INTERNATIONAL STANDARD ISO/IEC 15444-1:2000
TECHNICAL CORRIGENDUM 2

Published 2002-01-15

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

Information technology — JPEG 2000 image coding system —

Part 1:
Core coding system

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Système de codage d'image JPEG 2000 —

Partie 1: Système de codage de noyau

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to International Standard ISO/IEC 15444-1:2000 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 15444-1:2000/Cor.2:2002(E)**

INFORMATION TECHNOLOGY – JPEG 2000 IMAGE CODING SYSTEM: CORE CODING SYSTEM

TECHNICAL CORRIGENDUM 2

1) In the List of Tables, Table A-16 and Table A-15, change the order of Table A-15 and Table A-16.

2) In the List of Tables, Table A-16, change
"Progression order for the SPcod, SPcoc, and Ppoc parameters"
to
"Progression order for the SGcod, SPcoc, and Ppoc parameters"

3) In Annex A.6.1, Table A-17, change
"Multiple component transformation for the SPcod parameters"
to
"Multiple component transformation for the SGcod parameters"

4) In List of Tables, add "Table J-25 Recommended frequency weighting for multiple component (colour) images"

5) In clause 3.99, change
"tile-part index: The index of the current tile-part ranging from zero to the number of tile-parts minus in a given tile."
to
"tile-part index: The index of the current tile-part ranging from zero to the number of tile-parts minus one in a given tile."

6) In Annex A.2, Table A-2, the footnotes markers in the table do not correspond to the footnotes. The correct table has the below footnoting.

Table A-2 — List of markers and marker segments

	Symbol	Code	Main header	Tile-part header
Delimiting markers and marker segments				
Start of codestream	SOC	0xFF4F	required[a]	not allowed
Start of tile-part	SOT	0xFF90	not allowed	required
Start of data	SOD	0xFF93	not allowed	last marker
End of codestream	EOC	0xFFD9	not allowed	not allowed
Fixed information marker segments				
Image and tile size	SIZ	0xFF51	required	not allowed

Table A-2 — List of markers and marker segments *(continued)*

[a]	Symbol	Code	Main header	Tile-part header
Functional marker segments				
Coding style default	COD	0xFF52	required	optional
Coding style component	COC	0xFF53	optional	optional
Region-of-interest	RGN	0xFF5E	optional	optional
Quantization default	QCD	0xFF5C	required	optional
Quantization component	QCC	0xFF5D	optional	optional
Progression order change[b]	POC	0xFF5F	optional	optional
Pointer marker segments				
Tile-part lengths	TLM	0xFF55	optional	not allowed
Packet length, main header	PLM	0xFF57	optional	not allowed
Packet length, tile-part header	PLT	0xFF58	not allowed	optional
Packed packet headers, main header[c]	PPM	0xFF60	optional	not allowed
Packed packet headers, tile-part header[c]	PPT	0xFF61	not allowed	optional
In bit stream markers and marker segments				
Start of packet	SOP	0xFF91	not allowed	not allowed in tile-part header, optional in bit stream
End of packet header	EPH	0xFF92	optional inside PPM marker segment	optional inside PPT marker segment or in bit stream
Informational marker segments				
Component registration	CRG	0xFF63	optional	not allowed
Comment	COM	0xFF64	optional	optional

a. Required means the marker or marker segment shall be in this header, optional means it may be used.
b. The POC marker segment is required if there are progression order changes.
c. Either the PPM or PPT marker segment is required if the packet headers are not distributed in the bit stream. If the PPM marker segment is used then PPT marker segments shall not be used, and vice versa.

7) In Annex A.4.2, Table A-5, Psot value, correct

"12 — (2^{32}-1)"

to

"0, or 14 — (2^{32}-1)"

8) In Annex A.5.1, Lsiz definition, Equation A.1, correct
"Lcod"
to
"Lsiz"

9) In Annex A.6.1, Table A-16, table title, change
"Progression order for the SPcod, SPcoc, and Ppoc parameters"
to
"Progression order for the SGcod, SPcoc, and Ppoc parameters"

10) In Annex A.6.1, Table A-17, table title, correct
"Multiple component transformation for the SPcod parameters"
to
"Multiple component transformation for the SGcod parameters"

11) In Annex A.6.2, Figure A-11, add the definition "F^i-F^n — Precinct size"

12) In Annex A.6.6, Table A-32, correct
"$RSpoc^i$", "8", "0 — 33"
to
"$RSpoc^i$", "8", "0 — 32"

13) In Annex A.7.1, Table A-33, lower right cell, correct
"13 — 65 535
13 — $(2^{32}-1)$"
to
"14 — 65 535
14 — $(2^{32}-1)$"

14) In Annex A.8.1, Usage first paragraph, second sentence, correct
"Shall only be used if indicated in the proper COD marker segment (see Annex A.6.1)."
to
"Shall not be used unless indicated that it is allowed in the proper COD marker segment (see Annex A.6.1)."

