INTERNATIONAL STANDARD ISO/IEC 11172-2:1993
TECHNICAL CORRIGENDUM 2

Published 1999-12-01

Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s —

Part 2:
Video

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s —

Partie 2: Vidéo

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO/IEC 11172-2:1993 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

1) Replace subclause 2.4.3.2 (page 25, line 38) by:
"

The value for intra_quantizer_matrix[0] shall always be 8.
".

ICS 681.3.04(084.14)

Ref. No. ISO/IEC 11172-2:1993/Cor.2:1999(E)

Printed in Switzerland

2) Replace subclause 2.4.3.6 (page 30, line 2) by:
"

If macroblock_pattern is zero, cbp=0.
".

3)   Replace subclause 2.4.3.7 (page 30, line 25) by:
"

The number of bits in the following dct_dc_differential code, dct_dc_size_luminance, is derived according to the VLC table B.5a.
".

4)   Replace subclause 2.4.3.7 (page 30, line 29) by:
"

The number of bits in the following dct_dc_differential code, dct_dc_size_chrominance, is derived according to the VLC table B.5b.
".

5)   Replace subclause 2.4.3.7 (page 30, line 32) by:
"

If dct_dc_size_luminance or dct_dc_size_chrominance (as appropriate) is zero, then dct_dc_differential is not present in the bitstream.
".

6)   Replace subclause 2.4.3.7 (page 30, line 35) by:
"

If dct_dc_size_luminance or dct_dc_size_chrominance (as appropriate) is greater than zero, then dct_zz[0] is computed as follows from dct_dc_differential:
".

7)   Replace subclause 2.4.3.7 (page 30, line 37-43) by:
"
For luminance blocks:
        if (dct_dc_differential & (1 << (dct_dc_size_luminance-1))) dct_zz[0] = dct_dc_differential ;
        else dct_zz[0] = ((-1) << (dct_dc_size_luminance)) | (dct_dc_differential+1) ;

For chrominance blocks:
        if (dct_dc_differential & (1 << (dct_dc_size_chrominance-1))) dct_zz[0] = dct_dc_differential ;
        else dct_zz[0] = ((-1) << (dct_dc_size_chrominance)) | (dct_dc_differential+1) ;
".

8) Replace subclause 2.4.3.7 (page 30, table) by:
"
example for dct_dc_size_luminance = 3
".

9)   Replace subclause 2.4.4.4 (page 38, lines 4-5) by:
"

In B-pictures, the skipped macroblock is defined to have the same direction of prediction (forward, backward, or bidirectional) as the prior macroblock, differential motion vectors equal to zero, and no DCT coefficients.
".

10) Replace subclause D.6.5.2 (page 99, line 39) by:
"

Whereas in P-pictures skipped macroblocks have a motion vector equal to zero, in B-pictures skipped macroblocks have the same motion vector and the same direction of prediction (forward, backward, or bidirectional) as the previous macroblock, which cannot be intra coded.
".

11) Replace subclause D.6.7, Table D.22 by:
"

Table D.22 -- Example of the coded data elements needed to generate repeated pictures

Value (bits)	Mnemonic	Length (bits)
0000 0000 0000 0000 0000 0001 0000 0000	picture_start_code	32 bits
xxxx xxxx xx	temporal_reference	10 bits
010	picture_coding_type	3 bits
xxxx xxxx xxxx xxxx	vbv_delay	16 bits
0	full_pel_forward_vector	1 bit
001	forward_f_code	3 bits
0	Extra_bit_picture	1 bit
0000 00	stuffing	6 bits
0000 0000 0000 0000 0000 0001 0000 0001	slice_start_code	32 bits
0000 1	quantizer_scale	5 bits
1	macroblock_address_increment	1 bit
001	macroblock_type	3 bits
0	motion_horizontal_forward_code	1 bit
0	motion_vertical_forward_code	1 bit
0000 0001 000 (x 11)	macroblock_escape (x11)	121 bits
0000 0011 001	macroblock_address_increment	11 bits
001	macroblock_type	3 bits
0	motion_horizontal_forward_code	1 bit
0	motion_vertical_forward_code	1 bit
0	Extra_bit_slice	1 bit
000	stuffing	3 bits
	Total	256 bits

".

12) Replace subclause 2.3 (page 14, line 37) by:
"

2.3.1 Definition of bytealigned function
".

13) Replace subclause 2.3 (page 14, line 42) by:
"

2.3.2 Definition of nextbits function
".

14) Replace subclause 2.3 (page 14, line 46) by:
"

2.3.3 Definition of next_start_code function
".

15) Replace subclause 2.4.1 (page 15, line 5) by:
"

2.4.1.1 Video sequence
".

16) Replace subclause 2.4.1 (page 15, line 22) by:
"

2.4.1.2 Sequence header
".

17) Replace subclause 2.4.1 (page 15, line 27) by:
"
2.4.1.3 Group of pictures
".

18) Replace subclause 2.4.1 (page 16, line 15) by:
"
2.4.1.4 Picture
".

19) Replace subclause 2.4.1 (page 17, line 1) by:
"
2.4.1.5 Slice
".

20) Replace subclause 2.4.1 (page 17, line 10) by:
"
2.4.1.6 Macroblock
".

21) Replace subclause 2.4.1 (page 17, line 23) by:
"
2.4.1.7 Block
".

22) Replace subclause 2.4.1 (page 17, line 30) by:
"
2.4.1.8 Reserved, Forbidden and Marker bit
".

23) Replace subclause D.5.2 (page 68, line 20) by:
"
D.5.2.1 Group of pictures start code
".

24) Replace subclause D.5.2 (page 68, line 32) by:
"
D.5.2.2 Time code
".

25) Replace subclause D.5.2 (page 69, line 1) by:
"
D.5.2.3 Closed GOP
".

26) Replace subclause D.5.2 (page 69, line 24) by:
"
D.5.2.4 Broken link
".

27) Replace subclause D.5.2 (page 69, line 37) by:
"
D.5.2.5 Extension data
".

28) Replace subclause D.5.2 (page 69, line 49) by:
"
D.5.2.6 User data
".