，以及对空间(一幅图内)和时间(从一幅图到另一幅图)范围内的丢失或出错所造成的影响的限制。

## II -D. 13. 1 隐藏的可能性

隐藏技术可以在丢失/出错发生时,掩盖其造成的影响。有些隐藏方法使用任何编码比特流均可实现,而其它的则需依靠编码器来配置数据或提供附加信息,以使增强的性能可以实现。

### I -D. 13. 1. 1 时域预测隐藏

解码器可以通过从时空邻近数据预测丢失数据来提供错误隐藏。在假设邻近的块之间的图象特性通常是相似(在空域和时域)的前提下,解码器使用正确接收到的信息来预测显示什么内容以代替丢失/出错的数据,对于时域的情况,意味着从邻近的场或帧来进行预测。

#### 1. 从以前的帧中替换

最简单而又可行的方法是从以前的图象中同一位置上取一宏块来代替丢失的宏块。这种方法适于相对静止的图象区域,但对于运动区域,这种替换是很显眼的。由于双向预测的使用和图象解码顺序与图象显示顺序之间的差异,对于“以前的图象”的解释必须十分小心。如果在一幅 P-图或 I-图中,某一宏块丢失,则可以通过将以前的 P-图或 I-图的同一宏块的相应数据拷贝到这里来掩盖。这能够保证这幅图在用于进一步的预测之前是完整的。B-图中的丢失宏块可以由最后显示的图象代替,可以是任何类型的图象,或者由尚未显示,但在存储器中的下一幅 I-图或 P-图来代替。

#### 2. 运动补偿隐藏

针对邻近图象的隐藏可以通过预测丢失宏块的运动向量来改进,这主要靠受损图象中相邻宏块的运动向量来实现(假设这些宏块未丢失)。这可以改进运动图象区域中的“隐藏”,但这里有一个明显的问题,那就是如果出错宏块的邻近宏块均为内部宏块,则由于其并无运动向量,所以便不易预测出错宏块的运动向量了。编码器可以提供帮助以克服这个问题,这在 I -D. 12. 1. 1. 3 中有所讨论。

复杂的运动向量估计还会涉及到在丢失宏块上方或下方的相邻宏块的存储,这主要是为了可以及时地进行向前预测和向后预测(对于 B-图)。然后可以对来自上方和下方(如果可以得到)的运动向量求平均。

较简单一些的解码器可以只使用前向预测和/或来自丢失宏块的上方宏块的运动向量,这将省去存储和插值。

#### 3. 内部 MVs 的使用

如果丢失宏块的上方或下方的宏块为内部宏块,则通常不能使用上一节中所述的运动补偿隐藏技术,这是因为内部编码宏块不带运动向量。特别是在 I-图中,采用通常的计算和运动向量不可能实现这种隐藏。

编码过程可以进行扩展,以使内部宏块也包含运动向量。当然,对于某一特定宏块,这个运动向量和编码信息必须分开传输(如,在不同的分组包中),这样,当图象数据丢失后,仍可得到运动向量。

当“comcealment\_motion\_vectors”=1 时,运动向量与内部宏块一同传输,这样可以允许解码器具有改进的隐藏功能。内部宏块所带的“隐藏”运动向量仅用于位于该宏块正

下方的宏块。

为简单起见,内部宏块所带的隐藏运动向量通常都是前向的,并且在帧图中当帧运动向量看待,在场图中当场运动向量看待。

因此,准备生成隐藏运动向量的编码器应为给定的内部编码宏块传输用于隐藏(即预测,分别进行基于帧或场的向前预测),且位于内部宏块正下方的宏块的运动向量的帧或场运动向量。

隐藏运动向量主要打算用于 I 或 P 图,但从语法上讲,允许在 B 图中用到它们。B 图中的“隐藏”并不太重要,这是由于 B 图并不用作预测器,因此错误也不会传播到其它图象。所以,在 B 图中传输隐藏运动向量是一种浪费。

位于图象的底行的内部宏块所带的隐藏运动向量是没什么用的。然而,如果“concealment\_motion\_vectors”=1,则那些隐藏运动向量必须被传输。编码器可以使用向量(0,0)来使编码开销最小。

当使用隐藏运动向量时,使一个组块包含一行(或更少)的宏块是个好办法,这样,当一个组块或组块的某部分丢失时,“隐藏”便可限于少于一行的宏块之中。这就意味着连续丢失两行的宏块的可能性要小得多,所以使用隐藏向量而实现有效“隐藏”的机会增大了。

注意 当“concealment\_motion\_vectors”=1 时,如果传输了一个内部宏块,则 PMVs(运动向量预测器)未复位。通常由内部宏块复位 PMVs。

#### II -D. 13. 1. 2 空间预测隐藏

待预测的隐藏宏块也可以通过对同一幅图象中的相邻宏块进行插值来生成,[附录 I -D(17)]。这最适于高速运动的区域,此时时域预测不能奏效,也可为不能获得隐藏运动向量的内部宏块提供另一种隐藏方法(I -D. 12. 1. 1. 3)。对于场景切换后的单元丢失,这种方法也特别有用。

有几种方法可以实现空间插值,能够在空域或 DCT 域执行,但通常它只能用于预测丢失宏块的主要性能参数,如 DC 系数或最低 AC 系数。对于细微细节(高频)的预测很难成功,并且在快速运动的图象中没有什么价值。

空间预测的宏块隐藏也可以与分层编码方法联合(即,数据分割或 SNR 分级混合,见 I -D. 12. 1. 3)。如果在发生单元丢失时,宏块中的一些 DCT 系数从低层中恢复,则可以为错误隐藏而获得全部信息(从低层的同一宏块复原的 DCT 系数和从邻近宏块接收的所有 DCT 系数)。这对于由于带宽限制而仅包含 DC 系数的低层特别有用。

#### II -D. 13. 1. 3 有利于隐藏的分层编码

通过对编码视频信息进行安排而进一步改进隐藏过程,这样最重要的信息就最有可能被接收到。不太重要的信息的丢失可以被更有效地隐藏。通过使用具有不同优先级的存储设备和传输媒体可以实现这种方法(如 B-ISDN 中的优先级可控的基于单元的传输,或为不同的信息提供不同的错误保护和错误修改的地方)。由编码过程产生的各种分量,根据其丢失对重构图象的影响的不同而分为具有不同重要性的级。通过指明比特流分量的优先级,并按各分量的重要性分别对待,则可获得更高的错误“隐藏”性能。

以下几种策略可用来产生具有等级顺序的比特流或层。

数据分割 编码的宏块数据分为多层,这样,零区包含地址和控制信息以及顺序靠前

的 DC 系数,而一区包含高频 DCT 系数。

**SNR 分级混合** 两个系数集均被反量化,并在解码之前,在接收器处被加到一起。其中一个系数集是另一个系数集的量化误差的改进数据,也可能有其它的组合(包括对数据分割的仿效)。

**空域分级混合** 低层可以不用考虑增强层而编码,并可使用其它标准的编码方法(ISO/IEC 11172-2 等)。增强层包含来自基于低层的预测的编码差分信号。

**时域分级混合** 增强层定义了附加图,当它和基层再度复合时,可产生具有更高图象速率的组合图象序列。

当把按上述策略产生的层逐步叠加后,便可产生不断改善质量的重构信号。尽管这些编码技术与不分层的系统相比可能导致比特率的增长,但当遇到信道错误时,分层系统将显示出更高的性能。

单独考虑错误复原,这些按等级排序的层应根据质量来处理,这样一些功能(如对于给定整体比特率的图象质量)可以达到最优。在下面这些场合之一或更多的情况下,比特流分量可能有不同的处理:

- 编码器 可能使用不同的信道编码。
- 信道 对于不同的比特流分量,信道可能提供不同的单元/包丢失概率或不同的错误特性。
- 编码器 在各个比特流中,错误“隐藏”可能有不同的表现。

### 1. 数据分割的使用

数据分割可将宏块数据简单地分为两层。PBP 指针定义了每一层的内容。通常,0 数据区包含地址、控制信息以及低频 DCT 系数,而 1 数据区包含高频 DCT 系数。

在编码器中 PBP 指针的值对于每个组块可能都是不同的,这样,可以控制两个层次之间位的分配(例如,保持常量)。对于 I-帧,P-帧和 B-帧,这种分配是不同的。两个层次间的等级安排意味着对某些宏块,0 数据分区不包含 DCT 系数或运动向量。

如果对信道错误进行分配,以使 1 数据区接收大部分的错误,则可实现很好的容错性。

我们假定在解码器中可以检测出错误,这样就能够采取行动来防止错误被显示。对于 1 数据区,只简单地不显示出错数据(即,只使用 0 数据区)。通过高度可靠的传输可以使 0 数据区的数据丢失或出错减到最小。解码器的隐藏作用也是必要的。

### 2. SNR 分级编码的使用

SNR 分级编码根据是对一个还是两个层次进行解码来提供具有相同空间分辨率但不同的图象质量的两个层。这种技术主要是为了当缺省增强层时,仍可提供可用的低质量层。而且,如果错误能够主要限制于增强层中,则也可实现很好的错误复原。

当增强层中出错时,仅受影响的图象区域使用低层信号。特别是对于经常出现的错误,增强层的临时丢失或永久丢失,这种隐藏都是非常有效的,这是由于显示的信号不会产生像分块或运动抖动这样的非线性失真。

如果永久丢失了增强层,并且这样仅对低层解码,则在编码器中仅实现了一个 MC 预测循环的地方可能发生偏移。然而,在大多数的构型中这种偏移都是不可见的(例如,M

= 3, N = 12 通常会提供充分的修正)。

对于非常高的错误率,上层信号的临时或永久的丢失,SNR 分级系统的低层信号很适应“隐藏”。然而,当无差错时,上层质量就不能达到像分层方法(如数据分割)那样的子带质量。

### 3. 空间分级编码的使用

空间分级编码允许低层不用考虑增强层而编码,并且可以使用其它标准的编码方法(ISO/IEC 11172-2 等)。增强层包含来自基于低层的预测的编码差分信号。对于增强层中的错误,向上转换的低层信号可以直接作为受损图象区域的隐藏信息来使用。特别是对于经常发生的错误或增强层的临时丢失,这种隐藏信号相对来讲不会产生像分块(当高频 DCT 系数完全从低层中丢失时便会产生)或运动抖动(当运动信息从高优先级层中遗漏)这样的非线性失真。

如果没有错误发生,则向上变化的低层信号作为附加预测信号,采用适应于宏块的方法来改进上层的编码性能。因此,增强层比特流由量化的剩余时域或低层预测误差组成。

空域分级编码提供非常适用于在有很高的错误率或增强层信号临时丢失的情况下隐藏方法。然而,当两个层都用上时,增强图象质量通常不如其它的分层编码方法。

### 4. 时域分级编码的使用

时域分级是一种允许对视频帧分层的编码技术。在每层中,帧的空间分辨率是相同的,但每层的时域速率均低于源数据,而两个层的时域速率的组合等于源数据的时域速率。如果增强层中出现错误,则具有全空域分辨率的基层可以简单地被用于隐藏。特别是对于经常出错或增强层信号临时丢失这些情况,基层信号可以提供很好的隐藏性能。

像那些使用 SCIF 格式的电信应用领域,可以通过使用具有 SCIF 空间分辨率,但只有一半的时域分辨率,对基层编码来实现具有时域分级的高程度错误复原,剩余的对应于另一半时域分辨率的帧编码于增强层中。通常,赋予增强层较低的优先级,当其丢失时,基层解码帧可以通过帧重复来实现隐藏。这种“隐藏”仅导致全时域分辨率的暂时损失,但可保证全部空间分辨率和质量。

在诸如使用高时域分辨率逐行视频格式作为源信号的那些 HDTV 应用中,可以用时域分级来实现高程度的错误复原。那种应用被设想为需要两层,一个基层,一个增强层,它们均处理同样的图象格式(均为隔行的或均为逐行的),但这是以一半的时域速率进行的。不考虑其所选格式的增强层和基层的时域再复合通常可以产生源信号的完整的逐行时域分辨率。在 HDTV 传输中,如果低优先级的增强层信号被破坏了,则基层信号可以用于隐藏,既可以如逐行格式基层那样直接使用,也可以像隔行格式基层那样,变换场地位以后再使用。

通常,增强层数据可以被赋予低优先级,当其丢失时,基层解码帧可以通过帧重复或帧平均来实现“隐藏”。这种“隐藏”仅会导致临时的,不易被察觉的时域分辨率损失,但可以保持完整的空间分辨率和质量。

## II -D. 13. 2 空间定位

空间定位通过提供在宏块间差分编码的比特流元素的早期再同步,来将一幅图中的

错误传播的程度降到最小。

孤立的位错误可以通过无效码字来检测出,但一个错误序列可能被解码器的设计者允许解码。然而,很难预测对图象的影响(可能产生合法但不正确的码字),这样,为了控制错误,即使是仅知某一块数据中某处的一位出了错,我们也要对整个受影响的组块进行隐藏。

如果出现了长序列的错误(例如,分组包或单元丢失),则实际上唯一的方法就是丢弃数据,直到遇到下一个重新同步点(位于下一个组块或图象头的起始码)。通过提供更多的重新同步点,受数据丢失或出错影响的图象区域可以减小,并可降低对隐藏技术的要求,在牺牲代码效率的条件下使得错误更不可见。因此,错误的空间定位就要靠控制组块的大小尺寸,这是由于组块是带有重新同步点的最小编码单位(起始码)。

#### I -D. 13. 2. 1 小组块

实现错误空间定位的最基本的方法是减少(固定的)组块中的宏块数目。重新同步点的出现频率的增长会减少出现数据丢失时受影响的图象区域。由于在 MPEG 编码视频中通常都有组块结构,所以这种方法在任何传输或存储媒体,以及任何框架中都是有效的。

由于增加了额外开销信息,这种方法会导致一些编码效率的损失。对于基于 4Mbit/s 的 Rec. 601 图象格式,如果每个组块包含 11 个宏块,则这种损失为 3%,如包含 4 个宏块,则这种损失为 12%(这些百分比是相对于每组块使用 44 个宏块或使用一幅图象宽度的系统而计算出来的)。在 4Mbit/s 而无差错的情况下,对于每组块含 4 个宏块的图象,效率损失而造成的图象质量下降为最多 1dB,对于每组块含 11 个宏块的图象,为最多 0.2dB。但当  $CLR = 10^{-2}$  时,这种方法可实现 1~5dB 的质量提高,这也取决于所使用的隐藏方法(简单的宏块替换或运动补偿隐藏)。

从可被察觉的图象质量的观点来看,这种方法的性能主要取决于组块与图象的相对尺寸的大小。因此,应该考虑图象尺寸(以宏块表示),以及在编码效率和出错引起的图象质量降低程度之间取舍,来决定组块尺寸。

#### II -D. 13. 2. 2 自适应的组块尺寸

根据不同的编码模式、图象结构,对一个图象组块编码所需的位数会有很大差别。如果组块仅包含少数几个宏块,则一个传输分组包,甚至是很小的一个分组包或单元,都可包含几个组块。在一个传输分组包中提供多个重新同步点起不到什么作用。过于简单化的短组块方案存在着的另一个问题是,由于没有考虑到传输分组包的结构,在发生丢失以后的第一个有效的传输分组包可能包含一个组块的大部分信息,但由于起始码已丢失,所以就没什么用了。

对小组块方法的一个改进是采用自适应的组块尺寸。当编码器生成比特流时,它跟踪传输包内的数据内容,在每个传输分组包中,一有机会就开始放入组块。这种方法可以实现与“固定尺寸小组块”方法同样的错误空间定位,但具有更高的效率。

然而,这种方法只会给基于单元或分组包的传输,以及在大数据块上进行错误检测的情况带来好处。如果在发生错误时,一个以上的组块发生丢失,则小组块定位的频繁的重新同步点仅仅是一个浪费。如果孤立的位错误仅影响一个组块,则没有什么必要来调整组块大小。

此外,这种自适应组块尺寸技术需要编码器和打包者之间建立密切的联系,即每有一种新的包或单元,则产生一种新的组块。这样,因为在编码过程中仅会设定一种传输分组包结构,则这种方法对于某些应用(如,对存储的视频采用这种方法分发)便不适用了。

### II -D. 13. 3 时间定位

时间定位通过提供差分编码的图象的早期重新同步点,来使得在时间轴上的图象之间的错误传播降到最小,作到这一点的最显而易见的途径是使用内部模式编码。

#### I -D. 13. 3. 1 内部图

通过使用内部图,如果每 N 幅图是一幅内部编码图,且在每幅 I 图以前有( $M-1$ )幅 B 图被显示,则一幅解码图中的错误最多不会传播到( $N+M-1$ )幅图以上。

内部图在视频解码序列的同步上起到重要作用。尽管它对于时间定位是非常有用的,但为错误复原而加入额外的内部图必须谨慎从事(即减小 N)。内部图的编码需要大量的位,在比特流中相对占很大的一部分,从而,也就更有被数据丢失或数据错误影响的可能。

#### II -D. 13. 3. 2 内部组块

为了避免由内部图引起的附加延迟,需要低延迟的应用系统可能更愿意仅对图象的某些部分进行内部编码。这可以提供与内部图同样的错误复原。例如,每幅图从顶端到底端有固定数目的组块被内部编码,而对整幅图进行更新需经过 P 幅图。应该注意这种更新方法的三个方面的问题是:

- 尽管通常来讲,图象上的出错区域在 P 幅图(平均  $P/2$  幅图的期间)内会被消除,但有时运动补偿会使干扰绕过内部刷新,而一直延续到  $2P$  幅图。
- 为了保证错误不会传播到图象中已被刷新的区域,应对运动向量加以限制,限制向量的垂直分量以保证预测不会由图象“最老的”部分作出。
- 清除错误的视觉效果类似于汽车的挡风刮水器清除水雾。在无错序列中,有时也可见到这种刮水器效果,除非速率控制机构能够保证内部组块的质量与周围的非内部宏块相近。

### II -D. 13. 4 小结

表 II -D-3 总结了以上的错误复原技术,并附了应用指南。

不可能非常简明地指出错误复原性能,这是因为评价必定是主观的且是依实际应用系统而定的,所以最多只能给出一个指导。而且这几种不同的错误复原方法也很可能是组合起来使用的。以下的描述是一些单元丢失的试验结果,仅针对基于单元的视频信息的传输,作为对系统性能的一些指导意见。

从以前帧中简单替换宏块的方法与“小组块”方法组合起来(每个组块 4 个宏块),对于  $CLR = 10^{-5}$  的相当低的错误率的大多数序列来讲,可以提供足够的图象质量(4Mbit/s, 主框架, 主级别的系统)。

高级的运动补偿“隐藏”方法(对于丢失的宏块进行完全空域和时域的运动向量插值,并且在使用内部组块更新的 P 图中隐藏丢失,即  $N = \text{无穷大}, M = 1$ )在  $CLR = 10^{-3}$  时可以提供足够的图象质量(4Mbit/s, 主框架, 主级别的系统)。

表 II-D-3 错误隐藏技术的总结

分类	技术	框架/应用
隐藏	时域预测——从以前的图象中替换	任何框架。最适于静态图象
	时域预测——运动补偿	任何框架。采用运动向量预测的先进技术
	时域预测——使用隐藏MVs	任何框架,但内部MVs的计算是一个编码器选项
	空域预测	任何框架,不适于静态的复杂图象
	数据分割	目前未应用于框架中,但可以作为前/后处理加入。最小的开销和复杂性。根据所分配的比特率,低层可以不提供可用的图象
	SNR分级	SNR分级、空域分级、高框架。适用于非常高的错误率或临时不可获得增强层。实现起来相对比较简单
	空域分级	空域分级和高框架。适用于非常高的错误率或增强层临时不可获得
	时域分级	目前未应用于框架中。适用于非常高的错误率或增强层临时不可获得
空间定位	小组块	任何框架
	自适应组块大小	任何框架,但当包大小一定后,需知道传输特性
时间定位	内部图	任何框架,但有延迟影响
	内部组块	任何框架,但错误的延续可能长于内部图方法

在数据丢失更严重的环境中的操作可能需要使用分层的编码方法。对于高优先级的信息进行充分保护,这些方法可以在 CLR 高达  $10^{-2}$  甚至  $10^{-1}$  时也能提供足够的图象质量。作为 4Mbit/s 的主框架、主级别系统的后处理功能而实现的数据分割,如果每个分区分配到 50% 的比特率,且基层中没有数据丢失,则有例子可以表明,在  $CLR = 10^{-3}$  时会有 0.5dB 损失,在  $CLR = 10^{-2}$  时会有 1.5dB 损失,并且在两种情况中几乎没有视觉上的损失。

针对各种不同的分层编码方法,我们可以给出一些一般的评论,这可能是有用处的。一般来讲,光靠错误复原这一理由,是不能论证为什么引入最复杂的分层编码方法的。事实上,如果这些方法可以满足其它的一些系统要求,则我们实际上是同时也利用它来进行错误复原。数据分割实现起来很简单,而且除了在错误率太高(大于 10% 的丢失率),或增强层数据完全丢失时,它可以提供几乎与其它几种方法同样的错误复原。SNR 分级要稍微复杂一些,且比数据分割的代码效率略低些,但当增强层不在时,它可以更简单地提供具有可用质量的低层信号。空域分级要更复杂一些,但在牺牲总体效率(两层)的情况下,可以提供很好的低层图象质量。

## 附录 II -E 框架和级别的限定

(本附录不作为本推荐|国际标准必备的一部分)

### II -E. 1 框架中的语法元素限定

本节将本规范中定义的语法元素制成表格。每项均被分类,以指出它是否需要由与特定的框架和级别兼容的解码器来支持。注意,标准的对于兼容性的规范在 ISO/IEC 13818-4 中给出。

注意 本节是非正式的,仅作为 II -8 章中表述的各种标准的限制的一个总结。如果,由于制备这份文本时所产生的错误,使得 II -8 章与附录 II -E 有出入,则以 II -8 章为准。

在表 II -E-2 到表 II -E-22 中。所用到的一些缩写符号列于表 II -E-1 中。

表 II -E-1 附录 II -E 的表格中所用的缩写符号

缩写	用于	含 义
x	状态	必须被解码器支持
o	状态	不必被解码器支持
D	类型	具有依靠级别的参数的项
I	类型	不依靠框架中级别的项
P	类型	用于解码后的后处理的项;解码器必须能对包含这些项的比特流进行解码,但它们的用处超出了本推荐 国际标准

表 II-E-2 序列头

#	语法元素	状态						类型	
		HIGH							
SPATIAL									
SNR									
MAIN									
SIMPLE									
01	horizontal_size_value	x	x	x	x	x	D	见表 I-8-7	
02	vertical_size_value	x	x	x	x	x	D	见表 I-8-7	
03	aspect_ratio_information	x	x	x	x	x	P		
04	frame_rate_code	x	x	x	x	x	D	见表 I-8-7 和 I-8-6	
05	(pel rate) Note: this is not a syntactic element						D	见表 8-8, 象素速率是象素/行, 行/帧, 帧/秒的产物	
06	bit_rate_value	x	x	x	x	x	D	见表 I-8-9	
07	vbv_buffer_size_value	x	x	x	x	x	D	见表 I-8-10	
08	constrained_parameters_flag	x	x	x	x	x	I	如果限制为 MPEG-1, 则为“1”; 如果为 MPEG-2, 则为“0”	
09	load_intra_quantiser_matrix	x	x	x	x	x	I		
10	intra_quantiser_matrix[64]	x	x	x	x	x	I		
11	load_non_intra_quantiser_matrix	x	x	x	x	x	I		
12	non_intra_quantiser_matrix[64]	x	x	x	x	x	I		
13	sequence_extension()	x	x	x	x	x	I	如果为 MPEG-2, 则存在	
14	sequence_display_extension()	x	x	x	x	x	P		
15	sequence_scalable_extension()	O	O	x	x	x	I	分级层的最大数见表 I-8-11	
16	user_data()	x	x	x	x	x	I	解码器可以跳过这些数据	

表 II-E-3 序列扩展

#	语法元素	状态					类型	
		HIGH						
		SPATIAL						
		SNR						
		MAIN						
		SIMPLE						
01	profile_and_level_indication	x	x	x	x	x	D 框架: 8 个值之一 级别: 16 个值之一 换码位: 2 个值之一	
02	progressive_sequence	x	x	x	x	x	I	
03	chroma_format	x	x	x	x	x	I 见表 I-8-5	
04	horizontal_size_extension	x	x	x	x	x	D 与输入图象大小有关	
05	vertical_size_extension	x	x	x	x	x	D 与输入图象大小有关	
06	bit_rate_extension	x	x	x	x	x	D 与输入图象大小有关	
07	vbv_buffer_size_extension	x	x	x	x	x	D 与输入图象大小有关	
08	low_delay	x	x	x	x	x	I	
09	frame_rate_extension_n	x	x	x	x	x	I 对于所有定义的框架均置 0	
10	frame_rate_extension_d	x	x	x	x	x	I 对于所有定义的框架均置 0	

表 II-E-4 序列显示扩展单元

#	语法元素	状态					类型	
		HIGH						
		SPATIAL						
		SNR						
		MAIN						
		SIMPLE						
01	video_format	x	x	x	x	x	P	
02	colour_description	x	x	x	x	x	P 与输入格式有关	
03	colour_primaries	x	x	x	x	x	P	
04	transfer_characteristics	x	x	x	x	x	P	
05	matrix_coefficients	x	x	x	x	x	P	
06	display_horizontal_size	x	x	x	x	x	P 与输入格式有关	
07	display_vertical_size	x	x	x	x	x	P 与输入格式有关	

表 II-E-5 序列分级扩展

#	语法元素	状态						类型	
		HIGH			SPATIAL				
		SNR			MAIN				
		SIMPLE							
01	scalable_mode	o	o	x	x	x	I	SNR 框架; SNR 分级 空域或高框架; SNR 或空域分级	
02	layer_id	o	o	x	x	x	I		
	假如“空间分级”								
03	lower_layer_prediction_horizontal_size	o	o	o	x	x	D	亮度采样密度见表 II-8-8	
04	lower_layer_prediction_vertical_size	o	o	o	x	x	D	亮度采样密度见表 II-8-8	
05	horizontal_subsampling_factor_m	o	o	o	x	x	I		
06	horizontal_subsampling_factor_n	o	o	o	x	x	I		
07	vertical_subsampling_factor_m	o	o	o	x	x	I		
08	vertical_subsampling_factor_n	o	o	o	x	x	I		
	假如“时间分级”								
09	picture_mux_enable	o	o	o	o	o	I		
10	mux_to_progressive_sequence	o	o	o	o	o	I		
11	picture_mux_order	o	o	o	o	o	I		
12	picture_mux_factor	o	o	o	o	o	I		

表 II-E-6 图组头

#	语法元素	状态					类型	
		HIGH						
		SPATIAL						
		SNR						
		MAIN						
		SIMPLE						
01	time_code	x	x	x	x	x	I	
02	closed_gop	x	x	x	x	x	I	
03	broken_link	x	x	x	x	x	I	

表 II-E-7 图象头

#	语法元素	状态					类型	
		HIGH						
		SPATIAL						
		SNR						
		MAIN						
		SIMPLE						
01	temporal_reference	x	x	x	x	x	I	
02	picture_coding_type	x	x	x	x	x	I 简单框架:I,P 主,SNR、空域或高框架: I,P, B	
03	vbv_delay	x	x	x	x	x	I	
04	full_pel_forward_vector	x	x	x	x	x	I 对于 MPEG-2 为“0”	
05	forward_f_code	x	x	x	x	x	I 对于 MPEG-2 为“111”	
06	full_pel_backward_vector	x	x	x	x	x	I 对于 MPEG-2 为“0”	
07	backward_f_code	x	x	x	x	x	I 对于 MPEG-2 为“111”	
08	extra_information_picture	x	x	x	x	x	I	
09	picture_coding_extension()	x	x	x	x	x	I	
10	quant_matrix_extension()	x	x	x	x	x	I	
11	picture_display_extension()	x	x	x	x	x	P	
12	picture_spatial_scalable_extension()	o	o	o	x	x	I	
13	picture_temporal_scalable_extension() ( )	o	o	o	o	o	I	

表 II-E-8 图象编码扩展

#	语法元素	状态						类型
		HIGH			SPATIAL			
SNR			MAIN			SIMPLE		
#	语法元素	x	x	x	x	x	D	注释
01	f_code[0][0](forward horizontal)	x	x	x	x	x	D	低级别[1 : 7] 主级别[1 : 8] 高-1440 或高级别[1 : 9]
02	f_code[0][1](forward vertical)	x	x	x	x	x	D	低级别[1 : 4] 主,高-1440 或高级别[1 : 5]
03	f_code[1][0](backward horizontal)	x	x	x	x	x	D	低级别[1 : 7] 主级别[1 : 8] 高-1440 或高级别[1 : 9]
04	f_code[1][1](backward vertical)	x	x	x	x	x	D	纸级别[1 : 4] 主,高-1440 或高级别[1 : 5]
05	intra_dc_precision	x	x	x	x	x	I	简单、主、SNR 或空域框架 [8 : 10] 高框架[8 : 11]
06	picture_structure	x	x	x	x	x	I	
07	top_field_first	x	x	x	x	x	I	
08	frame_pred_frame_dct	x	x	x	x	x	I	
09	concealment_motion_vectors	x	x	x	x	x	I	
10	q_scale_type	x	x	x	x	x	I	
11	intra_vlc_format	x	x	x	x	x	I	
12	alternate_scan	x	x	x	x	x	I	
13	repeat_first_field	x	x	x	x	x	I	
14	chroma_420_type	x	x	x	x	x	P	
15	progressive_frame	x	x	x	x	x	P	
16	composite_display_flag	x	x	x	x	x	P	
17	v_axis	x	x	x	x	x	P	
18	field_sequence	x	x	x	x	x	P	
19	sub_carrier	x	x	x	x	x	P	
20	burst_amplitude	x	x	x	x	x	P	
21	sub_carrier_phase	x	x	x	x	x	P	

表 II -E-9 量化矩阵扩展

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	load_intra_quantiser_matrix	x	x	x	x	x	I		
02	intra_quantiser_matrix[64]	x	x	x	x	x	I		
03	load_non_intra_quantiser_matrix	x	x	x	x	x	I		
04	non_intra_quantiser_matrix[64]	x	x	x	x	x	I		
05	load_chroma_intra_quantiser_matrix	o	o	o	o	x	I		
06	chroma_intra_quantiser_matrix[64]	o	o	o	o	x	I		
07	load_chroma_non_intra_quantiser_matrix	o	o	o	o	x	I		
08	chroma_non_intra_quantiser_matrix[64]	o	o	o	o	x	I		

表 II -E-10 图象显示扩展

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	frame_centre_horizontal_offset	x	x	x	x	x	P	与输入格式有关	
02	frame_centre_vertical_offset	x	x	x	x	x	P	与输入格式有关	

表 II-E-11 图象时域分级扩展

#	语法元素	状态						类型	
		HIGH							
SPATIAL									
SNR									
MAIN									
SIMPLE									
01	reference_select_code	o	o	o	o	o	I		
02	forward_temporal_reference	o	o	o	o	o	I		
03	backward_temporal_reference	o	o	o	o	o	I		

表 II-E-12 图象空域分级扩展

#	语法元素	状态						类型	
		HIGH							
SPATIAL									
SNR									
MAIN									
SIMPLE									
01	lower_layer_temporal_reference	o	o	o	x	x	I		
02	lower_layer_horizontal_offset	o	o	o	x	x	D	与输入格式有关	
03	lower_layer_vertical_offset	o	o	o	x	x	D	与输入格式有关	
04	spatial_temporal_weight_code_table_index	o	o	o	x	x	I		
05	lower_layer_progressive_frame	o	o	o	x	x	I		
06	lower_layer_deinterlaced_field_select	o	o	o	x	x	I		

表 II-E-13 组 块 层

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	slice_vertical_position_extension	x	x	x	x	x	D	与输入格式有关	
02	priority_breakpoint	o	o	o	o	o	I	仅数据分割需要这项	
03	quantiser_scale_code	x	x	x	x	x	I		
04	intra_slice	x	x	x	x	x	I		
05	extra_information_slice	x	x	x	x	x	I	解码器可能跳过这项数据	
06	macroblock()	x	x	x	x	x	I		

表 II-E-14 宏 块 层

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	macroblock_escape	x	x	x	x	x	I		
02	macroblock_address_increment	x	x	x	x	x	I		
03	macroblock_modes()	x	x	x	x	x	I		
04	quantiser_scale_code	x	x	x	x	x	I		
05	motion_vectors(0)	x	x	x	x	x	I	前向运动向量	
06	motion_vectors(1)	o	x	x	x	x	I	后向运动向量	
07	coded_block_pattern()	x	x	x	x	x	I		
08	block(i)	x	x	x	x	x	I		

表 II-E-15 宏块模式

#	语法元素	状态						类型	
		HIGH							
SPATIAL									
SNR									
MAIN									
SIMPLE									
01	macroblock_type	x	x	x	x	x	I		
02	spatial-temporal-weight_code	o	o	o	x	x	I		
03	frame_motion_type	x	x	x	x	x	I	01: 基于场的预测 10: 基于帧的预测 11: 双基	
04	field_motion_type	x	x	x	x	x	I	01: 10: 16×8MC 11: 双基	
05	dct_type	x	x	x	x	x	I		

表 II-E-16 运动向量

#	语法元素	状态						类型	
		HIGH							
SPATIAL									
SNR									
MAIN									
SIMPLE									
01	motion_vertical_field_select	x	x	x	x	x	I		
02	motion_vector()	x	x	x	x	x	I		

表 II-E-17 运动向量

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	motion_horizontal_code	x	x	x	x	x	I		
02	motion_horizontal_r	x	x	x	x	x	I		
03	dmv_horizontal	x	x	x	x	x	I		
04	motion_vertical_code	x	x	x	x	x	I		
05	motion_vertical_r	x	x	x	x	x	I		
06	dmv_vertical	x	x	x	x	x	I		

表 II-E-18 编码块模式

#	语法元素	状态						类型	
		HIGH							
		SPATIAL							
		SNR							
		MAIN							
		SIMPLE							
01	coded_block_pattern_420	x	x	x	x	x	I		
02	coded_block_pattern_1	o	o	o	o	x	I	4 : 2 : 2	
03	coded_block_pattern_2	o	o	o	o	o	I	4 : 4 : 4	

表 II-E-19 块 层

#	语法元素	状态						类型									
		HIGH															
SPATIAL																	
SNR																	
MAIN																	
SIMPLE																	
01	DCT 系数	x	x	x	x	x	I										
02	块结尾	x	x	x	x	x	I										

## II-E. 2 允许的层组合(见 II-8.4.1)

下面的表(II-E-20~II-E-22)指出了应用于比特流每一层的参数限制,以及一个能够对每一层完整解码的解码器的框架/级别。每个表描述了框架/级别矩阵中的一个单一兼容点。

### 高框架

在高框架表中,假定 intra\_dc\_precision = 11 仅与 4 : 2 : 2 chroma\_format 一同使用。放松这一限制会使最小解码器总为高框架解码器,且交互操作能力差。

表 II-E-20 高框架@主级别

层#	layer_id	分级方式	色差格式	最大采样密度	最大亮度采样率	最大整体比特率/ $10^6$	最大整体VBV缓冲区	最小解码器
1	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
1	0	—	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
2	0	—	4 : 2 : 0	MP@ML	HP@ML	15	1 835 008	HP@ML
2	1	SNR	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
2	0	—	4 : 2 : 0	MP@ML	HP@ML	15	1 835 008	HP@ML
2	1	SNR	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
2	0	—	4 : 2 : 2	MP@ML	HP@ML	15	1 835 008	HP@ML
2	1	SNR	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
2	0	—	4 : 2 : 0	MP@LL	MP@LL	4	475 136	MP@LL
2	1	空域	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	0	—	4 : 2 : 0	MP@LL	MP@LL	3	360 448	MP@LL
3	1	SNR	4 : 2 : 0	MP@LL	MP@LL	4	475 136	SNR@LL
3	2	空域	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	0	—	4 : 2 : 0	MP@LL	MP@LL	4	475 136	MP@LL
3	1	SNR	4 : 2 : 2	MP@LL	MP@LL	15	1 835 008	HP@ML
3	2	空域	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
3	0	—	4 : 2 : 0	MP@LL	MP@LL	4	475 136	MP@LL
3	1	空域	4 : 2 : 0	MP@ML	HP@ML	15	1 835 008	HP@ML
3	2	SNR	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML

注意 高框架的主级别比其它框架的主级别能够支持更高的分辨率,它可以支持 NTSC 的所有显示行(483 行),而其它框架的“主”级别最多支持 30Hz 帧速率的 480 行显示行。在高框架、主级别,不允许在空域分级体系中将 4 : 2 : 2 色差作为低层编码(见表 I-8-11)。

