PROPOSED SMPTE STANDARD

for Television — Transporting MPEG-2 Recoding Information through 4:2:2 Component Digital Interfaces

Contents

1 Scope

This standard specifies an embedded transport mechanism for the MPEG-2 recoding data set as defined in SMPTE 327M for the representation of MPEG-2 recoding information in ITU-R BT.656, 4:2:2 component digital interfaces.

The recoding data set is derived from an ISO/IEC 13818-1 and 2 compliant MPEG bit stream during the decoding process, as described in ISO/IEC 13818-1 and -2.

For the minimum operation of this standard, the MPEG-2 recoding data set shall be spatially and temporally aligned to each decoded macroblock mapped into an ITU-R BT.656 interface.

This standard specifies the spatially and temporally aligned transport of the MPEG-2 recoding data set within the active picture area on ITU-R BT.656 interfaces for equipment that complies with ISO/IEC 13818-1 and -2, including 422P@ML and MP@ML for both the 625/50 and 525/60 video standards.

2 2 Normative references

The following standards 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 agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 327M, Television — MPEG-2 Recoding Data Set

ITU-R BT.601-5 (1994), Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios

ITU-R BT.656-4, Interfaces for Digital Component Video Signals in 525-Line and 625-Line Television Systems Operating at the 4:2:2 Level of Recommendation ITU-R BT.601 [Part A]

ITU-T H.222.0 with amend 1-2, ISO/IEC 13818-1:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

ITU-T H.262 with amend 1/corr 1 and amend 2/corr 2, ISO/IEC 13818-2:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

3 General

The principal application of this standard is to preserve the quality of the video signal when cascading

MPEG-2 decoders and coders for editing or transcoding purposes by feeding forward previous coding decisions.

The transport mechanism for the MPEG-2 recoding data set permits the simultaneous processing of both the video and the MPEG-2 recoding data set and, consequently, of the MPEG bit stream. This allows lossless cascading, frame accurate editing, and logo/caption insertion to be performed.

The information contained in the MPEG-2 recoding data set is defined in SMPTE 327M.

This recoding information shall be temporally locked to the decoded (or partially decoded) video to the nearest MPEG-2 frame or field depending on the picture structure of the coded MPEG-2 bit stream. It shall also be spatially locked with the decoded video to the nearest MPEG-2 macroblock within the decoded frame/field.

To accrue the full benefits of the recoding information when cascading via a 4:2:2 component digital interface, the following recommendations shall be adhered to:

- The transportation mechanism shall preserve at least the 8 most significant bits of active video. The mechanism outlined uses the least significant bit of each 10-bit chrominance sample to transmit the data through the ITU-R BT.656 interface.
- It is also necessary for the recoding information to be aligned with the decoded MPEG macroblocks in the decoded pictures, both spatially and temporally. This standard is based on producing an ITU-R BT.656 output and will cover MPEG bit streams up to and including 422P@ML and MP@ML for both 625/50 and 525/60 systems.

NOTE – Users should be aware that some existing equipment using ITU-R BT.656 interfaces alters or uses the tenth bit of the video samples for other (nonstandardized) purposes. Where this occurs, the recoder will be unable to benefit from the MPEG-2 recoding information.

4 Definitions

4.1 aligned: [Applied usually to recoding information] arranged so that information relating to a macroblock (or other region) is embedded within that region of the decoded signal.

4.2 bit stream: An ordered series of bits conforming to ISO/IEC 13818-2.

4.3 bottom field: One of two fields that comprise a frame. Each line of a bottom field is spatially located immediately below the corresponding line of the top field.

4.4 cascading: The process where video that has once been coded (compressed) is subsequently decoded and coded once more. This cascaded step could carry on for any number of generations.

4.5 chrominance: The color-difference samples defined in ITU-R BT.601.

4.6 coding: The process by which an uncompressed video sequence is compressed to a bit stream that conforms to ISO/IEC 13818-2.

4.7 column: A vertical column of macroblocks spanning the full height of the decoded picture (columns are numbered from left to right starting at zero).

4.8 CRC: Cyclic Redundancy Check. A class of error detecting codes used in this standard to detect errors when the recoding information and the decoded pictures are being edited or processed in an ITU-R BT.601 stream by external equipment.

4.9 decoded video, decoded picture: Output video generated by an ISO/IEC 13818-2 compliant decoder that complies with the ITU-R BT.656 interface standard.

4.10 decoder: A compressed bit stream decoder that complies with ISO/IEC 13818-2.

4.11 DTS: Decoder Time Stamp (see ISO/IEC 13818-1).

4.12 embedded: [Applied usually to recoding information] conveyed within a digital video signal so as to be capable of being passed through digital video equipment.

4.13 macroblock: Defined in ISO/IEC 13818-2 as a block of 16 × 16 luminance pixels.

4.14 macroblock rate information: This corresponds to the coding information from the ISO/IEC 13818-2 bit stream that relates to the individual macroblocks as defined in this standard.

4.15 MPEG profile/level: As defined in ISO/IEC 13818-2.

4.16 MPEG-2 recoding information: This comprises the elements defined in SMPTE 327M and the additional information described in this standard required for the practical transport and use of the MPEG-2 recoding data set.

4.17 picture: As defined in ISO/IEC 13818-2.

4.18 picture rate information: This corresponds to the coding information from the ISO/IEC 13818-2 bit stream that relates to the whole picture as defined in this standard.

4.19 PTS: Presentation Time Stamp (see ISO/IEC 13818-1).

4.20 recoding data set: The set of information defined in SMPTE 327M.

4.21 stripe: A horizontal row of macroblocks spanning the full width of the decoded picture (stripes are numbered from top to bottom starting at zero).

4.22 sufficient: [Applied usually to recoding data set] containing the necessary information to enable transparent recoding (in a mathematical sense) of the video signal.

4.23 top field: One of two fields that comprise a frame. Each line of a top field is spatially located immediately above the corresponding line of the bottom field.

4.24 transcoding: A conversion within the MPEG-2 stream domain, such as bit rate changing or changing the group of pictures (GOP) structure.

4.25 video: A signal conforming to ITU-R BT.601 in this standard.

5 MPEG-2 recoding information

MPEG-2 recoding information comprises:

– The elements defined in SMPTE 327M.

– Additional elements from both macroblock and picture rate information as described in this standard.

These additional elements are required to allow the video and the MPEG-2 recoding information to be edited in parallel using conventional ITU-R BT.601 based equipment.

The MPEG-2 recoding information is subsequently aligned within the macroblock as described in figure 1.

6 Macroblock rate information

The contents of the MPEG-2 recoding data set relating to each macroblock will be spatially as well as temporally aligned with the decoded video pixels relating to that macroblock.

The elements in table 1 are at picture rate in the MPEG-2 stream, but are inserted into MPEG-2 recoding information at both picture and macroblock level to create the alignment required to allow editing of the resultant ITU-R BT.601 signal.

The elements in table 2 are required in the transport mechanism to synchronize the video information and the MPEG-2 recoding information.

Where reference is made to elements in ISO/IEC 13818-2, these are indicated in the text in italics.

7 MPEG-2 recoding information transport mechanism

This comprises 256 bits of data per macroblock that shall be placed bit by bit onto the least significant bit of each 10-bit chrominance sample in the decoded video.

The decoded 4:2:2 component video macroblock corresponds to a matrix of 16 × 16 luminance components and two matrices of 8 × 16 color-difference components. Hence, in total there are 256 pixels of luminance and 256 pixels of chrominance in each macroblock. In ITU-R BT.656 video, the multiplexed order for these pixels is C_b,Y,C_r,Y, etc.

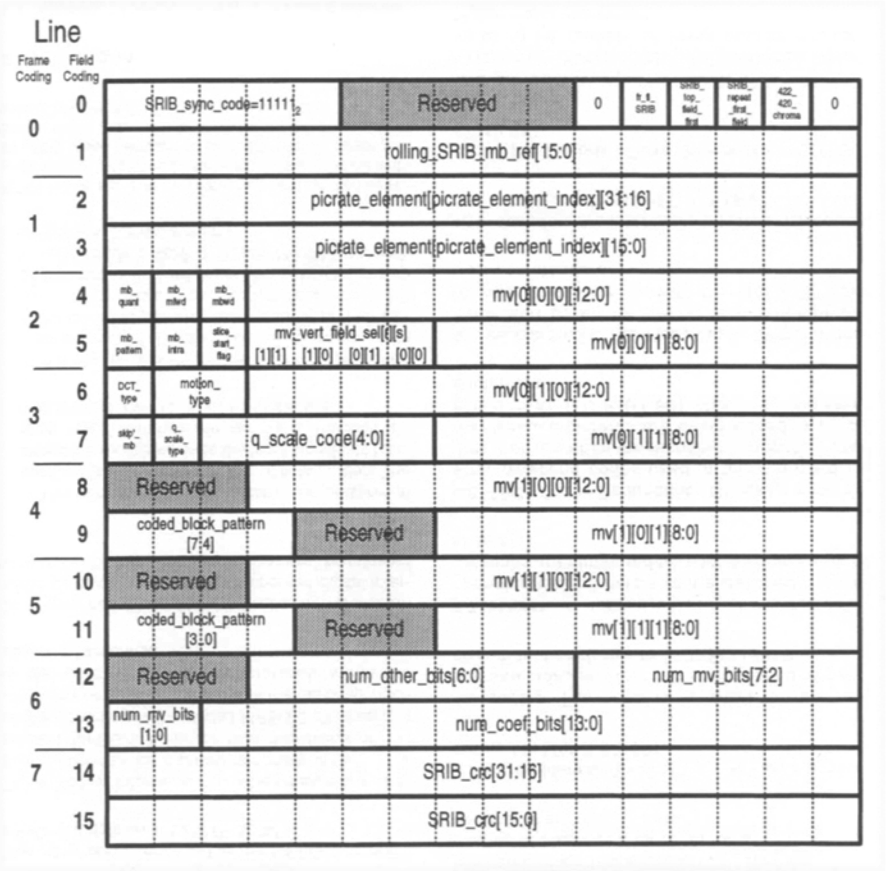

Figure 1 – Macroblock format of MPEG-2 recoding information

Table 1 – Additional macroblock rate information

Parameter	No. of bits	Definition
top_field_first	1-bit flag	Set to a value equal to *top_field_first* held in the original bit stream and also indicates, along with repeat_first_field, the temporal alignment of the recoding information with its associated video.
repeat_first_field	1-bit flag	Set to a value equal to *repeat_first_field* held in the original bit stream. The contents of the recoding information of the first field must also be repeated, as indicated by this flag.
422_420_chroma	1-bit flag	A value of 0 indicates the bit stream was 4:2:0 and chrominance up-sampling was performed to output 4:2:2 video. A value of 0 indicates the bit stream was 4:2:2 and no chrominance filtering was performed.
picrate_element picrate_element[31:16] picrate_element[15:0]	 16-bit ui 16-bit ui	Represents part of the picture rate information for the present picture and has its content dispersed within the decoded picture. This is defined in further detail in clause 8.
q_scale_type	1-bit flag	Set to a value equal to *q_scale_type* held in the original bit stream.
reserved		These blocks have no meaning and all the bits shall be set to 0.

