



INTERNATIONAL STANDARD ISO/IEC 13818-2:1996

TECHNICAL CORRIGENDUM 1

Published 1997-12-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Generic coding of moving pictures and associated audio information: Video

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des images animées et du son associé: Vidéo

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO/IEC 13818-2:1996 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 29, Coding of audio, picture, multimedia and hypermedia information.

ICS 35.040

Ref. No. ISO/IEC 13818-2:1996/Cor.1:1997(E)

Descriptors: data processing, moving pictures, image processing, video recording, video data, data converting, coding (data conversion).

© ISO/IEC 1997

Printed in Switzerland

Withdrawn

This page intentionally left blank

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – GENERIC CODING OF MOVING
PICTURES AND ASSOCIATED AUDIO INFORMATION: VIDEO

TECHNICAL CORRIGENDUM 1

Introduction

Each corrigendum item is followed by an informative background.

1) Subclause 6.1.1.6

In “Sequence header” subclause, second last paragraph, replace the sentence:

In the coded bitstream, a repeat sequence header may precede either an I-picture or a P-picture, but not a B-picture.

by:

In the coded bitstream, the first picture following a sequence header or a repeated sequence header shall be either an I-picture or a P-picture, but not a B-picture.

Background

This was an editorial oversight.

2) Subclause 6.2.4

Replace the syntax table for slice by:

slice() {	No. of bits	Mnemonic
slice_start_code	32	bslbf
if (vertical_size > 2800)		
slice_vertical_position_extension	3	uimsbf
if (<sequence_scalable_extension() is present in the bitstream>) {		
if (scalable_mode == 'data partitioning')		
priority_breakpoint	7	uimsbf
}		
quantiser_scale_code	5	uimsbf
if (nextbits() == '1') {		
slice_extension_flag	1	bslbf
intra_slice	1	uimsbf
slice_picture_id_enable	1	uimsbf
slice_picture_id	6	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()		
}		

Background

In some networks that do not have guaranteed bandwidth, e.g. Ethernet, FDDI, large bursts of errors may occur occasionally. If the errors are localized to one picture, then recovery can happen upon reception of the very next slice header. However, if the errors are spread over several pictures, then recovery may take a long time, since the decoder does not know what the decoding parameters are for the next available slice.

If the header information is available from a separately transmitted Data Partitioning layer, or available from the application, then error recovery can be speeded up considerably if the slice header contains information that an application can use to identify which picture the slice belongs to. The **slice_picture_id** provides this information.

3) Subclause 6.3.6

In colour_primaries, after Table 6-6, replace:

In the case that sequence_display_extension() is not present in the bitstream or colour_description is zero the chromaticity is assumed to be that corresponding to colour_primaries having the value 1.

by the following text:

In the case that `sequence_display_extension()` is not present in the bitstream or `colour_description` is zero, the chromaticity is assumed to be implicitly defined by the application.

In transfer_characteristics, after Table 6-8, replace:

In the case that `sequence_display_extension()` is not present in the bitstream or `colour_description` is zero the transfer characteristics are assumed to be those corresponding to transfer_characteristics having the value 1.

by the following text:

In the case that `sequence_display_extension()` is not present in the bitstream or `colour_description` is zero, the transfer characteristics are assumed to be implicitly defined by the application.

After Table 6-9 replace:

In the case that `sequence_display_extension()` is not present in the bitstream or `colour_description` is zero the matrix coefficients are assumed to be those corresponding to matrix_coefficients having the value 1.

by the following text and Note:

In the case that `sequence_display_extension()` is not present in the bitstream or `colour_description` is zero the matrix coefficients are assumed to be implicitly defined by the application.

NOTE – In applications which may have signals with more than one set of colour primaries, transfer characteristics, and/or matrix coefficients, it is recommended to transmit a sequence display extension with `colour_description` set to one, and to specify the appropriate values for the colorimetry parameters.

Background

In applications where only one set of colour primaries, transfer characteristics and matrix coefficients is used, there is not a need for the codec to pass colour description parameters. The current syntax includes default colour description parameters which could cause improper interpretation of many bitstreams.

A solution to this problem which embraces all existing implementations without introducing a new problem for other implementations is not to have a default definition, but instead to let the application define the default, as described in this corrigendum.

4) Subclause 6.3.10

a) Under the description of the syntax element `frame_pred_frame_dct`, replace the statement:

`frame_pred_frame_dct` shall be '1' if `progressive_frame` is '1'.

by:

`frame_pred_frame_dct` shall be '1' if `progressive_sequence` is '1'.

Under the description of the syntax element `progressive_frame`, replace the statement:

- `frame_pred_frame_dct` shall be 1.

by:

- if `progressive_sequence` is equal to one, then `frame_pred_frame_dct` shall be 1.

Background

The following text from MPEG95/044 explains the background for this corrigendum:

It appears that the restriction “`frame_pred_frame_dct` shall be 1 if `progressive_frame` is 1” causes problems when editing bitstreams, as demonstrated by the following example.