Society of Cable Telecommunications Engineers

ENGINEERING COMMITTEE
Digital Video Subcommittee

# AMERICAN NATIONAL STANDARD

# ANSI/SCTE 128 2010-a

# AVC Video Systems and Transport Constraints for Cable Television

## NOTICE



140 Philips Road

Exton, PA  19341

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## **Errata**

This version (2010-a) corrects editorial errors in the final approved document previously published as "ANSI/SCTE 128 2010", namely:

1) The missing legends for tables 9A, 9B, 9C were restored

2) The numbering of the Tables from 9A forward was corrected

There were no normative content changes.

# AVC Video Systems and Transport Constraints for Cable Television

## 1.0 SCOPE

This document defines the video coding and transport constraints on ITU-T Rec. H.264 | ISO/IEC 14496-10 [4] video compression (hereafter called "AVC") for Cable Television. In particular, this document describes the transmission of AVC coded video elementary streams in an MPEG-2 service multiplex (single or multi-program Transport Stream).

Note: The carriage of MPEG-2 video in the MPEG-2 service multiplex is described in SCTE 54 [1].

### 1.1 Background (Informative)

This document assists in creation of an AVC coded video elementary stream and its transport and is intended for broadcast purposes. There are other applications: time-shifting (e.g., PVR/DVR service), Video-on-Demand service, unicast, multicast, splicing (e.g., Ad-insertion) that could employ the specifications in this document. However, constraints specific to those applications are outside of the scope of this document.

## 2.0 NORMATIVE REFERENCES

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

### 2.1 SCTE References

[1] ANSI/SCTE 54 (2006), Digital Video Service Multiplex and Transport System Standard for Cable Television.

### 2.2 Standards from other Organizations

[2] ATSC A/65C, Program and System Information Protocol for Terrestrial Broadcast and Cable, Revision C, with Amendment No. 1, May 9, 2006; Section 6.9.2.

[3] ISO/IEC 13818-1, (2007), "Information Technology – Generic coding of moving pictures and associated audio – Part 1: Systems."

[4] ITU-T Rec. H.264 | ISO/IEC 14496-10, (2005), "Information Technology – Coding of audio visual objects – Part 10: Advanced Video Coding."

[5] CEA-608-C (2005), Line 21 Data Services.

[6] CEA-708-C (2006), Digital Television (DTV) Closed Captioning.

[7] ATSC A/53 Part 4:2007, Digital Television Standard, MPEG-2 Video System Characteristics, 3 January 2007.

[8] ETSI TS 101 154 V1.8.1 Digital Video Broadcasting (DVB): Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream, July, 2007.

[9] SMPTE 2016-1: Standard for Television – Format for Active Format Description and Bar Data.

[10] ISO/IEC 13818-2 (2000), Information Technology – Generic coding of moving pictures and associated audio - Part 2: Video

## 3.0 INFORMATIVE REFERENCES

The following documents may provide valuable information to the reader but are not required when complying with this standard.

### 3.1 SCTE References

[11] ANSI/SCTE 43 (2005), Digital Video Systems Characteristics Standard for Cable Television.

[12] ANSI/SCTE 21 (2001R2006), Standard For Carriage of NTSC VBI Data In Cable Digital Transport Streams.

[13] ANSI/SCTE 07 (2006), Digital Transmission Standard for Cable Television.

[14] DVS/714 [in preparation], Constraints on AVC Video Coding for Digital Program Insertion.

### 3.2 Standards from other Organizations

[15] SMPTE 170M (1999), Television – Composite Analog Video Signal – NTSC for Studio Applications.

[16] SMPTE 274M (2003), Standard for television, 1920 x 1080 Scanning and Interface.

[17] SMPTE 296M (2001), Standard for television, 1280 x 720 Scanning, Analog and Digital Representation, and Analog Interface.

[18] ITU-R BT.601-5 (1995), Encoding parameters of digital television for studios.

[19] ITU-R BT.709-4 (2000), Basic Parameter Values for the HDTV Standard for the Studio and for International Programme Exchange.

[20] ITU-T J.83B Digital (1997) and ERRATUM 1(1998), Digital Video Transmission Standard for Cable Television.

[21] CEA-CEB16: Active Format Description (AFD) & Bar Data Recommended Practice.

[22] SMPTE 125M (1995), Standard for television, Component Video Signal 4:2:2, Bit Parallel Digital Interface.

[23] SMPTE 293M (2003), Standard for television, 720x483 Active Line at 59.95 Hz Progressive Scan Production, Digital Representation.

[24] SMPTE 267M (1995), Standard for television, Bit Parallel Digital Interface- Component Video Signal 4:2:2 16x9 Aspect Ratio.

[25] ITU-T Rec. T.35, (2000), "Procedure for the allocation of ITU-T defined codes for non-standard facilities."

[26] ATSC A/53, Part 3:2007, "Service Multiplex and Transport Subsystem Characteristics", 3 January 2007.

[27] CEA-861-D "A DTV Profile for Uncompressed High Speed Digital Interfaces", Publication Date: July 1, 2006.

## 4.0 COMPLIANCE NOTATION

Throughout this document, there are words that are used to define the significance of particular requirements. These words are:

| | |
|---|---|
| "shall" | This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification. |
| "shall not" | This phrase means that the item is an absolute prohibition of this specification. |
| "should" | This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be |

| | |
|---|---|
| | understood and the case carefully weighted before choosing a different course. |
| "should not" | This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label. |
| "may" | This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item. |
| "forbidden" | The value specified shall never be used. |

This document contains symbolic references to syntactic elements used in the video and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., reserved), may contain the underscore character (e.g., constraint_set0_flag) and may consist of character strings that are not English words (e.g., pic_width_in_mbs_minus1).

# 5.0 DEFINITIONS AND ACRONYMS

## 5.1 Acronyms

The following definitions and acronyms are used in this document:

| | |
|---|---|
| ATSC | Advanced Television Systems Committee |
| AU | Access Unit |
| CPB | Coded Picture Buffer |
| DPB | Decoded Picture Buffer |
| DTS | Decoding Time Stamp |
| DTV | Digital Television |
| DVB | Digital Video Broadcasting |
| DVS | Digital Video Subcommittee |
| ESPI | Elementary_Stream_Priority_Indicator |
| FPP | Forward Predicted Picture |
| GOP | Group of Pictures |
| HDTV | High Definition Television |
| HRD | Hypothetical Reference Decoder |
| IDR | Instantaneous Decoding Refresh |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| MPEG | Moving Picture Experts Group |
| NAL | Network Abstraction Layer |
| PPS | Picture Parameter Set |
| PTS | Presentation Time Stamp |

QAM Quadrature Amplitude Modulation

RAI Random_Access_Indicator

SDTV Standard Definition Television

SEI Supplemental Enhancement Information

SPS Sequence Parameter Set

SRAP SCTE Random Access Point

T-STD Transport Stream System Target Decoder

TS Transport Stream

VBI Vertical Blanking Interval

VUI Video Usability Information

## 5.2 Definitions

AVC ITU-T Rec. H. 264 | ISO/IEC 14496-10 Advanced Video Coding standard

AVC Receiver The term "AVC Receiver" in this standard means a receiver having at least the attributes listed below:

1. Able to parse and decode the normative elements from AVC [4] that are specified with constraints in this standard;
2. Not adversely affected by the presence or absence of optional and informative elements from AVC [4];
3. Not adversely affected by the presence or absence of optional and informative elements in this standard;
4. Able to parse and process all elements from AVC [4] Annex D (SEI messages) and Annex E (VUI syntax elements) that are specified as normative in this standard and conveyed in-band; Note : These are optional elements in the AVC specification;
5. Able to parse and decode all the normative elements from ISO/IEC 13818-1 [3] that are normatively included and/or constrained by this standard;
6. Not adversely affected by the presence or absence of optional elements from ISO/IEC 13818-1 [3] (such as data in adaptation fields) that are specified with constraints in this standard;
7. Supports the processing of end_of_stream_rbsp() syntax element required by applications where another bitstream follows the end_of_stream NAL unit. The bitstream that follows will start with an IDR picture and may be accompanied by a time base discontinuity.

Note: The additional information from items 6 and 7 is optionally provided for the benefit of AVC receivers that include support for applications such as PVR, DPI and VOD.

Forward Predicted Picture A predicted picture that does not use any later-displayed picture as a reference.

Program An ISO-IEC 13818-1 MPEG-2 Program

SGOP A SCTE Group Of Pictures (SGOP) is the group of pictures spanning two consecutive SRAPs including the prior SRAP AU but not including the subsequent SRAP AU.

SRAP Picture An I- or IDR-picture that is part of an SRAP Access Unit.

Numerical formats are defined in the following table:

**Table 1: Numerical Format Definitions**

| Example Values | Description |
|---|---|
| 12345 | Example of a decimal value format |
| 0x2A | Example of a hexadecimal value format |
| '10010100' | Example of a string of binary digits |

## 6.0 MPEG-2 MULTIPLEX AND TRANSPORT CONSTRAINTS FOR AVC

This section and its subsections describe MPEG-2 System details pertaining to AVC that extends the specifications of SCTE 54 [1].

### 6.1 Services and Features

Note: As described in SCTE 54 [1] and other SCTE standards, the MPEG-2 Transport provides services and features enabled by information carried at the MPEG-2 Transport multiplex level and not at the video elementary stream component level. Some of these services are System Information and Program Guide, Emergency Alerts, and Specification of Private Data Services.

Note: The bitrate value for the AVC Bitstream is application dependent and limited by the contiguous bandwidth of the transmission channel. In the application of AVC transmission over a 64-QAM channel, bitrate value in combination with other bitstreams in the MPEG-2 Transport multiplex, conforms to a channel bitrate of less than or equal to 27.0 Mbps; in transmissions over 256-QAM channels to less than or equal to 38.8 Mbps.

This section describes additional services and features details pertaining to AVC.

### 6.2 MPEG-2 Systems Standard

#### 6.2.1 Video T-STD

Video T-STD for AVC shall be based on Section 2.14.3.1 of ISO/IEC 13818-1 [3] and shall follow the constraints for the profile and level encoded in the video elementary stream in Appendix A of AVC [4].

### 6.3 Assignment of identifiers

This section describes additional identifiers relevant to AVC video elementary stream components.

#### 6.3.1 AVC Stream Type Codes

The AVC stream type value shall be 0x1B.

### 6.3.2 Descriptors

#### 6.3.2.1 Video descriptor

AVC video is signaled by the AVC_video_descriptor() when required by ISO/IEC 13818-1[3] or as otherwise appropriate. This descriptor, when carried, shall be placed in the descriptor loop for the video program element of the PMT with a descriptor tag value of 0x28.

Certain services may include video elementary streams that contain one or more AVC still pictures that conform to the still picture model of ISO/IEC 13818-1 [3]. Any elementary stream containing still pictures shall include a AVC_video_descriptor() with AVC_still_present_flag set to "1" in accordance with section 2.6.64 of 13818-1[3]. Constraints for transmitting AVC still pictures are defined in section 9.0 of this document.