Table 2 – Additional transport elements

Parameter	No. of bits	Definition
srib_sync_code	5-bit flag	This is a fixed bit-string, 11111, which shall be used to indicate the left alignment of the first row of each macroblock.
fr_fl_srib	1-bit flag	This flag shall be set to the value 1 when *picture_structure* is set to frame picture structure (value 11) and indicates that the MPEG-2 recoding information is distributed over 16 frame lines. If *picture_structure* is not set to frame (value 11), then the flag is set to 0 and the recoding information is distributed over 16 field lines. This mechanism ensures that the recoding information remains spatially and temporally locked to the corresponding pixels in the decoded video frames/fields. The distribution of the recoding information for frame pictures and field pictures is further shown in figures 2 and 3.
rolling_srib_mb_ref	16-bit ui	This is a 16-bit modulo 65521 rolling reference value (see note). This rolling reference increments on every macroblock of the transmitted picture. The count shall be continuous across transmitted picture boundaries. This value shall be initialized at start-up to a number selectable between 0 and 65520 inclusive. This is to allow systems of decoders to be built with unique recoding information identifiers.
srib_crc srib_crc[31:16] srib_crc[15:0]	 16-bit ui 16-bit ui	A 32-bit CRC to allow error detection for each macroblock of the MPEG-2 recoding information. A model for the operation of this 32-bit CRC is defined in annex A of ISO/IEC 13818-1. The CRC is calculated on the 224 data bits of the macroblock of MPEG-2 recoding information taken in raster order, using the following generator polynomial: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

NOTE – The largest 16-bit derived prime number was chosen to ensure that there is a minimum coherence between the position of the macroblock and its equivalent address in subsequent pictures.

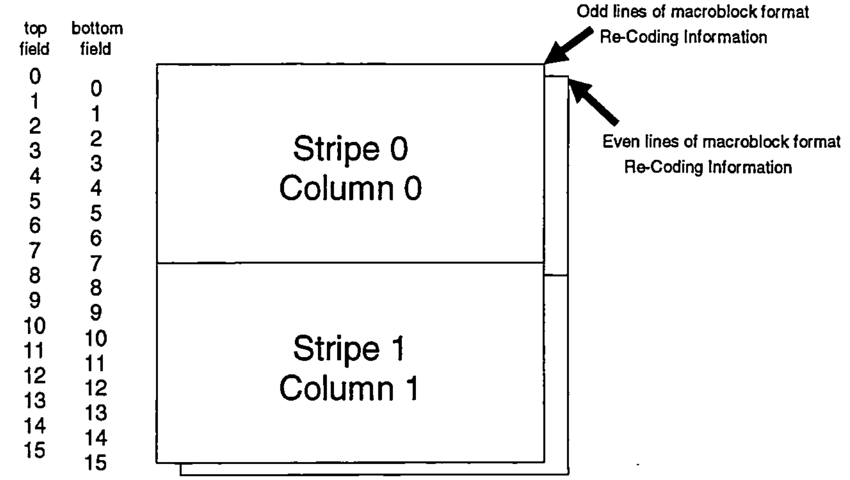

Figure 2 – Recoding information in frame coded pictures (fr_fl_srib=1)

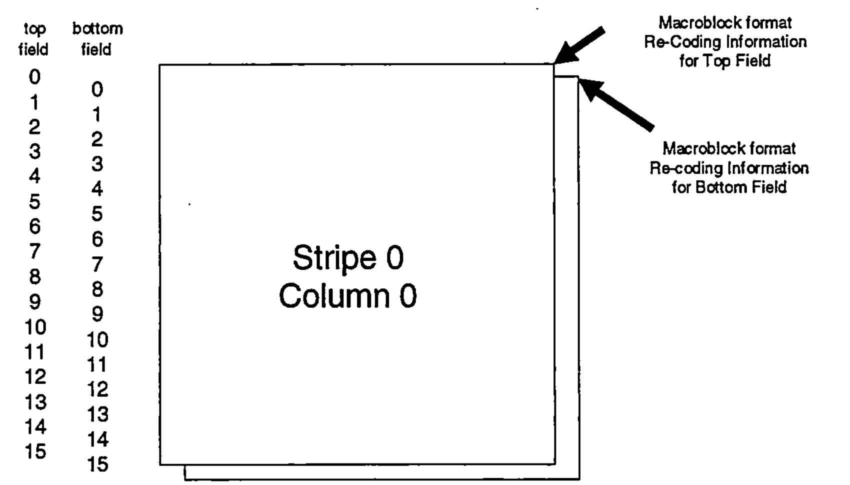

Figure 3 – Recoding information in field coded pictures (fr_fl_srib=0)

To increase the resilience of the detection of edits in the macroblock formatted recoding information, a form of parity scrambling is applied to the data before being placed in the least significant bit of the 10-bit chrominance samples.

The parity scrambling scheme used is shown in figure 4, where the bits represent the bits of 10-bit video (i.e., bits 2 through 9 correspond to the 8-bit video).

Consider the first few video samples in a stream as being C_b0, Y0, C_r0, Y1, C_b1, Y2, C_r1, Y3, etc. The parity for scrambling the macroblock formatted recoding information inserted into each chrominance sample is derived by combining the parity of the chrominance sample with the parity of the subsequent luminance sample in the video stream.

For example, the parity used to scramble the macroblock formatted recoding information for sample C_b0 above is the combined parity of C_b0 and Y0, the scrambling of data for C_r0 is taken from the combined parity of C_r0 and Y1, etc.

A detailed example of the macroblock formatted recoding information bits encoded into the ITU-R BT.656 samples for the first four luminance and associated chrominance samples is shown in table 3.

8 Picture rate information

This clause defines the picture rate information required in the recoding information. Table 4 shows the elements which must be included.

The picture rate information must all pass without error in order to achieve the best performance when recoding. Therefore, the picture rate information must be sent in a reasonably rugged way and it is for this reason that several copies are sent in the macroblock formatted recoding information distributed around the picture in the manner which will now be discussed.

In order to carry the full picture rate information, 4320 bits are required. Each macroblock of recoding information shall carry a 32-bit element of this picture rate information which will therefore require a total of 135 macroblocks in order to carry the full picture rate information. The number of copies which will then be distributed for the main picture types and video formats are given in table 5.

Other formats also exist, for example, in 422P@ML, the formats 720 x 608 pels for 625/50 frame and 720 x 512 pels for 525/60 frame. For these and other picture formats where a noninteger number of copies are available, incomplete copies will exist in the lower stripes of each picture.

The content of the picture rate information is described in SMPTE 327M and in more detail as regards the transport mechanism in table 4.

The distribution of the picture rate information must be such that common video processes such as caption or logo insertion permit the full extraction of the picture rate information. The equation given below gives the optimal distribution of picture rate elements in each macroblock of the MPEG-2 recoding information, regardless of the picture type and video format used, given the stripe and column address of that macroblock:

picrate_element_index = [(stripe % 3) * 45 + column + (27 * (stripe / 3))] % 135

The mathematical symbols are as defined in ISO/IEC 13818-2.

8.1 Picture rate elements

Table 4 shows a listing of the elements taken from the bit stream which will be included in the MPEG-2 recoding information.

For each element of the picture rate information, a category can be defined according to the way the information should be conveyed within the MPEG-2 recoding information. These categories are defined and numbered in the list below, with a definition of how this information is conveyed:

1) Represents values that are taken directly from the bit stream when present in the current picture, otherwise these values are undefined.

2) Represents values which might not be present in the bit stream for each picture, but which are required in each picture where MPEG-2 recoding information is embedded. These elements must be repeated in each picture based on previous values encountered in the bit stream.

3) Represents values that are not directly present in the bit stream, but must be derived for every picture where the recoding information is to be embedded.

The category of each picture rate information element is given for each element in table 4.

The 32-bit protection CRC to allow error detection for the picture rate information is defined by the model in annex A of ISO/IEC 13818-1. The CRC is calculated on the 4288 data bits of the embedded MPEG-2 recoding picture rate information taken in the order given in table 4.

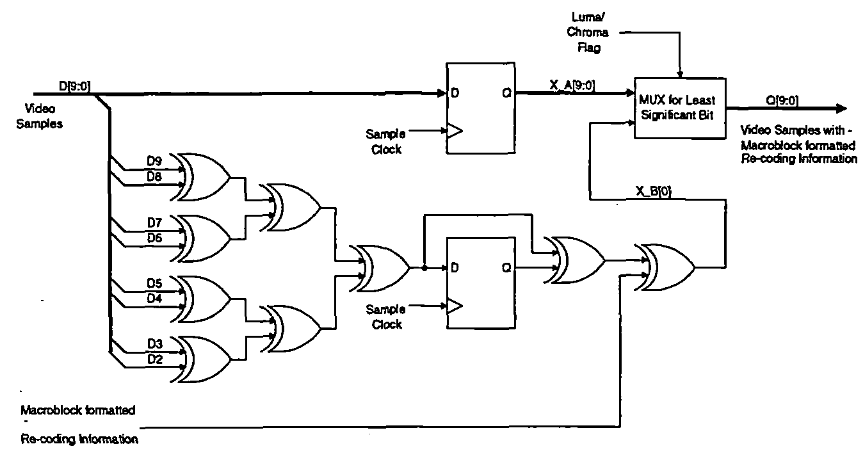

Figure 4 – Embedding of the macroblock formatted recoding information

Table 3 – Layout of the macroblock formatted recoding information within the video data

D9	$C_b[0][9]$	Y[0][9]	$C_r[0][9]$	Y[1][9]	$C_b[1][9]$	Y[2][9]	$C_r[1][9]$	Y[3][9]
D8	$C_b[0][8]$	Y[0][8]	$C_r[0][8]$	Y[1][8]	$C_b[1][8]$	Y[2][8]	$C_r[1][8]$	Y[3][8]
D7	$C_b[0][7]$	Y[0][7]	$C_r[0][7]$	Y[1][7]	$C_b[1][7]$	Y[2][7]	$C_r[1][7]$	Y[3][7]
D6	$C_b[0][6]$	Y[0][6]	$C_r[0][6]$	Y[1][6]	$C_b[1][6]$	Y[2][6]	$C_r[1][6]$	Y[3][6]
D5	$C_b[0][5]$	Y[0][5]	$C_r[0][5]$	Y[1][5]	$C_b[1][5]$	Y[2][5]	$C_r[1][5]$	Y[3][5]
D4	$C_b[0][4]$	Y[0][4]	$C_r[0][4]$	Y[1][4]	$C_b[1][4]$	Y[2][4]	$C_r[1][4]$	Y[3][4]
D3	$C_b[0][3]$	Y[0][3]	$C_r[0][3]$	Y[1][3]	$C_b[1][3]$	Y[2][3]	$C_r[1][3]$	Y[3][3]
D2	$C_b[0][2]$	Y[0][2]	$C_r[0][2]$	Y[1][2]	$C_b[1][2]$	Y[2][2]	$C_r[1][2]$	Y[3][2]
D1	$C_b[0][1]$	Y[0][1]	$C_r[0][1]$	Y[1][1]	$C_b[1][1]$	Y[2][1]	$C_r[1][1]$	Y[3][1]
D0	Macroblock formatted recoding information	Y[0][0]	Macroblock formatted recoding information	Y[1][0]	Macroblock formatted recoding information	Y[2][0]	Macroblock formatted recoding information	Y[3][0]

