INTERNATIONAL STANDARD ISO/IEC 23003-1:2007
TECHNICAL CORRIGENDUM 1

Published 2008-02-01

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

Information technology — MPEG audio technologies —

Part 1:
MPEG Surround

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Technologies audio MPEG —

Partie 1: Ambiance MPEG

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 23003-1:2007 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 23003-1:2007/Cor.1:2008(E)**

Published in Switzerland

In the following, changes in existing text and tables are highlighted by gray background.

Throughout the whole document, replace:

down-mix

by:

downmix

Throughout the whole document, replace:

down mix

by:

downmix

Throughout the whole document, replace:

Uimsbf

by:

uimsbf

Throughout the whole document, replace:

Vlclbf

by:

vlclbf

Throughout the whole document, replace:

UiMsbf

by:

uimsbf

Throughout the whole document, replace:

BsMsbf

by:

bsmsbf

Throughout the whole document, including subclause headings, replace:

signalling

by:

signaling

and replace:

Signalling

by:

Signaling

In subclause 3.1, insert at the beginning the new subclause 3.1.1 given below and increment the numbering of the following subclauses 3.1.x:

3.1.1
ATD
Arbitrary Tree Data, corresponds to CLD for arbitrary tree elements.

*In subclause 3.1.19, insert a line break between "***3.1.19***" and "***processing band***".*

In subclause 4.3.1.4, replace the title:

Enhanced Matrixed Mode of MPEG Surround

by:

Enhanced Matrix Mode of MPEG Surround

In subclause 4.3.1.4, in the first paragraph, replace:

includes an enhanced matrixed mode that creates

by:

includes an enhanced matrix mode that creates

In subclause 4.3.1.6, in the first paragraph, replace:

system supports matrixed encoded downmixes. The MPEG Surround encoder can create a stereo downmix that is matrixed encoded, and can thus be decode by legacy matrixed surround decoders. The MPEG Surround decoder, will invert the matrixed encoded downmix, and produce the multi-channel signal based on the inverted downmix and the spatial parameters, without any degradation in quality due to the matrixed encoded downmix.

by:

system supports matrix encoded downmixes. The MPEG Surround encoder can create a stereo downmix that is matrix encoded, and can thus be decode by legacy matrix surround decoders. The MPEG Surround decoder, will invert the matrix encoded downmix, and produce the multi-channel signal based on the inverted downmix and the spatial parameters, without any degradation in quality due to the matrix encoded downmix.

In subclause 4.3.1.6, replace:

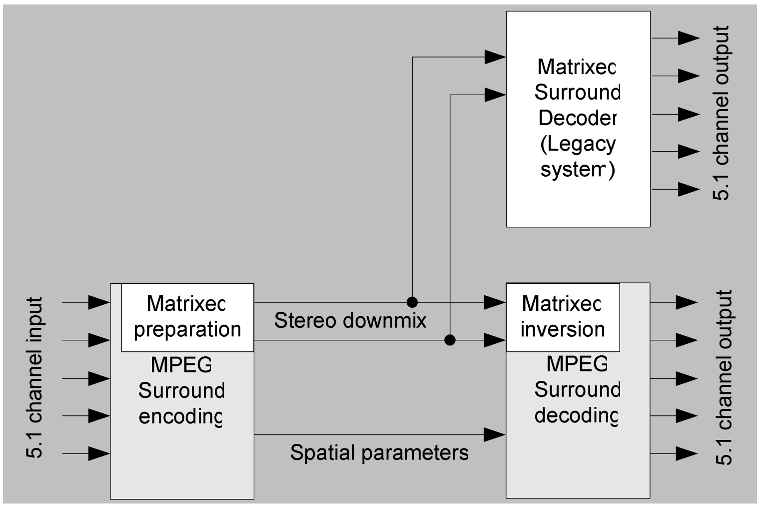


Figure 1 — Matrixed compatible MPEG Surround

by:

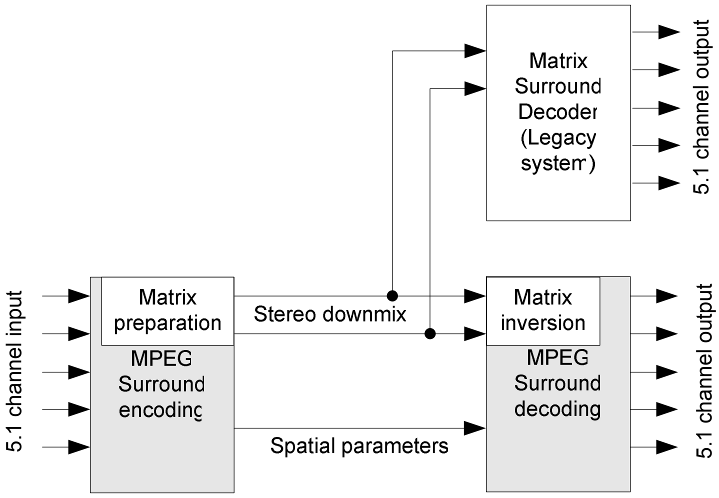


Figure 2 — Matrix compatible MPEG Surround

In subclause 4.3.1.7, in the first paragraph, replace:

the binaural encoded downmix is inverted similarly to the inversion of the matrixed compatible downmix.

by:

the binaural encoded downmix is inverted similarly to the inversion of the matrix compatible downmix.

In subclause 4.3.2, Table 2, replace:

Only supported over the complex valued part of the frequency range

by:

Only supported over a limited part of the frequency range

In subclause 4.3.2, replace the following row in Table 2:

Decorrelators		

by:

Decorrelators	Lattice decorrelators	PS decorrelators and low complexity lattice decorrelators

In subclause 4.4, replace:

The MPEG Surround decoder can be interfaced in either the time-domain or the QMF domain. In general, the MPEG Surround coder interfaces to the downmix channels by means of a 64 bands QMF domain frequency representation, identical to that standardized in ISO/IEC 14496-3 for High Quality Parametric Audio Coding. In the case the spatial coder is combined with HE-AAC, this QMF representation is directly available as an intermediate signal in the HE-AAC coder. In combination with alternative core coders, additional QMF analysis or synthesis modules, as defined in ISO/IEC 14496-3, are required for the spatial decoder and encoder, respectively. A Low Power MPEG Surround decoder cannot be connected in the QMF domain with a High Quality HE-AAC decoder, and vice versa a High Quality MPEG Surround decoder cannot be connected in the QMF domain with a Low Power HE-AAC decoder. For both combinations they shall be combined in the timedomain.

by:

The MPEG Surround decoder can be interfaced in either the time domain or the QMF domain. In general, the MPEG Surround coder interfaces to the downmix channels by means of a 64 band QMF domain representation, identical to that standardized in ISO/IEC 14496-3 subclause 4.6.18. In the case the spatial coder is combined with HE-AAC, this QMF representation is directly available as an intermediate signal in the HE-AAC coder. In combination with alternative downmix coders, additional QMF analysis or synthesis modules, as defined in subclause 6.3.2.1, are required for the spatial decoder and encoder, respectively.

It is not possible to connect HE-AAC and MPEG Surround in the QMF domain if the number of QMF bands is different. HE-AAC employs a 64 band QMF domain representation for the output signal, or a 32 band QMF domain representation if downsampled SBR is used. MPEG Surround employs a 32, 64, or 128 band QMF domain representation for the input signal, depending upon the sampling frequency used. In any combination where the number of QMF bands is not the same for HE-AAC and MPEG Surround they shall be connected in the time domain.

A Low Power MPEG Surround decoder cannot be connected in the QMF domain with a High Quality HE-AAC decoder, and vice versa a High Quality MPEG Surround decoder cannot be connected in the QMF domain with a Low Power HE-AAC decoder. Both combinations shall be connected in the time domain.

In subclause 4.5, remove the following sentence from the first paragraph:

The direct connection in the QMF domain is not possible in case of upsampled or downsampled operation of MPEG Surround, as defined in subclause 6.3.3.

In subclause 4.5, replace the following sentence in the fifth paragraph:

Note that special consideration is required if an MPEG Surround decoder and an HE-AAC decoder are connected in the time domain while a connection in the QMF domain would have been possible.

by:

Note that special consideration is required if an MPEG Surround decoder and an HE-AAC decoder are connected in the time domain while a connection in the QMF domain would have been possible according to subclause 4.4.

In subclause 4.6, replace the title:

Downmix gain

by:

Pre- and post-gains

In subclause 4.7.2, replace:

Five different hierarchical levels are defined, which allow for different numbers of input and output channels, for different ranges of sampling rates, and for a different bandwidth of the residual signal decoding. The level of the decoder must be equal to or larger than the level of the bitstream in order to ensure proper decoding. In addition, decoders of level 1 and 2 are capable of decoding all bitstreams of level 2 and 3, though at a possibly slightly reduced quality due to the limitations of the decoder. The quality and format of the output of an MPEG Surround decoder furthermore depends on the specific decoder configuration, as detailed below. However, these decoder configuration aspects are completely orthogonal to the different levels of this profile.

by:

Five different hierarchical levels are defined, which allow for different numbers of input and output channels, for different ranges of sampling rates, and for a different bandwidth of the residual signal decoding. The level of the decoder must be equal to or larger than the level of the bitstream in order to ensure proper decoding. In addition, decoders of level 1, 2 and 3 are capable of decoding all bitstreams of level 2, 3 and 4, though at a possibly slightly reduced quality due to the limitations of the decoder. The quality and format of the output of an MPEG Surround decoder furthermore depends on the specific decoder configuration, as detailed below. However, these decoder configuration aspects are completely orthogonal to the different levels of this profile.

In subclause 4.7.2, in Table 3, replace:

Residual coding based External Downmix compensation (subclause 6.5.2.3.4)

by:

Residual coding based External Downmix compensation (subclause 6.1.6)

In subclause 4.7.2, at the penultimate bullet in the third bullet list, replace:

Stereo output (specified in subclause 6.4.7.3).

by:

Stereo output (specified in subclause 6.4.7).

In subclause 4.7.2, at the last bullet in the third bullet list, replace:

Binaural output (specified in clause 7).

by:

Binaural output (specified in clause 6.11).

In subclause 4.7.2, replace:

Table 4 — Levels of the Baseline MPEG Surround profile

Level	**Tree configurations**	**Max. number output channels**	**Max. sampling rate [kHz]**	**Max. bandwidth residual coding [QMF bands]**	**Max. PCU High Quality decoder**	**Max. RCU High Quality decoder**	**Max. PCU Low Power decoder**	**Max. RCU Low Power decoder**
1	515, 525 (Note 1) (Note 4)	2.0	48	0 (Note 2)	12	5	6	4
2	515, 525 (Note 4)	5.1	48	0 (Note 2)	25	15	12	11
3	515, 525 (Note 4)	5.1	48	64 (Note 3)	25	15	12	11
4	515, 525, 757, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15
5	515, 525, 757, 727, plus arbitrary tree extension	32 incl. LFE	96	64 (Note 3)	123 (max. 70 at 48 kHz sampling)	61 (max. 38 at 48 kHz sampling)	80 (max. 44 at 48 kHz sampling)	53 (max. 32 at 48 kHz sampling)
Note 1: This level provides a 2-channel stereo output. Note 2: Residual coding data, if present in the bitstream, is not utilized, hence the residual decoding tool is not required. Note 3: A low power decoder utilizes only residual coding data for the first 8 QMF bands, corresponding to approximately 2.7 kHz bandwidth. Note 4: Arbitrary tree extension data, if present, is not utilized.								

by:

Table 4 — Levels of the Baseline MPEG Surround profile

Level	**Tree configurations**	**Max. number output channels**	**Max. sampling rate [kHz]**	**Max. bandwidth residual coding [QMF bands]**	**Max. PCU High Quality decoder**	**Max. RCU High Quality decoder**	**Max. PCU Low Power decoder**	**Max. RCU Low Power decoder**
1	515, 525, 727 (Note 1) (Note 4)	2.0	48	0 (Note 2)	12	5	6	4
2	515, 525, 727 (Note 4)	5.1	48	0 (Note 2)	25	15	12	11
3	515, 525, 727 (Note 4)	5.1	48	64 (Note 3)	25	15	12	11
4	515, 525, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15

5	515, 525, 757, 727 (Note 4)	7.1	48	64 (Note 3)	34	21	17	15
6	515, 525, 757, 727, plus arbitrary tree extension	32 incl. LFE	96	64 (Note 3)	123 (max. 70 at 48 kHz sampling)	61 (max. 38 at 48 kHz sampling)	80 (max. 44 at 48 kHz sampling)	53 (max. 32 at 48 kHz sampling)
Note 1: This level provides a 2-channel stereo output. Note 2: Residual coding data, if present in the bitstream, is not utilized, hence the residual decoding tool is not required. Note 3: A low power decoder utilizes only residual coding data for the first 8 QMF bands, corresponding to approximately 2.7 kHz bandwidth. Note 4: Arbitrary tree extension data, if present, is not utilized.								

In subclause 4.7.2, after the bullet "Support of stereo output for 515…" insert the following new bullet:

- Support of 5.1 output for 727 (subclause 6.4.8) is mandatory for levels 2 and 3.

In subclause 4.7.2 replace:

- Support for "Multi-slots HRTF convolution approach" (subclause 6.11.4.2.3) is mandatory for all levels of a high quality decoder (since this mode is not applicable in case of a low power decoder), and the complexity figures assume that a set of HRTF filters with a length of 128 samples (level 1) or 512 samples (levels 2 to 5) in the time domain is used.

by:

- Support for "Multi-slots HRTF convolution approach" (subclause 6.11.4.2.3) is mandatory for all levels of a high quality decoder (since this mode is not applicable in case of a low power decoder), and the complexity figures assume that a set of HRTF filters with a length of 128 samples (level 1) or 512 samples (levels 2 to 6) in the time domain is used.

In subclause 5.1, in all syntax tables, replace:

Else

by:

else

In subclause 5.1, replace Table 8:

```
Syntax                                                      No. of bits   Mnemonic
ParamHRTFset()
{
    bsHRTFfreqRes;                                          3             uimsbf

    bsHRTFasymmetric;                                       1             uimsbf
    for (hc=0; hc<HRTFnumChan; hc++) {                                    Note 1
        for (hb =0; hb <HRTFnumBands; hb ++) {                            Note 2
            bsHRTFlevelLeft[hc][hb];                        6             uimsbf
        }
        if (bsHRTFasymmetric) {
            for (hb =0; hb <HRTFnumBands; hb ++) {                        Note 2
                bsHRTFlevelRight[hc][hb];                   6             uimsbf
            }
        }
        bsHRTFphase[hc];                                    1             uimsbf
        if (bsHRTFphase[hc]) {
            for (hb =0; hb <HRTFnumPhase; hb ++) {                        Note 3
                bsHRTFphaseLR[hc][hb];                      6             uimsbf
            }
        }
        bsHRTFicc[hc];                                      1             uimsbf
        if (bsHRTFicc[hc]) {
            for (hb =0; hb <HRTFnumBands; hb ++) {                        Note 2
                bsHRTFiccLR[hc][hb];                        3             uimsbf
            }
        }
    }
}
Note 1: HRTFnumChan= 5.
Note 2: HRTFnumBands is defined in Table 53 and depends on bsHRTFfreqRes.
Note 3: HRTFnumPhase is defined in Table 53 and depends on bsHRTFfreqRes.
```

by:

Syntax	No. of bits	Mnemonic
ParamHRTFset()		
{		
bsHRTFfreqRes;	**3**	**uimsbf**
bsHRTFasymmetric;	**1**	**uimsbf**
for (hc=0; hc<HRTFnumChan; hc++) {		Note 1
for (hb =0; hb < HRTFnumBands; hb ++) {		Note 2
bsHRTFlevelLeft[hc][hb];	**6**	**uimsbf**
}		
if (bsHRTFasymmetric) {		
for (hb =0; hb <HRTFnumBands; hb ++) {		Note 2
bsHRTFlevelRight[hc][hb];	**6**	**uimsbf**
}		
}		
bsHRTFphase[hc];	**1**	**uimsbf**
if (bsHRTFphase[hc]) {		
for (hb =0; hb <HRTFnumPhase; hb ++) {		Note 3
bsHRTFphaseLR[hc][hb];	**6**	**uimsbf**
}		
}		
}		
}		
Note 1: HRTFnumChan = 5. Note 2: HRTFnumBands is defined in Table 53 and depends on bsHRTFfreqRes. Note 3: HRTFnumPhase is defined in Table 53 and depends on bsHRTFfreqRes.		