#### 6.3.2.2 Caption service descriptor

When caption services are delivered within the AVC video elementary stream (as specified in Section 8.0), the caption_service_descriptor(), as defined in Sec. 6.9.2 of ATSC A/65C [2], shall be present as described in SCTE 54 [1] section 5.8.3.6.

#### 6.3.2.3 SCTE Adaptation field data descriptor

When private data bytes of the adaptation field of the TS packets are in use, with tag, length, and data structures as defined in Section 6.4.3, the SCTE_adaptation_field_data_descriptor shall be placed in the descriptor loop for the video program element of the PMT. In the absence of such adaptation field private data, the descriptor shall not be included in the corresponding ES_info_loop of the PMT. The presence of the SCTE_adaptation_field_descriptor() shall mean that private_data_byte(s) carried in adaptation field data in this stream are in tag-length-data format, where the tag values are as defined in SCTE standards. The absence of this descriptor does not preclude private data in the adaptation header that could conform to MPEG or other standards. The SCTE adaptation field data descriptor shall be formatted per Table 2.

**Table 2: SCTE adaptation field data descriptor**

| Syntax | Number of bits | Identifier |
|---|---|---|
| SCTE_adaptation_field_data_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| } | | |

Where:

**descriptor_tag**: This value shall be set to 0x97

**descriptor_length:** This value is 0x00

Note: This descriptor varies from the signaling mechanism used by DVB[8] to indicate the presence of AU_Information. The descriptor establishes announcement for general and consistent syntax and semantics for the private data of the TS packet's adaptation field.

## 6.4 AVC Program Constraints

MPEG-2 Programs shall be constrained to carry at most one AVC video elementary stream component with a stream_type value of 0x1B.

### 6.4.1 SCTE Random Access Point (SRAP) Access Unit Composition

An **SCTE Random Access Point (SRAP)** access unit, or SRAP, demarcates a location within an AVC Bitstream where an AVC Receiver is able to begin decoding video. The spacing of successive random access points is an important contributor to channel change time, but is not the only factor contributing to channel change time. Other factors that contribute to channel change time include physical device tuning constraints, RF tuning, or conditional access operations.

An SRAP access unit is an AVC access unit shown pictorially in Figure 1. An SRAP access unit shall include exactly one Sequence Parameter Set (SPS) (that is active) with VUI and the Picture Parameter Set (PPS) that is required for decoding the associated picture. The SPS shall precede any SEI NAL units that may be present in an SRAP access unit.

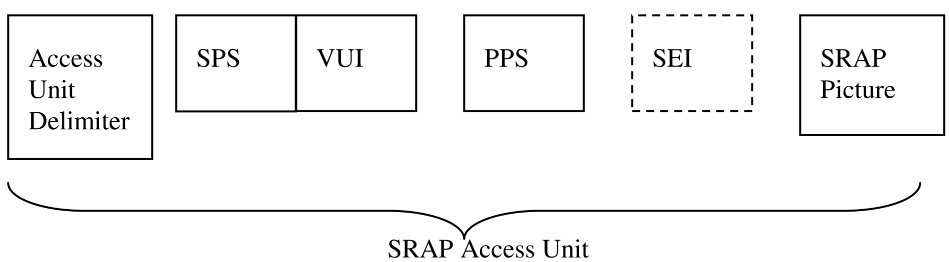


**Figure 1: NAL Unit Order for a Typical SRAP Access Unit**

The picture encoded within the SRAP access unit, the SRAP Picture, shall be either an I- or IDR-picture. In broadcast applications, this picture is typically a reference picture.

### 6.4.2 SRAP Transport Constraints

An SRAP shall meet the following transport constraints.

#### 6.4.2.1 TS Packet Header and Adaptation Field Constraints

A TS packet containing the PES packet header of an SRAP shall have an adaptation field. The payload_unit_start_indicator bit shall be set to '1' in the TS packet header and the adaptation_field_control bits shall be set to '11'(as per ISO/IEC 13818-1 [3]). In addition, the random_access_indicator bit in the Adaptation field of the TS packet that contains the PES packet header of the SRAP shall be set to '1' and follow the constraints as specified in ISO/IEC 13818-1 [3] in Subclause 2.4.3.5.

Per ISO/IEC 13818-1 [3], the elementary_stream_priority_indicator bit shall be set to '1' in the adaptation field of a TS packet containing the first slice start code of the SRAP Picture (which is an I or IDR picture with slice_type = 2 or slice_type = 7).

Both the random_access_indicator and elementary_stream_priority_indicator bits shall be set to '1' in the adaptation field of a TS packet containing the PES packet header of an SRAP if this TS packet also contains the first slice start code of the SRAP Picture. Otherwise, a TS packet with the elementary_stream_priority_indicator bit set to '1' shall immediately follow the TS packet with the random_access_indicator bit set to '1'.

Note 1: Setting of both a random_access_indicator and elementary_stream_priority_indicator bits for the access unit signifies an SRAP access unit.

Note 2: Multiple PPSs may be present in an SRAP access unit. The number of PPSs that may be present in an SRAP access unit is constrained by TS packet restrictions above (requiring both RAI and ESPI bits set in either the same TS packet or of successive TS packets). According to AVC [4], this requires all the bytes between the access unit delimiter NAL Unit and the start of the first slice of the SRAP Picture to be part of the payload of either the same TS packet or two successive TS packets.

6.4.2.2 SRAP Picture Decoding Time Stamp and SRAP Picture Presentation Time Stamp Constraints

The AVC Bitstream shall contain necessary elements such that all pictures with PTS greater than or equal to $DTS_{SRAP}$ + 0.5 seconds (where $DTS_{SRAP}$ represents the decoding time stamp of an SRAP Picture) are fully reconstructable and displayable when decoding starts at the SRAP picture.

> Note 1: This implies that any picture that has a PTS >= [$DTS_{SRAP}$ + 0.5 seconds] cannot be predicted directly or indirectly from reference pictures that were transmitted prior to the SRAP (i.e., with a lower value of DTS than $DTS_{SRAP}$). This also implies that any picture that was transmitted prior to the SRAP as well as any partially reconstructed pictures in the time interval [$DTS_{SRAP}$ , $DTS_{SRAP}$ + 0.5 seconds], cannot have a PTS that is greater than or equal to [$DTS_{SRAP}$ + 0.5 seconds].

The time difference between the receipt of an SRAP (actual value of PCR if present in the transport packet or computed value of PCR for the transport packet containing SRAP) and the DTS/PTS of its SRAP Picture is another key component in determining channel change time. The time difference between the receipt of an SRAP and the DTS of its SRAP Picture is also known as the initial video buffering delay of the AVC Bitstream in the CPB. Like in MPEG-2 video, the maximum possible initial video buffering delay is determined by the size of the CPB divided by the AVC Bitstream's actual bitrate. The initial video buffering delay shall be limited to 3 seconds or less. For applications requiring fast channel change or small initial delay after random access, the initial video buffering delay should be limited to one second or less.

> Note 2: The maximum delay of data through the T-STD is 10 seconds for AVC while it was 1 second for MPEG-2 video, and the ratio of CPB buffer size to maximum bitrate is also higher for AVC. In order to improve the channel change time or reduce the initial delay after random access for AVC, transmission systems should use a reasonable data delay without compromising the coding efficiency.

For broadcast applications where fast channel change is important, the $PTS_{SRAP}$ (where $PTS_{SRAP}$ represents the presentation time stamp of an SRAP Picture) should be less than or equal to $DTS_{SRAP}$ +0.5 seconds. This constraint bounds the time between decoding and presentation of an SRAP picture

6.4.2.3 Constraints on Decoding Time Stamps

For applications where fast channel change or random access is important, the maximum time interval between the decoding time stamp of successive SRAP Pictures shall be less than or equal to 1 second for integer frame rates, with appropriate adjustment (less than two pictures) for (1) non-integer frame rates and (2) small variabilities associated with scene change detection during encoding.

Note: The frequency at which SRAP access units are inserted into an AVC Bitstream is one of the key components in determining the channel change time and may simplify splicing and trick mode operations. If the interval between the Decoding Time Stamps of two successive SRAP Pictures is too small, such as 0.2 seconds, compression efficiency might be lowered significantly. On the other hand, if the interval between the Decoding Time Stamps of two successive SRAP Pictures is too large (such as 5 seconds), the time to effect a channel change or the initial delay after random access may be longer.

### 6.4.3 Adaptation Field Private Data

ISO/IEC 13818-1 [3] requires that the presence of an adaptation field be indicated by means of the adaptation_field_control, i.e. a 2-bit field in the header of the TS packet. The presence of private data bytes is signaled by means of the transport_private_data_flag coded at the beginning of the adaptation field.

When an adaptation field contains private data, the private_data_byte field shall contain the construct tag, length, data per Table 3.

**Table 3: private_data_byte**

| Syntax | No. of Bits | Format |
|---|---|---|
| private_data_byte { | | |
|     for i=0 to n { | | |
|         tag | 8 | bslbf |
|         length | 8 | bslbf |
|         If (tag==0xDF) | | |
|             format identifier | 32 | bslbf |
|         data() | var | |
|     } | | |
| } | | |

tag: tag shall take a value from Table 4: Tag Values

**Table 4: Tag Values**

| Tag Values | Description |
|---|---|
| 0x00 | Forbidden |
| 0x01 | Used by DVB |
| 0x02 | AU_Information |
| 0x03-0xDE | Reserved for future standardized use. See ATSC Code Points Registry in addition to this standard. |
| 0xDF | Registered Private Data |
| 0xE0-0xFE | User Private (unmanaged, therefore collisions between different users or applications may occur, except perhaps in totally closed systems) |
| 0xFF | Reserved for future extensions |

Note: The syntax and semantics for the tag value associated with AU_Information (tag value 0x02) can be found in Appendix A. This standard places no constraint on the definition of new tag values that conform to the structure defined herein. The syntax and semantics for other tag values, when defined, may be found in other SCTE or other standards.

Tag value 0xDF is registered private data, managed by the format_identifier field.

length: this field is the number of bytes following this length field

format_identifier: this field shall be as defined by ISO/IEC 13818-1, Section 2.6.9, Section 2.10, and Annex O. Only registered values are permitted.

data: one or more bytes corresponding to the tag value

Even though multiple collections of tag, length, data may be contained in consecutive TS packets, each collection of tag, length, data shall be contained within one TS packet only

The total number of private data bytes is specified by means of the transport_private_data_length, an 8-bit field that is directly followed by the private data bytes. The private data bytes may be composed of one or more data fields. This syntax does not allow gaps between two data fields.

#### 6.4.3.1 Optional Transport Adaptation Layer Information

Tag value 0x02 may be used to support PVR and other applications where additional information is placed in the adaptation field of TS packets as described in Appendix A. This information is optional; however when it is present, it shall follow the constraints specified in Appendix A.

