INTERNATIONAL STANDARD ISO/IEC 13818-2:1996
TECHNICAL CORRIGENDUM 2

Published 1997-12-15

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

TECHNICAL CORRIGENDUM 2

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

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO/IEC 13818-2:1995 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.2:1997(E)**

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

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

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

TECHNICAL CORRIGENDUM 2

Introduction

Some corrigenda items are followed by an informative background.

1) Subclause 3.44

Replace "in SI units" *by* "in spatial measurement units such as centimetre"

Background

SI (Système International) units are used by engineers to measure lots of different things. In this case, we are talking only of spatial measurement units such as centimetre. Using the term SI is not necessary and, furthermore, the abbreviation SI is not defined in this Recommendation | International Standard.

2) Subclause 3.78

Add the following definition immediately after 3.78:

3.79 Inverse DCT, IDCT: Inverse discrete cosine transform, as defined in Annex A.

and renumber the subsequent items.

3) Subclause 4.6

At the end of this subclause, after the definition of "vlclbf", *add the following text that clarifies the use of those abbreviations in the syntax clause:*

In the clauses describing the syntax, any syntactic element that can only take positive or unsigned values (such as a flag that can be equal to 0 or 1) is described with the mnemonic 'uimsbf'. If the syntactic element can have a negative value, it is described with the mnemonic 'simsbf'. If the syntactic element has a constant value (e.g. marker_bit) then it is described with the mnemonic 'bslbf'. If the syntactic element represents a variable-length code, it is described with the mnemonic 'vlclbf'.

4) Subclause 6.1.1.4

At the beginning of 6.1.1.4, insert the following text:

A coded picture is made of a picture header, the optional extensions immediately following it and the following picture data. A coded picture may be a coded frame or a coded field.

An I-frame picture or a pair of field pictures, where the first field picture is an I-picture and the second field picture is an I-picture or a P-picture, is called a coded I-frame.

A P-frame picture or a pair of P-field pictures is called a coded P-frame.

A B-frame picture or a pair of B-field pictures is called a coded B-frame.

A coded I-frame, a coded P-frame or a coded B-frame is called a coded frame.

Background

Currently coded I-frame, coded P-frame and coded B-frame are not defined by any text except in clause 3.

5) Subclause 6.2.1

In the fifth paragraph, change "Table 6-1 defines the slice start code values ..." *to* "Table 6-1 defines the start code values ...".

6) Subclause 6.2.2.6

Change the mnemonic for **time_code** *from* "bslbf" *to* "uimsbf".

7) Subclause 6.2.3.6

Change the mnemonic for **original_or_copy** *from* "bslbf" *to* "uimsbf".

Change the mnemonic for **copyright_flag** *from* "bslbf" *to* "uimsbf".

Change the mnemonic for **reserved** *from* "uimsbf" *to* "bslbf".

Background

As of today, only one possible value is permitted for those reserved bits.

8) Subclause 6.2.4

Change the mnemonic for **reserved_bits** *from* "uimsbf" *to* "bslbf".

Background

As of today, only one possible value is permitted for those reserved bits.

9) Subclause 6.2.6

Define the mnemonic for "First DCT coefficient" *and* "Subsequent DCT coefficients" *to be* "vlclbf".

10) Subclause 6.3.9

Replace the definition of **temporal_reference** *with the following:*

temporal_reference – The temporal_reference is a 10-bit unsigned integer associated with each coded picture.

The following simple specification applies only when low_delay is equal to zero.

When a coded frame is in the form of two field pictures, the temporal_reference associated with each picture shall be the same (it is called the temporal_reference of the coded frame). The temporal_reference of each coded frame shall increment by one modulo 1024 when examined in display order at the output of the decoding process, except when a group of pictures header occurs. Among the frames coded after a group of pictures header, the temporal_reference of the coded frame that is displayed first, shall be set to zero.

The following more general specification applies when low_delay is equal to zero or one.

If picture A is not a big picture, i.e. the VBV buffer is only examined once before the coded picture A is removed from the VBV buffer and if N is the temporal_reference of picture A, then the temporal_reference of picture B immediately following picture A in display order is equal to:

- 0 if there is a group of pictures header present between picture A and picture B (in coded order).
- (N + 1) % 1024 if picture B is a frame picture or is the first field of a pair of field pictures.
- N if picture B is the second field of a pair of field pictures.

When low_delay is equal to one, there may be situations where the VBV buffer shall be re-examined several times before removing a coded picture (referred to as a big picture) from the VBV buffer.