表 II-E-21 高框架@高-1440 级别

层 #	layer_id	分级方式	色差格式	最大采样密度	最大亮度采样率	最大整体比特率/ $10^6$	最大整体 VBV 缓冲区	最小解码器
1	0	—	4 : 2 : 0	MP@H-14	HP@H-14	80	9 781 248	HP@H-14
1	0	—	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
2	0	—	4 : 2 : 0	MP@H-14	HP@H-14	20	2 441 216	HP@H-14
2	1	SNR	4 : 2 : 0	MP@H-14	HP@H-14	80	9 781 248	HP@H-14
2	0	—	4 : 2 : 0	MP@H-14	MP@H-14	20	2 441 216	MP@H-14
2	1	SNR	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
2	0	—	4 : 2 : 2	MP@H-14	MP@H-14	20	2 441 216	MP@H-14
2	1	SNR	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
2	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	MP@H-14
2	1	空域	4 : 2 : 0	MP@H-14	HP@H-14	80	9 781 248	HP@H-14
2	0	—	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
2	1	空域	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	MP@H-14
3	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	SNR	4 : 2 : 0	MP@ML	HP@ML	60	7 340 032	HP@ML
3	2	空域	4 : 2 : 0	MP@H-14	HP@H-14	80	9 781 248	HP@H-14
3	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	SNR	4 : 2 : 2	MP@ML	HP@ML	60	7 340 032	HP@ML
3	2	空域	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
3	0	—	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	SNR	4 : 2 : 2	MP@ML	HP@ML	60	7 340 032	HP@ML
3	2	空域	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
3	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	空域	4 : 2 : 0	MP@H-14	HP@H-14	60	7 340 032	HP@H-14
3	2	SNR	4 : 2 : 0	MP@H-14	HP@H-14	80	9 781 248	HP@H-14
3	0	—	4 : 2 : 0	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	空域	4 : 2 : 0	MP@H-14	MP@H-14	60	7 340 032	HP@H-14
3	2	SNR	4 : 2 : 0	MP@H-14	MP@H-14	80	9 781 248	HP@H-14
3	0	—	4 : 2 : 2	MP@ML	HP@ML	20	2 441 216	HP@ML
3	1	空域	4 : 2 : 2	MP@H-14	MP@H-14	60	7 340 032	HP@H-14
3	2	SNR	4 : 2 : 2	MP@H-14	MP@H-14	80	9 781 248	HP@H-14

表 II -E-22 高框架@高级别

层 #	layer_id	分级方式	色差格式	最大采样密度	最大亮度采样率	最大整体比特率/ $10^6$	最大整体VBV缓冲区	最小解码器
1	0	—	4 : 2 : 0	MP@HL	MP@HL	100	12 222 464	HP@HL
1	0	—	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
2	0	—	4 : 2 : 0	MP@HL	HP@HL	25	3 047 424	HP@HL
2	1	SNR	4 : 2 : 0	MP@HL	HP@HL	100	12 222 464	HP@HL
2	0	—	4 : 2 : 0	MP@HL	MP@HL	25	3 047 424	HP@H-14
2	1	SNR	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
2	0	—	4 : 2 : 2	MP@HL	MP@HL	25	3 047 424	HP@HL
2	1	SNR	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
2	0	—	4 : 2 : 0	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
2	1	空域	4 : 2 : 0	MP@HL	HP@HL	100	12 222 464	HP@HL
2	0	—	4 : 2 : 2	HP@HL,LL	HP@ML	25	3 047 424	HP@H-14
2	1	空域	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 0	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
3	1	SNR	4 : 2 : 0	HP@HL,LL	HP@HL,LL	80	9 781 248	HP@H-14
3	2	空域	4 : 2 : 0	MP@HL	HP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 0	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
3	1	SNR	4 : 2 : 2	HP@HL,LL	HP@HL,LL	80	9 781 248	HP@H-14
3	2	空域	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 2	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
3	1	SNR	4 : 2 : 2	HP@HL,LL	HP@HL,LL	80	9 781 248	HP@H-14
3	2	空域	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 0	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
3	1	空域	4 : 2 : 0	MP@HL	HP@HL	80	9 781 248	HP@HL
3	2	SNR	4 : 2 : 0	MP@HL	HP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 0	HP@HL,LL	HP@HL,LL	25	3 047 424	HP@H-14
3	1	空域	4 : 2 : 0	MP@HL	MP@HL	80	9 781 248	HP@HL
3	2	SNR	4 : 2 : 0	MP@HL	MP@HL	100	12 222 464	HP@HL
3	0	—	4 : 2 : 2	HP@HL,LL	HP@ML	25	3 047 424	HP@H-14
3	1	空域	4 : 2 : 2	MP@HL	MP@HL	80	9 781 248	HP@HL
3	2	SNR	4 : 2 : 2	MP@HL	MP@HL	100	12 222 464	HP@HL

注意 ‘HP@HL,LL’ 表示采样密度/对于高框架,高级别,低层次的极限,限定的采样速率。



### **第三部分 音 频**



## III -0 引 言

### III -0.1 ISO/IEC11172-3 降低音频编码采样率的扩充

为了在非常低的比特率(<64kbit/s 每音频通道)下得到较好的音频质量,尤其是与CCITT 标准 G-722 的性能相比,ISO/IEC11172-3 的 I , II 和 III 层提供了三个附加的采样频率。附加的采样频率是 16kHz, 22.05kHz 和 24kHz。这使音频带宽可以接近 7.5kHz, 10.3kHz 和 11.25kHz。除了新定义的采样频率域、比特率索引域和比特分配表外,ISO/IEC 11172-3 的语法、语意和编码技术保持不变。如果在 ISO/IEC11172-3 头中的 ID 位等于零,则这些新的定义才有效。为了得到最好的音频性能,在解码中使用的听觉心理学模型的参数要相应修改。

在这些采样率下,音频帧的持续时间:

层	采样频率/kHz		
	16	22.05	24
I	24ms	17.41.. ms	16ms
II	72ms	52.24.. ms	48ms
III	36ms	26.12.. ms	24ms

### III -0.2 多通道音频的低比特率编码

#### III -0.2.1 通用多通道音频系统

单声道或立体声音频信号的低比特率编码标准是由 ISO/IEC11172-3 中的 MPEG-1 Audio 建立的。这个标准可用于存储介质或有限容量传输通道中的与图象信息联系的或单独的高质量数字化音频信号。

在仅需两个通道立体声时,ISO/IEC11172-3 音频编码标准可以和 MPEG-1 及 MPEG-2 Video 一起使用。MPEG-2 Audio(ISO/IEC 13818-3)提供了扩展到 3/2 多通道音频的能力和一个可选的低频增强通道(LFE)。

与传统的双声道音频系统相比,多通道音频系统提供了增强立体声音响的立体声性能。不仅在与画面配合的应用中,而且在仅有音频的应用中,人们普遍认为需要提高表现力。通用或兼容的多通道音频系统可用于卫星或陆地的电视广播,数字化音频广播(陆地或卫星)以及其它非广播媒介等,例如:

CATV 有线电视传播

CDAD 有限数字化音频传播

ENG	电子新闻采集(包括卫星新闻采集)
IPC	个人通信(视频会议、视频电话等)
ISM	交互式存储介质(光盘等)
NDB	网络数据库服务(通过 ATM 等)
DSM	数字化存储介质(数字化 VTR 等)
EC	电子电影院
HTT	家庭电视剧场
ISDN	综合业务数字网

似乎上面所述对制造商,生产者和顾客都非常有吸引力。

本文件描述了称为 ISO/多通道音频 MPEG 的音频编码系统。该系统可用于在存储介质或有限容量传输通道中传送高质量的数字化的多通道和/或多语种音频信息。其基本特征之一是与 ISO/IEC11172-3 编码的单声、立体声或双声道音频程序向后兼容。在 ISO/MPEG 音频小组和 ITU-R(先前的 CCIR)的特别小组 TG10/1,10/2 和 10/3 的考虑下,该系统的设计可用于不同的应用中。

### III -0. 2. 2 多通道音频的表示

#### III -0. 2. 2. 1 3/2 立体声加 LFE 格式

考虑到立体声音响的展示,ITU-R,SMPTE 和 EBU 的特别小组推荐使用一个附加的中央扬声器通道 C 和两个环绕扬声器通道 LS 和 RS,加上前面的左、右扬声器通道 L 和 R。这个音频格式称为“3/2-立体声”(3 个前端/2 个环绕扬声器通道),需要传输五个适当当地格式化的音频信号。

对于伴有画面的音频应用(如 HDTV),根据一般电影院中的经验,三个前面的扬声器通道保证了足够的方向稳定性和与画面相关的正面影象的清晰度。其最大的好处是“中心稳定”,这一点在任何位置都能得到保证,而其对大多数对话很重要。

另外,对于只有音频的应用,3/2-立体声格式可以提高双通道立体声的效果。附加的一对环绕扬声器通道提高了听觉环境的真实性。

一个低频增强通道(本文中称为 LFE 通道)可以选择性地加在任何一种配置中。该通道的目的是使选择该通道的听众在重放过程中,在频率和电平两方面扩展低频部分。这方面,与电影业在数字化声音系统中推出 LFE 通道的目的相同。

LFE 通道不能用于多通道声音播放的全部低频成分。LFE 通道是接收器中的可选项,因此只应含有可能具有高电平的低频声音效果。LFE 通道不包括在任何解码器的反矩阵操作中。LFE 通道的采样频率相当于主通道采样频率除以因子 96。在一个音频帧中有 12 个 LFE 采样值。LFE 通道能提供的信号范围从 15Hz 到 120Hz。

#### III -0. 2. 2. 2 兼容性

##### 向下兼容

在 ITU-R 建议 775“伴有画面或独立的多通道立体声音响音频系统,1992 年 11 月”中推荐了一个音频格式体系,该体系支持较低数目的扬声器和降低的表现性能(降到 2/0-立体声或者甚至单声),以及相应的向下混合方程集。可供选择的较低电平音频格式

有 3/1, 3/0, 2/2, 2/1, 2/0 和 1/0。这些可用于经济受限的或通道容量紧张的环境下。相应的扬声器安排是 3/2, 3/1, 3/0, 2/2, 2/1, 2/0 和 1/0。

#### 向后兼容性

在许多应用中, 经常采用非同时播送的方法传递附加的通道(中央, 环绕), 以达到扩展现有 2/0 立体声系统的目的。对现存接收器的这一向后兼容能力意味着矩阵兼容性的使用: 上一代的解码器必须重新生成两个常规的基本立体声信号  $L_o/R_o$ , 而多通道解码器从基本立体声信号和扩展信号中生成全部的 3/2 立体声, 为  $L'/C'/R'/LS'/RS'$ 。

#### III-0.2.2.3 多种语言能力

特别是对于 HDTV 的应用, 要求有多通道立体声性能和两种语言节目或多种语言解说词。这一标准提供了可在五通道声音系统中进行选择的音频通道配置。例如一个双语言 2/0 立体声节目或一个 2/0, 3/0 立体声加伴奏服务(如对听力困难者的“清晰对话”, 为视力受损者的解说词, 多种语言解说词等)。一个重要的配置是解说词对白(如通过中央扬声器)与一般音乐/效果立体声向下混合重现(例如记录影片, 体育报导)。

### III-0.2.3 多通道音频编码系统的基本参数

传输 3/2 声音系统的五个音频信号需要五个传输通道(但是对于比特率降低信号的内容, 则不必要互相独立)。为了使传输信号中的两个信号可独立提供立体声服务, 源声音信号在编码前被一个线性矩阵组合。组合后的信号(及其传输通道)以符号  $T_0, T_1, T_2, T_3$  和  $T_4$  区分。

#### III-0.2.3.1 与 ISO/IEC11172-3 的兼容性

多通道音频编码系统具有与 ISO/IEC11172-3 解码器向后和向前的兼容性。

对于多通道音频比特流, 向后兼容性是指 ISO/IEC11172-3 的解码器能正确解码基本的立体声信息。基本的立体声信息包括左通道和右通道, 它们组成了在所有通道中的音频信息的适当向下混合, 或者基本的立体声信息可以仅由多通道音频配置中的左声道和右声道组成。适当的向下混合方程由方程组(1)和(2), (3)和(4)及(5)和(6)给出。

$$L_o = L + x * C + y * LS \quad (1)$$

$$R_o = R + x * C + z * RS \quad (2)$$

或

$$L_o = L \quad (3)$$

$$R_o = R \quad (4)$$

或

$$L_o = L + x * C - y * jS \quad (5)$$

$$R_o = L + x * C + y * jS \quad (6)$$

其中,  $jS$  由计算  $LS$  和  $RS$  的单音成分得来, 带宽限制为 100~7000Hz, 半杜比 B 型编码, 90°相移。在出版此推荐|国际标准时, 通过使用方程(5)和(6)而与现有环绕声音解码器达到的兼容程度尚未得到验证。

向前兼容性是指 MPEG2 多通道音频解码器可以正确解码 ISO/IEC11172-3 音频比特流。

有可能为以下组合:

基本 Lo,Ro 立体声	多通道扩展
层 I	层 I mc
层 II	层 II mc
层 I	层 I mc

上表描述了基本的 Lo,Ro 立体声层 I , II 和 III 以及层 I mc 和层 II mc 的多通道扩展的组合。

ISO/MPEG 音频多通道系统与 ISO 标准 11172-3 全兼容。此兼容性由对基本立体声信息进行编码以适应 ISO/IEC11172-3, 同时开发 ISO/IEC11172-3 音频帧的辅助数据域以进行多通道扩展来实现。

完整的 ISO/IEC 11172-3 帧包含以下四种不同的信息：

- ISO/IEC 11172-3 音频帧的前 32 位的头信息。
- 循环冗余校验(CRC), 由 16 位组成, 紧跟在头信息后面(可选)。
- 音频数据, 对于层 I , 由位分配(BAL)、规格因子选择信息(SCFSI)、规格因子(SCF)和子带样本组成。
- 辅助数据, 由于大量的不同应用, 采用 ISO/IEC 11172-3 标准, 此域的长度和使用未作规定。

辅助数据域长度可变, 可以把通道 T2/T3/T4 的全部扩展信息压缩到辅助数据域的第一部分。如果 MC 编码器没有使用全部的辅助数据域进行多通道扩展信息的编码, 剩余的部分可用于其它辅助数据。

多通道扩展信息要求的比特率, 根据声音信号随帧的基准改变。总的比特率是固定的, 由 ISO/IEC 11172-3 规定。

### III -0. 2. 3. 2 音频输入/输出格式

采样率: 48,44.1 或 32kHz

量化: 最多 24bit/样本 PCM 分辨率

下列音频通道的组合可用于音频编码器的输入:

(1) 五通道, 用 3/2 配置

L,C,R 加两个环绕通道 LS,RS

(2) 五通道, 3/0+2/0 配置

第一个节目的 L,C,R 加第二个节目的 L2,R2

(3) 四通道, 3/1 配置

L,C,R 加单环绕通道 S

(4) 四通道, 2/2 配置

L,R 加双环绕通道 LS,RS

(5) 四通道, 2/0+2/0 配置

第一个节目的 L,R 加第二个节目的 L2,R2

(6) 三通道, 3/0 配置

L,C,R 无环绕

(7) 三通道, 2/1 配置

L, R 带单环绕通道 S

(8) 双通道, 2/0 配置

立体声或双通道方式(如 ISO/IEC11172-3)

(9) 单通道, 1/0 配置

单通道方式(如 ISO/IEC11172-3)

不同的音频输入信号的组合最多以五个通道进行编码和传输。五个通道 T0, T1, T2, T3 和 T4 中, T0 和 T1 是 ISO/IEC 11172-3 中的两个基本通道。传送向后兼容的信号 L<sub>0</sub>, R<sub>0</sub>。传输通道 T2, T3 和 T4 一起形成了多通道扩展信息, 在 ISO/IEC 11172-3 辅助数据域中一起传输。

多通道解码后, 最多五个音频通道被恢复, 可由听者选择表示为任何方便的格式:

(1) 五个通道: 3/2 配置

前向: 左(L)和右(R)通道加中央通道(C)

环绕: 左环绕(LS)和右环绕(RS)

(2) 四个通道: 3/1 配置

前向: 左(L)和右(R)通道加中央通道(C)

环绕: 单环绕(S)

(3) 四通道: 2/2 配置

前向: 左(L)和右(R)通道

环绕: 左环绕(LS)和右环绕(RS)

(4) 三通道: 2/1 配置

前向: 左(L)和右(R)通道

环绕: 单环绕(S)

(5) 三通道: 3/0 配置

前向: 左(L)和右(R)通道加中央通道(C)

环绕: 无环绕

(6) 双通道: 2/0 配置

前向: 左(L)和右(R)通道

环绕: 无环绕

(7) 一个通道: 1/0 配置

前向: 单通道(Mo)

环绕: 无环绕

对以上任何配置可以任意附加一个低频增强通道。

输出信号有可能要求是离散信号, 或按向下混合, 或根据 ITU-R 推荐 775 定义的向上校正方程进行组合。

### III-0.2.3.3 混合编码方式

#### 动态传输开关

为了使一起传输的信号 T0 和 T1 以及三个附加传输信号 T2, T3 和 T4 之间有更好

的互不相关性,灵活地选择 T2,T3 和 T4 通道是必要的。与频率范围无关,ISO/IEC 13818-3 允许在五个信号 L,C,R,Ls,Rs 中任选三个在 T2,T3,T4 中传输。

#### **动态交叉会话**

根据双耳听觉模型,有可能确定立体声信号部分而不管立体声播放的空间感觉。立体声-无关信号的组成部分未被屏蔽,但它们对声源定位不会有帮助。在人类声音系统的双耳处理器中,它们被忽略了。因此,任何立体声信号(L,C,R,Ls 或 Rs)的立体声-无关部分可以通过任何一个扬声器或配置中的几个扬声器重放,而不会影响立体声效果。这可以在许多频率区域内独立做到。

#### **自适应多通道预测**

为了利用统计学上的通道间相关性,用自适应多通道预测来减少冗余。在传输通道 T2,T3,T4 中不是传输实际的信号,而是相应的预测误差信号。预测器是二阶带延时补偿的。

#### **中央幻象编码**

由于人类音频系统仅使用音频信号中的强刺激进行高频的定位,因此有可能在前向的左和右声道中传输中央声道的高频部分,在中央扬声器的位置构成一个幻象源。

#### **III-0.2.3.4 编码解码参数**

编码、解码与 ISO/IEC 11172-3 类似。

编码方式:

3/2,3/0+2/0,3/1,2/0+2/0,3/0,2/2,2/1,2/0,1/0

第二立体声节目

最多 7 个附加多语言或解说词通道

联合服务

子带滤波器变换:

子带数目: 32

采样频率:  $F_s/32$

子带带宽:  $F_s/64$

MDCT 的附加分解(仅层 III)

频率分辨率: 每个子带的 6 或 18 个部分

LFEC 滤波器变换:

LFEC 个数: 1

采样频率:  $F_s/96$

LFEC 带宽: 125Hz

动态范围: 多于 20bit

## **III - 1 音频概述**

### **III - 1. 1 范 围**

本推荐|国际标准规定了 ISO/IEC 11172-3 的扩展,用于低频率采样、广播、传输和存储介质上的多通道高质量音频的编码表示,以及多通道高质量音频信号的解码方法。编码器的输入和解码器的输出与现有的 PCM 标准兼容。

### **III - 1. 2 标 准 参 考**

后续的 ITU-T/CCITT 推荐及国际标准包含的条目通过参照本文档来构成本推荐|国际标准的条目。在出版时,指明的版本是有效的。所有的推荐和标准都需要改正。我们鼓励赞同本推荐|国际标准的团体来调查下面列出的标准的最新版本的应用可能性。IEC 和 ISO 的成员保存有当前有效的国际标准的注册。ITU 的电信标准化局保存有当前有效的 ITU-T/CCITT 建议的清单。

#### **III - 1. 2. 1 相同的推荐|国际标准**

#### **III - 1. 2. 2 两个推荐|国际标准在技术内容上的等价性**

#### **III - 1. 2. 3 附加参考文献**

ISO/IEC 11172 “Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1. 5Mbit/s” Part 3 “Audio”

Recommendation 775 of CCIR 1992, “Multichannel stereophonic sound system with and without accompanying picture”

Recommendations and reports of the CCIR, 1990

XVIIth Plenary Assembly, Düsseldorf, 1990

Volume XI-Part1

Broadcasting Service(Television)

Rec. 601-1 “Encoding parameters of digital television for studios”

CCIR Volume X and XI Part 3

Recommendation 648: Recording of audio signals

CCIR Volume X and XI Part3

Report 955-2: Sound Broadcasting by satellite for portable and mobile receivers, including Annex IV Summary description of Advanced Digital System II . IEEE Draft Standard “Specification for the implementation of 8×8 inverse discrete cosine transform”

P1180/D2, July 18, 1990

IEC publication 908 : 198 “CD Digital Audio System”

## III-2 技术单元

### III-2.1 定义

本推荐|国际标准使用如下定义。若指定为适用于某一部分，在方括号内注明。

- (1) **ac 系数[视频]**: 所有在一维或两维内频率非零的 DCT 系数。
- (2) **访问单元[系统]**: 音频压缩时, 访问单元为音频访问单元。视频压缩时, 访问单元为一幅图象的编码表示。
- (3) **自适应分段[音频]**: 在可变时间段的音频信号的数字化表示的细分。
- (4) **自适应位分配[音频]**: 根据听觉心理模型, 在时间和频率变化方式的子带上的位的分配。
- (5) **自适应多通道预测[音频]**: 利用通道间统计学上的相关性降低多通道数据量的方法。
- (6) **自适应噪音分配[音频]**: 根据听觉心理模型在时间和频率变化的频带编码噪音的分配。
- (7) **别名[音频]**: 子-奈奎斯特采样的镜象信号成分。
- (8) **分解滤波槽[音频]**: 将宽带 PCM 音频信号变换为次采样的子带样本集的编码器的滤波槽。
- (9) **音频访问单元[音频]**: 对于层 I 和层 II, 音频访问单元定义为可自己解码的编码比特流的最小部分, 其中解码意味着声音的完全重建。对于层 II, 音频访问单元是可利用先前获得的主要信息进行解码的比特流部分。
- (10) **音频缓冲器[音频]**: 在系统目标解码器中存储压缩的音频数据的缓冲器。
- (11) **音频序列[音频]**: 一个不中断的音频帧序列, 其中的以下参数未改变:
  - ID
  - 层
  - 采样频率
  - 层 I 和 II 的比特率索引
- (12) **向后运动向量[视频]**: 用于从播放时间在后的参考画面作运动补偿的运动向量。
- (13) **巴克[音频]**: 临界波段率单元。巴克标尺是音频范围的频率标尺的非线性映象。基本符合人耳的频率选择性。
- (14) **双向预测编码图象; B-图象[视频]**: 利用以前和/或以后参考图象作运动补偿编码的图象。
- (15) **比特率**: 从存储介质到解码器输入的压缩的比特流的传输率。
- (16) **块压扩[音频]**: 在一定时间内的音频信号的数字化表示的标准化。

- (17) **块[视频]**: 8 行 8 列的正交象素块。
- (18) **界限[音频]**: 使用强立体声编码的最小部分波段。
- (19) **字节**: 8 位的序列。
- (20) **字节对齐**: 在编码比特流中,若某位的位置是 8 位的倍数,则其是字节对齐的。
- (21) **中央通道[音频]**: 用于稳定正面立体声映象中央单元的音频表现通道。
- (22) **通道**: 存储或传输 CD13818 比特流的数字化介质。
- (23) **通道[音频]**: 重现被传输的音频信号的数据序列。
- (24) **色度(分量)[视频]**: 一个矩阵或块或单一象素按 CCIR Rec 601 定义的方式表示与原色相关的两色差信号之一。
- (25) **编码音频比特流[音频]**: 在 CD 部分规定的音频信号的编码表示。
- (26) **编码视频比特流[视频]**: 在 CD 中规定的一幅或多幅图象的序列的编码表示。
- (27) **编码顺序[视频]**: 图象存储和解码的顺序。此顺序不必与播放顺序相同。
- (28) **编码表示**: 以编码形式表示的数据元素。
- (29) **编码参数[视频]**: 用户可定义的用以表示编码的视频比特流特性的参数集。比特流的特性用编码参数描述。解码器以其能解码的比特流为特性。
- (30) **分量[视频]**: 一个矩阵、块或单一象素用以构成一幅图象。它们来自三个矩阵(亮度和两个色度)中之一。
- (31) **压缩**: 减少表示一个数据项的位的数量。
- (32) **固定比特率视频编码[视频]**: 具有固定的平均比特率的压缩的视频比特流。
- (33) **固定比特率**: 从压缩比特流开始到结束,比特率固定的操作。
- (34) **约束参数[视频]**: ISO/IEC 11172-2 的 II -2.4.3.2 节中定义的编码参数集的值。
- (35) **受限系统参数流(CSPS)[系统]**: ISO/IEC 11172 多路流。在 ISO/IEC 11172-1 应用中定义了其约束条件。
- (36) **CRC**: 循环冗余校验。
- (37) **临界波段率[音频]**: 频率的听觉心理功能。在给定的可闻频率下,它与低于此频率的临界波段数成比例。临界波段率标尺的单位是巴克。
- (38) **临界波段[音频]**: 在谱域中与人耳的频率选择性相对应的听觉心理学量度。此选择性用巴克表示。
- (39) **数据元素**: 编码前及解码后表示的数据项。
- (40) **dc-系数[视频]**: 在二维中频率均为零的 DCT 系数。
- (41) **dc 编码图象;D 图象[视频]**: 仅用自身信息进行编码的图象。在编码表示的 DCT 系数中,只有 dc 系数。
- (42) **DCT 系数**: 确定的余弦基函数的幅值。
- (43) **解码流**: 对压缩比特流的解码重组。
- (44) **解码器输入缓冲[视频]**: 在视频缓冲检测器中规定的先进先出(FIFO)缓冲器。
- (45) **解码器输入率[视频]**: 视频缓冲校验器中说明的且在编码视频比特流中被编

码的数据率。

(46) **解码器**: 解码过程的具体化。

(47) **解码(过程)**: 在 ISO/IEC 11172 中定义的过程,读入编码比特流,产生解码的图象或音频样本。

(48) **解码时间标记;DTS[系统]**: 可能出现于包头的域,用于指示在系统目标解码器中访问单元被解码的时间。

(49) **反加强[音频]**: 在存储或传输后对音频信号进行的滤波,用以消除加强时产生的线性偏差。

(50) **反量化[视频]**: 当量化后的 DCT 系数在比特流中的表示被解码后,而未进行反 DCT 之前进行的变化过程。

(51) **数字存储介质; DSM**: 数字化存储器或传输设备或系统。

(52) **离散余弦变换;DCT[视频]**: 或正向离散余弦变换,或反向离散余弦变换。DCT 是可逆的、离散正交变换。反向 DCT 在 ISO/IEC 11172-2 附录 A 中定义。

(53) **播放顺序[视频]**: 解码的图象被播放的顺序。通常与编码器输入的顺序一样。

(54) **向下混合[音频]**: 对 n 个通道进行矩阵变换得到少于 n 的通道数。

(55) **双通道方式[音频]**: 一种将两个有独立节目内容(例如两种语言)的音频通道编码成一个比特流的方式。编码过程与立体声方式相同。

(56) **动态交叉会话[音频]**: 一种将立体声-无关信号成分拷贝到另一通道而实现多通道数据减少的方法。

(57) **动态传输通道切换[音频]**: 一种多通道数据减少方法,通过将正交信号成分分配到传输通道来实现。

(58) **编辑**: 用一个或多个压缩比特流产生新的压缩比特流的过程。编辑后的合适比特流必须满足 CD 中定义的要求。

(59) **基本流[系统]**: 一个编码的视频、音频或其它比特流的通项。

(60) **加强[音频]**: 在音频信号存储或传输前,进行的滤波,以提高高频部分的信噪比。

(61) **编码器**: 编码过程的具体化。

(62) **编码(过程)**: 本 CD 中定义的读入一个输入图象或音频样本流,产生有效的编码比特流的过程。本 CD 中未给出说明。

(63) **熵编码**: 对数字化表示的信号的长度可变的无损失编码,以减少冗余。

(64) **快进[视频]**: 按播放顺序比实时快速地播放一组或一组的部分图象。

(65) **FFT**: 快速傅里叶变换。离散傅里叶变换(一种正交变换)的一种快速算法。

(66) **滤波器组[音频]**: 覆盖整个音频频率范围的带通滤波器组。

(67) **固定分段[音频]**: 将数字化表示的音频信号细分成固定的时间段。

(68) **禁止**: “禁止”项被用来定义编码比特流,指示此值不会被使用。通常用来避免竞争起始码。

(69) **强制更新[视频]**: 宏块的从时域到时域的内部编码过程,保证在编码器与解码器间不会产生过多的不匹配误差。

- (70) **前向运动向量[视频]**: 以时间在前的参考图象作运动补偿的运动向量。
- (71) **帧[音频]**: 对应于音频访问单元的音频 PCM 样本的一部分音频信号。
- (72) **自由格式[音频]**: 除定义的比特率之外的任何比特率,都小于每层的最大有效比特率。
- (73) **未来参考图象[视频]**: 未来参考图象是按播放顺序在当前图象之后产生的参考图象。
- (74) **颗粒[层 II ][音频]**: 量化前整体考虑的全部 32 个次能带中的 25 个连续次能带样本集。它们对应于 96 个 PCM 样本。
- (75) **颗粒[层 III ][音频]**: 自身携带附带信息的 57 条频率线。
- (76) **图象组[视频]**: 用于辅助随机访问的一个或多个编码图象序列。在 ISO/IEC 11172-2 中定义的编码语法中,图象组是一个层。
- (77) **汉宁窗[音频]**: 对一块音频样本,在进行傅氏变换前逐点使用的时间函数。
- (78) **哈夫曼编码**: 熵编码的特殊方法。
- (79) **混合型滤波器组[音频]**: 次能带滤波器组与 MDCT 的组合集。
- (80) **IMDCT[音频]**: 反向改进离散余弦变换。
- (81) **强立体声[音频]**: 基于在高频部分仅保留左右声道的能量包络,利用立体声音频节目中的立体声无关性或冗余的方法。
- (82) **隔行[视频]**: 普通电视图象的特性。图象上交替的线表示时间上的不同瞬间。
- (83) **内部编码[视频]**: 对宏块或图象进行仅利用自身信息的编码。
- (84) **内部编码图象;I-图象[视频]**: 仅使用自身信息进行编码的图象。
- (85) **ISO/IEC 11172(多路)流[系统]**: 由零个或多个基本流按 ISO/IEC 11172-1 定义的方式组合成的比特流。
- (86) **联合立体声编码[音频]**: 利用立体声无关性或立体声冗余的编码。
- (87) **联合立体声模式[音频]**: 使用联合立体声编码算法的音频编码方法。
- (88) **层[音频]**: 在 CD 部分定义的音频系统的一个编码层次。
- (89) **层[视频和系统]**: 在 ISO/IEC 11172-1 和 ISO/IEC 11172-2 中定义的视频和系统规范的数据体系中的一层。
- (90) **低频增强通道[音频]**: 在多通道系统中用于低频效果的有限带宽通道。
- (91) **亮度(成分)[视频]**: 一个矩阵、块或单个象素按 CCIR Rec 601 定义的方式表示单色的信号和与基色相关的信号。亮度的符号是 Y。
- (92) **宏块[视频]**: 四个  $8 \times 8$  的亮度数据块和两个相应的  $8 \times 8$  的色度数据块组成图象亮度分量的  $16 \times 16$  块。宏块有时指象素数据,有时指象素值的编码表示和其它数据元素。由 ISO/IEC 11172-2 的宏块层的语法定义而定。根据上下文可以清楚地知其用途。
- (93) **映象[音频]**: 音频信号从时域到频域通过部分波段滤波与/或 MDCT 进行转换。
- (94) **掩盖[音频]**: 人类听觉系统的特性:一个音频信号在另一音频信号存在的情况下,无法被感知。
- (95) **掩盖阈值[音频]**: 频率和时间的函数低于该函数,人耳即无法感受到音频

信号。

(96) **MDCT[音频]**: 改进离散余弦变换,与时域中的“别名取消滤波器组”对应。

(97) **运动补偿[视频]**: 用运动向量来改善对象素值的预测效率。预测利用运动向量产生预先解码的象素值在过去或将来参考图象中的偏移值,以得到预测误差信号。

(98) **运动估计[视频]**: 在编码过程中估计运动向量的过程。

(99) **运动向量[视频]**: 用于运动补偿的二维向量,是从当前图象坐标位置到参考图象坐标位置的偏移量。

(100) **MS 立体声[音频]**: 一种利用立体声音频节目中的立体声无关性和冗余性的方法。对总和及差分信号编码而非对左右两个通道编码。

(101) **多通道[音频]**: 多个音频通道的组合,以生成空间声音场。

(102) **多语种[音频]**: 用不止一种语言来表现对话。

(103) **非内部编码[视频]**: 对宏块或图象编码,不仅使用自身信息,也使用其它时间产生的宏块和图象的信息。

(104) **非音色成分[音频]**: 音频信号的类噪声成分。

(105) **奈奎斯特采样**: 以信号带宽的两倍或以上的频率采样。

(106) **组[系统]**: 一个组包括组头和后面跟随的一个或多个分组。它是 ISO/IEC 11172-1 中系统编码语法描述的一个层。

(107) **分组数据[系统]**: 分组包的基本流中相邻的数据字节。

(108) **分组包头[系统]**: 用于表达分组数据中基本流信息的数据结构。

(109) **分组包[系统]**: 分组包包括一个分组包头及基本数据流中的相邻字节。它是 ISO/IEC11172-1 系统编码语法描述的一个层。

(110) **填充[音频]**: 通过在音频帧中有条件地加入间隔,将音频帧的平均时长调整为相应的 PCM 样本脉冲宽度的方法。

(111) **过去参考图象[视频]**: 过去参考图象是在播放顺序上先于当前图象的参考图象。

(112) **象素纵横比[视频]**: 象素在播放时垂直高度与水平宽度的比率。

(113) **象素[视频]**: 图象元素。

(114) **图象周期[视频]**: 图象率的倒数。

(115) **图象率[视频]**: 从解码过程输出的图象的速率。

(116) **图象[视频]**: 源的编码的或重建的影象数据。一个源的或重建的图象由三个 8 位数字的矩阵组成,代表亮度和两个色度信号。图象层是本 CD 的编码语法定义的一个层。注意,在本 CD 中术语“图象”总比术语“帧”或“场”优先使用。

(117) **多相滤波器组[音频]**: 一组有相同带宽、特殊相位关系的滤波器,允许滤波器组的高效实现。

(118) **预测[音频]**: 用预测器由其它通道的部分波段样本预测一个通道的部分波段样本。

(119) **预测[视频]**: 用预测器估算当前被解码的象素值或数据元素。

(120) **预测编码图象;P-图象[视频]**: 用过去的参考图象进行运动补偿预测编码的

图象。

- (121) **预测误差[视频]**: 一个像素或数据元素的实际值与预测值的差。
- (122) **预测器[视频]**: 先前解码的像素值或数据元素的线性组合。
- (123) **表现通道[音频]**: 解码器输出的音频通道, 对应于扬声器的左、中、右, 左环绕和右环绕。
- (124) **表现时间标记; PTS[系统]**: 有可能在分组包头中出现的域, 指示在系统目标解码器中表现单元出现的时刻。
- (125) **表现单元; PU[系统]**: 解码的音频访问单元或解码的图象。
- (126) **听觉心理模型[音频]**: 人类听觉系统的掩蔽行为的数学模型。
- (127) **量化矩阵[视频]**: 用于反量化器的一组 64 个 8 位值。
- (128) **量化 DCT 系数[视频]**: 反量化前的 DCT 系数。量化 DCT 系数的可变长编码表示作为压缩视频比特流的一部分存储。
- (129) **量化器比例因子[视频]**: 在比特流中描述的数据元素, 用于解码过程中的按比例反量化。
- (130) **随机访问**: 从任意点开始对编码比特流进行读取和解码的过程。
- (131) **参考图象[视频]**: 参考图象是按播放顺序与当前图象最邻近的 I- 或 P- 图象。
- (132) **重排缓冲区[视频]**: 在系统目标解码器中的缓冲区, 用于存储重建的 I- 图象或重建的 P- 图象。
- (133) **再量化[音频]**: 对编码的部分波段样本进行解码以恢复原始的量化值。
- (134) **保留**: 在定义编码比特流的项中用到“保留”, 指明此值可能在将来的 ISO/IEC 扩展中被用到。
- (135) **倒放[视频]**: 按播放的倒转顺序播放图象的过程。
- (136) **比例因子频带[音频]**: 在层 II 中用一个比例因子变比的一组频率线。
- (137) **比例因子索引[音频]**: 比例因子的一个数值码。
- (138) **比例因子[音频]**: 量化前一组值按此因子作比例变换。
- (139) **序列头[视频]**: 编码比特流中的一块数据, 包括一些数据元素的编码表示。
- (140) **附带信息**: 比特流中控制解码器的必要的信息。
- (141) **跳过的宏块[视频]**: 没有存数据的宏块。
- (142) **组块[视频]**: 一系列宏块。它是 ISO/IEC 11172-2 的编码语法定义的一个层。
- (143) **槽[音频]**: 槽是比特流的基本部分。在层 I, 一个槽等于 4 个字节, 在层 II 和 III 为一个字节。
- (144) **源流**: 压缩编码前的单一的非多路流。
- (145) **分布函数[音频]**: 描述掩蔽效应的频率分布的函数。
- (146) **起始编码[系统和视频]**: 嵌入编码比特流的 32 位唯一编码。有多种用途, 包括在编码语法中区分一些层。
- (147) **STD 输入缓冲区[系统]**: 系统目标解码器的一个先进先出缓冲区, 用于存储解码前的基本流的压缩数据。
- (148) **立体声-无关[音频]**: 立体声音频信号中对空间感无贡献的部分。