Table 4 – Picture rate elements

Parameter	Number format	No. of bits	Bit offset from	Bit offset to	Data category	Details
MPEG standard flag	1-bit flag	1	0	0	3	1=>MPEG1:0=>MPEG2
red_bw_flag	1-bit flag	1	1	1	3	Default = 0
red_bw_indicator	3-bit ui	3	2	4	3	Default = 000
header present flags	2 flags	2	5	6	3	Sequence header present flag, GOP header present flag
Extension start code flags	16 flags	16	7	22	3	Indicates if a given extension start code exists. The 16 flags correspond to the 16 entries in table 6.2 of ISO/IEC 13818-2 in the order they are listed.
Other start codes	3 flags	3	23	25	3	user_data_start_code sequence_error_code sequence_end_code
sequence header						
horizontal_size	14-bit uimsbf	14	26	39	2	Includes extension
vertical_size	14-bit uimsbf	14	40	53	2	Includes extension
aspect_ratio_information	4-bit uimsbf	4	54	57	2	
frame_rate_code	4-bit uimsbf	4	58	61	2	
bit_rate	30-bit uimsbf	30	62	91	2	Includes extension
vbv_buffer_size	18-bit uimsbf	18	92	109	2	Includes extension
constrained_parameter_flag	1-bit flag	1	110	110	2	
sequence extension						
profile_and_level_indication	8-bit uimsbf	8	111	118	2	
progressive_sequence	1-bit flag	1	119	119	2	
chroma_format	2-bit uimsbf	2	120	121	2	
low_delay	1-bit flag	1	122	122	2	
sequence display extension						
video_format	3-bit uimsbf	3	123	125	2	
color_description	1-bit flag	1	126	126	2	
color_primaries	8-bit uimsbf	8	127	134	2	
transfer_characteristics	8-bit uimsbf	8	135	142	2	
matrix_coefficients	8-bit uimsbf	8	143	150	2	
display_horizontal_size	14-bit uimsbf	14	151	164	2	
display_vertical_size	14-bit uimsbf	14	165	178	2	
group of pictures header						
time_code	25-bit field	25	179	203	2	
closed_gap	1-bit flag	1	204	204	2	
broken_link	1-bit flag	1	205	205	2	

(continued)

Table 4 – Picture rate elements (continued)

Parameter	Number format	No. of bits	Bit offset from	Bit offset to	Data category	Details
picture header						
temporal_reference	10-bit uimsbf	10	206	215	1	
picture_coding_type	3-bit uimsbf	3	216	218	1	
vbv_delay	16-bit uimsbf	16	219	234	1	To be calculated (see note 1)
full_pel_forward_vector	1-bit flag	1	235	235	1	
forward_f_code	3-bit uimsbf	3	236	236	1	
full_pel_backward_vector	1-bit flag	1	239	239	1	
backward_f_code	3-bit uimsbf	3	240	242	1	
picture coding extension						
forward_horizontal_f_code	4-bit uimsbf	4	243	246	1	
forward_vertical_f_code	4-bit uimsbf	4	247	250	1	
backward_horizontal_f_code	4-bit uimsbf	4	251	254	1	
backward_vertical_f_code	4-bit uimsbf	4	255	258	1	
intra_dc_precision	2-bit uimsbf	2	259	260	1	
picture_structure	2-bit uimsbf	2	261	262	1	
top_field_first	1-bit flag	1	263	263	1	
frame_pred_frame_dct	1-bit flag	1	264	264	1	
concealment_motion_vectors	1-bit flag	1	265	265	1	
q_scale_type	1-bit flag	1	266	266	1	
intra_vlc_format	1-bit flag	1	267	267	1	
alternate_scan	1-bit flag	1	268	268	1	
repeat_first_field	1-bit flag	1	269	269	1	
chroma_420_type	1-bit flag	1	270	270	1	
progressive frame	1-bit flag	1	271	271	1	
composite_display_flag	1-bit flag	1	272	272	1	
v-axis	1-bit flag	1	273	272	1	
field_sequence	3-bit uimsbf	3	274	276	1	
sub_carrier	1-bit flag	1	277	277	1	
burst_amplitude	7-bit uimsbf	7	278	284	1	
sub_carrier_phase	8-bit uimsbf	8	285	292	1	
quant matrix extension						See note 2.
load_intra_quantizer_matrix	1-bit flag	1	293	293	1	
load_non_intra_quantizer_matrix	1-bit flag	1	294	294	1	
load_chroma_intra_quantizer_matrix	1-bit flag	1	295	295	1	
load_chroma_non_intra_quantizer_matrix	1-bit flag	1	296	296	1	
intra_quantizer_matrix[64]	64* 0..255	512	297	808	2	
non_intra_quantizer_matrix[64]	64* 0..255	512	809	1320	2	
chroma_intra_quantizer_matrix[64]	64* 0..255	512	1321	1832	2	
chroma_non_intra_quantizer_matrix[64]	64* 0..255	512	1833	2344	2	

(continued)

Table 4 – Picture rate elements (concluded)

Parameter	Number format	No. of bits	Bit offset from	Bit offset to	Data category	Details
picture display extension						
frame_center_horizontal_offset_1	16-bit uimsbf	16	2345	2360	2	
frame_center_vertical_offset_1	16-bit uimsbf	16	2361	2376	2	
frame_center_horizontal_offset_2	16-bit uimsbf	16	2377	2392	2	
frame_center_vertical_offset_2	16-bit uimsbf	16	2393	2408	2	
frame_center_horizontal_offset_3	16-bit uimsbf	16	2409	2424	2	
frame_center_vertical_offset_3	16-bit uimsbf	16	2425	2440	2	
copyright extension						
Copyright flag	1-bit flag	1	2441	2441	2	
Copyright identifier	8-bit code	8	2442	2449	2	
Original or copy	1-bit flag	8	2450	2450	2	
Copyright number	64-bit uimsbf	64	2451	2514	2	
PTS/DTS						
PTS_DTS_flag	2-bit flag	2	2515	2516	1	
PTS value	33-bit uimsbf	33	2517	2549	2	
DTS value	33-bit uimsbf	33	2550	2582	2	
spare reserved bits						
Spare	41-bit uimsbf	41	2583	2623		
user data area						
User data		1664	2624	4287	2	
picture rate information CRC						
32-bit protection CRC	32-bit uimsbf	32	4288	4319		

NOTES
1 This value shall be calculated as defined in SMPTE 327M.
2 Refer to SMPTE 327M for further details.

Table 5 – Repetition of picture rate information

Video format	Pels	No. of MBs	No. of copies
625/50 frame	720 × 576	1620	12
625/50 field	720 × 288	810	6
525/60 frame	720 × 480	1350	10
525/60 field	720 × 240	675	5

PROPOSED SMPTE STANDARD

for Television — MPEG-2 Video Recoding Data Set

SMPTE 327M

Table of contents

1 Scope

This standard specifies the content of the picture related recoding data set for the representation of ISO/IEC 13818-2 MPEG coding information for the purpose of optimally cascading decoders and recoders at any bit rate or GOP structure. The coding information is as derived from an ISO/IEC 13818 compliant MPEG bit stream during the picture decoding process, as described in ISO/IEC 13818-2.

The scope and operation of this standard are the definition of the content of a sufficient recoding data set which may be derived in decoders that comply with ISO/IEC 13818-2, including all nonscalable profiles defined in ISO/IEC 13818-2.

To allow the resynchronization of the video and its associated audio or data after processing, a mechanism using some additional information derived from ISO/IEC 13818-1 is also included in this standard.

This sufficient data set may be transported by various means defined in other SMPTE standards.

2 Normative references

The following standards 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 agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ITU-T H.222.0 with amend 1-2, ISO/IEC 13818-1:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 1: Systems

ITU-T H.262 with amend 1/corr 1 and amend 2/corr 2, ISO/IEC 13818-2:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 2: Video

3 General

The principal application of this standard is to preserve the quality of the video signal when cascading MPEG-2 decoders and coders (including transcoding) by feeding forward previous coding decisions.

The MPEG-2 recoding data set is described as sufficient when it contains the data required that, in combination with an MPEG-2 decoded or partially decoded picture, allows bit accurate recreation of the previously picture coded bit stream.

The various methods whereby the data set is transported alongside the MPEG-2 decoded or partially decoded picture information are the subject of other standards.

THIS PROPOSAL IS PUBLISHED FOR COMMENT ONLY

This standard describes the content of the MPEG-2 recoding data set as extracted or derived from an MPEG-2 bit stream.

The information required in the sufficient MPEG-2 recoding data set can be broken down into three parts. These three parts consist of the picture rate information (described in clause 5), the macroblock rate information (described in clause 6), and some additional information (described in clause 7).

4 Definitions

4.1 bit stream: An ordered series of bits conforming to ISO/IEC 13818-2.

4.2 bottom field: One of two fields that comprise a frame. Each line of a bottom field is spatially located immediately below the corresponding line of the top field.

4.3 cascading: The process where video that has once been coded (compressed) is subsequently decoded and coded once more. This cascaded step could carry on for any number of generations.

4.4 coding: The process by which an uncompressed video sequence is compressed to a bit stream that conforms to ISO/IEC 13818-2.

4.5 column: A vertical column of macroblocks spanning the full height of the decoded picture (columns are numbered from left to right starting at 0).

4.6 decoder: A compressed bit stream decoder that complies with ISO/IEC 13818-2.

4.7 DTS: Decoding time stamp (see ISO/IEC 13818-1).

4.8 macroblock: Defined in ISO/IEC 13818-2 as a block of 16 x 16 luminance pixels.

4.9 macroblock rate information: This corresponds to the coding information from the ISO/IEC 13818-2 bit stream that relates to the individual macroblocks as defined in this standard.

4.10 MPEG profile / level: As defined in ISO/IEC 13818-2.

4.11 picture: As defined in ISO/IEC 13818-2.

4.12 picture rate information: In this standard, this corresponds to the coding information from the ISO/IEC 13818-2 bit stream that relates to the whole picture plus some additional information derived from sequence and GOP headers.

4.13 PTS: Presentation time stamp (see ISO/IEC 13818-1).

4.14 recoding data set: The set of information derived from an ISO/IEC 13818-1/2 compliant stream which, when accompanied by decoded or partially decoded video information, assists in the recreation of the original bit stream as defined in this standard.

4.15 stripe: A horizontal row of macroblocks spanning the full width of the decoded picture (stripes are numbered from top to bottom starting at 0).

4.16 sufficient: [As applied usually to the recoding data set] allows bit accurate recreation of the previously picture coded bit stream when used in combination with the decoded or partially decoded picture.

4.17 top field: One of two fields that comprise a frame. Each line of a top field is spatially located immediately above the corresponding line of the bottom field.

4.18 transcoding: A conversion within the MPEG-2 stream domain, such as bit-rate changing or changing the group of pictures (GOP) structure.

5 Picture rate information

This clause defines the minimum picture rate information required by the MPEG-2 recoding data set. Table 1 lists the elements taken from the bit stream which shall be included.

