INTERNATIONAL STANDARD ISO/IEC 14496-3:2005/Amd.3:2006
TECHNICAL CORRIGENDUM 1

Published 2008-12-01

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

Information technology — Coding of audio-visual objects —
Part 3:
Audio

AMENDMENT 3: Scalable Lossless Coding (SLS)

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des objets audiovisuels —

Partie 3: Codage audio

AMENDEMENT 3: Codage extensible sans perte (SLS)

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 14496-3:2005/Amd.3:2006 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Replace 12.5.5.2.1 entirely, with the following (where differences to the existing text are highlighted in grey):

12.5.5.2.1 Overview

The residual integer spectral data vector is decoded from the LLE data stream *lle_data*(). Firstly, all scale factor bands with *lazy_bp* > 0 are BPGC/CBAC decoded, where the amplitude and sign of the residual spectral data *res* is bit-plane decoded starting from the maximum bit-plane *max_bp* and progressing to lower bit-planes until bit-plane 0 for each scale factor band. Subsequently, the low energy mode decoding is invoked to decode the remaining scale factor bands with *lazy_bp* <= 0.

ICS 35.040 **Ref. No. ISO/IEC 14496-3:2005/Amd.3:2006/Cor.1:2008(E)**

© ISO/IEC 2008 – All rights reserved

Published in Switzerland

Documents
Document Sharing Hub

Full document details: ISO_IEC 14496-3_2005_AMD3_2006_COR1_2008

The SLS decoder can provide the functionality of fine-grain scalability (FGS) by truncating the LLE bitstream. Moreover, it allows to decode additional symbols beyond the point of truncation by exploiting the properties of arithmetic coding.

In 12.5.5.2.2 replace the text up to Table 12.19 with the following (where differences to the existing text are highlighted in grey):

12.5.5.2.2 BPGC/CBAC decoding process

The BPGC decoding or CBAC decoding process is performed on scale factor bands for which *lazy_bp*>0. The BPGC/CBAC bit-plane decoding process is used to decode the bit-plane symbols for reconstructing the residual integer spectral data *res*. The bit-plane decoding process is started from *max_bp* for each sfb, and progressively proceeds to lower bit-planes. For the first NUM_BP bit-plane scans the bit-plane symbols are arithmetic decoded as illustrated in the following pseudo code:

```
/* preparing the help element */
for (g=0;g<num_window_groups;g++){
    for (sfb = 0;sfb<num_sfb;sfb++){
        width = swb_offset[g][sfb+1] - swb_offset[g][sfb];
        for (win = 0;win <window_group_len[g];win++) {
            for (bin=0;bin<width;bin++){
                is_sig[g][win][sfb][bin] =
                (quant[g][sfb][win][bin])&&(band_type[g][sfb]==ImplicitBand)?1:0;
                /* sign will be determined implicitly if is_sig == 1 */
                amp[g][win][sfb][bin] = 0;
                sign[g][win][sfb][bin] = 1;
            }
        }
        cur_bp[g][sfb] = max_bp[g][sfb];
    }
}

/* BPGC/CBAC decoding */
while ((max_bp[g][sfb] - cur_bp[g][sfb]<LAZY_BP) && (cur_bp[g][sfb] >= 0)){
    for (g=0;g<num_window_groups;g++){
        for (sfb = 0;sfb<num_sfb;sfb++){
            if ((cur_bp[g][sfb]>=0) && (lazy_bp[g][sfb] > 0)){
                width = swb_offset[g][sfb+1] - swb_offset[g][sfb];
                for (win=0;win<window_group_len[g];win++){
                    for (bin=0;bin<width;bin++){
                        if (!is_lle_ics_eof ()){
                            if (interval[g][win][sfb][bin] >
                                amp[g][win][sfb][bin] + (1<<cur_bp[g][sfb])
                            {
                                freq = determine_frequency();
                                sym = decode(freq);
                                amp[g][win][sfb][bin] += sym << cur_bp[g][sfb];
                                 /* decode bit-plane cur_bp*/
                                if ((!is_sig[g][win][sfb][bin]) && (sym )) {
                                        /* decode sign bit of res if necessary */
                                        sign[g][win][sfb][bin] = (decode(freq_sign))? -1:1;
                                        is_sig[g][win][sfb][bin] = 1;

                                }
                            }
                        }
                        else {
                            smart_decoding_cbac_bpgc();
                        }
                    }
                }
                cur_bp[g][sfb]--; /* progress to next bit-plane */
            }
        }
    }
}
```


© ISO/IEC 2008 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 14496-3_2005_AMD3_2006_COR1_2008