## 6.5 PES constraints

Each PES packet shall contain only one AVC access unit start, as defined in Sections 2.1.3 and 2.14.1 of 13818-1[3]. The AVC access unit start shall occur in the same TS packet as the PES packet header, unless to do so would require bit stuffing. In this case the AVC access unit start shall occur in the next TS packet of the bitstream with the same

PID. Each PES packet header shall contain a PTS and DTS if DTS differs from the PTS. PES packetization shall comply to ISO/IEC 13818-1 even under system time base or continuity counter discontinuities signaled by setting discontinuity_indicator to '1' in the adaptation header.

Note: Per 13818-1, the payload_unit_start_indicator bit is set to '1' in the TS packet header of a TS packet containing a PES packet header. The payload of this TS packet will commence with the first byte of the PES packet.

# 7.0 AVC VIDEO CONSTRAINTS

## 7.1 Possible video inputs

While not required by this standard, there are certain television production standards, shown in Table 5, that define video formats that relate to compression formats specified by this standard.

**Table 5: Standardized Video Input Formats**

| Video standard | Active lines | Active samples/ line |
|---|---|---|
| SMPTE 274M [15] | 1080 | 1920 |
| SMPTE 296M [17] | 720 | 1280 |
| ITU-R BT.601-5 [18] | 483[1] | 720 |

The compression formats may be derived from one or more appropriate video input formats. It may be anticipated that additional video production standards may be developed in the future that extend the number of possible input formats.

## 7.2 Source coding specification

The AVC video compression algorithm shall conform to the High or Main Profile syntax of AVC[4]. AVC is specified herein as bitstreams compliant to a constrained set of High or Main Profile at Level 3.0, 4.0, or 4.2 (level_idc equal to 30, 40 or 42 respectively). Unless specified otherwise in this document, the allowable parameters shall be bounded by the upper limits specified in the AVC Specification [4].[2]

Profiles and levels shall be constrained as shown in Tables 6, 9A, 9B and 9C (indicated values for profile_idc and level_idc).

Additionally, AVC bitstreams shall meet the constraints and specifications described in this document. AVC bitstreams shall utilize the SEI and the VUI syntactic elements defined in AVC [4] Annexes D and E respectively in accordance with this specification. VUI and SEI messages expected to be processed by an AVC Receiver are specified herein. Some VUI and SEI messages are optional and may be ignored by the AVC Receiver as specified

[1]The number of active lines is not specified in ITU-R.601-5 [18][19]. "483" is the original number of active lines specified in the NTSC standard. However current accepted practice in North America allows the line count to be anywhere from 480 to 486.

[2] See ISO/IEC 14496-10 [4], Annex A for more information regarding profiles and levels.

herein. AVC Receivers should be made under the assumption that any legal structure as permitted by AVC may occur in the broadcast stream even if presently reserved or unused.

7.2.1 Constraints with respect to AVC

The tables in the following sections list the allowed values for each of the AVC syntactic elements that are restricted beyond the limits imposed by High Profile @ Level 4.0 or 4.2 in the AVC Specification.

7.2.1.1 Sequence Parameter Set (SPS) constraints

For each SRAP, there shall be one active Sequence Parameter Set (SPS) present in the bit stream. Table 6 identifies parameters in the Sequence Parameter Set of a bit stream that shall be constrained by the video subsystem and lists the allowed values for each.

**Table 6: Sequence Parameter Set Constraints**

| Parameter Set Syntactic Element | Allowed Value |
|---|---|
| profile_idc | 100 or 77 |
| constraint_set0_flag | 0 |
| constraint_set1_flag | 0 (when profile_idc is 100) and<br>1 (when profile_idc is 77) |
| constraint_set2_flag | 0 |
| constraint_set3_flag | 0 |
| level_idc | See Tables 9A, 9B and 9C |
| num_ref_frames | Less than or equal to MaxFrameBuffers (See Tables 9A, 9B and 9C) |
| chroma_format_idc | 1 (when profile_idc is 100)<br>N/A (when profile_idc is 77) |
| gaps_in_frame_number_value_allowed_flag | 0 |
| pic_width_in_mbs_minus1 | See Tables 9A, 9B and 9C |
| pic_height_in_map_units_minus1 | See Tables 9A, 9B and 9C |
| vui_parameters_present_flag | 1 |

All AVC Receivers are expected to be capable of processing AVC Bitstreams that have profile_idc(s) of 100 in accordance with the parameters and constraints set herein. Note that these AVC Receivers should process bitstreams with profile_idc = 77 also.

The time interval between consecutive changes in pairs of pic_width_in_mbs_minus1 and pic_height_in_map_units_minus1 shall be greater than or equal to one second.

7.2.1.2 Video Usability Information (VUI) Constraints

The AVC Receiver is expected to process the following VUI syntax elements:

**Table 7: VUI Constraints**

| VUI Header Syntactic Element | Allowed Value |
|---|---|
| aspect_ratio_idc | See Tables 9A, 9B and 9C |
| colour_primaries | see below |

| | |
|---|---|
| transfer_characteristics | see below |
| matrix_coefficients | see below |
| chroma_sample_loc_type_top_field | used |
| chroma_sample_loc_type_bottom_field | used |
| num_units_in_tick | See Table 11 |
| time_scale | See Table 11 |
| fixed_frame_rate_flag | 1 ( equals 0 for Low Delay mode and still pictures) |
| pic_struct_present_flag | used |
| max_dec_frame_buffering | equal to MaxFrameBuffers (See Tables 9A, 9B and 9C) (if present) |

While any appropriate values for each of the following 3 parameters in the VUI: colour_primaries, transfer_characteristics, and matrix_coefficients, as defined in Tables E-3, E-4, and E-5 of AVC [4], are allowed in the transmitted bit stream, it is noted that ITU-R BT.709 [19] and SMPTE 170M [15] are the most likely to be in common use.

The preferred values for colour_primaries, transfer_characteristics and matrix_coefficients are defined to be ITU-R BT.709 [19] for the first two row entries in Table 5. For the bottom row entry in Table 5, the preferred values for colour_primaries, transfer_characteristics and matrix_coefficients are defined to be SMPTE 170M[15].

Note: Syntactical elements that are used require that the immediate parent "xxx_present_flag", if it exists, to be enabled (for example, the colour_description_present flag).

7.2.1.3 Picture Parameter Constraints and Level Limits

AVC Bitstreams shall not include non-paired fields (as defined in AVC).

All pictures in AVC Bitstreams shall be displayable pictures.

Between two SRAPs, the content of a picture parameter set with a particular pic_parameter_set_id shall not change. That is, if more than one picture parameter set is present in the bitstream and these picture parameter sets are different from each other, then each picture parameter set shall have a different pic_parameter_set_id.

7.2.1.4 Supplemental Enhancement Information (SEI) Constraints

**Table 8: SEI Constraints**

| SEI Header Syntactic Element | Usage Constraints |
|---|---|
| | |
| Picture Timing SEI message | Optional, but required if picture structure information is carried |
| | |
| User data registered by ITU-T Rec. T.35[25] SEI message | Required for carriage of AFD, closed captioning, and/or bar data structures |

For bitstreams that carry the picture structure information (such as film mode), the pic_struct_present_flag shall be set to '1' in the VUI. If the pic_struct_present_flag is set to '1' in the VUI, then per AVC[4] a picture timing SEI is required to be associated with each access unit in the coded video sequence. If the coded video sequence does not require picture structure information, then the pic_struct_present_flag should be set to '0' in the VUI. This flag in the VUI allows use of a picture timing SEI message with only the picture structure information without the need to include HRD information (such as CPB and DPB delay or initial values of the delay in the buffering period SEI).

The Buffering Period SEI message is optional and may be ignored since this duplicates the functionality defined in the MPEG-2 transport level. The Pan-scan SEI message is optional but not recommended. See Section 8.2.3. All other SEI messages are optional.

When supporting AFD, bar data, and closed captioning (see section 8.0 for more details), SEI user_data_registered_itu_t_t35 shall be used.

#### 7.2.1.5 Compression format constraints

Tables 9A, 9B and 9C list the allowed compression formats and constraints for associated parameters (for non low delay mode applications). Table 9A covers Level 3.0 formats, Table 9B covers Level 4.0 formats, and Table 9C covers Level 4.2 formats. AVC Receivers that are capable of decoding Level 4.0 formats are also expected to be capable of decoding Level 3.0 formats. AVC Receivers that are capable of decoding Level 4.2 formats are also expected to be capable of decoding Level 4.0 and Level 3.0 formats. See Section 7.2.1.6 which specifies additional constraints for low delay mode applications.

The value of "MaxFrameBuffers" is specified in Tables 9A, 9B and 9C below. For each of the resolutions in Tables 9A, 9B and 9C, the coded video sequence shall not require the units of frame buffers in the DPB (Decoded Picture Buffer) to be greater than MaxFrameBuffers to enable the output of the decoded pictures at the specified output times.

The syntax element num_ref_frames in the AVC Sequence Parameter Set shall be set to a value less than or equal to the value MaxFrameBuffers. If the syntax element max_dec_frame_buffering is present in the VUI parameters syntax structure of the sequence parameter set, its value shall be set equal to MaxFrameBuffers. If the syntax element max_dec_frame_buffering is not present in the VUI parameters syntax structure of the sequence parameter set, the bitstream shall still obey the same constraints as if the syntax element max_dec_frame_buffering had been present and equal to MaxFrameBuffers.