15) In Annex B.10.3, after the first paragraph, there could be ambiguity about the functioning of the code-block inclusion tag tree. After the last paragraph add,
"Note — If a packet is marked as empty then no code-blocks may contribute to the corresponding layer. If the next packet is not marked as empty, the code-block inclusion information (defined in Annex B.10.4) for the previous layer with the empty bit set has to be included. The code-block inclusion information for code-blocks which have not yet been included in any packet is encoded using a tag tree whose entries are initialized with the layer number of the first layer to which the code-block contributes. Thus, the tag tree will have redundant information identifying whether or not the code-block contributes to both the current layer and the layer in which the packet was marked as empty."

16) In Annex B.10.4, third paragraph, first sentence, change
"For code-blocks that have not been previously included in any packet, this information is signalled with a separate tag tree code for each precinct."
to
"For code-blocks that have not been previously included in any packet, this information is signalled with a separate tag tree code for each precinct as confined to a subband."

17) In Annex G.1.1, Equation G.1, change

"$I(x,y) \leftarrow I(x,y) - 2^{Ssiz^i - 1}$ "

to

"$I(x,y) \leftarrow I(x,y) - 2^{Ssiz^i}$ "

18) In Annex G.1.2, Equation G.2, change

"$I(x,y) \leftarrow I(x,y) + 2^{Ssiz^i - 1}$ "

to

"$I(x,y) \leftarrow I(x,y) + 2^{Ssiz^i}$ "

19) In Annex I.5.1, third paragraph, delete the sentence

"The control-Z character in the type stops file display under MS-DOS."

20) In Annex I.5.3.1, BPC definition, second paragraph, change

"If the bit depth is the same for all components, then this parameter specifies that bit depth and shall be equivalent to the values of the $Ssiz^i$ fields in the SIZ marker in the codestream (which shall all be equal). If the components vary in bit depth, then the value of this field shall be 255 and the JP2 Header box shall also contain a Bits Per Component box defining the bit depth of each component (as defined in Annex I.5.3.2)."

to

"If the bit depth and the sign are the same for all components, then this parameter specifies that bit depth and shall be equivalent to the values of the $Ssiz^i$ fields in the SIZ marker in the codestream (which shall all be equal). If the components vary in bit depth and/or sign, then the value of this field shall be 255 and the JP2 Header box shall also contain a Bits Per Component box defining the bit depth of each component (as defined in Annex I.5.3.2)."

21) In Annex I.5.3.4, change "

I.5.3.4 Palette box

The palette specified in this box is applied to a single component to convert it into multiple components. The colourspace of the components generated by the palette is then interpreted based on the values of the Colour Specification boxes in the JP2 Header box in the file. The mapping of an actual component from the codestream through the palette is specified in the Component Mapping box. If the JP2 Header box contains a Palette box, then it shall also contain a Component Mapping box. If the JP2 Header box does not contain a Palette box, then it shall not contain a Component Mapping box.

The type of the Palette box shall be 'pclr' (0x7063 6C72). The contents of this box shall be as follows:

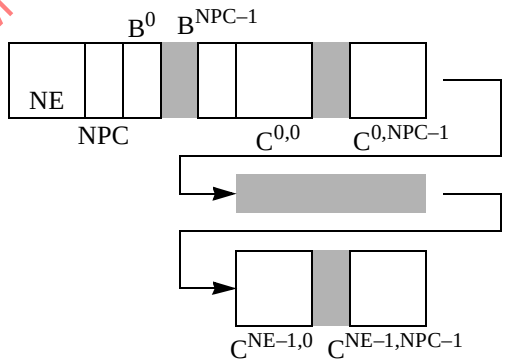

Figure I-11 — Organization of the contents of the Palette box

NE: Number of entries in the table. This value shall be in the range 1 to 1 024 and is encoded as a 2-byte big endian unsigned integer.

NPC: Number of components created by the application of the palette. For example, if the palette turns a single index component into a three-component RGB image, then the value of this field shall be 3. This field is encoded as a 1-byte unsigned integer.