In subclause 5.1, in Table 20 and Table 21, replace:

numTempShapeChan is defined by Table 40 and

by:

numTempShapeChan is defined by Table 67 and

In subclause 5.1, Table 26, replace:

Pilot = 1Dhuff_dec(hcodPilot_XXX, **bsCodeW**);

by:

Pilot = 1Dhuff_dec(hcodFirstBand_XXX, **bsCodeW**);

In subclause 5.1, Table 26, replace:

Note 1: XXX is to be replaced by the value of dataType (CLD, ICC or CPC, ATD) .

by:

Note 1: XXX is to be replaced by the value of dataType (CLD, ICC, CPC, ATD) .

In subclause 5.1, replace Table 34:

Table 34 — Syntax of ResidualData()

```
Syntax                                                                        No. of bits  Mnemonic
ResidualData()
{
    for (i=0; i<numOttBoxes+numTttBoxes; i++) {
        if (bsResidualPresent[i]) {
            if (i<numOttBoxes) {
                for (ps=0; ps<numParamSets; ps++) {                                        Note 2
                    bsIccDiffPresent[i][ps];                                  1            Uimsbf
                    if (bsIccDiffPresent[i][ps]) {
                        for (pb=0; pb<bsResidualBands[i]; pb++) {
                         IccDiff[i][ps][pb] =                                 1..7         vlclbf
                                1Dhuff_dec(hcod1D_ICC_Diff,bsCodeW);                       Note 3
                        }
                    }
                }
            }
            tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1);                   Note 4
            for (rf=0; rf<bsResidualFramesPerSpatialFrame; rf++)
                individual_channel_stream(0);                                              Note 1
                if (window_sequence == EIGHT_SHORT_SEQUENCE) &&                            Note 5
                        ((tempExtraFrame == 18) || (tempExtraFrame == 24) ||
                        (tempExtraFrame == 30)) {
                    individual_channel_stream(0);                                          Note 1
                }
            }
        }
    }
}
```

Note1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.
Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.
Note 3: 1Dhuff_dec() is defined in Annex A.1.
Note 4: numSlots is defined by numSlots = bsFrameLength +1. Furthermore the division shall be interpreted as ANSI C integer division.
Note 5: individual_channel_stream(0) determines the value of window_sequence.

by:

Table 34 — Syntax of ResidualData()

```
Syntax                                                                    No. of bits   Mnemonic
ResidualData()
{
    for (k=0; k<numOttBoxes+numTttBoxes; k++) {
        if (bsTreeConfig==3 || bsTreeConfig==4) {
            i = resReorder[k];                                                          Note 6
        }
        else
            i = k;
        if (bsResidualPresent[i]) {
            if (i<numOttBoxes) {
                for (ps=0; ps<numParamSets; ps++) {                                     Note 2
                    bsIccDiffPresent[i][ps];                              1             Uimsbf
                    if (bsIccDiffPresent[i][ps]) {
                        for (pb=0; pb<bsResidualBands[i]; pb++) {
                          IccDiff[i][ps][pb] =                            1..7          vlclbf
                                1Dhuff_dec(hcod1D_ICC_Diff,bsCodeW);                    Note 3
                        }
                    }
                }
            }
            tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1);                Note 4
            for (rf=0; rf<bsResidualFramesPerSpatialFrame; rf++)
                individual_channel_stream(0);                                           Note 1
                if (window_sequence == EIGHT_SHORT_SEQUENCE) &&                         Note 5
                        ((tempExtraFrame == 18) || (tempExtraFrame == 24) ||
                        (tempExtraFrame == 30)) {
                    individual_channel_stream(0);                                       Note 1
                }
            }
        }
    }
}
```

Note1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.
Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.
Note 3: 1Dhuff_dec() is defined in Annex A.1.
Note 4: numSlots is defined by numSlots = bsFrameLength +1. Furthermore the division shall be interpreted as ANSI C integer division.
Note 5: individual_channel_stream(0) determines the value of window_sequence.
Note 6: resReorder[k] for 0 <= k < 6 is the following constant integer array: {0, 1, 2, 5, 3, 4}.

In subclause 5.1, in Table 36, Note 5, replace:

profile bitsream syntax

by:

profile bitstream syntax

In subclause 5.2, in Table 41, in the second row, replace:

default fine quantization for CLD, ICC, CPC

by:

default fine quantization for CLD, ICC, CPC, ATD

In subclause 5.2, replace:

bsTttBandsHigh	Same as bsTttBandsLow but for high band range. The high band range is bsTttBandsLow <= pb < **bsTttBandsHigh**.

by:

bsTttBandsHigh	Helper variable, same as bsTttBandsLow but for high band range. The high band range is bsTttBandsLow <= pb < bsTttBandsHigh.

In subclause 5.2, remove the definition of bsHRTFicc and bsHRTFiccLR.

In subclause 5.2, Table 54 in the heading, replace:

bsSacExtTyp

by:

bsSacExtType

In subclause 5.2, Table 71, replace:

XXX (dataType)	**numQuantStepsXXXCoarse**	**numQuantStepsXXXFine**
CLD	16	31
ICC	4	8
CPC	26	51

by:

XXX (dataType)	**numQuantStepsXXXCoarse**	**numQuantStepsXXXFine**
CLD, ATD	15	31
ICC	4	8
CPC	26	51

In subclause 5.2, Table 72, replace:

numQuantSteps	**maxGrpLen**
3	5
6	5
7	6
11	2
13	4
19	4
25	3
51	4
any other value	1

by:

numQuantSteps	maxGrpLen
3	5
7	6
11	2
13	4
19	4
25	3
51	4
any other value	1

In subclause 5.2, Table 76, replace:

LavIdx	**lavTabCLD [LavIdx]**	**lavTabICC [LavIdx]**	**lavTabCPC [LavIdx]**
0	3	1	3
1	5	3	6
2	7	5	9
3	9	7	12

by:

LavIdx	**lavTabCLD [LavIdx], lavTabATD [LavIdx]**	**lavTabICC [LavIdx]**	**lavTabCPC [LavIdx]**
0	3	1	3
1	5	3	6
2	7	5	9
3	9	7	12

In subclause 5.2, remove the definition of hcodPilot_XXX

In subclause 5.2, replace definition of hcodFirstBand_XXX:

One-dimensional Huffman code (Table A.2, Table A.3, and Table A.4) used for coding of data the data type of which is determined by the value of XXX. It is applied for coding of the lowest frequency band whenever differential coding in frequency direction is applied.

by:

One-dimensional Huffman code (Table A.2, Table A.3, and Table A.4)used for coding of data the data type of which is determined by the value of XXX. It is applied for coding of the lowest frequency band whenever pilot-based coding or differential coding in frequency direction is applied.

In subclause 5.2, replace:

bsXXXpcm[pi][ps][pb]	PCM coded indices where XXX is to be replaced by CLD, ICC, or CPC. The indices are offset so that they cannot be negative.
bsXXXmsbDiff[pi][ps][pb]	Differentially coded most significant bits of a quantization index of data type XXX (where XXX can be either CLD or ICC or CPC) belonging to parameter index pi, parameter set ps and parameter band pb.

bsXXXlsb[pi][ps][pb]	Least significant bit of a quantization index of data type XXX [...]. May only be 1 in case of data type CLD and fine quantization; otherwise always 0.
idxXXX[pi][ps][pb]	Quantized spatial parameter (as index, can be negative) for the pi:th XXX parameter for the ps:th parameter set (0 <= ps < numParamSets) and the pb:th parameter band (0 <= pb < numBands). XXX is to be replaced by CLD, ICC, or CPC.

by:

bsXXXpcm[pi][ps][pb]	PCM coded indices where XXX is to be replaced by CLD, ICC, CPC, or ATD. The indices are offset so that they cannot be negative.
bsXXXmsbDiff[pi][ps][pb]	Differentially coded most significant bits of a quantization index of data type XXX (where XXX can be CLD. ICC, CPC, or ATD) belonging to parameter index pi, parameter set ps and parameter band pb.
bsXXXlsb[pi][ps][pb]	Least significant bit of a quantization index of data type XXX (where XXX can be CLD. ICC, CPC, or ATD). May only be 1 in case of data type CPC and fine quantization; otherwise always 0.
idxXXX[pi][ps][pb]	Quantized spatial parameter (as index, can be negative) for the pith XXX parameter for the ps:th parameter set (0 <= ps < numParamSets) and the pbth parameter band (0 <= pb < numBands). XXX is to be replaced by CLD, ICC, CPC, or ATD.

In subclause 5.2, directly after the description of 'individual_channel_stream()' add a description for 'channel_pair_element()', including a new Table 79A:

MPEG-2 AAC Low Complexity profile channel_pair_element() according to the syntax defined in Table 14 (and related tables) of subclause 6.3 of ISO/IEC 13818-7. Decoding of a channel_pair_element() also determines the value of **window_sequence**, according to ISO/IEC 13818-7.
A second channel_pair_element() is present when **window_sequence** (determined by the first channel_pair_element()) equals EIGHT_SHORT_SEQUENCE and **tempExtraFrame** equals 18, 24 or 30. In this case, the value of **window_sequence** determined by the second channel_pair_element() shall equal EIGHT_SHORT_SEQUENCE.
Restrictions apply to the elements of the channel_pair_element() syntax. The restriction applied to the elements of the channel_pair_element() syntax is given in Table 79A.

Table 79A – Restrictions in syntax of channel_pair_element()

Tool	**Allowed value**
common_window	1

Furthermore, the restrictions applied to the elements ics_info(), defined in Table 15 of subclause 6.3 of ISO/IEC 13818-7, and individual_channel_stream(), defined in Table 16 of subclause 6.3 of ISO/IEC 13818-7, as sub-elements of the channel_pair-element(), are given in Table 77 and Table 78. In addition to the existing restriction on the maximum number of scalefactor bands to which Temporal Noise Shaping is applied, (the constant TNS_MAX_BANDS defined in Table 33 of subclause 7.1.6 of ISO/IEC 13818-7), the lowest scalefactor band where Temporal Noise Shaping is applied is restricted and depends on the sampling rate (derived from **bsResidualSamplingFrequencyIndex**) and whether **window_sequence** indicates a long or short window, as specified in Table 79.

In subclause 5.2, Table 81, replace:

numInChan					
1		2		6	
numAacEl	AacEl	numAacEl	AacEl	numAacEl	AacEl
1	'0'	1	'1'	2	'111'

by:

numInChan					
1		2		6	
numAacEl	AacEl	numAacEl	AacEl	numAacEl	AacEl
1	'0'	1	'1'	3	'111'

In subclause 6.1, replace the heading:

6.1 Compressed data stream decoding and de-quantization

by:

6.1 Bitstream decoding and de-quantization

In subclause 6.1.2, replace the heading:

6.1.2 Decoding of CLD ATD, ICC, CPC, and arbitrary downmix gain parameters

by:

6.1.2 Decoding of CLD, ATD, ICC, CPC, and arbitrary downmix gain parameters

In subclause 6.1.2.1, replace:

pi = parameter instance having values in the range which, for CLD, ICC and CPC have the range 0 .. numOttBoxes+4*numTttBoxes+numInChan-1, and for ATD have the range 0 .. numOttBoxesAT-1

by:

pi = parameter instance having values in the following ranges:

- CLD: 0..numOttBoxes+4*numTttBoxes+numInChan-1
- ICC: 0..numOttBoxes+2*numTttBoxes-1
- CPC: 0..4*numTttBoxes-1
- ATD: 0..numOttBoxesAT-1

In subclause 6.1.2.1, last paragraph, replace:

The following process described in the subclauses below is carried out for all instances pi of all parameter types CLD, ATD, ICC, CPC, where XXX denotes the actual parameter type.

by:

The following process described in the subclauses below is carried out for all instances pi of all parameter types CLD, ATD, ICC, CPC, where XXX denotes the actual parameter type. Note that, if the bsOneIcc flag is set to one, one coherence information (ICC) of the original audio signal applicable to all OTT boxes is contained within the parametric representation of the decoder tree; in particular, for 5-1-5 configurations only one coherence information of the original audio signal is transmitted within the bitstream.

In subclause 6.1.2.2, at the end of the subclause, add:

In case bsOneIcc is set to one, copy single ICC to all OTT parameter instances.

```
if (bsOneICC) {
  firstPiNonLfe = -1;
  for (pi=0; pi<numOttBoxes; pi++) {
    if (!ottModeLfe[pi]) {
      firstPiNonLfe = pi;
      break;
    }
  }
  if (firstPiNonLfe != -1) {
    for (pi=0; pi<numOttBoxes; pi++) {
      if (!ottModeLfe[pi]) {
        for (ps=0; ps<numParamSets; ps++) {
          for (pb=0; pb<numBands; pb++) {
            idxICC[pi][ps][pb] = idxICC[firstPiNonLfe][ps][pb];
          }
        }
      }
    }
  }
}
```

In subclause 6.1.2.3.2, replace:

```
if (bsPilotCoding[pi][setIdx]) {
```

by:

```
if (bsPilotCodingXXX[pi][setIdx]) {
```

In subclause 6.1.8, in the text following table 84, replace:

In case of a 5-2-5 configuration the EdQ tool is not used.

by:

In case of a 5-2-5, 7-2-7, or 7-5-7 configuration the EdQ tool is not used.

In subclause 6.1.11.1, replace:

In order to account for coarse quantization and low up-date rate of spatial parameters (CLD, ICC and CPC), smoothing can be applied. The MPEG Surround decoder performs the smoothing on the matrices resulting from the received parameters rather than directly on the parameters. The resulting effect is the same. The smoothing is performed on the matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ by first order IIR filtering of the parameter bands, for which smoothing is active. The actual smoothing process as well as those matrices are defined in subclauses 6.5.2 and 6.5.3 for matrix $\mathbf{W}_1$ and $\mathbf{W}_2$, respectively.

by:

In order to account for coarse quantization and low up-date rate of spatial parameters (CLD, ICC, CPC, and ATD), smoothing can be applied. The MPEG Surround decoder performs the smoothing on the matrices resulting from the received parameters rather than directly on the parameters. The resulting effect is the same. The smoothing is performed on the matrices $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ by first order IIR filtering of the parameter bands, for which smoothing is active. The actual smoothing process as well as those matrices are defined in subclauses 6.5.2, 6.5.3, and 6.5.4 for matrix $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$, respectively.