For the bit order and number format of this information, refer to the corresponding transport mechanism.

Table 1 – Picture rate elements

Parameter	No. of bits (See note 1)	Notes
Sequence header present flag	1-bit flag	
GOP header present flag	1 flag	
Extension start code flags	16 flags	Indicates if a given extension start code exists. The 16 flags correspond to the 16 entries in table 6.2 of ISO/IEC 13818-2 in the order they are listed.
Other start code flags	3 flags	user_data_start_code, sequence_error_code, sequence_end_code
sequence header		
horizontal_size	14-bit ui	Includes extension
vertical_size	14-bit ui	Includes extension
aspect_ratio_information	4-bit ui	
frame_rate_code	4-bit ui	
bit_rate	30-bit ui	
vbv_buffer_size	18-bit ui	Includes extension
constrained_parameters_flag	1-bit flag	
sequence extension		
profile_and_level_indication	8-bit ui	
progressive_sequence	1-bit flag	
chroma_format	2-bit ui	
low_delay	1-bit flag	
sequence display extension		
video_format	3-bit ui	
color_description	1-bit flag	
color_primaries	8-bit ui	
transfer characteristics	8-bit ui	
matrix_coefficients	8-bit ui	
display_horizontal_size	14-bit ui	
display_vertical_size	14-bit ui	
group of pictures header		
time_code	25-bit field	
closed_gop	1-bit flag	
broken_link	1-bit flag	
picture header		
temporal_reference	10-bit ui	
picture_coding_type	3-bit ui	
vbv_delay	16-bit ui	See note 2
full_pel_forward_vector	1-bit flag	
forward_f_code	3-bit ui	
full_pel_backward_vector	1-bit flag	
backward_f_code	3-bit ui	
picture coding extension		
forward_horizontal_f_code	4-bit ui	
forward_vertical_f_code	4-bit ui	
backward_horizontal_f_code	4-bit ui	
backward_vertical_f_code	4-bit ui	

(continued)

Table 1 – Picture rate elements (concluded)

Parameter	No. of bits (See note 1)	Notes
intra_dc_precision	2-bit ui	
picture_structure	2-bit ui	
top_field_first	1-bit flag	
frame_pred_frame_dct	1-bit flag	
concealment_motion_vectors	1-bit flag	
q_scale_type	1-bit flag	
intra_vlc_format	1-bit flag	
alternate_scan	1-bit flag	
repeat_first_field	1-bit flag	
chroma_420_type	1-bit flag	
progressive_frame	1-bit flag	
composite_display_flag	1-bit flag	
v_axis	1-bit flag	
field_sequence	3-bit ui	
sub_carrier	1-bit flag	
burst_amplitude	7-bit ui	
sub_carrier_phase	8-bit ui	
quant matrix extension	See note 3	
load_intra_quantizer_matrix	1-bit field	
load_non_intra_quantizer_matrix	1-bit flag	
load_chroma_intra_quantizer_matrix	1-bit flag	
load_chroma_non_intra_quantizer_matrix	1-bit flag	
intra_quantizer_matrix [64]	64* 0..255	
non_intra_quantizer_matrix [64]	64* 0..255	
chroma_intra_quantizer_matrix [64]	64* 0..255	
chroma_non_intra_quantizer_matrix [64]	64* 0..255	
picture display extension		
frame_center_horizontal_offset_1	16-bit ui	
frame_center_vertical_offset_1	16-bit ui	
frame_center_horizontal_offset_2	16-bit ui	
frame_center_vertical_offset_2	16-bit ui	
frame_center_horizontal_offset_3	16-bit ui	
frame_center_vertical_offset_3	16-bit ui	

NOTES
1 The order and number format of each element is defined in the appropriate transport mechanism.
2 This value shall be recalculated as vbv_delay = DTS-PCR, where this calculation is done at the points where the DTS is received. If no DTS is present for the current picture, then the value of DTS shall be calculated as defined in reference clause 2.5.2.3 of ISO/IEC 13818-1.
3 The following elements of the bit stream may be present in either the sequence header or the quant matrix extension (or both). The information required by the MPEG-2 recoding data set is exactly that required by an MPEG-2 video decoder. That is, in the case of 4:2:2 profile, the quantizer matrix values shall be taken from the last set of values received in the bit stream for each of the four matrices, unless a sequence header was received at the beginning of the current picture and no matrix was loaded in either the sequence header or the quant matrix extension, in which case the values will be taken from the default tables in ISO/IEC 13818-2. The load_quantizer_matrix flags shall be set to 1 if, and only if, either of the two possible occurrences of those flags in the bit stream takes the value 1. In the case of the Main Profile, exactly the same procedure is followed except that all values relating to chroma matrices are undefined.

6 Macroblock rate information

The MPEG-2 recoding data set contains macroblock rate elements that are either extracted or derived from the MPEG-2 bit stream. Table 2 contains the information directly extracted from the MPEG-2 bit stream while table 3 contains the elements that are derived.

Together these lists show all the data elements that are required for each macroblock in order to create the sufficient MPEG-2 recoding data set.

6.1 Information extracted from the original bit stream

Where reference is made to elements in ISO/IEC 13818-2, these are indicated in the text in italics (see table 2).

6.2 Information derived from the original bit stream

The following information is not available directly from the original MPEG-2 stream, but can be derived from the bit stream during the decoding process. Where reference is made to elements in ISO/IEC 13818-2, these are indicated in the text in italics (see table 3).

Table 2 – Information extracted from MPEG-2 bit stream

Parameter	No. of bits	Definition
mb_quant	1-bit flag	Value equal to *macroblock_quant*, derived from *macroblock_type*.
mb_mfwd	1-bit flag	Value equal to *macroblock_motion_forward*, derived from *macroblock_type*.
mb_mbwd	1-bit flag	Value equal to *macroblock_motion_backward*, derived from *macroblock_type*.
mb_pattern	1-bit flag	Value equal to *macroblock_pattern*, derived from *macroblock_type*.
mb_intra	1-bit flag	Value equal to *macroblock_intra*, derived from *macroblock_type*.
mb_vert_field_sel[r][s]	4-bit ui	Value equal to *motion_vertical_field_select[r][s]*.
dct_type	1-bit flag	Value equivalent to *dct_type* held in the original bit stream.
motion_type	2-bit ui	Value equivalent to *frame_motion_type* in frame pictures and *field_motion_type* in field pictures when present in the original bit stream. If neither *frame_motion_type* nor *field_motion_type* is present in the bit stream, then the value of motion_type is not defined.
q_scale_code	5-bit ui	Value equal to *q_scale_code*.
coded_block_pattern[7...0]	8-bit ui	Value equivalent to *coded_block_pattern[7...0]* held in the original bit stream.
mv [r][s][t] mv [0][0][0] mv [0][0][1] mv [0][1][0] mv [0][1][1] mv [1][0][0] mv [1][0][1] mv [1][1][0] mv [1][1][1]	 13-bit si 9-bit si 13-bit si 9-bit si 13-bit si 9-bit si 13-bit si 9-bit si	Values corresponding to the derived motion vectors *vector'[r][s][t]*, as defined in clause 7.6.3 of ISO/IEC 13818-2. The meanings associated with the dimensions in the arrays mv [r] [s] [t] and *vector'[r][s][t]* are defined in table 7-7 of ISO/IEC 13818-2. The number format is twos complement and is specified to an accuracy of one half sample.

Table 3 – Information derived from MPEG-2 bit stream

Parameter	No. of bits	Definition
skipped_mb	1-bit flag	Derived from the values of *macroblock_escape* and *macroblock_address_increment* as defined in clause 6.3.17 of ISO/IEC 13818-2. skipped_mb will be set to 1 if the current macroblock has been skipped; otherwise it is 0.
slice_start_flag	1-bit flag	Set to the value 1 if the current macroblock is the first macroblock in a slice; otherwise it is set to 0.
num_coef_bits	14-bit ui	Set to the number of bits used by all codes in the syntax elements *coded_block_pattern()* and *block()*, within the syntax element *macroblock()*, defined in clause 6.2.5 of ISO/IEC 13818-2.
num_mv_bits num_mv_bits [7:2] num_mv_bits [1:0]	 6-bit ui 2-bit ui	Set to the number of bits used by all codes in the syntax elements *motion_vectors(0)* and *motion_vectors(1)* within the syntax element *macroblock()*, defined in clause 6.2.5 of ISO/IEC 13818-2.
num_other_bits	7-bit ui	Set to the number of bits used in the syntax element *macroblock()*, defined in clause 6.2.5 of ISO/IEC 13818-2, excluding those codes accounted for by num_mv_bits and num_coef_bits.

7 Transport of the MPEG-2 recoding data set

The previous clauses describe the necessary information for the sufficient MPEG-2 recoding data set. However, additional information and suitable transport mechanisms are required in order to be able to fully exploit this information in practical systems. The following additional information shall be available in all transport mechanisms:

– User data: This is unbounded by ISO/IEC 13818-2. A defined capacity shall be made available by the transport system employed, so that essential elements can be passed with the MPEG-2 recoding data set.

– Copyright: The copyright extension defined in ISO/IEC 13818-2 is not necessary for cascading operations. However, it is necessary that this information should be propagated through the system and it, therefore, forms an integral part of the standard.

7.1 Additional information derived from a packetized elementary stream

Where the original source was derived from a packetized elementary stream, the following additional information must be included:

– PTS/DTS as defined in ISO/IEC 13818-1. As audio, video, and other services are frequently split apart and pass through independent signal processing paths, it is necessary to add this information to enable the reconstruction of the complete stream after such processing.

7.2 Additional information derived from an elementary stream

Where this information is derived from an elementary stream, the appropriate mechanism is described in SMPTE 328M.

8 Compressed stream format for bandwidth reduced systems

There are, however, circumstances in which the transmission of the sufficient data set described in this

standard is not possible. Some existing equipment has a very low capacity for the transmission of the recoding data and this limit will have an impact on subsequent stages including the MPEG-2 recoding process.

Examples of such equipment are those which are not transparent in the active video region and/or are restricted in their capacity to carry the additional data. However, such equipment may still be able to transport some recoding information.

Transporting a limited set of recoding data can still have a beneficial effect on the final recoding stage though clearly not as effective as the use of the sufficient data set. There are two methods to reduce the bit rate of the sufficient recoding data set:

The first step is to use variable-length coding (VLC) for those parts of the recoding data which have redundancy.

A full description of this mechanism is defined in SMPTE 329M.

Where this remains insufficient for the reduced bandwidth available, use of a subset of the recoding data, considering the balance and trade-off between the band limiting effect and the recoded picture quality, may be used.

To ensure interoperablity between the various transport mechanisms proposed for the MPEG-2 recoding data set, it is essential to ensure the consistency of the reduced bandwidth syntax representation. This is defined in the reduced bandwidth indicator table (table 4) and shall be incorported into all transport mechanisms using this data set.

Reduced bandwidth operation shall be signaled in the following manner:

red_bw_flag shall be set to 0 in normal operation. For reduced bandwidth operation, this flag shall be set to 1.

Where the red_bw_flag is set to 1, then the red_bw_indicator shall be examined to determine what information shall be present from the MPEG-2 recoding data set as defined in table 4.