If picture A is a big picture and if K is the number of times that the VBV buffer is re-examined as defined in C.7 (K > 0), if N is the temporal_reference of picture A, then the temporal_reference of picture B immediately following picture A in display order is equal to:

- K % 1024 if there is a group of pictures header present between picture A and picture B (in coded order).
- (N + K + 1) % 1024 if picture B is a frame picture or is the first field of a pair of field pictures.
- (N + K) % 1024 if picture B is the second field of a pair of field pictures.

NOTE 1 – If the big picture is the first field of a frame coded with field pictures, then the temporal_reference of the two field pictures of that coded frame are not identical.

Background

A minor technical error was found in 6.3.9, definition of temporal reference in the presence of big pictures. The current specification is incorrect when the big picture happens to be the first field of a pair of field pictures, like in the following example:

The sequence consists of six field pictures.

Picture T1 (first field of the second coded frame) is big.


```
decode     T0  B0  --  --  T1  B2  T3  B3
display      T0      T0      T1      T3      top field
                 B0      B0      B2      B3  bottom field
```

Clearly, in this case, the temporal reference of the bottom field following T1 (T1 is the big field-picture) should be 2 (B2), but according to the current text, it would have to be 3, which is incorrect. The definition of temporal_reference should be changed to handle the case of the first field of a pair of field pictures being a big picture.

In Table 6-12, add a missing space between "IEC" *and* "11172".

11) Subclause 6.3.10

Right before the paragraph for **top_field_first**, *add the following two bullets:*

- vbv_delay;
- temporal_reference.

12) Subclause 6.3.11

Identify by "u" *the horizontal axis and by* "v" *the vertical axis, as in Figure 7-2.*

Background

Axis of the default quantiser matrices are not identified. They should match the convention of 7.3.

Under **intra_quantiser_matrix**, *replace* "The first value shall always be 8." *by* "The first value shall always be 8 (values 1 to 7 and 9 to 255 are reserved)."

Under **chroma_intra_quantiser_matrix**, *replace* "The first value shall always be 8." *by* "The first value shall always be 8 (values 1 to 7 and 9 to 255 are reserved)."

13) Subclause 6.3.17

The first line in the NOTE, add a missing space between IEC *and* 11172.

Remove the second bullet that is misleading:

- The first and last macroblock of a slice shall not be skipped.

Add the following text at the end of 6.3.17 just before 6.3.17.1:

It should be noted that the syntax does not allow the first and last macroblock of a slice to be skipped.

14) Subclause 6.3.17.3

Under **dmvector[t]**, *change* "as described in 7.6.3.1" *to* "as described in 7.6.3.6".

15) Subclause 6.3.17.4

Under "if (chroma_format = = '4:4:4')", *change* "for (i = 8; i < 12; i++)" *to* "for (i = 6; i < 12; i++)".

Two paragraphs before Table 6-20, replace "... defined in Figures 6-8, 6-9 and 6-10" *by* "... defined in Figures 6-10, 6-11 and 6-12".

16) Clause 7

Remove the second paragraph "With the exception of the Inverse Discrete Transform ... this Specification."

Background

The above paragraph belongs to ISO/IEC 13818-4 (Conformance).

This corrigendum provides a solution to a genuine problem identified in ITU-T Rec. H.262 | ISO/IEC 13818-2 (MPEG-2 Video) known as the IDCT internal register overflow problem. The proposed solution is a compromise between various possible solutions discussed by email by members of the Video group.

Without this correction, it is not possible to define unambiguously what is a conformant video bitstream and what is a conformant video decoder in ISO/IEC 13818-4 (Conformance).

The proposed corrigendum clarifies the definition and requirements on the IDCT used by a conforming decoding process.

It was demonstrated that, in some cases, compliant bitstreams cannot be decoded properly by hardware-based decoders that have an IDCT that complies with the requirements currently in ITU-T Rec. H.262 | ISO/IEC 13818-2 (Annex A). Because it is required that a compliant decoder decode properly (i.e. with the arithmetic accuracy requirements of part-2), a compliant bitstream (that has the profile and level of the decoder), this situation is causing a contradiction. The solution to this problem is to change the specification and requirements on the IDCT used by the decoding process.

The correction does not put unreasonable burden on decoder designs to cover cases that are very infrequent, where some internal registers used in some hardware IDCT implementations could overflow.

The correction does not call for non-conformance of bitstreams that would cause such IDCT implementations to overflow. Instead, it just specifies that the output of the decoding process is undefined in this case (i.e. IDCT internal register overflow may happen).

However, the correction introduces a new requirement that IDCT be reasonably precise (i.e. does not cause internal register overflow) for "reasonable" video bitstreams. The correction contains an unambiguous definition of those "reasonable" video bitstreams.