B^i: This parameter specifies the bit depth of generated component *i*, encoded as a 1-byte big endian integer. The low 7-bits of the value indicate the bit depth of this component. The high-bit indicates whether the component is signed or unsigned. If the high-bit is 1, then the component contains signed values. If the high-bit is 0, then the component contains unsigned values. The number of B^i values shall be the same as the value of the NPC field.

C^{ji}: The generated component value for entry *j* for component *i*. C^{ji} values are organized in component major order; all of the component values for entry *j* are grouped together, followed by all of the entries for component *j*+1. The size of C^{ji} is the value specified by field B^i. The number of components shall be the same as the NPC field. The number of C^{ji} values shall be the number of created components (the NPC field) times the number of entries in the palette (NE). If the value of B^i is not a multiple of 8, then each C^{ji} value is padded with zeros to a multiple of 8 bits and the actual value shall be stored in the low-order bits of the padded value. For example, if the value of B^i is 10 bits, then the individual C^{ji} values shall be stored in the low 10 bits of a 16 bit field.

Table I-12 — Format of the contents of the Palette box

Field name	Size (bits)	Value
NE	16	1 — 1 024
NPC	8	1 — 255
B^i	8	See Table Table I-13 —
C^{ji}	Variable	Variable

Table I-13 — B^i values

Values (bits) MSB LSB	Component sample precision
x000 0000 — x010 0101	Component bit depth = value + 1. From 1 bit deep through 38 bits deep respectively (counting the sign bit, if appropriate)
0xxx xxxx	Components are unsigned values
1xxx xxxx	Components are signed values
	All other values reserved for ISO use.

"to"

I.5.3.4 Palette box

This box specifies a palette that can be used to create channels from components. However, the Palette box does not specify the creation of any particular channel; the creation of channels based on the application of the palette to a component is specified by the Component Mapping box. The colourspace or meaning of the generated channel is specified by the Channel Definition box (or specified through the defaults defined in the specification of the Channel Definition box if the Channel Definition box does not exist). If the JP2 Header box contains a Palette box, then it shall also contain a Component Mapping box. If the JP2 Header box does not contain a Palette box, then it shall not contain a Component Mapping box.

The type of the Palette box shall be ‘pclr’ (0x7063 6C72). The contents of this box shall be as follows:

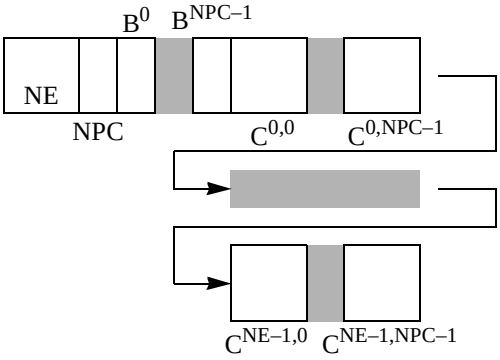

Figure I-11 — Organization of the contents of the Palette box

NE: Number of entries in the table. This value shall be in the range 1 to 1 024 and is encoded as a 2-byte big endian unsigned integer.

NPC: Number of palette columns specified in the Palette box. For example, if the palette is to be used to map a single index component into a three-component RGB image, then the value of this field shall be 3. This field is encoded as a 1-byte unsigned integer.

B^i: This parameter specifies the bit depth of values created by palette column *i*, encoded as a 1-byte big endian integer. The low 7-bits of the value indicate the bit depth of this palette column. The high-bit indicates whether the palette column is signed or unsigned. If the high-bit is 1, then the palette column contains signed values. If the high-bit is 0, then the palette column contains unsigned values. The number of B^i values shall be the same as the value of the NPC field.

C^{ji}: The value for entry *j* for palette column *i*. C^{ji} values are organized in entry major order; all of the values for entry *j* are grouped together, followed by all of the values for entry *j*+1. In the example given above, this table would therefore read $R_1, G_1, B_1, R_2, G_2, B_2$, etc. The size of C^{ji} is the value specified by field B^i. The number of palette columns shall be the same as the NPC field. The number of C^{ji} values shall be the number of palette columns (the NPC field) times the number of entries in the palette (NE). If the value of B^i is not a multiple of 8, then each C^{ji} value is padded with zeros to a multiple of 8 bits and the actual value shall be stored in the low-order bits of the padded value. For example, if the value of B^i is 10 bits, then the individual C^{ji} values shall be stored in the low 10 bits of a 16 bit field.

Table I-12 — Format of the contents of the Palette box

Field name	Size (bits)	Value
NE	16	1 — 1 024
NPC	8	1 — 255
B^i	8	See Table Table I-13 —
C^{ji}	Variable	Variable