**Table 9A: Level 3.0 Compression Format Constraints (level_idc = 30)**

| vertical size | horizontal size | PicWidthInMbs | PicHeightInMbs | MaxFrameBuffers [4] | aspect_ratio_idc | Display aspect ratio | Allowed frame rates | Progressive interlaced |
|---|---|---|---|---|---|---|---|---|
| 480 | 720 | 45 | 30 | 6 | 5 | 16:9 | 1,2,4,5 | P |
| 480 | 720 | 45 | 30 | 6 | 3 | 4:3 | 1,2,4,5 | P |
| 480 | 720 | 45 | 30 | 6 | 5 | 16:9 | 4,5 | I |
| 480 | 720 | 45 | 30 | 6 | 3 | 4:3 | 4,5 | I |
| 480 | 704 | 44 | 30 | 6 | 5 | 16:9 | 1,2,4,5 | P |
| 480 | 704 | 44 | 30 | 6 | 3 | 4:3 | 1,2,4,5 | P |
| 480 | 704 | 44 | 30 | 6 | 5 | 16:9 | 4,5 | I |
| 480 | 704 | 44 | 30 | 6 | 3 | 4:3 | 4,5 | I |
| 480 | 640 | 40 | 30 | 6 | 1 | 4:3 | 1,2,4,5 | P |
| 480 | 640 | 40 | 30 | 6 | 1 | 4:3 | 4,5 | I |
| 480 | 544 | 34 | 30 | 6 | 5 | 4:3 | 1,4 | P |
| 480 | 544 | 34 | 30 | 6 | 5 | 4:3 | 4 | I |
| 480 | 528 | 33 | 30 | 6 | 5 | 4:3 | 1,4 | P |
| 480 | 528 | 33 | 30 | 6 | 5 | 4:3 | 4 | I |
| 480 | 352 | 22 | 30 | 6 | 7 | 4:3 | 1,4 | P |
| 480 | 352 | 22 | 30 | 6 | 7 | 4:3 | 4 | I |

**Legend:**
frame rate: 1 = 23.976 Hz, 2 = 24 Hz, 4 = 29.97 Hz, 5 = 30 Hz, 7 = 59.94 Hz, 8 = 60 Hz
aspect_ratio_idc: 1 = 1:1 [square samples], 3 = 10:11, 5 = 40:33, 7 = 20:11, 14= 4:3

**Table 9B: Level 4.0 Compression Format Constraints (level_idc = 40)**

| vertical size | horizontal size | PicWidthInMbs | PicHeightInMbs | MaxFrameBuffers [4] | aspect_ratio_idc | Display aspect ratio | Allowed frame rates | Progressive interlaced |
|---|---|---|---|---|---|---|---|---|
| 1080 | 1920 | 120 | 68 | 4 | 1 | 16:9 | 1,2,4,5 | P |
| 1080 | 1920 | 120 | 68 | 4 | 1 | 16:9 | 4,5 | I |
| 1080 | 1440 | 90 | 68 | 4 | 14 | 16:9 | 1,2,4,5 | P |
| 1080 | 1440 | 90 | 68 | 4 | 14 | 16:9 | 4,5 | I |
| 720 | 1280 | 80 | 45 | 9 | 1 | 16:9 | 1,2,4,5,7,8 | P |
| 480 | 720 | 45 | 30 | 9 | 5 | 16:9 | 7,8 | P |
| 480 | 720 | 45 | 30 | 9 | 3 | 4:3 | 7,8 | P |
| 480 | 704 | 44 | 30 | 9 | 5 | 16:9 | 7,8 | P |
| 480 | 704 | 44 | 30 | 9 | 3 | 16:9 | 7,8 | P |
| 480 | 640 | 40 | 30 | 9 | 1 | 4:3 | 7,8 | P |

**Legend:**
frame rate: 1 = 23.976 Hz, 2 = 24 Hz, 4 = 29.97 Hz, 5 = 30 Hz, 7 = 59.94 Hz, 8 = 60 Hz
aspect_ratio_idc: 1 = 1:1 [square samples], 3 = 10:11, 5 = 40:33, 7 = 20:11, 14= 4:3

**Table 9C: Level 4.2 Compression Format Constraints (level_idc = 42)**

| vertical size | horizontal size | PicWidthInMbs | PicHeightInMbs | MaxFrameBuffers [4] | aspect_ratio_idc | Display aspect ratio | Allowed frame rates | Progressive interlaced |
|---|---|---|---|---|---|---|---|---|
| 1080 | 1920 | 120 | 68 | 4 | 1 | 16:9 | 7,8 | P |
| 1080 | 1440 | 90 | 68 | 4 | 14 | 16:9 | 7,8 | P |

**Legend:**
frame rate: 7 = 59.94 Hz, 8 = 60 Hz
aspect_ratio_idc: 1 = 1:1 [square samples], 3 = 10:11, 5 = 40:33, 7 = 20:11, 14= 4:3

For pictures with vertical sizes of 1080, 1088 lines shall be coded in order to satisfy the AVC requirement that the coded vertical size be a multiple of 16 (progressive scan) or 32 (interlaced scan). The bottom 8 lines shall be disregarded by a decoder. The value of frame_crop_top_offset shall be 0 and frame_crop_bottom_offset shall be 2*(1 + frame_mbs_only_flag).

The maximum values of Max Frame Size, Max Video Bit Rate, MaxCPB and MaxDPB shall not exceed the values shown in Table 10. These values are based on the highest picture resolutions specified in Tables 9A, 9B and 9C. Values for Max Video Bit Rate and MaxCPB should follow the constraints listed in Table 10 unless limited by the contiguous bandwidth of the transmission channel minus any additional data overhead needs.

**Table 10: Level and Computed Values to Support Table 9A, 9B and 9C**

| Level | Max Frame Size (MacroBlocks) | Max Video Bit Rate | MaxCPB | MaxDPB ( units of 1024 bytes ) |
|---|---|---|---|---|
| Level 3.0 | 1350 | 15 | 15 | 3037.5 |
| Level 4.0 | 8160 | 30 | 37.5 | 12440 |
| Level 4.2 | 8160 | 30 | 37.5 | 12440 |

Note: Bitrates and CPB size calculations performed per ISO/IEC 14496-10 [4] Annex A and ISO/IEC 13818-1 [3] Section 2.14.3.1

Table 11 lists time_scale and num_units_in_tick need to set for Progressive and Interlaced frame rates.

**Table 11: Time_scale & num_units_in_tick settings for Frame Rates**

| Frame Rate (Hz) | Interlaced/ Progressive | time_scale | num_units_in_tick |
|---|---|---|---|
| 23.976 | P | 48000 | 1001 |
| 24 | P | 48 | 1 |
| 29.97 | P | 60000 | 1001 |
| 30 | P | 60 | 1 |
| 29.97 | I | 60000 | 1001 |
| 30 | I | 60 | 1 |
| 59.94 | P | 120000 | 1001 |
| 60 | P | 120 | 1 |

7.2.1.6 Low Delay Mode

Low Delay mode corresponds to low_delay_hrd_flag = '1' and is signaled by fixed_frame_rate_flag = '0' in the VUI (per Table 7). Low Delay mode shall satisfy all of the following:

1. All pictures shall be an IDR, I, or FPP.

   Note: AVC receivers may ignore pic_struct (if present in the picture timing SEI) for Low Delay mode applications. In some cases, pic_struct values (1, 2, 5 or 6) could cause field parity issues in receivers when decoded pictures are repeated.

2. All transmitted pictures shall be displayable pictures and transmitted in display order. For most applications, the PTS is expected to be equal to the DTS and thus DTS is not present in the PES packet header.

3. The maximum number of reference pictures shall be one less than non-Low Delay mode.

4. The last fully reconstructed picture in Low Delay mode shall be displayed indefinitely until a subsequent picture is fully reconstructed and available for display.

   Note: If required, AVC receivers can determine the display frame rate from the VUI parameters num_units_in_tick and time_scale (see Table 11).

5. The T-STD management for Low Delay mode shall be consistent with the extensions for AVC specified in section 2.14.3 of 13818-1[3].

   Note: Per Annex E of AVC [4], low_delay_hrd_flag can either be present in the VUI or conveyed by other means. If low_delay_hrd_flag is present in the VUI, then (per Annex D and Annex E of AVC) bitstreams must include buffering period SEI and picture_timing SEI with the appropriate values of CPB and DPB delay values for each access unit. If low_delay_hrd_flag is present in the VUI and set to '1', then AVC receivers must use the CPB and DPB delay values from the picture timing SEI for T-STD management instead of the PTS and DTS values coded in the PES header of each access unit (per section 2.14.3 of 13818-1 [3]). If low_delay_hrd_flag is not present in the VUI and fixed_frame_rate_flag is set to '0', AVC receivers are expected to assume Low Delay mode (I.E:, low_delay_hrd_flag = '1'which allows buffer underflow) and may use the PTS and DTS values coded in the PES header for T-STD management.

6. The fixed_frame_rate_flag shall be set to zero for transmission, however an AVC Receiver may ignore the fixed_frame_rate_flag in Low Delay mode.

7.2.1.7 Program Splicing Constraint

System processes (such as digital ad insertion and program splicing) may require a resolution change in the AVC stream within the same program that results in a seamless or near-seamless behavior in the AVC receiver. When a user of this standard wishes to facilitate such a change, the AVC elementary stream shall be encoded in accordance with these additional constraints (also see DVS 714 [14]):

If such seamless or near-seamless behavior in the AVC receiver is desired, then level_idc and the vertical picture size in the AVC elementary stream should not change within the same program (also see DVS 714 [14]).

> Note: profile changes, display aspect ratio changes, frame rate changes, and interlaced/progressive transitions (in either order) should be avoided as they may result in disruption of the decoder's video output.

For transmissions that conform to the above constraints, the AVC Receiver is expected to manage the MaxDpbSize (defined in [4]) as constrained through MaxFrameBuffers in Tables 9A, 9B and 9C, the MaxDPB as constrained in Table 10, and process the no_output_of_prior_pics_flag in the IDR picture of sequence after the transition correctly. In all other cases the AVC Receiver may infer no_output_of_prior_pics_flag to be '1' and clear the DPB.

## 8.0 CARRIAGE OF CAPTIONING, AFD, AND BAR DATA

The carriage of closed captions, AFD, and bar data when present shall be carried as specified in the following sections.

### 8.1 Encoding and transport of caption, active format description (AFD) and bar data

Advanced DTV closed captions (CEA-708 [6]), when present, shall be encoded in accordance with CEA-708 and shall be transported as specified in Section 8.1.1. Line 21 caption data, encoded in accordance with CEA-608[5], when present shall be transported as specified in CEA-708 and Section 8.1.1.

Note: CEA-708 requires a fixed bandwidth of 9600 bits per second for the closed caption payload data. Bandwidth calculations should anticipate this requirement.

#### 8.1.1 Caption, AFD and Bar Data Syntax

Caption, AFD and bar data shall be carried in the SEI raw byte sequence payload (RBSP) syntax of the video Elementary Stream. Table 12 describes the common data syntax (see AVC, Annex D.1.5 and D.2.5 [4]).

**Table 12: Common Data Syntax[3]**

| Syntax | No. of Bits | Format |
|---|---|---|
| user_data_registered_itu_t_t35 ( ) { | | |
| itu_t_t35_country_code | 8 | bslbf |
| itu_t_t35_provider_code | 16 | bslbf |
| user_identifier | 32 | bslbf |
| user_structure() | | |
| } | | |

Note that SEI payloads carrying a SEI payloadType of 4 and containing a 32-bit field following the itu_t_t35_provider_code which has a value other than user_identifier may be present in an SCTE-compliant AVC video bit stream. Receiving devices are expected to process this field and use it to determine the syntax and semantics of the user data construct to follow.

Receiving devices are expected to silently discard any unrecognized SEI payloads encountered in the video bit stream. For example, if an unrecognized 32-bit identifier is seen following the itu_t_t35_provider_code, or an unrecognized 8-bit user_data_type_code (see Section 8.2) is seen following the ATSC1_data, data should be discarded until another SEI payload is seen or the RBSP terminates.

Note: The values specified below for both itu_t_t35_country_code and itu_t_35_provider_code are the assigned values for the purposes of this standard. This does not imply that other uses of this SEI construct will not also be used for other applications. See ITU-T Recommendation T.35 [25] for additional information.