In clause 7, replace the 3rd paragraph:

"The IDCT is defined statistically ... given in Annex A."

by:

"The IDCT function f[y][x] used in the decoding process may be any of several approximations of the saturated mathematical integer-number IDCT defined in Annex A. Requirements on the accuracy of the IDCT function used in the decoding process are specified in Annex A."

Background

See the background information above.

17) Subclause 7.4

In Figure 7-4, change "quant scale code" *to* "quantiser scale code".

18) Subclause 7.5

Replace the paragraph:

Once the DCT coefficients, F[v][u], are reconstructed, the inverse DCT transform defined in Annex A shall be applied to obtain the inverse transformed values, f[y][x]. These values shall be saturated so that:

$$-256 \leq f[y][x] \leq 255, \text{ for all } x, y$$

by:

Once the DCT coefficients, F[v][u] are reconstructed, an IDCT transform that conforms to the specifications of Annex A shall be applied to obtain the inverse transformed values f[y][x].

Background

See the background information of **16**) of this technical corrigendum.

19) Subclause 7.6

In Figure 7-5, replace "Half-pel Prediction Filtering" *by* "Half-sample Prediction Filtering".

Background

The term "pel" is banned from this specification. The term "sample" shall be used instead.

20) Subclause 7.6.3.1

Add the following text at the end of 7.6.3.1:

The vector'[r][s][t] value considered in this subclause is the one obtained from the pseudo code above. In case of dual-prime, this restriction that vector'[r][s][t] shall be in the range [low:high] does not apply to the scaled motion vectors vector'[2:3][0][0:1] defined in 7.6.3.6. Other restrictions on motion vectors, including scaled dual-prime motion vectors are specified in 7.6.3.8 and 8.3.

Background

This correction is to lift an ambiguity in 7.6.3.1.

The restriction to the range [low:high] expressed in 7.6.3.1 applies only to:

- the coded differential motion vector (delta);
- the reconstructed motion vector (vector'[r][s][t]) obtained by the pseudo-code in this subclause (i.e. in case of dual-prime, the reconstructed motion vector before scaling);
- the updated value of the motion vector predictor (PMV[r][s][t]).

The problem is that the text in 7.6.3.1 mentions the vector'[r][s][t] value, which may be confused with the scaled dual-prime motion vectors vector'[r][0][0] and vector'[r][0][1] values defined later in 7.6.3.6.

The only restrictions on the dual-prime motion vectors are:

1) The vertical coordinate of the transmitted field motion vector must be in half of the range supported by the vertical f_code (as described in 7.6.3.2).

2) Field vectors used for dual prime prediction should point inside the reference picture (as described in 7.6.3.8). This implies a restriction on the horizontal coordinate of the dual-prime motion vectors, including the scaled-up vector).

3) There is a restriction on the vertical coordinate of field vectors used for dual prime prediction based on the maximum vertical f-code supported by the profile and level (expressed in Table 8-8 of 8.3 and clarified in technical corrigendum 1).

21) Subclause 7.7

In Figure 7-13, replace "Half-pel Prediction Filtering" *by* "Half-sample Prediction Filtering".

Background

The term "pel" is banned from this specification. The term "sample" shall be used instead.

22) Subclause 7.7.3.3

In the second paragraph, replace "The Table 7-16 defines ..." *by* "Table 7-16 defines ...".

23) Subclause 7.9

In Figure 7-16, replace "Half-pel Prediction Filtering" *by* "Half-sample Prediction Filtering".

Background

The term "pel" is banned from this specification. The term "sample" shall be used instead.

24) Subclause 8.1

Remove the sentence:

When a bitstream conforming to ISO/IEC 11172-2 ... the constrained_parameter_flag shall be set.

Background

This sentence does not belong here and, furthermore, it is confusing, since this is a requirement already defined in ISO/IEC 11172-2.

25) Subclause 8.2

Replace the text in the paragraph right after Table 8-6:

In this context a macroblock is deemed to start with the first bit of the macroblock_address_increment (or macroblock_escape, if any) and continues until the last bit of the "End of block" symbol of the last coded block (or the last bit of the coded_block_pattern() if there are no coded blocks)macroblock() syntactic structure. The bits required to represent any slice() that precedes (or follows) the macroblock are not counted as part of the macroblock.

by the following text:

In this context, a macroblock is deemed to start with the first bit of the macroblock_address_increment (or macroblock_escape, if any) and continues until the last bit of the macroblock() syntactic structure. The bits required to represent any slice() that precedes (or follows) the macroblock are not counted as part of the macroblock.

Background

The text in the paragraph right before Table 8-6 does not work when a macroblock is "MC not-coded".

26) Subclause 8.2.1

Remove the last two paragraphs:

The High profile is also distinguished by having different constraints on luminance sample rate, maximum bit rate and VBV buffer size. Refer to Tables 8-12, 8-13 and 8-14.

