INTERNATIONAL STANDARD ISO/IEC 14496-16:2011
TECHNICAL CORRIGENDUM 1

Published 2015-04-01

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 16:
Animation Framework eXtension (AFX)

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Codage des objets audiovisuels —

Partie 16: Extension du cadre d'animation (AFX)

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 14496-16:2011 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 29, *Coding of Audio, Picture, Multimedia and Hypermedia Information*.

Replace section 5.2.5.3.5.1 Syntax

"Class IntArrayDecoder (numberOfdata, dim)
{
….
}"
with

```
Class IntArrayDecoder(numberOfdata, dim)
{
  Bit(4) predictionMode;
  Bit(4) binarizationMode;
  If ((binarizationMode == 0) && (predictionMode == 0))          // FL
  {
```

ICS 35.040 **Ref. No. ISO/IEC 14496-16:2011/Cor.1:2015(E)**

Published in Switzerland

```
    unsinged int (32) streamSizeFL;
    Bit(8) QP;
    for(i=0;i< numberOfdata *dim;i++)
      bit (QP) nData; // simple QBCR
  }
  else if (binarizationMode == 1)          // BPC
  {
    unsinged int (32) streamSizeBPC;
    If (predictionMode==3)      bit(1-7) predictor;
    bit (5) prefixSize;
    for(i=0;i< numberOfdata *dim;i++)
      {BPDecoder(prefixSize) nDifData;
      If ((predictionMode==1,4,5)&&(nDifData != 0))     bit(1) nSign;
    }
  }
  else if  (binarizationMode == 2)         // 4C
  {
      unsinged int (32) streamSize4C;
      for(i=0;i< numberOfdata *dim;i++)
      {
            If (predictionMode==3)
            {
               bit(3) predictor;
                bit (1) terminationBit;
                while (terminationBit)
                {
                      bit(3)      threeBitsFL;
                      bit(1)      terminationBit;
                }
            }
            else
            {
                Do
                {
                      bit(3)      threeBitsFL;
                      bit(1)      terminationBit;
                }
                while (terminationBit)
            }
        If((predictionMode==1,4,5)&&(difValue!=0))     bit(1) signBit;
        }
  }
  else if  (binarizationMode == 3)         // AC
  {
        unsinged int (32) streamSizeAC;
        for(i=0;i< numberOfdata *dim;i++)
        {
            If (predictionMode==3)      ACDecoder(8) predictor
            ACDecoder(1<<QP) nValue
            ACDecoder(2) hasnext
            If (nValue!= 0)      ACDecoder(1) nSign
        }
  }
  else if  (binarizationMode == 4)         // AC/EGk
  {
        unsinged int (32) streamSizeACEGk;
        unsinged int (8) K
        unsinged int (8) M
        for(i=0;i< numberOfdata *dim;i++)
        {
```