Table 4 – Reduced bandwidth indicators

red_bw_indicator_	num_coef_bits, num_mv_bits, num_other_bits	q_scale_code, q_scale_type	motion_type, mv_vert_field_sel[r][s], mv[r][s][t]	mb_mfwd, mb_mbwd	mb_pattern	coded_block_pattern	mb_intra	slice_start	dct_type	mb_quant	skipped mb
Indicator 0	0	1	1	1	1	1	1	1	1	1	1
Indicator 1	0	1	1	1	1	0	1	0	1	0	0
Indicator 2	0	1	0	0	0	0	0	0	0	0	0
Indicator 3	0	0	0	0	0	0	0	0	0	0	0

NOTES
1 0 = this information is not present.
2 1 = this information exists.

Annex A (informative)
Transport mechanisms

This annex contains information about the various proposed transport mechanisms for the MPEG-2 recoding data set and the relationships among these standards (see figure A.1).

A.1 Transport mechanism for macroblock based editing

To allow for the editing of the MPEG-2 recoding data set to occur in parallel with the decoded video, the following key features will be required of the transport mechanism:

- Spatial alignment of the decoded video and the MPEG-2 recoding data set on a macroblock basis;
- Temporal alignment of the decoded video and the MPEG-2 recoding data set on a macroblock and picture basis;
- A system to accompany or embed this information into the ITU-R BT.601 stream without adversely affecting the picture quality for standard definition systems and a similar system for high-definition systems;
- This system must also allow the MPEG-2 recoding data set to be destroyed in the areas of the picture where two or more video signals overlap.

For this type of transport mechanism, all macroblock rate information must be available on a macroblock basis.

If in the MPEG-2 decoding process, there is a need to derive or infer certain elements for use in the decoding of the current macroblock (e.g., skipped macroblock), then all the elements described in clause 6 must be set to those used by the decoding process, as defined in ISO/IEC 13818-2.

A full description of the standard defined version of this transport mechanism is given in SMPTE 319M.

A.2 History information

There are instances where a video signal is first coded by a long GOP, low bit rate coder to get the cost benefit of the source media. This may then be converted to a short GOP, high bit rate stream to get functional benefits, such as splicing and VTR stunt play modes. The short GOP signal may then be converted back to a long GOP low bit rate stream at the output.

In such cases, conversion may be done in the compressed domain and it is useful to keep the first set of long GOP coding data present in the short GOP compressed bit stream and pass this to the next compression recoder to achieve the best picture quality at each generation. The carriage of this recoding data from the last but one encoder is called history recoding data. For the reasons outlined in clause 8, a bandwidth limited recoding data set may also be required for this history.

It is also clear that the history recoding data may be carried either in the baseband (ITU-R BT.601 for standard definition signals) domain or the compressed bit stream domain depending on the application. Methods for the carriage of history recoding data will be required for both the baseband and bit stream domains.

The system required to reduce the bandwidth is described in SMPTE 329M.

The method of transporting this signal in the baseband domain is defined in SMPTE 319M.

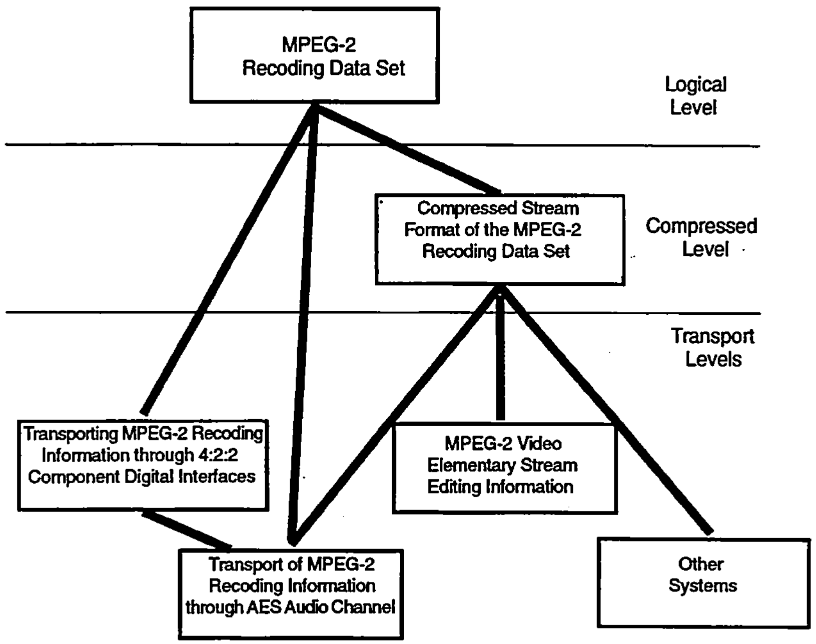

Figure A.1 – Relationships among SMPTE standards

Annex B (informative)
Bibliography

SMPTE 319M, Television — Transporting MPEG-2 Recoding Information through 4:2:2 Component Digital Interfaces

SMPTE 328M, Television — MPEG-2 Video Elementary Stream Editing Information

SMPTE 329M, Television — MPEG-2 Video Recoding Data Set — Compressed Stream Format

ITU-R BT.601-5 (1994), Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios

PROPOSED SMPTE STANDARD

for Television — MPEG-2 Video Elementary Stream Editing Information

1 Scope

This standard defines the MPEG video elementary stream (ES) information to facilitate seamless edits under defined circumstances. The video ES, as defined by the MPEG standards, are supplemented with additional information for professional studio applications. Supplementary information will be carried within the sequence header and the user data area of the video ES. This standard defines the data to be carried and the location of the data.

2 Normative references

The following standards 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 agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI/SMPTE 293M-1996, Television — 720 × 483 Active Line at 59.94-Hz Progressive Scan Production — Digital Representation

ANSI/SMPTE 296M-1997, Television — 1280 × 720 Scanning, Analog and Digital Representation and Analog Interface

SMPTE 12M-1999, Television, Audio and Film — Time and Control Code

SMPTE 274M-1998, Television — 1920 × 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates

SMPTE 309M-1999, Television — Transmission of Date and Time Zone Information in Binary Groups of Time and Control Code

SMPTE 326M, Television — SDTI Content Package Format (SDTI-CP)

SMPTE 327M, Television — MPEG-2 Video Recoding Data Set

SMPTE RP 186-1995, Video Index Information Coding for 525- and 625-Line Television Systems

SMPTE RP 202, Video Alignment for MPEG-2 Coding

ISO/IEC 13818-2:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

ITU-R BT.601-5 (10/95), Studio Encoding Parameters of Digital Television for Standard 4:3 and Widescreen 16:9 Aspect Ratios

ITU-R BT.656-4 (02/98), Interfaces for Digital Component Video Signals in 525-Line and 625-Line Television Systems Operating at the 4:2:2 Level of Recommendation ITU-R BT.601 [Part A]

3 MPEG ES

ISO/IEC 13818-2 defines the characteristics of the MPEG elementary stream (ES). When used in a professional environment where seamless edits (splices) are required, supplementary data needs to be carried by the bit stream.

THIS PROPOSAL IS PUBLISHED FOR COMMENT ONLY

3.1 General

Seamless frame accurate editing of compressed video is most easily accomplished with use of short GOP structures. Longer GOP structures can be edited by decoding and reencoding, by transcoding to shorter GOP structures, or with more involved processing, edited directly. The best approach will be determined by a range of application-specific considerations.

3.2 Sequence header

ISO/IEC 13818-2 does not define the repetition frequency of the sequence header. To be compliant with this standard, the sequence header shall exist at every I frame.

3.3 VBV_delay

An accurate vbv_delay value shall be carried in the video elementary stream. The value of 0xffff shall not be allowed. The vbv_delay value may be relied upon for remultiplexiing elementary streams into transport streams.

4 MPEG ES-syntax elements

The following syntax elements, and the functional descriptions, shall be inserted in the MPEG ES in the user data area:

4.1 V/H coding phase

The current implementation of MPEG does not specify the horizontal and vertical coding phase. This standard requires that the vertical and horizontal coding phase be known in order for decoding and peripheral equipment to process correctly the signal.

V and H coding information shall be included only for SDTV signals where the coding phase is not compliant with SMPTE RP 202. For HDTV signals, H/V coding phase information is defined by SMPTE RP 202.

4.2 Time code

Provision is made for the insertion of two time codes complying with SMPTE 12M. At least one time code, reference date time stamp (as defined in SMPTE 326M) shall be carried as a means of maintaining synchronization with other content or metadata streams. Carriage of a second time code is optional. Compliant decoders shall have the capability to decode the two time codes.

4.3 Picture order

Picture order information specifies the picture duration and is the equivalent to the PTS/DTS present in the MPEG transport stream. Picture order value is counted by field units. Picture order information will be used by multiplexers and remultiplexers to know the picture duration and to be able to parse the repeat_first_field information. In some cases, the latency of the system will be minimized using the picture order information.

4.4 Video Index

Video index, as defined by SMPTE RP 186, shall be carried if present on the baseband signal. Information carried by video index should be preserved during any coding, recoding, editing, or transcoding process. It is envisioned that the data described in the forthcoming SMPTE metadata dictionary (SMPTE 335M) will be handled by the transport mechanism described in SMPTE 326M. These parametric data will include all of the parameters currently coded in video index, although the data representation of some items may be different.

4.5 Ancillary data

Data that is carried in the vertical interval of the baseband signal should be preserved. Ancillary data may consist of more than 23 consecutive zeros. To prevent this condition, a marker shall be inserted every 22 bits.

4.6 History data

History data, consisting of original and subsequent encoding parameters that may be useful in transcoding or reencoding, may be carried by the bit stream. SMPTE 327M defines the content of the history data information. History data may consist of more than 23 consecutive zeros. To prevent this condition, a marker shall be inserted every 22 bits.

4.7 User data

User data is defined by ISO/IEC 13818-2.

5 MPEG ES-syntax

MPEG_ES_editing_information is inserted in the sequence or/and picture user data area of the video ES. All data fields follow the unique 16-bit header which specifies this stream as being compliant with this standard. The syntax shall be flexible and extensible. All syntax elements are distinguished by data ID. Data ID specifies the length and characteristics of the following syntax element.

The V/H coding phase and defined control flags shall be transmitted in user_data of the sequence layer. Others shall be transmitted in user_data of the picture layer. In the picture layer, time_code() and picture_order() will be placed first because these are relatively small amounts of data and could be handled by software. The following data types, video_index(), ancillary_data(), and history_data() are positioned later in the bit stream because these data types potentially can be large in size and may require hardware support. User_data() shall be placed at the end (tail).

NOTE – All syntactic elements described in this standard follow the pseudo-code form, as defined in ISO/IEC 13818.

5.1 Data_ID

8 bits: This syntax element specifies the length and characteristics of the following; 0x00 is forbidden.

