



**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*.

*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:

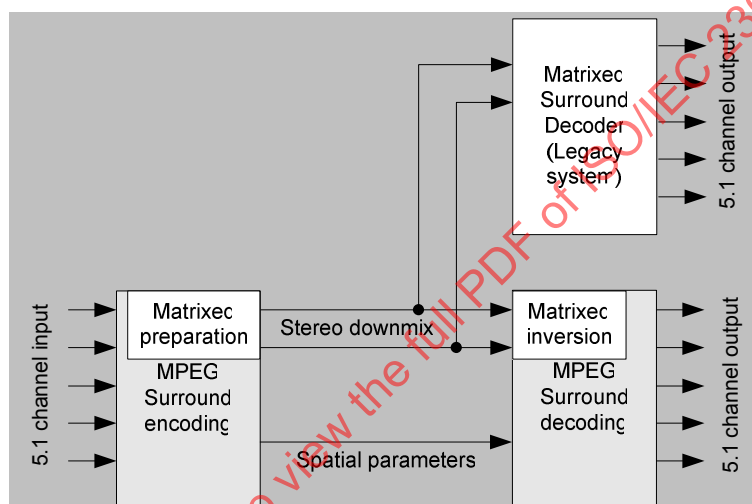


Figure 1 — Matrixed compatible MPEG Surround

by:

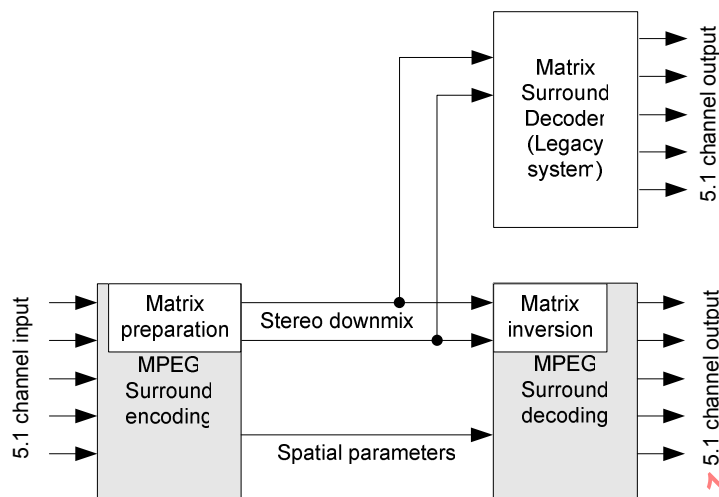


Figure 2 — Matrix compatible MPEG Surround

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

by:

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

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                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------|
| <pre> ResidualData() {   for (i=0; i&lt;numOttBoxes+numTttBoxes; i++) {     if (bsResidualPresent[i]) {       if (i&lt;numOttBoxes) {         for (ps=0; ps&lt;numParamSets; ps++) {           <b>bslccDiffPresent</b>[i][ps];           if (bslccDiffPresent[i][ps]) {             for (pb=0; pb&lt;bsResidualBands[i]; pb++) {               lccDiff[i][ps][pb] =                 1Dhuff_dec(hcod1D_ICC_Diff,<b>bsCodeW</b>);             }           }         }       }       tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1);       for (rf=0; rf&lt;bsResidualFramesPerSpatialFrame; rf++)         individual_channel_stream(0);       if (window_sequence == EIGHT_SHORT_SEQUENCE) &amp;&amp;         ((tempExtraFrame == 18)    (tempExtraFrame == 24)             (tempExtraFrame == 30)) {         individual_channel_stream(0);       }     }   } } </pre> | <p>1</p> <p>1..7</p> | <p>Note 2<br/><b>Uimsbf</b></p> <p>Note 3</p> <p>Note 4</p> <p>Note 1<br/>Note 5</p> <p>Note 1</p> |
| <p>Note 1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.</p> <p>Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.</p> <p>Note 3: 1Dhuff_dec() is defined in Annex A.1.</p> <p>Note 4: numSlots is defined by numSlots = bsFrameLength + 1. Furthermore the division shall be interpreted as ANSI C integer division.</p> <p>Note 5: individual_channel_stream(0) determines the value of window_sequence.</p>                                                                                                                                                                                                                                                                                                                                                   |                      |                                                                                                    |

by:

**Table 34 — Syntax of ResidualData()**

| Syntax                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | No. of bits                        | Mnemonic                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <pre> ResidualData() {     for (k=0; k&lt;numOttBoxes+numTttBoxes; k++) {         if (bsTreeConfig==3    bsTreeConfig==4) {             i = resReorder[k];         }         else             i = k;         if (bsResidualPresent[i]) {             if (i&lt;numOttBoxes) {                 for (ps=0; ps&lt;numParamSets; ps++) {                     <b>bslccDiffPresent[i][ps];</b>                     if (bslccDiffPresent[i][ps]) {                         for (pb=0; pb&lt;bsResidualBands[i]; pb++) {                             lccDiff[i][ps][pb] =                                 1Dhuff_dec(hcod1D_ICC_Diff,<b>bsCodeW</b>);                         }                     }                 }             }             tempExtraFrame=numSlots/(bsResidualFramesPerSpatialFrame+1);             for (rf=0; rf&lt;bsResidualFramesPerSpatialFrame; rf++)                 individual_channel_stream(0);             if (window_sequence == EIGHT_SHORT_SEQUENCE) &amp;&amp;                 ((tempExtraFrame == 18)    (tempExtraFrame == 24)                     (tempExtraFrame == 30)) {                 individual_channel_stream(0);             }         }     } } </pre> | <p><b>1</b></p> <p><b>1..7</b></p> | <p>Note 6</p> <p>Note 2<br/><b>Uimsbf</b></p> <p><b>vlclbf</b><br/>Note 3</p> <p>Note 4</p> <p>Note 1</p> <p>Note 5</p> <p>Note 1</p> |
| <p>Note 1: individual_channel_stream(0) according to MPEG-2 AAC Low Complexity profile bitstream syntax described in subclause 6.3 of ISO/IEC 13818-7.</p> <p>Note 2: numParamSets is defined by numParamSets = bsNumParamSets + 1.</p> <p>Note 3: 1Dhuff_dec() is defined in Annex A.1.</p> <p>Note 4: numSlots is defined by numSlots = bsFrameLength + 1. Furthermore the division shall be interpreted as ANSI C integer division.</p> <p>Note 5: individual_channel_stream(0) determines the value of window_sequence.</p> <p>Note 6: resReorder[k] for 0 ≤ k &lt; 6 is the following constant integer array: {0, 1, 2, 5, 3, 4}.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                    |                                                                                                                                       |

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  $\text{bsTttBandsLow} \leq \text{pb} < \text{bsTttBandsHigh}$ .

by:

**bsTttBandsHigh** Helper variable, same as bsTttBandsLow but for high band range. The high band range is  $\text{bsTttBandsLow} \leq \text{pb} < \text{bsTttBandsHigh}$ .

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<br>[LavIdx] | lavTabICC<br>[LavIdx] | lavTabCPC<br>[LavIdx] |
|--------|-----------------------|-----------------------|-----------------------|
| 0      | 3                     | 1                     | 3                     |
| 1      | 5                     | 3                     | 6                     |
| 2      | 7                     | 5                     | 9                     |
| 3      | 9                     | 7                     | 12                    |

by:

| LavIdx | lavTabCLD<br>[LavIdx],<br>lavTabATD<br>[LavIdx] | lavTabICC<br>[LavIdx] | lavTabCPC<br>[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 ( $0 \leq ps < \text{numParamSets}$ ) and the pb:th parameter band ( $0 \leq pb < \text{numBands}$ ). 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 pi:th XXX parameter for the ps:th parameter set ( $0 \leq ps < \text{numParamSets}$ ) and the pb:th parameter band ( $0 \leq pb < \text{numBands}$ ). 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        |       |
| numAacEI  | AacEI | numAacEI | AacEI | numAacEI | AacEI |
| 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 (!ottModelLfe[pi]) {
            firstPiNonLfe = pi;
            break;
        }
    }
    if (firstPiNonLfe != -1) {
        for (pi=0; pi<numOttBoxes; pi++) {
            if (!ottModelLfe[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 (bsPilotCoding[xxx][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:

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

by:

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

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:

| <b>numSlots</b> | <i>bsResidualFramesPerSpatialFrame</i><br><b>or</b><br><i>bsArbitraryDownmixResidualFramesPerSpatialFrame</i> |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| 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 | <i>bsResidualFramesPerSpatialFrame</i><br>or<br><i>bsArbitraryDownmixResidualFramesPerSpatialFrame</i> |
|----------|--------------------------------------------------------------------------------------------------------|
| 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 ( $CLD_{dm}$ )

by:

and an Interchannel Cross Correlation ( $ICC_{dm}$ )

In subclause 6.2.2, second paragraph, replace:

stereo hybrid QMF-domain input signals ( $x_X^{n,k}, x_i^{n,k}$ ) following

by:

stereo hybrid QMF-domain input signals ( $x_{L_0}^{n,k}, x_{R_0}^{n,k}$ ) 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

$$Q_X^n = \sum_{k=16}^{42} x_X^{n,k} (x_X^{n,k})^*, \quad X \in \{L_0, R_0\}$$

From  $Q_X^n$ , two analysis state parameters  $r_{X,1}^n$  and  $r_{X,2}^n$  for  $X \in \{L_0, R_0\}$  are subsequently updated according to:

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

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

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:

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

by:

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

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

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

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

In subclause 6.2.4, replace:

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

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

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

by:

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

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

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

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  $\mathbf{dm}_{CLD}(m) < 0$ ;
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 2<sup>nd</sup> 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.