(149) **立体声方式[音频]**: 一种方式,其中构成“立体声对”(左和右)的两个音频通道被编码于一个比特流中。编码过程与双通道方式一样。

(150) **填充(位);填充(字节)**: 可以插入到压缩的比特流中,而在解码过程中被去除的码字。填充的目的是为了增加流的比特率。

(151) **子波段[音频]**: 音频频率波段的细分。

(152) **子波段滤波器组[音频]**: 一组覆盖整个音频频率范围的带通滤波器。在此 CD 的这部分,子波段滤波器组是一多相滤波器组。

(153) **子波段样本[音频]**: 音频编码器中的子波段滤波器组产生滤波的输入音频流的子采样表示。滤波后的样本叫子样本。从 384 个时间连续的输入音频样本中,32 个部分波段中的每个都有 12 个连续的子样本。

(154) **环绕通道[音频]**: 附加于前向通道(L 和 R 或 L,R 和 C)的音频表现通道,用 来增强空间感。

(155) **同步字[音频]**: 嵌入在音频比特流中的 12 位码,用以确定帧的开始。

(156) **合成滤波器组[音频]**: 解码器中的滤波器组,用来从子波段样本中重构 PCM 音频信号。

(157) **系统头[系统]**: 系统头是在 ISO/IEC 11172-1 中定义的数据结构,携带的信息 概括了 ISO/IEC 11172 多路流的系统特性。

(158) **系统目标解码器;STD[系统]**: 解码过程的假设参考模型,用以描述 ISO/IEC 11172 多路比特流的语义。

(159) **时间标记[系统]**: 指示一个事件的时刻的一个项。

(160) **三联体[音频]**: 一个子波段中的三个连续子样本。32 个部分波段中的任一三 联体构成一个颗粒。

(161) **音色成分[音频]**: 音频信号的正弦状成分。

(162) **可变比特率**: 在对压缩比特流解码期间比特率随时间变化的操作。

(163) **可变长编码;VLC**: 可逆的编码过程。对频率高的事件用较短的码字,对频率 低的事件用较长的码字。

(164) **视频缓冲检测器;VBV[视频]**: 假想的解码器,在概念上与编码器输出相连。 目的是提供对编码器或编辑过程可能产生的数据率的可变性的约束。

(165) **视频序列[视频]**: 一个或多组图象的序列。是 ISO/IEC 11172-2 编码语法定义 的一个层。

(166) **Z 形扫描顺序[视频]**: 一特定的 DCT 系数顺序,(近似)从最低频率到最高频 率。

## III -2. 2 符号和缩写

本推荐 | 国际标准使用的数学运算符号与 C 语言类似。但取整和四舍五入的整数除 法是特殊定义的。位操作定义假设有两个整数成分。编号和计数循环从零开始。

### III -2. 2. 1 算术运算符

+	加
-	减(二元运算)或负(一元运算)
++	增加
--	减少
*	乘
^	幂
/	整数除法,结果向零取整。例如, $7/4$ 和 $-7/-4$ 取整为 1, $-7/4$ 和 $7/-4$ 取整为 -1。
//	整数除法,结果取最近的整数。除非特殊说明,半整数值舍入背向零。例如: $3//2$ 为 2, $-3//2$ 为 -2。
DIV	整数除法,结果向 $-\infty$ 取整
	绝对值 $ x  = x$ 当 $x > 0$ $ x  = 0$ 当 $x = 0$ $ x  = -x$ 当 $x < 0$
%	取模 只对正数有定义
Sign()	取符号 $\text{Sign}(x) = 1$ 当 $x > 0$ $\text{Sign}(x) = 0$ 当 $x = 0$ $\text{Sign}(x) = -1$ 当 $x < 0$
MINT()	最近整数。返回实值参数的最近的整数值。半整数值背向零舍入。
sin	正弦
cos	余弦
exp	指数
$\sqrt{\phantom{x}}$	平方根
$\log_{10}$	10 为底的对数
$\log_e$	e 为底的对数
$\log_2$	2 为底的对数

### III -2. 2. 2 逻辑运算符

	逻辑或
&&	逻辑与
!	逻辑非

### III -2. 2. 3 关系运算符

>	大于
$\geq$	大于等于
<	小于

<= 小于等于  
 == 等于  
 != 不等于  
 $\max[\dots]$  参数表中的最大值  
 $\min[\dots]$  参数表中的最小值

### III -2. 2. 4 位操作符

使用位操作符假设有两个操作数。

& 与  
 | 或  
 >> 带符号扩展右移  
 << 补零左移

### III -2. 2. 5 赋值

= 赋值运算符

### III -2. 2. 6 助记符

下面定义的助记符用以描述编码比特流中的不同数据类型。

bslbf	位串, 左位在前, 其中左按 ISO/IEC 11172 中写的位串的顺序。位串是带单引号的 1 和 0 串。如 '1000 0001'。位串内的空格是便于阅读的, 无特殊意义
center_chan	中央通道索引
center_limited	变量, 指示中央子波段是否未传输。用于中央通道的幻象编码情况下。
ch	通道。如果 ch 值为 0。表示立体信号的左通道或两个独立通道的第一个通道。(音频)
gr	在音频层 I 中为 $3 * 32$ 个子波段样本的颗粒。在音频层 II 中为 $18 * 32$ 子样本的颗粒。(音频)
L,C,R,LS,RS	左、中、右、左环绕和右环绕音频信号
$L^w, C^w, R^w, LS^w, RS^w$	加权左、中、右、左环绕和右环绕音频信号。加权出于两个原因: ① 所有信号在经过编码时, 为避免计算兼容立体声信号时过载而被消弱。② 矩阵方程中包含有削弱因子, 以及其它过程, 如相移。 实际编码和传输的信号是加权和处理过的。在译码器中恢复正常
left_sur_chan	左环绕通道索引
main_data	比特流的 main_data 部分包含比例因子、哈夫曼编码数据和辅助信息。(音频)

mono_sur_chan	单环绕通道索引。此索引与左环绕通道索引为同一索引。
msblimit	使用的最大子波段。
nch	通道数。单通道方式为 1, 其它方式为 2。(音频)
nmch	多通道扩展部分的通道数
dyn_cross	表示对一定的传输通道和子波段使用了动态交叉会话
npredcoeff	使用的预测系数个数
part2_length	main_data 中用于比例因子的位数。(音频)
right_sur_chan	右环绕通道索引
rpchof	余多项式系数, 高阶在前。(音频)
sb	子波段。(音频)
sbgr	根据子项 III-2.5.2.10 中子波段组表的独立的子波段组
sblimit	最低子波段数。未分配位。(音频)
scfsi	比例因子选择信息。(音频)
switch_point_l	使用窗口开关时用到的比例因子频带数(长块比例因子频带)。(音频)
switch_point_s	使用窗口开关时用到的比例因子频带数(短块比例因子频带)。(音频)
tc	传输通道。(音频)
uimsbf	无符号整数, 高有效位在前
vlclbf	变长编码, 左位在前, 其中“左”系指 VLC 码的书写顺序
window	当 block_type == 2, 0 <= window <= 2 时, 实际时间槽数目。(音频)
多字节字的顺序是高有效字节在前。	

### III-2.2.7 常量

$\pi$	3.14159265358...
e	2.71828182845...

## III-2.3 描述比特流语法的方法

由解码器恢复的比特流在 III-2.4.1 和 III-2.5.1 节中描述。比特流中的每个数据项用黑体。通过名字、按位的长度以及其类型和传输顺序的助记符来描述。

比特流中解码的数据元素所导致的操作依赖于其值及之前解码的数据元素。数据元素的解码及所用的状态变量的定义在 III-2.4.2, III-2.4.3, III-2.5.2 和 III-2.5.3 节中描述。下面的结构表示数据元素以标准类型出现时的情形。

注意本语法用“C”-码规定: 变量或表达式为非零值时等价于条件为真。

```
while(condition){           若条件为真, 则数据元素组紧接着数据流产生。如此重复直
    data_element             到条件非真
```

```

...
}

do {
    data_element ...
}while(condition)

if (condition) {
    data_element
    ...
}
else {
    data_element
    ...
}

for (expr1;expr2;expr3)
{
    data_element
    ...
}

```

数据元素总是至少产生一次。  
数据元素重复直至条件非真

若条件为真,在数据流中产生第一组数据元素  
若条件非真,在数据流中产生第二组数据元素

**expr1** 是指定的循环初始表达式,一般规定计数器的初始状态  
**expr2** 是指定的每次循环前的测试条件。条件非真时循环终止  
**expr3** 是每次循环后的操作,一般是增加计数器

注意,本结构的最通常用法为:

```

for (i=0;i<n;i++) {
    data_element
    ...
}

```

数据元素组产生 n 次。数据元素组内的条件结构可能依赖循环控制变量 i 的值。i 第一次出现时被置为 0,第二次增加到 1,如此往复

注,数据元素组中可能含有嵌套的条件结构。为简便起见,当后面只有一个数据元素时{}省略。

data_element[]	data_element[]是一数组数据。
data_element[n]	数据元素的个数由上下文而定。
data_element[m][n]	data_element[n]是数组数据的第 n+1 个元素
data_element[1][m][n]	data_element[m][n]是二维数组的第 m+1,n+1 个元素
data_element[m..n]	data_element[1][m][n]是三维数组的第 1+1,m+1,n+1 个元素
	data_element[m..n]是位 m 到位 n 之间包括的位

虽然语法用过程项表示,但不能认为Ⅲ-2.4.3节实现了可靠的解码过程。它只是定义了一个正确的无错误的比特流输入。实际的解码器必须包含查寻起始码的方法才能开始正确解码。

#### bytealigned(字节对整)函数的定义

如果当前位置在字节的边界,bytealigned 函数返回 1,即比特流中的下一位是一个字节的起始位;否则返回 0。

#### nextbits 函数的定义

nextbits()将位串与比特流中将要解码的下一位进行比较。

#### next\_start\_code 函数定义

next\_start\_code 函数去掉填充的零位和零字节,并且定位下一个起始码。

语 法	位数	助记符
<pre>next_start_code(){     while(! bytealigned())         zero_bit     while (nextbits() !='0000 0000 0000 0000 0000 0001')         zero_byte }</pre>	'1' 8	'0' '00000000'

此函数检查当前位置是否是字节对准的。若不是,则存在填充的零位。此后到起始码前有可能有几个零字节。因此,起始码总是字节对准的,且有可能跟在零填充位之后。

## III-2.4 ISO/IEC 11172-3 降低采样频率音频编码扩展的要求

### III-2.4.1 编码音频比特流语法的说明

#### III-2.4.1.1 层 I , II

见 ISO/IEC 11172-3,子项 2.4.1

#### III-2.4.1.2 层 III

语 法	位 数	助记符
audio_data()		
{		
main_data_begin	8	uimsbf
if (mode==single_channel)		
private_bits	1	bslbf
else		
private_bits	2	bslbf
for (ch=0;ch<nch;ch++) {		
part2_3_length[ch]	12	uimsbf
big_values[ch]	9	uimsbf
global_gain[ch]	8	uimsbf

续表

语 法	位 数	助记符
scalefac_compress[ch]	9	bslbf
window_switching_flag[ch]	1	bslbf
if(window_switching_flag[ch]) {		
block_type[ch]	2	bslbf
mixed_block_flag[ch]	1	uimsbf
for(region=0;region<2;region++)		
table_select[ch][region]	5	bslbf
for(window=0;window<3;window++)		
subblock_gain[ch][window]	3	uimsbf
}		
else {		
for(region=0;region<3;region++)		
table_select[ch][region]	5	bslbf
region0_count[ch]	4	bslbf
region1_count[ch]	3	bslbf
}		
scalefac_scale[ch]	1	bslbf
count1table_select[ch]	1	bslbf
}		
main_data()	1	

主数据比特流定义如下。audio\_data()中的 main\_data 域包含主数据比特流中的字节。但是,由于层Ⅲ使用的哈夫曼编码的可变特性,本帧的主数据一般不跟在本帧的头和附带信息之后。本帧的 main\_data 起始位置在比特流中帧头之前由 main\_data\_begin 的值给定的负偏移值处。(见定义 main\_data\_begin)哈夫曼编码位见 ISO/IEC 11172-3 的 2.4.1.7 节。

语 法	位 数	助记符
main_data()		
{		
for (ch=0;ch<nch;ch++) {		
if((window_switching_flag[ch]==1)&&		
(block_type[ch]==2)){		
if (mixed_block_flag[ch]) {		
for (sfb=0;sfb<8;sfb++)		
scalefac_l[ch][sfb]	0..4	uimsbf
for (sfb=3;sfb<12;sfb++)		
for(window=0;window<3;window++)		
scalefac_s[ch][sfb][window]	0..5	uimsbf
}		
else {		

续表

语 法	位 数	助记符
<pre> for (sfb=0;sfb&lt;12;sfb++)     for (window=0;window&lt;3;window++)         scalefac_s[ch][sfb][window]     } } else{     for(sfb=0;sfb&lt;21;sfb++)         scalefac_l[ch][sfb] } Huffmancodebits() } for(b=0;b&lt;no_of_ancillary_bits;b++)     ancillary_bit } </pre>	0..5	uimsbf
	0..5	uimsbf
	1	bslbf

### III-2.4.2 音频比特流语法的语义

#### III-2.4.2.1 通用音频序列

见 ISO/IEC 11172-3,2.4.2.1 节

#### III-2.4.2.2 音频帧

见 ISO/IEC 11172-3,2.4.2.2 节

#### III-2.4.2.3 头

对所有层,前 32 位(4 字节)为头信息。

**同步字** 见 ISO/IEC 11172-3,2.4.2.3 节。

**ID** 用来标识算法的一个位。对 ISO/IEC 11172-3 降低采样频率扩展为“1”。

**层** 见 ISO/IEC 11172-3,2.4.2.3 节

**protection\_bit** 见 ISO/IEC 11172-3,2.4.2.3 节

**bitrate\_index** 指明比特率。全零标明自由格式情况,使用固定比特率,不必列出。

“固定”意味着一帧包含 N 或 N+1 个槽,取决于填充位的值。bitrate\_index 是表的索引,在不同层是不同的。bitrate\_index 指明了所有的比特率而不考虑方式(立体声,联合立体声,双声道,单声道)。当层 I , II 和 III 在自由格式时,不要求解码器支持高于 256kbit/s , 160kbit/s , 1600kbit/s 的比特率。

bitrate_index	当 $F_s = 16, 22.05, 24 \text{kHz}$ 时, 规定的比特率/kbits $^{-1}$	
	层 I	层 II, 层 III
'0000'	自由	自由
'0001'	32	8
'0010'	48	16
'0011'	56	24

续表

bitrate_index	当 $F_s = 16, 22.05, 24 \text{kHz}$ 时, 规定的比特率/kbits $^{-1}$	
	层 I	层 II, 层 III
'0100'	64	32
'0101'	80	40
'0110'	96	48
'0111'	112	56
'1000'	128	64
'1001'	144	80
'1010'	160	96
'1011'	176	112
'1100'	192	128
'1101'	224	144
'1110'	256	160
'1111'	禁止	禁止

**sampling\_frequency** 指明采样频率, 依据下表, 对于 ID='0', 音频解码器复位时也许要改变采样频率。

sampling_frequency	规定的频率/kHz
'00'	22.05
'01'	24
'10'	16
'11'	保留

**padding\_bit** 见 ISO/IEC 11172-3, 2.4.2.3 节。对于采样频率为 22.05kHz, 填充是必须的, 对自由格式有可能要求填充。

**private\_bit** 见 ISO/IEC 11172-3, 2.4.2.3 节

**mode** 见 ISO/IEC 11172-3, 2.4.2.3 节

**mode\_extension** 见 ISO/IEC 11172-3, 2.4.2.3 节

**copyright** 见 ISO/IEC 11172-3, 2.4.2.3 节

**original/copy** 见 ISO/IEC 11172-3, 2.4.2.3 节

**emphasis** 见 ISO/IEC 11172-3, 2.4.2.3 节

#### III - 2.4.2.4 错误校验

对于层 I 和层 II, 见 ISO/IEC 11172-3, 2.4.2.4 节

对于层 III, 用于计算错误校验的位是:

头的位 16..31  
单通道方式音频数据位 0..71  
其它方式音频数据位 0..135

#### III-2.4.2.5 音频数据层 I

见 ISO/IEC 11172-3,2.4.2.5 节

#### III-2.4.2.6 音频数据层 II

见 ISO/IEC 11172-3,2.4.2.6 节

#### III-2.4.2.7 音频数据层 III

除 scalefac\_compress 不同的定义之外,见 ISO/IEC 11172-3,2.4.2.7 节。**scalefac\_compress[ch]** 选择传输比例因子所用的位数,以及设置成复位预标志。若预标志已设置,则如 ISO/IEC 11172-3(表 B.6,附录 B)所述,把一个表的值加到比例因子上。

#### III-2.4.2.8 辅助数据

见 ISO/IEC 11172-3,2.4.1.8 节

### III-2.4.3 音频解码过程

#### III-2.4.3.1 音频解码层 I, II

见 ISO/IEC 11172-3,2.4.3 节。对于层 I, 在本推荐 | 国际标准中使用 ISO/IEC 11172-3 中的表 B.1(每一子波段的可能量化,层 I),而不用表 B.2(层 II 位分配表)。

#### III-2.4.3.2 音频解码层 III

层 III 低采样频率解码除以下几点不同外,与 ISO/IEC 11172-3 的层 II 一样:

- 在短块(block\_type == 2)的情况下,对强立体声的低边界的解码在每一窗口内单独进行。即,与 ISO/IEC 11172-3 2.4.3.4 节不同,强度界限的计算适用于每一短窗口的值,且允许每一短窗口的单独强立体声解码。
- 强立体声解码过程的第 4 步和第 5 步改动如下:

④  $R_i := L_i * k_r$

⑤  $L_i := L_i * k_l$

$k_l$  和  $k_r$  的值通过传输的比例因子 is\_pos\_sb 的值计算得到:

```
if(is_pos_sb == 0)  k_l = 1.0  k_r = 1.0
else if (is_pos_sb % 2 == 1)  k_l = i_0^(is_pos_sb+1)/2  k_r = 1.0
else      k_l = 1.0      k_r = i_0^is_pos_sb/2
```

基本强立体声解码因子  $i_0$  由 intensity\_scale 确定(当 intensity\_scale == 1 时为  $1/\sqrt{2}$ ,否则为  $1/\sqrt{\sqrt{2}}$ )。

- ISO/IEC 11172-3 2.4.3.4 节的“比例因子”段落由下段代替:

比例因子

比例因子是根据 slen1,slen2,slen3 和 slen4 和 nr\_of\_sfb1,nr\_of\_sfb2,nr\_of\_sfb3 和 nr\_of\_sfb4 解码的。它们是由 scalefac\_compress 的值确定的。

编码比例因子所用的位数称为 part2\_length,计算如下:

```

part2_length =nr_of_sfb1 *slen1 + nr_of_sfb2 *slen2 + nr_of_sfb3 *slen3
+ nr_of_sfb4 *slen4

```

比例因子分四部分传输。每部分的比例因子个数(nr\_of\_sfb1, nr\_of\_sfb2, nr\_of\_sfb3 和 nr\_of\_sfb4), 每部分的比例因子长度(slen1, slen2, slen3 和 slen4), 以及预标记, 均由下面的过程从 scalefac\_compress 解码得到:

```

if(! (((mode_extension==01) || (mode_extension==11))&&(ch==1))){
    if(scalefac_compress<400){
        slen1=(scalefac_compress>>4)/5
        slen2=(scalefac_compress>>4)%5
        slen3=(scalefac_compress%16)>>2
        slen4=scalefac_compress%4
        preflag=0
        block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
        0,1,3      x             6           5           5           5
        2          0             9           9           9           9
        2          1             8           9           9           9
    }
    if(400<=scalefac_compress<500){
        slen1=((scalefac_compress-400)>>2)/5
        slen2=((scalefac_compress-400)>>2)%5
        slen3=(scalefac_compress-400)%4
        slen4=0
        preflag=0
        block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
        0,1,3      x             6           5           7           3
        2          0             9           9           12          6
        2          1             8           9           12          6
    }
    if(500<=scalefac_compress<512){
        slen1=(scalefac_compress-500)/3
        slen2=(scalefac_compress-500)%3
        slen3=0
        slen4=0
        preflag=1
        block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
        0,1,3      x             11          10          0           0
        2          0             18          18          0           0
        2          1             17          18          0           0
    }
}
if((mode_extension==01) || (mode_extension==11))&&(ch==1){

```

```

intensity_scale=scalefac_compress%2
int_scalefac_compress=scalefac_compress>>1
if(int_scalefac_compress<180){
    slen1=int_scalefac_compress/36
    slen2=(int_scalefac_compress%36)/6
    slen3=(int_scalefac_compress%36)%6
    slen4=0
    preflag=0
    block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
    0,1,3   x                 7          7          7          0
    2       0                 12         12         12         0
    2       1                 8          15         12         0
}
if(180<=int_scalefac_compress<244){
    slen1=((int_scalefac_compress-180)%64)>>4
    slen2=((int_scalefac_compress-180)%16)>>2
    slen3=(int_scalefac_compress-180)%4
    slen4=0
    preflag=0
    block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
    0,1,3   x                 6          6          6          3
    2       0                 12         9          9          6
    2       1                 8          12         9          6
}
if(244<=int_scalefac_compress<255){
    slen1=(int_scalefac_compress-244)/3
    slen2=(int_scalefac_compress-244)%3
    slen3=0
    slen4=0
    preflag=0
    block_type mixed_block_flag nr_of_sfb1 nr_of_sfb2 nr_of_sfb3 nr_of_sfb4
    0,1,3   x                 8          8          5          0
    2       0                 15         12         9          0
    2       1                 8          18         9          0
}
}

```

## III - 2. 5 多通道低比特率音频编码的要求

### III - 2. 5. 1 编码音频比特流语法说明

#### III - 2. 5. 1. 1 音频序列

见 ISO/IEC 11172-3 2. 4. 1. 1 节。

#### III - 2. 5. 1. 2 层 I 音频帧

```
frame()
{
    mpeg1_header()
    mpeg1_error_check()
    mpeg1_audio_data()
    mc_extension_part1()
    mpeg1_header()
    mpeg1_error_check()
    mpeg1_audio_data()
    mc_extension_part2()
    mpeg1_header()
    mpeg1_error_check()
    mpeg1_audio_data()
    mc_extension_part3()
    mpeg1_ancillary_data()
}
```

#### III - 2. 5. 1. 3 层 II , III 音频帧

```
frame()
{
    mpeg1_header()
    mpeg1_error_check()
    mpeg1_audio_data()
    mc_extension()
    mpeg1_ancillary_data()
}
```

#### III - 2. 5. 1. 4 MC\_extension

```
mc_extension()
{
    mc_header()
    mc_error_check()
    mc_composite_status_info()
    mc_audio_data()
```

```

if(layer<3)
    ml_audio_data()
}

```

### III-2.5.1.5 MPEG1头

见 ISO/IEC 11172-3 2.4.1.3 节

### III-2.5.1.6 MPEG1错误校验

见 ISO/IEC 11172-3 2.4.1.4 节

### III-2.5.1.7 MPEG1音频数据

见 ISO/IEC 11172-3 2.4.1.5 节, 2.4.1.6 节, 2.4.1.7 节

### III-2.5.1.8 MC头

语 法	位数	助记符
mc_header()		
{		
center	2	bslbf
surround	2	bslbf
lfe	1	bslbf
audio_mix	1	bslbf
dematrix_procedure	2	bslbf
no_of_multi_lingual_ch	3	uimsbf
multi_lingual_fs	1	bslbf
multi_lingual_layer	1	bslbf
ext_bit_stream_present	1	bslbf
if ext_bit_stream_present==‘1’		
n_ad_bytes	8	uimsbf
}		

### III-2.5.1.9 MC 错误校验

语 法	位数	助记符
mc_error_check()		
{		
mc_crc_check	16	rpchof
}		

### III-2.5.1.10 MC 合成状态信息, 层 I , II

语 法	位数	助记符
mc_composite_status_info()		
{		
tc_sbgr_select	1	bslbf

续表

语 法	位数	助记符
<b>dyn_cross_on</b>	1	bslbf
<b>mc_prediction_on</b>	1	bslbf
if tc_sbgr_select == 1		
{		
<b>tc_allocation</b>	2..3	uimsbf
for (sbgr=0; sbgr<12; sbgr++)		
tc_allocation[sbgr] = tc_allocation		
}		
else for (sbgr=0; sbgr<12; sbgr++)		
<b>tc_allocation[sbgr]</b>	2..3	uimsbf
if <b>dyn_cross_on</b> == 1		
{		
<b>dyn_cross_LR</b>	1	bslbf
for (sbgr=0; sbgr<12; sbgr++)		
<b>dyn_cross_mode[sbgr]</b>	1..5	bslbf
}		
if <b>mc_prediction_on</b> == 1		
{		
for (sbgr=0; sbgr<8; sbgr++)		
{		
<b>mc_prediction[sbgr]</b>	1	bslbf
if (mc_prediction[sbgr] == 1)		
for (pci=0; pci<npredcoef; pci++)		
predsi[sbgr, pci]	2	bslbf
}		
}		
}		

### III-2.5.1.11 MC 合成状态信息, 层Ⅲ

语 法	位数	助记符
<b>mc_composite_status_info()</b>		
{		
<b>mc_data_begin</b>	11	uimsbf
for (gr=0; gr<2; gr++)		
for (ch=2; ch<4; ch++) {		
<b>seg_list_present[gr][ch]</b>	1	bslbf
<b>tc_present[gr][ch]</b>	1	bslbf
<b>block_type[gr][ch]</b>	2	bslbf
}		
if (center != '00') {		
for (gr=0; gr<2; gr++) {		

续表

语 法	位数	助记符
<b>seg_list_present[gr][center Chan]</b>	1	bslbf
<b>tc_present[gr][center Chan]</b>	1	bslbf
<b>block_type[gr][center Chan]</b>	2	bslbf
}		
}		
if(surround=='01') {		
for(gr=0;gr<2;gr++) {		
<b>seg_list_present[gr][mono_surr Chan]</b>	1	bslbf
<b>tc_present[gr][mono_surr Chan]</b>	1	bslbf
<b>block_type[gr][mono_surr Chan]</b>	2	bslbf
}		
}		
if(surround=='10'    surround=='11') {		
for(gr=0;gr<2;gr++)		
for(ch=left_surr Chan;ch<=right_surr Chan;ch++) {		
<b>seg_list_present[gr][ch]</b>	1	bslbf
<b>tc_present[gr][ch]</b>	1	bslbf
<b>block_type[gr][ch]</b>	2	bslbf
}		
}		
if (dematrix_procedure != 3)		
<b>dematrix_length</b>	4	bslbf
else		
dematrix_length=0		
for(sbgr=0;sbgr<dematrix_length;sbgr++)		
<b>dematrix_select[sbgr]</b>	3..4	bslbf
for (gr=0;gr<2;gr++)		
for(ch=2;ch<7;ch++) {		
if(ch_present(ch) && seg_list_present[gr][ch]) {		
<b>seg_list_nodef[gr][ch]</b>	1	bslbf
if(seg_list_nodef[gr][ch]) {		
if(gr==1 && seg_list_present[gr_0][ch] &&		
seg_list_nodef[gr_0][ch]) {		
<b>segment_list_repeat[ch]</b>	1	bslbf
if(! segment_list_repeat[ch]) {		
<b>segment_list(gr, ch)</b>		
}		
}		
else		
<b>segment_list(gr, ch)</b>		
}		
}		

续表

语 法	位数	助记符
<b>mc_prediction_on</b>	1	bslbf
if(mc_prediction_on){		
for (sbgr=0;sbgr<15;sbgr++)		
<b>mc_prediction[sbgr]</b>	1	bslbf
for (sbgr=0;sbgr<15;sbgr++){		
if(mc_prediction_sbgr[sbgr]){		
for (pci=0;pci<npredcoef;pci++)		
<b>predsi[sbgr][pci]</b>	1	bslbf
}		
}		
for (sbgr=0;sbgr<15;sbgr++){		
for (pci=0;pci<npredcoef;pci++)		
if(predsi[sbgr][pci])		
<b>pred_coef[sbgr][pci]</b>	3	uimsbf
}		
}		
}		

段列表语法定义如下：

语 法	位数	助记符
segment_list(gr,ch)		
{		
seg=0		
sbgr=dematrix_length		
if(block_type[gr][ch]==2) sbgr_cnt=12 else sbgr_cnt=15		
<b>attenuation_range[gr][ch]</b>	2	uimsbf
<b>attenuation_scale[gr][ch]</b>	1	uimsbf
while(sbgr<sbgr_cnt){		
<b>seg_length[gr][ch][seg]</b>	4	uimsbf
if(seg_length[gr][ch][seg]==0)		
break;		
<b>tc_select[gr][ch][seg]</b>	3	uimsbf
if(tc_select[gr][ch][seg]! =7 && tc_select[gr][ch][seg]! =ch)		
for(sbgr1=sbgr;sbgr1<sbgr+seg_length[gr][ch][seg];sbgr1--)		
<b>attenuation[gr][ch][seg][sbgr1]</b>	2..5	uimsbf
sbgr+=seg_length[gr][ch][seg]		
seg++		
}		
}		

### III -2. 5. 1. 12 MC 音频数据, 层 I 和层 II

语 法	位数	助记符
mc_audio_data()		
<pre> if(lfe==1)     <b>lfe_allocation</b>     for(sb=0;sb&lt;msblimit;sb++)         for(mch=0;mch&lt;nmch;mch++)             if(! center_limited[mch,sb] &amp;&amp; ! dyn_cross[mch,sb])                 allocation[mch,sb]             else allocation[mch,sb]=0     for(sb=0;sb&lt;msblimit;sb++)         for (mch=0;mch&lt;nmch;mch++)             if(allocation[mch,sb]!=0    dyn_cross[mch,sb])                 scfsi[mch,sb] </pre>	4	uimsbf
<pre> if mc_prediction_on==1     for(sbgr=0;sbgr&lt;8;sbgr++)         if(mc_prediction[sbgr]==1)             for(px=0;px&lt;npred;px++)                 if(predsi[sbgr,px]!=0)                     delay_comp[sbgr,px]                     for(pci=0;pci&lt;predsi[sbgr,px];pci++)                         pred_coef[sbgr,px,pci] </pre>	2..4	uimsbf
<pre> if(lfe==1)     <b>lfe_scalefactor</b>     for(sb=0;sb&lt;msblimit;sb++)         for(mch=0;mch&lt;nmch;mch++)             if(allocation[mch,sb]!=0    dyn_cross[mch,sb])                 {                     if(scfsi[mch,sb]==0)                         { scalefactor[mch,sb,0]                             scalefactor[mch,sb,1]                             scalefactor[mch,sb,2] }                     if(scfsi[mch,sb]==1    scfsi[mch,sb]==3)                         { scalefactor[mch,sb,0]                             scalefactor[mch,sb,2] }                     if(scfsi[mch,sb]==2)                         scalefactor[mch,sb,0]                 }     for(gr=0;gr&lt;12;gr++)     {         if(lfe=='1')             <b>if_sample[gr]</b> </pre>	2	uimsbf
	3	uimsbf
	8	uimsbf
	6	uimsbf
	2..15	uimsbf

续表

语 法	位数	助记符
<pre> for(sb=0;sb&lt;msblimit;sb++)     for(mch=0;mch&lt;nmch;mch++)         if(allocation[mch,sb]! = 0)         {             if(grouping[mch,sb])                 samplecode[mch,sb,gr]             else for (s=0;s&lt;3;s++)                 sample[mch,sb,3 * gr+s]         }     } } </pre>	5..10 2..16	uimsbf uimsbf

## III -2. 5. 1. 13 MC 音频数据, 层Ⅲ

语 法	位数	助记符
<pre> mc_audio_data() {     if(dematrix_procedure! = 3)         matrix_attenuation_present     else         matrix_attenuation_present=0     if(matrix_attenuation_present){         for(gr=0;gr&lt;2;gr++)             for(ch=2;ch&lt;7;ch++)                 if(block_type[gr][ch]==2)                     for(sbgr=dematrix_length;sbgr&lt;12;sbgr++)                         if(js_carrier[gr][ch][sbgr]){                             matrix_attenuation_l[gr][ch][sbgr]                             matrix_attenuation_r[gr][ch][sbgr]                         }                     else                         for(sbgr=dematrix_length;sbgr&lt;15;sbgr++)                             if(js_carrier[gr][ch][sbgr]){                                 matrix_attenuation_l[gr][ch][sbgr]                                 matrix_attenuation_r[gr][ch][sbgr]                             }     }      for(tc=2;tc&lt;7;tc++)         for(scfsi_band=0;scfsi_band&lt;4;scfsi_band++)             if(tc_present[gr_0][tc] &amp;&amp; tc_present[gr_1][tc])                 scfsi[tc][scfsi_band]             else </pre>	1 3 3 3 3 3 1	bslbf bslbf bslbf bslbf bslbf bslbf bslbf

续表

语 法	位数	助记符
scfsi[tc][scfsi_band] = 0		
for(gr=0;gr<2;gr++) {		
for(tc=2;tc<7;tc++) {		
if(tc_present[gr][tc]) {		
part2_3_length[gr][tc]	12	uimsbf
big_values[gr][tc]	9	uimsbf
global_gain[gr][tc]	8	uimsbf
scalefac_compress[gr][tc]	4	bslbf
if(block_type[gr][tc] != 0) {		
for(region=0;region<2;region++)		
table_select[gr][tc][region]	5	bslbf
if(block_type[gr][tc] == 2) {		
for(window=0>window<3>window++)		
subblock_gain[gr][tc][window]	3	uimsbf
}		
}		
else {		
for(region=0;region<3;region++)		
table_select[gr][tc][region]	5	bslbf
region0_count[gr][tc]	4	bslbf
region1_count[gr][tc]	3	bslbf
}		
preflag[gr][tc]	1	bslbf
scalefac_scale[gr][tc]	1	bslbf
count1table_select[gr][tc]	1	bslbf
}		
}		
}		
if(lfe)		
LFE_header()		
if(no_of_multi_lingual_ch != 0)		
ML_header()		
while(!bytealigned())		
byte_align_bit		
mc_main_data()		
if(lfe)		
lfe_main_data()		
if(no_of_multi_lingual_ch != 0)		
ml_main_data()		
}		

主数据比特流定义如下:mc\_audio\_data()中的 mc\_main\_data 域包含从主数据比特流来的字节。但是,由于层Ⅲ所使用的哈夫曼编码的变长特性,一帧的主数据并非跟在此帧的头和附带信息之后。一帧的 mc\_main\_data 的起始位置在由 mc\_data\_begin 给出的

负偏移确定的帧头的前面。(见 ISO/IEC 11172-3 中 main\_data\_begin 的定义。)

语 法	位数	助记符
<pre> mc_main_data() {     for(gr=0;gr&lt;2;gr++){         for(tc=2;tc&lt;7;tc++){             if(tc_present[gr][tc]){                 if(block_type[gr][tc]==2){                     for(sfb=0;sfb&lt;12;sfb++)                         for(window=0;window&lt;3;window++)                             if(data_present[gr][tc][sfb][window]){                                 scalefac_s[gr][tc][sfb][window]                             }                 }             else{                 if((scfsi[tc][0]==0)    (gr==0))                     for(sfb=0;sfb&lt;6;sfb++)                 if(data_present[gr][tc][sfb]){                     scalefac_l[gr][tc][sfb]                 }                 if((scfsi[tc][1]==0)    (gr==0))                     for(sfb=6;sfb&lt;11;sfb++)                 if(data_present[gr][tc][sfb]){                     scalefac_l[gr][tc][sfb]                 }                 if((scfsi[tc][2]==0)    (gr==0))                     for(sfb=11;sfb&lt;16;sfb++)                 if(data_present[gr][tc][sfb]){                     scalefac_l[gr][tc][sfb]                 }                 if((scfsi[tc][3]==0)    (gr==0))                     for(sfb=16;sfb&lt;21;sfb++)                 if(data_present[gr][tc][sfb]){                     scalefac_l[gr][tc][sfb]                 }             }         }     } } Huffmancodebits() } } } </pre>	0..4	uimsbf

续表

语 法	位数	助记符
<pre> <b>Huffmancodebits()</b>   for(1=0;1&lt;big_values * 2;1+=2){     <b>hcod[ x ][ y ]</b>     if( x =15 &amp;&amp; linbits&gt;0)       <b>linbitsx</b>     if(x!=0)       <b>signx</b>     if( y ==15 &amp;&amp; linbits&gt;0)       <b>linbitsy</b>     if(y!=0)       <b>signy</b>     is[1]=x     is[1+1]=y   }   for(;l&lt;big_values * 2+count1 * 4;1+=4){     <b>hcod[ v ][ w ][ x ][ y ]</b>     if(v!=0)       <b>signv</b>     if(w!=0)       <b>signw</b>     if(x!=0)       <b>signx</b>     if(y!=0)       <b>signy</b>     is[1]=v     is[1+1]=w     is[1+2]=x     is[1+3]=y   }   for(;l&lt;576;l++)     is[l]=0 } </pre>		