Data_ID	Data_type
00	FORBIDDEN
01	V-phase
02	H-phase
03	Time code 1
04	Time code 2
05	Picture order
06	Video index
07	Ancillary data
08	History data
...	...
80	Control flags
...	...
FF	User data

5.2 Higher syntactic structure

Syntax	Bits	Mnemonic
MPEG_ES_editing_information(i){		
SMPTE_header	16	"0x0001"
If (i == 0){/* follows sequence user data */		
While (nextbits() ! = "0x00"){		
If (nextbits() == "V-phase")		
V-phase()	...	
Else if (nextbits() == "H-phase")		
H-phase()	...	
Else if (nextbits() == "Control flags")		
Control flags()	...	
Else		
Break; / * Do nothing */		
}		
}		
Else if (i == 2){ /* follows picture user data */		
While (nextbits() ! = "0x00"){		
If (nextbits() == "Time code 1" II Nextbits() == "Time code 2")		
Time_code()	...	
Else if (nextbits() == "Picture_order"		
Picture_order()	...	
Else		
Break; /* Do nothing */		
}		
While (nextbits() ! = "0x00"){		
If (nextbits() == "Video index")		
Video_index()	...	
Else if (nextbits() == "Ancillary data")		
Ancillary_data()	...	
Else if (nextbits() == "History data")		
History_data()	...	
Else		
Break; / * Do nothing * /		
}		
}		
If (nextbits() == "User data")		
User_data()	...	
Next_start_code()		
}		

5.3 V coding phase

Syntax	Bits	Mnemonic
V-phase(){		
Data_ID	8	bslbf
V-phase	16	uimsbf
}		

V-phase (16 bits): V-phase is an unsigned integer that specifies the top line of the coded frame. When present, V-phase shall be carried in the user data of the sequence layer. For SDTV signals, V-phase shall be carried if the vertical coding phase is not compliant with SMPTE RP 202. For HDTV signals, V coding phase is defined by the SMPTE standard and the V-phase user data shall not be carried.

5.4 H coding phase

Syntax	Bits	Mnemonic
H-phase(){		
Data_ID	8	bslbf
H-phase	8	uimsbf
}		

H-phase (8 bits): H-phase is an unsigned integer that specifies the sample number of the first sample in a video line as defined in ITU-R BT.601. When present, H-phase shall be carried in the user data of the sequence layer. For SDTV signals, H-phase shall be carried if the horizontal coding phase is not compliant with SMPTE RP 202. For HDTV signals, H coding phase is defined by the SMPTE standard and the H-phase user data shall not be carried.

5.5 Control flags

Syntax	Bits	Mnemonic
Control flags(){		
Data_ID	8	bslbf
Picture_order_presence	1	uimsbf
Reserved	7	
}		

Picture_order_presence (1 bit): If this flag is set to 1, all picture order information shall be carried in the user data area of every picture. In this case, the multiplexer may use the picture order information for multiplexing, and low-delay multiplexing will be possible.

5.6 Time coding data

Syntax	Bits	Mnemonic
Time_code(){		
Data_ID	8	bslbf
Time_code [63..48]	16	uimsbf
Marker_bit	1	bslbf
Time_code [47..32]	16	uimsbf
Marker_bit	1	bslbf
Time_code [31..16]	16	uimsbf
Marker_bit	1	bslbf
Time_code [15..0]	16	uimsbf
Marker_bit	1	bslbf
Reserved_bits	4	bslbf
}		

Time_code (64 bits): The time-code format defined below shall comply with ANSI/SMPTE 12M. Two time codes may be carried. At least one time code (reference time/date stamp [SMPTE 309M]) shall be carried. All time codes specified shall be presented at the time the corresponding picture is established during the encoding process.

NOTE — The 64-bit contents of the time code are mapped as follows:

Syntax	Bits	Mnemonic
Time_code [63..0]{		
Color frame flag	1	bslbf
Drop frame flag (NTSC)/unused (PAL)	1	bslbf
TV frame tens	2	uimsbf
TV frame units	4	uimsbf
Field phase (NTSC)/binary group flag 0 (PAL)	1	bslbf
TV seconds tens	3	uimsbf
TV seconds units	4	uimsbf
Binary group flag 0 (NTSC)/binary group flag 2 (PAL)	1	bslbf
TV minutes tens	3	uimsbf
TV minutes units	4	uimsbf
Binary group flag 2 (NTSC)/field phase (PAL)	1	bslbf
Binary group flag 1 (NTSC)/binary group flag 1 (PAL)	1	bslbf
TV hours tens	2	uimsbf
TV hours units	4	uimsbf
2nd binary group	4	uimsbf
1st binary group	4	uimsbf
4th binary group	4	uimsbf
3rd binary group	4	uimsbf
6th binary group	4	uimsbf
5th binary group	4	uimsbf
8th binary group	4	uimsbf
7th binary group	4	uimsbf
}	4	uimsbf

5.7 Picture order

Syntax	Bits	Mnemonic
Picture_order(){		
Data_ID	8	bslbf
DTS_presence	1	bslbf
PTS_counter	7	uimsbf
If (DTS_presence == "1"){		
Marker_bits	1	bslbf
DTS_counter	7	uimsbf
}		
}		

DTS_presence (1 bit): If DTS_presence is set to 1, DTS_counter field shall exist.

PTS_counter (7 bits): This is a 7-bit unsigned integer which is equivalent to PTS counted by field units. This is a modulo 128 counter and shall increment according to picture duration taking into account repeat_first_field and reordering delay caused by B-picture.

DTS_counter (7 bits): This is a 7-bit unsigned integer which is equivalent to DTS counted by field units. This is a modulo 128 counter and shall increment according to picture duration taking into account repeat_first_field and reordering delay caused by B-picture.

Annex A presents two examples of PTS/DTS counter operation for long GOP and short GOP bit stream formats.

5.8 Video index

Syntax	Bits	Mnemonic
Video_index(){		
Data_ID	8	bslbf
Field_ID	2	bslbf
Line_number	14	uimsbf
Video_index_length	8	uimsbf
Marker_bits	1	bslbf
For (j=0; j<video_index_length; j++){		
Video_index_payload	22	uimsbf
Marker_bits	1	bslbf
}		
While (!bytealigned())		
Zero_bit	1	"0"
}		

Field_ID (2 bits): If the progressive_sequence flag is set to 0 (interlace), field_ID shall specify the field index counted by field unit in the presentation order of this picture. The counter of the earliest field shall be set to 0. If repeat_first_field is set to 0, the picture shall include two fields indexed as 0 and 1. If repeat_first_field is set to 1, the picture shall include three fields indexed as 0, 1, and 2.

If progressive_sequence flag is set to 1 (progressive), field_ID shall specify the frame index counted by progressive frame unit in the presentation order of this picture. The counter of the earliest frame shall be set to 0. If repeat_first_field and top_field_first are set to 0, the picture shall include only one progressive frame indexed as 0. If repeat_first_field is set to 1 and top_field_first are set to 0, the picture shall include two progressive frames indexed as 0 and 1. If repeat_first_field is set to 1 and top_field_first is set to 1, the picture shall include three progressive frames indexed as 0, 1, and 2.

Line_number (14 bits): This specifies the absolute frame-based line number of the video_index data; 0 is not allowed. The line number shall be specified according to the relevant video standard (ITU-R BT.656, SMPTE 274M, ANSI/SMPTE 293M, ANSI/SMPTE 296M).

Video_index_length (8 bits): This specifies the loop count of the following video_index_payload.

Video_index_payload (22 bits): These contain the payload of the video_index data. Video_index_payload is defined in SMPTE RP 186 for SDTV.

5.9 Ancillary data

Syntax	Bits	Mnemonic
Ancillary_data(){		
Data_ID	8	bslbf
Field_ID	2	bslbf
Line_number	14	uimsbf
Ancillary_data_length	16	uimsbf
Marker_bits	1	bslbf
For (j=0; j<ancillary_data_length; j++){		
Ancillary_data_payload	22	uimsbf
Marker_bits	1	bslbf
}		
While (!bytealigned())		
Zero_bit	1	"0"
}		

Field _ID (2 bits): If the progressive_sequence flag is set to 0 (interlace), field_ID shall specify the field index counted by the field unit in the presentation order of

this picture. The counter of the earliest field shall be set to 0. If repeat_first_field is set to 0, the picture shall include two fields indexed as 0 and 1. If repeat_first_field is set to 1, the picture shall include three fields indexed as 0, 1, and 2.

If the progressive_sequence flag is set to 1 (progressive), field _ID shall specify the frame index counted by the progressive frame unit in the presenttion order of this picture. The counter of the earliest frame shall be set to 0. If repreat_first_field and top_field_first are set to 0, the picture shall include only one progressive frame indexed as 0. If repeat_first_field is set to 1 and top_field_first is set to 0, the picture shall include two progressive frames indexed as 0 and 1. If repeat_first_field is set to 1 and top_field_first is set to 1, the picture shall include three progressive frames indexed as 0, 1, and 2.

Line_number (14 bits): This specifies the absolute frame-based line number of the ancillary data; 0 is not allowed. The line number shall be specified according to the relevant video standard (ITU-R BT.656, SMPTE 274M, ANSI/SMPTE 293M, ANSI/SMPTE 296M).

Ancillary_data_length (16 bits): This specifies the loop count of the following ancillary data_payload.

Ancillary_data_payload (22 bits): These contain the payload of the ancillary data.

5.10 History data

Syntax	Bits	Mnemonic
History_data(){		
Data_ID	8	bslbf
Marker_bits	1	bslbf
History_data_length	23	uimsbf
Marker_bits	1	bslbf
For (j=0; j<History_data_length; j++){		
History_data_payload	22	uimsbf
Marker_bits	1	bslbf
}		
While (!bytealigned())		
Zero_bit	1	"0"
}		

History_data_length (23 bits): This specifies the loop count of the following ancillary data_payload.

History_data__payload (22 bits): These bits contain the payload of the ancillary data. History data will be encapsulated in this field. The contents are defined in SMPTE 327M.

5.11 User data

Syntax	Bits	Mnemonic
User_data(){		
Data_ID	8	bslbf
While (nextbits() ! = "0x000001")		
User_data	8	uimsbf
}		
}		

User data (8 bits): The user data byte count is variable. User data shall not emulate the MPEG start codes (see ISO/IEC 13818-2). The end of user data is detected when the next start code appears. User data shall be placed at the end (tail) of the edit information.

Annex A (informative)
Usage of PTS and DTS_counter

A.1 Example 1: Long GOP

The following is an example of PTS and DTS_counter taken from the beginning of a video sequence. In this example, there are two coded B-frames between successive coded P-frames and also two coded B-frames between successive coded I- and P-frames and all pictures are frame pictures with a 3-2 pulldown operation.

At the encoder input:

Frame No.	1	2	3	4	5	6	7	8	9	10	11	12	13
Picture type	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1

At the encoder output, in the coded bit stream, and at the decoder input:

Frame No.	1	4	2	3	7	5	6	10	8	9	13	11	12
Picture type	I	P	B	B	P	B	B	I	B	B	P	B	B
Repeat_first_field	1	0	0	1	1	1	0	0	0	1	1	1	0
Top_field_first	1	1	0	0	0	1	0	0	1	1	1	0	1
DTS_counter	125	0	3	5	8	10	13	15	18	20	23	25	28

At the decoder output:

Frame No.	1	2	3	4	5	6	7	8	9	10	11	12	13
Picture type	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1
PTS_counter	0	3	5	8	10	13	15	18	20	23	25	28	30

A.2 Example 2: Short GOP

The following is an example of PTS and DTS_counter taken from the beginning of a short GOP video sequence. In this example, the counters are wrapping around at 20 instead of 128.