[3] Shaded cells in this table indicate syntactic and semantic additions to the ISO/IEC 14496-10 Standard [4]

### 8.1.2 Caption, AFD and Bar Data Semantics

itu_t_t35_country_code – A fixed 8-bit field, the value of which shall be 0xB5.

itu_t_35_provider_code – A fixed 16-bit field registered by the ATSC. The value shall be 0x0031.

user_identifier – This is a 32 bit code that indicates the contents of the user_structure() as indicated in Table 13.

user_structure() – This is a variable length data structure defined by the value of user_identifier and Table 13

**Table 13: user_identifier**

| user_identifier | user_structure() |
|---|---|
| 0x47413934 (“GA94”) | ATSC1_data() |
| 0x44544731 (“DTG1”) | afd_data() |
| all other values | SCTE/ATSC Reserved |

## 8.2 ATSC1_data() Syntax

Table 14 describes the ATSC1_data() syntax which shall be used.

**Table 14: ATSC1_data() Syntax**

| Syntax | No. of Bits | Format |
|---|---|---|
| ATSC1_data( ) { | | |
| user_data_type_code | 8 | uimsbf |
| user_data_type_structure() | var | |
| marker_bits | 8 | '11111111' |
| } | | |

### 8.2.1 ATSC1_data() Semantics

user_data_type_code – An 8-bit value that identifies the type of user data to follow in the user_data_type_structure(). The values are defined in Table 15.

**Table 15: user_data_type_code**

| user_data_type_code | user_data_type_structure() |
|---|---|
| 0x00 – 0x02 | SCTE/ATSC Reserved |
| 0x03 | cc_data() [1] |
| 0x04 | SCTE/ATSC Reserved |
| 0x05 | SCTE/ATSC Reserved |

| | |
|---|---|
| 0x06 | bar_data() |
| 0x07 – 0xFF | SCTE/ATSC Reserved |
| Footnote:<br>[1] Table 16 below is included for the convenience of the reader and is identical to Table 2 of CEA-708-C and may be removed in a future edition of this standard. | |

user_data_type_structure – This is a variable length set of data defined by the value of user_data_type_code and Table 15.

8.2.2 Encoding and Transport of Caption Data[4]

Table 16 describes the syntax and the semantics of caption data.

**Table 16: Caption Data Syntax**

| Syntax | No. of Bits | Format |
|---|---|---|
| cc_data() { | | |
| reserved | 1 | ’1’ |
| process_cc_data_flag | 1 | bslbf |
| zero_bit | 1 | '0' [1] |
| cc_count | 5 | uimsbf |
| reserved | 8 | ‘1111 1111’ |
| for ( i=0 ; i < cc_count ; i++ ) { | | |
| one_bit | 1 | ‘1’ |
| reserved | 4 | ‘1111’ |
| cc_valid | 1 | bslbf |
| cc_type | 2 | bslbf |
| cc_data_1 | 8 | bslbf |
| cc_data_2 | 8 | bslbf |
| } | | |
| } | | |
| Footnote:<br>[1] For backwards compatibility, this bit shall be zero, not one. | | |

[4] The syntax and semantics of cc_data() may be moved to CEA-708 [6]. At that point this standard should be amended to delete this entire section in deference to the CEA-708 definition. This syntax is bit-compatible with the existing syntax defined by previous versions of CEA-708.

process_cc_data_flag – This flag is set to indicate whether it is necessary to process the cc_data. If it is set to 1, the cc_data has to be parsed and its meaning has to be processed. When it is set to 0, the cc_data may be discarded.

zero_bit – This bit shall be '0' to maintain backwards compatibility with previous versions of CEA-708 [6].

cc_count – This 5-bit integer indicates the number of closed caption constructs following this field. It may have values 0 through 31. The value of cc_count shall be set according to the frame rate and coded picture structure (field or frame) such that a fixed bandwidth of 9600 bits per second is maintained for the closed caption payload data. Sixteen (16) bits of closed caption payload data are carried in each pair of the fields cc_data_1 and cc_data_2.

one_bit – This bit shall be '1' to maintain backwards compatibility with previous versions of CEA-708 [6].

cc_valid – This flag shall be set to '1' to indicate that the two closed caption data bytes that follow are valid. If set to '0' the two data bytes are invalid, as defined in [6].

cc_type – Denotes the type of the two closed caption data bytes that follow, as defined in [6].

cc_data_1 – The first byte of a closed caption data pair as defined in [6].

cc_data_2 – The second byte of a closed caption data pair as defined in [6].

### 8.2.3 Encoding and transport of bar data

Bar data, when present, shall be encoded and transported using the ATSC1_data() structure defined in Table 14 and the assigned value for user_data_type_code shown in Table 15. Table 17 describes the syntax of bar data. Bar data should be included in an SEI message whenever the rectangular picture area containing useful information does not extend to the full height or width of the coded frame and AFD alone is insufficient to describe the extent of the image. See Section 8.2.4.

When bar_data() is present in the Video Elementary Stream, the SEI pan_scan_rect() parameters in the SEI RBSP syntax (AVC, Annex D.1.3 and D.2.3 [4]) shall not be present. Bar data is to be preferred over the use of the SEI pan_scan_rect().

At an SRAP, unless AFD data is present specifying otherwise, the absence of bar data shall indicate that the rectangular picture area containing useful information extends to the full height and width of the coded frame.

Bar data is constrained (below) to be signaled in pairs, either top and bottom bars or left and right bars, but not both pairs at once. Bars may be unequal in size. One bar of a pair may be zero width or height.

**Table 17: Bar Data Syntax**

| Syntax | No. of Bits | Format |
|---|---|---|
| bar_data() { | | |
| top_bar_flag | 1 | bslbf |
| bottom_bar_flag | 1 | bslbf |
| left_bar_flag | 1 | bslbf |
| right_bar_flag | 1 | bslbf |
| Reserved | 4 | '1111' |
| if (top_bar_flag == '1') { | | |
| marker_bits | 2 | '11' |
| line_number_end_of_top_bar | 14 | uimsbf |
| } | | |
| if (bottom_bar_flag == '1') { | | |
| marker_bits | 2 | '11' |
| line_number_start_of_bottom_bar | 14 | uimsbf |
| } | | |
| if (left_bar_flag == '1') { | | |
| marker_bits | 2 | '11' |
| pixel_number_end_of_left_bar | 14 | uimsbf |
| } | | |
| if (right_bar_flag == '1') { | | |
| marker_bits | 2 | '11' |
| pixel_number_start_of_right_bar | 14 | uimsbf |
| } | | |
| } | | |

Designation of line numbers for line_number_end_of_top_bar and line_number_start_of_bottom_bar is video format-dependent and shall conform to the applicable standard indicated in Table 18.

top_bar_flag – This flag shall indicate, when set to '1', that the top bar data is present. If left_bar_flag is '1', this flag shall be set to '0'.

bottom_bar_flag – This flag shall indicate, when set to '1', that the bottom bar data is present. This flag shall have the same value as top_bar_flag.

left_bar_flag – This flag shall indicate, when set to '1', that the left bar data is present. If top_bar_flag is '1', this flag shall be set to '0'.

right_bar_flag – This flag shall indicate, when set to '1', that the right bar data is present. This flag shall have the same value as left_bar_flag.

line_number_end_of_top_bar – A 14-bit unsigned integer value representing the last line of a horizontal letterbox bar area at the top of the reconstructed frame. Designation of line numbers shall be as defined in Table 18.

line_number_start_of_bottom_bar – A 14-bit unsigned integer value representing the first line of a horizontal letterbox bar area at the bottom of the reconstructed frame. Designation of line numbers shall be as defined in Table 18.

pixel_number_end_of_left_bar – A 14-bit unsigned integer value representing the last horizontal luminance sample of a vertical pillarbox bar area at the left side of the reconstructed frame. Pixels shall be numbered from zero, starting with the leftmost pixel.

pixel_number_start_of_right_bar – A 14-bit unsigned integer value representing the first horizontal luminance sample of a vertical pillarbox bar area at the right side of the reconstructed frame. Pixels shall be numbered from zero, starting with the leftmost pixel.

The range of line numbers and pixels within the coded frame for each image format shall be as specified in Table 2 of SMPTE 2016-1[9] as extended by Table 18 below. Information from SMPTE 2016-1 Table 2 is contained in the following table.

**Table 18: Line Number Designation (Informative)**

| Video Format | Applicable Standard | Coding Range, lines | Coded Lines | | |
|---|---|---|---|---|---|
| | | | First Field | Second Field | Frame |
| 480 Interlaced | SMPTE 125M [22] | 480 | 23 - 262 | 286 -525 | |
| 480 Progressive | SMPTE 293M [23] | 480 | | | 45 - 524 |
| 720 Progressive | SMPTE 296M [17] | 720 | | | 26 - 745 |
| 1080 Interlaced | SMPTE 274M [15] | 1088 | 21 - 560 | 584 - 1123 | |
| 1080 Progressive | SMPTE 274M [15] | 1088 | | | 42 - 1121 |

Note: The first two rows of this table are based on 720x483 SMPTE production formats. CEA-861[27] standardizes 720x480 video formats for consumer AVC receivers, using the same line number designation as the SMPTE standards but with 3 less active video lines at the bottom of the picture.

#### 8.2.3.1 Recommended Receiver Response to Bar Data

Receiving device designers are strongly encouraged to study Consumer Electronics Association (CEA) bulletin CEB16 [21], which contains recommendations regarding the processing of bar data.

### 8.2.4 Encoding and transport of active format description data

Active format description data, when present, shall be encoded and transported in accordance with Annex A of ATSC A/53 Part 4 [7]. Some of the text from A/53 Part 4 [7] is reproduced in this section for the convenience of the reader. Active Format Description (AFD) should be included in an SEI message whenever the rectangular picture area containing useful information does not extend to the full height or width of the coded frame. AFD data may also be included in user data when the rectangular picture area containing useful information extends to the full height and width of the coded frame.

When present, the AFD shall be carried within the SEI RBSP of the video Elementary Stream. For each SRAP Picture the default aspect ratio of the area of interest shall be set as signalled by the Supplemental Enhancement Information parameters. After introduction, an AFD shall remain in effect until the next SRAP or until another AFD value is introduced. Receivers should interpret the absence of AFD in a sequence start to mean the active format is the same as the coded frame, corresponding to AFD value '1000' (see Table 19).

> Note: The AFD syntax as shown here, starting with the afd_data of Table 19: Active Format Description Syntax for AVC video (which is the user_structure() of Table 12: Common Data Syntax ) is syntactically identical to that specified in ETSI TS 101 154 V1.8.1 [8], and is reprinted here with permission. Semantics are documented in Section 8.2.6 and some are intentionally different.

### 8.2.5 AFD Syntax

afd_data() shall be carried as specified in Section 8.1. Table 19 describes the syntax of the Active Format Description.