In subclause 6.1.11.2, replace:

$$\mathbf{S}(l,m) = \mathbf{smgData}[l][m],$$

by:

$$\mathbf{S}(l,m) = \begin{cases} 0 & ,m < \max(\mathbf{m}_{\mathrm{resPar}}) \\ \mathbf{smgData}[l][m] & ,m \geq \max(\mathbf{m}_{\mathrm{resPar}}) \end{cases},$$

In subclause 6.1.11.3, replace:

For encoder controlled smoothing one of four time constants can be signaled in the bitstream, while the automatic mode uses a fixed value of 256 time slots:

by:

For encoder controlled smoothing one of four time constants can be signaled in the bitstream:

In subclause 6.1.13, Table 87, replace:

numSlots	*bsResidualFramesPerSpatialFrame* **or** *bsArbitraryDownmixResidualFramesPerSpatialFrame*
15	0
16	0
18	0
24	0
30	0
32	0, 1
36	1
48	1
60	1
64	1, 3
72	2

by:

numSlots	*bsResidualFramesPerSpatialFrame* **or** *bsArbitraryDownmixResidualFramesPerSpatialFrame*
15	0
16	0
18	0
24	0
30	0
32	0, 1
36	1
48	1
60	1
64	1, 3
72	3

In subclause 6.2.1, second paragraph, replace:

and an Interchannel Cross Correlation $\left(CLD_{\mathrm{dm}}\right)$

by:

and an Interchannel Cross Correlation $\left(ICC_{\mathrm{dm}}\right)$

In subclause 6.2.2, second paragraph, replace:

stereo hybrid QMF-domain input signals $\left(\boldsymbol{x}_{\boldsymbol{X}}^{\boldsymbol{n},\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{n},\boldsymbol{k}}\right)$ following

by:

stereo hybrid QMF-domain input signals $\left(\boldsymbol{x}_{\boldsymbol{L}_0}^{\boldsymbol{n},\boldsymbol{k}}, \boldsymbol{x}_{\boldsymbol{R}_0}^{\boldsymbol{n},\boldsymbol{k}}\right)$ following

In subclause 6.2.2, second paragraph, remove the following text:

Furthermore, a slot-based energy Q_X^n for each slot *n* is computed from both stereo hybrid QMF-domain input signals ($x_X^{n,k}$ $x_i^{n,k}$) following

$$Q_X^n = \sum_{k=16}^{42} x_X^{n,k} \left(x_X^{n,k}\right)^* \ , \ X \in \left\{\mathrm{L}_0, \mathrm{R}_0\right\}$$

From Q_X^n, two analysis state parameters $r_{X,1}^n$ and $r_{X,2}^n$ for $X \in \left\{\mathrm{L}_0, \mathrm{R}_0\right\}$ are subsequently updated according to:

$$r_{X,1}^n = c_r r_{X,1}^{n-1} + \left(1 - c_r\right)\left(Q_X^n\right)^2,$$

$$r_{X,2}^n = c_r r_{X,2}^{n-1} + \left(1 - c_r\right)\left(Q_X^n - Q_X^{n-1}\right)^2,$$

with

$$c_r = \exp\left\{\frac{-64}{0.3F_s}\right\}.$$

In subclause 6.2.4, second paragraph, replace:

are given in Table A.31.

by:

are given in Table A.33.

In subclause 6.2.4, third paragraph, replace:

according to Table A.31.

by:

according to Table A.33.

In subclause 6.2.4, replace:

$$\begin{aligned}\mathbf{CPC}(m) = \; & \left(1-w_1(m)\right)\left(1-w_2(m)\right)\mathbf{CLD}\left[T_{CPC}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor,\left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right)\right]+\\ & w_1(m)\left(1-w_2(m)\right)\mathbf{CLD}\left[T_{CPC}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil,\left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right)\right]+\\ & \left(1-w_1(m)\right)w_2(m)\mathbf{CLD}\left[T_{CPC}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor,\left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right)\right]+\\ & w_1(m)w_2(m)\mathbf{CLD}\left[T_{CPC}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil,\left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right)\right]\end{aligned}$$

by:

$$\begin{aligned}\alpha^{l,m} &= \mathbf{CPC}\left[\left\lfloor T_{CPC1}^{0,0}+T_{CPC1}^{0,1}+T_{CPC1}^{1,0}+T_{CPC1}^{1,1}+0.5\right\rfloor\right],\ \mathit{with}\\ T_{CPC1}^{0,0} &= \left(1-w_1(m)\right)\cdot\left(1-w_2(m)\right)\cdot T_{CPC1}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor,\left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right),\\ T_{CPC1}^{0,1} &= w_1(m)\cdot\left(1-w_2(m)\right)\cdot T_{CPC1}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil,\left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right),\\ T_{CPC1}^{1,0} &= \left(1-w_1(m)\right)\cdot w_2(m)\cdot T_{CPC1}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor,\left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right),\\ T_{CPC1}^{1,1} &= w_1(m)\cdot w_2(m)\cdot T_{CPC1}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil,\left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right).\end{aligned}$$

,

$$\begin{aligned}
\beta^{l,m} &= \mathbf{CPC}\left\lfloor\!\left\lfloor T_{CPC2}^{0,0} + T_{CPC2}^{0,1} + T_{CPC2}^{1,0} + T_{CPC2}^{1,1} + 0.5 \right\rfloor\!\right\rfloor, \mathit{with} \\
T_{CPC2}^{0,0} &= \left(1 - w_1(m)\right)\cdot\left(1 - w_2(m)\right)\cdot T_{CPC2}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right), \\
T_{CPC2}^{0,1} &= w_1(m)\cdot\left(1 - w_2(m)\right)\cdot T_{CPC2}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right), \qquad , \\
T_{CPC2}^{1,0} &= \left(1 - w_1(m)\right)\cdot w_2(m)\cdot T_{CPC2}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right), \\
T_{CPC2}^{1,1} &= w_1(m)\cdot w_2(m)\cdot T_{CPC2}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right).
\end{aligned}$$

$$\begin{aligned}
CLD_X^{l,m} &= \mathbf{CLD}\left\lfloor\!\left\lfloor T_{CLD}^{0,0} + T_{CLD}^{0,1} + T_{CLD}^{1,0} + T_{CLD}^{1,1} + 0.5 \right\rfloor\!\right\rfloor \\
T_{CLD}^{0,0} &= \left(1 - w_1(m)\right)\cdot\left(1 - w_2(m)\right)\cdot T_{CLD}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right) \\
T_{CLD}^{0,1} &= w_1(m)\cdot\left(1 - w_2(m)\right)\cdot T_{CLD}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right) \qquad , \\
T_{CLD}^{1,0} &= \left(1 - w_1(m)\right)\cdot w_2(m)\cdot T_{CLD}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right) \\
T_{CLD}^{1,1} &= w_1(m)\cdot w_2(m)\cdot T_{CLD}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right)
\end{aligned}$$

$$\begin{aligned}
ICC_X^{l,m} &= \mathbf{ICC}\left\lfloor\!\left\lfloor T_{ICC}^{0,0} + T_{ICC}^{0,1} + T_{ICC}^{1,0} + T_{ICC}^{1,1} + 0.5 \right\rfloor\!\right\rfloor \\
T_{ICC}^{0,0} &= \left(1 - w_1(m)\right)\cdot\left(1 - w_2(m)\right)\cdot T_{ICC}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right) \\
T_{ICC}^{0,1} &= w_1(m)\cdot\left(1 - w_2(m)\right)\cdot T_{ICC}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m)\right\rfloor\right) \qquad , \\
T_{ICC}^{1,0} &= \left(1 - w_1(m)\right)\cdot w_2(m)\cdot T_{ICC}\left(\left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m)\right\rfloor, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right) \\
T_{ICC}^{1,1} &= w_1(m)\cdot w_2(m)\cdot T_{ICC}\left(\left\lceil \mathbf{dm}_{\mathrm{CLI}}(m)\right\rceil, \left\lceil \mathbf{dm}_{\mathrm{ICI}}(m)\right\rceil\right)
\end{aligned}$$

In subclause 6.2.4, replace:

where $\lfloor . \rfloor$ denotes the 'floor' function, $\lceil . \rceil$ the 'ceil' function, and w_1 and w_2 the interpolation weights:

$$w_1 = \mathbf{dm}_{\mathrm{CLI}}(m) - \left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m) \right\rfloor,$$

$$w_2 = \mathbf{dm}_{\mathrm{ICI}}(m) - \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m) \right\rfloor.$$

by:

where $\lfloor . \rfloor$ denotes the 'floor' function, $\lceil . \rceil$ the 'ceil' function, $\mathbf{CLD}[.]$ is given by Table 82, $\mathbf{ICC}[.]$ is given by Table 83, $\mathbf{CPC}[.]$ is given by Table 84, $T_{CPC1}(.)$, $T_{CPC2}(.)$, $T_{CLD}(.)$ and $T_{ICC}(.)$ are given by Table A.33, $X = \{1,2\}$, and w_1 and w_2 the interpolation weights:

$$w_1 = \mathbf{dm}_{\mathrm{CLI}}(m) - \left\lfloor \mathbf{dm}_{\mathrm{CLI}}(m) \right\rfloor,$$

$$w_2 = \mathbf{dm}_{\mathrm{ICI}}(m) - \left\lfloor \mathbf{dm}_{\mathrm{ICI}}(m) \right\rfloor.$$

In subclause 6.2.4, replace:

This process is performed for two CPC parameters, one CLD and one ICC parameter, according to Table. A.31. These parameters are used to generate the spatial synthesis matrices M_1 and M_2, under the following constraints:

1. Generation of M_1 and M_2 is based on the '525' mode, using matrix compatibility (bsMatrixMode=1, see subclause 6.5.2.4), and a prediction based up-mix (bsTttModeLow =1, see subclause 6.5), and where $\gamma^{l,m}$ is assumed to be $\gamma^{l,m} = 1$;

2. The same CLD and ICC value is used for CLD_1 / CLD_2 and ICC_1 / ICC_2 , for both OTT boxes respectively;

3. The CPC parameters for processing band *m* have to be interchanged if $CLD_{dm}(m) < 0$.

by:

The resulting parameters are used to generate the spatial synthesis matrices M_1 and M_2, under the following constraints:

1. Generation of M_1 and M_2 is based on the '525' mode, using matrix compatibility (bsMatrixMode=1, see subclause 6.5.2.4), and a prediction based up-mix (bsTttModeLow =1, see subclause 6.5), and where $\gamma^{l,m}$ is assumed to be $\gamma^{l,m} = 1$, and surround channel gains of +1 (bsFixedGainsSur=0; see subclause 5.1);

2. The CPC parameters for processing band *m* have to be interchanged if $\mathbf{dm}_{\mathrm{CLD}}(m) < 0$;

3. For $X = 0$, $CLD_{X,nq}^{l,m} = \mathbf{CLD}[15]$ and $ICC_{X,nq}^{l,m} = \mathbf{ICC}[0]$ for all m and l.

In subclause 6.2.5, remove the sentence:

The temporal processing flags for the whole frame are based on the analysis states of slot '-1'.

In subclause 6.3.3, in the penultimate paragraph, replace:

Decorrelation (see subclause 5.5.2) for the upper 64 QMF bands is done by means of a 1 QMF sample delay...

by:

Decorrelation (see subclause 6.6) for the upper 64 QMF bands is done by means of a 1 QMF sample delay...

In subclause 6.3.3, remove the last paragraph:

For downsampled and upsampled operation of MPEG Surround in combination with an HE-AAC coded downmix signal, it is necessary to connect the HE-AAC decoder to the MPEG Surround decoder in the time domain.

In subclause 6.4.1, in the 2nd paragraph replace:

Every decorrelator can, for certain frequency regions as indicated by the compressed data stream, be replace by a residual signal.

by:

Every decorrelator can, for certain frequency regions as indicated by the bitstream, be replaced by a residual signal.

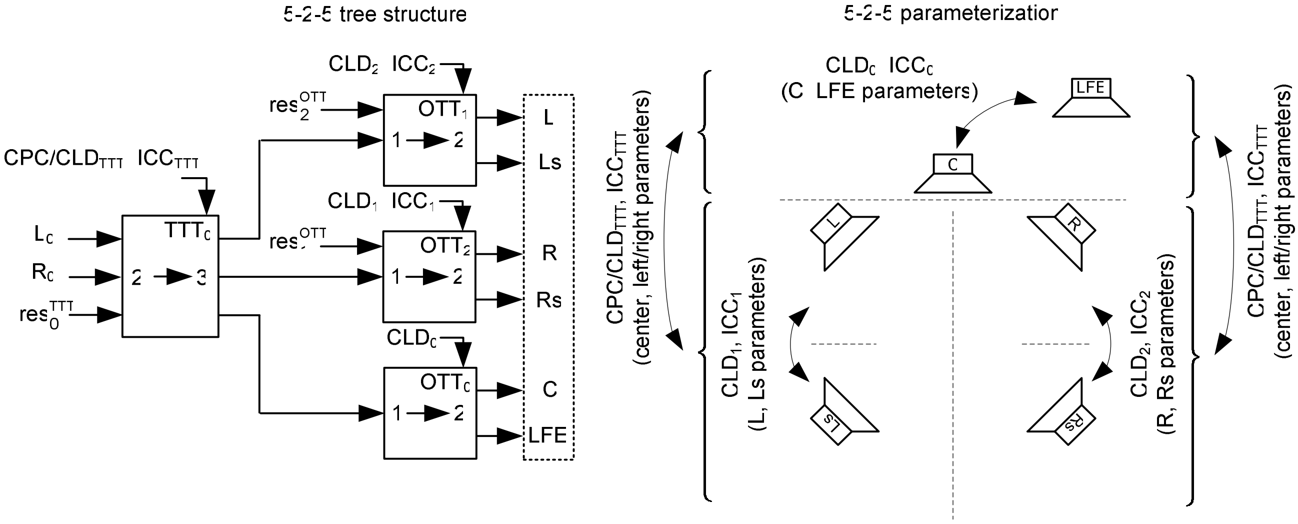
In subclause 6.4.1, in the 2nd paragraph replace:

by means of signalling in the data stream by activated.

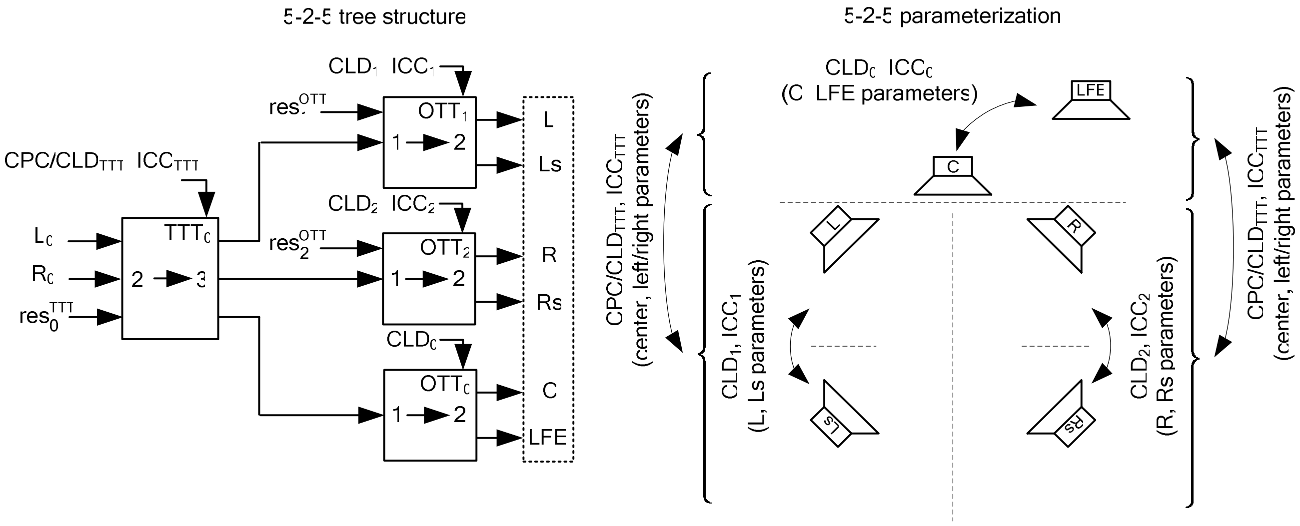
by:

by means of signaling in the data stream be activated.