语 法	位数	助记符
<b>LFE_header()</b>		
<b>lfe_hc_len</b>	8	uimsbf
<b>lfe_gain</b>	8	uimsbf
<b>lfe_table_select</b>	5	uimsbf

续表

语 法	位数	助记符
<pre> lfe_main_data() {     for(l=0;l&lt;lfe_bigval;l++)         hcod[ x ][ y ]         if( x ==15 &amp;&amp; linbits&gt;0)             linbitsx         if(x!=0)             signx         if( y ==15 &amp;&amp; linbits&gt;0)             linbitsy         if(y!=0)             signy         is_lfe[gr_0][1]=x         is_lfe[gr_1][1]=y     }     while(l&lt;6){         is_lfe[gr_0][1]=0;         is_lfe[gr_1][1]=0;         l++;     } } </pre>	0...19 1...13 1 1	blsbf uimsbf blsbf blsbf

### III-2.5.1.14 ML 音频数据, 层 I 和层 II

语 法	位数	助记符
<pre> ml_audio_data() {     for(sb=0;sb&lt;csblimit;sb++)         for(mlch=0;mlch&lt;nmlch;mlch++)             allocation[mlch,sb]     for(sb=0;sb&lt;mlsblimit;sb++)         for(mlch=0;mlch&lt;nmlch;mlch++)             if(allocation[mlch,sb]!=0)                 scfsi[mlch,sb]     for(sb=0;sb&lt;mlsblimit;sb++)         for(mlch=0;mlch&lt;nmlch;mlch++)             if(allocation[mlch,sb]!=0)             {                 if(scfsi[mlch,sb]==0)                     { scalefactor[mlch,sb,0]                       scalefactor[mlch,sb,1]                       scalefactor[mlch,sb,2] }                 if(scfsi[mlch,sb]==1    scfsi[mlch,sb]==3) </pre>	2..4 2 6 6 6	uimsbf uimsbf uimsbf uimsbf

续表

语 法	位数	助记符
{ scalefactor[mlch, sb, 0] scalefactor[mlch, sb, 2]} if(scfsi[mlch, sb] == 2) scalefactor[mlch, sb, 0] } for(gr=0;gr<ngr;gr++) for(sb=0;sb<mlsblimit;sb++) for(mlch=0;mlch<nmlch;mlch++) if(allocation[mlch, sb] != 0) { if(grouping[mlch, sb]) samplecode[mlch, sb, gr] else for(s=0;s<3;s++) sample[mlch, sb, 3 * gr + s] } }	6 6 6 5..10 2..16	uimsbf uimsbf uimsbf uimsbf uimsbf

### III -2. 5. 1. 15 ML 头, 层 III

语 法	位数	助记符
ML_header() { if(multi_lingual_fs == 0) ngr = 2 else ngr = 1 for(gr=0;gr<ngr;gr++){ for(ch=0;ch<nch;ch++){ part2_3_length[gr][ch] big_values[gr][ch] global_gain[gr][ch] scalefac_compress[gr][ch] window_switching_flag[gr][ch] if(window_switching_flag[gr][ch]){ block_type[gr][ch] mixed_block_flag[gr][ch] for(region=0;region<2;region++) table_select[gr][ch][region] for(window=0;window<3;window++) subblock_gain[gr][ch][window] } else{ for(region=0;region<3;region++) table_select[gr][ch][region] region0_count[gr][ch] region1_count[gr][ch]	12 9 8 4 1 2 1 5 3 5 4 3	uimsbf uimsbf uimsbf bslbf bslbf bslbf uimsbf uimsbf bslbf uimsbf bslbf bslbf bslbf

续表

语 法	位数	助记符
}		
preflag[gr][ch]	1	bslbf
scalefac_scale[gr][ch]	1	bslbf
count1table_select[gr][ch]	1	bslbf
}		
}		
}		

### III-2.5.1.16 ML 主数据,层Ⅲ

若 multilingual\_fs==0,见 ISO/IEC 11172-3 的 2.4.1.7 节。

若 multilingual\_fs==1,见 2.4.1.2 节。

使用 ML 主数据,nch 设置成 no\_of\_multi\_lingual\_ch。

### III-2.5.1.17 MPEG1 辅助数据

若 ext\_bit\_stream\_present==1,下面语法有效。

语 法	位数	助记符
MPEG1_ancillary_data() { if ext_bit_stream_present==1 { for(b=0;b<8*n_ad_bytes;b++) ancillary_bit } }	1	bslbf

若 ext\_bit\_stream\_present==0,见 ISO/IEC 11172-3 的 2.4.1.8 节

### III-2.5.1.18 Ext-frame

```
if ext_bit_stream_present=='1'  
  ext_frame()  
  {  
    ext_header()  
    ext_data()  
    ext_ancillary_data()  
  }
```

### III-2.5.1.19 Ext\_header

语 法	位数	助记符
ext_header()		
{		
ext_syncword	12	bslbf
ext_crc_check	16	bslbf

续表

语 法	位数	助记符
<b>ext_length</b>	11	uimsbf
<b>reserved_bit</b>	1	bslbf
}		

### III-2.5.1.20 Ext\_ancillary\_data

语 法	位数	助记符
<b>ext_ancillary_data()</b> { <b>for(b=0;b&lt;no_of_ancillary_bits;b++)</b> <b>ext_ancillary_bit</b> } }	1	bslbf

## III-2.5.2 音频比特流语法的语义

### III-2.5.2.1 通用音频序列

**frame** 一部分比特流,可依靠自身解码。它包含每一编码音频通道的 1152 个音频样本,LFE 通道的 12 个样本,每个多语种通道的 1152 或 576 个样本。它以一个同步字开始,在层 I 的第三个同步字之前和层 II 或层 III 的下一个同步字之前结束,它由整数个槽组成。(层 I 中为 4 个字节,层 II 或层 III 中为 1 个字节。)

### III-2.5.2.2 层 I 音频帧

**mpeg1\_header** 比特流中包含同步和状态信息的部分。

**mpeg1\_error\_check** 比特流中的一部分,包含有比特流中 MPEG1 部分的错误检测信息。

**mpeg1\_audio\_data** 比特流中的一部分,含有比特流中 MPEG1 部分的音频样本信息。

**mc\_extension\_part1, mc\_extension\_part2, mc\_extension\_part3** 这三部分构成了完整的一帧的多通道扩展域'mc\_extension',包含 mc\_header, mc\_error\_check, mc\_composite\_status\_info, mc\_audio\_data 和 ml\_audio\_data。mc\_extension\_part1 和 mc\_extension\_part2 的结尾各包含一位零。

**mpeg1\_ancillary\_data** 比特流中用于辅助数据的部分。

### III-2.5.2.3 层 II, III 音频帧

**mpeg1\_header** 见 III-2.5.2.2 节

**mpeg1\_error\_check** 见 III-2.5.2.2 节

**mpeg1\_audio\_data** 见 III-2.5.2.2 节

**mc\_extension** 这部分构成了多通道扩展域,包括 mc\_header, mc\_error\_check, mc\_composite\_status\_info, mc\_audio\_data 和 ml\_audio\_data。

**mpeg1\_ancillary\_data** 见 III-2.5.2.2 节

### **III-2.5.2.4 MC 扩展**

**mc\_header** 比特流中包含关于比特流的多通道和多语种扩展的信息的部分。

**mc\_error\_check** 比特流中包含关于比特流多通道扩展错误校验信息的部分。

**mc\_composite\_status\_info** 比特流中包含合成编码方式状态信息的部分。

**mc\_audio\_data** 比特流中包含关于比特流多通道扩展音频样本信息的部分。

**ml\_audio\_data** 比特流中包含关于比特流解说词扩展音频样本信息的部分。

### **III-2.5.2.5 MPEG1 头**

见 ISO/IEC 11172-3 的 2.4.2.3 节

### **III-2.5.2.6 MPEG1 错误校验**

见 ISO/IEC 11172-3 的 2.4.2.4 节

### **III-2.5.2.7 MPEG1 音频数据**

见 ISO/IEC 11172-3 的 2.4.2.5 节, 2.4.2.6 节, 2.4.2.7 节。

### **III-2.5.2.8 MC 头**

**center** 两位, 用以指示在多路通道中是否有中央通道及其带宽。

'00' 无中央通道

'01' 有中央通道

'10' 无定义

'11' 中央限制带宽的(幻象编码)

若中央通道信号是限制带宽的, 子波段 11 以上的子波段根本不传输。解码器必须将变量 **center\_limited[mch, sb]** 设成 true(真), 对那些子波段的分配设为 0:

```
if(center == '11')
    for(sb = 12; sb < msblimit; sb++)
        center_limited[center, sb] = true;
```

**surround** 两位, 用以指示多路通道中是否有环绕通道。

'00' 无环绕

'01' 单环绕

'10' 立体声环绕

'11' 无环绕, 但有第二立体声节目

**lfe** 一位, 指示是否有低频效果通道。

'0' 无低频效果通道

'1' 有低频效果通道

**audio\_mix** 一位, 指示信号是否为大听音空间混合, 像剧院, 或为小听音空间混合, 像居室。解码器不必关心此位, 但重放系统要用到此位。

'0' 音频节目为大听音空间混合

'1' 音频节目为小听音空间混合

**dematrix\_procedure** 两位, 指示在解码器中用的是什么反矩阵过程。dematrix\_procedure 影响 tc\_allocation 解码和反规范过程。

'00' 过程 0

'01' 过程 1  
 '10' 过程 2  
 '11' 过程 3

值'10'只能与 3/1 或 3/2 配置组合应用。

**no\_of\_multi\_lingual\_ch** 一个无符号整型,其中的三位指示 mc 扩展比特流中的多语种或解说词通道。

**multi\_lingual\_fs** 一位,指示多语种通道的采样频率是否与主音频通道相同。若采样频率选择为  $1/2 * Fs$ (主音频通道)则为 1,若两者采样频率相同则为‘0’。

**multi\_lingual\_layer** 一位,指示层 I\_ml 还是层 II\_ml 被使用。层 I\_ml 总是与层 I 一起使用的。

ISO 11172-3 基本立体声	multi_lingual_layer	层
层 I	X	层 I_ml
层 I	0	层 I_ml
层 I	1	层 I_ml
层 II	0	层 II_ml
层 II	1	层 II_ml

**ext\_bit\_stream\_present** 一位,指示是否存在比特流扩展,当完整的多通道和多语种信息不适合某个 MPEG1 兼容比特流时,比特流扩展中包含多通道和多语种信息的剩余部分。

**n\_ad\_bytes** 构成一个无符号整数的 8 位,指明了 MPEG1 兼容的辅助数据域使用了多少字节。

### III-2.5.2.9 MC 错误校验

**mc\_crc\_check** 针对错误检测的 16 位强制校验字。也用于检测是否有可用的多通道或多语种信息。在层 I,多通道 crc 校验的计算从多通道首部的第一位开始,到比例因子域的最后一一位止。在层 II,从多通道首部的第一位开始计算,到 scfsi 域的最后一一位止。在层 III,从多通道首部第一位开始,到 ML\_header(包含)的最后一一位止。

### III-2.5.2.10 MC 组成状态信息层 I, II

**tc\_sbgr\_select** 一位,表示 tc\_allocation 对所有子波段还是单独子波段组有效。若对所有子波段有效,则为“1”;若对单独子波段组有效则为‘0’。下表所示为根据子波段组 sbgr 的子波段的分配。

sbgr	子波段组中包含的子波段
0	0
1	1
2	2

续表

sbgr	子波段组中包含的子波段
3	3
4	4
5	5
6	6
7	7
8	8...9
9	10...11
10	12...15
11	16...31

**dyn\_cross\_on** 一位,指示是否使用动态交叉会话。若用则为‘1’,否则为‘0’。

**mc\_prediction\_on** 一位,指示是否用到 mc\_prediction。用则为‘1’,否则为‘0’。

**tc\_allocation,tc\_allocation[sbgr]** 包含传输通道分配的信息。分别对应所有子波段或子波段组 sbgr 中的子波段。T0 总包含 Lo 而 T1 总包含 Ro。dematrix\_procedure 为‘11’表示 tc\_allocation[sbgr]==0;若 dematrix\_procedure==‘10’,tc\_allocation 只有为 0,1,2 有效。若用到幻象编码,中央通道须包含在用到的子波段组的附加传输通道中,如:对那些子波段组 tc\_allocation 的值限制为

- 0,3,4,5 在 3/2 方式
- 0,3,4 在 3/1 方式
- 0 在 3/0 和 3/0+2/0 方式

#### (1) 3/2 配置(nmch==3)

tc_allocation	T2	T3	T4
0	C <sup>w</sup>	LS <sup>w</sup>	RS <sup>w</sup>
1	L <sup>w</sup>	LS <sup>w</sup>	RS <sup>w</sup>
2	R <sup>w</sup>	LS <sup>w</sup>	RS <sup>w</sup>
3	C <sup>w</sup>	L <sup>w</sup>	RS <sup>w</sup>
4	C <sup>w</sup>	LS <sup>w</sup>	R <sup>w</sup>
5	C <sup>w</sup>	L <sup>w</sup>	R <sup>w</sup>
6	R <sup>w</sup>	L <sup>w</sup>	RS <sup>w</sup>
7	L <sup>w</sup>	LS <sup>w</sup>	R <sup>w</sup>

(2) 3/1 配置(nmch==2)

tc_allocation	T2	T3
0	C <sup>w</sup>	S <sup>w</sup>
1	L <sup>w</sup>	S <sup>w</sup>
2	R <sup>w</sup>	S <sup>w</sup>
3	C <sup>w</sup>	L <sup>w</sup>
4	C <sup>w</sup>	R <sup>w</sup>

(3) 3/0(+2/0)配置(nmch==1 在 3/0 方式,nmch==3 在 3/0+2/0 方式)

tc_allocation	T2
0	C <sup>w</sup>
1	L <sup>w</sup>
2	R <sup>w</sup>

在第二立体声节目的情况下,T3 包含第二立体声节目的 L2,T4 包含 R2。

(4) 2/2 配置(nmch==2)

tc_allocation	T2	T3
0	LS <sup>w</sup>	RS <sup>w</sup>
1	L <sup>w</sup>	RS <sup>w</sup>
2	LS <sup>w</sup>	R <sup>w</sup>
3	L <sup>w</sup>	R <sup>w</sup>

(5) 2/1 配置(nmch==1)

tc_allocation	T2
0	S <sup>w</sup>
1	L <sup>w</sup>
2	R <sup>w</sup>

(6) 2/0(+2/0)配置(nmch==0 在 2/0 方式,nmch==2 在 2/0+2/0 方式)在第二立体声节目时,T2 包含 L2,T3 包含 R2。

(7) 1/0(+2/0)配置(在 1/0 方式 nmch==0,在 1/0+2/0 方式,nmch==2)在第二立体声节目里,T1 包含 L2,T2 包含 R2。

**dyn\_cross\_LR** 一位,指示 C<sup>w</sup> 是从 Lo(若 dyn\_cross\_LR=='0')还是从 Ro(若 dyn\_

`cross_LR == '1')`拷贝。

**`dyn_cross_mode[sbgr]`** 一至四位,指示对于子波段组 `sbgr` 的子波段,哪一传输通道的动态交叉会话是活动的,对于那些子波段,位分配和子波段样本不在比特流中。但是比例因子和比例因子选择信息总是有的。这些子波段的分配被设置为零。本域的位数取决于通道的配置。通道可以是 3/2(1),3/1(2),3/0(3),2/2(4) 和 2/1(5)。下表给出了所有不出现的传输通道的模式。若传输通道  $T_i$  不出现(在表中以‘—’表示),对应于此音频通道的已重量化但未反比例变换的子波段样本将根据以下原则拷贝。

- 在表中同一行若有一项  $T_{ij}, j$  通道中的子波段样本从  $i$  通道复制。
- 否则  $L^w$  和  $LS^w$  从  $Lo$  复制。  
 $R^w$  和  $RS^w$  从  $Ro$  复制  
若  $dyn\_cross\_LR == 0$  则  $C^w$  从  $Lo$  复制,若  $dyn\_cross\_LR == 1$  则  $C^w$  从  $Ro$  复制。

初始情况下,对于所有通道的所有子波段,变量 `dyn_cross[Tx,sb]` 被设置成 `false`。然后,对于位分配和样本未被传输的子波段,变量 `dyn_cross[mch_sb]` 被置为 `true`:

```
for(sb=lim1;sb<=lim2;sb++)
    dyn_cross[Tx,sb]=true;
```

其中 `lim1` 和 `lim2` 代表子波段组的边界。

(1) 3/2 配置,域长 4 位:

<code>dyn_cross_mod</code> <code>e[sbgr]</code>	传输通道		
0	$T_2$	$T_3$	$T_4$
1	$T_2$	$T_3$	—
2	$T_2$	—	$T_4$
3	—	$T_3$	$T_4$
4	$T_2$	—	—
5	—	$T_3$	—
6	—	—	$T_4$
7	—	—	—
8	$T_2$	$T_{34}$	—
9	$T_{23}$	—	$T_4$
10	$T_{24}$	$T_3$	—
11	$T_{23}$	—	—
12	$T_{24}$	—	—
13	—	$T_{34}$	—
14	$T_{234}$	—	—
15	禁止		

{无动态交叉会话}

(2) 3/1 配置,域长 3 位:

dyn_cross_mod e[sbgr]	传输通道	
0	T2	T3
1	T2	—
2	—	T3
3	—	—
4	T23	—
5	禁止	
6	禁止	
7	禁止	

{无动态交叉会话}

(3) 3/0 配置,域长 1 位:

dyn_cross_mod e[sbgr]	传输通道	
0	T2	—
1	—	—

{无动态交叉会话}

(4) 2/2 配置,域长 3 位:

dyn_cross_mod e[sbgr]	传输通道	
0	T2	T3
1	T2	—
2	—	T3
3	—	—
4	T23	—
5	禁止	
6	禁止	
7	禁止	

{无动态交叉会话}

(5) 2/1 配置,域长 1 位:

<b>dyn_cross_mod</b>	传输通道
e[sbgr]	
0	T2
1	—

{无动态交叉会话}

若有第二立体声节目,dyn\_cross\_mode[sbgr]域的长度要加1。附加位的意义是:

'0' 第二立体声节目无动态交叉会话

'1' R2 的子波段样本(2/0+2/0 配置时 T3 通道,3/0+2/0 配置时 T4 通道)从 L2  
(2/0+2/0 配置时 T2 通道,3/0+2/0 配置时 T3 通道)预测得到。

**mc\_prediction[sbgr]** 一位,指示在子波段组 sbgr 中是否使用了预测降低冗余。mc\_prediction 的使用仅限于子波段组 0 到 7。若使用冗余降低则为'1',否则为'0'。

**predsi[sbgr,px]** 预测器选择信息。指示子波段组 sbgr 中由 px 索引的预测器是否被使用。若使用了,有多少系数被传递。

00	未使用预测器
01	传递 1 个系数
10	传递 2 个系数
11	传递 3 个系数

使用的预测器的最大数目 npred 取决于传输通道分配(tc\_allocation)和动态交叉会话(dyn\_cross\_mode)。npred 的值如下:

### 3/2 配置

tc_allocation	动态交叉会话															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	3	3	2	2	1	1	0	4	3	3	2	2	2	—	—
1	3	2	2	2	1	1	1	0	3	3	3	2	2	2	—	—
2	3	2	2	2	1	1	1	0	3	3	3	2	2	2	—	—
3	4	3	3	2	2	1	1	0	4	3	3	2	2	2	—	—
4	4	3	3	2	2	1	1	0	4	3	3	2	2	2	—	—
5	4	3	3	2	2	1	1	0	4	3	3	2	2	2	—	—
6	3	2	2	2	1	1	1	0	3	3	3	2	2	2	—	—
7	3	2	2	2	1	1	1	0	3	3	3	2	2	2	—	—

### 3/1 配置

tc_allocation	动态交叉会话							
	0	1	2	3	4	5	6	7
0	4	2	2	0	2	—	—	—
1	3	1	2	0	2	—	—	—
2	3	1	2	0	2	—	—	—
3	3	2	1	0	2	—	—	—
4	3	2	1	0	2	—	—	—

### 3/0 配置

tc_allocation	动态交叉会话	
	0	1
0	2	0
1	1	0
2	1	0

### 2/2 配置

tc_allocation	动态交叉会话							
	0	1	2	3	4	5	6	7
0	2	1	1	0	2	—	—	—
1	2	1	1	0	2	—	—	—
2	2	1	1	0	2	—	—	—
3	2	1	1	0	2	—	—	—

### 2/1 配置

tc_allocation	动态交叉会话	
	0	1
0	2	0
1	1	0
2	1	0

### III-2.5.2.11 MC 合成状态信息层Ⅲ

**mc\_data\_begin** 11 位,指示相对于实际帧,第一个字节的负偏移,以字节表示。帧内属于 MPEG-1 部分的字节的数目,mc\_header,mc\_error\_check 和 mc\_composite\_status\_info 未计在内。这意味着,若 `mc_data_begin == 0`,则 `mc_main_data` 从最后一个 `byte_aligned_bit` 后开始。

**seg\_list\_present[gr][ch]** 只有通道被标记为在 `mc_header()` 中存在时才传输。若 `seg_list_present` 未标记(只对最多两个通道合法),相应的通道通过对兼容的左/右传输通道进行反矩阵过程而得到重构。

**seg\_list\_nodef[gr][ch]** 指示 `segmentlist` 是否被传输,或是否使用了缺省段表。缺省的段表指出该通道在相应的通道内完全传输。

**segment\_list\_repeat[ch]** 指出了第二个颗粒的段表是否与第一个相同。只有在第一个颗粒的段表已被传输且不为缺省类型时,才传输此变量。

**tc\_present[gr][ch]** 指明是否可在比特流中包含有关 `tc`(传输通道)的信息。`seg_list_present` 与 `tc_present` 的区别在于传输通道数可以少于输出通道数,即使通道是由反矩阵重构的。一个通道含有一个段表,但是,相应的 `tc` 不必通过强立体声重建。有 `tc_present` 设置的通道可以参考 `tc_select`。

`tc_present == 1` 表明通道存在。对于在 `mc_header()` 中标记为不存在的音频通道,`tc_present` 的值赋为零。

**ch\_present(ch)** 一个函数,指明音频通道 `ch` 是否如 `mc_header()` 中指明的那样存在。

**block\_type[gr][ch]** 指示颗粒/通道的窗口类型(见滤波器组层Ⅲ的说明)。

block_type[gr]	窗类型
0	正常块
1	起始块
2	3 个短窗
3	结束块

`block_type` 给出了块中的值的汇总,以及变换的长度和计数信息(见图 III-A.4 简图,附录 C 有分析的说明)。多相滤波器组在 ISO/IEC 11172-3 2.4.3 节中说明。

对于长块的情况(`block_type` 不等于 2),IMDCT 对每 18 个输入值产生 36 个输出值。输出根据 `block_type` 而被窗口化,前一部分与前一块的最后一部分重叠。结果向量是一个波段的多相滤波器组的合成部分的输入。

对于短块(`block_type` 为 2),每块进行三次变换产生 12 个输出值。三个向量被窗口化和重叠。结果向量的两端均连接 6 个零产生一个长为 36 的向量,就像一个长变换的输出那样。

如果 `block_type` 不为零,其它一些变量设置为缺省值:

`region0_count=7` (`block_type==1 or 3` 时)

`region0_count=8` (`block_type==2`)

region1\_count=36 big\_value 区中的其它剩余值包含于 region 1 中。

**dematrix\_length** 反矩阵通道在其中显式传输的 scalefactor\_band\_groups 的个数。最开始的 dematrix\_length 个 scalefactorband\_groups 没有联合视频信息(tc\_select)被传输。如果 dematrix\_length==0, 要通过反矩阵而重构的通道由 seg\_list\_present 决定。

### 3/2,3/1 和 2/2 配置 (4 位)

dematrix_select:	L	R	C	LS/S	RS	对 3/2 有效	对 3/1 有效	对 2/2 有效
0	0	0	0	0	0	y	y	y
1	x	0	0	0	0	y	y	y
2	0	x	0	0	0	y	y	y
3	x	x	0	0	0	y	y	y
4	0	0	x	0	0	y	y	n
5	x	0	x	0	0	y	y	n
6	0	x	x	0	0	y	y	n
7	0	0	0	x	0	y	y	y
8	0	x	0	x	0	y	y	y
9	0	0	x	x	0	y	n	n
10	0	0	0	0	x	y	n	y
11	x	0	0	0	x	y	n	y
12	0	0	x	0	x	y	n	n
13	0	0	0	x	x	y	n	y
14	x	0	0	x	0	n	y	n
15	—	—	—	—	—	n	n	n

### 3/0 和 2/1 配置(3 位)

dematrix_select:	L	R	C/S	对 3/0 有效	对 2/1 有效
0	0	0	0	y	y
1	x	0	0	y	y
2	x	x	0	y	y
3	0	0	x	y	y
4	x	0	x	y	y
5	0	x	x	y	y
6	—	—	—	n	n
7	—	—	—	n	n

**scalefactorband\_group** 为了传输 dematrix\_length 和段表,“比例因子带”被组成组。下表给出了长块(block\_type==0,1,3)和短块(block\_type==2)的组。对于短块,scalefactor band\_group 包含所有三个子块的相应值。

比例因子带中的每个 scalefactorband\_group(sbgr)的宽度和开始值:

sbgr #	长 块 (block_type == 0,1,3)		短 块 (block_type == 2)	
	sbgr width	sbgr begin	sbgr width	sbgr begin
0	3	0	1	0
1	3	3	1	1
2	3	6	1	2
3	1	9	1	3
4	1	10	1	4
5	1	11	1	5
6	1	12	1	6
7	1	13	1	7
8	1	14	1	8
9	1	15	1	9
10	1	16	1	10
11	1	17	2	11
12	1	18	—	13
13	1	19	—	—
14	2	20	—	—
15	—	22	—	—

**attenuation\_range[gr][ch]** 段表的衰减有四个不同的范围。下表指明衰减范围：

attenuation_range:	衰减的位数
0	2
1	3
2	4
3	5

**attenuation\_scale[gr][ch]** 衰减有两个不同步长。attenuation\_scale 决定步长。attenuation\_scale == 0, 步长为  $1/\sqrt{\sqrt{2}}$ 。attenuation\_scale == 1, 步长为  $1/\sqrt{2}$ 。

**seg\_length[gr][ch][seg]** 是 scalefactorband 组的数目, 与 tc\_select 相乘衰减后被复制到通道(ch)中。若 seg\_length == 0, tc\_select 和衰减就不被传输。未被选中的 scalefactorband\_groups 被设置为零。

**tc\_select[gr][ch][seg]** 指明被传输的通道数, 是段表处理的来源。tc\_select == 7 表明此段中的值是由反矩阵重建的。

**attenuation[gr][ch][seg][sbgr]** 对于每个 scalefactorband\_group, 要传输一个 attenuation(衰减), 对通道进行汇集。attenuation(衰减)的宽度从 2 位到 5 位, 可变, 由 attenuation\_range 指定。attenuation(衰减)的步长由 attenuation\_scale 决定, 从  $\sqrt{2}$  到  $\sqrt{\sqrt{2}}$  可变。若 tc\_select == 7, 说明通道进行反矩阵运算, 不传输 attenuation(衰减)。若 tc\_select == ch, 则传输的通道为选中的通道, 不传输 attenuation(衰减)。

**mc\_prediction\_on** 一位, 指示是否使用了 mc\_prediction。若用了则为 '1', 否则为 '0'。

**mc\_prediction[sbgr]** 一位, 指示在子波段组 sbgr 中是否使用了预测进行多通道冗余降低。若用了则为 '1', 否则为 '0'。

**predsi[sbgr,pci]** 预测器选择信息。指明在子波段组 sbgr 中以 pci 索引的预测器系数是否被传送。若系数被传输则为 '1', 否则为 '0'。

**pred\_coef[sbgr,pci]** 用于子波段组 sbgr 和索引 pci 中的子波段的实际预测系数。

### III-2.5.2.12 MC 音频数据层 I, II

**lfe\_allocation** 包含用于低频增强通道采样的量化器的信息。此域中的位形成无符号整数, 用于 ISO/IEC11172-3 的表 B.2“层 I 位分配表”中相关表的索引(对于子波段 0)。它给出了量化级数。若 Fs 为 48kHz, 应使用 ISO/IEC 11172-3 的表 B.2.a, 若 Fs 为 44.1kHz 或 32kHz 则用表 B.2.b, 不考虑比特率。

**allocation[mch,sb]** 包含多通道扩展通道 mch 的子波段 sb 采样的量化器的信息。与此分配域相关的子波段的数目 bound[mch] 取决于 mc\_coding\_mode 和 mc\_coding\_mode\_extension。此域中的位形成一个无符号整数, 作为 ISO/IEC 11172-3 的表 B.2“层 I 位分配表”中相关表的索引。它给出了量化级数。若 Fs 等于 48kHz, 应用表 B.2.a, 若 Fs 等于 44.1kHz 或 32kHz, 应用表 B.2.b, 不考虑比特率。

**scfsi[mch,sb]** 比例因子选择信息。给出了有关多通道扩展通道 mch 的子波段 sb 的被传送的比例因子数的信息。对每一子波段, 帧被分成三个相同的部分, 每部分有 12 个子波段样本。

'00' 对应于 0,1,2 部分的三个比例因子被传送。

'01' 传送两个比例因子, 第一个对部分 0,1 有效, 第二个对部分 2 有效。

'10' 传送一个比例因子, 对所有三个部分有效。

'11' 传送一个比例因子, 第一个对部分 0 有效, 第二个对部分 1 和 2 有效。

**delay\_comp[sbgr,px]** 3 位, 规定了在子波段组 sbgr 和预测器索引 px 中用于延迟补偿的 0,1,2,...,7 个子波段样本的移位。

**pred\_coef[sbgr,px,pci]** 在子波段组 sbgr 和预测器索引 px 中的最高到第二阶的预测器的实际系数。

**lf\_scalefactor** 指明低频效果通道中的再量化样本进行倍增所需的因子。六位, 构成无符号整数, 作为 ISO/IEC 11172-3 表 B.1“层 I, II 比例因子”的索引。

**scalefactor[mch,sb,p]** 指明多通道扩展通道 mch 的帧的 P 部分子波段 sb 中再量化样本进行倍增所需的因子。六位构成一无符号整数, 作为 ISO/IEC 11172-3 表 B.1“层 I, II 比例因子”的索引。

**lf\_sample[gr]** 低频增强通道的颗粒 gr 中的单样本编码表示。

**samplecode[mch, sb, gr]** 多通道扩展通道 mch 的子波段 sb 的颗粒 gr 的三个连续样本的编码表示。

**sample[mch, sb, s]** 多通道扩展通道 mch 的子波段 sb 的样本 s 的编码表示。

### III-2.5.2.13 MC 音频数据层 III

**data\_present[gr][tc][sf]** 一个映象,描述实际被传输的数据(依赖于颗粒、传输通道和比例因子波段)。此映象不传输,但是通过确定由 dematrix\_select 或 segment\_list 所参考的 scalefactorband 而在解码器恢复。

**js\_carrier[gr][tc][sbgr]** 一个映象,描述哪个 scalefactorband\_group 数据(取决于颗粒,传输通道和比例因子波段组)被用作联合立体声传输的载体。此映象不传输,但是通过参考 tc\_select!=ch 而确定的 scalefactorband\_group,该映象可在解码器中恢复。

**matrix\_attenuation\_present** 指明 matrix\_attenuation 是否被传输。若传输则为‘1’。

**matrix\_attenuation\_lr[gr][ch][sbgr]** 在联合立体声编码的情况下,需要校正的值进行兼容向下混合信号 Lo 和 Ro 的能量保护。在解码器中为正确进行反矩阵,进行了衰减。实际的衰减因子计算如下:

$$\text{attenuation} = 1 / (\sqrt{\sqrt{2}} * \text{matrix\_attenuation\_l/r})$$

对于使用 Lo(Ro)通道的反矩阵运算,要使用 matrix\_attenuation\_l(matrix\_attenuation\_r)。反矩阵操作的改进在解码过程中说明。

**scfsi[tc][scfsi\_band]** 在层 III,比例因子选择信息的作用与在音频层 I 中类似。主要的区别在于使用变量 scfsi\_band 可以使 scfsi 适用于比例因子组而非单一的比例因子。比例因子在颗粒中的应用受控于 scfsi。只有当通道在两个颗粒中均传输的情况下,比例因子选择信息才传输。其它的被置为零。

scfsi[scfsi_band]	
'0'	对于每个颗粒比例因子被传输
'1'	对于颗粒 0 和 1 比例因子被传输

若短窗被打开,即对于一个颗粒 block\_type==2,那么对于此帧 scfsi 总为零。

**scfsi\_band** 控制针对比例因子组的比例因子选择信息(scfsi\_bands)的使用。

scfsi_band	比例因子波段(见表 III-B.8)
0	0,1,2,3,4,5
1	6,7,8,9,10
2	11...15
3	16...20

**part2\_3\_length[gr][tc]** 此值包含在比例因子和哈夫曼编码中使用的 main\_data 的位数。

**big\_values[gr][tc]** 按不同的哈夫曼码表编码的每一个颗粒的谱值。整个频率范围从零到奈奎斯特频率被分成几个区间，用不同的码表编码。根据最大的量化值进行分割，这是基于较高频率的值幅度较低或根本不需全部编码的假设。从高频率开始，对量化值为零的对进行计数，称为“rzero”，然后，对量化值绝对值不超过 1（例如：只有 3 个可能的量化值）的四元组进行计数，称为“count1”，此后，剩下偶数个值。最后，在向下扩展到零的谱的区间中的值对的数目称为“big\_values”。这个范围内的最大绝对值为 8191，下图表明了此分割：

```
xxxxxxxxxxxxxxxxxxxxx-----00000000000000000000000000000000  
|           |           |           |  
1           bigvalues * 2 :bigvalues * 2+count1 * 4           iblen
```

值 000 为全零, 其数目为 2 的倍数。

值---为-1,0 或+1,其数目是 4 的倍数。

值  $xxx$  不限，其数目是 2 的倍数。

Iblen 是 576。

**global\_gain[gr][tc]** 量化器的步长值信息，在附带信息变量 global\_gain 中传输。它是用对数量化的。对于 global\_gain 的使用，参考 ISO/IEC 11172-3 的 2.4.3.4 节“再量化和全标度的公式”中的公式。

**scalefac\_compress[gr][tc]** 选择比例因子的传输位数,根据下表选择:

若 block-type 为 0,1,或 3:

slen1: 比例因子段 0 到 10 的比例因子長度。

slen2: 比例因子段 11 到 20 的比例因子长度。

若 block-type 为 2;

slen1: 比例因字段 0 到 5 的比例因子长度。

slen2: 比例因字段 6 到 11 的比例因子长度。

scalefac_compress[gr]	slen1	slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2

续表

scalefac_compress[gr]	slen1	slen2
13	3	3
14	4	2
15	4	3

**table\_select[gr][tc][region]** 根据最大量化值和信号的局部统计使用不同的哈夫曼码表。表Ⅲ-B.7中给出了全部可能的32个表。

**subblock\_gain[gr][tc][window]** 指示一个子块的相对全局增益的增益偏移(量化:因子4)。只与类型2的块(短窗)一起传递。在解码器中,子块的值被除以4(subblock\_gain[window])见ISO/IEC 11172-3 2.4.3.4节——再量化和全标度公式。

**region0\_count[gr][tc]** 对谱的进一步分割以增强哈夫曼编码器的性能。此分割是对big\_values中描述的区间的细分。细分的目的是获得更高的错误鲁棒性和编码效率。使用了三个区间,称为区间0,1和2。根据最大量化值和局部信号统计,每个区间使用不同的哈夫曼码表。

region0\_count和region1\_count的值用于指示区间的边界。区间边界根据比例因子段的谱分割进行对整。

region0\_count域含的值比区间0中的比例因子段数少1。在短块条件下,每个比例因子段计数三次,每个短窗一次,因此当region0\_count值为8表明区间1从比例因子段3开始。

若block\_type==2,颗粒中的比例因子段的总数为 $12 * 3 = 36$ 。若block\_type!=2,比例因子段数为21。

**region1\_count[gr][tc]** region1\_count比区间1的比例因子段数少1。另外,若block\_type==2,代表不同时间槽的比例因子段分别计数。

**preflag[gr][tc]** 这是对附加高频量化值放大的捷径。若preflag被设置,则在比例因子上加一个表的值(见表Ⅲ-8.6)。这与对再量化比例因子用表值倍乘是一样的。若block\_type==2(短块),不使用preflag。