**Table 19: Active Format Description Syntax for AVC video**

| Syntax | No. of Bits | Format |
|---|---|---|
| afd_data() { | | |
| zero_bit | 1 | '0' |
| active_format_flag | 1 | bslbf |
| alignment_bits | 6 | '00 0001' |
| if (active_format_flag == '1') { | | |
| reserved | 4 | '1111' |
| active_format | 4 | bslbf |
| } | | |

### 8.2.6 AFD Semantics

active_format_flag – A 1 bit flag. A value of '1' indicates that an active format is described in this data structure.

active_format – A 4 bit field describing the area of interest in terms of its aspect ratio within the coded frame as defined in AVC [4]. Table 20 defines the coding of the active_format field that shall be used.

The active_format is used by the receiver in conjunction with picture size and shape information as indicated in the sequence parameter set RBSP and the VUI parameters. In particular, the picture width, picture height, frame cropping information, and sample aspect ratio are important for proper use of active_format. (see AVC [4].)

The combination of source aspect ratio and active_format allows the receiver to identify whether the area of interest is the whole of the frame (e.g., source aspect ratio 16:9, active_format 16:9 center), a letterbox within the frame (e.g., source aspect ratio 4:3, active_format 16:9 center), or a pillarbox within the frame (e.g., source aspect ratio 16:9, active_format 4:3 center).

**Table 20: Active Format**

| **active_format** | **Description** | |
|---|---|---|
| | **4:3 coded frames** | **16:9 coded frames** |

| | | |
|---|---|---|
| ‘0000’ | undefined (see below) | undefined (see below) |
| ‘0001’ | Reserved | Reserved |
| ‘0010’ – ‘0011’ | Not recommended | Not recommended |
| ‘0100’ | Aspect ratio greater than 16:9 (see below) | Aspect ratio greater than 16:9 (see below) |
| ‘0101’ – ‘0111’ | Reserved | Reserved |
| ‘1000’ | 4:3 full frame image | 16:9 full frame image |
| ‘1001’ | 4:3 full frame image | 4:3 pillarbox image |
| ‘1010’ | 16:9 letterbox image | 16:9 full frame image |
| ‘1011’ | 14:9 letterbox image | 14:9 pillarbox image |
| ‘1100’ | Reserved | Reserved |
| ‘1101’ | 4:3 full frame image, alternative 14:9 center | 4:3 pillarbox image, alternative 14:9 center |
| ‘1110’ | 16:9 letterbox image, alternative 14:9 center | 16:9 full frame image, alternative 14:9 center |
| ‘1111’ | 16:9 letterbox image, alternative 4:3 center | 16:9 full frame image, alternative 4:3 center |

AFD '0000' indicates that information is not available and is undefined. Unless bar data is available, DTV receivers and video equipment should interpret the active image area as being the same as that of the coded frame.

AFD ‘0000’, when accompanied by bar data, signals that the image’s aspect ratio is narrower than 16:9, but is not either 4:3 or 14:9. The bar data should be used to determine the extent of the image.

AFD ‘0100’, which should be accompanied by bar data, signals that the image’s aspect ratio is wider than 16:9, as is typically the case with widescreen features. The bar data should be used to determine the height of the image.

Use of ‘0010’ or ‘0011’ is not recommended in the SCTE television system. Values ‘0001’, ‘0101’ through ‘0111’ and ‘1100’ are reserved.

8.2.7 Recommended Receiver Response to AFD

Receiving device designers are strongly encouraged to study the Consumer Electronics Association (CEA) bulletin CEB16 [21], which contains recommendations regarding the processing of AFD. In several instances, a variety of design choices are possible when processing a given AFD value for display and the recommendation identifies one preferred method.

8.2.8 Relationship Between Bar Data and AFD (Informative)

Certain combinations of Active Format Description and bar data may be present in an SEI message (either, neither, or both). Note that AFD data may not always exactly match bar data because AFD only deals with 4:3, 14:9, and 16:9 aspect ratios while bar data may represent nearly any aspect ratio. When AFD and bar data are present together, AFD should be used in preference to bar data, except in the cases of AFD ‘0000’ and ‘0100’, where bar data should be used in concert with AFD as described above.

## 9.0 SUPPORT FOR AVC STILL PICTURES

AVC still pictures may be used in transport multiplex and when used shall be constrained as follows:

- The still picture coding shall comply with the definition in Section 2.1.5 of 13818-1 [3] . In addition, still picture applications should conform to the video coding constraints specified in tables 7, 8, 9A, 9B and 9C.
- Low_delay_hrd_flag (as defined in AVC [4]) may be either set to '0' or '1'. Still picture applications should follow the constraints specified in section 7.2.1.6 .
- A PES packet shall contain one and only one complete access unit with a still picture, which shall be aligned to the PES packet header. The PES packet header shall contain a coded PTS value.
- The time interval between successive still pictures shall be less than or equal to 60 seconds.
- PMT for this program element shall include the AVC_video_descriptor with the AVC_still_present_flag set to '1'. In addition, fixed_frame_rate_flag is set to '0' in the VUI (per Table 7).

# APPENDIX A AU_information in Adaptation Field Private Data

## A.1 Introduction

This Appendix contains optional information based on ETSI TS 101 154 V1.8.1 Annex D[8], which information, if present, shall meet the constraints herein.

## A.2 Requirements

The AU_information(), if present, shall be carried in the private data section within the adaptation field of a TS packet. A descriptor format is used to allow future extensions and additions to the information carried, and to allow other information to be carried in the adaptation field's private data.

The purpose of the AU_information() is to convey information about that access unit that is of use to applications, e.g., PVR applications. All the information provided in AU_information() should be considered "helper" information rather than definitive information. Thus, if there are any conflicts between AU_information() and the actual stream, then the information in the stream shall take precedence over the information in this descriptor. However, such a conflict should be considered an error condition and as such should not occur. The AU_information() when present, shall be in the TS packet that contains the start of each access unit of an AVC [4] video stream, except for the SRAP access unit, where it may be present either in the same TS packet or in the previous TS packet.

The format of AU_information() shall be as shown in Table 21.

**Table 21: AU_information data field**

| Syntax | No. of Bits | Mnemonic |
|---|---|---|
| AU_information () { | | |
| data_field_tag | 8 | uimsbf |
| data_field_length | 8 | uimsbf |
| if (data_field_length >= 1) { | | |
| AU_coding_format | 4 | uimsbf |
| AU_coding_type_information | 4 | bslbf |
| } | | |
| if (data_field_length >= 2) { | | |
| AU_ref_pic_idc | 2 | uimsbf |
| AU_pic_struct | 2 | bsblf |
| AU_PTS_present_flag | 1 | bslbf |
| AU_profile_info_present_flag | 1 | bslbf |
| AU_stream_info_present_flag | 1 | bslbf |
| AU_trick_mode_info_present_flag | 1 | bslbf |
| } | | |
| if (AU_PTS_present_flag == '1') { | | |
| AU_PTS_32 | 32 | uimsbf |
| } | | |

| Syntax | No. of Bits | Mnemonic |
|---|---|---|
| if (AU_stream_info_present_flag == '1') { | | |
| reserved_zero | 4 | '0000' |
| AU_frame_rate_code | 4 | uismbf |
| } | | |
| if (AU_profile_info_present_flag == ‘1’) { | | |
| AU_profile | 8 | uismbf |
| AU_constraint_set0_flag | 1 | bslbf |
| AU_constraint_set1_flag | 1 | bslbf |
| AU_constraint_set2_flag | 1 | bslbf |
| AU_AVC_compatible_flags | 5 | bslbf |
| AU_level | 8 | uismbf |
| } | | |
| if (AU_trick_mode_info_present_flag == ‘1’) { | | |
| AU_max_I_picture_size | 12 | uismbf |
| AU_nominal_I_period | 8 | uismbf |
| AU_max_I_period | 8 | uismbf |
| Reserved_zero | 4 | '0000' |
| } | | |
| if (*data parsed* < data_field_length) { | | |
| AU_Pulldown_info_present_flag | 1 | bslbf |
| reserved_zero | 6 | ‘00’ |
| AU_flags_extension_1 | 1 | bslbf |
| } | | |
| if (AU_Pulldown_info_present_flag == ‘1’) { | | |
| reserved | 4 | ‘000’ |
| AU_Pulldown_info | 4 | bslbf |
| } | | |
| for(i=0; i<n; i++) { | | |
| AU_reserved_byte | 8 | bslbf |
| } | | |
| } | | |

## A.3 Semantics

data_field_tag **-** this field shall have the value 0x02.

data_field_length - this field is the number of descriptor bytes following this length field. The values 0 and 1 may be used to signal short versions of the descriptor. The value 0 means that no fields after the data_field_length are sent. The value 1 means that only the fields AU_coding_format and AU_coding_type_information are present.

AU_coding_format - This field shall signal the coding format used by the elementary stream carried by this TS packet. The values are as show in Table 22. The value "Undefined" means that the coding type is not specified. If this field is not transmitted then the value "undefined" shall be inferred.

**Table 22: AU_coding_format values**

| Value | Stream Type |
|---|---|
| 0 | Undefined |
| 1 | Forbidden |
| 2 | AVC video stream as defined in AVC[4] |
| 3 | Used by DVB |
| 4-0xF | reserved |

AU_coding_type_information - indicates the elementary stream slice types that are present in the immediately following access unit. For AVC video [4], this field shall be interpreted as a four bit field with the syntax shown in Table 23. If this field is not transmitted, then no information about the coding type may be inferred (i.e., any slice type may be present).

The value '0' in this field means "undefined". This shall only be used where the value is to be supplied by a subsequent stage in delivery or processing.

**Table 23: AU_coding_type_information for AVC video**

| Syntax | No. of Bits | Mnemonic |
|---|---|---|
| AU_IDR_slice_present_flag | 1 | bslbf |
| AU_I_slice_present_flag | 1 | bslbf |
| AU_P_slice_present_flag | 1 | bslbf |
| AU_B_slice_present_flag | 1 | bslbf |

AU_ref_pic_idc - This field indicates if any part of the access unit is required in the reconstruction of other access units. The value '00' means that it is not used by other access units. In the case of AVC [4], the value shall be the nal_ref_idc field in the NAL header used for the access unit. If this field is absent, then inferences shall not be made about whether or not the AU is used as a reference to other AUs.

AU_pic_struct - This field shall be set to '01' if the access unit is a top field picture, '10' if it is a bottom field. Otherwise, it shall be set to '00'. '11' value is reserved. If this field is absent, inferences shall not may be made about the picture structure (i.e., it may be any format).

Note: AU_Pulldown_info may provide additional information in this area.

AU_PTS_present_flag - This field shall be set to '1' when the AU_PTS_32 value is present in the descriptor, otherwise it shall be set to the value '0'. If this field is not present, the value '0' shall be inferred.

AU_profile_info_present_flag - This field shall be set to '1' when the AU_profile_idc and AU_level_idc values are present in the descriptor, otherwise it shall be set to the value '0'. If this field is not present, the value '0' shall be inferred.