At the encoder input:

Frame No.	1	2	3	4	5	6	7	8	9
Picture type	I	B	I	B	I	B	I	B	I
Repeat_first_field	0	1	0	1	0	1	0	1	0
Top_field_first	1	1	0	0	1	1	0	0	1

At the encoder output, in the coded bit stream, and at the decoder input:

Frame No.	1	3	2	5	4	7	6	9	8
Picture type	I	I	B	I	B	I	B	I	B
Repeat_first_field	0	0	1	0	1	0	1	0	1
Top_field_first	1	0	1	1	0	0	1	1	0
DTS_counter	18	0	2	5	7	10	12	15	17

At the decoder output:

Frame No.	1	2	3	4	5	6	7	8	9
Picture type	I	B	I	B	I	B	I	B	I
Repeat_first_field	0	1	0	1	0	1	0	1	0
Top_field_first	1	1	0	0	1	1	0	0	1
PTS_counter	0	2	5	7	10	12	15	17	0

Annex B (informative)
Bibliography

SMPTE 291M-1998, Television — Ancillary Data Packet and Space Formatting

SMPTE 308M-1998, Television — MPEG-2 4:2:2 Profile at High Level

SMPTE 329M, Television — MPEG-2 Video Recoding Data Set — Compressed Stream Format

Forthcoming SMPTE 335M, Television — Metadata Dictionary

ISO/IEC 13818-1:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

PROPOSED SMPTE STANDARD

SMPTE 329M

for Television — MPEG-2 Video Recoding Data Set — Compressed Stream Format

Table of contents

1 Scope

This standard specifies the stream format of the MPEG-2 recoding data set for the representation of compressed ISO/IEC 13818-2 MPEG coding information, as used in applications requiring transport systems of reduced data capacity.

The coding information is derived from an ISO/IEC 13818-2 compliant MPEG bit stream during the decoding process, as described in ISO/IEC 13818-2.

The information based on this stream format may be transported by various means; for example, the elementary stream format defined in SMPTE 328M.

2 Normative references

The following standards 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 agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 327M, Television—MPEG-2 Video Recoding Data Set

ITU-T H.262, ISO/IEC 13818-2:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

3 General

The main application of this standard is to preserve the quality of the video signal when cascading decoders and coders for editing or transcoding purposes by feeding forward previous coding decisions.

The MPEG-2 recoding data set in combination with an MPEG-2 decoded or partially decoded picture is effective for implementing editing or transcoding of the MPEG-2 bit stream. There are, however, applications in which the transmission of all the recoding data set is not possible. Some current equipment may have restricted capacity for the transmission of the recoding data. This limitation has an impact on subsequent compression stages which can make use of the MPEG-2 recoding process.

In order to decrease the bit rate for the recoding data set, the MPEG-2 recoding data set is converted into an MPEG-like stream, which is called the compressed stream format of the MPEG-2 recoding set. This standard defines this stream format.

The compressed stream format of the MPEG-2 recoding set much resembles an MPEG-2 video stream, except that the compressed stream format of the MPEG-2 recoding set does not have DCT coefficients which occupy the major part of an MPEG video stream.

THIS PROPOSAL IS PUBLISHED FOR COMMENT ONLY

By adopting an MPEG-like form for the compressed stream format of the MPEG-2 recoding set, use is made of an efficient compression for this data set for storage and transmission operations. Moreover, it may be possible to reuse the MPEG decoding devices to decode this compressed stream format of the MPEG-2 recoding set.

The compressed format of the MPEG-2 recoding set syntactical structure is modified by red_bw_flag and red_bw_indicators. The selection of one of the variations of the syntactical structure will be determined by the application by taking into account a balance between bit rate and the number of elements in the recoding data set.

This standard does not describe the transport mechanism, but describes the stream structure of the compressed stream format of the MPEG-2 recoding set. This stream format is independent of application, and all the transport information in the reduced bandwidth recoding data transportation system should be based on this stream.

The transport mechanism depends on application, which should be defined in other standards documents. The first of these is the MPEG-2 elementary stream editing information, SMPTE 328M.

4 Definitions

4.1 bit stream: An ordered series of bits conforming to ISO/IEC 13818-2.

4.2 cascading: This describes the process where video that has once been coded (compressed) is subsequently decoded and coded once more. This cascading step could carry on for any number of generations.

4.3 coding: This is the process by which an uncompressed video sequence is compressed to a bit stream that conforms to the ISO/IEC 13818-2 standard.

4.4 compressed stream format of the MPEG-2 recoding data set: This is the compressed information of the recoding data set which is used in reduced bandwidth recoding data transportation systems.

4.5 decoder: A compressed bit stream decoder that complies with the ISO/IEC 13818-2 standard.

4.6 macroblock: Defined in the ISO/IEC 13818-2 standard as a block of 16 × 16 luminance pixels.

4.7 MPEG profile/level: As defined in ISO/IEC 13818-2.

4.8 picture: As defined in ISO/IEC 13818-2.

4.9 recoding data set: The set of information derived from an ISO/IEC 13818-2 compliant stream which, when accompanied by decoded or partially decoded video information, assists in the recreation of the original MPEG-2 stream.

4.10 reduced bandwidth recoding data transportation system: This is the system that does not have enough bandwidth capacity for transporting a full set of the recoding data set, but is able to transport the compressed stream format of the MPEG-2 recoding data set.

4.11 reduced bandwidth stream: The highest syntactic structure of the reduced MPEG-2 recoding data set. It contains the recoding information for one picture.

4.12 stripe: A horizontal row of macroblocks spanning the full width of the decoded picture (stripes are numbered from top to bottom starting at zero).

4.13 transcoding: A conversion within the MPEG-2 stream domain, such as bit rate changing or changing the group of pictures (GOP) structure.

5 Compressed stream format of the MPEG-2 recoding data set

5.1 Compressed stream format of the MPEG-2 recoding set syntax

The compressed stream format of the MPEG-2 recoding set is a subset of the ISO/IEC 13818-2 stream and is extracted from the original MPEG-2 video stream.

5.1.1. Start codes

Start codes are specific bit patterns that do not otherwise occur in the compressed stream format of the MPEG-2 recoding data set.

Each start code consists of a start code prefix followed by a start code value. The start code prefix is a string of 23 bits with the value zero followed by a single bit with the value one.

The start code value is an eight-bit integer that identifies the type of start code. Most types of start code have just one start code value. However, slice_start_code is represented by many start code values. In this case, the start code value is the slice_vertical_position for the slice.

All start codes shall be byte aligned. When necessary, this shall be achieved by inserting bits with the value zero before the start code prefix such that the first bit of the start code prefix is the first (most significant) bit of a byte.

Table 1 defines the start code values for the start codes used in the compressed stream format of MPEG-2 recoding set.

The use of the start codes is defined in the following syntax description with the exception of the sequence_error_code. The sequence_error_code has been allocated for use by a media interface to indicate where uncorrectable errors have been detected (see tables 2 - 22).

5.2 Compressed stream format of the MPEG-2 recoding set semantics

Almost all of the syntax elements in the compressed stream format of the MPEG-2 recoding set have the same meaning as those defined in ISO/IEC 13818-2. Some elements are newly defined and some have a different meaning. Definitions for these elements are given below.

5.2.1 Recoding stream information

Re_coding_stream_info_ID: A 16-bit integer that is used for identification of re_coding_stream_info(). The value is 1001 0001 1110 1100 (0 x 91 ec).

red_bw_flag: A 1-bit flag. Its definition is given in SMPTE 327M.

red_bw_indicator: A 2-bit integer defined in SMPTE 327M.

num_other_bits: A 7-bit integer. This syntax element is defined in SMPTE 327M.

num_mv_bits: An 8-bit integer defined in SMPTE 327M.

num_coef_bits: A 14-bit integer. This syntax element is defined in SMPTE 327M.

num_coef_bits, num_mv_bits, and num_other_bits exist when red_bw_flag is 0. These three data sets are repeated by the number of macroblocks. The data sets match the macroblocks respectively, which are scanned from left top to right bottom horizontally first, as in MPEG-2.

The value of red_bw_indicator specifies the level of reduced information and it ranges from 0 to 3. As the value is increased, the more information is reduced.

As the compressed stream format of the MPEG-2 recoding data set syntax structure depends on this red_bw_flag and red_bw_indicator, the decoder shall recognize these elements in the current bit stream. Then it shall parse the bit stream accordingly.

The details of each red_bw_indicator value are described as follows:

– Indicator 0: This stream transports all the recoding data set, except that the information such as num_coef_bits, num_mv_bits, and num_other_bits are not transported. All the recoding data set could be derived from this stream. Therefore, the position of slice_start, skipped_mb, and mb_quant should be the same as in the original stream.

– Indicator 1: This stream is the same as indicator 0, except that coded_block_pattern() information is not transported. This stream is not guaranteed to have the values of slice_start, skipped_mb, and mb_quant to be the same as in the original stream.

– Indicator 2: This stream transports only quantizer_scale_code information in slice() and macroblock(). Other information which is carried in slice() and macroblock() is not guaranteed to be the same as in the original stream.

– Indicator 3: The information of slice() and macroblock() is not transported by this stream.

The relation between red_bw_indicator and the elements of the MPEG-2 recoding data set described in SMPTE 327M is shown in table 23.

Table 1 – Start code value

Name	Start code value (hexadecimal)
picture_start_code	00
slice_start_code	01 through AF
reserved	B0
reserved	B1
user_data_start_code	B2
sequence_header_code	B3
sequence_error_code	B4
extension_start_code	B5
reserved	B6
sequence_end_code	B7
group_start_code	B8

Table 2 – Compressed stream format of the MPEG-2 recoding set

compressed_stream_format_of_MPEG_2_re_coding_set() {	No. of bits	Mnemonic
next_start_code()		
sequence_header()		
sequence_extension()		
extension_and_user_data(0)		
if (nextbits() == group_start_code) {		
group_of_pictures_header()		
extension_and_user_data(1)		
}		
picture_header()		
picture_coding_extension()		
re_coding_stream_info()		
extensions_and_user_data(2)		
if (! red_bw_flag II (red_bw_indicator <=2))		
picture_data()		
sequence_end_code	32	bslbf
}		

Table 3 – Sequence header

sequence_header() {	No. of bits	Mnemonic
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	bslbf
load_intra_quantizer_matrix	1	uimsbf
if (load_intra_quantizer_matrix)		
intra_quantizer_matrix[64]	8*64	uimsbf
load_non_intra_quantizer_matrix	1	uimsbf
if (load_non_intra_quantizer_matrix)		
non_intra_quantizer_matrix[64]	8*64	uimsbf
next_start_code()		
}		

Table 4 – Sequence extension

sequence_extension() {	No. of bits	Mnemonic
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()		
}		

Table 5 – Extension and user data

extension_and_user_data(i) {	No. of bits	Mnemonic
while ((nextbits() == extension_start_code) II		
(nextbits() == user_data_start_code)) {		
if ((i ! == 1) && (nextbits() == extension_start_code))		
extension_data(i)		
if (nextbits() == user_data_start_code)		
user_data()		
}		
}		