**scalefac\_scale[gr][tc]** 比例因子根据scalefac\_scale用步长2或 $\sqrt{2}$ 进行对数量化。下表说明了对于每种步长在再量化方程中使用的比例因子乘数。

scalefac_scale[gr]	scalefac_multiplier
0	0.5
1	1

**count1table\_select[gr][tc]** 此标志选择量化值不超过1的四元组区间的两种可能的哈夫曼码表。

count1table_select[gr]	
0	表Ⅲ-B.7-A
1	表Ⅲ-B.7-B

**scalefac\_l[gr][tc][sfb], scalefac\_s[gr][tc][sfb][window], is\_pos[sfb]** 比例因子用来给量化噪音着色。若量化噪音以正确的形状着色,那么将被完全掩盖。不像层 I 和 II,层 III 比例因子与量化信号的局部最大值无关。在层 III,比例因子用于解码器,以获得组值的分割因子。对于层 III,组延伸到几个频率线。这些组称为比例因字段,且被选来尽可能地近似严格意义上的段。

scalefac\_compress 表显示比例因子 0...10 的范围是 0 到 15(最大长度为 4 位),比例因子 11...21 的范围是 0 到 7(最大长度为 3 位)。将谱细分为比例因字段对于每一块长和采样频率是固定的,且存储在编码器和解码器的表中(见表 III-B.8)。表中最高频率线以上的频率线的比例因子为零,意味着实际倍乘因子是 1.0。

比例因子是用对数量化的。量化步长用 scalefac\_scale 设置。

未被传输通道选中的比例因字段的比例因子不被传输。这意味着比例因子将被打包在一起传输,在解码或矩阵求逆时解包。

**byte\_align\_bit** 用于 mc\_main\_data 字节对整的私用位。

**huffmancodebits()** 哈夫曼编码数据。

huffmancodebits() 的语法表明了量化值如何编码。在 big\_values 分区内,绝对值小于 15 的量化值对直接进行哈夫曼编码。编码由表 III-B.7 的哈夫曼表 0 到 31 得到。值(x,y)总是成对编码。如果对量化值等于或大于 15 的值编码,这些值根据哈夫曼码被编码成不同的域。若一个值对的一个或两个值非零,则在码字中加一个或两个符号位。对于 big\_values 分区的哈夫曼表有三个组成参数:

hcod[x][y] 对应于值 x,y 的哈夫曼码表入口。

hlen[x][y] 对应于值 x,y 的哈夫曼码表长度。

linbits 当 linbitsx 或 linbitsy 被编码时,表示其长度。

huffmancodebits 的语法包括以下域和参数:

signv v 的符号(正为 0,负为 1)

signw w 的符号(正为 0,负为 1)

signx x 的符号(正为 0,负为 1)

signy y 的符号(正为 0,负为 1)

linbitsx 用于对值大于等于 15 的 x 值编码。只有在 hcoc 中 |x| 等于 15 时,才对此域编码。若 linbits 为零,则当 |x| == 15 时,实际上没有位被编码,则 linbitsx 值定义为零。

linbitsy 用于 y,与 linbitsx 相同。

is[1] 对于频率线 1 的量化值。

只对大于等于 15 的值编码时才使用 linbitsx 和 linbitsy 域。此域按无符号整数解释,且被加上 15 以获得编码值。对于块中最大值小于 15 的表,不用 linbitsx 和 linbitsy。注意,15 也可以用 linbits 为零的哈夫曼表编码。此时,由于 linbits 为零,linbitsx 域或 linbitsy 域实际未编码。

在 count1 分区内,值小于等于 1 的四元组被编码。用表 III-B.7 中的表 A 和 B 进行哈夫曼编码。对于每一非零值,在哈夫曼编码符号后面加一个符号位。

count1 分区的哈夫曼表由以下参数组成:

**hcod[v][w][x][y]** 对应于值 v,w,x,y 的哈夫曼码表入口。

**hlen[v][w][x][y]** 对应于值 v,w,x,y 的哈夫曼码表长度。

哈夫曼码表 B 不是真的四维码,因为它由平凡码(trivial code)构成:0 用 1 编码,1 用 0 编码。

count1 分区以上的量化值为零,因此不编码。

此文件中用的参数“count1”指示在 count1 区间的哈夫曼编码数。然而与 big\_value 区间不同,在此语法中,count1 分区中的值个数不被编码。只有当颗粒中全部的位(如 part\_3\_length 规定)被用尽才能知道 count1 区间的结尾,且 count1 的值在对 count1 区间解码后才能清楚地知道。

哈夫曼数据的顺序依赖于颗粒的 block\_type。若 block\_type 是 0,1 或 3,哈夫曼编码数据按频率增加顺序排序。

若 block\_type == 2(短块),哈夫曼编码数据的顺序与此颗粒的比例因子值顺序相同。哈夫曼编码数据用于顺序的比例因子段,从比例因子段 0 开始。在每个比例因子段内,数据用于顺序的时间窗,从窗 0 开始到窗 2 结束。在每个窗内,量化值按频率增加的顺序安排。

**lfe\_table\_select** 确定用于对低频增强通道谱值解码的哈夫曼码表。解释与 table\_select 相同。

**lfe\_hc\_len** 确定用于两个颗粒的低频增强通道谱值的哈夫曼编码的总长度。

**lfe\_gain** 确定低频增强通道的量化步长。解释与 global\_gain 相同。

**lfe\_main\_data()** 包括两个颗粒的低频增强通道的哈夫曼编码谱值。**lfe\_main\_data()** 解释就像 Huffmancodebits() 结构那样,只含 big\_values 和 zero\_values。与 Huffmancodebits 的 count1 相似,在 lfe\_main\_data() 中的哈夫曼编码个数(如 lfe\_bigval)不传输。取而代之,在所有 lfe\_hc\_len 中指明的位用尽之后,它通过哈夫曼解码恢复。与 Huffmancodebits() 结构不同,解码值 x 和 y 分别表示颗粒 0 和 1 的谱系数值。

### III-2.5.2.14 ML 音频数据层 I 和层 II

**allocation[mlch,sb]** 包含在多语种扩展通道 mlch 中的子波段 sb 的样本量化器的信息。此域中的位形成无符号整数,用作 ISO/IEC 11172-3 表 B. 2“层 I 位分配表”中相关表的索引,表中给出了量化级数。若 Fs 为 48kHz,用表 B. 2. a,若 Fs 等于 44. 1kHz 或 32kHz,用表 B. 2. b,不考虑比特率。若在多语种通道中使用半采样频率(multi\_lingual\_fs == 1)用本推荐|国际标准中的表 III-B. 1

**scfsi[mlch,sb]** 比例因子选择信息。给出多语种扩展通道 mlch 中子波段 sb 的被传送的比例因子数的信息。对于每个子波段,帧被分成三个相同的部分,每部分 12 个样本。

'00' 传输三个比例因子,相应于 0,1,2。

'01' 传输两个比例因子,第一个对部分 0 和 1 有效,第二个对部分 2 有效。

'10' 传输一个比例因子,对三部分有效。

'11' 传输两个比例因子,第一个对部分 0 有效,第二个对部分 1 和 2 有效。

**scalefactor[mlch,sb,p]** 指明多语种扩展通道 mlch 的帧的 p 部分子波段 sb 的再量化样本应倍乘的因子。6 个位构成无符号整数,作为 ISO/IEC 11172-3 表 B. 1“层 I, II 比例因子”的索引。

**samplecode[mlch,sb,gr]** 多语种扩展通道 mlch 的子波段 sb 的颗粒 gr 的三个连续样本的编码表示。若 multi\_lingual\_fs 等于‘0’(全采样频率), 颗粒数 ngr 等于 12; 若 multi\_lingual\_fs 等于‘1’(半采样频率), ngr 等于 6。

**sample[mlch,sb,s]** 多语种扩展通道 mlch 的子波段 sb 的样本 s 的编码表示。

### III-2.5.2.15 ML 音频数据层Ⅲ

见 ISO/IEC 11172-3 的 2.4.2.7 节或本推荐|国际标准的 III-2.4.2.7 节, 依赖于 multi\_lingual\_fs。

### III-2.5.2.16 MPEG-1 辅助数据

见 ISO/IEC 11172-3 的 2.4.2.8 节。

### III-2.5.2.17 扩展帧

**ext\_header** 扩展比特流中含同步和状态信息的部分。

**ext\_data** 比特流中的多通道/多语种域部分, 包括在 MPEG1 兼容比特流部分中不能传输的位。

**ext\_ancillary\_data** 扩展比特流中可用于运载辅助数据的部分。

### III-2.5.2.18 扩展头部

**ext\_synword** 12 位的串‘ ’, 用作 MPEG 1 兼容比特流和扩展比特流的同步。

**ext\_crc\_check** 强制的 16 位校验字,CRC\_check 的计算从 ext\_length 字节的第一位开始, 与后续的位组合在一起共 128 位。

**ext\_length** 11 位的数, 指明扩展帧的总字节数。此数不超过 1536 个字节。

**reserved\_bit** 保留, 为将来使用。

## III-2.5.3 音频解码过程

### III-2.5.3.1 概述

一般解码过程与 ISO/IEC 11172-3 的 2.4.3 节很相似。包含位分配、解码、解码比例因子选择信息、解码比例因子、在层 I 和 II 对子波段样本进行再量化、解码附带信息、解码比例因子、哈夫曼解码、再量化、记录、滤波器组合成和层 III 的别名减少。

根据 ISO/IEC 11172-3 的 2.4.3 节, 第一步是对向后兼容信号 Lo,Ro 解码。MPEG-1 辅助数据开始假设为包含编码多通道扩展信号。如果强制的 CRC\_check 产生有效结果, 那么开始多通道解码。每三个连续的 ISO/IEC 11172-3 层 I 帧中只有一个含有多通道头部。多通道扩展的前 12 位组成多通道头部, 提供中央通道、环绕通道、LFE 通道的存在信息, 后跟逆矩阵过程, 包含在多通道扩展比特流中的多语种通道数, 多语种通道的采样频率和用于多语种通道的编码层。

本推荐|国际标准提供了对 ISO/IEC 11172-3 定义的三个层的比特率扩展的可能性, 而且与此标准保持向后兼容性。这可从包含多通道/多语种数据的剩余数据的扩展比特流得到。此比特流的结构在附录 III-A.2 中描述。MPEG-1 兼容比特流至少应包含 MPEG 1 音频数据和 MC 头部。

必须遵循直接紧跟 mc\_header 的 CRC\_chedu 字的误差检测法是与用于 ISO/IEC 11172-3 中的方法一样的, 并在 ISO/IEC 11172-3 中的 2.4.3.1 节中描述。

### III-2.5.3.2 合成编码方式

#### 1. 传输通道开关

传输通道(tc\_allocation)的音频通道的分配对整个带宽或根据 tc\_sbgr\_select 的值对单独子波段组有效。根据配置,tc\_allocation 域确定哪个音频通道包含在传输通道中。对每种可能性,为了获得输出通道,一个解码矩阵被用于所有传输通道的子波段域。矩阵在下面给出。结果是信号仍需反规范化(见 III-2.5.3.3 节)。若 dematrix\_procedure = '11'(见 III-2.5.2.8 节)被选中,则所有信号可直接从传输通道获得,不需要逆矩阵。在此条件下,使用 tc\_allocation 的缺省值‘0’。若 dematrix\_procedure == ‘10’,在环绕通道中需要以下过程。

(1) 在 3/2 配置下,计算单音环绕信号:

$$jS^w = 0.5 * (jLS^w + jRS^w)$$

(2) 在  $jS^w$  上加一带通滤波器,限制带宽范围在 100—7000Hz 之间。

(3) 限带结果信号被用于逆矩阵运算。

(4) 在 3/2 配置时要对  $jLS^w$  和  $jRS^w$  进行下述处理,或在 3/1 配置时对  $jS^w$  作输出前处理(在逆矩阵前不进行这些操作):

① -90 度相移。

② 半杜比 B-型解码。

#### 解码矩阵:

下述的矩阵方程对不同的通道配置均有效。逆矩阵方程对第二立体声节目无效。

3/2 配置,逆矩阵过程等于‘00’和‘01’:

tc_allocation	解码矩阵
0	$L^w = Lo - T2 - T3$
	$R^w = Ro - T2 - T4$
	$C^w = T2$
	$LS^w = T3$
	$RS^w = T4$

tc_allocation	解码矩阵
1	$C^w = Lo - T2 - T3$
	$R^w = Ro - C^w - T4$
	$L^w = T2$
	$LS^w = T3$
	$RS^w = T4$

tc_allocation	解码矩阵
2	$C^w = Ro - T2 - T4$
	$L^w = Lo - C^w - T3$
	$R^w = T2$
	$LS^w = T3$
	$RS^w = T4$

tc_allocation	解码矩阵
3	$LS^w = Lo - T3 - T2$
	$R^w = Ro - T2 - T4$
	$C^w = T2$
	$L^w = T3$
	$RS^w = T4$

tc_allocation	解码矩阵
4	$L^w = Lo - T_2 - T_3$
	$RS^w = Ro - T_4 - T_2$
	$C^w = T_2$
	$LS^w = T_3$
	$R^w = T_4$

tc_allocation	解码矩阵
5	$LS^w = Lo - T_3 - T_2$
	$RS^w = Ro - T_4 - T_2$
	$C^w = T_2$
	$L^w = T_3$
	$R^w = T_4$

tc_allocation	解码矩阵
6	$C^w = Ro - T_2 - T_4$
	$LS^w = L_o - T_3 - C^w$
	$R^w = T_2$
	$L^w = T_3$
	$RS^w = T_4$

tc_allocation	解码矩阵
7	$C^w = Lo - T_2 - T_3$
	$RS^w = Ro - T_4 - C^w$
	$L^w = T_2$
	$LS^w = T_3$
	$R^w = T_4$

3/2 配置,逆矩阵过程等于‘10’:

tc_allocation	解码矩阵
0	$L^w = Lo - T_2 + jS^w$
	$R^w = Ro - T_2 - jS^w$
	$C^w = T_2$
	$jLS^w = T_3$
	$jRS^w = T_4$

tc_allocation	解码矩阵
1	$C^w = Lo - T_2 + jS^w$
	$R^w = Ro - C^w - jS^w$
	$L^w = T_2$
	$jLS^w = T_3$
	$jRS^w = T_4$

tc_allocation	解码矩阵
2	$C^w = Ro - T_2 - jS^w$
	$L^w = Lo - C^w + jS^w$
	$R^w = T_2$
	$jLS^w = T_3$
	$jRS^w = T_4$

3/1 配置,逆矩阵过程等于‘00’和‘01’:

tc_allocation	解码矩阵
0	$L^w = L_o - T_2 - T_3$
	$R^w = R_o - T_2 - T_3$
	$C^w = T_2$
	$S^w = T_3$

tc_allocation	解码矩阵
1	$C^w = L_o - T_2 - T_3$
	$R^w = R_o - C^w - T_3$
	$L^w = T_2$
	$S^w = T_3$

tc_allocation	解码矩阵
2	$C^w = R_o - T_2 - T_3$
	$L^w = L_o - C^w - T_3$
	$R^w = T_2$
	$S^w = T_3$

tc_allocation	解码矩阵
3	$S^w = L_o - T_2 - T_3$
	$R^w = R_o - T_2 - S^w$
	$C^w = T_2$
	$L^w = T_3$

tc_allocation	解码矩阵
4	$S^w = R_o - T_2 - T_3$
	$L^w = L_o - T_2 - S^w$
	$C^w = T_2$
	$R^w = T_3$

3/1 配置,逆矩阵过程等于‘1 0’:

tc_allocation	解码矩阵
0	$L^w = L_o - T_2 + jS^w$
	$R^w = R_o - T_2 - jS^w$
	$C^w = T_2$
	$jS^w = T_3$

tc_allocation	解码矩阵
1	$C^w = L_o - T_2 + jS^w$
	$R^w = R_o - C^w - jS^w$
	$L^w = T_2$
	$jS^w = T_3$

tc_allocation	解码矩阵
2	$C^w = R_o - T_2 - jS^w$
	$L^w = L_o - C^w + jS^w$
	$R^w = T_2$
	$jS^w = T_3$

3/0 配置:

tc_allocation	解码矩阵
0	$L^w = Lo - T_2$
	$R^w = Ro - T_2$
	$C^w = T_2$

tc_allocation	解码矩阵
1	$C^w = Lo - T_2$
	$R^w = Ro - C^w$
	$L^w = T_2$

tc_allocation	解码矩阵
2	$C^w = Ro - T_2$
	$L^w = Lo - C^w$
	$R^w = T_2$

2/2 配置：

tc_allocation	解码矩阵
0	$L^w = Lo - T_2$
	$R^w = Ro - T_3$
	$LS^w = T_2$
	$RS^w = T_3$

tc_allocation	解码矩阵
1	$R^w = Ro - T_3$
	$LS^w = Lo - T_2$
	$L^w = T_2$
	$RS^w = T_3$

tc_allocation	解码矩阵
2	$L^w = Lo - T_2$
	$RS^w = Ro - T_3$
	$LS^w = T_2$
	$R^w = T_3$

tc_allocation	解码矩阵
3	$LS^w = Lo - T_2$
	$RS^w = Ro - T_3$
	$L^w = T_2$
	$R^w = T_3$

2/1 配置：

tc_allocation	解码矩阵
0	$L^w = Lo - T_2$
	$R^w = Ro - T_2$
	$S^w = T_2$

tc_allocation	解码矩阵
1	$S^w = Lo - T_2$
	$R^w = Ro - S^w$
	$L^w = T_2$

tc_allocation	解码矩阵
2	$S^w = R_o - T_2$
	$L^w = L_o - S^w$
	$R^w = T_2$

## 2. 动态交叉会话

如果某一通道的一定的子波段组的动态交叉会话被允许，则位分配和编码子波段样本将不含在比特流内，但是必须从相应的传输通道的被传输的子波段样本复制。dyn\_cross\_mode 域指明了子波段样本从哪一个通道复制到哪个通道。但是比特流中仍含有比例因子选择信息和用于对子波段样本再标度的比例因子信息。

下列规则用于不同的配置：

### 3/2 配置

如果无通道  $T_2$ ，可能被分配到此通道上的表现通道 C 或 L 就通过把传输的  $L_o$  的子波段样本乘以  $T_2$  中的比例因子得到。如果所缺的表现通道是 R，此通道通过把  $R_o$  的子波段样本乘以  $T_2$  中的比例因子得到。如果缺少包含 L 或  $L_s$  的传输通道  $T_3$ ，则表现通道 L 或  $L_s$  将通过把  $L_o$  的子波段样本乘以  $T_3$  的比例因子得到。如果缺少传输通道  $T_4$ ，包含于其中的 R 或  $R_s$  将通过把  $R_o$  的子波段样本乘以  $T_4$  中的比例因子得到。表中的项  $TS_{ij}$  表示传输通道 j 中的子波段样本需从 i 中复制得到。传输通道  $T_1$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以比例因子  $scf_i$  得到。传输通道  $T_j$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以比例因子  $scf_j$  得到。其余解码与无动态交叉会话时相同。

### 3/1 配置

如果缺少传输通道  $T_2$ ，有可能分配于此传输通道的表现通道 L 或 C 由  $L_o$  的子波段样本乘以  $T_2$  中的比例因子得到。如果缺少表现通道 R，由  $R_o$  的子波段样本乘以  $T_2$  中的比例因子得到。如果缺少表现通道 S，由  $L_o$  和  $R_o$  的子波段样本乘以  $T_2$  中的比例因子得到。除了  $T_3$  的比例因子将被用于子波段样本的再量化以外，相同的过程对传输通道  $T_3$  也有效。

表中的项  $TS_{ij}$  意味着传输通道 j 内的样本须从 i 中复制。传输通道  $T_i$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以比例因子  $scf_i$  得到。传输通道  $T_j$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以  $scf_j$  得到。其余的解码与无动态交叉会话时相同。

### 3/0 配置

如果缺少传输通道  $T_2$ ，有可能分配于此传输通道的表现通道 L 或 C 由  $L_o$  的子波段样本乘以  $T_2$  的比例因子得到。如果缺少表现通道 R，由  $R_o$  的子波段样本乘以  $T_2$  的比例因子得到，其余的解码与无动态交叉会话时期相同。

### 2/2 配置

如果缺少传输通道  $T_2$ ，有可能分配给此传输通道的表现通道 L 或  $L_s$  由  $L_o$  的子波段样本乘以  $T_2$  的比例因子获得。如果缺少传输通道  $T_3$ ，有可能分配于此传输通道的表现通道 R 或  $R_s$  由  $R_o$  的子波段样本乘以  $T_3$  的比例因子得到。

表中的项  $TS_{ij}$  表示传输通道  $j$  的子波段样本须由  $i$  复制。传输通道  $T_i$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以比例因子  $scf_i$  得到。传输通道  $T_i$  的合成滤波器的输入样本由子波段样本  $TS_{ij}$  乘以比例因子  $scf_i$  得到。其余的解码过程与无动态交叉会话时相同。

## 2/1 配置

如果缺少传输通道  $T_2$ , 有可能分配于此传输通道的表现通道  $L$  由  $L_0$  的子波段样本乘以  $T_2$  的比例因子获得。如果缺少表现通道  $R$ , 由  $R_0$  的子波段样本乘以  $T_2$  的比例因子获得。如果缺少表现通道  $S$ , 由  $L_0$  和  $R_0$  的子样本乘以  $T_2$  的比例因子获得。其余的解码与无动态交叉会话时相同。

### 3. MC-预测

如果  $MC\_prediction\_on$  位被设置, 且同时  $MC\_prediction[sbgr]$  位被设置, 则  $predsi[sbgr][px]$  位将决定哪个预测器被使用, 以及每个子波段组  $sbgr$  有多少系数  $pred\_coef[sbgr][px][pc]$  被传输。如果  $predsi[sbgr][px]$  是 1, 2 或 3, 延迟补偿  $delay\_comp[sbgr][px]$  和下面的 1, 2 或 3 预测器系数索引必须从比特流中读出。预测器系数索引根据下述方程反量化:

$$pred\_coef[sbgr][px] = (x - 127) / 32$$

如果被传输的系数少于 3 个, 余下的  $pred\_coef[sbgr][px][pc]$  被设置为零。如果  $predsi[sbgr][px]$  是 0, 所有相应的  $pred\_coef[sbgr][px][pc]$  设成 0。

存储在  $pred\_coef[sbgr][px][pc]$  中的预测器系数的含义取决于  $tc\_allocation$  和  $dyn\_cross\_mode$ , 如下:

以  $T_2, T_3, T_4$  为序

- 如果传输通道含有  $C^W$ , 则后两个最多含 3 个系数的集合是  $pred\_coef\_C0[sbgr][k]$  和  $pred\_coef\_C1[sbgr][k]$
- 如果传输通道含有  $S^W$ , 则后两个最多含 3 个系数的集合是  $pred\_coef\_S0[sbgr][k]$  和  $pred\_coef\_S1[sbgr][k]$
- 如果传输通道含有  $LS^W$ , 则后面一个最多含 3 个系数的集合是  $pred\_coef\_LS[sbgr][k]$
- 如果传输通道含有  $RS^W$ , 则后面一个最多含 3 个系数的集合是  $pred\_coef\_RS[sbgr][k]$
- 如果传输通道含有  $L^W$ , 则后面一个最多含 3 个系数的集合是  $pred\_coef\_L[sbgr][k]$
- 如果传输通道含有  $R^W$ , 则后面一个最多含 3 个系数的集合是  $pred\_coef\_R[sbgr][k]$
- 如果传输通道含有  $Txy$ , 则后两个最多含 3 个系数的集合是  $pred\_coef\_T0[sbgr][k]$  和  $pred\_coef\_T1[sbgr][k]$

根据传输通道分配( $tc\_allocation$ )和动态交叉会话( $dyn\_cross\_mode$ )依赖于传输通道  $T_2, T_3, T_4$  中传输的是哪个信号, 计算相应的三个, 两个或一个预测信号。

$$C^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} C0[sbgr,k] * T0(n-\text{delay\_comp} - k) + \sum_{k=0}^2 \text{pred\_coef\_} C1[sbgr,k] * T1(n-\text{delay\_comp} - k)$$

$$S^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} S0[sbgr,k] * T0(n-\text{delay\_comp} - k) + \sum_{k=0}^2 \text{pred\_coef\_} S1[sbgr,k] * T1(n-\text{delay\_comp} - k)$$

$$LS^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} LS[sbgr,k] * T0(n-\text{delay\_comp} - k)$$

$$RS^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} RS[sbgr,k] * T1(n-\text{delay\_comp} - k)$$

$$L^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} L[sbgr,k] * T0(n-\text{delay\_comp} - k)$$

$$R^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} R[sbgr,k] * T1(n-\text{delay\_comp} - k)$$

$$Txy^{w^A}(n) = \sum_{k=0}^2 \text{pred\_coef\_} T0[sbgr,k] * T0(n-\text{delay\_comp} - k) + \sum_{k=0}^2 \text{pred\_coef\_} T1[sbgr,k] * T1(n-\text{delay\_comp} - k)$$

通过在预测信号上加上传输的预测误差信号,子波段组 sbgr 的信号用下面相应的三个,两个或一个方程重建:

$$C^w = C^{w^A} + \epsilon_C$$

$$LS^w = LS^{w^A} + \epsilon_{LS}$$

$$RS^w = RS^{w^A} + \epsilon_{RS}$$

$$L^w = L^{w^A} + \epsilon_L$$

$$R^w = R^{w^A} + \epsilon_R$$

$$Txy = Txy^{w^A} + \epsilon_{t_{xy}}$$

### III. 2. 5. 3. 3 反规范化过程

在解码器中,首先,所有的加权信号  $L^w, C^w, R^w, LS^w, RS^w$  必须乘以消权因子以消去权。然后,这些信号才能被乘以反规范化因子,恢复在编码器端进行的衰减,以避免计算兼容信号时过载。

dematrix_procedure	信 号	无权因子	逆规范化因子
0,2	$L^w, R^w$	1	$1 + \sqrt{2}$
	$C^w, LS^w, RS^w$	$\sqrt{2}$	
1	$L^w, R^w$	1	$1.5 + 0.5 * \sqrt{2}$
	$LS^w, RS^w$	2	
	$C^w$	$\sqrt{2}$	
3	$L^w, R^w, C^w, LS^w, RS^w$	1	1

### III-2. 5. 3. 4 合成子波段滤波器

见 ISO/IEC 11172-3 的 2.4.3.2.2 节

### III-2.5.3.5 层Ⅲ解码

#### 1. 层Ⅲ段表

在最小配置下使用少量位时,段表语法允许灵活的多通道信号联合立体声编码。其主旨是从传输通道(TC<sub>s</sub>)的谱数据池中组成每个音频输出通道。这在通道谱(段)的不同部分是不相同的。对于每一段,要传输源 TC 的长度和编号(seg\_length,相应以 scalefactor-band-groups 和 tc\_select 为单位)。对于层Ⅲ,定义了以下 TC 编号:

TC#	通 道	符 号
0	Lo	left_comp_chan
1	Ro	right_comp_chan
2	L	left_chan
3	R	right_chan
4	C	center_chan
5	Ls/S	left_surr_chan,mono_surr_chan
6	Rs	right_surr_chan
7	“反矩阵”	

为了传输第二立体声节目(surround == '11'),TC #5 和 #6 分别用于左和右通道。如果用了 dematrix\_procedure 3(无矩阵),左和右通道信号分别在 TC #0 和 #1 中传输而非在 TC #2 和 #3 中。

对于每个 TC,存在一个与 MPEG-1 层Ⅲ音频通道相似的数据结构。例如:附带信息和哈夫曼编码谱值。tc\_present 标志指示哪个通道被传输了,例如,在 mc\_audio 比特流中含有多少附带信息和主信息。对于 MPEG-2,每个通道的附带信息的总量是变的。除这些区别外,哈夫曼编码值的解码工作与 MPEG-1 解码器一样。

一个音频输出通道 ch 的每个段有一个相应于 TC 的映象(tc\_select == ch),但是为了合成编码,它被赋值给不同的 TC。在此条件下,在 TC 谱数据上加一个传输的衰减值,以恢复音频输出通道的谱数据。在特殊情况下,tc\_select == 7 时,相应的段由逆矩阵运算重建。

对于几种段表类型,定义了捷径:

- seglist\_present == 0 表明所有被覆盖的 scalefactorband-groups 的数据的段表由逆矩阵重建。(即最大段长,tc\_select == 7)
- seglist\_nodef == 0 表明所有被覆盖的 scalefactorband-groups 的数据有一简单“缺省”段表在相应的 TC 内传输。(即最大段长,tc\_select == ch)
- seglist\_repeat == 1 表明在颗粒 #1 中使用已传输的颗粒 #0 的相同的段表。

段表可以只对一个颗粒有效,也可以如 segment\_list\_repeat 所标,对帧内的两个颗粒均有效。seg\_length 为 0 表明段表结束,通道谱的剩余部分设为零。

对于高于 scalefactorband\_group 边界(由 dematrix\_length 表示)的频率,段表用于

标明将合成编码的通道。当 scalefactorband\_groups 低于 dematrix\_length 时, 使用一种不太灵活的方法分配实际传输通道, 此方法不允许合成编码。

dematrix\_select 是个 3..4 位的值, 有 14 种可能的分配(对于 3/2 配置)。它用于说明哪个通道被逆矩阵, 哪个被传输。这里也许包含两个、一个甚至没有通道通过矩阵求逆而被重建。由于段表是为每个颗粒传输的, 所以 dematrix\_select 对两个颗粒都有效。

## 2. 层 III 的解码过程

如果可得到一个扩展比特流, 其访问单元将包括 mc\_composite\_status\_info 和 mc\_audio\_data 的部分。其内容在 MPEG-1 兼容比特流的主数据部分将与 mc\_composite\_status\_info 与/或 mc\_audio\_data 相连。mc\_data\_begin 指针的目标在包含相连比特流的缓冲区中计算。

解码过程包含以下几步:

### 缺省段表类型的扩充

这步通过对 seg\_list\_present, seg\_list\_nodef 和 seg\_list\_repeat 估算来进行。如果这些语法单元指明使用了捷径, 那么根据 2.5.3.5 节中的 1 项说明的简化定义, 对完整的 segment\_list 表示进行扩充。

### 解码映象的构造

构造一个描述了实际传输哪一个 TC 谱数据(依赖于颗粒, 传输通道和比例因子波段)的映象 data\_present[gr][ts][sb]。这一步通过确定由 dematrix\_select 或 segment\_lists(作为比例因子波段组的一部分)参考的比例因子带来完成。

另外, 构成一个描述了用作联合立体声传输的载体的 TC 谱数据(依赖于颗粒, 传输通道和比例因子波段组)的映象。这一步通过确定参考 tc\_select != ch 的子波段组来对每个音频通道 ch 作出。

### TC 信息的解码

tc\_present 指明了所有通道的 TC 数据再量化。这就像在层 III MPEG-1 解码器中使用以下单元的信息一样: block\_type, scalefac-1, scalefac-s, scfsi, part2\_3\_length, big\_values, global\_gain, scalefac\_compress\_table\_select, subblock\_gain, region\_0\_count, region\_1\_count, preflag, scalefac\_scale, count1table\_select。再量化操作在 ISO/IEC 11172-3 的 2.4.3.4 节中描述。解码数据是各自的音频输出通道的原频谱信息, 其中所有属于比例因子波段的系数(data\_present[gr][tc][sfb]==0)都被略去。

### 多通道预测解码

多通道预测的解码类似于在层 I 和层 II 中完成的那样, 与每个比例因子波段组 sbgr 无关。如果 mc\_prediction\_on 为“关”, 对任何比例因子波段组均不需要预测解码。如果 mc\_prediction\_sbgr[sbgr] 标志为“关”, 则相应的比例因子波段组不使用预测, 也不传输进一步的预测器信息。对每帧预测信息只传输一次, 用于两个颗粒。

### 可能的预测组合和预测器系数数目的计算

对于每个比例因子波段组 sbgr, 可能的预测组合依据以下规则计算:

如果(1)颗粒之一的数据被传输(data\_present[gr][mch][sfb(sbgr)]!=0 || data\_present[gr-1][ch][sfb(sbgr)]!=0)及(2)源和目的通道有相同的块类型, 则对于多通道

预测,每个通道都有可能成为目的通道。

对于每个可能的目的通道可能有一个或两个源通道(和预测器系数):

目标通道	源通道数	源通道
L	1	Lo
R	1	Ro
C,S	2	Lo,Ro
Ls	1	Lo
Rs	1	Ro

在联合立体声编码时( $js\_carrier[gr\_1][ch][sbgr] != 0$ ),两个源通道 Lo 和 Ro 被作为可能的源通道。npredcoef 的值表明了一个比例因子波段组的可能的预测器系数的总数。对于短块(block\_type == 2),11 以上的比例因子波段组的 npredcoef 定义为零(即:高于定义的比例因子波段组的数目)。

对每个可能的系数,在预测器选择信息 predsi[sbgr][]中有一位被传输,对于可能的系数的位,根据使用标准通道分配顺序的目标通道来排序,即:L,R,C,Ls,Rs。如果对于目的通道有两个可能的源通道,第一位表示 Lo,第二位表示 Ro 源通道。

如果 predsi[sbgr][pci] 为 0。相应的系数 pred\_coef[sbgr][pci] 设为 0。否则传输一个系数。系数的顺序与 predsi 信息的顺序相同,即:系数根据目的通道(粗排序)和源通道(细排序)排序。系数根据下表反量化:

传输值	反量化值
0	-0.61199
1	-0.24565
2	0.24565
3	0.61199
4	1.15831
5	1.97304
6	3.18805
7	5

### 预测信号的计算

在每个涉及到的目的通道内,预测信号计算如下,并加到传输的预测误差信号上:

$$L += pred\_coef\_L[sbgr] * Lo$$

$$R += pred\_coef\_R[sbgr] * Ro$$

$$C += pred\_coef\_C1[sbgr] * Lo + pred\_coef\_C2[sbgr] * Ro$$

```

Ls += pred_coef_Ls[sbgr] * Lo
Rs += pred_coef_Rs[sbgr] * Ro

```

### 对于联合立体声编码

JS += pred\_coef\_JS1[sbgr] \* Lo + pred\_coef\_JS2[sbgr] \* Ro

附加预测信号只对数据在每个通道分别传输(data\_present[gr][ch][sbgr]! = 0)的颗粒起作用。

#### • 通道数据的解码

每个音频数据的输出根据其段表和 dematrix\_select 配置从解码的 TC 数据中汇集。所有通过逆矩阵重建的比例因子波段组将被省略。data\_present 映象用来在目的通道的谱缓冲区中将编码的谱值从 TC 数据指向正确的比例因子波段组位置。

对于组合编码段(即:tc\_select! = ch && tc\_select! = 7),使用如下的传输衰减值对谱数据作比例操作:

确定基本衰减因子  $a_0 (1/\sqrt{2} \text{ 当 } \text{attenuation\_scale} == 1, \text{ 否则 } 1/\sqrt{\sqrt{2}})$

用实际衰减因子 a 作比例操作:

$$a = \begin{cases} a_0^{\text{attenuation}} & \text{for } \text{attenuation} < 0.75 * \text{max\_attenuation} \\ a_0^{\text{attenuation}-\text{max\_attenuation}} & \text{for } \text{attenuation} \geq 0.75 * \text{max\_attenuation} \end{cases}$$

with  $\text{max\_attenuation} = 2^{\text{attenuation\_range}+2}$

#### • 逆矩阵

逆矩阵用于重建所缺的比例因子波段组(只对 dematrix\_procedure<3 有效;对第二立体声节目无效,即:sourround==3)。

对于开始的 dematrix\_length 个比例因子波段组,逆矩阵部分由整个帧传输的 dematrix\_select 值确定。此边界以上,由段表的段与 tc\_select==7 定义。逆矩阵通过从 3/2 立体声配置的向下混合方程恢复零个、一个或两个通道而实现:

$$Lo = \alpha * (L + \beta * C + \gamma * Ls) \quad \text{及} \quad Ro = \alpha * (R + \beta * C + \gamma * Rs)$$

或在 3/1 立体声配置

$$Lo = \alpha * (L + \beta * C + \gamma * S) \quad \text{及} \quad Ro = \alpha * (R + \beta * C + \gamma * S)$$

其中  $\alpha$  是所有通道的总衰减,  $\beta$  和  $\gamma$  是中央和环绕通道的衰减因子。对于其它立体声配置,向下混合方程可由这些公式中之一得到,认为缺少的音频通道为零。在 dematrix\_procedure=='10' 条件下,逆矩阵方程要更改,如 III-2.5.3.2 节中 1 项所述。

对每个逆矩阵过程规定的衰减因子值:

dematrix_procedure	$\alpha$	$\beta$	$\gamma$
0	$1/(1+\sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$
1	$1/(1.5+0.5*\sqrt{2})$	$1/\sqrt{2}$	0.5
2	$1/(1+\sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$

幻象编码中央通道:

在中央通道的幻象编码时(`center == '11'`)，在逆矩阵的中央通道中出现的编码噪音可通过限制逆矩阵的中央通道带宽来抑制，如下所示：

采样频率/Hz	中央通道内有效的线数
48000	230
44100	238
32000	296

此步骤在第二通道逆矩阵之前进行。

联合立体声逆矩阵的修正：

如果 `matrix_attenuation_present` 标志为 `on`，通道逆矩阵的标准过程将被更改。对于逆矩阵操作，所有联合立体声编码的比例因子波段组数据由一个衰减因子预先作比例变换。此比例运算对逆矩阵方程的 Lo 和 Ro 两部分独立进行。

比例因子 m 由传输的 `matrix_attenuation` 值确定：

$$ml = 2^{-0.25 \cdot \text{matrix\_attenuation\_l}[\text{js\_ch}][\text{sbgr}]}$$

$$mr = 2^{-0.25 \cdot \text{matrix\_attenuation\_r}[\text{js\_ch}][\text{sbgr}]}$$

其中，`js_ch` 标明了传输联合立体声编码信号的实际谱数据的 TC；`sbgr` 表示比例因子波段组索引。

本过程对于联合立体声编码 L 和 C 有如下说明。谱数据在 L 通道的 TC 中传输(例如：TC #2)。因此，C 由相同的数据使用对应的衰减值重建。逆矩阵之前，L 和 C 由因子 m 进行比例变行。此比例操作不作用于最后的输出通道数据，如图 III-1 所示。

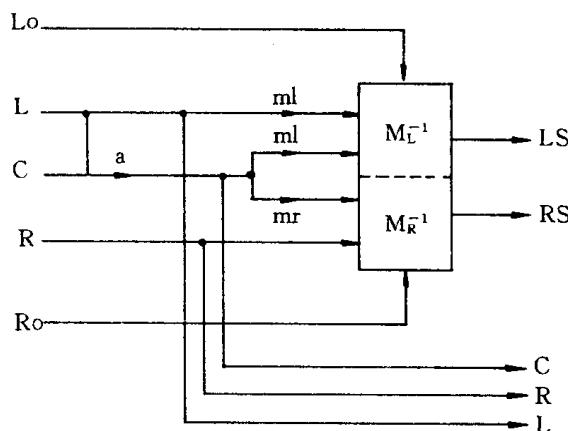


图 III-1 联合立体声逆矩阵的修改过程

#### • 合成滤波器组

应用合成滤波器组(见 ISO/IEC 11172-3 的 2.4.3.4 节)

#### 3. 层 III 的 LFE 的解码

LFE 值从简化的层 III 型比特流解码得到。

- 哈夫曼编码值的解码由 lfe\_table\_select 指示的哈夫曼码表进行。
- 传输的哈夫曼编码的解码进行到 lfe\_hc\_len 指示的所有位用尽才结束。此过程结束后, lfe\_bigval 的值已知道。为清楚起见, 此参数被引入来指示低频通道传输的谱数据使用的哈夫曼码字的数目。解码成分 x 和 y, 解释为颗粒 0 和 1 的相应谱系数的值。
- 依次, 反量化采用与 TC 数据的反量化相似的方法实现。为此, 使用 lfe\_gain, 比例因子和子块增益假设为零。
- 作为低频增强通道的合成滤波器组。用于短块(block\_type == 2)的数据重建的逆 MDCT (IMDCT) 被使用, 作为如 ISO/IEC 11172-3 的 2.4.3.4 节描述的合成混合滤波器组的一部分。因此, 在 ISO/IEC 11172-3 的 2.4.3.4 节/加窗(d)中描述的窗口类型被用于每个颗粒的 12 个 IMDCT 输出样本。由于每个颗粒只有一个窗口类型, “overlap add”处理简化成:

$\text{result}_i = y_i + s_i \quad \text{for } i=0 \text{ to } 5$

$s_i = y_{i+6} \quad \text{for } i=0 \text{ to } 5$

#### 4. 层 III ML 数据的解码

如要 multilingual\_fs == 0, 见 ISO/IEC 11172-3 的 2.4.3.4 节

如果 multilingual\_fs == 1, 见 III-2.4.3.2 节

为了应用 ML 主数据, nch 设成 no\_of\_multi\_lingual\_ch。

## 附录 III -A 图

(此附录为本推荐|国际标准的必需部分)

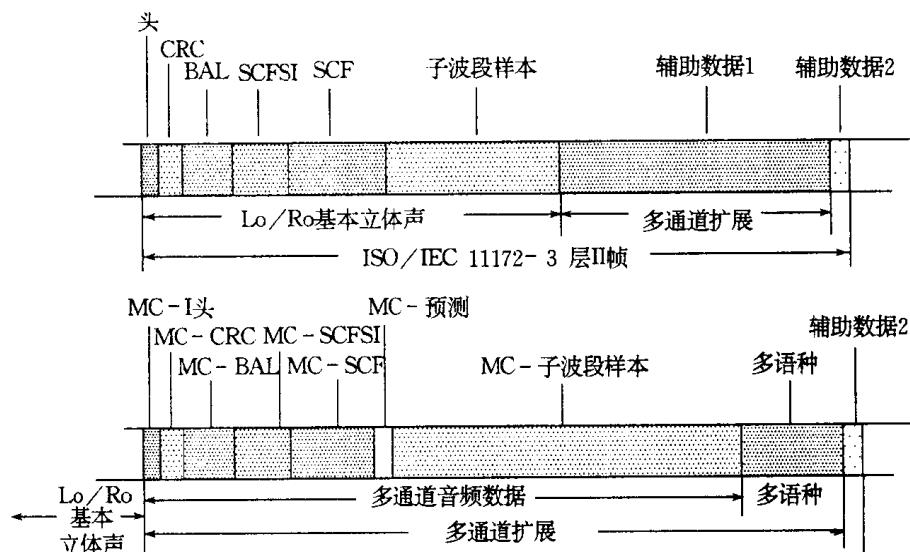


图 III-A-1 ISO 13818-3 层 II 多通道扩展结构,与 ISO 11172-3 层 I 向后兼容

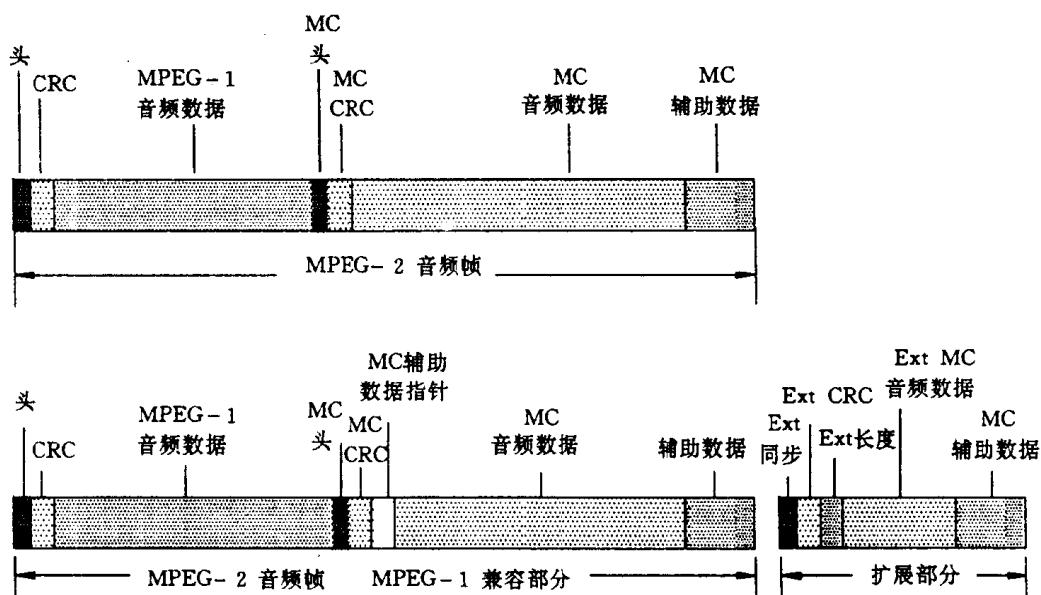


图 III-A-2 ISO 13818-3 多通道扩展层 I 结构,使用 ISO/IEC 11172-3  
兼容比特流和扩展比特流

## 附录 III -B 表

(此附录为本推荐 | 国际的必需部分)

**表 III -B-1 层 II , 每个子波段可能的量化采样频率 16, 22.05, 24kHz  
索引**

sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	—	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	16383
1	4	—	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	16383
2	4	—	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	16383
3	4	—	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	16383
4	3	—	3	5	9	15	31	63	127								
5	3	—	3	5	9	15	31	63	127								
6	3	—	3	5	9	15	31	63	127								
7	3	—	3	5	9	15	31	63	127								
8	3	—	3	5	9	15	31	63	127								
9	3	—	3	5	9	15	31	63	127								
10	3	—	3	5	9	15	31	63	127								
11	2	—	3	5	9												
12	2	—	3	5	9												
13	2	—	3	5	9												
14	2	—	3	5	9												
15	2	—	3	5	9												
16	2	—	3	5	9												
17	2	—	3	5	9												
18	2	—	3	5	9												
19	2	—	3	5	9												
20	2	—	3	5	9												
21	2	—	3	5	9												
22	2	—	3	5	9												
23	2	—	3	5	9												

续表

sb	nbal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
24	2	—	3	5	9												
25	2	—	3	5	9												
26	2	—	3	5	9												
27	2	—	3	5	9												
28	2	—	3	5	9												
29	2	—	3	5	9												
30	0	—															
31	0	—															

sblimit=30

Sum of nbal=75

表III-B-2 层III比例因子波段

本表列出了每个比例因子波段的带宽。对于长窗(类型0,1或3)每个采样频率有21个波段,对于短窗有12个波段。

16kHz采样频率,长块,线数576

比例因子波段	带宽	开始索引	结束索引
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	20	96	115
12	24	116	139
13	28	140	167
14	32	168	199

续表

比例因子波段	带宽	开始索引	结束索引
15	38	200	237
16	46	238	283
17	52	284	335
18	60	336	395
19	68	396	463
20	58	464	521
21	54	522	575

**16kHz 采样频率,短块,线数 192**

比例因子波段	带宽	开始索引	结束索引
0	4	0	3
1	4	4	7
2	4	8	11
3	6	12	17
4	8	18	25
5	10	26	35
6	12	36	47
7	14	48	61
8	18	62	79
9	24	80	103
10	30	104	133
11	40	134	173
12	18	174	191

**22. 05kHz 采样频率,长块,线数 576**

比例因子波段	带宽	开始索引	结束索引
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29

续表

比例因子波段	带宽	开始索引	结束索引
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	20	96	115
12	24	116	139
13	28	140	167
14	32	168	199
15	38	200	237
16	46	238	283
17	52	284	335
18	60	336	395
19	68	396	463
20	58	464	521
21	54	522	575

## 22. 05kHz 采样频率, 短块, 线数 192

比例因子波段	带宽	开始索引	结束索引
0	4	0	3
1	4	4	7
2	4	8	11
3	6	12	17
4	6	18	23
5	8	24	31
6	10	32	41
7	14	42	55
8	18	56	73
9	26	74	99
10	32	100	131
11	42	132	173
12	18	174	191

**24kHz 采样频率, 长块, 线数 576**

比例因子波段	带宽	开始索引	结束索引
0	6	0	5
1	6	6	11
2	6	12	17
3	6	18	23
4	6	24	29
5	6	30	35
6	8	36	43
7	10	44	53
8	12	54	65
9	14	66	79
10	16	80	95
11	18	96	113
12	22	114	135
13	26	136	161
14	32	162	193
15	38	194	231
16	46	232	277
17	54	278	331
18	62	332	393
19	70	394	463
20	76	464	539
21	36	540	575

**24kHz 采样频率, 短块, 线数 192**

比例因子波段	带宽	开始索引	结束索引
0	4	0	4
1	4	4	8
2	4	8	12
3	6	12	18
4	8	18	26
5	10	26	36

续表

比例因子波段	带宽	开始索引	结束索引
6	12	36	48
7	14	48	62
8	18	62	80
9	24	80	104
10	32	104	136
11	44	136	180
12	12	180	192

## 附录 III-C 编码过程

(本附录非此推荐|国际标准的必备部分)

### III-C.1 低采样频率扩展

在此附录中说明本推荐|国际标准的编码器与 ISO/IEC 11172-3 编码器的区别。

#### III-C.1.1 低采样频率,层 I

与 ISO/IEC 11172-3 中描述的编码器的唯一区别在于格式和听觉心理模型。编码的子波段信息以帧的形式传输,每帧由槽组成。在层 I,一个槽由 32 位组成。一帧的槽数取决于采样率和比特率。每帧包含 384 个原始输入样本的信息。

Fs/kHz	帧大小/ms
24	16
22.05	17.415..
16	24

一帧的槽数可用以下公式计算:

$$\text{每帧槽数}(N) = \text{bitrate} * 12 / \text{Fs}$$

用于低采样频率的听觉心理模型须更改。见附录 III-D.1。

#### III-C.1.2 低采样频率,层 II

与 ISO/IEC 11172-3 中描述的编码器的区别在于格式、可能的量化及听觉心理模型。编码的子波段信息以帧的形式传输,每帧由槽组成。在层 II,一个槽包含 8 个位。一帧的槽数依赖于采样频率和比特率。每帧含有 384 个原始输入信号样本的信息。

Fs/kHz	帧大小/ms
24	48
22.05	52.245..
16	72

一帧的槽数可用以下公式计算:

$$\text{每帧槽数}(N) = \text{bitrate} * 144 / \text{Fs}$$

本文件必须使用 ISO/IEC 11172-3 表 B.1“每个子波段的可能的量化,层 II”,而非表 B.2“层 II 位分配表,每个子波段的可能的量化”。

对于低采样频率,听觉心理模型需要修改。见附录 III-D.1。

### III-C. 1.3 低采样频率,层 III

与 11172-3 层 II 描述的编码过程的区别在于比例因子波段表的改变,由于帧设计的变化而省略的一些附带信息,以及听觉心理模型上某些表的改进。11172-3 中描述的基本步骤除计算比例因子选择信息外全都适用。

## III-C. 2 多通道扩展

在本附录的这部分中,描述了两个适合多通道编码器的例子。一个用于层 I 和 II ,另一个用于层 III。例子对 5+1 通道配置有效。(例如:左、中、右、左环绕、右环绕的低频增强通道),对于多语种扩展也与同层的多通道扩展一样。

### III-C. 2.1 多通道扩展,层 I, II

#### III-C. 2.1.1 滤波器组

使用的滤波器组与 ISO/IEC 11172-3 的一样,如: 对所有层的 32 波段多项滤波器组,在层 II ,后跟一个子波段信号的 MDCT。子波段滤波器被用于全部五个通道。

#### III-C. 2.1.2 比例因子的计算

比例因子以及层 II 的比例因子的选择信息的计算与 ISO/IEC 11172-3 中的方法完全一样。

#### III-C. 2.1.3 听觉心理模型

ISO/IEC 11172-3 中描述的两个听觉心理模型在此也适用。对于全部五个通道,要计算所有子波段的 signal-to-mask 比率。

#### III-C. 2.1.4 预矫正

预矫正(或预量化)是为了防止在解码器进行逆矩阵时出现音频通道的未掩盖和预期之外的噪声。这种噪音的出现是由于解码器进行逆矩阵的信号是与编码器的矩阵处理用的不同的多通道扩展信号。在解码器中只能得到量化的样本。在编码器的矩阵运算之前,对这些样本进行预量化,可以避免可听见的人为噪音。可采用以下步骤:

对于每个子波段组:

步骤 1: 传输通道切换过程: 多通道扩展信号 T2, T3, T4 的选择及相关的 tc\_allocation;

如果 tc\_allocation[sbgr] 等于 1 或 7:

步骤 2: 根据掩盖阈值计算对 T2 和 T3 编码和解码;

步骤 3: 用预矫正的 T2 和 T3 进行矩阵运算以获得 Lo;

步骤 4: 预矫正的中央信号的计算,它将在对 Lo 编码解码后在解码器端得到。

步骤 5: 用预矫正后的中央信号和 T4 作矩阵运算以获得 Ro。

如果 tc\_allocation[sbgr] 等于 2 或 6:

步骤 2: 根据掩盖阈值计算对 T2 和 T3 编码和解码;

步骤 3：用预矫正的 T2 和 T3 进行矩阵运算以获得 Ro；

步骤 4：预矫正的中央信号的计算，它将在对 Ro 编码解码后在解码器端得到；

步骤 5：用预矫正后的中央信号和 T4 作矩阵运算以获得 Lo。

如果 tc\_allocation[sbgr] 等于 0,3,4 或 5

步骤 2：根据掩盖阈值计算对 T2,T3,T4 编码和解码；

步骤 3：用预矫正的 T2,T3,T4 作矩阵运算以获得兼容对(Lo,Ro)

如果中央信号在某一子波段组中占有控制地位，建议在附加的传输通道中只使用不含中央信号的 tc\_allocations。

### III-C. 2. 1. 5 矩阵运算

首先，在计算兼容立体声信号时，所有的信号都必须衰减，以避免过载。衰减因子依选取的矩阵过程而定。

过程 0,2:  $1/(1+\sqrt{2})$

过程 1:  $1/(1.5+0.5 * \sqrt{2})$

过程 3: 1

其次，中央、左环绕和右环绕信号在计算兼容立体声前必须衰减。衰减因子为：

过程 0,2: C,Ls,Rs  $1/\sqrt{2}$

过程 1: C  $1/\sqrt{2}$

Ls,Rs 0.5

过程 3: C,Ls,Rs 1

衰减后的信号为  $C^W, Ls^W, Rs^W$ 。

接着，兼容信号计算如下：

过程 0,1:  $Lo = L^W + C^W + LS^W$

$Ro = R^W + C^W + RS^W$

过程 2:  $Lo = L^W + C^W - jS^W$

$Ro = R^W + C^W + jS^W$

过程 3:  $Lo = L^W$

$Ro = R^W$

其中  $jS^W$  是从  $LS^W$  和  $RS^W$  经过计算单音成分，带宽限制在 100~7000Hz，半杜比 B 型编码，90°相移后得到的。

### III-C. 2. 1. 6 动态传输通道切换

在逆矩阵处理时为避免可听见的人为噪音，有必要选择正确的传输通道分配。这适用于矩阵过程 0,1 和 2。一个简单但有效的方法是选择那些有最低的子波段组比例因子的传输通道 T2,T3,T4。在由一个以上子波段构成的子波段组中，首先必须为每个信号确定在子波段组的所有子波段中最大的比例因子。然后，三个有最低的最大比例因子的信号（最高比例因子索引）被分配到传输通道 T2,T3,T4。如果传输通道分配对于所有子波段组相同或几乎相同，tc\_sbgr\_select 位可设为 0，此时只有一个 tc\_allocation 必须为全部子波段组传送。

### III-C. 2. 1. 7 动态交叉会话

根据人耳的双耳模型,两个或多通道立体声信号的成分可以被确定为一个大的扩展,这与对应的立体声表现的空间感无关。立体声无关信号成分未被掩盖,而在另一方面,它们对声源的定位不起作用。因此,并非所有的传输通道,只是那些含有立体声无关成分的部分,在所有时间内才必须传输。此时,多通道立体声信号(L,C,R,LS 或 RS)的任何通道可以被其它通道代替。这可以在子波段组中进行,其中最多有 12 个组,或甚至在整个音频通道内进行。在解码端,此通道或其一部分必须通过任一表现通道重建,或通过几个表现通道,而不影响立体声印象。

层 I 和 II 中使用的动态交叉会话方法是基于强立体声编码的概念,在 ISO/IEC 11172-3 附录 G 中有所描述,但是允许不同的通道之间有更大的灵活性,且具有频率段上的更高的分辨率。动态交叉会话可以在一定的比特率条件下提高音频质量,或在一定音质水平下减少多通道音频信号的比特率。此方法会产生一些可以忽略的解码器的复杂性,但不影响编码器和解码器延迟。

动态交叉会话是基于已知的听觉心理效果的。一方面,此方法使用像强立体声编码那样的效应,即在高频部分定位主要靠瞬时包络而非音频信号的瞬时精确结构;另一方面,动态交叉会话基于以下事实,只有音频信号的瞬时包络的快速变化对于定位是重要的。然而,越是稳定的部分对于定位作用就越小。这意味着在一定的频谱区间内,对于一定的时间间隔,交叉会话是允许的。那些信号必须用编码器中的信号分析来区分。可设成“单音”,只用一个通道传输。信号可以在子波段组的基础上区分。多通道扩展部分最多有三个传输通道可以被替换。

只有相应的比例因子和 scfsi 在那些会被动态交叉会话在解码器中替换的通道中传输,而非位分配和子波段样本。结果是,在基本通道 Lo/Ro 中可获得全部的所谓“Gestalt”信息的影像,在扩展通道内只有相关的立体声信息被传输。

III-3.5.2.10 节的动态交叉会话表的  $T_{xy}$  表示 tc\_allocation 表(也在 III-3.5.2.10 节给出)中指出的表现通道的子波段样本应被加在一起,如 ISO/IEC 11172-3 附录 G 所描述。位分配和子波段样本在传输通道  $T_x$  中传输。对应于  $T_x$  和  $T_y$  的表现通道的比例因子和 scfsi 必须在  $T_x$  和  $T_y$  中传输。它允许为两个通道传输电平控制信息,以重建相应于  $T_x$  和  $T_y$  的两个表现通道的瞬时坡度。动态交叉会话表中的表项允许非常灵活地使用强立体声编码。

### III-C. 2. 1. 8 自适应的多通道预测

自适应的多通道预测用于减少通道间的冗余。(所有的计算在帧基础上对一帧进行。)

在使用多通道预测时,传输通道 T2..T4 的信号从 MPEG 1 兼容部分比特流(Lo, Ro)的信号预测得到。代替在子波段组内的实际信号,传输的是预测误差、预测系数和延时补偿。

可能的预测方程是:

$$C^w(n) = \sum_{k=0}^2 pred\_coef\_C0[sbgr,k] * T0(n - delay\_comp - k) + \sum_{k=0}^2 pred\_coef\_C1[sbgr,k] * T1(n - delay\_comp - k)$$

$$\begin{aligned}
S^{w^A}(n) &= \sum_{k=0}^2 \text{pred\_coef\_} S0[\text{sbgr}, k] * T0(n - \text{delay\_comp} - k) + \sum_{k=0}^2 \text{pred\_coef\_} \\
&\quad S1[\text{sbgr}, k] * T1(n - \text{delay\_comp} - k) \\
LS^{w^A}(n) &= \sum_{k=0}^2 \text{pred\_coef\_} Ls[\text{sbgr}, k] * T0(n - \text{delay\_comp} - k) \\
RS^{w^A}(n) &= \sum_{k=0}^2 \text{pred\_coef\_} Rs[\text{sbgr}, k] * T1(n - \text{delay\_comp} - k) \\
L^{w^A}(n) &= \sum_{k=0}^2 \text{pred\_coef\_} L[\text{sbgr}, k] * T0(n - \text{delay\_comp} - k) \\
R^{w^A}(n) &= \sum_{k=0}^2 \text{pred\_coef\_} R[\text{sbgr}, k] * T1(n - \text{delay\_comp} - k)
\end{aligned}$$

其中  $S$  是两个传输通道的信号和, 用于某些动态交叉会话的情况。取代  $L^w, R^w, C^w, LS^w, RS^w, S^w$ , 下列预测误差被传递。

$$\begin{aligned}
\epsilon_l &= L^w - L^{w^A} \\
\epsilon_r &= R^w - R^{w^A} \\
\epsilon_c &= C^w - C^{w^A} \\
\epsilon_{ls} &= LS^w - LS^{w^A} \\
\epsilon_{rs} &= RS^w - RS^{w^A} \\
\epsilon_s &= S^w - S^{w^A}
\end{aligned}$$

通过计算预测器系数  $\text{pred\_coef}[\text{sbgr}, px, pci]$ , 使预测误差信号的指数最小, 以产生优化的预测增益。预测增益是原始信号与相应的预测误差信号的能量的比率, 以 dB 为单位。下面给出详细说明。

在比较了实际预测增益和要求的预测器系数编码的附带信息量之后, 决定在每帧对哪个子波段组和哪个信号使用  $L^w, R^w, C^w, LS^w, RS^w$  和  $S^w$  的预测。需要 8 位来对应于 1.34dB 预测增益的预测器系数编码。

如果预测误差信号取代原始信号被传输, 则用于位分配过程的 SMR 值由于计算预测增益而减少。为了提供位分配所必须的 SCFSI 信息, 必须计算传输的预测误差信号的“预备”版本。

为了避免不同量化误差的累积, 建议在计算“最后”的预测误差信号之前量化和反量化信号  $Lo, Ro$  和预测器系数。因此, 预测误差信号在编码器和解码器中一样。

传输的信号  $T0, T1, T2, T3, T4$  的编码通常用“分配”“SCFSI”、“比例因子”和“采样”来实现。

### 帧的编码

{

- 子波段滤波;
- 矩阵运算;
- 比例因子计算;
- 传输模式计算(SCFSI);

- 用听觉心理模型计算 SMR 的值;
- 传输通道分配;
- 动态交叉会话;
- 延迟补偿, 预测器系数和预测增益的计算;
- 预测器选择信号(PREDSI)的计算;
- 修改 SMR 值的计算;
- 预测器系数的量化;
- 预备预测误差信号的计算;
- 比例因子计算;
- 传输模式计算(SCFSI);
- 位分配(用修改的 SMR 值);
- 子波段样本量化;
- 子波段样本反量化;
- 最终预测误差信号的计算(使用反量化的子波段样本);
- 比例因子计算;
- 传输模式计算(SCFSI);
- 子波段样本量化;
- 比特流格式化;

}

预测器系数、预测增益和预测器选择信息的计算。

下述的类 C 语言的描述是预测计算的简单的例子, 针对传输通道 T2,T3,T4 包含各自的 C,LS,RS, 动态交叉会话不起作用, 只有零阶预测的情况, 无延迟补偿。此过程的输出由系数 coef\_0,coef\_1,coef\_2,coef\_3 和相应的预测器选择信息 predsi[0..3]组成, 在此例中,coef\_0..coef\_3 的含义为:

```
pred_coef_C0=coef_0;
pred_coef_C1=coef_1;
pred_coef_LS=coef_2;
pred_coef_RS=coef_3;
```

下述过程在其它情况下类似。

```
for(sbgr=0; sbgr<12; sbgr++){
    /* calculation of variances and correlation functions
       using short-term estimates */
    st1=st2=sc=sls=srs=0;
    ct1c=ct2c=ct1t2=ct1ls=ct2rs=0;
    numsb=((sbgr==11)? sblimits: sbgr_min[sbgr+1])-sbgr_min[sbgr];
    for (sb=sbgr_min[sbgr];sb<sbgr_min[sbgr]+numsb; sb++)
        for (gr =0; gr<3; gr++)
```

```

for (i=0; i<12; i++){
    st1+=sqr(sb_sample[0][gr][i][sb]);
    st2+=sqr(sb_sample[1][gr][i][sb]);
    sc +=sqr(sb_sample[2][gr][i][sb]);
    sls+=sqr(sb_sample[3][gr][i][sb]);
    srs+=sqr(sb_sample[4][gr][i][sb]);
    ct1c+=sb_sample[0][gr][i][sb] * sb_sample[2][gr][i][sb];
    ct2c+=sb_sample[1][gr][i][sb] * sb_sample[2][gr][i][sb];
    ct1t2+=sb_sample[0][gr][i][sb] * sb_sample[1][gr][i]
        [sb];
    ct1ls+=sb_sample[0][gr][i][sb] * sb_sample[3][gr][i][sb];
    ct2rs+=sb_sample[1][gr][i][sb] * sb_sample[4][gr][i]
        [sb];
}

st1=sqrt(st1/(3 * 12 * numsb));
st2=sqrt(st2/(3 * 12 * numsb));
sc=sqrt(sc/(3 * 12 * numsb));
sls=sqrt(sls/(3 * 12 * numsb));
srs=sqrt(srs/(3 * 12 * numsb));
st1=(st1>MIN_S)? st1: MIN_S; /* to avoid division by 0 */
st2=(st2>MIN_S)? st2: MIN_S;
sc=(sc>MIN_S)? sc: MIN_S;
sls=(sls>MIN_S)? sls: MIN_S;
srs=(srs>MIN_S)? srs: MIN_S;
ct1c=ct1c/(st1 * sc);
ct2c=ct2c/(st2 * sc);
ct1t2=ct1t2/(st1 * st2);
ct1ls=ct1ls/(st1 * sls);
ct2rs=ct2rs/(st2 * srs);
/* calculation of predictor coefficients */
coef_0=sc/st1 * ct1c;
coef_1=sc/st2 * ct2c;
coef_2X0=sc/st1 * (ct1c-ct2c * ct1t2)/(1-sqr(ct1t2));
coef_2X1=sc/st2 * (ct2c-ct1c * ct1t2)/(1-sqr(ct1t2));
coef_2=sls/st1 * ct1ls;
coef_3=srs/st2 * ct2rs;
/* calculation of prediction gains */

```

```

/* problem: if sbgr contains more than one subband the */
/* prediction gain can be different in the subbands!!! */
gain_0=10 * log(1/(1-sqr(ct1c)));
gain_1=10 * log(1/(1-sqr(ct2c)));
gain_2=10 * log(1/(1-sqr(ct1ls)));
gain_3=10 * log(1/(1-sqr(ct2rs)));
temp=sqr(sc)-2 * (coef_x0 * ct1c * st1 * sc)-2 * (coef_x1 * ct2c
* st2 * sc)+2 * (coef_x0 * coef_x1 * ct1t2 * st1 * st2)+sqr
(coef_x0 * st1)+sqr(coef_x1 * st2);
gain_01=10 * log (sqr(sc)/temp);
/* calculation of predictor select information */
maxgain=0;
maxmode=0;
if (gain_0 -SI_COEF/numsb>maxgain){
    maxgain=gain_0-SI_COEF/numsb;
    maxmode=1;
}
if (gain_1 -SI_COEF/numsb>maxgain){
    maxgain=gain_1-SI_COEF/numsb;
    maxmode=2;
}
if (gain_01-2 * SI_COEF/numsb>maxgain) {
    maxgain=gain_01-2 * SI_COEF/numsb;
    maxmode=3;
}
switch (maxmode) {
case 0:
    temp_pred_gain[0]=0;
    predsi[0]=0;
    predsi[1]=0;
    break;
case 1:
    temp_pred_gain[0]=gain_0;
    predsi[0]=1;
    predsi[1]=0;
    pred_coef[sbgr][0]=coef_0;
    break;
case 2:

```

```

        temp_pred_gain[0]=gain_1;
        predsi[0]=0;
        predsi[1]=1;
        pred_coef[sbgr][1]=coef_1;
        break;

    case 3:
        temp_pred_gain[0]=gain_01;
        predsi[0]=1;
        predsi[1]=1;
        pred_coef[sbgr][0]=coef_x0;
        pred_coef[sbgr][1]=coef_x1;
        break;
    }

    if (gain_2>SI_COEF/numsb) {
        temp_pred_gain[1]=gain_2;
        predsi[2]=1;
        pred_coef[sbgr][2]=coef_2;
    }
    else{
        temp_pred_gain[1]=0;
        predsi[2]=0;
    }

    if (gain_3>SI_COEF/numsb) {
        temp_pred_gain[2]=gain_3;
        predsi[3]=1;
        pred_coef[sbgr][3]=coef_3;
    }
    else{
        temp_pred_gain[2]=0;
        predsi[3]=0;
    }

    /* simplifying assumption: prediction gain is the same in */
    /* all subbands of one subband group */
    for (sb=sbgr_min[sbgr]; sb<sbgr_min[sbgr]+numsb; sb++)
        for (i=0; i<3; i++)
            pred_gain[i][sb]=temp_pred_gain[i];
    /* modification of the SMR values according to the prediction gain */

```

```

/* i.e. : SMR is reduced by the prediction gain */
for (sb = sbgr_min[sbgr]; sb < sbgr_min[sbgr] + numsb; sb++)
    for (i=0; i<3; i++)
        smr[i+2][sb] = pred_gain[i][sb];
} /* for (sbgr=0; sbgr<12; sbgr++) */

```

### III-C. 2. 1. 9 中央通道的幻象编码

如果存在位缺乏, 使用中央通道幻象编码可以用不引人注目的方法提供明显的增益。为了得到低频和高频部分, 中央信号是经过低通和高通滤波的。中央通道的高频部分衰减了 3dB 加到左和右通道上。滤波和叠加应在 PCM 值域内进行, 以避免在子波段边界产生混淆, 这是由于幻象编码在此边界以上进行。多通道比特流的中央位被设置为‘11’。只有位分配、比例因子选择信息、比例因子和低频部分的采样数据被实际传输。

### III-C. 2. 1. 10 位分配

位分配过程与 ISO/IEC 11172-3 的相似, 但是现在用于 5 个通道, 同时还有一个低频增强通道。在层 I, 因为兼容部分要求三个位分配, 而扩展部分只要求一个位分配, 因此稍有不同。一种简单的方法就是对于向后兼容部分的每三个连续的层 I 的帧用相同的位分配, 把附带信息需求的位和此部分的样本变成三倍。此后, 可以同层 II 一样处理。从全部可得的位的总数里, 必须去掉 2 位, 因为在每三个连续帧的前两个后面插入了一个位 0。这是为了在面向位的通道无更多帧时作同步用的。

### III-C. 2. 1. 11 多语种

多语种通道的编码可以和比特流中的兼容的多通道数据以相同的采样频率进行, 或者采用半采样率。在后一种情况, 可以在降低带宽的条件下获得编码效率的显著的增加。如果输入信号的带宽已受限, 如语音信号, 则此带宽限制并非真正的不利因素。

如果使用了全部的采样频率, 则可以根据 ISO/IEC 11172-3 编码, 只有一个例外: 不可能进行强立体声编码, 及最多复合七个通道。如果用半采样率, 则可以用 C. 1. 2 中所描述的低采样率扩展编码, 只有一个例外: 不可能进行强立体声编码, 最多复合七个通道, 以及帧内包含子波段样本的一半, 因此时间减半。

### III-C. 2. 1. 12 格式化

多通道/多语种比特流须根据 III-3. 5. 1 节的语法格式化。在层 I, 多通道比特流被直接插在向后兼容部分的后面。帧的剩余的数据可用于辅助数据。在层 I, 多通道比特流基本由三部分组成, 分在三个层 I 的帧内。第一部分直接从向后兼容的比特流结尾开始, 在下一个同步字前一位结束。帧的最后一位设为零。第二部分直接跟在下一帧的向后兼容部分后面, 在那帧结束前一位结束, 同样最后一位为零。第三部分直接跟在下一帧的向后兼容部分后面, 在那帧结束前结束, 剩余位用作辅助数据。

## III-C. 2 多通道扩展层 III

### III-C. 2. 2. 1 听觉心理模型

ISO/IEC 11172-3 中描述的两个听觉心理模型在此仍然适用。对所有五个通道和兼

容通道,计算全部比例因子波段的门槛值。如果用矩阵过程进行了编码(即: dematrix\_procedure<3),则为了优化系统操作,所有的通道的块类型必须一样。在至少有一个通道具备窗口开关的条件时,以上所述便可通过将 ISO/IEC 11172-3 的 C. 1. 5. 3/2 中所描述的窗口开关序列应用于所有通道来实现。

### III-C. 2. 2. 2 滤波器组

使用的滤波器组同 ISO/IEC 11172-3。即: 32 波段的多相滤波器组,后跟子波段信号的 MDCT,及减少混淆的处理(见 ISO/IEC 11172-3 的 C. 1. 5. 3/3 节)。根据听觉心理模型计算出的块类型,滤波器组用于所有五个通道。

### III-C. 2. 2. 3 段表处理

段表是引入联合立体声编码的一种通用方法,到一个通道的输出可以由另一个通道的数据经比例变换得到,段表处理应用的要求是所有的通道有相同的块类型。我们建议在除 dematrix\_procedure==3 以外的多通道信号编码中用此方法。在这种条件下,所有通过合成编码分组在一起的通道应有相同的块类型。

虽然语法允许几个段在一个块内包含不同的联合立体声方式,但有可能限制处于高频率的段使用 segment\_List。这是对此处描述的编码器的建议。

联合立体声编码的应用在联合立体声检测过程的控制下进行,以便确定通道之间的最佳的联合立体声组合。可变的 dematrix\_length 指明了自适应逆矩阵和联合立体声处理的分离点。

联合立体声检测用于所有可能的 dematrix\_length 值,从 0 到 14。Dematrix\_length 被设置成与最低的 dematrix\_length 索引相同,其中联合立体声检测在满足了虚拟的无削弱的图象映象要求的同时,显示了从立体声编码的预期增益。

联合立体声检测通过对最佳联合立体声组合的搜索来完成。用所有可实现的通道的组合(像 L+Rs,R+Rs,L+C+Ls,R+C+Rs,Ls+Rs 等),来比较模拟的联合立体声组合与原始的联合立体声组合。比较通过估算原始的和模拟联合立体声信号的短时能量进行。如果相关的能量偏差大于 0.03,则此组合不适合。与此同时,使用感知熵(PE)估计联合立体声编码降低比特率的可能性。通过组合可以提供最小的质量损失,正如短时能量指明的那样,同时,最大 PE 增益项被选择。

对于选择的联合立体声组合的传输,一个通道被用于此组合的“载体”。此运载通道包含联合立体声组合的谱信息。载体通道是从所有组合通道中选择出来的具有最高的能量的组合通道。

### III-C. 2. 2. 4 动态传输通道切换

为了避免逆矩阵过程中可听见的人为噪音,有必要选择正确的传输通道分配。有几种可能的方法:

- 整个传输通道的选择可用“seglist\_present”语法的带少量位的附带信息。对于有效的层Ⅲ比特流,编码器可通过设置 seglist\_present[] 为零选择最多两个通道进行逆矩阵运算。此情况下,对应的 tc\_present[] 可设置为 0,指示无进一步的附带信息被传输。
- 为了获得更好的逆矩阵配置的控制,传输通道的选择可以比例因子波段组为基础

在比例因子波段组上进行。这可用 dematrix\_select 语法得到。对于高于 dematrix\_length 的比例因子波段组,选择对应的段的 tc\_select 值为 7,可得到相同的效果。

选择过程可基于如下条件: 对于每个通道,通过听觉心理模型计算掩蔽能力(掩蔽阈值,xmin),用于 MPEG-1 层Ⅲ编码(见 ISO/IEC 11172-3,C. 1. 5. 3 节)。从所有的通道中,具有最强的掩蔽能力的两个通道被选通过作使用逆矩阵重建,因此无须传输。如果其中一个通道为中央通道,且计算的掩蔽阈值差超过 6dB,则只有具有最强掩蔽能力的通道被逆矩阵。

### III-C. 2. 2. 5 矩阵运算

兼容的立体声信号 Lo/Ro 由多通道信号计算如下:

$$\text{过程 } 0,1,3: \quad Lo = \alpha * (L + \beta * C + \gamma * Ls)$$

$$Ro = \alpha * (R + \beta * C + \gamma * Rs)$$

$$\text{过程 } 2: \quad Lo = \alpha * (L + \beta * C - \gamma * jS)$$

$$Ro = \alpha * (R + \beta * C + \gamma * jS)$$

其中  $jS$  由  $Ls$  和  $Rs$  通过计算单音成分而得到,带宽限制为 100~7000Hz,半杜比 B 型编码,90°相移。

在上述方程中, $\alpha$  是对所有通道的全体衰减, $\beta$  和  $\gamma$  是中央和环绕信号的衰减因子。对每个逆矩阵过程,衰减因子规定为:

dematrix_procedure	$\alpha$	$\beta$	$\gamma$
0	$1/(1+\sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$
1	$1/(1.5+0.5*\sqrt{2})$	$1/\sqrt{2}$	0.5
2	$1/(1+\sqrt{2})$	$1/\sqrt{2}$	$1/\sqrt{2}$
3	1	0	0

请注意,与层Ⅰ和层Ⅱ不同,所有多通道立体声信号 L,R,C,Ls,RS,S 的处理都没有加权过程。

### III-C. 2. 2. 6 自适应多通道预测

在层Ⅲ多通道编码中可用与层Ⅰ和层Ⅱ相同的自适应多通道预测,除了一个例外:它的预测过程作用于混合型滤波器组的输出值。

### III-C. 2. 2. 7 量化和编码

对于连续的编码,所有五个输入通道和两个兼容通道的输出数据被转换成 TC 表示。这通过去除所有滤波器组的输出通道谱的频谱部分来实现,谱部分不必传输。有两种情况,谱部分排除在传输之外:

- 在解码器中由逆矩阵重建的谱数据将被排除于 TC 的传输之外。这根据动态传输切换的结果进行。
- 在联合立体声编码条件下,只有涉及到的通道数据的运载部分在 TC 中传输。所

有其它涉及到的通道数据在解码器中由联合立体声处理过程通过段表语法重建。

汇集 TC 数据后,所有的 TC 被以同样的方式量化,如层Ⅱ立体声编码器的通道谱用 ISO/IEC 11172-3 的 C. 1. 5. 4 节描述的迭代的策略。对应通道的阈值和用听觉心理模型计算的比例因子波段用作迭代目标(即:对每个比例因子波段的最大允许偏差,xmin)。根据计算得到的其它通道的阈值水平、更复杂的编码策略会涉及改进迭代目标。

编码的 TC 中的位分配根据其在感知熵(PE)中的相对作用进行:

$$tc\_bits_{ch} = \frac{pe_{ch}}{\sum_i pe_i} \cdot total\_bits$$

其中  $tc\_bits$  表明 TC#Ch 的位分配,  $pe_i$  表示通道 i 的总感知熵,  $total\_bits$  是此颗粒中依据比特率和采样频率可能得到的总位数。有关感知熵的定义见 ISO/IEC 11172-3 的 C. 1. 5. 3/2. 1

### III-C. 2. 2. 8 多语种扩展

编码依赖于 multi\_lingual\_fs 的选择,如 ISO 11172-3 中所述或 C. 1. 3 中描述的改进。

## 附录 III -D 听觉心理模型

(此附录不是本推荐|国际标准的必要部分)

### III -D. 1 用于低采样频率的听觉心理模型 1

对于低采样频率扩展来说,听觉心理模型 1 所需的适应性改变并不大。在此重复描述听觉心理模型,并作了必要的修改。

听觉心理模型的计算要与对应的层相适应。此例对层 I 和 II 有效。此模型可以修改以适应层 III。

听觉心理模型在层 I 或 II 的应用中没有根本的区别:

层 I : 对每个具有 12 个子波段或 384 个 PCM 输入样本的块计算新的位分配。

层 II : 对三个共有 36 个子波段样本,对应  $3 * 384(1152)$  个输入 PCM 样本的块计算新的位分配。

32 个子波段的位分配基于全部子波段的 signal-to-mask-ratio 的计算。因此,确定每个子波段的最大信号电平和最小掩蔽阈值是必要的。最小的掩蔽阈值来自输入 PCM 信号的 FFT,后面跟随听觉心理模型的计算。

FFT 与子波段滤波器操作并行进行,作为对子波段滤波器组的低频谱选择性缺乏的补偿。此技术既提供了对编码音频(具有最小预回声优化窗的多相滤波器),足够的时间分辨率,也提供了足够的计算掩蔽阈值的谱分辨率。频率和电平的混叠干扰可以计算出来。这对需要一些位以去除混叠成分的解码器的子波段的最小比特率的计算是必要的。只在编码器中需要计算更高频率分辨率的附加复杂性,在解码器中无附加的延迟或复杂性。

signal-to-mask-ratio 的计算基于以下步骤:

- 步骤 1 FFT 从时域到频域转换的计算;
- 步骤 2 在每个子波段中的声压水平的确定;
- 步骤 3 安静阈值的确定(绝对阈值);
- 步骤 4 寻找音频信号中音调(更像正弦形状)的和非音调(更像噪音)的成分;
- 步骤 5 掩模的换算,获得相关的掩模;
- 步骤 6 各自掩蔽阈值的计算;
- 步骤 7 全局掩蔽阈值的确定;
- 步骤 8 每个子波段最小掩蔽阈值的确定;
- 步骤 9 signal-to-mask 比率的计算

这些步骤将作进一步讨论,假设 24kHz 的采样频率,除非另作说明。对于另两个采样频率,所有提到的频率要作相应的变化。

#### 步骤 1 谱的计算

FFT 在原则上与 ISO/IEC 11172-3 的相同,但由于采样频率不同,以毫秒为单位表示的长度不同。

FFT 的技术数据:

	层 I	层 II
变换长度 N	512 样本	1024 样本
窗宽 假如 $F_s = 24\text{kHz}$	21.33ms	42.67ms
窗宽 假如 $F_s = 22.05\text{kHz}$	23.22ms	46.44ms
窗宽 假如 $F_s = 16\text{kHz}$	32ms	64ms
频率分辨率	$F_s/512$	$F_s/1024$

Hann 窗,  $h(i)$ :

$$h(i) = \sqrt{8/3} * 0.5 * \{1 - \cos[2\pi(i)/N]\} \quad 0 <= i <= N-1$$

能量密度谱 X(k):

$$X(k) = 10 * \log_{10} \left| \frac{1}{N} \sum_{l=0}^{N-1} h(l) * s(l) * e^{(-jkl * 2\pi/N)} \right|^2 \text{dB} \quad k = 0 \dots N/2,$$

其中  $s(l)$  是输入信号

在最大值对应 96dB 时实现参考水平为 96dB SPL(声压级)的规范。

## 步骤 2 声压级的确定

在子波段 n 中的声压级  $L_{sb}$  计算如下:

$$L_{sb}(n) = \text{MAX}[X(k), 20 * \log(\text{scf}_{max}(n) * 32768) - 10] \text{ dB}$$

X(k) 在子波段 n 内

其中 X(k) 是 FFT 的索引为 k 的谱线的声压级,是在相应的子波段 n 的频率范围中的最大幅度下。表达式  $\text{scf}_{max}(n)$  是层 I 的比例因子,它在层 II 是一帧内子波段 n 的三个比例因子的最大值。“-10dB”项修正了峰和 RMS 电平间的差。对每个子波段 n 都计算声压级  $L_{sb}(n)$ 。

下面的两种可替换的计算  $L_{sb}(n)$  的方法,提供了获得更佳编码器性能的潜力,但是此技术尚未进行正规的音频质量测试。

子波段 n 内的交替的声压级  $L_{sb}$  计算如下:

$$L_{sb}(n) = \text{MAX}[X_{spl}(n), 20 * \lg(\text{scf}_{max}(n) * 32768) - 10] \text{ dB}$$

with

$$X_{spl}(n) = 10 * \log_{10} \left( \sum_k 10^{X(k)/10} \right) \text{dB}$$

k 在子波段 n 内。

其中  $X_{spl}(n)$  是子波段 n 对应的交替的声压级。

## 步骤 3 考虑安静阈值

安静阈值  $LT_q(k)$ ,或称为绝对阈值,可在表“频率,临界比特率和绝对阈值”(层 I 的表 II-D.1a, II-D.1b, II-D.1C; 层 II 的表 II-D.1d, II-D.1e, II-D.1f)获得。这些表依赖于输入 PCM 信号的采样率。在频域可得到每个样本的值,其中计算了掩蔽阈值。

## 步骤 4 寻找音调和非音调成分

掩蔽成分的音调对掩蔽阈值有影响。由于这个原因,区分音调和非音调成分是值得的。为计算全局的掩蔽阈值,有必要从 FFT 谱中获取音调和非音调成分。

这一步从局部最大值的确定开始,然后,在一个临界波段带宽内抽取音调成分(正弦),且计算非音调成分的强度。临界波段的边界在表“临界波段边界”中给出(表 II-D. 2a, II-D. 2b, II-D. 2c; 层 I 的表 II-D. 2d, II-D. 2e, II-D. 2f)。

临界波段的带宽在低频时随着只有 0.1kHz 带宽的中心频率变化,在高频时随着 4kHz 带宽的中心频率变化。由听觉心理专家得知,耳朵在低频区比在高频区有更好的频率分辨率。要确定一个局部最大值是否为音调成分,可以检查此局部最大值周围的频率范围 df。频率范围如下给出:

采样率 16kHz

$df = 62.5\text{Hz}$	$0\text{kHz} < f \leq 3.0\text{kHz}$
$df = 93.75\text{Hz}$	$3.0\text{kHz} < f \leq 6.0\text{kHz}$
$df = 187.5\text{Hz}$	$6.0\text{kHz} < f \leq 7.5\text{kHz}$

采样率 22.05kHz

$df = 86.133\text{Hz}$	$0\text{kHz} < f \leq 2.756\text{kHz}$
$df = 129.199\text{Hz}$	$2.756\text{kHz} < f \leq 5.512\text{kHz}$
$df = 258.398\text{Hz}$	$5.512\text{kHz} < f \leq 10.336\text{kHz}$

采样率 24kHz

$df = 93.750\text{Hz}$	$0\text{kHz} < f \leq 3.0\text{kHz}$
$df = 140.63\text{Hz}$	$3.0\text{kHz} < f \leq 6.0\text{kHz}$
$df = 281.25\text{Hz}$	$6.0\text{kHz} < f \leq 11.250\text{kHz}$

为了列出音调或非音调的谱线  $X(k)$ , 执行下列操作:

(1) 标出局部最大值

谱线  $X(k)$  被标为局部最大值, 当

$X(k) > X(k-1)$  且  $X(k) \geq X(k+1)$

(2) 列出音调成分, 计算声压级

局部最大值被列入音调成分, 当

$X(k) - X(k+j) \geq 7\text{dB}$

其中 j 依据下述选择:

层 I,  $F_s = 16\text{kHz}$ :

$j = -2, +2$	对于 $2 < k < 96$
$j = -3, -2, +2, +3$	对于 $96 \leq k < 192$
$j = -6, \dots, -2, +2, \dots, +6$	对于 $192 \leq k < 250$

层 II,  $F_s = 16\text{kHz}$ :

$j = -4, +4$	对于 $4 < k < 192$
$j = -6, \dots, -2, +2, \dots, +6$	对于 $192 \leq k < 384$
$j = -12, \dots, -2, +2, \dots, +12$	对于 $384 \leq k < 500$

层 I ,Fs=22.05,24kHz:

j=-2,+2	对于 2<k<64
j=-3,-2,+2,+3	对于 64<=k<128
j=-6,...,-2,+2,...,+6	对于 128<=k<250

层 II ,Fs=22.05,24kHz:

j=-4,+4	对于 4<k<128
j=-6,...,-2,+2,...,+6	对于 128<=k<256
j=-12,...,-2,+2,...,+12	对于 256<=k<500

如果 X(k)是一个音调成分,列出下列参数:

谱线索引号 k;

声压级  $X_{tm}(k)=10 * \log_{10}\{10^{\frac{X(k-1)}{10}} + 10^{\frac{X(k)}{10}} + 10^{\frac{X(k+1)}{10}}\}$ ,以 dB 为单位;

音调标志。

接着,被检查的频率范围内的所有频率线设置为 $-\infty$ dB

### (3) 列出非音调成分,计算能量

从剩余的谱线内计算非音调(噪音)成分。要计算谱线 X(k)的非音调成分,用表“临界波段边界”(层 I 的表 III-D. 2a, III-D. 2b, III-D. 2c; 层 II 的表 III-D. 2d, III-D. 2e, III-D. 2f) 确定临界波段 z(k)。对于采样率 16kHz,用 21 个临界波段,对于采样率 22.05kHz 和 24kHz 用 23 个临界波段。在每个临界波段内,谱线(音调成分被置零后剩余的)的能量被求和,构成对应临界波段的新的非音调成分  $X_{nm}(k)$  的声压级。

下列参数被列出:

最靠近临界波段几何平均的谱线的索引 K;

声压级  $X_{nm}(k)$  以 dB 为单位;

非音调标志。

## 步骤 5 音调和非音调掩蔽成分换算

换算过程是用于减少计算全局掩蔽阈值的掩模的数目。

(1) 音调成分  $X_{tm}(k)$  或非音调成分  $X_{nm}(k)$  被用于计算掩蔽阈值,仅当

$$X_{tm}(k) >= LT_q(k) \text{ 或 } X_{nm}(k) >= LT_q(k)$$

在此表达式中,  $LT_q(k)$  是频率索引 k 的绝对阈值(或安静阈值)。这些值在层 I 的表 III-D. 1a, III-D. 1b, III-D. 1c; 层 II 的表 III-D. 1d, III-D. 1e, III-D. 1f 中给出。

(2) 在距离小于 0.5 Bark 内换算两个或更多的音调成分:保持成分的最高能量,从音调成分表中除去较小的成分。进行此操作时,使用一个宽度为 0.5 Bark 的临界波段内的滑动窗口。接着,索引 j 用来指示与组合的换算表的相关的音调或非音调掩蔽成分。

## 步骤 6 各自掩蔽阈值的计算

在原始的 N/2 频率域由 k 索引的样本中,仅有由 i 索引的样本的子集,被用于全局掩蔽阈值的计算。用到的样本在层 I 的表 III-D. 1a, III-D. 1b, III-D. 1c; 层 II 的表 III-D. 1d, III-D. 1e, III-D. 1f 中示出。

层 I : 对于对应于被初始的六个子波段覆盖的频率区域的频率线,不用子采样。对于

对应于其后的六个子波段的频率区域,要考虑每个频率区域的第二个谱线。最后,对于其后的18个子波段,要考虑每个频率区域的第四个谱线,(见层I的表III-D.1a, III-D.1b, III-D.1c)

层II: 在对应于被最初的三个子波段覆盖的频率区间的频率线,不用子采样。在对应于被其后的三个子波段覆盖的频率区间,考虑每个频率区域的第二个谱线。在对应于其后的六个子波段的频率区间,考虑每个频率区域的第四个谱线。最后,对于其后的18个子波段,考虑每个频率区域的第8个谱线(见层I的表III-D.1d, III-D.1e, III-D.1f)

样本数n在子采样频率域内依赖于层。对于层I,n等于108,对于层II,n等于132。

每个音调和非音调成分被赋值为以i索引的,且与原始谱线X(k)代表的频率最近的值。此索引i在层I的表III-D.1a, III-D.1b, III-D.1c,层II的表III-D.1d, III-D.1e, III-D.1f给出。

两个音调和非音调成分的各自掩蔽阈值由以下表达式给出:

$$LT_{tm}[z(j), z(i)] = X_{tm}[z(j)] + av_{tm}[z(j)] + vf[z(j), z(i)] dB$$

$$LT_{nm}[z(j), z(i)] = X_{nm}[z(j)] + av_{nm}[z(j)] + vf[z(j), z(i)] dB$$

在此公式中,LT<sub>tm</sub>和LT<sub>nm</sub>是处于z Bark临界波段率的各自掩蔽阈值,掩蔽成分在临界波段率掩模z<sub>m</sub> Bark处。值以dB为单位,可正,可负。项X<sub>tm</sub>[z(j)]是对应临界波段率z(j)的索引号为j的掩蔽成分的声压级。项av是掩蔽索引,vf是掩蔽成分X<sub>tm</sub>[z(j)]的掩蔽函数。掩蔽索引av与音调和非音调掩模(av<sub>tm</sub>和av<sub>nm</sub>)不同。

对于音调掩模,由以下公式给出:

$$av_{tm} = -1.525 - 0.275 * z(j) - 4.5 dB$$

对于非音调掩模

$$av_{nm} = -1.525 - 0.175 * z(j) - 0.5 dB$$

掩蔽函数vf由不同的较低和较高的斜率确定特性,取决于以Bark为单位的到掩模的距离dz=z(i)-z(j)。在此表达式中,i是谱线的索引,在其上计算掩蔽函数,j是相应的掩模索引。临界波段率z(j)和z(i)可在层I的表III-D.1a, III-D.1b, III-D.1c,层II的表III-D.1d, III-D.1e, III-D.1f中找到。掩蔽函数与音调和非音调掩模相同,如下:

$$vf = 17 * (dz + 1) - (0.4 * X[z(j)] + 6) dB \quad \text{对于 } -3 \leq dz < -1 \text{ Bark}$$

$$vf = (0.4 * X[z(j)] + 6) * dz dB \quad \text{对于 } -1 \leq dz < 0 \text{ Bark}$$

$$vf = -17 * dz dB \quad \text{对于 } 0 \leq dz < 1 \text{ Bark}$$

$$vf = -(dz - 1) * (17 - 0.15 * X[z(j)]) - 17 dB \quad \text{对于 } 1 \leq dz < 8 \text{ Bark}$$

在这些表达式中X[z(j)]是第j个掩蔽成分的声压级,以dB为单位。由于实现的复杂性,掩蔽将不再考虑,如果dz<-3Bark或dz>=8Bark(LT<sub>tm</sub>和LT<sub>nm</sub>被设成-∞ dB,在此范围以外)。

## 步骤7 全局掩蔽阈值LT<sub>g</sub>的计算

第i个频率样本的全局掩蔽阈值LT<sub>g</sub>(i)从每个j音调和非音调掩模的各自掩蔽阈值的较高或较低斜率及安静阈值LT<sub>q</sub>(i)得到。也可在层I的表III-D.1a, III-D.1b, III-D.1c;层II的表III-D.1d, III-D.1e, III-D.1f中得到。全局阈值通过对相应的各自掩蔽阈值和安静阈值的能量求和得到。

$$LT_g(i) = 10 \log_{10} \left( 10^{LT_q(i)/10} + \sum_{j=1}^m 10^{LT_{tm}(z(j), z(i))/10} + \sum_{j=1}^n 10^{LT_{nm}(z(j), z(i))/10} \right)$$

音调掩模的总数为 m, 非音调掩模的总数为 n, 对给定的 i,j 的范围可减小, 以正好包围距离 i 为 -8 到 +3 Bark 的掩蔽成分。此范围外  $LT_{tm}$  和  $LT_{nm}$  是  $-\infty$  dB

#### 步骤 8 最小掩蔽阈值的确定

子波段 n 的最小掩蔽阈值由下式给出:

$$LT_{\min}(n) = \text{MIN}(LT_g(i)) \text{ dB}$$

$f(i)$  在子波段 n 中

其中  $f(i)$  是第 i 个频率样本的频率。 $f(i)$  在层 I 的表 II-D. 1a, II-D. 1b, II-D. 1c; 层 II 的表 II-D. 1d, II-D. 1e, II-D. 1f 中。对每个子波段计算最小掩蔽水平  $LT_{\min}(n)$ 。

#### 步骤 9 signal-to-mask-ratio 的计算

对每个子波段 n 计算 signal-to-mask-ratio

$$SMR_{sb}(n) = L_{sb}(n) - LT_{\min}(n) \text{ dB}$$

表III-D.1a 频率,临界波段速率和绝对阈值

此表在16kHz采样率对层I有效

索引号 i	频率 Hz	临界波段率 z	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z	绝对阈值 dB
1	31.25	0.309	58.23	55	1937.50	12.898	0.02
2	62.50	0.617	33.44	56	2000.00	13.104	-0.25
3	93.75	0.925	24.17	57	2062.50	13.302	-0.54
4	125.00	1.232	19.20	58	2125.00	13.493	-0.83
5	156.25	1.538	16.05	59	2187.50	13.678	-1.12
6	187.50	1.842	13.87	60	2250.00	13.855	-1.43
7	218.75	2.145	12.26	61	2312.50	14.027	-1.73
8	250.00	2.445	11.01	62	2375.00	14.193	-2.04
9	281.25	2.742	10.01	63	2437.50	14.354	-2.34
10	312.50	3.037	9.20	64	2500.00	14.509	-2.64
11	343.75	3.329	8.52	65	2562.50	14.660	-2.93
12	375.00	3.618	7.94	66	2625.00	14.807	-3.22
13	406.25	3.903	7.44	67	2687.50	14.949	-3.49
14	437.50	4.185	7.00	68	2750.00	15.087	-3.74
15	468.75	4.463	6.62	69	2812.50	15.221	-3.98
16	500.00	4.736	6.28	70	2875.00	15.351	-4.20
17	531.25	5.006	5.97	71	2937.50	15.478	-4.40
18	562.50	5.272	5.70	72	3000.00	15.602	-4.57
19	593.75	5.533	5.44	73	3125.00	15.841	-4.82
20	625.00	5.789	5.21	74	3250.00	16.069	-4.96
21	656.25	6.041	5.00	75	3375.00	16.287	-4.98
22	687.50	6.289	4.80	76	3500.00	16.496	-4.90
23	718.75	6.532	4.62	77	3625.00	16.697	-4.70
24	750.00	6.770	4.45	78	3750.00	16.891	-4.39
25	781.25	7.004	4.29	79	3875.00	17.078	-3.99
26	812.50	7.233	4.14	80	4000.00	17.259	-3.51
27	843.75	7.457	4.00	81	4125.00	17.434	-2.99
28	875.00	7.677	3.86	82	4250.00	17.605	-2.45
29	906.25	7.892	3.73	83	4375.00	17.770	-1.90
30	937.50	8.103	3.61	84	4500.00	17.932	-1.37
31	968.75	8.309	3.49	85	4625.00	18.089	-0.86
32	1000.00	8.511	3.37	86	4750.00	18.242	-0.39
33	1031.25	8.708	3.26	87	4875.00	18.392	0.03
34	1062.50	8.901	3.15	88	5000.00	18.539	0.40
35	1093.75	9.090	3.04	89	5125.00	18.682	0.72
36	1125.00	9.275	2.93	90	5250.00	18.823	1.00
37	1156.25	9.456	2.83	91	5375.00	18.960	1.24
38	1187.50	9.632	2.73	92	5500.00	19.095	1.44
39	1218.75	9.805	2.63	93	5625.00	19.226	1.62
40	1250.00	9.974	2.53	94	5750.00	19.356	1.78
41	1281.25	10.139	2.42	95	5875.00	19.482	1.92
42	1312.50	10.301	2.32	96	6000.00	19.606	2.05
43	1343.75	10.459	2.22	97	6125.00	19.728	2.18
44	1375.00	10.614	2.12	98	6250.00	19.847	2.30
45	1406.25	10.765	2.02	99	6375.00	19.964	2.42
46	1437.50	10.913	1.92	100	6500.00	20.079	2.55
47	1468.75	11.058	1.81	101	6625.00	20.191	2.69
48	1500.00	11.199	1.71	102	6750.00	20.300	2.82
49	1562.50	11.474	1.49	103	6875.00	20.408	2.97
50	1625.00	11.736	1.27	104	7000.00	20.513	3.13
51	1687.50	11.988	1.04	105	7125.00	20.616	3.29
52	1750.00	12.230	0.80	106	7250.00	20.717	3.46
53	1812.50	12.461	0.55	107	7375.00	20.815	3.65
54	1875.00	12.684	0.29	108	7500.00	20.912	3.84

表III-D.1b 频率,临界波段速率和绝对阈值

此表在 22.05kHz 采样率对层 I 有效

索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB
1	43.07	0.425	45.05	55	2670.12	14.909	-3.41
2	86.13	0.850	25.87	56	2756.25	15.100	-3.77
3	129.20	1.273	18.70	57	2842.38	15.284	-4.09
4	172.27	1.694	14.85	58	2928.52	15.460	-4.37
5	215.33	2.112	12.41	59	3014.65	15.631	-4.60
6	258.40	2.525	10.72	60	3100.78	15.796	-4.78
7	301.46	2.934	9.47	61	3186.91	15.955	-4.91
8	344.53	3.337	8.50	62	3273.05	16.110	-4.97
9	387.60	3.733	7.73	63	3359.18	16.260	-4.98
10	430.66	4.124	7.10	64	3445.31	16.406	-4.96
11	473.73	4.507	6.56	65	3531.45	16.547	-4.88
12	516.80	4.882	6.11	66	3617.58	16.685	-4.74
13	559.86	5.249	5.72	67	3703.71	16.820	-4.54
14	602.93	5.608	5.37	68	3789.84	16.951	-4.30
15	646.00	5.959	5.07	69	3875.98	17.079	-4.02
16	689.06	6.301	4.79	70	3962.11	17.205	-3.71
17	732.13	6.634	4.55	71	4048.24	17.327	-3.37
18	775.20	6.959	4.32	72	4134.38	17.447	-3.00
19	818.26	7.274	4.11	73	4306.64	17.680	-2.25
20	861.33	7.581	3.92	74	4478.91	17.905	-1.50
21	904.39	7.879	3.74	75	4651.17	18.121	-0.81
22	947.46	8.169	3.57	76	4823.44	18.331	-0.18
23	990.53	8.450	3.40	77	4995.70	18.534	0.35
24	1033.59	8.723	3.25	78	5167.97	18.731	0.79
25	1076.66	8.987	3.10	79	5340.23	18.922	1.15
26	1119.73	9.244	2.95	80	5512.50	19.108	1.44
27	1162.79	9.493	2.81	81	5684.77	19.289	1.68
28	1205.86	9.734	2.67	82	5857.03	19.464	1.89
29	1248.93	9.968	2.53	83	6029.30	19.635	2.07
30	1291.99	10.195	2.39	84	6201.56	19.801	2.24
31	1335.06	10.416	2.25	85	6373.83	19.963	2.41
32	1378.13	10.629	2.11	86	6546.09	20.120	2.59
33	1421.19	10.836	1.97	87	6718.36	20.273	2.78
34	1464.26	11.037	1.83	88	6890.63	20.421	2.98
35	1507.32	11.232	1.68	89	7062.89	20.565	3.19
36	1550.39	11.421	1.53	90	7235.16	20.705	3.43
37	1593.46	11.605	1.38	91	7407.42	20.840	3.68
38	1636.52	11.783	1.23	92	7579.69	20.972	3.95
39	1679.59	11.957	1.07	93	7751.95	21.099	4.24
40	1722.66	12.125	0.90	94	7924.22	21.222	4.56
41	1765.72	12.289	0.74	95	8096.48	21.342	4.89
42	1808.79	12.448	0.56	96	8268.75	21.457	5.25
43	1851.86	12.603	0.39	97	8441.02	21.569	5.64
44	1894.92	12.753	0.21	98	8613.28	21.677	6.05
45	1937.99	12.900	0.02	99	8785.55	21.781	6.48
46	1981.05	13.042	-0.17	100	8957.81	21.882	6.95
47	2024.12	13.181	-0.36	101	9130.08	21.980	7.44
48	2067.19	13.317	-0.56	102	9302.34	22.074	7.96
49	2153.32	13.578	-0.96	103	9474.61	22.165	8.52
50	2239.45	13.826	-1.38	104	9646.88	22.253	9.10
51	2325.59	14.062	-1.79	105	9819.14	22.338	9.72
52	2411.72	14.288	-2.21	106	9991.41	22.420	10.37
53	2479.85	14.504	-2.63	107	10163.67	22.499	11.06
54	2583.98	14.711	-3.03	108	10335.94	22.576	11.79

表III-D. 1c 频率, 临界波段速率和绝对阈值

此表在 24kHz 采样率对层 I 有效

索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB
1	46.88	0.463	42.10	55	2906.25	15.415	-4.30
2	93.75	0.925	21.17	56	3000.00	15.602	-4.57
3	140.63	1.385	17.47	57	3093.75	15.783	-4.77
4	187.50	1.842	13.87	58	3187.50	15.956	-4.91
5	234.38	2.295	11.60	59	3281.25	16.124	-4.98
6	281.25	2.742	10.01	60	3375.00	16.287	-4.98
7	328.13	3.184	8.84	61	3468.75	16.445	-4.94
8	375.00	3.618	7.94	62	3562.50	16.598	-4.84
9	421.88	4.045	7.22	63	3656.25	16.746	-4.66
10	468.75	4.463	6.62	64	3750.00	16.891	-4.43
11	515.63	4.872	6.12	65	3843.75	17.032	-4.15
12	562.50	5.272	5.70	66	3937.50	17.169	-3.82
13	609.38	5.661	5.33	67	4031.25	17.303	-3.45
14	656.25	6.041	5.00	68	4125.00	17.434	-3.06
15	703.13	6.411	4.71	69	4218.75	17.563	-2.66
16	750.00	6.770	4.45	70	4312.50	17.688	-2.24
17	796.88	7.119	4.21	71	4406.25	17.811	-1.83
18	843.75	7.457	4.00	72	4500.00	17.932	-1.43
19	890.63	7.785	3.79	73	4687.50	18.166	-0.68
20	937.50	8.103	3.61	74	4875.00	18.392	-0.02
21	984.38	8.410	3.43	75	5062.50	18.611	0.52
22	1031.25	8.708	3.26	76	5250.00	18.823	0.97
23	1078.13	8.996	3.09	77	5437.50	19.028	1.32
24	1125.00	9.275	2.93	78	5625.00	19.226	1.60
25	1171.88	9.544	2.78	79	5812.50	19.419	1.83
26	1218.75	9.805	2.63	80	6000.00	19.606	2.03
27	1265.63	10.057	2.47	81	6187.50	19.788	2.22
28	1312.50	10.301	2.32	82	6375.00	19.964	2.41
29	1359.38	10.537	2.17	83	6562.50	20.135	2.60
30	1406.25	10.765	2.02	84	6750.00	20.300	2.81
31	1453.13	10.986	1.86	85	6937.50	20.461	3.03
32	1500.00	11.199	1.71	86	7125.00	20.616	3.27
33	1546.88	11.406	1.55	87	7312.50	20.766	3.53
34	1593.75	11.606	1.38	88	7500.00	20.912	3.82
35	1640.63	11.800	1.21	89	7687.50	21.052	4.12
36	1687.50	11.988	1.04	90	7875.00	21.188	4.46
37	1734.38	12.170	0.86	91	8062.50	21.318	4.82
38	1781.25	12.347	0.67	92	8250.00	21.445	5.20
39	1828.13	12.518	0.49	93	8437.50	21.567	5.62
40	1875.00	12.684	0.29	94	8625.00	21.684	6.07
41	1921.88	12.845	0.09	95	8812.50	21.797	6.54
42	1968.75	13.002	-0.11	96	9000.00	21.906	7.06
43	2015.63	13.154	-0.32	97	9187.50	22.012	7.60
44	2062.50	13.302	-0.54	98	9375.00	22.113	8.18
45	2109.38	13.446	-0.75	99	9562.50	22.210	8.80
46	2156.25	13.586	-0.97	100	9750.00	22.304	9.46
47	2203.13	13.723	-1.20	101	9937.50	22.395	10.15
48	2250.00	13.855	-1.43	102	10 125.00	22.482	10.89
49	2343.75	14.111	-1.88	103	10 312.50	22.566	11.67
50	2437.50	14.354	-2.34	104	10 500.00	22.646	12.50
51	2531.25	14.585	-2.79	105	10 687.50	22.724	13.37
52	2625.00	14.807	-3.22	106	10 875.00	22.799	14.29
53	2718.75	15.018	-3.62	107	11 062.50	22.871	15.26
54	2812.50	15.221	-3.98	108	11 250.00	22.941	16.28

表III-D. 1d 频率, 临界波段速率和绝对阈值

此表在 16kHz 采样率对层 I 有效

索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB
1	15.63	0.154	68.00	67	1343.75	10.459	2.22
2	31.25	0.309	58.23	68	1375.00	10.614	2.12
3	46.88	0.463	42.10	69	1406.25	10.765	2.02
4	62.50	0.617	33.44	70	1437.50	10.913	1.92
5	78.13	0.771	27.97	71	1468.75	11.058	1.81
6	93.75	0.925	24.17	72	1500.00	11.199	1.71
7	109.38	1.079	21.36	73	1562.50	11.474	1.49
8	125.00	1.232	19.20	74	1625.00	11.736	1.27
9	140.63	1.385	17.47	75	1687.50	11.988	1.04
10	156.25	1.538	16.05	76	1750.00	12.230	0.80
11	171.88	1.690	14.87	77	1812.50	12.461	0.55
12	187.50	1.842	13.87	78	1875.00	12.684	0.29
13	203.13	1.994	13.01	79	1937.50	12.898	0.02
14	218.75	2.145	12.26	80	2000.00	13.104	-0.25
15	234.38	2.295	11.60	81	2062.50	13.302	-0.54
16	250.00	2.445	11.01	82	2125.00	13.493	-0.83
17	265.63	2.594	10.49	83	2187.50	13.678	-1.12
18	281.25	2.742	10.01	84	2250.00	13.855	-1.43
19	296.88	2.890	9.59	85	2312.50	14.027	-1.73
20	312.50	3.037	9.20	86	2375.00	14.193	-2.04
21	328.13	3.184	8.84	87	2437.50	14.354	-2.34
22	343.75	3.329	8.52	88	2500.00	14.509	-2.64
23	359.38	3.474	8.22	89	2562.50	14.660	-2.93
24	375.00	3.618	7.94	90	2625.00	14.807	-3.22
25	390.63	3.761	7.68	91	2687.50	14.949	-3.49
26	406.25	3.903	7.44	92	2750.00	15.087	-3.74
27	421.88	4.045	7.22	93	2812.50	15.221	-3.98
28	437.50	4.185	7.00	94	2875.00	15.351	-4.20
29	453.13	4.324	6.81	95	2937.50	15.478	-4.40
30	468.75	4.463	6.62	96	3000.00	15.602	-4.57
31	484.38	4.600	6.44	97	3125.00	15.841	-4.82
32	500.00	4.736	6.28	98	3250.00	16.069	-4.96
33	515.63	4.872	6.12	99	3375.00	16.287	-4.98
34	531.25	5.006	5.97	100	3500.00	16.496	-4.88
35	546.88	5.139	5.83	101	3625.00	16.697	-4.66
36	562.50	5.272	5.70	102	3750.00	16.891	-4.34
37	578.13	5.403	5.57	103	3875.00	17.078	-3.93
38	593.75	5.533	5.44	104	4000.00	17.259	-3.45
39	609.38	5.661	5.33	105	4125.00	17.434	-2.93
40	625.00	5.789	5.21	106	4250.00	17.605	-2.38
41	640.63	5.916	5.10	107	4375.00	17.770	-1.83
42	656.25	6.041	5.00	108	4500.00	17.932	-1.30
43	671.88	6.166	4.90	109	4625.00	18.089	-0.80
44	687.50	6.289	4.80	110	4750.00	18.242	-0.34
45	703.13	6.411	4.71	111	4875.00	18.392	0.07
46	718.75	6.532	4.62	112	5000.00	18.539	0.44
47	734.38	6.651	4.53	113	5125.00	18.682	0.76
48	750.00	6.770	4.45	114	5250.00	18.823	1.03
49	761.25	7.004	4.29	115	5375.00	18.960	1.26
50	812.50	7.233	4.14	116	5500.00	19.095	1.47
51	843.75	7.457	4.00	117	5625.00	19.226	1.64
52	875.00	7.677	3.86	118	5750.00	19.356	1.80
53	906.25	7.892	3.73	119	5875.00	19.482	1.94
54	937.50	8.103	3.61	120	6000.00	19.606	2.07
55	968.75	8.309	3.49	121	6125.00	19.728	2.19
56	1000.00	8.511	3.37	122	6250.00	19.847	2.32
57	1031.25	8.708	3.26	123	6375.00	19.964	2.44
58	1062.50	8.901	3.15	124	6500.00	20.079	2.57
59	1093.75	9.090	3.04	125	6625.00	20.191	2.70
60	1125.00	9.275	2.93	126	6750.00	20.300	2.84
61	1156.25	9.456	2.83	127	6875.00	20.408	2.99
62	1187.50	9.632	2.73	128	7000.00	20.513	3.15
63	1218.75	9.805	2.63	129	7125.00	20.616	3.31
64	1250.00	9.974	2.53	130	7250.00	20.717	3.49
65	1281.25	10.139	2.42	131	7375.00	20.815	3.67
66	1312.50	10.301	2.32	132	7500.00	20.912	3.87