AU_stream_info_present_flag - This field shall be set to '1' when the AU_frame_rate_code value is present in the descriptor, otherwise it shall be set to the value '0'. If this field is not present, the value ‘0’ shall be inferred.

AU_trick_mode_info_present_flag - This field shall be set to '1' when the AU_max_I_picture_size and AU_max_I_period are present in the descriptor, otherwise it shall be set to the value '0'. If this field is not present, the value ‘0’ shall be inferred.

AU_PTS_32 - the 32 most significant bits of the 33-bit PTS that applies to the access unit to which this descriptor applies.

AU_frame_rate_code - this field indicates the video frame rate in the stream carried by packets with the current PID. In the case of video, this shall be encoded as in Table 6-4, Section 6.3.3 of ISO/IEC 13818-2[10]. The values in this table are informatively replicated on Table 24 below. If the field is absent inferences shall not be made at this layer about the frame rate.

**Table 24: Informative Frame Rate values taken from table 6-4 of 13818-2[10]**

| AU_frame_rate_code | **Corresponding Frame Rate (Hz)** |
|---|---|
| 0 | Forbidden |
| 1 | 24/1.001 |
| 2 | 24 |
| 3 | 25 |
| 4 | 30/1.001 |
| 5 | 30 |
| 6 | 50 |
| 7 | 60/1.001 |
| 8 | 60 |
| 9 to 0xF | Reserved |

AU_profile - this field conveys the profile to which the access unit conforms. For AVC video [4] this carries the profile_idc value as defined in AVC [4].

The value ‘0’ in this field means “undefined”. This shall only be used where the value is to be supplied by a subsequent stage in delivery or processing.

Constraint_set0_flag, constraint_set1_flag, constraints_set2_flag, AVC compatible flags **-** These fields carry the same semantics as the fields of the same name in the AVC_video_descriptor() in Section 2.6.65 of 13818-1[3], which in turn have semantics defined in AVC[4], section 7.4.2.1. Note that with High profile, the first bit in AVC_compatible_flags carries constraint_set3_flag.

AU_level **-** this field conveys the level to which the access unit conforms. For AVC video [4] this carries the level_idc value as defined AVC, Annex A [4].

The value ‘0’ in this field means “undefined”. This shall only be used where the value is to be supplied by a subsequent stage in delivery or processing.

AU_max_I_picture_size **-** this field conveys the maximum intra picture size, in units of 16x1024 bits, which may be found in the current bitstream. This value, according to profile and level, shall comply with AVC [4] limits. The value 0 shall be forbidden.

AU_nominal_I_period - this field conveys the nominal distance between two consecutive I or IDR pictures, counted in frame pictures. The value 0 shall be forbidden.

AU_max_I_period - this field conveys the maximum distance that may be found in the stream between two consecutive I/IDR pictures, counted in frame pictures. The value 0 shall be forbidden.

AU_Pulldown_info_present_flag - This flag indicates if information about 3:2 pulldown for this AU is present

If this field is not present, then the value ‘0’ should be inferred.

AU_Pulldown_info - This field contains the five bits carried in the AVC structure signalling the AU’s display characteristics, specifically the pic_struct field of the SEI message pic_timing. The default value for this field shall be as shown in  Table 25 below.

**Table 25: AU_Pulldown_info default values**

| AU_pic_struct | AU_Pulldown_info default value |
|---|---|
| 00 | 0 (progressive-scan formats) |
| 00 | 3 (interlaced-scan formats) |
| 01 | 1 |
| 10 | 2 |
| 11 | reserved |

# APPENDIX B Encoding Guidelines to Enable Trick Play Support of AVC Streams (Informative)

## B.1 Introduction

### B.1.1 Overview

This appendix discusses informative guidelines on the encoding of AVC bitstreams to enable support of trick play modes. MPEG-2 personal video recording devices are increasingly being used in the marketplace and it is reasonable to expect this trend to continue. It is important to recognize that the unofficial widely-adopted methods of MPEG-2 encoding directly enabled many of the techniques currently used to achieve trick mode functionality. Note that MPEG-2 video may be encoded in a manner that makes PVR very difficult but since most encoders encoded bitstreams in a "PVR-friendly" manner, this was not an issue with MPEG-2 bitstreams. Currently, the lack of syntax and semantics constraints on AVC bitstreams combined with the rich set of video coding tools in AVC allows for a wide variety of potential bitstreams with some being very problematic for any type of sophisticated bitstream manipulation such as the trick modes in AVC PVR implementations. For these reasons, the guidelines in this appendix were constructed to assist encoders to create AVC bitstreams that are "PVR-friendly". Note that this appendix is informative since it is understood that enabling trick play support is an optional feature that may or may not be appropriate depending on its intended use.

### B.1.2 Technical Requirements

One class of trick play modes consists of the desire to play back the video at a speed that is a multiple of real-time playback. Let a Nx trick play mode (where N is a positive number greater than 1) represent video playback at a speed of N times real-time playback. For example, a 3x trick play mode may be desired which would allow a user to fast forward through a program three times as fast as normal playback, i.e., in one-third the time. It is often desired for these trick modes to be relatively "smooth", i.e., an Nx trick mode (where N is an positive integer) requires (at least approximately) every Nth picture in the bitstream to be displayed. For example, repeating every thirtieth picture ten times would not constitute a "smooth" 3x trick mode using this definition. This "smooth" requirement need not be required for very fast trick modes like 15x or 30x fast forward since the human visual system is unable to process such rapid motion. However, this requirement may be desirable for trick modes such as 2x and 3x fast forward to obtain the satisfactory visual appearance of moving objects during the trick play.

In general, without any encoding constraints, the minimum requirement to implement trick modes is for the decoding to be done at the same speed as the desired trick mode to ensure that every prediction region is available for use in the motion compensation process, e.g., a decoder that runs at three times the normal speed of decoding is needed to guarantee 3x fast forward functionality. Note that this is a significant increase from the minimum requirement needed for normal playback. This approach has been done before for trick play with MPEG-2 standard definition content but is not practical or cost effective for many current and future applications. For example, decoding HD AVC video at three times the normal decoding speed is currently not possible in a cost-efficient fashion and even if this increased capability were made available in the future, it may not be desirable because of the increased cost relative to the minimum requirement for normal playback. This leads to a key technical assumption for the cost-effective implementation of trick play modes:

- Encoding intended for trick-play will be done in such a way that it does not burden decoders to decode pictures at a rate faster than normal playback to implement a trick play mode.

## B.2 Discardable Pictures

Many PVR implementations drop pictures in the bitstream (i.e., skip over and do not present these pictures to the decoder) to circumvent the need to decode bitstreams at speeds that are a multiple of real-time decoding. The visual effect of decoding at a multiple of real-time decoding may then be achieved using a normal decoder. This is only possible if all the dropped pictures are not needed as reference frames for the pictures that are to be displayed. Pictures that can be dropped without affecting the decoding of other pictures are termed “discardable” pictures. The following sections will discuss how the “discardable” pictures concept was exploited in MPEG-2 trick play implementations and then how this same concept may be used to implement AVC trick play.

### B.2.1 MPEG-2 Discardable Pictures

In the MPEG-2 video standard, B-pictures are not allowed to be used as reference pictures for motion compensation. This has a significant benefit for trick play modes since any B-pictures in a MPEG-2 bitstream may be dropped without affecting the decodability of other pictures. The “discardability” property of B-pictures is commonly used by many MPEG-2 trick mode implementations.

Figure 2 illustrates the unofficial but widely-adopted MPEG-2 GOP structure, the IBBP GOP structure, which has two B-pictures placed between every pair of anchor I- and/or P-pictures. By dropping the B-pictures in this type of stream and passing the remaining pictures to the decoder, the visual effect of 3x fast forward trick play may be implemented with a decoder running at normal playback speed.

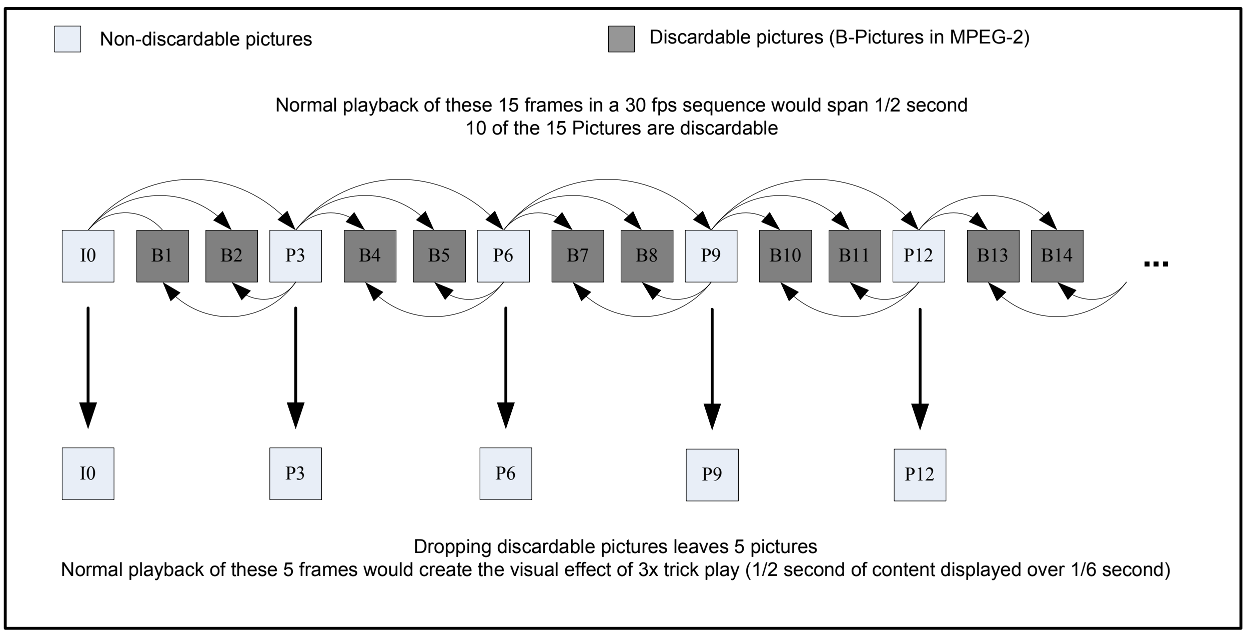


**Figure 2: Example of achieving a 3x trickplay mode from a common MPEG-2 GOP structure (IBBP)**

Figure 3 illustrates a MPEG-2 GOP structure, the IPPP GOP structure, where no B pictures are placed between every pair of anchor I- and/or P- pictures. Note that this structure is compliant to MPEG-2 but the technique of dropping B-pictures described above does not create a 3x trick play mode with this MPEG-2 coding structure since there are not enough B-pictures to drop (there is only one discardable picture at the end of the GOP). In this case, a decoder that is able to run at N times normal decoding speed is necessary to support N times fast forward trick play since every picture is dependent on the previous picture in the GOP.