Decoders that are Simple profile @ Main level compliant shall be capable of decoding Main profile @ Low level bitstreams.

Background

They do not belong in this subclause.

27) Subclause 8.4.1

Replace garbled text

- Adding 4:2:2 chroma format to a 4:2:0 lower ... Spatial scale.

by the following text:

- Chroma simulcast, which allows to add 4:2:2 chroma information to a 4:2:0 base layer and defined in 7.8 is implemented with SNR scalability.

Add an item before Table 8.10 with the following text:

- (level – 1) is defined as follows:
  - if level is Main, (level – 1) is Low;
  - if level is High – 1440, (level – 1) is Main;
  - if level is High, (level – 1) is High – 1440.

28) Subclause 8.5

In Table 8-15, replace "ISO/IEC 11172" *by* "ISO/IEC 11172-2".

29) Subclause 8.6

A new subclause 8.6 with the following text should be created between Table 8-14 and Table 8-15:

8.6 Compatibility requirements on decoders

Table 8-15 defines the requirements on compatibility for decoders. There is a requirement that a decoder of a profile and level represented by a column in Table 8-15 be capable of decoding correctly all bitstreams with profile and level indication marked by an X in the column. In case of scalable hierarchy of bitstreams, the profile and level indication are that of the upper layer.

30) Annex A

Change the title from "Discrete cosine transform" *to* "Inverse discrete cosine transform".

Background

Annex A defines normative requirement for the IDCT. The definition of DCT in Annex A is purely informative.

Before "The inverse DCT (IDCT) is defined as:", *insert the following paragraph:*

The definition of the DCT (also called forward DCT) is purely informative. Forward DCT is not used by the decoding process described by this specification.

Background

See the background information of **16)** of this technical corrigendum.

Replace "The inverse DCT (IDCT) is defined as:" *by* "The mathematical real-number IDCT is defined as:".

Background

See the background information of **16)** of this technical corrigendum.

Remove the two paragraphs after the formula "f(x, y) = 2/N ..." *and add the following text after the formula:*

f(x, y) is a real number.

The mathematical integer-number IDCT is defined as:

$$f'(x, y) = \text{round}(f(x, y))$$

where round() is the rounding to the nearest integer, with half-integer values rounded away from zero. No clamping or saturation is performed.

The saturated mathematical integer-number IDCT is defined as:

$$f''(x, y) = \text{saturate}(f'(x, y))$$

where saturate() is the saturation in the range [–256, 255], defined as:

$$\text{saturate}(x) = \begin{cases} -256 & x < -256 \\ 255 & x > 255 \\ x & -256 \le x \le 255 \end{cases}$$

The IDCT function f[y][x] used in the decoding process may be any of several approximations of the saturated mathematical integer-number IDCT f "(x, y), provided that it meets all of the following requirements:

1) The IDCT function f[y][x] used in the decoding process shall have values always in the range [–256, 255].

2) The IDCT function f[y][x] used in the decoding process shall conform to the IEEE Standard Specification for the implementation of 8 by 8 Inverse Discrete Cosine Transform, Std 1180-1990, 6 December 1990.

3) This item applies only when input blocks of DCT coefficients cause all the 64 values output of the mathematical integer-number IDCT f '(x, y) to be in the range [–384, 383]. When f '(x, y) > 256, f[y][x] shall be equal to 255 and when f '(x, y) < –257, f[y][x] shall be equal to –256. For all values of f '(x, y) in the range [–257, 256] the absolute difference between f[y][x] and f "(x, y) shall not be larger than 2.

4) Let F be the set of 4096 blocks Bi[y][x] (i = 0 … 4095) defined as follows:

   a) Bi[0][0] = i – 2048.

   b) Bi[7][7] = 1 if Bi[0][0] is even, Bi[7][7] = 0 if Bi[0][0] is odd.

   c) All other coefficients Bi[y][x] other than Bi[0][0] and Bi[7][7] are equal to 0.

   For each block Bi[y][x] that belongs to set F defined above, an IDCT that conforms to this specification (see Annex A) shall output a block f[y][x] that has a peak error of 1 or less compared to the reference saturated mathematical integer-number IDCT f "(x, y). In other words, | f[y][x] –f "(x, y) | shall be < = 1 for all x and y.

In addition to these requirements, the following is a recommendation on the accuracy of the IDCT function f[y][x].

5) When item 3) does not apply, i.e. for input blocks of DCT coefficients causing the output of the mathematical integer-number IDCT f '(x, y) to contain one or more values out of the range [–384, 383], it is desirable that f[y][x] be as close as possible to f "(x, y) for all bitstreams produced by reasonably well-designed encoders.