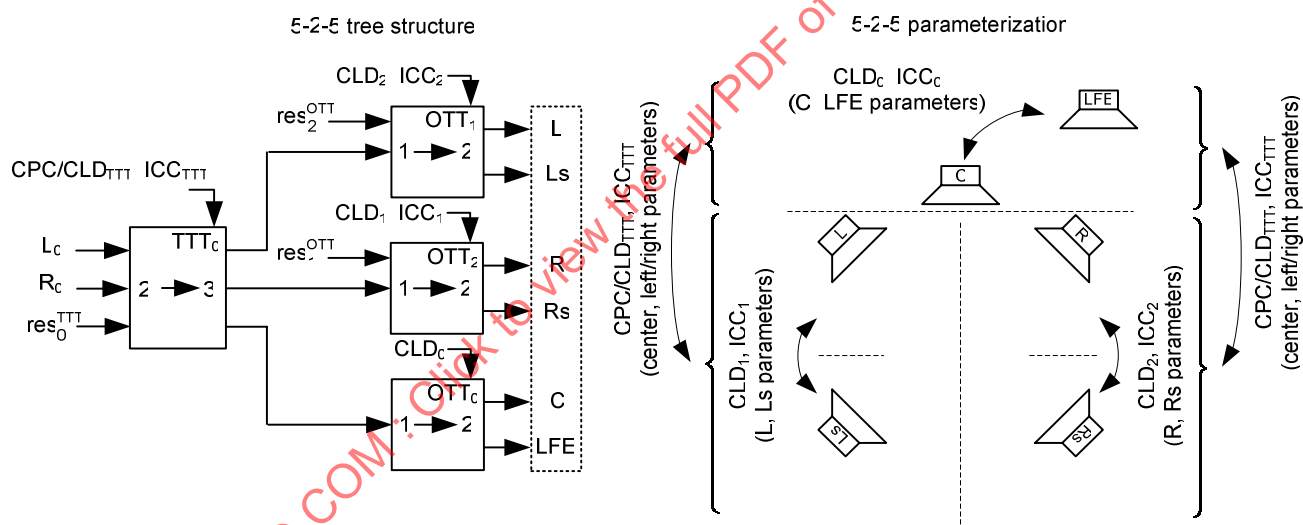
In subclause 6.4.1, in the 2<sup>nd</sup> paragraph replace:

by means of **signalling** in the data stream **by** activated.

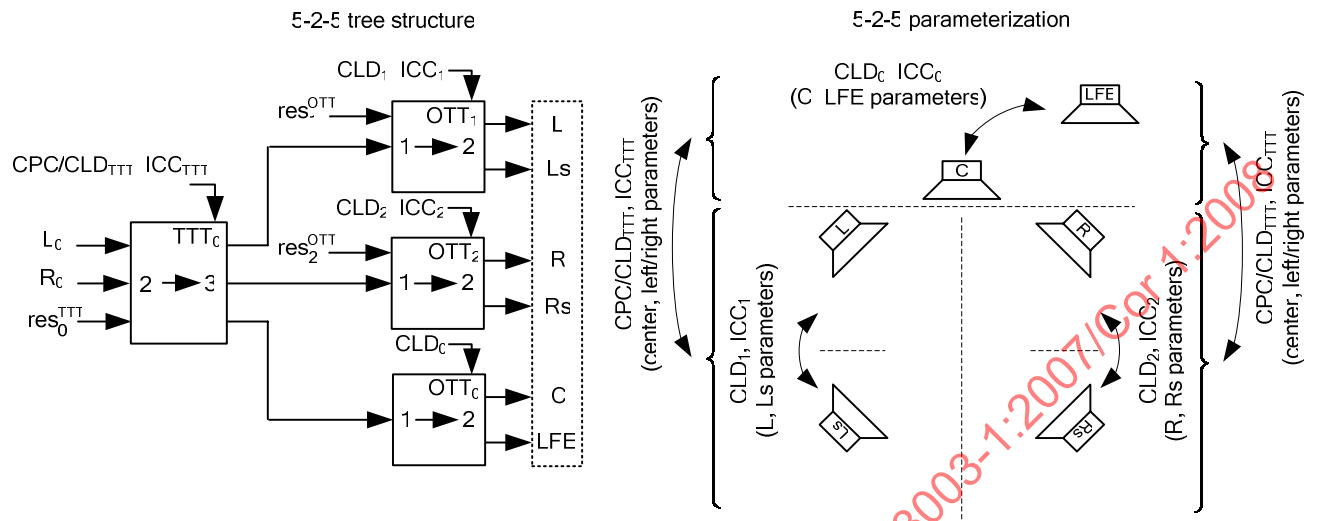
by:

by means of **signaling** in the data stream **be** activated.

In subclause 6.4.3.1, replace Figure 26:



by:



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  $\mathbf{v}_m^{n,k}$ .

by:

with the exception of  $\mathbf{v}_{M_0}^{n,k}$ .

In subclause 6.4.7.3, in the last sentence, replace:

given the  $CLD_q^{l,m}$  and  $ICC_q^{l,m}$  parameters calculated in the previous subclauses

by:

given the  $CLD_q^{l,m}$ ,  $ICC_q^{l,m}$  and  $\mathbf{g}^{l,m}$  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<sub>1</sub> or 7-2-7<sub>2</sub> 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<sub>3</sub> and OTT<sub>4</sub> 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<sub>3</sub> and OTT<sub>4</sub> 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

| <b>bsTreeConfig</b>                                                    | Mapped 5-2-5<br><b>bsTempShapeEnableChannel(ch)</b> | Combination of original 7-2-7<br><b>bsTempShapeEnableChannel(ch')</b> |
|------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------|
| 3 (7-2-7 <sub>1</sub> )                                                | L                                                   | L' AND Lc'                                                            |
|                                                                        | R                                                   | R' AND Rc'                                                            |
| 4 (7-2-7 <sub>2</sub> )                                                | 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.

$$XXX[ch][n] = \text{envReshapeData}[ch_1][n] + \text{envReshapeData}[ch_2][n]$$

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

where SIGN() denoted the signum function.

**Table 88B — Mapping of GES envReshapeData**

| <b>bsTreeConfig</b>                                                    | <b>ch</b> | <b>ch<sub>1</sub>;ch<sub>2</sub></b> |
|------------------------------------------------------------------------|-----------|--------------------------------------|
| 3 (7-2-7 <sub>1</sub> )                                                | L         | L';Lc'                               |
|                                                                        | R         | R';Rc'                               |
| 4 (7-2-7 <sub>2</sub> )                                                | 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:

$$\beta^m$$

by:

$$\beta^{l,m}$$

In subclause 6.5.2.2.3.3, replace:

$$\text{bsTttModeLow}(0) = 3$$

by:

$$\text{bsTttModeLow}(0) \in \{2, 3\}$$

In subclause 6.5.2.2.3.3, replace:

$$\text{bsTttModeLow}(0) = 5$$

by:

$$\text{bsTttModeLow}(0) \in \{4, 5\}$$

In subclause 6.5.2.2.3.3 replace:

$$\text{For } \text{bsTttModeLow}(0) \geq 3$$

by:

$$\text{For } \text{bsTttModeLow}(0) \geq 2$$

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<sub>2</sub> configuration the  $\mathbf{R}_2^{l,m}$  matrix is defined according to:

$$\mathbf{R}_2^{l,m} = \begin{bmatrix} \kappa_{TT0}^{l,m} H1 I_{OTT1}^{l,m} & 0 & 0 & H1 2_{OTT1}^{l,m} & 0 & g_{TT0}^{l,m} c_{1,OTT1}^{l,m} & 0 & 0 \\ \kappa_{TT0}^{l,m} H1 I_{OTT1}^{l,m} H1 I_{OTT3}^{l,m} & 0 & 0 & H1 2_{OTT1}^{l,m} H1 I_{OTT3}^{l,m} & 0 & g_{TT0}^{l,m} c_{1,OTT1}^{l,m} H1 I_{OTT3}^{l,m} & H1 2_{OTT3}^{l,m} & 0 \\ \kappa_{TT0}^{l,m} H1 I_{OTT1}^{l,m} H2 I_{OTT3}^{l,m} & 0 & 0 & H1 2_{OTT1}^{l,m} H2 I_{OTT3}^{l,m} & 0 & g_{TT0}^{l,m} c_{1,OTT1}^{l,m} H2 I_{OTT3}^{l,m} & H2 2_{OTT3}^{l,m} & 0 \\ 0 & \kappa_{TT0}^{l,m} H1 I_{OTT2}^{l,m} & 0 & 0 & H1 2_{OTT2}^{l,m} & g_{TT0}^{l,m} c_{1,OTT2}^{l,m} & 0 & 0 \\ 0 & \kappa_{TT0}^{l,m} H1 I_{OTT2}^{l,m} H1 I_{OTT4}^{l,m} & 0 & 0 & H1 2_{OTT2}^{l,m} H1 I_{OTT4}^{l,m} & g_{TT0}^{l,m} c_{1,OTT2}^{l,m} H1 I_{OTT4}^{l,m} & 0 & H1 2_{OTT4}^{l,m} \\ 0 & \kappa_{TT0}^{l,m} H1 I_{OTT2}^{l,m} H2 I_{OTT4}^{l,m} & 0 & 0 & H1 2_{OTT2}^{l,m} H2 I_{OTT4}^{l,m} & g_{TT0}^{l,m} c_{1,OTT2}^{l,m} H2 I_{OTT4}^{l,m} & 0 & H2 2_{OTT4}^{l,m} \\ 0 & 0 & \kappa_{TT0}^{l,m} c_{1,OTT0}^{l,m} & 0 & 0 & -\sqrt{2} g_{TT0}^{l,m} c_{1,OTT0}^{l,m} & 0 & 0 \\ 0 & 0 & c_{2,OTT0}^{l,m} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

by:

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

$$\mathbf{R}_2^{l,m} = \begin{bmatrix} \kappa_{TTT_0}^{l,m} H1I_{OTT_1}^{l,m} & 0 & 0 & H12_{OTT_1}^{l,m} & 0 & g_{TTT_0}^{l,m} c_{1,OTT_1}^{l,m} & 0 & 0 \\ H2I_{OTT_1}^{l,m} H1I_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_1}^{l,m} H1I_{OTT_3}^{l,m} & 0 & 0 & H12_{OTT_3}^{l,m} & 0 \\ H2I_{OTT_1}^{l,m} H2I_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_1}^{l,m} H2I_{OTT_3}^{l,m} & 0 & 0 & H22_{OTT_3}^{l,m} & 0 \\ 0 & \kappa_{TTT_0}^{l,m} H1I_{OTT_2}^{l,m} & 0 & 0 & H12_{OTT_2}^{l,m} & g_{TTT_0}^{l,m} c_{1,OTT_2}^{l,m} & 0 & 0 \\ 0 & H2I_{OTT_2}^{l,m} H1I_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_2}^{l,m} H1I_{OTT_4}^{l,m} & 0 & 0 & H12_{OTT_4}^{l,m} \\ 0 & H2I_{OTT_2}^{l,m} H2I_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_2}^{l,m} H2I_{OTT_4}^{l,m} & 0 & 0 & H22_{OTT_4}^{l,m} \\ 0 & 0 & \kappa_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 & -\sqrt{2} g_{TTT_0}^{l,m} c_{1,OTT_0}^{l,m} & 0 & 0 \\ 0 & 0 & c_{2,OTT_0}^{l,m} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In subclause 6.6.1, replace Table 89, below:

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

| configuration      | Decorrelator $X = 0, \dots, 9$ |                        |                        |                        |                        |   |   |   |   |   |
|--------------------|--------------------------------|------------------------|------------------------|------------------------|------------------------|---|---|---|---|---|
|                    | 0                              | 1                      | 2                      | 3                      | 4                      | 5 | 6 | 7 | 8 | 9 |
| 5-1-5 <sub>1</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_2^{\text{OTT}} ( )$ |                        |   |   |   |   |   |
| 5-1-5 <sub>2</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |                        |   |   |   |   |   |
| 5-2-5              | $D_2^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_0^{\text{TTT}} ( )$ |                        |                        |   |   |   |   |   |
| 7-2-7 <sub>1</sub> | $D_2^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_0^{\text{TTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |   |   |   |   |   |
| 7-2-7 <sub>2</sub> | $D_2^{\text{OTT}} ( )$         | $D_0^{\text{TTT}} ( )$ | $D_0^{\text{TTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |   |   |   |   |   |
| 7-5-7 <sub>1</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ |                        |                        |                        |   |   |   |   |   |
| 7-5-7 <sub>2</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ |                        |                        |                        |   |   |   |   |   |

by:

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

|                    | Decorrelator $X = 0, \dots, 9$ |                        |                        |                        |                        |   |   |   |   |   |
|--------------------|--------------------------------|------------------------|------------------------|------------------------|------------------------|---|---|---|---|---|
| configuration      | 0                              | 1                      | 2                      | 3                      | 4                      | 5 | 6 | 7 | 8 | 9 |
| 5-1-5 <sub>1</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_2^{\text{OTT}} ( )$ |                        |   |   |   |   |   |
| 5-1-5 <sub>2</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |                        |   |   |   |   |   |
| 5-2-5              | $D_1^{\text{OTT}} ( )$         | $D_2^{\text{OTT}} ( )$ | $D_0^{\text{ITT}} ( )$ |                        |                        |   |   |   |   |   |
| 7-2-7 <sub>1</sub> | $D_1^{\text{OTT}} ( )$         | $D_2^{\text{OTT}} ( )$ | $D_0^{\text{ITT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |   |   |   |   |   |
| 7-2-7 <sub>2</sub> | $D_1^{\text{OTT}} ( )$         | $D_2^{\text{OTT}} ( )$ | $D_0^{\text{ITT}} ( )$ | $D_3^{\text{OTT}} ( )$ | $D_4^{\text{OTT}} ( )$ |   |   |   |   |   |
| 7-5-7 <sub>1</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ |                        |                        |                        |   |   |   |   |   |
| 7-5-7 <sub>2</sub> | $D_0^{\text{OTT}} ( )$         | $D_1^{\text{OTT}} ( )$ |                        |                        |                        |   |   |   |   |   |

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:

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

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

by:

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

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

In subclause 6.7.3, in the second paragraph, replace:

$$0 \leq n < \text{numSlots} \quad 6 \leq sb < 24$$

by:

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

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  $Env_{diffuse\_hold,ch} = \frac{1}{\varepsilon}$ .

In subclause 6.7.5, in the first paragraph, replace:

For 5-1-5

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

For 5-2-5, 7-2-7<sub>1</sub>, 7-2-7<sub>2</sub>, 7-5-7<sub>1</sub> and 7-5-7<sub>2</sub>

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

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

by:

For 5-1-5

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

For 5-2-5, 7-2-7<sub>1</sub>, 7-2-7<sub>2</sub>, 7-5-7<sub>1</sub> and 7-5-7<sub>2</sub>

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

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

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:

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

In subclause 6.8.2, replace:

$$Env_{ZZZ, ch_{ZZZ}}(n) = \begin{cases} \frac{Env_{input, L_0}(n) + Env_{input, R_0}(n)}{2} & \begin{cases} ZZZ = input \\ ch_{input} = C_0 \end{cases} \\ \sqrt{\frac{EnvAbs_{ZZZ, ch_{ZZZ}}^n}{Env_{ZZZ, ch_{ZZZ}}(n)}} & , otherwise \end{cases}$$

by:

$$Env_{ZZZ, ch_{ZZZ}}(n) = \begin{cases} \frac{Env_{input, L_0}(n) + Env_{input, R_0}(n)}{2} & \begin{cases} ZZZ = input \\ ch_{input} = C_0 \end{cases} \\ \sqrt{\frac{EnvAbs_{ZZZ, ch_{ZZZ}}^n}{Env_{ZZZ, ch_{ZZZ}}(n) + \varepsilon}} & , otherwise \end{cases}$$

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:

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

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

by:

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

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

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:

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

where

$\alpha = \exp\left(-\frac{dt}{\tau}\right)$  and  $dt = \frac{64}{F_s}$  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:

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

The aliasing condition is set according to the following rule

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

by:

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

where  $\alpha = \exp\left(-\frac{dt}{\tau}\right)$  and  $dt = \frac{64}{F_s}$  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:

$$w_{\text{real}}^{n,m} = \sum_{ch} \hat{x}_{\text{real}}^{n-1,m}(ch)$$

The aliasing condition is set according to the following rule

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

In subclause 6.10.4.2, replace:

$$\mathbf{M}_1^{n,k} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{t}(l-1) \leq n < \mathbf{t}(l), 0 \leq l < L, 0 \leq k < K.$$

by:

$$\mathbf{M}_1^{n,k} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \mathbf{G}_{LP,1}^{l,m}, \mathbf{t}(l-1) \leq n < \mathbf{t}(l), 0 \leq l < L, 0 \leq k < K,$$

where

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

In subclause 6.10.4.3, replace:

Where

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

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

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

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

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

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

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

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

| <i>Col</i>    |                  | 1 | 2 | 3 | 4 |
|---------------|------------------|---|---|---|---|
| $resIdx(col)$ | 515 <sub>1</sub> | 0 | 1 | 3 | 2 |
|               | 515 <sub>2</sub> | 0 | 1 | 3 | 4 |

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

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

In subclause 6.10.5, replace:

For  $m < M_{\text{proc}}^c$ , 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  $m \geq M_{\text{proc}}^c$ , 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  $m < M_{\text{proc}}^c - 1$ , 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  $m \geq M_{\text{proc}}^c - 1$ , 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:

$$IPD_B^{l,m} = \begin{cases} 0 & , m > 12, ICC_{\text{Btmp}}^{l,m} < 0.6 \\ \arg\left(\langle L_B R_B^* \rangle^{l,m}\right) & , \text{otherwise} \end{cases}$$

where

$$ICC_{\text{Btmp}}^{l,m} = \frac{|\langle L_B R_B^* \rangle^{l,m}|}{\sigma_L^{l,m} \sigma_R^{l,m}}$$

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

by:

$$IPD_B^{l,m} = \begin{cases} 0 & , m \geq 12, ICC_{\text{Btmp}}^{l,m} < 0.6 \\ \arg\left(\langle L_B R_B^* \rangle^{l,m}\right) & , \text{otherwise} \end{cases}$$

where

$$ICC_{\text{Btmp}}^{l,m} = \frac{|\langle L_B R_B^* \rangle^{l,m}|}{\sigma_L^{l,m} \sigma_R^{l,m}}$$

and  $\Re(x)$  denotes the real value of  $x$ . For high-frequency processing bands (band  $m = 12$  onwards), the  $IPD_B$  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  $(\sigma_X^{l,m})^2$  for  $X \in \{L, R\}$ , 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  $(\sigma_X^{l,m})^2$  for each virtual loudspeaker  $X \in \{L, R, C, Ls, Rs\}$ , according to:

$$(\sigma_L^{l,m})^2 = r_1(CLD_0^{l,m})r_1(CLD_1^{l,m})r_1(CLD_3^{l,m})$$

$$(\sigma_R^{l,m})^2 = r_1(CLD_0^{l,m})r_1(CLD_1^{l,m})r_2(CLD_3^{l,m})$$

$$(\sigma_C^{l,m})^2 = r_1(CLD_0^{l,m})r_2(CLD_1^{l,m})$$

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

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

with

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

and

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

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

and where

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

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

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

$$\begin{aligned} (\sigma_L^{l,m})^2 = & (P_{L,C}^m)^2 (\sigma_C^{l,m})^2 + (P_{L,L}^m)^2 (\sigma_L^{l,m})^2 + (P_{L,Ls}^m)^2 (\sigma_{Ls}^{l,m})^2 + (P_{L,R}^m)^2 (\sigma_R^{l,m})^2 + (P_{L,Rs}^m)^2 (\sigma_{Rs}^{l,m})^2 + \dots \\ & 2P_{L,L}^m P_{L,R}^m \rho_L^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_3^{l,m} \cos(\phi_R^m) + \dots \\ & 2P_{L,Ls}^m P_{L,Rs}^m \rho_{Ls}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_2^{l,m} \cos(\phi_{Rs}^m) \end{aligned}$$

$$\begin{aligned} (\sigma_R^{l,m})^2 = & (P_{R,C}^m)^2 (\sigma_C^{l,m})^2 + (P_{R,L}^m)^2 (\sigma_L^{l,m})^2 + (P_{R,Ls}^m)^2 (\sigma_{Ls}^{l,m})^2 + (P_{R,R}^m)^2 (\sigma_R^{l,m})^2 + (P_{R,Rs}^m)^2 (\sigma_{Rs}^{l,m})^2 + \dots \\ & 2P_{R,L}^m P_{R,R}^m \rho_L^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_3^{l,m} \cos(\phi_L^m) + \dots \\ & 2P_{R,Ls}^m P_{R,Rs}^m \rho_{Ls}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_2^{l,m} \cos(\phi_{Ls}^m) \end{aligned}$$

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

and where

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

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

The complex cross-spectrum  $\langle L_B R_B^* \rangle^{l,m}$  for each spatial parameter set is estimated for each parameter position and processing band as:

$$\begin{aligned} \langle L_B R_B^* \rangle = & (\sigma_C^{l,m})^2 P_{L,C}^m P_{R,C}^m \rho_C^m \exp(j\phi_C^m) + \dots \\ & (\sigma_L^{l,m})^2 P_{L,L}^m P_{R,L}^m \rho_L^m \exp(j\phi_L^m) + \dots \\ & (\sigma_R^{l,m})^2 P_{L,R}^m P_{R,R}^m \rho_R^m \exp(j\phi_R^m) + \dots \\ & (\sigma_{Ls}^{l,m})^2 P_{L,Ls}^m P_{R,Ls}^m \rho_{Ls}^m \exp(j\phi_{Ls}^m) + \dots \\ & (\sigma_{Rs}^{l,m})^2 P_{L,Rs}^m P_{R,Rs}^m \rho_{Rs}^m \exp(j\phi_{Rs}^m) + \dots \\ & P_{L,L}^m P_{R,R}^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_3 + \dots \\ & P_{L,Ls}^m P_{R,Rs}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_2 + \dots \\ & P_{L,R}^m P_{R,L}^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_3 \rho_L^m \rho_R^m \exp(j(\phi_L^m + \phi_R^m)) + \dots \\ & P_{L,Rs}^m P_{R,Ls}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_2 \rho_{Ls}^m \rho_{Rs}^m \exp(j(\phi_{Ls}^m + \phi_{Rs}^m)) \end{aligned}$$