Note that the problematic effect on PVR of a bitstream with a coding structure as shown in Figure 3 has often been overlooked and not usually an issue because this type of MPEG-2 GOP structure is rarely used in broadcast applications.

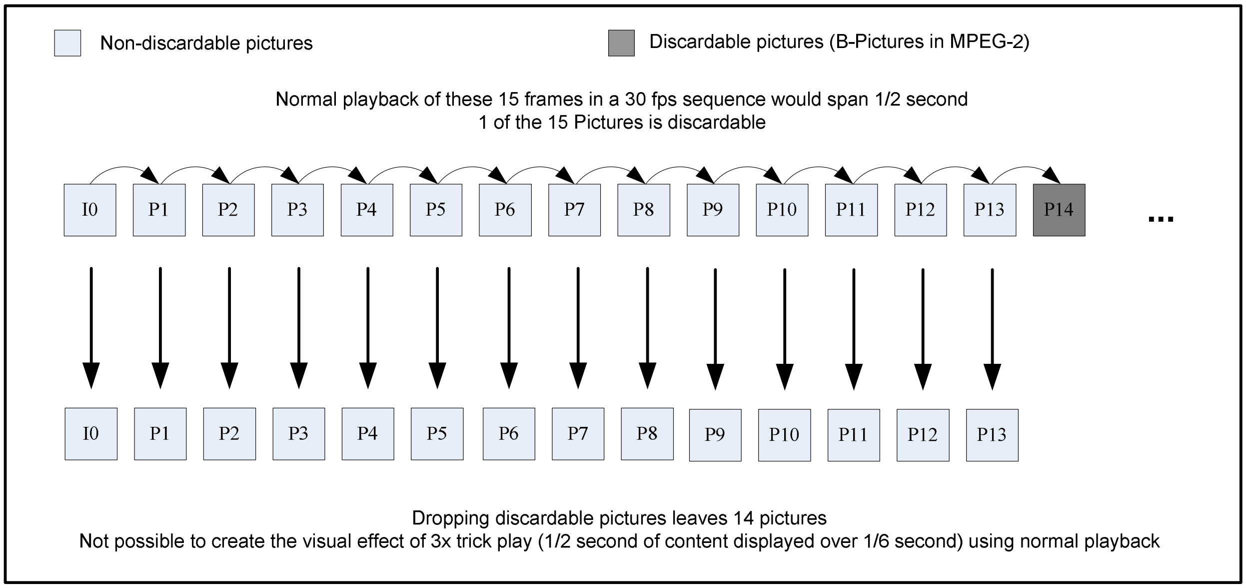


**Figure 3: Example of a compliant MPEG-2 GOP structure (IPPP) that is unable to achieve 3x trick play by discarding pictures**

### B.2.2 AVC Discardable Pictures

The AVC compression standard has some substantial differences compared to MPEG-2 that significantly affect the picture coding structure and complicate trick mode implementations. These include the fact that B-pictures may be used as reference pictures for prediction, i.e., not all B-pictures are discardable as in MPEG-2. Note that the discardability of pictures is specifically indicated in the AVC standard by the nal_ref_idc flag in the NAL header (nal_ref_idc = 0 indicates a discardable picture). Therefore, for AVC bitstreams, the important factor in trick mode functionality is the location of discardable pictures, not the location of B-pictures as in MPEG-2. The presence of discardable pictures determines the feasibility of dropping pictures that are not needed for display to achieve the visual effect of a trick play mode.

### B.2.3 Discardable Pictures and Trick Play Speeds

The percentage of pictures in the bitstream that are discardable determines the maximum trick play speed that could be achieved by just dropping discardable pictures while operating the decoder at normal processing speeds. The formula below may be used to associate the percentage of discardable pictures with the maximum trick play speed that could be achieved by dropping discardable pictures:

Trick Play Speed = 100/ (100 – X) where X is the percentage of discardable pictures

Examples using common ratios of discardable pictures are listed in Table 26:

**Table 26: Discardable Picture Percentages and Maximum Achievable Trick Play Speeds by discard process**

| Percentage of Discardable Pictures | Maximum Trick Play Speed Achievable By Dropping Pictures |
|---|---|
| 16% (1/6 of the pictures) | 1.2x |
| 20% (1/5 of the pictures) | 1.25x |
| 25% (1/4 of the pictures) | 1.33x |
| 33% (1/3 of the pictures) | 1.5x |
| 50% (1/2 of the pictures) | 2x |
| 66% (2/3 of the pictures) | 3x |
| 75% (3/4 of the pictures) | 4x |

Note: Trick play speeds slower than the maximum achievable by dropping pictures may always be created by choosing to display some of the discardable pictures.

### B.2.4 Smooth Trick Play and Compression Efficiency

Constraining a certain percentage of pictures in the bitstream to be discardable is necessary to enable the technique of dropping discardable pictures to achieve a trick play mode. However, it is important to recognize that determining the interval period between pictures where this percentage is constrained has a tradeoff between whether a smooth trick play is achieved and the coding structure which is able to impact coding efficiency. For example, Figure 4 and Figure 5 both illustrate coding structures with 66% of its pictures as discardable pictures (in both cases 10 of the 15 total pictures are discarded).

Figure 4 has a more regular discardable picture structure and represents the further requirement of 2 out of every 3 pictures to be discardable. Dropping the discardable pictures in Figure 4 may result in smooth 3x playback since every third picture in the original stream remains. However, note that the tradeoff for the ability to create a smooth 3x trick play is that the discardable picture structure places a tight constraint on the encoding which could reduce compression efficiency

Ten out of the 15 total pictures in Figure 5 are discardable as in Figure 4, but its discardable picture structure is not as regular. Dropping the discardable pictures in Figure 5 might not result in a smooth trick play experience as in Figure 4. However, note that dropping discardable pictures may still be used to achieve the visual effect of playing through the content at three times the speed (since 5 frames remain) but without the serious constraint on the encoding.

Note: Although structure may not always guarantee smooth playback, there are methods that could create an appearance of smoother playback by means outside of this appendix.

To enable trick play support and still facilitate maximum compression efficiency, the percentage of discardable pictures should be calculated over the length of a SGOP (which, at the maximum 1 second time interval between the Decoding Time Stamps of two successive SRAP pictures, may be up to 60 pictures). Encoding for the smoothest trick-play should distribute discardable pictures evenly in time throughout the SGOP.

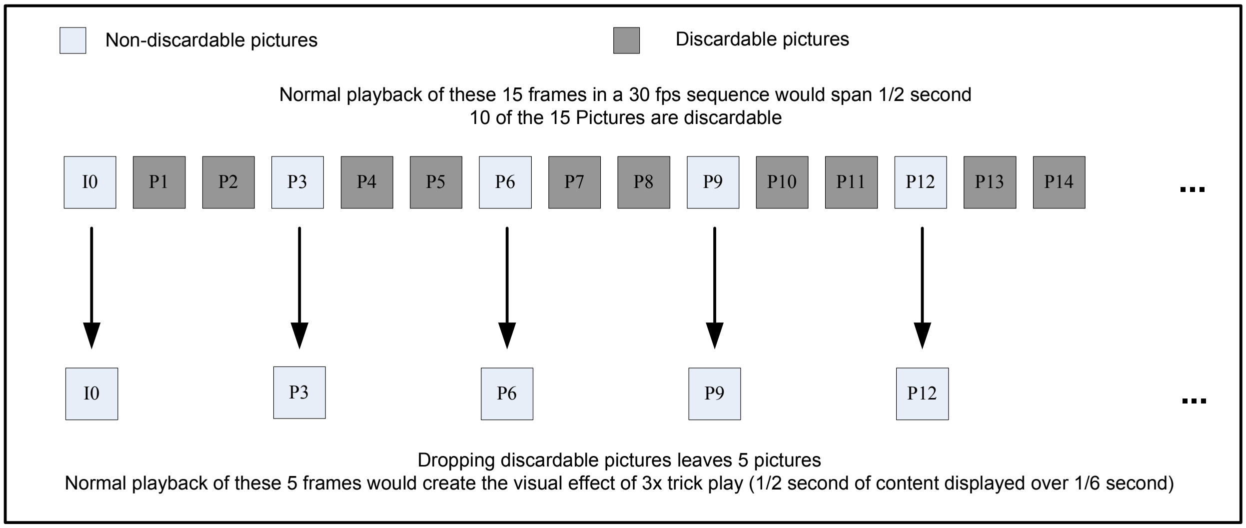


**Figure 4: Coding Structure with 2 Out of Every 3 Pictures as Discardable Pictures (The Discardable Pictures Are Inserted Consistently)**

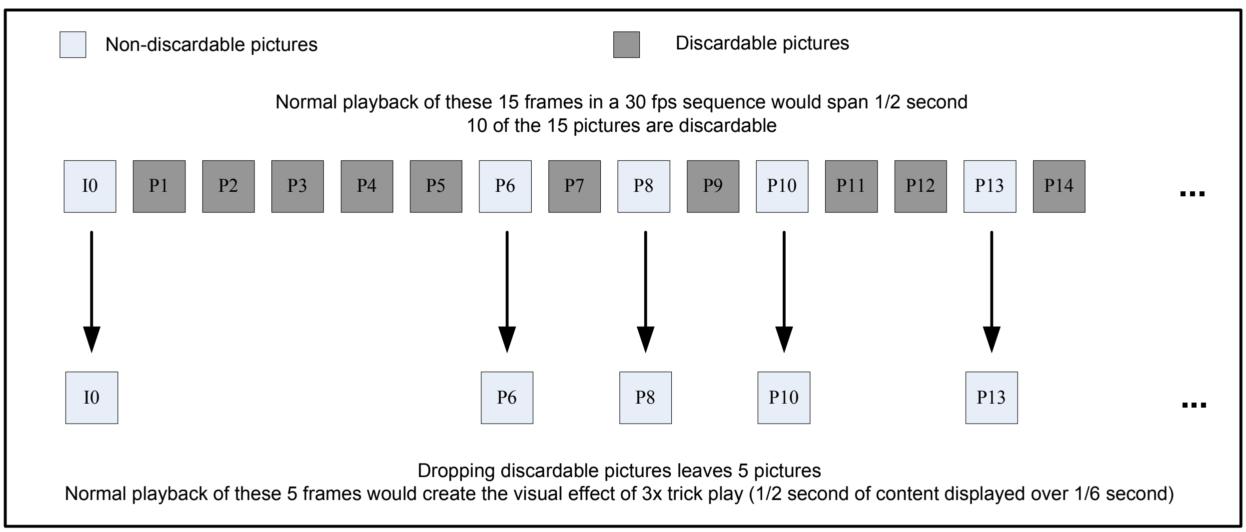


**Figure 5: Coding Structure with 10 out of Every 15 Pictures as Discardable Pictures (The Discardable Pictures Are Not Inserted Consistently)**