表 III-D. 1e 频率、临界波段速率和绝对阈值

此表在 22.05kHz 采样率对层 I 有效

索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB
1	21.53	0.213	68.00	67	1851.86	12.603	0.39
2	43.07	0.425	45.05	68	1894.92	12.753	0.21
3	64.60	0.638	32.57	69	1937.99	12.900	0.02
4	86.13	0.850	25.87	70	1981.05	13.042	-0.17
5	107.67	1.062	21.63	71	2024.12	13.181	-0.36
6	129.20	1.273	18.70	72	2067.19	13.317	-0.56
7	150.73	1.484	16.52	73	2153.32	13.578	-0.96
8	172.27	1.694	14.85	74	2239.45	13.826	-1.38
9	193.80	1.903	13.51	75	2325.59	14.062	-1.79
10	215.33	2.112	12.41	76	2411.72	14.288	-2.21
11	236.87	2.319	11.50	77	2497.85	14.504	-2.63
12	258.40	2.525	10.72	78	2583.98	14.711	-3.03
13	279.93	2.730	10.05	79	2670.12	14.909	-3.41
14	301.46	2.934	9.47	80	2756.25	15.100	-3.77
15	323.00	3.136	8.96	81	2842.38	15.284	-4.09
16	344.53	3.337	8.50	82	2928.52	15.460	-4.37
17	366.06	3.536	8.10	83	3014.65	15.631	-4.60
18	387.60	3.733	7.73	84	3100.78	15.796	-4.78
19	409.13	3.929	7.40	85	3186.91	15.955	-4.91
20	430.66	4.124	7.10	86	3273.05	16.110	-4.97
21	452.20	4.316	6.82	87	3359.18	16.260	-4.98
22	473.73	4.507	6.56	88	3445.31	16.406	-4.94
23	495.26	4.695	6.33	89	3531.45	16.547	-4.85
24	516.80	4.882	6.11	90	3617.58	16.685	-4.69
25	538.33	5.067	5.91	91	3703.71	16.820	-4.49
26	559.86	5.249	5.72	92	3789.84	16.951	-4.24
27	581.40	5.430	5.54	93	3875.98	17.079	-3.95
28	602.93	5.608	5.37	94	3962.11	17.205	-3.63
29	624.46	5.785	5.22	95	4048.24	17.327	-3.28
30	646.00	5.959	5.07	96	4134.38	17.447	-2.91
31	667.53	6.131	4.93	97	4306.64	17.680	-2.16
32	689.06	6.301	4.79	98	4478.91	17.905	-1.41
33	710.60	6.469	4.67	99	4651.17	18.121	-0.72
34	732.13	6.634	4.55	100	4823.44	18.331	-0.11
35	753.66	6.798	4.43	101	4995.70	18.534	0.41
36	775.20	6.959	4.32	102	5167.97	18.731	0.84
37	796.73	7.118	4.21	103	5340.23	18.922	1.19
38	818.26	7.274	4.11	104	5512.50	19.108	1.48
39	839.79	7.429	4.01	105	5684.77	19.289	1.71
40	861.33	7.581	3.92	106	5857.03	19.464	1.91
41	882.86	7.731	3.83	107	6029.30	19.635	2.09
42	904.39	7.879	3.74	108	6201.56	19.801	2.26
43	925.93	8.025	3.65	109	6373.83	19.963	2.43
44	947.46	8.169	3.57	110	6546.09	20.120	2.61
45	968.99	8.310	3.48	111	6718.36	20.273	2.80
46	990.53	8.450	3.40	112	6890.63	20.421	3.00
47	1012.06	8.587	3.33	113	7062.89	20.565	3.22
48	1033.59	8.723	3.25	114	7235.16	20.705	3.46
49	1076.66	8.987	3.10	115	7407.42	20.840	3.71
50	1119.73	9.244	2.95	116	7579.69	20.972	3.98
51	1162.79	9.493	2.81	117	7751.95	21.099	4.28
52	1205.86	9.734	2.67	118	7924.22	21.222	4.60
53	1248.93	9.968	2.53	119	8096.48	21.342	4.94
54	1291.99	10.195	2.39	120	8268.75	21.457	5.30
55	1335.06	10.416	2.25	121	8441.02	21.569	5.69
56	1378.13	10.629	2.11	122	8613.28	21.677	6.10
57	1421.19	10.836	1.97	123	8785.55	21.781	6.54
58	1464.26	11.037	1.83	124	8957.81	21.882	7.01
59	1507.32	11.232	1.68	125	9130.08	21.980	7.50
60	1550.39	11.421	1.53	126	9302.34	22.074	8.03
61	1593.46	11.605	1.38	127	9474.61	22.165	8.59
62	1636.52	11.783	1.23	128	9646.88	22.253	9.18
63	1679.59	11.957	1.07	129	9819.14	22.338	9.80
64	1722.66	12.125	0.90	130	9991.41	22.420	10.46
65	1765.72	12.289	0.74	131	10163.67	22.499	11.15
66	1808.79	12.448	0.56	132	10335.94	22.576	11.88

表 II-D. 1f 频率、临界波段速率和绝对阈值

此表在 24kHz 采样率对层 I 有效

索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB	索引号 i	频率 Hz	临界波段率 z Bark	绝对阈值 dB
1	23.44	0.232	68.00	67	2015.63	13.154	-0.32
2	46.88	0.463	42.10	68	2062.50	13.302	-0.54
3	70.31	0.694	30.43	69	2109.38	13.446	-0.75
4	93.75	0.925	24.17	70	2156.25	13.586	-0.97
5	117.19	1.156	20.22	71	2203.13	13.723	-1.20
6	140.63	1.385	17.47	72	2250.00	13.855	-1.43
7	164.06	1.614	15.44	73	2343.75	14.111	-1.88
8	187.50	1.842	13.87	74	2437.50	14.354	-2.34
9	210.94	2.069	12.62	75	2531.25	14.585	-2.79
10	234.38	2.295	11.60	76	2625.00	14.807	-3.22
11	257.81	2.519	10.74	77	2718.75	15.018	-3.62
12	281.25	2.742	10.01	78	2812.50	15.221	-3.98
13	304.69	2.964	9.39	79	2906.25	15.415	-4.30
14	328.13	3.184	8.84	80	3000.00	15.602	-4.57
15	351.56	3.402	8.37	81	3093.75	15.783	-4.77
16	375.00	3.618	7.94	82	3187.50	15.956	-4.91
17	398.44	3.832	7.56	83	3281.25	16.124	-4.98
18	421.88	4.045	7.22	84	3375.00	16.287	-4.98
19	445.31	4.255	6.90	85	3468.75	16.445	-4.92
20	468.75	4.463	6.62	86	3562.50	16.598	-4.80
21	492.19	4.668	6.36	87	3656.25	16.746	-4.61
22	515.63	4.872	6.12	88	3750.00	16.891	-4.36
23	539.06	5.073	5.90	89	3843.75	17.032	-4.07
24	562.50	5.272	5.70	90	3937.50	17.169	-3.73
25	585.94	5.468	5.50	91	4031.25	17.303	-3.36
26	609.38	5.661	5.33	92	4125.00	17.434	-2.96
27	632.81	5.853	5.16	93	4218.75	17.563	-2.55
28	656.25	6.041	5.00	94	4312.50	17.688	-2.14
29	679.69	6.227	4.85	95	4406.25	17.811	-1.73
30	703.13	6.411	4.71	96	4500.00	17.932	-1.33
31	726.56	6.592	4.58	97	4687.50	18.166	-0.59
32	750.00	6.770	4.45	98	4875.00	18.392	0.05
33	773.44	6.946	4.33	99	5062.50	18.611	0.58
34	796.88	7.119	4.21	100	5250.00	18.823	1.01
35	820.31	7.289	4.10	101	5437.50	19.028	1.36
36	843.75	7.457	4.00	102	5625.00	19.226	1.63
37	867.19	7.622	3.89	103	5812.50	19.419	1.86
38	890.63	7.785	3.79	104	6000.00	19.606	2.06
39	914.06	7.945	3.70	105	6187.50	19.788	2.25
40	937.50	8.103	3.61	106	6375.00	19.964	2.43
41	960.94	8.258	3.51	107	6562.50	20.135	2.63
42	984.38	8.410	3.43	108	6750.00	20.300	2.83
43	1007.81	8.560	3.34	109	6937.50	20.461	3.06
44	1031.25	8.708	3.26	110	7125.00	20.616	3.30
45	1054.69	8.853	3.17	111	7312.50	20.766	3.57
46	1078.13	8.996	3.09	112	7500.00	20.912	3.85
47	1101.56	9.137	3.01	113	7687.50	21.052	4.16
48	1125.00	9.275	2.93	114	7875.00	21.188	4.50
49	1171.88	9.544	2.78	115	8062.50	21.318	4.86
50	1218.75	9.805	2.63	116	8250.00	21.445	5.25
51	1265.63	10.057	2.47	117	8437.50	21.567	5.67
52	1312.50	10.301	2.32	118	8625.00	21.684	6.12
53	1359.38	10.537	2.17	119	8812.50	21.797	6.61
54	1406.25	10.765	2.02	120	9000.00	21.906	7.12
55	1453.13	10.986	1.86	121	9187.50	22.012	7.67
56	1500.00	11.199	1.71	122	9375.00	22.113	8.26
57	1546.88	11.406	1.55	123	9562.50	22.210	8.88
58	1593.75	11.606	1.38	124	9750.00	22.304	9.54
59	1640.63	11.800	1.21	125	9937.50	22.395	10.24
60	1687.50	11.988	1.04	126	10125.00	22.482	10.98
61	1734.38	12.170	0.86	127	10312.50	22.566	11.77
62	1781.25	12.347	0.67	128	10500.00	22.646	12.60
63	1828.13	12.518	0.49	129	10687.50	22.724	13.48
64	1875.00	12.684	0.29	130	10875.00	22.799	14.41
65	1921.88	12.845	0.09	131	11062.50	22.871	15.38
66	1968.75	13.002	-0.11	132	11250.00	22.941	16.41

## III -D. 2 临界波段边界的表

**表 III-D. 2a 临界波段边界**

此表在 16kHz 采样率对层 I 有效

这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{\text{Bark}}$
0	3	93.75	0.925
1	7	218.75	2.145
2	10	312.50	3.037
3	13	406.25	3.903
4	17	531.25	5.006
5	21	656.25	6.041
6	25	781.25	7.004
7	30	937.50	8.103
8	35	1093.75	9.090
9	40	1250.00	9.974
10	47	1468.75	11.058
11	51	1687.50	11.988
12	55	1937.50	12.898
13	61	2312.50	14.027
14	67	2687.50	14.949
15	74	3250.00	16.069
16	79	3875.00	17.078
17	84	4500.00	17.932
18	91	5375.00	18.960
19	99	6375.00	19.964
20	108	7500.00	20.912

**表 III-D. 2b 临界波段边界**

此表在 22.05kHz 采样率对层 I 有效

这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{\text{Bark}}$
0	2	86.13	0.850
1	5	215.33	2.112
2	7	301.46	2.934
3	10	430.66	4.124
4	12	516.80	4.882
5	15	646.00	5.959
6	18	775.20	6.959
7	21	904.39	7.879
8	25	1076.66	8.987
9	29	1248.93	9.968
10	34	1464.26	11.037
11	39	1679.59	11.957
12	46	1981.05	13.042
13	51	2325.59	14.062
14	55	2670.12	14.909
15	61	3186.91	15.955
16	68	3789.84	16.951
17	74	4478.91	17.905
18	79	5340.23	18.922
19	85	6373.83	19.963
20	92	7579.69	20.972
21	101	9130.08	21.980
22	108	10 335.94	22.576

**表Ⅲ-D. 2c 临界波段边界**

此表在 24kHz 采样率对层 I 有效

这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{\text{Bark}}$
0	2	93.75	0.925
1	7	187.50	1.842
2	7	328.13	3.184
3	9	421.88	4.045
4	11	515.63	4.872
5	14	656.25	6.041
6	17	796.88	7.119
7	20	937.50	8.103
8	23	1078.13	8.996
9	27	1265.63	10.057
10	31	1453.13	10.986
11	36	1687.50	11.988
12	42	1968.75	13.002
13	49	2343.75	14.111
14	53	2718.75	15.018
15	58	3187.50	15.956
16	65	3843.75	17.032
17	72	4500.00	17.932
18	77	5437.50	19.028
19	82	6375.50	19.964
20	89	7687.50	21.052
21	97	9187.50	22.012
22	108	11250.00	22.941

**表Ⅲ-D. 2d 临界波段边界**

此表在 16kHz 采样率对层 I 有效

这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{\text{Bark}}$
0	6	93.75	0.925
1	13	203.13	1.994
2	20	312.50	3.037
3	27	421.88	4.045
4	34	531.25	5.006
5	42	656.25	6.041
6	49	781.25	7.004
7	54	937.50	8.103
8	59	1093.75	9.090
9	64	1250.00	9.974
10	71	1468.75	11.058
11	75	1687.50	11.988
12	79	1937.50	12.898
13	85	2312.50	14.027
14	91	2687.50	14.949
15	98	3250.00	16.069
16	103	3875.00	17.078
17	108	4500.00	17.932
18	115	5375.00	18.960
19	123	6375.00	19.964
20	132	7500.00	20.912

**表Ⅲ-D. 2e 临界波段边界**

此表在 22.05kHz 采样率对层 I 有效  
这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{Bark}$
0	5	107.67	1.062
1	9	193.80	1.903
2	14	301.46	2.934
3	19	409.13	3.929
4	25	538.33	5.067
5	30	646.00	5.959
6	36	775.20	6.959
7	43	925.93	8.025
8	49	1076.66	8.987
9	53	1248.93	9.968
10	58	1464.26	11.037
11	63	1679.59	11.957
12	70	1981.05	13.042
13	75	2325.59	14.062
14	79	2670.12	14.909
15	85	3186.91	15.955
16	92	3789.84	16.951
17	98	4478.91	17.905
18	103	5340.23	18.922
19	109	6373.83	19.963
20	116	7579.69	20.972
21	125	9130.08	21.980
22	132	10335.94	22.576

**表Ⅲ-D. 2f 临界波段边界**

此表在 24kHz 采样率对层 I 有效  
这些频率代表每个临界波段的顶端

序号	表 F&CB 索引	频率 Hz	$\frac{z}{Bark}$
0	4	93.75	0.925
1	9	210.94	2.069
2	13	304.69	2.964
3	18	421.88	4.045
4	23	539.06	5.073
5	28	656.25	6.041
6	33	773.44	6.946
7	39	914.06	7.945
8	46	1078.13	8.996
9	51	1265.63	10.057
10	55	1453.13	10.986
11	60	1687.50	11.988
12	66	1968.75	13.002
13	73	2343.75	14.111
14	77	2718.75	15.018
15	82	3187.50	15.956
16	89	3843.75	17.032
17	96	4500.00	17.932
18	101	5437.50	19.028
19	106	6375.00	19.964
20	113	7687.50	21.052
21	121	9187.50	22.012
22	132	11250.00	22.941

### III -D. 3 针对较低采样频率的听觉心理模型 2

针对较低采样频率的听觉心理模型 2 与 ISO/IEC 11172-3 中描述的听觉心理模型 2 相同, 只除了一些例外。下面的表用来代替表 III -C. 7. a... III -C. 8. e, 与层 III 一同使用:

表 III -D. 3a Sampling frequency = 24kHz 长块

序号	FFT-lines	minval	qthr	norm	bval
0	1	24.5	4.532	0.970	0.000
1	1	24.5	4.532	0.755	0.469
2	1	24.5	4.532	0.738	0.937
3	1	24.5	0.904	0.730	1.406
4	1	24.5	0.904	0.724	1.875
5	1	20	0.090	0.723	2.344
6	1	20	0.090	0.723	2.812
7	1	20	0.029	0.723	3.281
8	1	20	0.029	0.718	3.750
9	1	20	0.009	0.690	4.199
10	1	20	0.009	0.660	4.625
11	1	18	0.009	0.641	5.047
12	1	18	0.009	0.600	5.437
13	1	18	0.009	0.584	5.828
14	1	12	0.009	0.531	6.187
15	1	12	0.009	0.537	6.522
16	2	6	0.018	0.857	7.174
17	2	6	0.018	0.858	7.800
18	2	3	0.018	0.853	8.402
19	2	3	0.018	0.824	8.966
20	2	3	0.018	0.778	9.483
21	2	3	0.018	0.740	9.966
22	2	0	0.018	0.709	10.426
23	2	0	0.018	0.676	10.866
24	2	0	0.018	0.632	11.279
25	2	0	0.018	0.592	11.669
26	2	0	0.018	0.553	12.042
27	2	0	0.018	0.510	12.386
28	2	0	0.018	0.513	12.721
29	3	0	0.027	0.608	13.115
30	3	0	0.027	0.673	13.561
31	3	0	0.027	0.636	13.983
32	3	0	0.027	0.586	14.371
33	3	0	0.027	0.571	14.741
34	4	0	0.036	0.616	15.140
35	4	0	0.036	0.640	15.562
36	4	0	0.036	0.597	15.962
37	4	0	0.036	0.538	16.324
38	4	0	0.036	0.512	16.665
39	5	0	0.045	0.528	17.020

续表

序号	FFT-lines	minval	qthr	序号 rm	bval
40	5	0	0.045	0.516	17.373
41	5	0	0.045	0.493	17.708
42	6	0	0.054	0.499	18.045
43	7	0	0.063	0.525	18.398
44	7	0	0.063	0.541	18.762
45	8	0	0.072	0.528	19.120
46	8	0	0.072	0.510	19.466
47	8	0	0.072	0.506	19.807
48	10	0	0.180	0.525	20.159
49	10	0	0.180	0.536	20.522
50	10	0	0.180	0.518	20.873
51	13	0	0.372	0.501	21.214
52	13	0	0.372	0.496	21.553
53	14	0	0.400	0.497	21.892
54	18	0	1.628	0.495	22.231
55	18	0	1.628	0.494	22.569
56	20	0	1.808	0.497	22.909
57	25	0	22.607	0.494	23.248
58	25	0	22.607	0.487	23.583
59	35	0	31.650	0.483	23.915
60	67	0	605.867	0.482	24.246
61	67	0	605.867	0.524	24.576

表III-D. 3b Sampling-frequency=22.05kHz 长块

序号	FFT-lines	minval	qthr	序号 rm	bval
0	1	24.5	4.532	0.951	0.000
1	1	24.5	4.532	0.700	0.431
2	1	24.5	4.532	0.681	0.861
3	1	24.5	0.904	0.675	1.292
4	1	24.5	0.904	0.667	1.723
5	1	20	0.090	0.665	2.153
6	1	20	0.090	0.664	2.584
7	1	20	0.029	0.664	3.015
8	1	20	0.029	0.664	3.445
9	1	20	0.029	0.655	3.876
10	1	20	0.009	0.616	4.279
11	1	20	0.009	0.597	4.670
12	1	18	0.009	0.578	5.057
13	1	18	0.009	0.541	5.415
14	1	18	0.009	0.575	5.774
15	2	12	0.018	0.856	6.422
16	2	6	0.018	0.846	7.026
17	2	6	0.018	0.840	7.609

续表

序号	FFT-lines	minval	qthr	序号 rm	bval
18	2	3	0.018	0.822	8.168
19	2	3	0.018	0.800	8.710
20	2	3	0.018	0.753	9.207
21	2	3	0.018	0.704	9.662
22	2	0	0.018	0.674	10.099
23	2	0	0.018	0.640	10.515
24	2	0	0.018	0.609	10.917
25	2	0	0.018	0.566	11.293
26	2	0	0.018	0.535	11.652
27	2	0	0.018	0.531	11.997
28	3	0	0.027	0.615	12.394
29	3	0	0.027	0.686	12.850
30	3	0	0.027	0.650	13.277
31	3	0	0.027	0.611	13.681
32	3	0	0.027	0.567	14.062
33	3	0	0.027	0.520	14.411
34	3	0	0.027	0.513	14.751
35	4	0	0.036	0.557	15.119
36	4	0	0.036	0.584	15.508
37	4	0	0.036	0.570	15.883
38	5	0	0.045	0.579	16.263
39	5	0	0.045	0.585	16.654
40	5	0	0.045	0.548	17.020
41	6	0	0.054	0.536	17.374
42	6	0	0.054	0.550	17.744
43	7	0	0.063	0.532	18.104
44	7	0	0.063	0.504	18.447
45	7	0	0.063	0.496	18.781
46	9	0	0.081	0.516	19.130
47	9	0	0.081	0.527	19.487
48	9	0	0.081	0.516	19.838
49	10	0	0.180	0.497	20.179
50	10	0	0.180	0.489	20.510
51	11	0	0.198	0.502	20.852
52	14	0	0.400	0.502	21.196
53	14	0	0.400	0.491	21.531
54	15	0	0.429	0.497	21.870
55	20	0	1.808	0.504	22.214
56	20	0	1.808	0.504	22.558
57	21	0	1.899	0.495	22.898
58	27	0	24.415	0.486	23.232
59	27	0	24.415	0.484	23.564
60	36	0	32.554	0.483	23.897
61	73	0	660.124	0.475	24.229
62	18	0	162.770	0.515	24.542

表III-D.3c Sampling frequency=16kHz 长块

序号	FFT-lines	minval	qthr	序号 rm	bval
0	2	24.5	9.064	0.997	0.312
1	2	24.5	9.064	0.893	0.937
2	2	24.5	1.808	0.881	1.562
3	2	20	0.181	0.873	2.187
4	2	20	0.181	0.872	2.812
5	2	20	0.057	0.871	3.437
6	2	20	0.018	0.860	4.045
7	2	20	0.018	0.839	4.625
8	2	18	0.018	0.812	5.173
9	2	18	0.018	0.784	5.698
10	2	12	0.018	0.741	6.184
11	2	12	0.018	0.697	6.634
12	2	6	0.018	0.674	7.070
13	2	6	0.018	0.651	7.492
14	2	6	0.018	0.633	7.905
15	2	3	0.018	0.611	8.305
16	2	3	0.018	0.589	8.695
17	2	3	0.018	0.575	9.064
18	3	3	0.027	0.654	9.483
19	3	3	0.027	0.724	9.966
20	3	0	0.027	0.701	10.425
21	3	0	0.027	0.673	10.866
22	3	0	0.027	0.631	11.279
23	3	0	0.027	0.592	11.669
24	3	0	0.027	0.553	12.042
25	3	0	0.027	0.510	12.386
26	3	0	0.027	0.505	12.721
27	4	0	0.036	0.562	13.091
28	4	0	0.036	0.598	13.488
29	4	0	0.036	0.589	13.873
30	5	0	0.045	0.607	14.268
31	5	0	0.045	0.620	14.679
32	5	0	0.045	0.580	15.067
33	5	0	0.045	0.532	15.424
34	5	0	0.045	0.517	15.771
35	6	0	0.054	0.517	16.120
36	6	0	0.054	0.509	16.466
37	6	0	0.054	0.506	16.807
38	8	0	0.072	0.522	17.158
39	8	0	0.072	0.531	17.518

续表

序号	FFT-lines	minval	qthr	norm	bval
40	8	0	0.072	0.519	17.869
41	10	0	0.090	0.512	18.215
42	10	0	0.090	0.509	18.562
43	10	0	0.090	0.497	18.902
44	12	0	0.108	0.494	19.239
45	12	0	0.108	0.501	19.579
46	13	0	0.117	0.507	19.925
47	14	0	0.252	0.502	20.269
48	14	0	0.252	0.493	20.606
49	16	0	0.289	0.497	20.944
50	20	0	0.572	0.506	21.288
51	20	0	0.572	0.510	21.635
52	23	0	0.658	0.504	21.979
53	27	0	2.441	0.496	22.319
54	27	0	2.441	0.493	22.656
55	32	0	2.894	0.490	22.993
56	37	0	33.458	0.483	23.326
57	37	0	33.458	0.458	23.656
58	12	0	10.851	0.500	23.937

表III-D.3d Sampling frequency=24kHz 短块

序号	FFT-lines	qthr	norm	SNR/dB	bval
0	1	4.532	0.970	-8.240	0.000
1	1	0.904	0.755	-8.240	1.875
2	1	0.029	0.738	-8.240	3.750
3	1	0.009	0.730	-8.240	5.437
4	1	0.009	0.724	-8.240	6.857
5	1	0.009	0.723	-8.240	8.109
6	1	0.009	0.723	-8.240	9.237
7	1	0.009	0.723	-8.240	10.202
8	1	0.009	0.718	-8.240	11.083
9	1	0.009	0.690	-8.240	11.864
10	1	0.009	0.660	-7.447	12.553
11	1	0.009	0.641	-7.447	13.195
12	1	0.009	0.600	-7.447	13.781
13	1	0.009	0.584	-7.447	14.309
14	1	0.009	0.532	-7.447	14.803
15	1	0.009	0.537	-7.447	15.250
16	1	0.009	0.857	-7.447	15.667
17	1	0.009	0.858	-7.447	16.068
18	1	0.009	0.853	-7.447	16.409

续表

序号	FFT-lines	qthr	norm	SNR/dB	bval
19	2	0.018	0.824	-7.447	17.044
20	2	0.018	0.778	-6.990	17.607
21	2	0.018	0.740	-6.990	18.097
22	2	0.018	0.709	-6.990	18.528
23	2	0.018	0.676	-6.990	18.930
24	2	0.018	0.632	-6.990	19.295
25	2	0.018	0.592	-6.990	19.636
26	3	0.054	0.553	-6.990	20.038
27	3	0.054	0.510	-6.990	20.486
28	3	0.054	0.513	-6.990	20.900
29	4	0.114	0.608	-6.990	21.305
30	4	0.114	0.673	-6.020	21.722
31	5	0.452	0.637	-6.020	22.128
32	5	0.452	0.586	-6.020	22.512
33	5	0.452	0.571	-6.020	22.877
34	7	6.330	0.616	-5.229	23.241
35	7	6.330	0.640	-5.229	23.616
36	11	9.947	0.597	-5.229	23.974
37	17	153.727	0.538	-5.229	24.312

表Ⅲ-D. 3e Sampling-frequency=22.05kHz 短块

序号	FFT-lines	qthr	norm	SNR/dB	bval
0	1	4.532	0.952	-8.240	0.000
1	1	0.904	0.700	-8.240	1.723
2	1	0.029	0.681	-8.240	3.445
3	1	0.009	0.675	-8.240	5.057
4	1	0.009	0.667	-8.240	6.422
5	1	0.009	0.665	-8.240	7.609
6	1	0.009	0.664	-8.240	8.710
7	1	0.009	0.664	-8.240	9.662
8	1	0.009	0.664	-8.240	10.515
9	1	0.009	0.655	-8.240	11.293
10	1	0.009	0.616	-7.447	12.009
11	1	0.009	0.597	-7.447	12.625
12	1	0.009	0.578	-7.447	13.210
13	1	0.009	0.541	-7.447	13.748
14	1	0.009	0.575	-7.447	14.241
15	1	0.009	0.856	-7.447	14.695
16	1	0.009	0.846	-7.447	15.125
17	1	0.009	0.840	-7.447	15.508
18	1	0.009	0.822	-7.447	15.891
19	2	0.018	0.800	-7.447	16.537

续表

序号	FFT-lines	qthr	norm	SNR/dB	bval
20	2	0.018	0.753	-6.990	17.112
21	2	0.018	0.704	-6.990	17.620
22	2	0.018	0.674	-6.990	18.073
23	2	0.018	0.640	-6.990	18.470
24	2	0.018	0.609	-6.990	18.849
25	3	0.027	0.566	-6.990	19.271
26	3	0.027	0.535	-6.990	19.741
27	3	0.054	0.531	-6.990	20.177
28	3	0.054	0.615	-6.990	20.576
29	3	0.054	0.686	-6.990	20.950
30	4	0.114	0.650	-6.020	21.316
31	4	0.114	0.612	-6.020	21.699
32	5	0.452	0.567	-6.020	22.078
33	5	0.452	0.520	-6.020	22.438
34	5	0.452	0.513	-5.229	22.782
35	7	6.330	0.557	-5.229	23.133
36	7	6.330	0.584	-5.229	23.484
37	7	6.330	0.570	-5.229	23.828
38	19	171.813	0.578	-4.559	24.173

表III-D. 3f Sampling-frequency=16kHz 短块

序号	FFT-lines	qthr	norm	SNR/dB	bval
0	1	4.532	0.997	-8.240	0.000
1	1	0.904	0.893	-8.240	1.250
2	1	0.090	0.881	-8.240	2.500
3	1	0.029	0.873	-8.240	3.750
4	1	0.009	0.872	-8.240	4.909
5	1	0.009	0.871	-8.240	5.958
6	1	0.009	0.860	-8.240	6.857
7	1	0.009	0.839	-8.240	7.700
8	1	0.009	0.812	-8.240	8.500
9	1	0.009	0.784	-8.240	9.237
10	1	0.009	0.741	-7.447	9.895
11	1	0.009	0.697	-7.447	10.500
12	1	0.009	0.674	-7.447	11.083
13	1	0.009	0.651	-7.447	11.604
14	1	0.009	0.633	-7.447	12.107
15	1	0.009	0.611	-7.447	12.554
16	1	0.009	0.589	-7.447	13.000
17	1	0.009	0.575	-7.447	13.391
18	1	0.009	0.654	-7.447	13.781
19	2	0.018	0.724	-7.447	14.474

续表

序号	FFT-lines	qthr	norm	SNR/dB	bval
20	2	0.018	0.701	-6.990	15.096
21	2	0.018	0.673	-6.990	15.667
22	2	0.018	0.631	-6.990	16.177
23	2	0.018	0.592	-6.990	16.636
24	2	0.018	0.553	-6.990	17.057
25	2	0.018	0.510	-6.990	17.429
26	2	0.018	0.506	-6.990	17.786
27	3	0.027	0.562	-6.990	18.177
28	3	0.027	0.598	-6.990	18.597
29	3	0.027	0.589	-6.990	18.994
30	3	0.027	0.607	-6.020	19.352
31	3	0.027	0.620	-6.020	19.693
32	4	0.072	0.580	-6.020	20.066
33	4	0.072	0.532	-6.020	20.461
34	4	0.072	0.517	-5.229	20.841
35	5	0.143	0.517	-5.229	21.201
36	5	0.143	0.509	-5.229	21.549
37	6	0.172	0.506	-5.229	21.911
38	7	0.633	0.522	-4.559	22.275
39	7	0.633	0.531	-4.559	22.625
40	8	0.723	0.519	-3.980	22.971
41	10	9.043	0.512	-3.980	23.321

### III-D.4 将阈值计算分区转化为比例因子波段的表

表 III-D.4a Sampling-frequency=24kHz 长块

序号 sb	cbw	bu	bo	w1	w2
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.639
5	2	17	19	0.361	0.417
6	3	19	22	0.583	0.083
7	2	22	24	0.917	0.750
8	3	24	27	0.250	0.417
9	3	27	30	0.583	0.648
10	3	30	33	0.352	0.611
11	3	33	36	0.389	0.625

续表

序号 sb	cbw	bu	bo	w1	w2
12	4	36	40	0.375	0.144
13	3	40	43	0.856	0.389
14	3	43	46	0.611	0.160
15	3	46	49	0.840	0.217
16	3	49	52	0.783	0.184
17	2	52	54	0.816	0.886
18	3	54	57	0.114	0.313
19	2	57	59	0.687	0.452
20	1	59	60	0.548	0.908

表Ⅲ-D. 4b Sampling\_frequency=22.05kHz 长块

序号 sb	cbw	bu	bo	w1	w2
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.139
5	1	17	18	0.861	0.917
6	3	18	21	0.083	0.583
7	3	21	24	0.417	0.250
8	3	24	27	0.750	0.805
9	3	27	30	0.194	0.574
10	3	30	33	0.426	0.537
11	3	33	36	0.463	0.819
12	4	36	40	0.180	0.100
13	3	40	43	0.900	0.468
14	3	43	46	0.532	0.623
15	3	46	49	0.376	0.450
16	3	49	52	0.550	0.552
17	3	52	55	0.448	0.403
18	2	55	57	0.597	0.643
19	2	57	59	0.357	0.722
20	2	59	61	0.278	0.960

表Ⅲ-D. 4c Sampling\_frequency=16kHz 长块

序号 sb	cbw	bu	bo	w1	w2
0	1	0	2	1.000	0.528
1	2	2	4	0.472	0.305
2	2	4	6	0.694	0.083

续表

序号 sb	cbw	bu	bo	w1	w2
3	1	6	7	0.917	0.861
4	2	7	9	0.139	0.639
5	2	9	11	0.361	0.417
6	3	11	14	0.583	0.083
7	2	14	16	0.917	0.750
8	3	16	19	0.250	0.870
9	3	19	22	0.130	0.833
10	4	22	26	0.167	0.389
11	4	26	30	0.611	0.478
12	4	30	34	0.522	0.033
13	3	34	37	0.967	0.917
14	4	37	41	0.083	0.617
15	3	41	44	0.383	0.995
16	4	44	48	0.005	0.274
17	3	48	51	0.726	0.480
18	3	51	54	0.519	0.261
19	2	54	56	0.739	0.884
20	2	56	58	0.116	1.000

表Ⅲ-D. 4d Sampling frequency=24kHz 短块

序号 sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	4	15	19	0.833	0.583
6	3	19	22	0.417	0.917
7	4	22	26	0.083	0.944
8	4	26	30	0.055	0.042
9	2	30	32	0.958	0.567
10	3	32	35	0.433	0.167
11	2	35	37	0.833	0.618

表Ⅲ-D. 4e Sampling frequency=22.05kHz 短块

序号 sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500

续表

序号 sb	cbw	bu	bo	w1	w2
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	3	20	23	0.750	0.583
7	4	23	27	0.417	0.055
8	3	27	30	0.944	0.375
9	3	30	33	0.625	0.300
10	3	33	36	0.700	0.167
11	2	36	38	0.833	1.000

表 II -D. 4f Sampling-frequency = 16kHz 短块

序号 sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	4	20	24	0.750	0.250
7	5	24	29	0.750	0.055
8	4	29	33	0.944	0.375
9	4	33	37	0.625	0.472
10	3	37	40	0.528	0.937
11	1	40	41	0.062	1.000

## 附录 III-E 专利所有者清单

(此附录不作为本推荐|国际标准的必备部分)

用户请注意：在本推荐|国际标准内的某些处理——与可能要求使用此推荐|国际标准专利权范围内的发明权是一致的。

出版此推荐|国际标准，并不代表对有关这个主张或任何专利权的合法性。然而，此表中列出的每一个公司在下述的权限下与 ITTF 保留有归档授权许可证的意愿声明，即它们对获得此许可的申请者应持有理的和无差别的条款和条件。

关于这些专利的信息可以从以下组织获得。

该表总结了已接受的正式的专利陈述，并指出该陈述应用于标准的若干部分。该表包括有已提交非正式陈述的组织。然而，如果没有‘X’出现，就表示至今尚未收到该组织的正式陈述。

Company	ISO/IEC 13818-2	ISO/IEC 13818-3	ISO/IEC 13818-1
AT&T	X	X	X
BBC Research Department			
Bellcore	X		
Belgian Science Policy Office	X	X	X
BOSCH	X	X	X
CCETT			
CSELT	X		
David Sar 序号 ff Research Center	X	X	X
Deutsche Thomson-Brandt GmbH	X	X	X
France Telecom CNET			
Fraunhofer Gesellschaft		X	X
GC Tech 序号 logy Corporation	X	X	X
General Instruments			
Goldstar			
Hitachi. Ltd.			
International Bussiness Machines Corporation	X	X	X

续表

Company	ISO/IEC 13818-2	ISO/IEC 13818-3	ISO/IEC 13818-1
IRT		X	
KDD	X		
Massachusetts Institute of Tech 序号 logy	X	X	X
Matsushita Electric Industrial Co. , Ltd.	X	X	X
Mitsubishi Electric Corporation			
National Transcommunications Limited			
NEC Corporation		X	
Nippon Hoso Kyokai	X		
Nippon Telegraph and Telephone	X		
Nokia Research Center	X		
Norwegian Telecom Research	X		
Philips Consumer Electronics	X	X	X
OKI			
Qualcomm Incorporated	X		
Royal PTT Nederland N. V. ,PTT Research(NL)	X	X	X
Samsung Electronics			
Scientific Atlanta	X	X	X
Siemens AG	X		
Sharp Corporation			
Sony Corporation			
Texas Instruments			
Thomson Consumer Electronics			
Toshiba Corporation	X		
TV/Com	X	X	X
Victor Company of Japan Limited			