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

and where

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

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

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  $(\sigma_X^{l,m})^2$  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  $(\sigma_X^{l,m})^2$  for each virtual loudspeaker  $X = \{L, R, C, Ls, Rs\}$ , according to:

$$(\sigma_L^{l,m})^2 = r_1(CLD_0^{l,m})r_1(CLD_1^{l,m})r_1(CLD_3^{l,m})$$

$$(\sigma_R^{l,m})^2 = r_1(CLD_0^{l,m})r_2(CLD_1^{l,m})r_1(CLD_4^{l,m})$$

$$(\sigma_C^{l,m})^2 = r_2(CLD_0^{l,m})$$

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

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

with

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

and

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

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

and where

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

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

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

$$\begin{aligned} (\sigma_L^{l,m})^2 = & (P_{L,C}^m)^2 (\sigma_C^{l,m})^2 + (P_{L,L}^m)^2 (\sigma_L^{l,m})^2 + (P_{L,Ls}^m)^2 (\sigma_{Ls}^{l,m})^2 + (P_{L,R}^m)^2 (\sigma_R^{l,m})^2 + (P_{L,Rs}^m)^2 (\sigma_{Rs}^{l,m})^2 + \dots \\ & 2P_{L,L}^m P_{L,R}^m \rho_R^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_1^{l,m} \cos(\phi_R^m) + \dots \\ & 2P_{L,Ls}^m P_{L,Rs}^m \rho_{Rs}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_1^{l,m} \cos(\phi_{Rs}^m) \end{aligned}$$

$$\begin{aligned} (\sigma_R^{l,m})^2 = & (P_{R,C}^m)^2 (\sigma_C^{l,m})^2 + (P_{R,L}^m)^2 (\sigma_L^{l,m})^2 + (P_{R,Ls}^m)^2 (\sigma_{Ls}^{l,m})^2 + (P_{R,R}^m)^2 (\sigma_R^{l,m})^2 + (P_{R,Rs}^m)^2 (\sigma_{Rs}^{l,m})^2 + \dots \\ & 2P_{R,L}^m P_{R,R}^m \rho_L^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_1^{l,m} \cos(\phi_L^m) + \dots \\ & 2P_{R,Ls}^m P_{R,Rs}^m \rho_{Ls}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_1^{l,m} \cos(\phi_{Ls}^m) \end{aligned}$$

for  $0 \leq m < M_{\text{Proc}}, 0 \leq l < L$ ,

and where

$$ICC_1^{l,m} = \mathbf{D}_{\text{ICC}}(1, l, m), \quad 0 \leq m < M_{\text{Proc}}, 0 \leq l < L$$

The complex cross-spectrum  $\langle L_B R_B^* \rangle^{l,m}$  for each spatial parameter set is estimated for each parameter position and processing band as:

$$\begin{aligned} \langle L_B R_B^* \rangle = & (\sigma_C^{l,m})^2 P_{L,C}^m P_{R,C}^m \rho_C^m \exp(j\phi_C^m) + \dots \\ & (\sigma_L^{l,m})^2 P_{L,L}^m P_{R,L}^m \rho_L^m \exp(j\phi_L^m) + \dots \\ & (\sigma_R^{l,m})^2 P_{L,R}^m P_{R,R}^m \rho_R^m \exp(j\phi_R^m) + \dots \\ & (\sigma_{Ls}^{l,m})^2 P_{L,Ls}^m P_{R,Ls}^m \rho_{Ls}^m \exp(j\phi_{Ls}^m) + \dots \\ & (\sigma_{Rs}^{l,m})^2 P_{L,Rs}^m P_{R,Rs}^m \rho_{Rs}^m \exp(j\phi_{Rs}^m) + \dots \\ & P_{L,L}^m P_{R,R}^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_1 + \dots \\ & P_{L,Ls}^m P_{R,Rs}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_1 + \dots \\ & P_{L,R}^m P_{R,L}^m \sigma_L^{l,m} \sigma_R^{l,m} ICC_1 \rho_L^m \rho_R^m \exp(j(\phi_L^m + \phi_R^m)) + \dots \\ & P_{L,Rs}^m P_{R,Ls}^m \sigma_{Ls}^{l,m} \sigma_{Rs}^{l,m} ICC_1 \rho_{Ls}^m \rho_{Rs}^m \exp(j(\phi_{Ls}^m + \phi_{Rs}^m)) \end{aligned}$$

for  $0 \leq m < M_{\text{Proc}}, 0 \leq l < L$ ,

and where

$$ICC_1^{l,m} = \mathbf{D}_{\text{ICC}}(1, l, m), \quad 0 \leq m < M_{\text{Proc}}, 0 \leq l < L.$$

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  $y_{L_0}^{n,k}, y_{R_0}^{n,k}$ , 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<sub>1</sub> or 7-2-7<sub>2</sub> 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<sub>3</sub> and OTT<sub>4</sub> 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  $P_{Y,X}^m$ ,  $\phi_X^m$  and  $\rho_X^m$ , for  $X \in \{L, R, C, Ls, Rs\} \dots$

by:

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

In subclause 6.11.4.2.2.2, replace:

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

by:

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

In subclause 6.11.4.2.2.2, replace:

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

by:

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

In subclause 6.11.4.2.2.2, replace:

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

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:

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

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:

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

by:

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

In subclause 6.11.4.2.2.2, in the last paragraph, replace:

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

by:

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