INTERNATIONAL STANDARD

ISO/IEC 23094-1

First edition
2020-10

AMENDMENT 1
2023-08

Information technology — General video coding —

Part 1: Essential video coding

AMENDMENT 1: Green metadata supplemental enhancement information

Technologies de l'information — Codage vidéo général —

Partie 1: Codage vidéo essentiel

AMENDEMENT 1: Informations d'amélioration complémentaires des métadonnées vertes

Reference number
ISO/IEC 23094-1:2020/Amd. 1:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Information technology — General video coding —

Part 1:
Essential video coding

AMENDMENT 1: Green metadata supplemental enhancement information

Clause 2

Add the following normative references:

Rec. ITU-T H.273 | ISO/IEC 23091-2, *Information technology — Coding-independent code points — Part 2: Video*

ISO/IEC 23001-11, *Information Technology — MPEG Systems technologies — Part 11: Energy-efficient media consumption (green metadata)*

3.71, Note 1 to entry

Replace "Annex A" with "Annex B".

6.4.4

Delete "The neighbouring block is coded in intra or intra block copy mode".

7.2

Delete "(i.e., the position of the payload_bit_equal_to_one syntax element)".

7.3.2.8

Delete the following:

rbsp_stop_one_bit /* equal to 1 */	f(1)

7.3.2.9

Delete the following:

alignment_bit_equal_to_one /* equal to 1 */	f(1)

7.3.8.3

Replace:

coding_unit(x0, y0, log2CbWidth, log2CbHeight, ctDepth, cuQpDeltaCode)	

with:

coding_unit(x0, y0, log2CbWidth, log2CbHeight, ctDepth, cuQpDeltaCode, treeType, predModeConstraint)	

7.3.8.4

Add a comma after "treeType" in the first line.

7.3.8.5

Replace:

if((isSplit | | CuPredMode = = MODE_INTRA | | cbf_cb | | cbf_cr) && treeType != DUAL_TREE_CHROMA)	

with:

if(((isSplit | | CuPredMode = = MODE_INTRA | | cbf_cb | | cbf_cr) && treeType == SINGLE_TREE) | | treeType = = DUAL_TREE_LUMA)	

7.4.3.8

Delete "**rbsp_stop_one_bit** shall be equal to 1."

7.4.3.9

Delete "**alignment_bit_equal_to_one** shall be equal to 1."

7.4.9.3

After the bullet "log2CbWidthTemp is less than 2" add another bullet with text

"ratioWH is greater than 2".

After the bullet "log2CbHeightTemp is less than 2" add another bullet with text "ratioWH is greater than 2".

7.4.9.4

Replace "PRED_MODE_CONSTRAINT_INTRA" with "PRED_MODE_CONSTRAINT_INTRA_IBC".

8.3.1

Replace, in Formula (163), "/" with "÷".

8.5.1

Replace “RefPicList0[refIdxL0]” with “RefPicList[0][refIdxL0]”.

Replace “RefPicList1[refIdxL1]” with “RefPicList[1][refIdxL1]”.

8.5.2.1

Delete “Let the variable LX be RefPicListX, with X being 0 or 1, of the current picture”.

8.5.2.3

Replace “RefPicListX” with "RefPicList[X]”.

8.5.2.3.9

Replace “RefList0” with “RefPicList[0]”.

Replace “RefPicList[1]” with “RefPicList[1][1]”.

Replace “currPocDiffL0” with “currPocDiffL0_0”.

Replace “currPocDiffL1” with “currPocDiffL0_1”.

Replace “RefPicList0” with “RefPicList[0]”.

Replace “RefPicList1” with “RefPicList[1]”.

8.5.2.4.1

Replace “RefPicListX” with “RefPicList[X]”.

8.5.2.4.3

Replace “RefPicListX” with ”RefPicList[X]”.

8.5.2.5

Replace “RefPicList0” with “RefPicList[0]”.

Replace “RefPicList1” with “RefPicList[1]”.

8.5.4.2

Replace “RefPicListX” with ”RefPicList[X].

8.7.3

Replace, in Formulae (1056) and (1057), “–5” with “–9”.

8.7.4.2

Add the following input parameter:

— a variable trType specifying the transform kernel type.

In the first paragraph, replace “Type” with “type”.

8.8.2.3

Replace, before Formula (1136), the following:

— If refIdx0L0 is equal to refIdx0L1 and refIdx1L0 is equal to refIdx1L1, the following applies:

with:

— If RefPicList[0][refIdx0L0] is equal to RefPicList[0][refIdx1L0] and RefPicList[1][refIdx0L1] is equal to RefPicList[1][refIdx1L1], the following applies:

Replace, before Formula (1137), the following:

— Otherwise, if refIdx0L0 is equal to refIdx1L1 and refIdx0L1 is equal to refIdx1L0, the following applies:

with:

— Otherwise, if RefPicList[0][refIdx0L0] is equal to RefPicList[1][refIdx1L1] and RefPicList[1][refIdx0L1] is equal to RefPicList[0][refIdx1L0], the following applies:

Replace, before Formula (1191), the following:

— If refIdx0L0 is equal to refIdx0L1 and refIdx1L0 is equal to refIdx1L1, the following applies:

with:

— If RefPicList[0][refIdx0L0] is equal to RefPicList[0][refIdx1L0] and RefPicList[1][refIdx0L1] is equal to RefPicList[1][refIdx1L1], the following applies:

Replace, before Formula (1192), the following:

— Otherwise, if refIdx0L0 is equal to refIdx1L1 and refIdx0L1 is equal to refIdx1L0, the following applies:

with:

— Otherwise, if RefPicList[0][refIdx0L0] is equal to RefPicList[1][refIdx1L1] and RefPicList[1][refIdx0L1] is equal to RefPicList[0][refIdx1L0], the following applies:

8.8.3.4

Replace, after Formula (1229), the following:

— If refIdx0L0 is equal to refIdx1L0 and refIdx0L1 is equal to refIdx1L1 or if refIdx0L0 is equal to refIdx1L1 and refIdx0L1 is equal to refIdx1L0, the variable bS is derived as follows:

with:

— If RefPicList[0][refIdx0L0] is equal to RefPicList[0][refIdx1L0] and RefPicList[1][refIdx0L1] is equal to RefPicList[1][refIdx1L1], or if RefPicList[0][refIdx0L0] is equal to RefPicList[1][refIdx1L1] and RefPicList[1][refIdx0L1] is equal to RefPicList[0][refIdx1L0], the variable bS is derived as follows:

Replace, before Formula (1230), the following:

— If refIdx0L0 is equal to refIdx0L1, the variable bS is derived by:

with:

— If RefPicList[0][refIdx0L0] is equal to RefPicList[1][refIdx0L1], the variable bS is derived by:

Replace, before Formula (1231), the following:

— Otherwise, if refIdx0L0 is equal to refIdx1L0 and refIdx0L1 is equal to refIdx1L1, the variable bS is derived by:

with:

— Otherwise, if RefPicList[0][refIdx0L0] is equal to RefPicList[0][refIdx1L0] and RefPicList[1][refIdx0L1] is equal to RefPicList[1][refIdx1L1], the variable bS is derived by:

Replace, before Formula (1232), the following:

— Otherwise, if refIdx0L0 is equal to refIdx1L1 and refIdx0L1 is equal to refIdx1L0, the variable bS is derived by:

with:

— Otherwise, if RefPicList[0][refIdx0L0] is equal to RefPicList[1][refIdx1L1] and RefPicList[1][refIdx0L1] is equal to RefPicList[0][refIdx1L0], the variable bS is derived by:

8.8.4.5

Replace “Otherwise, (availableL is equal to TRUE), samples recPictureOut are derived as follows” with “Otherwise (availableL is equal to TRUE), for y = 0..blkHeight − 1, samples recPictureOut are derived as follows”.

Replace “Otherwise (availableR is equal to TRUE), samples recPictureOut are derived as follows” with “Otherwise (availableR is equal to TRUE), for y = 0..blkHeight − 1, samples recPictureOut are derived as follows”.

Replace “If availableT is equal to FALSE, for x = −3..blkWidth + 2 and, samples recPictureOut are derived as follows” with “If availableT is equal to FALSE, for x = −3..blkWidth + 2, samples recPictureOut are derived as follows”.

Replace Formulae (1337), (1338) and (1339) with the following:

recPictureOut[xCtb + x, yCtb − 3] = recPictureOut[xCtb + x, yCtb + 3] (1337)

recPictureOut[xCtb + x, yCtb − 2] = recPictureOut[xCtb + x, yCtb + 2] (1338)

recPictureOut[xCtb + x, yCtb − 1] = recPictureOut[xCtb + x, yCtb + 1] (1339)

Replace "Otherwise (availableT is equal to TRUE), samples recPictureOut are derived as follows" with "Otherwise (availableT is equal to TRUE), for x = 0..blkWidth − 1, samples recPictureOut are derived as follows".

Add the following after Formula (1342):

— If availableL is equal to FALSE, for y = yCtb − 3..yCtb − 1, samples recPictureOut are derived as follows:

recPictureOut[xCtb − 3, y] = recPicture[xCtb, y] (1477)

recPictureOut[xCtb − 2, y] = recPicture[xCtb, y] (1478)

recPictureOut[xCtb − 1, y] = recPicture[xCtb, y] (1479)

— Otherwise (availableL is equal to TRUE), for y = yCtb − 3..yCtb − 1, samples recPictureOut are derived as follows:

recPictureOut[xCtb − 3, y] = recPicture[xCtb − 3, y] (1480)

recPictureOut[xCtb − 2, y] = recPicture[xCtb − 2, y] (1481)

recPictureOut[xCtb − 1, y] = recPicture[xCtb − 1, y] (1482)

— If availableR is equal to FALSE, for y = yCtb − 3..yCtb − 1, samples recPictureOut are derived as follows:

recPictureOut[xCtb + blkWidth + 2, y] =
recPicture[xCtb + blkWidth − 1, y] (1483)

recPictureOut[xCtb + blkWidth + 1, y] =
recPicture[xCtb + blkWidth − 1, y] (1484)

recPictureOut[xCtb + blkWidth, y] =
recPicture[xCtb + blkWidth − 1, y] (1485)

— Otherwise (availableR is equal to TRUE), for y = yCtb − 3..yCtb − 1, samples recPictureOut are derived as follows:

recPictureOut[xCtb + blkWidth + 2, y] =
recPicture[xCtb + blkWidth + 2, y] (1486)

recPictureOut[xCtb + blkWidth + 1, y] =
recPicture[xCtb + blkWidth + 1, y] (1487)

recPictureOut[xCtb + blkWidth, y] =
recPicture[xCtb + blkWidth, y] (1488)

Replace "If availableB is equal to FALSE, for x = −3..blkWidth + 2 and, samples recPictureOut are derived as follows" with "If availableB is equal to FALSE, for x = −3..blkWidth + 2, samples recPictureOut are derived as follows".

Replace Formulae (1343) to (1345) with the following:

recPictureOut[xCtb + x, yCtb + blkHeight + 2] =
recPictureOut[xCtb + x, yCtb + blkHeight − 4] (1343)

recPictureOut[xCtb + x, yCtb + blkHeight + 1] =
recPictureOut[xCtb + x, yCtb + blkHeight − 3] (1344)

recPictureOut[xCtb + x, yCtb + blkHeight] =
recPictureOut[xCtb + x, yCtb + blkHeight − 2] (1345)

Replace "Otherwise (availableB is equal to TRUE), samples recPictureOut are derived as follows" with "Otherwise (availableB is equal to TRUE), for x = 0..blkWidth − 1, samples recPictureOut are derived as follows".

Add the following after Formula (1348):

— If availableL is equal to FALSE, for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureOut are derived as follows:

recPictureOut[xCtb − 3, y] = recPicture[xCtb, y] (1489)

recPictureOut[xCtb − 2, y] = recPicture[xCtb, y] (1490)

recPictureOut[xCtb − 1, y] = recPicture[xCtb, y] (1491)

— Otherwise (availableL is equal to TRUE), for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureOut are derived as follows:

recPictureOut[xCtb − 3, y] = recPicture[xCtb − 3, y] (1492)

recPictureOut[xCtb − 2, y] = recPicture[xCtb − 2, y] (1493)

recPictureOut[xCtb − 1, y] = recPicture[xCtb − 1, y] (1494)

— If availableR is equal to FALSE, for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureOut are derived as follows:

recPictureOut[xCtb + blkWidth + 2, y] =
recPicture[xCtb + blkWidth − 1, y] (1495)

recPictureOut[xCtb + blkWidth + 1, y] =
recPicture[xCtb + blkWidth − 1, y] (1496)

recPictureOut[xCtb + blkWidth, y] =
recPicture[xCtb + blkWidth − 1, y] (1497)

— Otherwise (availableR is equal to TRUE), for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureOut are derived as follows:

recPictureOut[xCtb + blkWidth + 2, y] =
recPicture[xCtb + blkWidth + 2, y] (1498)

recPictureOut[xCtb + blkWidth + 1, y] =
recPicture[xCtb + blkWidth + 1, y] (1499)

recPictureOut[xCtb + blkWidth, y] =
recPicture[xCtb + blkWidth, y] (1500)

8.8.4.6

Replace "Otherwise (availableL is equal to TRUE)," with "Otherwise (availableL is equal to TRUE), for y = 0..blkHeight − 1, samples recPictureChOut are derived as follows:".

Replace "Otherwise (availableR is equal to TRUE)," with "Otherwise (availableR is equal to TRUE), for y = 0..blkHeight − 1, samples recPictureChOut are derived as follows:".

Replace "If availableT is equal to FALSE, and loop_filter_across_tiles_enabled_flag is equal to FALSE, for x = −3..blkWidth + 2 and, samples recPictureChOut are derived as follows" with "If availableT is equal to FALSE, for x = −3..blkWidth + 2, samples recPictureChOut are derived as follows".

Replace Formulae (1362) to (1364) with the following:

recPictureChOut[xCtb + x, yCtb − 3] =
recPictureChOut[xCtb + x, yCtb + 3] (1362)

recPictureChOut[xCtb + x, yCtb − 2] =
recPictureChOut[xCtb + x, yCtb + 2] (1363)

recPictureChOut[xCtb + x, yCtb − 1] =
recPictureChOut[xCtb + x, yCtb + 1] (1364)

Replace "Otherwise (availableT is equal to TRUE)," with "Otherwise (availableT is equal to TRUE), for x = 0..blkWidth − 1, samples recPictureChOut are derived as follows:".

Add the following after Formula (1367):

— If availableL is equal to FALSE, for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb − 3, y] = recPictureCh[xCtb, y] (1501)

recPictureChOut[xCtb − 2, y] = recPictureCh[xCtb, y] (1502)

recPictureChOut[xCtb − 1, y] = recPictureCh[xCtb, y] (1503)

— Otherwise (availableL is equal to TRUE), for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb − 3, y] = recPictureCh[xCtb − 3, y] (1504)

recPictureChOut[xCtb − 2, y] = recPictureCh[xCtb − 2, y] (1505)

recPictureChOut[xCtb − 1, y] = recPictureCh[xCtb − 1, y] (1506)

— If availableR is equal to FALSE, for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb + blkWidth + 2, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1507)

recPictureChOut[xCtb + blkWidth + 1, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1508)

recPictureChOut[xCtb + blkWidth, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1509)

— Otherwise (availableR is equal to TRUE), for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb + blkWidth + 2, y] =
recPictureCh[xCtb + blkWidth + 2, y] (1510)

recPictureChOut[xCtb + blkWidth + 1, y] =
recPictureCh[xCtb + blkWidth + 1, y] (1511)

recPictureChOut[xCtb + blkWidth, y] =
recPictureCh[xCtb + blkWidth, y] (1512)

Replace “If availableB is equal to FALSE, for x = −3..blkWidth + 2 and, samples recPictureChOut are derived as follows” with “If availableB is equal to FALSE, for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows”.

Replace Formulae (1368) to (1370) with the following:

recPictureChOut[xCtb + x, yCtb + blkHeight + 2] =
recPictureChOut[xCtb + x, yCtb + blkHeight − 4] (1368)

recPictureChOut[xCtb + x, yCtb + blkHeight + 1] =
recPictureChOut[xCtb + x, yCtb + blkHeight − 3] (1369)

recPictureChOut[xCtb + x, yCtb + blkHeight] =
recPictureChOut[xCtb + x, yCtb + blkHeight − 2] (1370)

Replace “Otherwise (availableB is equal to TRUE),” with “Otherwise (availableB is equal to TRUE), for y = yCtb − 3..yCtb − 1, samples recPictureChOut are derived as follows:”.

Add the following after Formula (1373):

— If availableL is equal to FALSE, for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb − 3, y] = recPictureCh[xCtb, y] (1513)

recPictureChOut[xCtb − 2, y] = recPictureCh[xCtb, y] (1514)

recPictureChOut[xCtb − 1, y] = recPictureCh[xCtb, y] (1515)

— Otherwise (availableL is equal to TRUE), for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb − 3, y] = recPictureCh[xCtb − 3, y] (1516)

recPictureChOut[xCtb − 2, y] = recPictureCh[xCtb − 2, y] (1517)

recPictureChOut[xCtb − 1, y] = recPictureCh[xCtb − 1, y] (1518)

— If availableR is equal to FALSE, for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb + blkWidth + 2, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1519)

recPictureChOut[xCtb + blkWidth + 1, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1520)

recPictureChOut[xCtb + blkWidth, y] =
recPictureCh[xCtb + blkWidth − 1, y] (1521)

— Otherwise (availableR is equal to TRUE), for y = yCtb + blkHeight..yCtb + blkHeight + 2, samples recPictureChOut are derived as follows:

recPictureChOut[xCtb + blkWidth + 2, y] =
recPictureCh[xCtb + blkWidth + 2, y] (1522)

recPictureChOut[xCtb + blkWidth + 1, y] =
recPictureCh[xCtb + blkWidth + 1, y] (1523)

recPictureChOut[xCtb + blkWidth, y] =
recPictureCh[xCtb + blkWidth, y] (1524)

D.2.1

Replace the text with the following

sei_payload(payloadType, payloadSize) {	**Descriptor**
if(payloadType = = 0)	
buffering_period(payloadSize)	
else if(payloadType = = 1)	
pic_timing(payloadSize)	
else if(payloadType = = 4)	
user_data_registered_itu_t_t35(payloadSize)	
else if(payloadType = = 5)	
user_data_unregistered(payloadSize)	
else if(payloadType = = 6)	
recovery_point(payloadSize)	
else if(payloadType = = 56)	