In subclause 6.4.3.1, replace Figure 26:



by:


5-2-5 tree structure

5-2-5 parameterization

In subclause 6.4.4.2.1, replace:

The elements of $\mathbf{M}_2^{n,k}$ are defined in subclause 6.4.4…

by:

The elements of $\mathbf{M}_2^{n,k}$ are defined in subclause 6.5.3…

In subclause 6.4.5.2.1, replace:

The elements of $\mathbf{M}_2^{n,k}$ are defined in subclause 6.4.5…

by:

The elements of $\mathbf{M}_2^{n,k}$ are defined in subclause 6.5.3…

In subclause 6.4.7.3, below the first formula, replace:

with the exception of $\boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{n},\boldsymbol{k}}$.

by:

with the exception of $\boldsymbol{v}_{\boldsymbol{M}_0}^{\boldsymbol{n},\boldsymbol{k}}$.

In subclause 6.4.7.3, in the last sentence, replace:

given the $CLD_{\mathrm{q}}^{l,m}$ and $ICC_{\mathrm{q}}^{l,m}$ parameters calculated in the previous subclauses

by:

given the $CLD_{\mathrm{q}}^{l,m}$, $ICC_{\mathrm{q}}^{l,m}$ and $\boldsymbol{g}^{\boldsymbol{l},\boldsymbol{m}}$ parameters calculated in the previous subclauses.

After subclause 6.4.7, append the following new subclause 6.4.8:

6.4.8 5.1 output from 7-2-7 configurations

6.4.8.1 Introduction

The following subclauses describe the decoding process that obtains a 5.1 channel output from a 7.1 channel MPEG Surround bitstream using the 7-2-7 tree configuration. It is based on a mapping of the side information, followed by the normal decoding process for the 5-2-5 tree configuration.

6.4.8.2 Mapping of 7-2-7 tree parameters to 5-2-5 tree structure

For the decoding of a bitstream having 7-2-7 content on a decoder that is limited to 5.1 channel playback, the underlying 7-2-7_1 or 7-2-7_2 decoder tree structure is pruned to be congruent to the 5-2-5 structure. In this specific case, the spatial parameters (including possibly present residual signals) attributed to the OTT boxes OTT_3 and OTT_4 and the parametric data specific to the two additional upmix channels are not used in the actual upmix process (though they are parsed by the bitstream decoder). The upmix process is therefore fully described by the 5-2-5 upmix specification. Additionally, a parameter preprocessing is performed for the STP and the GES control data, if present, that maps it to the 5.1 output channels of the 5-2-5 tree structure.

6.4.8.3 Mapping of temporal shaping tool side information

6.4.8.3.1 General mapping

The temporal shaping flags **bsTempShapeEnableChannel** are mapped to form new flags suitable for the 5-2-5 structure. Specifically, the flags attributed to the output channels of the OTT boxes OTT_3 and OTT_4 are combined in a pair-wise fashion. For all other output channels the flags are not affected and used unaltered. The mapping is described in Table 88A.

Table 88A — Mapping of bsTempShapeEnableChannel flags

bsTreeConfig	Mapped 5-2-5 **bsTempShapeEnableChannel(ch)**	Combination of original 7-2-7 **bsTempShapeEnableChannel(ch')**
3 (7-2-7_1)	L	L' AND Lc'
	R	R' AND Rc'
4 (7-2-7_2)	Ls	Ls' AND Lsr'
	Rs	Rs' AND Rsr'
Note 1: Data of channels not listed in this table is copied unaltered. Note 2: “AND” denotes the logical AND operation.		

6.4.8.3.2 The STP tool

The STP processing is done as defined for the 5-2-5 configuration using the mapped parameters as described in 6.4.8.3.1.

6.4.8.3.2.1 The GES tool

The GES processing is done as defined for the 5-2-5 configuration using the mapped parameters as described in 6.4.8.3.1. Additionally, the transmitted envelope reshaping data is combined using the mapping described in the following formula and Table 88B. For all other output channels, the envelope reshaping data not affected and used unaltered.

$$XXX\left[ch\right]\left[n\right] = \mathbf{envReshapeData}\left[ch_1\right]\left[n\right] + \mathbf{envReshapeData}\left[ch_2\right]\left[n\right]$$

$$\mathbf{envReshapeData}\left[ch\right]\left[n\right] = INT\left(\frac{XXX\left[ch\right]\left[n\right]}{2.0} + 0.5 \cdot SIGN\left(XXX\left[ch\right]\left[n\right]\right)\right)$$

where SIGN() denoted the signum function.

Table 88B — Mapping of GES envReshapeData

bsTreeConfig	ch	ch_1;ch_2
3 (7-2-7_1)	L	L';Lc'
	R	R';Rc'
4 (7-2-7_2)	Ls	Ls';Lsr'
	Rs	Rs';Rsr'
Note 1: Data of channels not listed in this table is copied unaltered.		

In subclause 6.5.2.2.3.2, replace in last equation:

β^{lm}

by:

$\beta^{l,m}$

In subclause 6.5.2.2.3.3, replace:

$$\mathbf{bsTttModeLow}\left(0\right) = 3$$

by:

$$\mathbf{bsTttModeLow}\left(0\right) \in \left\{2,3\right\}$$

In subclause 6.5.2.2.3.3, replace:

$$\mathbf{bsTttModeLow}\left(0\right) = 5$$

by:

$$\mathbf{bsTttModeLow}\left(0\right) \in \left\{4,5\right\}$$

In subclause 6.5.2.2.3.3 replace:

For $\mathbf{bsTttModeLow}\left(0\right) \geq 3$

by:

For $\mathbf{bsTttModeLow}\left(0\right) \geq 2$

In subclause 6.5.2.3.1, first paragraph, replace:

parameterized external downmix compensation (bsArbitraryDownmix = 1), and residual coding based external downmix compensation (bsArbitraryDownmix = 2).

A decoder that does not support residual based external downmix compensation shall operate as if $bsArbitraryDownmix = 1$.

by:

parameterized external downmix compensation (bsArbitraryDownmix = 1), and residual coding based external downmix compensation (bsArbitraryDownmix = 1 plus a spatial extension of type *bsSacExtType* = 1 present).

A decoder that does not support residual based external downmix compensation shall perform only parametrized external downmix compensation when arbitrary downmix residual extension frames (*bsSacExtType* = 1) are present.

In subclause 6.5.2.3.4, in the title, replace:

downmix compensation *bsArbitraryDownmix* = 2

by:

downmix compensation *bsArbitraryDownmix* = 1 and spatial extension of type *bsSacExtType* = 1 present

In subclause 6.5.3.2, at the end of the paragraph, add:

Note, that in case the OTT box is operated in LFE mode, $ICC_X^{l,m}$ is set to 1.

In subclause 6.5.3.6, replace:

and for the 7-2-7$_2$ configuration the $\mathbf{R}_2^{l,m}$ matrix is defined according to:

$$\mathbf{R}_2^{l,m} = \begin{bmatrix}
\kappa_{TTT_0}^{l,m} H11_{OTT_1}^{l,m} & 0 & 0 & H12_{OTT_1}^{l,m} & 0 & g_{TTT_0}^{l,m} c_{1,OTT_1}^{l,m} & 0 & 0 \\
\kappa_{TTT_0}^{l,m} H11_{OTT_1}^{l,m} H11_{OTT_3}^{l,m} & 0 & 0 & H12_{OTT_1}^{l,m} H11_{OTT_3}^{l,m} & 0 & g_{TTT_0}^{l,m} c_{1,OTT_1}^{l,m} H11_{OTT_3}^{l,m} & H12_{OTT_3}^{l,m} & 0 \\
\kappa_{TTT_0}^{l,m} H11_{OTT_1}^{l,m} H21_{OTT_3}^{l,m} & 0 & 0 & H12_{OTT_1}^{l,m} H21_{OTT_3}^{l,m} & 0 & g_{TTT_0}^{l,m} c_{1,OTT_1}^{l,m} H21_{OTT_3}^{l,m} & H22_{OTT_3}^{l,m} & 0 \\
0 & \kappa_{TTT_0}^{l,m} H11_{OTT_2}^{l,m} & 0 & 0 & H12_{OTT_2}^{l,m} & g_{TTT_0}^{l,m} c_{1,OTT_2}^{l,m} & 0 & 0 \\
0 & \kappa_{TTT_0}^{l,m} H11_{OTT_2}^{l,m} H11_{OTT_4}^{l,m} & 0 & 0 & H12_{OTT_2}^{l,m} H11_{OTT_4}^{l,m} & g_{TTT_0}^{l,m} c_{1,OTT_2}^{l,m} H11_{OTT_4}^{l,m} & 0 & H12_{OTT_4}^{l,m} \\
0 & \kappa_{TTT_0}^{l,m} H11_{OTT_2}^{l,m} H21_{OTT_4}^{l,m} & 0 & 0 & H12_{OTT_2}^{l,m} H21_{OTT_4}^{l,m} & g_{TTT_0}^{l,m} c_{1,OTT_2}^{l,m} H21_{OTT_4}^{l,m} & 0 & H22_{OTT_4}^{l,m} \\
0 & 0 & \kappa_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 & -\sqrt{2} g_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 \\
0 & 0 & c_{2,OTT_0}^{l,m} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

by:

and for the 7-2-7_2 configuration the $\mathbf{R}_2^{l,m}$ matrix is defined according to:

$$\mathbf{R}_2^{l,m} = \begin{bmatrix} \kappa_{TTT_0}^{l,m} H11_{OTT_1}^{l,m} & 0 & 0 & H12_{OTT_1}^{l,m} & 0 & g_{TTT_0}^{l,m} c_{1,OTT_1}^{l,m} & 0 & 0 \\ H21_{OTT_1}^{l,m} H11_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_1}^{l,m} H11_{OTT_3}^{l,m} & 0 & 0 & H12_{OTT_3}^{l,m} & 0 \\ H21_{OTT_1}^{l,m} H21_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_1}^{l,m} H21_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_3}^{l,m} & 0 \\ 0 & \kappa_{TTT_0}^{l,m} H11_{OTT_2}^{l,m} & 0 & 0 & H12_{OTT_2}^{l,m} & g_{TTT_0}^{l,m} c_{1,OTT_2}^{l,m} & 0 & 0 \\ 0 & H21_{OTT_2}^{l,m} H11_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_2}^{l,m} H11_{OTT_4}^{l,m} & 0 & 0 & H12_{OTT_4}^{l,m} \\ 0 & H21_{OTT_2}^{l,m} H21_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_2}^{l,m} H21_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_4}^{l,m} \\ 0 & 0 & \kappa_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 & -\sqrt{2} g_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 \\ 0 & 0 & c_{2,OTT_0}^{l,m} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In subclause 6.6.1, replace Table 89, below:

Table 89 — Decorrelator-index as a function of decoder configuration

	Decorrelator $X = 0,...,9$									
configuration	0	1	2	3	4	5	6	7	8	9
5-1-5_1	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_2^{\mathrm{OTT}}(\ )$						
5-1-5_2	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$						
5-2-5	$D_2^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$							
7-2-7_1	$D_2^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$					
7-2-7_2	$D_2^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$					
7-5-7_1	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$								
7-5-7_2	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$								

by:

Table 89 — Decorrelator-index as a function of decoder configuration

	Decorrelator $X = 0,...,9$									
configuration	0	1	2	3	4	5	6	7	8	9
5-1-5$_1$	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_2^{\mathrm{OTT}}(\ )$						
5-1-5$_2$	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$						
5-2-5	$D_1^{\mathrm{OTT}}(\ )$	$D_2^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$							
7-2-7$_1$	$D_1^{\mathrm{OTT}}(\ )$	$D_2^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$					
7-2-7$_2$	$D_1^{\mathrm{OTT}}(\ )$	$D_2^{\mathrm{OTT}}(\ )$	$D_0^{\mathrm{TTT}}(\ )$	$D_3^{\mathrm{OTT}}(\ )$	$D_4^{\mathrm{OTT}}(\ )$					
7-5-7$_1$	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$								
7-5-7$_2$	$D_0^{\mathrm{OTT}}(\ )$	$D_1^{\mathrm{OTT}}(\ )$								

In subclause 6.6.2.1, in the first paragraph, replace:

In principle the frequency axis...

by:

The frequency axis...

In subclause 6.7.2, replace:

$$\tilde{\mathbf{z}}_{\mathrm{direct}}^{n,sb} = \begin{cases} \mathbf{z}_{\mathrm{direct}}^{n,sb} + \mathbf{z}_{\mathrm{diffuse}}^{n,sb} & ,0 \le sb < 5 \\ \mathbf{z}_{\mathrm{direct}}^{n,sb} & ,5 \le sb < 64 \end{cases}$$

$$\tilde{\mathbf{z}}_{\mathrm{diffuse}}^{n,sb} = \begin{cases} 0 & ,0 \le sb < 5 \\ \mathbf{z}_{\mathrm{diffuse}}^{n,sb} & ,5 \le sb < 64 \end{cases}$$

by:

$$\tilde{\mathbf{z}}_{\mathrm{direct}}^{n,sb} = \begin{cases} \mathbf{z}_{\mathrm{direct}}^{n,sb} + \mathbf{z}_{\mathrm{diffuse}}^{n,sb} & ,0 \le sb < 5 \\ \mathbf{z}_{\mathrm{direct}}^{n,sb} & ,5 \le sb < 64 \\ \mathbf{z}_{\mathrm{direct}}^{n,sb} + \mathbf{z}_{\mathrm{diffuse}}^{n,sb} & ,64 \le sb \end{cases}$$

$$\tilde{\mathbf{z}}_{\mathrm{diffuse}}^{n,sb} = \begin{cases} 0 & ,0 \le sb < 5 \\ \mathbf{z}_{\mathrm{diffuse}}^{n,sb} & ,5 \le sb < 64 \\ 0 & ,64 \le sb \end{cases}$$

In subclause 6.7.3, in the second paragraph, replace:

$$0 \le n < numSlots \quad 6 \le sb < 24$$

by:

$$0 \le n < numSlots, \quad 6 \le sb \le 24$$

In subclause 6.7.4, replace:

where a bandpass factor BP^{sb} and a spectral flattening factor GF^{sb} are defined in Table 95.

Table 95 – Defining BP^{sb} and GF^{sb}

Sb	BP^{sb}	GF^{sb}	*Sb*	BP^{sb}	GF^{sb}
0	0.0000	0.0000	13	0.9984	0.0075
1	0.0005	0.0000	14	0.9908	0.0086
2	0.0092	0.0000	15	0.9639	0.0099
3	0.0587	0.0000	16	0.8952	0.0111
4	0.2580	0.0000	17	0.7711	0.0125
5	0.7392	0.0000	18	0.6127	0.0141
6	0.9791	0.0001	19	0.4609	0.0158
7	0.9993	0.0009	20	0.3391	0.0179
8	1.0000	0.0019	21	0.2493	0.0203
9	1.0000	0.0029	22	0.1848	0.0232
10	1.0000	0.0040	23	0.1387	0.0266
11	1.0000	0.0052	24	0.1053	0.0307
12	0.9999	0.0063			

by:

where a bandpass factor BP^{sb} and a spectral flattening factor GF^{sb} are defined in Table 95; the spectral flattening factors GF^{sb} have the same effect as if a flattened signal Z_{flat} was derived, having a flatter spectrum than the corresponding signal Z.

Table 95 – Definition of BP^{sb} and GF^{sb}

sb	BP^{sb}	GF^{sb}	*sb*	BP^{sb}	GF^{sb}
0	0.0000	0.00000000	13	0.9984	0.00005625
1	0.0005	0.00000000	14	0.9908	0.00007396
2	0.0092	0.00000000	15	0.9639	0.00009801
3	0.0587	0.00000000	16	0.8952	0.00012321
4	0.2580	0.00000000	17	0.7711	0.00015625
5	0.7392	0.00000000	18	0.6127	0.00019881
6	0.9791	0.00000001	19	0.4609	0.00024964
7	0.9993	0.00000081	20	0.3391	0.00032041

8	1.0000	0.00000361	21	0.2493	0.00041209
9	1.0000	0.00000841	22	0.1848	0.00053824
10	1.0000	0.00001600	23	0.1387	0.00070756
11	1.0000	0.00002704	24	0.1053	0.00094249
12	0.9999	0.00003969			

In subclause 6.7.4, replace the fourth equation:

Error! Objects cannot be created from editing field codes.

by:

Error! Objects cannot be created from editing field codes.

The value of **Error! Objects cannot be created from editing field codes.** for the first slot of the first frame is initialized as $Env_{\text{diffuse\_hold},ch} = \frac{1}{\varepsilon}$.

In subclause 6.7.5, in the first paragraph, replace:

For 5-1-5

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \varepsilon}} \qquad , ch \in \{ch_m\}$$

For 5-2-5, 7-2-7_1, 7-2-7_2, 7-5-7_1 and 7-5-7_2

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm\_l}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \varepsilon}} \quad , ch \in \{ch_l\}$$

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm\_r}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \varepsilon}} \quad , ch \in \{ch_r\}$$

by:

For 5-1-5

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \tilde{\varepsilon}}} \qquad , \tilde{\varepsilon} = 10^{-9}, ch \in \{ch_m\}$$

For 5-2-5, 7-2-7_1, 7-2-7_2, 7-5-7_1 and 7-5-7_2

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm\_l}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \tilde{\varepsilon}}} \quad , \tilde{\varepsilon} = 10^{-9}, ch \in \{ch_l\}$$

$$scale_{ch}^{n} = \sqrt{\frac{E_{\text{direct\_norm\_r}}^{n}}{E_{\text{diffuse\_norm},ch}^{n} + \tilde{\varepsilon}}} \quad , \tilde{\varepsilon} = 10^{-9}, ch \in \{ch_r\}$$

In subclause 6.7.5, in the last paragraph, replace:

initialized as **Error! Objects cannot be created from editing field codes.** .

by:

initialized as $scale_{\text{smooth},ch}(-1) = 1$.

In subclause 6.7.6, in the second paragraph, replace:

processed by STP, no scaling is performed:

Error! Objects cannot be created from editing field codes.

In the same step the diffuse signal is added to the direct signals as follows,

Error! Objects cannot be created from editing field codes.

For channels that are not processed by STP,

Error! Objects cannot be created from editing field codes.

by:

processed by STP no scaling is performed. In the same step the diffuse signal is added to the direct signals as follows:

$$\tilde{z}_{ch}^{n,sb} = \begin{cases} \tilde{z}_{\text{direct},ch}^{n,sb} + \tilde{z}_{\text{diffuse},ch}^{n,sb} \cdot scale_{\text{smooth},ch}(n) \cdot BP^{sb} & ,0 \le sb < 9 & ,if\ \textbf{bsTempShapeEnableChannel}(ch) = 1 \\ \tilde{z}_{\text{direct},ch}^{n,sb} + \tilde{z}_{\text{diffuse},ch}^{n,sb} \cdot scale_{\text{smooth},ch}(n) & ,9 \le sb < 64 & ,if\ \textbf{bsTempShapeEnableChannel}(ch) = 1 \\ \tilde{z}_{\text{direct},ch}^{n,sb} + \tilde{z}_{\text{diffuse},ch}^{n,sb} & ,64 \le sb & ,if\ \textbf{bsTempShapeEnableChannel}(ch) = 1 \\ \tilde{z}_{\text{direct},ch}^{n,sb} + \tilde{z}_{\text{diffuse},ch}^{n,sb} & ,0 \le sb & ,otherwise \end{cases}$$

In subclause 6.8.2, replace:

$$Env_{ZZZ,ch_{ZZZ}}(n)=\begin{cases}\dfrac{Env_{\mathrm{input},\mathrm{L}_0}(n)+Env_{\mathrm{input},\mathrm{R}_0}(n)}{2} & ,\begin{cases}ZZZ=\mathrm{input}\\ ch_{\mathrm{input}}=\mathrm{C}_0\end{cases}\\ \sqrt{\dfrac{EnvAbs_{ZZZ,ch_{ZZZ}}^{n}}{\overline{Env}_{ZZZ,ch_{ZZZ}}(n)}} & ,otherwise\end{cases}$$

by:

$$Env_{ZZZ,ch_{ZZZ}}(n)=\begin{cases}\dfrac{Env_{\mathrm{input},\mathrm{L}_0}(n)+Env_{\mathrm{input},\mathrm{R}_0}(n)}{2} & ,\begin{cases}ZZZ=\mathrm{input}\\ ch_{\mathrm{input}}=\mathrm{C}_0\end{cases}\\ \sqrt{\dfrac{EnvAbs_{ZZZ,ch_{ZZZ}}^{n}}{\overline{Env}_{ZZZ,ch_{ZZZ}}(n)+\varepsilon}} & ,otherwise\end{cases}$$

In subclause 6.8.2, the last equation, replace:

$$\overline{Env}_{ZZZ,ch_{ZZZ}}(-1)=32768^{2}$$

by:

$$\overline{Env}_{ZZZ,ch_{ZZZ}}(-1)=\frac{1}{\varepsilon}$$

In subclause 6.8.3.2, the second equation, replace:

$$ampRatio_{ch_{\mathrm{output}}}^{n}=\sqrt{\frac{\sum_{k=8}^{K-1}\left|y_{\mathrm{diffuse},ch_{\mathrm{output}}}^{n,k}\right|^{2}}{\sum_{k=8}^{K-1}\left|y_{\mathrm{direct},ch_{\mathrm{output}}}^{n,k}\right|^{2}+\varepsilon}}$$

$$g_{ch_{\mathrm{output}}}^{n}=\frac{envRatio_{ch_{\mathrm{output}}}^{n}\cdot Env_{\mathrm{output},Dch\left(ch_{\mathrm{output}}\right)}(n)}{Env_{\mathrm{input},ch_{\mathrm{output}}}(n)}$$

by:

$$ampRatio_{ch_{\mathrm{output}}}^{n}=\sqrt{\frac{\sum_{k=8}^{K-1}\left|y_{\mathrm{diffuse},ch_{\mathrm{output}}}^{n,k}\right|^{2}}{\sum_{k=8}^{K-1}\left|y_{\mathrm{direct},ch_{\mathrm{output}}}^{n,k}\right|^{2}+\varepsilon}}$$

$$g_{ch_{\mathrm{output}}}^{n}=\frac{envRatio_{ch_{\mathrm{output}}}^{n}\cdot Env_{\mathrm{output},Dch\left(ch_{\mathrm{output}}\right)}(n)}{Env_{\mathrm{input},ch_{\mathrm{output}}}(n)+\tilde{\varepsilon}}\quad,\tilde{\varepsilon}=10^{-9}$$

In subclause 6.9.1, in the second paragraph, replace:

The 1024 (or 2048 in case two elements have been decoded, see Table 22, Table 34 and Table 100) MDCT coefficients are first...

by:

The 1024 (or 2048 in case two elements have been decoded, see Table 34, Table 36 and Table 100) MDCT coefficients are first...

In subclause 6.10.3.2, replace:

$$\begin{cases} \mathbf{r}^{m}\left(0\right)=\mathbf{r}^{m}\left(-1\right)\alpha^{t(1)+1}+\sum_{n=0}^{t(1)}\alpha^{t(1)-n}\hat{w}_{\text{real}}^{n,m}\hat{w}_{\text{real}}^{n-1,m}, & l=0 \\ \mathbf{r}^{m}\left(l\right)=\mathbf{r}^{m}\left(l-1\right)\alpha^{t(l+1)+1}+\sum_{n=t(l)+1}^{t(l+1)}\alpha^{t(l+1)-n}\hat{w}_{\text{real}}^{n,m}\hat{w}_{\text{real}}^{n-1,m}, & 1\leq l<L \end{cases}$$

where

$\alpha=\exp\left(-\frac{dt}{\tau}\right)$ and $dt=\frac{64}{F_{\text{S}}}$ and $\tau=23.22$. Negative time index (-1) for a subband sample equals the last subband sample from the previous frame. In the same manner, negative time index for the correlation sum equals the last sum from the previous frame. Moreover, the real-valued QMF signal is computed as the sum of the downmix input channels, according to:

$$\hat{w}_{\text{real}}^{n,m}=\sum_{ch}\hat{\mathbf{x}}_{\text{real}}^{n,m}\left(ch\right)$$

The aliasing condition is set according to the following rule

$$\mathbf{a}^{m}\left(l\right)=\begin{cases} 1, & if \quad \mathbf{r}^{m}\left(l\right) > \quad thres,\ \mathbf{r}^{m+1}\left(l\right) > \quad thres,\ m\ odd \\ 1, & if \quad \mathbf{r}^{m}\left(l\right) < \ \text{-}thres,\ \mathbf{r}^{m+1}\left(l\right) < \ \text{-}thres,\ m\ even \\ 0, & otherwise \end{cases}$$

by:

$$\begin{cases} \mathbf{r}^{m}\left(0\right)=\mathbf{r}^{m}\left(-1\right)\alpha^{t(1)+1}+\sum_{n=0}^{t(1)}\alpha^{t(1)-n}w_{\text{real}}^{n,m}w_{\text{real}}^{n-1,m}, & l=0 \\ \mathbf{r}^{m}\left(l\right)=\mathbf{r}^{m}\left(l-1\right)\alpha^{t(l+1)+1}+\sum_{n=t(l)+1}^{t(l+1)}\alpha^{t(l+1)-n}w_{\text{real}}^{n,m}w_{\text{real}}^{n-1,m}, & 1\leq l<L \end{cases}$$

where $\alpha = \exp\left(-\frac{dt}{\tau}\right)$ and $dt = \frac{64}{F_{\mathrm{S}}}$ and $\tau = 23.22$. Negative time index (-1) for a subband sample equals the last subband sample from the previous frame. In the same manner, negative time index for the correlation sum equals the last sum from the previous frame. Moreover, the real-valued QMF signal is computed as the sum of the downmix input channels, according to:

$$w_{\mathrm{real}}^{n,m} = \sum_{ch} \hat{\mathbf{x}}_{\mathrm{real}}^{n-11,m}\left(ch\right)$$

The aliasing condition is set according to the following rule

$$\mathbf{a}^{m}\left(l\right) = \begin{cases} 1, & if \quad \mathbf{r}^{m}\left(l-1\right) \; > \; thres, \; \mathbf{r}^{m+1}\left(l-1\right) \; > \; thres, \; m \; odd \\ 1, & if \quad \mathbf{r}^{m}\left(l-1\right) \; < \; \text{-}thres, \; \mathbf{r}^{m+1}\left(l-1\right) \; < \; \text{-}thres, \; m \; even \\ 0, & otherwise \end{cases}$$

In subclause 6.10.4.2, replace:

$$\mathbf{M}_{1}^{n,k} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; \mathbf{t}(l-1) \le n < \mathbf{t}(l), 0 \le l < L, 0 \le k < K \, .$$

by:

$$\mathbf{M}_{1}^{n,k} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \mathbf{G}_{LP,1}^{l,m}, \; \mathbf{t}(l-1) \le n < \mathbf{t}(l), 0 \le l < L, 0 \le k < K \, ,$$

where

$$\mathbf{G}_{LP,1}^{l,m} = \begin{cases} \mathbf{G}_{1}^{l,m} & \mathbf{bsArbitraryDownmix} = 1 \; and \; \mathbf{bsArbitraryDownmixResidualBands} > 0, „ \\ 1 & otherwise. \end{cases}$$

In subclause 6.10.4.3, replace:

Where

$$\mathbf{R}_{2\mathrm{LP}}^{l,m} = \mathbf{R}_2^{l,m}\mathbf{I}_5\mathbf{R}_1^{l,m}\mathbf{G}_1^{l,m}$$

and where $\mathbf{R}_2^{l,m}$, $\mathbf{R}_1^{l,m}$, and $\mathbf{G}_1^{l,m}$ are defined in subclause 5.5, and where $\mathbf{I}_5$ is the 5x5 unity matrix.

by the following text, which also includes a new Table 102A:

Where

$$\mathbf{R}_{2\mathrm{LP}}^{l,m}(row,col) = \begin{cases} \mathbf{R}_2^{l,m}(row,col) & \text{for} \quad m < resBands_{col} \\ \hat{\mathbf{R}}_{2\mathrm{LP}}^{l,m}(row,col) & \text{for} \quad m \geq resBands_{col} \end{cases},$$

row and *col* are the row and column of the respective matrix element,

$$\hat{\mathbf{R}}_{2LP}^{l,m} = \mathbf{R}_2^{l,m} diag\left(\mathbf{R}_1^{l,m}\mathbf{G}_{LP,2}^{l,m}\right),$$

$$resBands_c = \begin{cases} \min\left(\mathbf{m}_{resProc}\left(resIdx(col)\right), M_{proc}^{c} - 1\right) & \text{if} \qquad col > 0 \\ 0 & \text{otherwise} \end{cases},$$

and *resIdx*(*col*) is defined according to Table 102A.

Table 102A — *resIdx*(*col*) mapping function

Col		1	2	3	4
resIdx(*col*)	515_1	0	1	3	2
	515_2	0	1	3	4

$\mathbf{R}_2^{l,m}$ and $\mathbf{R}_1^{l,m}$ are defined in subclause 6.5, and

$$\mathbf{G}_{LP,2}^{l,m} = \begin{cases} 1 & \textbf{bsArbitraryDownmix} = 1 \ \textit{and} \ \textbf{bsArbitraryDownmixResidualBands} > 0, \\ \mathbf{G}_1^{l,m} & \textit{otherwise.} \end{cases}$$

In subclause 6.10.5, replace:

For $m < M_{\mathrm{Proc}}^{c}$, i.e., for the complex-valued bands, the entries $h_{11}^{l.m}$, $h_{12}^{l.m}$, $h_{21}^{l.m}$ and $h_{22}^{l.m}$ are described by the equations in subclause 6.5.2.4. For $m \geq M_{\mathrm{proc}}^{\mathrm{c}}$, i.e., for the real-valued bands, the entries $h_{11}^{l.m}$, $h_{12}^{l.m}$, $h_{21}^{l.m}$ and $h_{22}^{l.m}$ are described as following.

by:

For $m < M_{\mathrm{proc}}^{\mathrm{c}} - 1$, i.e., for the complex-valued bands, the entries $h_{11}^{l.m}$, $h_{12}^{l.m}$, $h_{21}^{l.m}$ and $h_{22}^{l.m}$ are described by the equations in subclause 6.5.2.4. For $m \geq M_{\mathrm{proc}}^{\mathrm{c}} - 1$, i.e., for the real-valued bands, the entries $h_{11}^{l.m}$, $h_{12}^{l.m}$, $h_{21}^{l.m}$ and $h_{22}^{l.m}$ are described as following.

In subclause 6.10.7.1, append the following new paragraph:

For stereo output processing as defined in subclause 6.4.7.3, the low power parametric stereo decorrelator defined in subclause 6.10.7.2 shall be used as D_0.

In subclause 6.11.4.1.2, replace:

$$IPD_{\mathrm{B}}^{l,m} = \begin{cases} 0 & , m > 12, ICC_{\mathrm{Btmp}}^{l,m} < 0.6 \\ \arg\left(\left\langle L_{\mathrm{B}} R_{\mathrm{B}}^{*} \right\rangle^{l,m} \right) & , otherwise \end{cases} .$$

where

$$ICC_{\mathrm{Btmp}}^{l,m} = \frac{\left| \left\langle L_{\mathrm{B}} R_{\mathrm{B}}^{*} \right\rangle^{l,m} \right|}{\sigma_{\mathrm{L}}^{l,m} \sigma_{\mathrm{R}}^{l,m}}$$

and $\Re(x)$ denotes the real value of x. For high-frequency processing bands (band $M_{\mathrm{proc}}^{\mathrm{c}}$ onwards), the $\mathrm{IPD_B}$ value is set to zero.

by:

$$IPD_{\mathrm{B}}^{l,m} = \begin{cases} 0 & , m \geq 12, ICC_{\mathrm{Btmp}}^{l,m} < 0.6 \\ \arg\left(\left\langle L_{\mathrm{B}} R_{\mathrm{B}}^{*} \right\rangle^{l,m} \right) & , otherwise \end{cases} ,$$

where

$$ICC_{\mathrm{Btmp}}^{l,m} = \frac{\left| \left\langle L_{\mathrm{B}} R_{\mathrm{B}}^{*} \right\rangle^{l,m} \right|}{\sigma_{\mathrm{L}}^{l,m} \sigma_{\mathrm{R}}^{l,m}}$$

and $\Re(x)$ denotes the real value of x. For high-frequency processing bands (band m = 12 onwards), the $\mathrm{IPD_B}$ value is set to zero.

In subclause 6.11.4.1.3.1, replace the complete subclause by:

In order to define the (relative) power $\left(\sigma_{\mathrm{X}}^{l,m}\right)^2$ for $X \in \{\mathrm{L},\mathrm{R}\}$, i.e. the left and right 3D/binaural output signal, (with respect to the mono input signal), the transmitted CLD parameters are converted to (relative) signal powers $\left(\sigma_{\mathrm{X}}^{l,m}\right)^2$ for each virtual loudspeaker $X \in \{\mathrm{L},\mathrm{R},\mathrm{C},\mathrm{Ls},\mathrm{Rs}\}$, according to:

$$\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_1\left(CLD_1^{l,m}\right) r_1\left(CLD_3^{l,m}\right)$$

$$\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_1\left(CLD_1^{l,m}\right) r_2\left(CLD_3^{l,m}\right)$$

$$\left(\sigma_{\mathrm{C}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_2\left(CLD_1^{l,m}\right)$$

$$\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2 = r_2\left(CLD_0^{l,m}\right) r_1\left(CLD_2^{l,m}\right) g_s^2$$

$$\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2 = r_2\left(CLD_0^{l,m}\right) r_2\left(CLD_2^{l,m}\right) g_s^2$$

with

$$r_1\left(CLD\right) = \frac{10^{CLD/10}}{1+10^{CLD/10}},$$

and

$$r_2\left(CLD\right) = \frac{1}{1+10^{CLD/10}},$$

for $0 \le m < M_{proc}, 0 \le l < L$,

and where

$$CLD_X^{l,m} = \mathbf{D}_{CLD}\left(X,l,m\right), \quad 0 \le X < 4, 0 \le m < M_{proc}, 0 \le l < L.$$

The constants g_s represents the surround gain factor according to **bsFixedGainsSur** as defined in Table 43.

Given the above, and the HRTF parameters $P_{\mathrm{Y,X}}^{m}$, ϕ_{X}^{m} and ρ_{X}^{m}, the (relative) power $\left(\sigma_{\mathrm{L}}^{l,m}\right)^2$ and $\left(\sigma_{\mathrm{R}}^{l,m}\right)^2$ for the left and right 3D/binaural output signal (with respect to the mono input signal) is given by:

$$\begin{aligned}\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 = {} & \left(P_{\mathrm{L,C}}^{m}\right)^2\left(\sigma_{\mathrm{C}}^{l,m}\right)^2+\left(P_{\mathrm{L,L}}^{m}\right)^2\left(\sigma_{\mathrm{L}}^{l,m}\right)^2+\left(P_{\mathrm{L,Ls}}^{m}\right)^2\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2+\left(P_{\mathrm{L,R}}^{m}\right)^2\left(\sigma_{\mathrm{R}}^{l,m}\right)^2+\left(P_{\mathrm{L,Rs}}^{m}\right)^2\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2+\ldots\\ & 2P_{\mathrm{L,L}}^{m}P_{\mathrm{L,R}}^{m}\rho_{\mathrm{R}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_3^{l,m}\cos\left(\phi_{\mathrm{R}}^{m}\right)+\ldots\\ & 2P_{\mathrm{L,Ls}}^{m}P_{\mathrm{L,Rs}}^{m}\rho_{\mathrm{Rs}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_2^{l,m}\cos\left(\phi_{\mathrm{Rs}}^{m}\right)\end{aligned}$$

$$\begin{aligned}\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 = {} & \left(P_{\mathrm{R,C}}^{m}\right)^2\left(\sigma_{\mathrm{C}}^{l,m}\right)^2+\left(P_{\mathrm{R,L}}^{m}\right)^2\left(\sigma_{\mathrm{L}}^{l,m}\right)^2+\left(P_{\mathrm{R,Ls}}^{m}\right)^2\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2+\left(P_{\mathrm{R,R}}^{m}\right)^2\left(\sigma_{\mathrm{R}}^{l,m}\right)^2+\left(P_{\mathrm{R,Rs}}^{m}\right)^2\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2+\ldots\\ & 2P_{\mathrm{R,L}}^{m}P_{\mathrm{R,R}}^{m}\rho_{\mathrm{L}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_3^{l,m}\cos\left(\phi_{\mathrm{L}}^{m}\right)+\ldots\\ & 2P_{\mathrm{R,Ls}}^{m}P_{\mathrm{R,Rs}}^{m}\rho_{\mathrm{Ls}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_2^{l,m}\cos\left(\phi_{\mathrm{Ls}}^{m}\right)\end{aligned}$$

for $0 \le m < M_{\mathrm{Proc}}, 0 \le l < L$,

and where

$$ICC_3^{l,m} = \mathbf{D}_{ICC}\left(3,l,m\right), \quad ,0 \le m < M_{proc}, 0 \le l < L$$

$$ICC_2^{l,m} = \mathbf{D}_{ICC}\left(2,l,m\right), \quad ,0 \le m < M_{proc}, 0 \le l < L\,.$$

The complex cross-spectrum $\left\langle L_{\mathrm{B}}R_{\mathrm{B}}^{*}\right\rangle^{l,m}$ for each spatial parameter set is estimated for each parameter position and processing band as:

$$\begin{aligned}\left\langle L_{\mathrm{B}}R_{\mathrm{B}}^{*}\right\rangle = {} & \left(\sigma_{\mathrm{C}}^{l,m}\right)^2P_{\mathrm{L,C}}^{m}P_{\mathrm{R,C}}^{m}\rho_{\mathrm{C}}^{m}\exp\left(j\phi_{\mathrm{C}}^{m}\right)+\ldots\\ & \left(\sigma_{\mathrm{L}}^{l,m}\right)^2P_{\mathrm{L,L}}^{m}P_{\mathrm{R,L}}^{m}\rho_{\mathrm{L}}^{m}\exp\left(j\phi_{\mathrm{L}}^{m}\right)+\ldots\\ & \left(\sigma_{\mathrm{R}}^{l,m}\right)^2P_{\mathrm{L,R}}^{m}P_{\mathrm{R,R}}^{m}\rho_{\mathrm{R}}^{m}\exp\left(j\phi_{\mathrm{R}}^{m}\right)+\ldots\\ & \left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2P_{\mathrm{L,Ls}}^{m}P_{\mathrm{R,Ls}}^{m}\rho_{\mathrm{Ls}}^{m}\exp\left(j\phi_{\mathrm{Ls}}^{m}\right)+\ldots\\ & \left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2P_{\mathrm{L,Rs}}^{m}P_{\mathrm{R,Rs}}^{m}\rho_{\mathrm{Rs}}^{m}\exp\left(j\phi_{\mathrm{Rs}}^{m}\right)+\ldots\\ & P_{\mathrm{L,L}}^{m}P_{\mathrm{R,R}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_3+\ldots\\ & P_{\mathrm{L,Ls}}^{m}P_{\mathrm{R,Rs}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_2+\ldots\\ & P_{\mathrm{L,R}}^{m}P_{\mathrm{R,L}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_3\rho_{\mathrm{L}}^{m}\rho_{\mathrm{R}}^{m}\exp\left(j\left(\phi_{\mathrm{L}}^{m}+\phi_{\mathrm{R}}^{m}\right)\right)+\ldots\\ & P_{\mathrm{L,Rs}}^{m}P_{\mathrm{R,Ls}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_2\rho_{\mathrm{Ls}}^{m}\rho_{\mathrm{Rs}}^{m}\exp\left(j\left(\phi_{\mathrm{Ls}}^{m}+\phi_{\mathrm{Rs}}^{m}\right)\right)\end{aligned}$$

for $0 \le m < M_{\mathrm{Proc}}, 0 \le l < L$,

and where

$$ICC_3^{l,m} = \mathbf{D}_{ICC}\left(3,l,m\right), \quad ,0 \le m < M_{proc}, 0 \le l < L$$

$$ICC_2^{l,m} = \mathbf{D}_{ICC}\left(2,l,m\right), \quad ,0 \le m < M_{proc}, 0 \le l < L\,.$$

In subclause 6.11.4.1.3.2, replace the complete subclause by:

Similar to the 5151 configuration, in order to define the (relative) power $\left(\sigma_{\mathrm{X}}^{l,m}\right)^2$ for X=L,R, i.e. the left and right 3D/binaural output signal, (with respect to the mono input signal), the transmitted CLD parameters are converted to (relative) signal powers $\left(\sigma_{\mathrm{X}}^{l,m}\right)^2$ for each virtual loudspeaker $X = \{L, R, C, Ls, Rs\}$, according to:

$$\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_1\left(CLD_1^{l,m}\right) r_1\left(CLD_3^{l,m}\right)$$

$$\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_2\left(CLD_1^{l,m}\right) r_1\left(CLD_4^{l,m}\right)$$

$$\left(\sigma_{\mathrm{C}}^{l,m}\right)^2 = r_2\left(CLD_0^{l,m}\right)$$

$$\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_1\left(CLD_1^{l,m}\right) r_2\left(CLD_3^{l,m}\right) g_s^2$$

$$\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2 = r_1\left(CLD_0^{l,m}\right) r_2\left(CLD_1^{l,m}\right) r_2\left(CLD_4^{l,m}\right) g_s^2$$

with

$$r_1\left(CLD\right) = \frac{10^{CLD/10}}{1+10^{CLD/10}} \text{ ,}$$

and

$$r_2\left(CLD\right) = \frac{1}{1+10^{CLD/10}} \text{ ,}$$

for $0 \le m < M_{proc}, 0 \le l < L$,

and where

$$CLD_X^{l,m} = \mathbf{D}_{CLD}\left(X, l, m\right), \quad 0 \le X < 5, 0 \le m < M_{proc}, 0 \le l < L \text{ .}$$

The constant g_s represents the surround gain factor according to **bsFixedGainsSur** as defined in Table 43.

Given the above, and the HRTF parameters $P_{\mathrm{Y,X}}^m$, ϕ_{X}^m and ρ_{X}^m , the (relative) power $\left(\sigma_{\mathrm{L}}^{l,m}\right)^2$ and $\left(\sigma_{\mathrm{R}}^{l,m}\right)^2$ for the left and right 3D/binaural output signals (with respect to the mono input signal) is given by:

$$\begin{aligned}\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 = {} & \left(P_{\mathrm{L,C}}^m\right)^2\left(\sigma_{\mathrm{C}}^{l,m}\right)^2 + \left(P_{\mathrm{L,L}}^m\right)^2\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 + \left(P_{\mathrm{L,Ls}}^m\right)^2\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2 + \left(P_{\mathrm{L,R}}^m\right)^2\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 + \left(P_{\mathrm{L,Rs}}^m\right)^2\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2 + \ldots \\ & 2P_{\mathrm{L,L}}^m P_{\mathrm{L,R}}^m \rho_{\mathrm{R}}^m \sigma_{\mathrm{L}}^{l,m} \sigma_{\mathrm{R}}^{l,m} ICC_1^{l,m} \cos\left(\phi_{\mathrm{R}}^m\right) + \ldots \\ & 2P_{\mathrm{L,Ls}}^m P_{\mathrm{L,Rs}}^m \rho_{\mathrm{Rs}}^m \sigma_{\mathrm{Ls}}^{l,m} \sigma_{\mathrm{Rs}}^{l,m} ICC_1^{l,m} \cos\left(\phi_{\mathrm{Rs}}^m\right)\end{aligned}$$

$$\begin{aligned}\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 = \; & \left(P_{\mathrm{R,C}}^{m}\right)^2\left(\sigma_{\mathrm{C}}^{l,m}\right)^2 + \left(P_{\mathrm{R,L}}^{m}\right)^2\left(\sigma_{\mathrm{L}}^{l,m}\right)^2 + \left(P_{\mathrm{R,Ls}}^{m}\right)^2\left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2 + \left(P_{\mathrm{R,R}}^{m}\right)^2\left(\sigma_{\mathrm{R}}^{l,m}\right)^2 + \left(P_{\mathrm{R,Rs}}^{m}\right)^2\left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2 + \ldots \\ & 2P_{\mathrm{R,L}}^{m}P_{\mathrm{R,R}}^{m}\rho_{\mathrm{L}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_{1}^{l,m}\cos\left(\phi_{\mathrm{L}}^{m}\right) + \ldots \\ & 2P_{\mathrm{R,Ls}}^{m}P_{\mathrm{R,Rs}}^{m}\rho_{\mathrm{Ls}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_{1}^{l,m}\cos\left(\phi_{\mathrm{Ls}}^{m}\right)\end{aligned}$$

for $0 \le m < M_{\mathrm{Proc}}, 0 \le l < L$,

and where

$$ICC_{1}^{l,m} = \mathbf{D}_{\mathrm{ICC}}\left(1,l,m\right), \quad ,0 \le m < M_{\mathrm{Proc}}, 0 \le l < L$$

The complex cross-spectrum $\left\langle L_{\mathrm{B}}R_{\mathrm{B}}^{*}\right\rangle^{l,m}$ for each spatial parameter set is estimated for each parameter position and processing band as:

$$\begin{aligned}\left\langle L_{\mathrm{B}}R_{\mathrm{B}}^{*}\right\rangle = \; & \left(\sigma_{\mathrm{C}}^{l,m}\right)^2 P_{\mathrm{L,C}}^{m}P_{\mathrm{R,C}}^{m}\rho_{\mathrm{C}}^{m}\exp\left(j\phi_{\mathrm{C}}^{m}\right) + \ldots \\ & \left(\sigma_{\mathrm{L}}^{l,m}\right)^2 P_{\mathrm{L,L}}^{m}P_{\mathrm{R,L}}^{m}\rho_{\mathrm{L}}^{m}\exp\left(j\phi_{\mathrm{L}}^{m}\right) + \ldots \\ & \left(\sigma_{\mathrm{R}}^{l,m}\right)^2 P_{\mathrm{L,R}}^{m}P_{\mathrm{R,R}}^{m}\rho_{\mathrm{R}}^{m}\exp\left(j\phi_{\mathrm{R}}^{m}\right) + \ldots \\ & \left(\sigma_{\mathrm{Ls}}^{l,m}\right)^2 P_{\mathrm{L,Ls}}^{m}P_{\mathrm{R,Ls}}^{m}\rho_{\mathrm{Ls}}^{m}\exp\left(j\phi_{\mathrm{Ls}}^{m}\right) + \ldots \\ & \left(\sigma_{\mathrm{Rs}}^{l,m}\right)^2 P_{\mathrm{L,Rs}}^{m}P_{\mathrm{R,Rs}}^{m}\rho_{\mathrm{Rs}}^{m}\exp\left(j\phi_{\mathrm{Rs}}^{m}\right) + \ldots \\ & P_{\mathrm{L,L}}^{m}P_{\mathrm{R,R}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_{1} + \ldots \\ & P_{\mathrm{L,Ls}}^{m}P_{\mathrm{R,Rs}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_{1} + \ldots \\ & P_{\mathrm{L,R}}^{m}P_{\mathrm{R,L}}^{m}\sigma_{\mathrm{L}}^{l,m}\sigma_{\mathrm{R}}^{l,m}ICC_{1}\rho_{\mathrm{L}}^{m}\rho_{\mathrm{R}}^{m}\exp\left(j\left(\phi_{\mathrm{L}}^{m} + \phi_{\mathrm{R}}^{m}\right)\right) + \ldots \\ & P_{\mathrm{L,Rs}}^{m}P_{\mathrm{R,Ls}}^{m}\sigma_{\mathrm{Ls}}^{l,m}\sigma_{\mathrm{Rs}}^{l,m}ICC_{1}\rho_{\mathrm{Ls}}^{m}\rho_{\mathrm{Rs}}^{m}\exp\left(j\left(\phi_{\mathrm{Ls}}^{m} + \phi_{\mathrm{Rs}}^{m}\right)\right)\end{aligned}$$

for $0 \le m < M_{\mathrm{Proc}}, 0 \le l < L$,

and where

$$ICC_{1}^{l,m} = \mathbf{D}_{\mathrm{ICC}}\left(1,l,m\right), \quad ,0 \le m < M_{\mathrm{Proc}}, 0 \le l < L.$$

In subclause 6.11.4.2.1, replace:

The stereo based binaural decoder can be implemented by a single-slot parametric approach (similar to the mono configurations), or by a multi-slot convolution approach. The stereo based configuration requires a (conventional) stereo (MPEG surround) downmix as input signal $y_{\mathrm{L}_0}^{n,k}$, $y_{\mathrm{R}_0}^{n,k}$, combined with spatial parameters that enable a 5.1-channel reconstruction. The downmix signal can be a matrixed surround compatible stereo downmix.

by:

The stereo based binaural decoder can be implemented by a single-slot parametric approach (similar to the mono configurations), or by a multi-slot convolution approach. The stereo based configuration requires spatial parameters that enable a 5.1-channel reconstruction. For bitstreams using a 7-2-7 configuration, the underlying 7-2-7_1 or 7-2-7_2 decoder tree structure is pruned to be congruent to the 5-2-5 structure (see also subclause 6.4.8), and hence the spatial parameters attributed to the OTT boxes OTT_3 and OTT_4 are not used in the binaural mixing process. The stereo based configuration furthermore requires a (conventional) stereo (MPEG surround) downmix as input signal $y_{L_0}^{n,k}$, $y_{R_0}^{n,k}$. The downmix signal can be a matrix surround compatible stereo downmix.

In subclause 6.11.4.2.1, in the second paragraph, replace:

For both configurations (normal downmix or matrixed surround compatible downmix),

by:

For both configurations (normal downmix or matrix surround compatible downmix),

In subclause 6.11.4.2.2.1, replace:

The HRTF parameters $P_{Y,X}^{m}$, ϕ_{X}^{m} and ρ_{X}^{m}, for $X \in \{L,R,C,Ls,Rs\}$ ...

by:

The HRTF parameters $P_{Y,X}^{m}$ and ϕ_{X}^{m}, for $X \in \{L,R,C,Ls,Rs\}$ ...

In subclause 6.11.4.2.2.2, replace:

$$\left(\sigma_{Ls}^{l,m}\right)^2 = r_2\left(CLD_1^{l,m}\right) / g_s^2$$

by:

$$\left(\sigma_{Ls}^{l,m}\right)^2 = r_2\left(CLD_1^{l,m}\right) \cdot g_s^2$$

In subclause 6.11.4.2.2.2, replace:

$$\left(\sigma_{Rs}^{l,m}\right)^2 = r_2\left(CLD_2^{l,m}\right) / g_s^2$$

by:

$$\left(\sigma_{Rs}^{l,m}\right)^2 = r_2\left(CLD_2^{l,m}\right) \cdot g_s^2$$

In subclause 6.11.4.2.2.2, replace:

$$h_{\mathrm{L,C}}^{l,m} = P_{\mathrm{L,C}}^{m} e^{+j\phi_{\mathrm{C}}^{m}/2} \frac{1}{g_c}$$

by:

$$h_{L,C}^{l,m} = P_{L,C}^{m} \cdot e^{+j\phi_{C}^{m}/2}$$

In subclause 6.11.4.2.2.2, replace:

$$h_{\mathrm{R,C}}^{l,m} = P_{\mathrm{R,C}}^{m} e^{+j\phi_{\mathrm{C}}^{m}/2} \frac{1}{g_c}$$

by:

$$h_{R,C}^{l,m} = P_{R,C}^{m} \cdot e^{-j\phi_{C}^{m}/2}$$

In subclause 6.11.4.2.2.2, replace:

$$\mathbf{H}_1^{l,k} = \begin{bmatrix} \mathbf{h}_{11}^{l,k} & \mathbf{h}_{12}^{l,k} \\ \mathbf{h}_{21}^{l,k} & \mathbf{h}_{22}^{l,k} \end{bmatrix} = \begin{bmatrix} h_{\mathrm{L,L}}^{l,\kappa(k)} & h_{\mathrm{L,R}}^{l,\kappa(k)} & h_{\mathrm{L,C}}^{l,\kappa(k)} \\ h_{\mathrm{R,L}}^{l,\kappa(k)} & h_{\mathrm{L,R}}^{l,\kappa(k)} & h_{\mathrm{R,C}}^{l,\kappa(k)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{W}_{\mathrm{temp}}^{l,\kappa(k)}, \quad 0 \le k < K, 0 \le l < L$$

by:

$$\mathbf{H}_1^{l,k} = \begin{bmatrix} \mathbf{h}_{11}^{l,k} & \mathbf{h}_{12}^{l,k} \\ \mathbf{h}_{21}^{l,k} & \mathbf{h}_{22}^{l,k} \end{bmatrix} = \begin{bmatrix} h_{\mathrm{L,L}}^{l,\kappa(k)} & h_{\mathrm{L,R}}^{l,\kappa(k)} & h_{\mathrm{L,C}}^{l,\kappa(k)} \\ h_{\mathrm{R,L}}^{l,\kappa(k)} & h_{\mathrm{R,R}}^{l,\kappa(k)} & h_{\mathrm{R,C}}^{l,\kappa(k)} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{W}_{\mathrm{temp}}^{l,\kappa(k)}, \quad 0 \le k < K, 0 \le l < L$$

In subclause 6.11.4.2.2.2, in the last paragraph, replace:

where $\mathbf{W}_{\mathrm{temp}}$ is the upmix matrix as defined in subclause 6.5.3.

by:

where $\mathbf{W}_{\mathrm{temp}}$ is the upmix matrix as defined in subclause 6.5.2.

In subclause 6.11.4.2.3.4, replace:

The relative powers of front vs. surround pairs (e.g., $\left(\sigma_{\mathrm{L}}^{l,k}\right)^2 \Big/ \left(\sigma_{\mathrm{Ls}}^{l,k}\right)^2$ and $\left(\sigma_{\mathrm{R}}^{l,k}\right)^2 \Big/ \left(\sigma_{\mathrm{Rs}}^{l,k}\right)^2$) is similarly to the single slot case reconstructed on the basis of transmitted CLD parameters, $\mathbf{D}_{\mathrm{CLD}}(1,l,m)$ and $\mathbf{D}_{\mathrm{CLD}}(2,l,m)$, albeit mapped from the parameter band resolution to the hybrid filterbank resolution, according to:

$$\left(\sigma_{\mathrm{L}}^{l,k}\right)^2 = r_1\left(CLD_1^{l,\kappa(k)}\right) \qquad \left(\sigma_{\mathrm{Ls}}^{l,k}\right)^2 = r_2\left(CLD_1^{l,\kappa(k)}\right)\Big/ g_s^2$$

$$\left(\sigma_{\mathrm{R}}^{l,k}\right)^2 = r_1\left(CLD_2^{l,\kappa(k)}\right) \qquad \left(\sigma_{\mathrm{Rs}}^{l,k}\right)^2 = r_2\left(CLD_2^{l,\kappa(k)}\right)\Big/ g_s^2$$

with g_s the down mix gain for surround channels, and where

by:

The relative powers of front vs. surround pairs (e.g., $\left(\sigma_{\mathrm{L}}^{l,k}\right)^2 \Big/ \left(\sigma_{\mathrm{Ls}}^{l,k}\right)^2$ and $\left(\sigma_{\mathrm{R}}^{l,k}\right)^2 \Big/ \left(\sigma_{\mathrm{Rs}}^{l,k}\right)^2$) is similarly to the single slot case reconstructed on the basis of transmitted CLD parameters, $\mathbf{D}_{\mathrm{CLD}}(1,l,m)$ and $\mathbf{D}_{\mathrm{CLD}}(2,l,m)$, albeit mapped from the parameter band resolution to the hybrid filterbank resolution, according to:

$$\left(\sigma_{\mathrm{L}}^{l,k}\right)^2 = r_1\left(CLD_1^{l,\kappa(k)}\right) \qquad \left(\sigma_{\mathrm{Ls}}^{l,k}\right)^2 = r_2\left(CLD_1^{l,\kappa(k)}\right) g_s^2$$

$$\left(\sigma_{\mathrm{R}}^{l,k}\right)^2 = r_1\left(CLD_2^{l,\kappa(k)}\right) \qquad \left(\sigma_{\mathrm{Rs}}^{l,k}\right)^2 = r_2\left(CLD_2^{l,\kappa(k)}\right) g_s^2$$

with g_s the surround gain factor according to **bsFixedGainsSur** as defined in Table 43, and where

In subclause 6.11.4.2.3.4, replace:

where $\mathbf{W}_{\mathrm{temp}}$ is the upmix matrix as defined in subclause 6.5.3.

by:

where $\mathbf{W}_{\mathrm{temp}}$ is the upmix matrix as defined in subclause 6.5.2.

In subclause 6.11.4.2.3.4, replace:

Second, energy estimates are formed for $Y \in \{\mathrm{L,R}\}$ according to

$$E_{\mathrm{Y}}^{l,k} = b^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,L}}^{l,k}\right\|^2 + a^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,R}}^{l,k}\right\|^2 + a^{l,k}b^{l,k}\left\|\mathbf{h}_{\mathrm{Y,C}}^{l,k}\right\|^2,$$

$$\Delta E_{\mathrm{Y}}^{l,k} = a^{l,k}b^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,L}}^{l,k} + \mathbf{h}_{\mathrm{Y,R}}^{l,k} - \mathbf{h}_{\mathrm{Y,C}}^{l,k}\right\|^2,$$

by:

Second, energy estimates are formed for $Y = \{\mathrm{L,R}\}$ according to

$$E_{\mathrm{Y}}^{l,k} = b^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,L}}^{l,k}\right\|^2 + a^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,R}}^{l,k}\right\|^2 + 2\cdot a^{l,k}b^{l,k}\left\|\mathbf{h}_{\mathrm{Y,C}}^{l,k}\right\|^2,$$

$$\Delta E_{\mathrm{Y}}^{l,k} = a^{l,k}b^{l,k}\left(1-c^{l,k}\right)\left\|\mathbf{h}_{\mathrm{Y,L}}^{l,k} + \mathbf{h}_{\mathrm{Y,R}}^{l,k} - \sqrt{2}\cdot\mathbf{h}_{\mathrm{Y,C}}^{l,k}\right\|^2,$$

In subclause 6.11.5, replace the title:

3D decoder for 5.1 channel reconstruction

by:

3D decoder for multi-channel reconstruction

In subclause 6.11.5, replace:

The matrix operation Q should not be applied individually but can be merged in the subsequent matrix operations of the spatial decoder. Specifically, the signals are processed by a 3x2 matrix M_1 when entering the spatial decoder. For 3D downmix inversion, the matrix $\mathbf{M}_1'$ of the spatial decoder that should be applied comprises the product of the 3D inversion matrix and the original matrix $\mathbf{M}_1$:

$$\mathbf{M}_1' = \mathbf{M}_1\mathbf{Q}\,.$$

The processing matrix $\mathbf{Q}$ is derived from the spatial parameters and the HRTF parameters. $\mathbf{Q}$ is the inverse matrix of the encoder processing matrix $\mathbf{H}$:

by:

The matrix operation $\mathbf{Q}$ should not be applied individually but can be merged in the subsequent matrix operations of the spatial decoder. Specifically, the downmix input signals are processed by a 3x2 matrix M_1 as defined in subclause 6.5.2. In case of 3D inversion, the elements of matrix **Q** replace the elements of the matrix compatibility inversion matrix **H**:

$$h_{ij}^{l,m} = q_{ij}^{l,m}, \quad i,j \in \{1,2\}$$

The processing matrix $\mathbf{Q}$ is derived from the spatial parameters and the HRTF parameters. $\mathbf{Q}$ is the inverse matrix of the encoder processing matrix $\mathbf{H}_1$.

In subclause 6.11.5, replace:

$$\mathbf{Q}_{\mathrm{tmp}}^{l,k} = \left(\mathbf{H}^{l,k}\right)^{-1} = \begin{bmatrix} q_{11}^{l,k} & q_{12}^{l,k} \\ q_{21}^{l,k} & q_{22}^{l,k} \end{bmatrix}$$

by:

$$\mathbf{Q}_{\mathrm{tmp}}^{l,k} = \left(\mathbf{H}_1^{l,k}\right)^{-1} = \begin{bmatrix} q_{11}^{l,k} & q_{12}^{l,k} \\ q_{21}^{l,k} & q_{22}^{l,k} \end{bmatrix}$$

In subclause 6.11.5, replace:

decoder the HRTF parameters $P_{Y,X}^{m}$, ϕ_{X}^{m} and ρ_{X}^{m} can be transmitted in the bitstream and are hence defined dependant on the bitstream elements as:

$$P_{L,X}^{m} = 10^{\frac{\mathbf{bsHRTFlevelleft}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) - 16}{20}}$$

$$P_{\mathrm{R},X}^{m} = \begin{cases} P_{\mathrm{L},i(X)}^{m} & , if\ bsHRFTasymmetric = 0 \\ 10^{\frac{\mathbf{bsHRTFlevelright}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) - 46}{20}} & , if\ bsHRFTasymmetric = 1 \end{cases}$$

$$\phi_{X}^{m} = \begin{cases} \frac{\pi}{32} \cdot \mathbf{bsHRTFphaseLR}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) & , if\ \mathbf{bsHRTFphase}\left(X\right) = 1, m < 13 \\ \boldsymbol{\varphi}_{\mathrm{default}}\left(X, m\right) & , if\ \mathbf{bsHRTFphase}\left(X\right) = 0, m < 13 \\ 0 & , m \geq 13 \end{cases}$$

$$\rho_{X}^{m} = \begin{cases} deq\left(\mathbf{bsHRTiccLR}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right), ICC\right) & , if\ \mathbf{bsHRTFicc}\left(X\right) = 1 \\ \boldsymbol{\rho}_{\mathrm{default}}\left(X, m\right) & , if\ \mathbf{bsHRTFicc}\left(X\right) = 0 \end{cases}$$

for $0 \leq m < M_{\mathrm{proc}}$

by:

decoder the HRTF parameters $P_{Y,X}^{m}$ and ϕ_{X}^{m} can be transmitted in the bitstream and are hence defined dependent on the bitstream elements as:

$$P_{L,X}^{m} = 10^{\frac{\mathbf{bsHRTFlevelleft}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) - 16}{20}}$$

$$P_{\mathrm{R},X}^{m} = \begin{cases} P_{\mathrm{L},i(X)}^{m} & , if\ bsHRFTasymmetric = 0 \\ 10^{\frac{\mathbf{bsHRTFlevelright}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) - 46}{20}} & , if\ bsHRFTasymmetric = 1 \end{cases}$$

$$\phi_{X}^{m} = \begin{cases} \frac{\pi}{32} \cdot \mathbf{bsHRTFphaseLR}\left(X, mapfunc\left(m, M_{\mathrm{HRTF}}\right)\right) & , if\ \mathbf{bsHRTFphase}\left(X\right) = 1, m < 13 \\ \boldsymbol{\varphi}_{\mathrm{default}}\left(X, m\right) & , if\ \mathbf{bsHRTFphase}\left(X\right) = 0, m < 13 \\ 0 & , m \geq 13 \end{cases}$$

for $0 \leq m < M_{\mathrm{proc}}$

In subclause 6.11.5, replace:

The default tables $\boldsymbol{\varphi}_{\mathrm{default}}\left(X,m\right)$ and $\boldsymbol{\rho}_{\mathrm{default}}\left(X,m\right)$ are defined in Table A.34 and Table A.35.

by:

The default table $\boldsymbol{\varphi}_{\mathrm{default}}\left(X,m\right)$ is defined in Table A.34.

In subclause 7.2.4, replace:

```
Syntax                                    No. of bits   Mnemonic
AncDataElement()
{
    ancSyncword; /* 0x8E4 */                  12          bslbf
    ancType;                                  2           uimsbf
    ancStart;                                 1           uimsbf
    ancStop;                                  1           uimsbf
    cnt = ancLenBytes;                        8           uimsbf
    if (cnt==255) {
        cnt += ancLenBytesAdd;                16          uimsbf
    }
    ancCrcWord;                               8           uimsbf
    for (i=0; i<cnt; i++) {
        ancDataSegmentByte[i];                8           bslbf
    }
}
```

by:

```
Syntax                                    No. of bits   Mnemonic
AncDataElement()
{
    ancSyncword; /* 0x472 */                  11          bslbf
    ancCrcLen;                                1           uimsbf
    ancType;                                  2           uimsbf
    ancStart;                                 1           uimsbf
    ancStop;                                  1           uimsbf
    cnt = ancLenBytes;                        8           uimsbf
    if (cnt==255) {
        cnt += ancLenBytesAdd;                16          uimsbf
    }
    if ( ancCrcLen == 0 ) {
        ancCrcWord;                           8           rpchof
    }
    else {
        ancCrcWord;                           16          rcphof
    }
    for (i=0; i<cnt; i++) {
        ancDataSegmentByte[i];                8           bslbf
    }
}
```

In subclause 7.2.4, replace:

ancSyncword	Identification syncword. Shall be 0x8E4.

by:

ancSyncword	Identification syncword. Shall be 0x472.
ancCrcLen	Indicates the length of ancCrcWord: 0: 8 bit, 1: 16 bit.

In subclause 7.2.4, replace:

ancCrcWord	ancCrcWord is defined by the generator polynomial G(x) = x^8+x^2+x+1 and the initial value for the CRC calculation is 0xFF. The CRC covers all bits in the AncDataElement() excluding ancSyncword and the ancCrcWord itself.

by:

ancCrcWord	ancCrcWord is defined by the generator polynomial G(x) = x^8+x^2+x+1 and the initial value for the CRC calculation is 0xFF if an 8 bit CRC is signaled by ancCrcLen. For the 16 bit CRC, the generator polynomial is G(x) = x^16 + x^15 + x^2 + 1 and the initial value is 0xFFFF. The CRC covers all bits in the AncDataElement() excluding ancSyncword, ancCrcLen, and the ancCrcWord itself.

In subclause 7.2.5, Table 109, replace:

sacTimeAlign	**16**	**uimsbf**

by:

sacTimeAlign	**16**	**simsbf**

In subclause 7.2.5, replace:

sacTimeAlign	Identifies the PCM sample …

by:

sacTimeAlign	Signed integer in the range -32768..32767, identifying the PCM sample...

In subclause 7.3.2, Table 111, change the mnemonic for element **bsDBHeaderCrc** *from* UiMsbf *to* rcphof*.*

In subclause 7.3.2, Table 113, change the mnemonic for element **bsDBDataCrc** *from* UiMsbf *to* rcphof*.*

In subclause 7.3.3, in the ***bsBDSubframes*** *description, replace:*

multiple of 64 are allowed In addition

by:

multiple of 64 are allowed. In addition

In subclause 7.3.3, in the description of ***bsBDAlloc****, replace:*

The three bits constitute an unsigned integer with values 0...7.

by:

The four bits constitute an unsigned integer with values 0...15.

In subclause 7.3.3, in the description of ***bsBDType****, replace:*

The types 1..5 are reserved

by:

The types 2...5 are reserved

In subclause 7.3.4.7, replace:

In the case that the buried data payload is of type MPEG Surround, the SpatialFrame data buried in one PCM frame shall be applied to the previously received PCM frame.

by:

In the case that the buried data payload is of type MPEG Surround frame or MPEG Surround header+frame, the SpatialFrame data buried in one PCM buried data frame shall be applied to a PCM frame having the same length as the PCM buried data frame and having an offset in PCM samples specified by the value of sacTimeAlign (see subclause 7.2.5).

In Annex A.1, replace Table A.1:

Table name	Offset	LAV	Notes
hcodFirstBand_CLD	15	15	
hcodFirstBand_ICC	0	7	Note 1
hcodFirstBand_CPC	10	15	
hcod1D_CLD_YY	0	15	Note 1
hcod1D_ICC_YY	0	7	Note 1
hcod1D_CPC_YY	0	15	Note 1
hcond2D_CLD_YY_ZZ_LL_escape	N/A	N/A	
hcond2D_ICC_YY_ZZ_LL_escape	N/A	N/A	
hcond2D_CPC_YY_ZZ_LL_escape	N/A	N/A	
hcond2D_CLD_YY_ZZ_03	0	3	Note 1
hcond2D_CLD_YY_ZZ_05	0	5	Note 1
hcond2D_CLD_YY_ZZ_07	0	7	Note 1
hcond2D_CLD_YY_ZZ_09	0	9	Note 1
hcond2D_ICC_YY_ZZ_01	0	1	Note 1
hcond2D_ICC_YY_ZZ_03	0	3	Note 1
hcond2D_ICC_YY_ZZ_05	0	5	Note 1
hcond2D_ICC_YY_ZZ_07	0	7	Note 1
hcond2D_CPC_YY_ZZ_03	0	3	Note 1
hcond2D_CPC_YY_ZZ_06	0	6	Note 1
hcond2D_CPC_YY_ZZ_09	0	9	Note 1
hcond2D_CPC_YY_ZZ_12	0	12	Note 1
hcod1D_ICC_Diff	0	7	Note 1
hcodLavIdx	0	N/A	
hcod2D_EnvRes	0	N/A	
Note 1: Data can only have non-negative values for this table.			

by:

Table name	**Offset**	**LAV**	**Notes**
hcodFirstBand_CLD	15	15	Note 2
hcodFirstBand_ICC	0	7	Note 1
hcodFirstBand_CPC	10	15	
hcod1D_CLD_YY	0	15	Notes 1, 2
hcod1D_ICC_YY	0	7	Note 1
hcod1D_CPC_YY	0	15	Note 1
hcond2D_CLD_YY_ZZ_LL_escape	N/A	N/A	Note 2
hcond2D_ICC_YY_ZZ_LL_escape	N/A	N/A	
hcond2D_CPC_YY_ZZ_LL_escape	N/A	N/A	
hcond2D_CLD_YY_ZZ_03	0	3	Notes 1, 2
hcond2D_CLD_YY_ZZ_05	0	5	Notes 1, 2
hcond2D_CLD_YY_ZZ_07	0	7	Notes 1, 2
hcond2D_CLD_YY_ZZ_09	0	9	Notes 1, 2
hcond2D_ICC_YY_ZZ_01	0	1	Note 1
hcond2D_ICC_YY_ZZ_03	0	3	Note 1
hcond2D_ICC_YY_ZZ_05	0	5	Note 1
hcond2D_ICC_YY_ZZ_07	0	7	Note 1
hcond2D_CPC_YY_ZZ_03	0	3	Note 1
hcond2D_CPC_YY_ZZ_06	0	6	Note 1
hcond2D_CPC_YY_ZZ_09	0	9	Note 1
hcond2D_CPC_YY_ZZ_12	0	12	Note 1
hcod1D_ICC_Diff	0	7	Note 1
hcodLavIdx	0	N/A	
hcod2D_EnvRes	0	N/A	
Note 1: Data can only have non-negative values for this table. Note 2: The tables for data type CLD apply also for data type ATD.			

In Annex A.2, replace incorrect entries in Table 32:

	0	1	2	3	4	5	6	7	8	9	10
$q(k)$	0	0	0	0	0	0	1	1	2	2	3
k	11	12	13	14	15	16	17	18	19	20	21
$q(k)$	3	5	6	7	8	9	10	11	12	13	14
k	22	23	24	25	26	27	28	28	30	31	32
$q(k)$	15	16	17	18	19	20	21	22	23	24	25
k	33	34	35	36	37	38	39	40	41	42	43
$q(k)$	26	27	28	29	30	31	32	33	34	35	36
k	44	45	46	47	48	49	50	51	52	53	54
$q(k)$	37	38	39	40	41	42	43	44	45	46	47
k	55	56	57	58	58	60	61	62	63	64	65
$q(k)$	48	49	50	51	52	53	54	55	56	57	58
k	66	67	68	69	70						
$q(k)$	59	60	61	62	63						

by:

k	0	1	2	3	4	5	6	7	8	9	10
$q(k)$	0	0	0	0	0	0	1	1	2	2	3
k	11	12	13	14	15	16	17	18	19	20	21
$q(k)$	4	5	6	7	8	9	10	11	12	13	14
k	22	23	24	25	26	27	28	29	30	31	32
$q(k)$	15	16	17	18	19	20	21	22	23	24	25
k	33	34	35	36	37	38	39	40	41	42	43
$q(k)$	26	27	28	29	30	31	32	33	34	35	36
k	44	45	46	47	48	49	50	51	52	53	54
$q(k)$	37	38	39	40	41	42	43	44	45	46	47
k	55	56	57	58	59	60	61	62	63	64	65
$q(k)$	48	49	50	51	52	53	54	55	56	57	58
k	66	67	68	69	70						
$q(k)$	59	60	61	62	63						

In Annex F.1, in the third paragraph, replace:

The 5151 systems combines

by:

The 5151 system combines

In subclause F.5.1, replace:

In the stereo based system, the last module of the encoder performs a down-mix from three (x_1, x_2, x_3) to two down-mix channels, combined with generation of corresponding spatial parameters. Each down-mix channel is a linear combination of the input signals x_1, x_2, and x_3. The energy-based mode transmits CLD parameter sets with encoder parameters

by:

In the stereo based system, the last module of the encoder performs a down-mix from three (x_1, x_2, x_3) to two down-mix channels, combined with generation of corresponding spatial parameters. Each down-mix channel is a linear combination of the input signals x_1, x_2, and x_3. Only the down-mix channels and the corresponding spatial parameters are transmitted to the decoder. Effectively this means that one channel is discarded prior to transmission. The TTT module supports two methods to retrieve three output signals (l, r, c) given the two down-mix channels and corresponding spatial parameters.

The energy-based mode transmits CLD parameter sets with encoder parameters

In subclause F.5.2.2.1, replace:

A second mode of operation for the TTT element is based on transmission of (elements of the) up-mix matrix directly. Furthermore, the TTT prediction mode describes an up-mix matrix that does allow cross-talk (i.e., non-zero values of m_{21} and m_{12}) in the up-mix process. The up-mix matrix, M_{CPC} is given by:

$$\begin{bmatrix} l \\ r \\ c \end{bmatrix} = M_{CPC} \begin{bmatrix} l_0 \\ r_0 \end{bmatrix},$$

with

$$M_{CPC} = \frac{1}{3} \begin{bmatrix} c_1 + 2 & c_2 - 1 \\ c_1 - 1 & c_2 + 2 \\ 1 - c_1 & 1 - c_2 \end{bmatrix}.$$

Here, c_1 and c_2 represent the transmitted CPC parameters. As can be observed, the output signal triplet (l, r, c) consists of linear combinations of the down-mix signal pair (l_0, r_0), in which the CPC parameters describe the up-mix matrix M_{CPC}.

by the following text, which also includes a new Figure F.1A:

Figure F.1A illustrates the combined encoder and decoder processing for the TTT module in prediction mode. A third channel $\hat{c}_0$ is generated in the encoder by means of the fixed coefficient invertible down-mix matrix D:

$$\begin{bmatrix} l_0 \\ r_0 \\ \hat{c}_0 \end{bmatrix} = D \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1}{2}\sqrt{2} \\ 0 & 1 & \frac{1}{2}\sqrt{2} \\ 1 & 1 & -\frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The third channel $\hat{c}_0$ is discarded in the encoder, whereas the down-mix channels (l_o, r_o) are transmitted to the decoder together with corresponding CPC prediction parameters c_1 and c_2 that describe the estimation of a third channel c_0 from two down-mix channels (l_0, r_0) conforming to

$$c_0 = c_1 l_0 + c_2 r_0 .$$

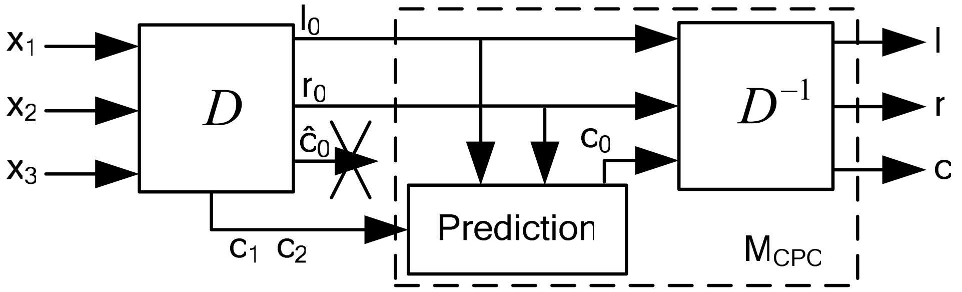


Figure F.1A – Combined encoder and decoder processing TTT module

Together with the down-mix channels (l_0, r_0), c_0 is used to reconstruct the output channels (l, r, c) by means of the inverse of down-mix matrix D as was used in the encoder:

$$\begin{bmatrix} l \\ r \\ c \end{bmatrix} = D^{-1} \begin{bmatrix} l_0 \\ r_0 \\ c_0 \end{bmatrix},$$

where

$$D^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} \end{bmatrix}.$$

Inserting the expression for c_0 results in

$$\begin{bmatrix} l \\ r \\ c \end{bmatrix} = D^{-1}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ c_1 & c_2 \end{bmatrix}\begin{bmatrix} l_0 \\ r_0 \end{bmatrix} = M_{CPC}\begin{bmatrix} l_0 \\ r_0 \end{bmatrix},$$

where the up-mix matrix, M_{CPC} is given by:

$$M_{CPC} = \frac{1}{3}\begin{bmatrix} c_1 + 2 & c_2 - 1 \\ c_1 - 1 & c_2 + 2 \\ (1 - c_1)\sqrt{2} & (1 - c_2)\sqrt{2} \end{bmatrix}.$$

The up-mix matrix is an integral part of the R1 matrix that is used to derive the pre-matrix M1 (see subclause 6.5).

In subclause F.8.3, in the first paragraph, replace:

to calculate a matrixed-surround compatible downmix.

by:

to calculate a matrix-surround compatible downmix.

In Annex G.1 and G.3, replace:

bit stream

by:

bitstream

In Annex H.3.2, in the third paragraph, replace:

IsCenterLFE() : a function to check if current R-OTT box is related to Centre/LFE outputs. The ‘0’ means the outputs of current R-OTT box are Centre/LFE and the ‘1’ means the outputs of current R-OTT box is Centre/LFE.

by:

IsCenterLFE() : a function to check if current R-OTT box is related to Centre/LFE outputs. The ‘0’ means the outputs of current R-OTT box are not Centre/LFE and the ‘1’ means the outputs of current R-OTT box are Centre/LFE.