Table 6 – Extension data

extension_data(i) {	No. of bits	Mnemonic
while (nextbits() == extension_start_code) {		
extension_start_code	32	bslbf
if (i == 0) /* follows sequence_extension() */		
sequence_display_extension()		
/* NOTE – i never takes the value 1 because extension_data()		
never follows a group_of_pictures_header() */		
if (i == 2) { /* follows picture_coding_extension() */		
if (nextbits() == "Quant Matrix Extension ID")		
quant_matrix_extension()		
else if (nextbits() == "Copyright Extension ID")		
copyright_extension()		
else		
picture_display_extension()		
}		
}		
}		

Table 7 – Sequence display extension

sequence_display_extension() {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
video_format	3	uimsbf
color_description	1	uimsbf
if (color_description) {		
color_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()		
}		

Table 8 – Quant matrix extension

quant_matrix_extension() {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
load_intra_quantizer_matrix	1	uimsbf
if (load_intra_quantizer_matrix)		
intra_quantizer_matrix[64]	8*64	uimsbf
load_non_intra_quantizer_matrix	1	uimsbf
if (load_non_intra_quantizer_matrix)		
non_intra_quantizer_matrix[64]	8*64	uimsbf
load_chroma_intra_quantizer_matrix	1	uimsbf
if (load_chroma_intra_quantizer_matrix)		
chroma_intra_quantizer_matrix[64]	8*64	uimsbf
load_chroma_non_intra_quantizer_matrix	1	uimsbf
if (load_chroma_non_intra_quantizer_matrix)		
chroma_non_intra_quantizer_matrix[64]	8*64	uimsbf
next_start_code()		
}		

Table 9 – Picture display extension

picture_display_extension() {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
for (i = 0; i<number_of_frame_center_offsets; i++) {		
frame_center_horizontal_offset	16	simsbf
marker_bit	1	bslbf
frame_center_vertical_offset	16	simsbf
marker_bit	1	bslbf
}		
next_start_code()		
}		

Table 10 – Copyright extension

copyright_extension() {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
copyright_flag	1	bslbf
copyright_identifier	8	uimsbf
original_or_copy	1	bslbf
reserved	7	uimsbf
marker_bit	1	bslbf
copyright_number_1	20	uimsbf
marker_bit	1	bslbf
copyright_number_2	22	uimsbf
marker_bit	1	bslbf
copyright_number_3	22	uimsbf
next_start_code()		
}		

Table 11 – User data

user_data() {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
while (nextbits() ! = 0000 0000 0000 0000 0000 0001) {		
user_data	8	uimsbf
}		
next_start_code()		
}		

Table 12 – Group of pictures header

group_of_pictures_header() {	No. of bits	Mnemonic
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code()		
}		

Table 13 – Picture header

picture_header() {	No. of bits	Mnemonic
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	bslbf
forward_f_code	3	bslbf
}		
if (picture_coding_type == 3) {		
full_pel_backward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while (nextbits() == 1) {		
extra_bit_picture /* with the value 1 */	1	uimsbf
extra_information_picture	8	uimsbf
}		
extra_bit_picture /* with the value 0 */	1	uimsbf
next_start_code()		
}		

Table 14 – Picture coding extension

picture_coding_extension() {	No. of bits	Mnemonic
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
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()		
}		

Table 15 – Recoding stream information

re_coding_stream_info() {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
re_coding_stream_info_ID	16	bslbf
red_bw_flag	1	uimsbf
if (red_bw_flag)		
red_bw_indicator	2	uimsbf
if (! red_bw_flag) {		
for (i=0; i<number_of_macroblock; i++) {		
marker_bit	1	bslbf
num_other_bits	7	uimsbf
marker_bit	1	bslbf
num_mv_bits	8	uimsbf
marker_bit	1	bslbf
num_coef_bits	14	uimsbf
}		
}		
next_start_code()		
}		

Table 16 – Picture data

picture_data() {	No. of bits	Mnemonic
do {		
slice()		
} while (nextbits() == slice_start_code)		
next_start_code()		
}		

Table 17 – Slice

slice() {	No. of bits	Mnemonic
slice_start_code	32	bslbf
quantizer_scale_code	5	uimsbf
if (nextbits() == 1) {		
intra_slice_flag	1	bslbf
intra_slice	1	uimsbf
reserved_bits	7	uimsbf
while (nextbits() == 1) {		
extra_bit_slice /* with the value 1 */	1	uimsbf
extra_information_slice	8	uimsbf
}		
}		
extra_bit_slice /* with the value 0 */	1	uimsbf
do {		
macroblock()		
} while (nextbits() ! = 000 0000 0000 0000 0000 0000)		
next_start_code()		
}		

Table 18 – Macroblock

macroblock() {	No. of bits	Mnemonic
while (nextbits() == 0000 0001 000)		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	vlclbf
macroblock_modes()		
if (macroblock_quant)		
quantizer_scale_code	5	uimsbf
if (! red_bw_flag II		
(red_bw_flag && (red_bw_indicator <=1))) {		
if (macroblock_motion_forward II		
(macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if (macroblock_motion_backward)		
motion_vectors(1)		
if (macroblock_intra && concealment_motion_vectors)		
marker_bit	1	bslbf
}		
if (macroblock_pattern &&		
(! red_bw_flag II		
(red_bw_flag && (red_bw_indicator == 0))))		
coded_block_pattern()		
}		

Table 19 – Macroblock modes

macroblock_modes() {	No. of bits	Mnemonic
macroblock_type	1-9	vlclbf
if (! red_bw_flag II		
(red_bw_flag && (red_bw_indicator <=1))) {		
if (macroblock_motion_forward II		
(macroblock_motion_backward) {		
if (picture_structure == frame) {		
if (frame_pred_frame_dct == 0)		
frame_motion_type	2	uimsbf
} else {		
field_motion_type	2	uimsbf
}		
}		
if (picture_structure == Frame picture) &&)		
(frame_pred_frame_dct == 0) &&		
(dct_type_flag == 1) &&		
(! red_bw_flag II		
(red_bw_flag && (red_bw_indicator <= 1)))) {		
dct_type	1	uimsbf
}		
}		

Table 20 – Motion vectors

motion_vectors (s) {	No. of bits	Mnemonic
if (motion_vector_count == 1) {		
if ((mv_format == field) && (dmv ! = 1))		
motion_vertical_field_select[0][s]	1	uimsbf
motion_vector(0,s)		
} else {		
motion_vertical_field_select[0][s]	1	uimsbf
motion_vector(0,s)		
motion_vertical_field_select[1][s]	1	uimsbf
motion_vector(1,s)		
}		
}		

Table 21 – Motion vector

motion_vector (r,s) {	No. of bits	Mnemonic
motion_code[r][s][0]	1-11	vlclbf
if ((f_code[s][0] ! = 1) && (motion_code[r][s][0] ! = 0))		
motion_residual[r][s][0]	1-8	uimsbf
if (dmv == 1)		
dmvector[0]	1-2	vlclbf
motion_code[r][s][1]	1-11	vlclbf
if ((f_code[s][1] ! = 1) && (motion_code[r][s][1] ! = 0))		
motion_residual[r][s][1]	1-8	uimsbf
if (dmv == 1)		
dmvector[1]	1-2	vlclbf
}		

Table 22 – Coded block pattern

coded_block_pattern() {	No. of bits	Mnemonic
coded_block_pattern_420	3-9	vlclbf
if (chroma_format == 4:2:2)		
coded_block_pattern_1	2	uimsbf
if (chroma_format == 4:4:4)		
coded_block_pattern_2	6	uimsbf
}		

Table 23 – Reduced bandwidth indicators

red_bw_indicator_	num_coef_bits, num_mv_bits, num_other_bits	q_scale_code, q_scale_type	motion_type, mv_vert_field_sel[r][s], mv[r][s][t]	mb_mfwd, mb_mbwd	mb_pattern	coded_block_pattern	mb_intra	slice_start	dct_type	mb_quant	skipped_mb
Indicator 0	0	1	1	1	1	1	1	1	1	1	1
Indicator 1	0	1	1	1	1	0	1	X	1	X	X
Indicator 2	0	1	0	X	X	0	X	X	0	X	X
Indicator 3	0	0	0	0	0	0	0	0	0	0	0

NOTE – 0 = This information is not present.
1 = This information is present and corresponds to that of the original data stream.
X = This information may be present, but is not guaranteed to correspond to that of the original data stream.

5.2.2 Picture data

Picture_data() consists of more than zero slice(). When red_bw_indicator is 3, picture data() is not transported by this stream. It means that this compressed stream format of the MPEG-2 recoding set is intended not to transport all syntax elements below picture_data(), but to transport picture_type, etc.

5.2.3 Slice

When red_bw_flag is 0 or red_bw_indicator is less than or equal to 2, the elements in slice() exist. However, the validity of slice_start_code depends on red_bw_indicator.

When red_bw_flag is 0 or red_bw_indicator is equal to 0, slice start information shall be the same as in the original stream.

5.2.4 Macroblock

The elements in macroblock() are varied by red_bw_indicator, although macroblock_escape, macroblock_address_increment, and macroblock_modes() always exist. However, the validity of macroblock_escape and macroblock_address_increment depend on red_bw_flag and red_bw_indicator. When red_bw_flag is 0 or red_bw_indicator is equal to 0, skipped_mb information derived from macroblock_escape and macro-block_address_increment shall be the same as in the original stream.

When red_bw_indicator is of value equal to or more than 2, motion_vectors() shall not exist.

When red_bw_indicator is equal to or more than 1, coded_block_pattern() shall not exist.

5.2.5 Macroblock modes

The elements in macroblock_modes() are varied by red_bw_flag and red_bw_indicator, although macroblock_type always exists.

When red_bw_indicator is equal to or more than 2, frame_motion_type, field_motion_type, and dct_type shall not exist.

The validity of the parameter derived from macroblock_type depends on red_bw_flag and red_bw_indicator.

macroblock_quant – When red_bw_flag is 0 or red_bw_indicator is 0, macroblock_quant is the same as in the original stream.

When red_bw_flag is set to 1 and red_bw_indicator is 1 or 2, this specifies the existence of quantizer_scale_code in macroblock(). The data may not be the same as in the original stream.

macroblock_motion_forward, macroblock_motion_backward – When red_bw_flag is 0 or red_bw_indicator is 0 or 1, macroblock_motion_forward and macroblock_motion_backward have the same meaning as in the original stream. Otherwise, these are not guaranteed

macroblock_pattern – When red_bw_flag is 0 or red_bw_indicator is 0, macroblock_pattern is the same as in the original stream. When red_bw_indicator is 1, this specifies the existence of dct_type. When red_bw_indicator is 2, this information is not guaranteed.

macroblock_intra – When red_bw_flag is 0 or red_bw_indicator is 0 or 1, macroblock_intra is the same as in the original stream. Otherwise, this information is not guaranteed.

Annex A (informative)
Bibliography

SMPTE 308M-1998, Television — MPEG-2 4:2:2 Profile at High Level

SMPTE 328M, Television — MPEG-2 Video Elementary Stream Editing Information

ITU-T H.222.0, ISO/IEC 13818-1:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems