INTERNATIONAL STANDARD

ISO/IEC 15938-3

First edition
2002-05-15

AMENDMENT 4
2010-10-15

Information technology — Multimedia content description interface —

Part 3: **Visual**

AMENDMENT 4: Video signature tools

Technologies de l'information — Interface de description du contenu multimédia —

Partie 3: Visuel

AMENDEMENT 4: Outils de vidéosignature

Reference number
ISO/IEC 15938-3:2002/Amd.4:2010(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 4 to ISO/IEC 15938-3:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Multimedia content description interface —

Part 3: **Visual**

AMENDMENT 4: Video signature tools

Replace 1.2 with:

1.2 Overview of Visual Description Tools

This part of ISO/IEC 15938 specifies tools for description of visual content, including still images, video and 3D models. These tools are defined by their syntax in DDL and binary representations and semantics associated with the syntactic elements. They enable description of the visual features of the visual material, such as color, texture, shape, motion, localization of the described objects in the image or video sequence and also unique and robust identification of visual material. An overview of the visual description tools is shown in Figure 1.

The basic structure description tools include five supporting tools of visual descriptions defined in Clauses 6-11. They are categorized into two groups, descriptor containers and basic supporting tools. The former consists of three datatypes, GridLayout providing efficient representations of visual features on grids, TimeSeries representing temporal arrays of several descriptions, GofGopFeature describes representative descriptions over video segment, and MultipleView describing a 3D object using several pictures captured from different view angles. The latter contains two tools, Spatial2DcoordinateSystem used to specify the 2D coordinate system and TemporalInterpolation indicating the interpolation method between two samples on a time axis.

The remaining description tools, except for the FaceRecognition and ImageSignature descriptors, are associated with visual features and are grouped into five feature categories: Color, Texture, Shape, Motion and Localization.

The color description tools include five color descriptors to represent different aspects of color features: representative colors (DominantColor), color distribution (ScalableColor), spatial distribution of colors (ColorLayout and ColorStructure) and perceptual feeling of illumination color (ColorTemperature). It also contains three supporting tools, ColorSpace and ColorQuantization used in DominantColor and IlluminationInvariantColor to extend four color descriptors, DominantColor, ScalableColor, ColorLayout and ColorStructure, to support illumination invariant similarity matching. An extension of ScalableColor to a group of frames or pictures (GoFGoPColor) is also included in this group. All the color descriptors can be extracted from arbitrarily shaped regions.

The texture description tools facilitate browsing (TextureBrowsing) and similarity retrieval (HomogeneousTexture and EdgeHistogram) using the texture of a still or moving image region. All the texture descriptors can be extracted from arbitrarily shaped regions.

The shape description tools include two descriptors that characterize different shape features of a 2D object or region. The RegionShape descriptor captures the distribution of all pixels within a region and the Contour Shape descriptor characterizes the shape properties of the contour of an object. The extension of RegionShape is also defined as ShapeVariation to describe temporal variation of shape over video segment. The Shape3D and Perceptual 3D Shape descriptors provide 3-dimensional shape information; the former represents an intrinsic shape characterization of 3D mesh models, and the latter represents part-based representation of a 3D object.

The motion description tools include four descriptors that characterize various aspects of motion. The CameraMotion descriptor specifies a set of basic camera operations such as, for example, panning and tilting. The motion of a key point (pixel) from a moving object or region can be characterized by the MotionTrajectory descriptor. The ParametricMotion descriptor characterizes an evolution of an arbitrarily shaped region over time in terms of a 2D geometric transformation. Finally, the MotionActivity descriptor captures the pace of the motion in the sequence, as perceived by the viewer. All motion descriptors except for CameraMotion can be extracted from arbitrarily shaped regions.

The localization description tools can be used to indicate regions of interest in the spatial (RegionLocator) and spatio-temporal (SpatioTemporalLocator) domains.

The FaceRecognition descriptor and the Advanced Face Recognition descriptor are not associated with any particular visual feature and can be used to describe a human face for applications requiring the matching and retrieval of face images.

The signature descriptors provide a "fingerprint" that uniquely identifies image and video content. The signatures are robust (unchanging) across a wide range of common editing operations, but are sufficiently different for every item of "original" content to allow unique and reliable identification – just like human fingerprints. There are two visual signatures; the ImageSignature and VideoSignature are descriptors for images and videos respectively. The signatures have no direct association with specific visual features such as colour, shape or texture.

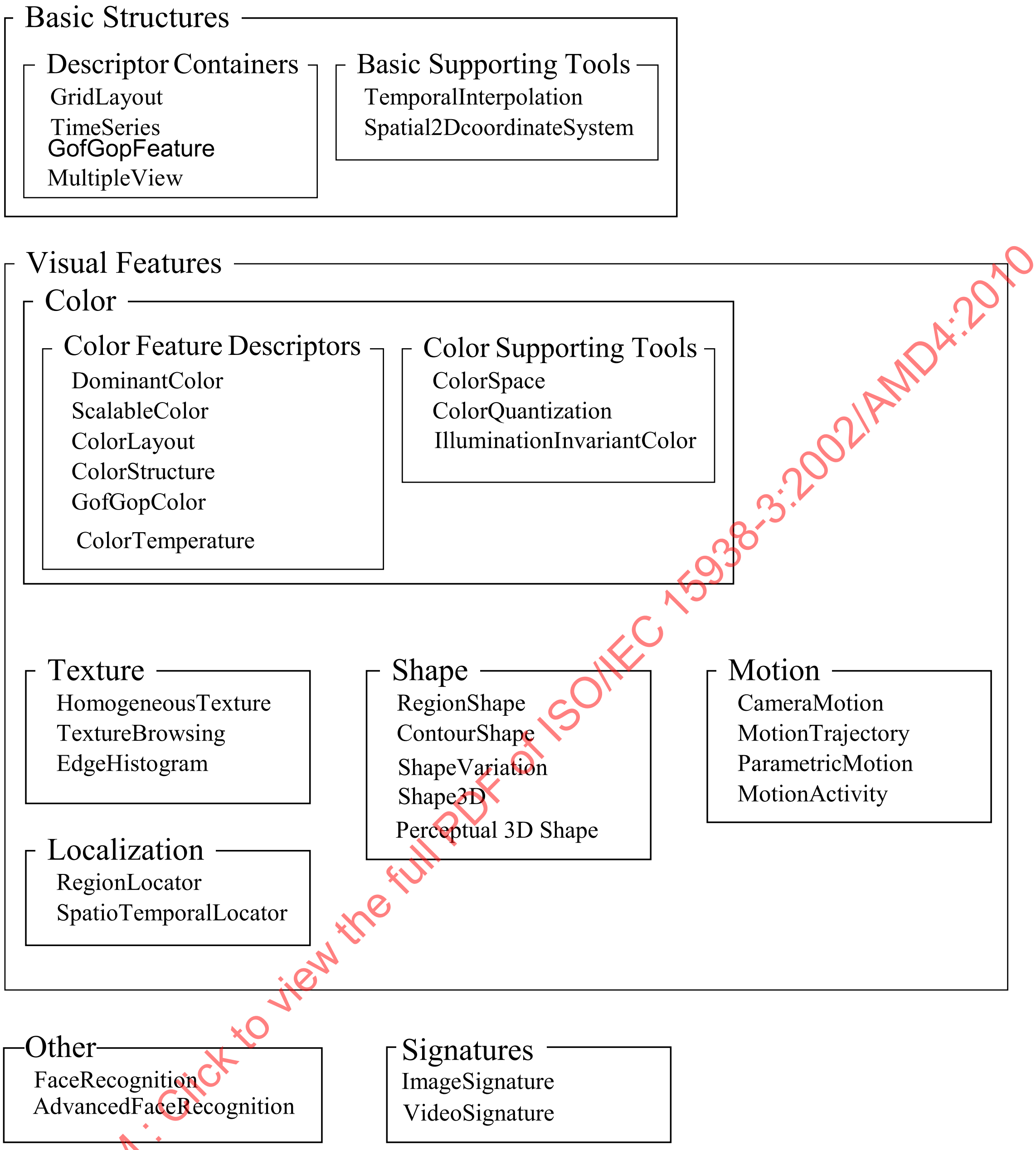

Figure 1 — Overview of Visual Description Tools

In 3.3, extend the definitions:

floor Maximum integer number less than or equal to the given floating point number

Replace 4.2.2 with:

4.2.2 Generic binary representation

The use of the video-specific syntax is signalled using the codec configuration mechanism defined in ISO/IEC 15938-1. The following classification scheme is defined for this purpose.

```
<ClassificationScheme uri="urn:mpeg:mpeg7:cs:VisualDescriptorCodecCS:2001">
   <Term termID="1">
      <Name xml:lang="en">MPEG7CameraMotion</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Camera Motion
             Codec</Definition>
   </Term>
   <Term termID="2">
      <Name xml:lang="en">MPEG7ColorLayout</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Layout
             Codec</Definition>
   </Term>
   <Term termID="3">
      <Name xml:lang="en">MPEG7ColorQuantization</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Quantization
             Codec</Definition>
   </Term>
   <Term termID="4">
      <Name xml:lang="en">MPEG7ColorSpace</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Space
             Codec</Definition>
   </Term>
   <Term termID="5">
      <Name xml:lang="en">MPEG7ColorStructure</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Color Structure
             Codec</Definition>
   </Term>
   <Term termID="6">
      <Name xml:lang="en">MPEG7ContourShape</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Contour Shape
             Codec</Definition>
   </Term>
   <Term termID="7">
      <Name xml:lang="en">MPEG7DominantColor</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Dominant Color
             Codec</Definition>
   </Term>
   <Term termID="8">
      <Name xml:lang="en">MPEG7EdgeHistogram</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Edge Histogram
             Codec</Definition>
   </Term>
   <Term termID="9">
      <Name xml:lang="en">MPEG7FaceRecognition</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Face Recognition
             Codec</Definition>
   </Term>
   <Term termID="10">
      <Name xml:lang="en">MPEG7FoFGoPColor</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary GoFGoP Color
             Codec</Definition>
   </Term>
   <Term termID="11">
      <Name xml:lang="en">MPEG7GridLayout</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Grid Layout
             Codec</Definition>
   </Term>
   <Term termID="12">
      <Name xml:lang="en">MPEG7HomogeneousTexture</Name>
      <Definition xml:lang="en">ISO/IEC 15938-3 Binary Homogeneous Texture
             Codec</Definition>
```

```
    </Term>
    <Term termID="13">
       <Name xml:lang="en">MPEG7IrregularVisualTimeSeries</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Irregular Time Series
              Codec</Definition>
    </Term>
    <Term termID="14">
       <Name xml:lang="en">MPEG7MotionActivity</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Motion Activity
              Codec</Definition>
    </Term>
    <Term termID="15">
       <Name xml:lang="en">MPEG7MotionTrajectory</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Motion Trajectory
              Codec</Definition>
    </Term>
    <Term termID="16">
       <Name xml:lang="en">MPEG7MultipleView</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Multiple View
              Codec</Definition>
    </Term>
    <Term termID="17">
       <Name xml:lang="en">MPEG7ParametricMotion</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Parametric Motion
              Codec</Definition>
    </Term>
    <Term termID="18">
       <Name xml:lang="en">MPEG7RegionLocator</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Region Locator
              Codec</Definition>
    </Term>
    <Term termID="19">
       <Name xml:lang="en">MPEG7RegionShape</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Region Shape
              Codec</Definition>
    </Term>
    <Term termID="20">
       <Name xml:lang="en">MPEG7RegularVisualTimeSeries</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Regular Time Series
              Codec</Definition>
    </Term>
    <Term termID="21">
       <Name xml:lang="en">MPEG7ScalableColor</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Scalable Color
              Codec</Definition>
    </Term>
    <Term termID="22">
       <Name xml:lang="en">MPEG7Shape3D</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Shape 3D
              Codec</Definition>
    </Term>
    <Term termID="23">
       <Name xml:lang="en">MPEG7Spatial2DCoordinateSystem</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary Spatial 2D Coordinate
              System Codec</Definition>
    </Term>
    <Term termID="24">
       <Name xml:lang="en">MPEG7SpatioTemporalLocator</Name>
       <Definition xml:lang="en">ISO/IEC 15938-3 Binary SpatioTemporal Locator
              Codec</Definition>
    </Term>
    <Term termID="25">
        <Name xml:lang="en">MPEG7TemporalInterpolation</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Temporal Interpolation
                Codec</Definition>
    </Term>
    <Term termID="26">
        <Name xml:lang="en">MPEG7TextureBrowsing</Name>
        <Definition xml:lang="en">ISO/IEC 15938-3 Binary Texture Browsing
                Codec</Definition>
    </Term>
    <Term termID=”27”>
        <Name xml:lang=”en”>MPEG7GofGopFeature</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Gof Gop Feature
Codec</Definition>
    </Term>
    <Term termID=”28”>
        <Name xml:lang=”en”>MPEG7ColorTemperature</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Color Temperature
Codec</Definition>
    </Term>
    <Term termID=”29”>
        <Name xml:lang=”en”>MPEG7ShapeVariation</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Shape Variation
Codec</Definition>
    </Term>
    <Term termID=”30”>
        <Name xml:lang=”en”>MPEG7IlluminationInvariantColor</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Illumination Invariant
Color Codec</Definition>
    </Term>
    <Term termID=”31”>
        <Name xml:lang=”en”>MPEG7AdvancedFaceRecognition</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Advanced Face Recognition
Codec</Definition>
    </Term>
    <Term termID=”32”>
        <Name xml:lang=”en”>MPEG7Perceptual3DShape</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Perceptual 3D Shape
Codec</Definition>
    </Term>
    <Term termID=”33”>
        <Name xml:lang=”en”>MPEG7ImageSignature</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Image Signature
Codec</Definition>
    </Term>
    <Term termID=”34”>
        <Name xml:lang=”en”>MPEG7VideoSignature</Name>
        <Definition xml:lang=”en”>ISO/IEC 15938-3 Binary Video Signature
Codec</Definition>
    </Term>
</ClassificationScheme>
```

In 5.2.4, replace Table 1 with:

Table 1 — Assignment of IDs to descriptors

ID	Descriptor
0	Forbidden
1	CameraMotion
2	ColorLayout
3	ColorSpace
4	ColorStructure
5	ColorQuantization
6	ContourShape
7	DominantColor
8	EdgeHistogram
9	FaceRecognition
10	GoFGoPColor
11	GridLayout
12	HomogeneousTexture
13	IrregularVisualTimeSeries
14	MotionActivity
15	MotionTrajectory
16	MultipleView
17	ParametricMotion
18	RegionLocator
19	RegionShape
20	RegularVisualTimeSeries
21	ScalableColor
22	Shape3D
23	Spatial2DCoordinateSystem
24	SpatioTemporalLocator
25	TemporalInterpolation
26	TextureBrowsing
27	GofGopFeature
28	ColorTemperature
29	ShapeVariation
30	IlluminationInvariantColor
31	AdvancedFaceRecognition
32	Perceptual3DShape
33	ImageSignature
34	VideoSignature
35-255	Reserved

After 11.3, add the following:

11.4 Video Signature

11.4.1 Introduction

The visual content descriptors in Sections 6-9 are very useful when trying to find videos with similar content. These descriptors are intended to be general and were found to be unsuitable for the task of finding duplicate content. The video signature descriptor is designed to identify duplicate video content. This descriptor is robust (unchanging) to a wide range of common video editing operations, but is sufficiently different for every "original" content to identify it uniquely and reliably – just like human fingerprints.

The video signature is composed of three main elements,

- a frame signature,
- a set of compact summary frame signatures - referred to as words
- and a group-of-frames representation for a temporal segment -referred to as a bag-of-words.

A video is assumed to be made up of a set of frames (or pictures) each representing a single temporal sample. A frame is made of a set of pixels each representing a single spatial sample. The frame signature is extracted from each frame of a video. It is a 380 dimensional vector of base-3 ternary values that describe the intensities and the intensity inter-relations between pixel regions in the frames. Each dimension can be characterized as a mean, first or second order operator.

Words are compact, 1 byte, representations of the frame signature. All possible combinations of values for a word are referred to as the vocabulary. The words provide a summary representation of the frame.

A bag-of-words representation is often used in text searching to compare the similarity between two documents. It ignores the ordering of the text and therefore provides some robustness to editing. For the video signature a bag-of-words records the occurrence of *words* within a temporal segment of frames. The bag-of-words therefore provides a coarse descriptor for the temporal segment.

The video signature descriptor syntax provides support for description of single or multiple static spatial regions within the frame. Each spatial region is a rectangular region having arbitrary position and size, with edges parallel to the edges of the frame. Each spatial region may have its own start and end media times. This feature is useful when describing content such as videos with picture-in-picture, where the entire frame region can be described as the first spatial region and the picture-in-picture region can be described as the second spatial region.

The extraction procedure shall be applied to each spatial region independently. Specifically, only pixels within the spatial region are processed to extract the video signature.

11.4.2 DDL representation syntax

```
<complexType name="VideoSignatureType" final="#all">
  <complexContent>
    <extension base="mpeg7:VisualDType">
      <sequence>
        <element name="VideoSignatureRegion" maxOccurs="4294967295">
          <complexType>
            <sequence>
              <element name="VideoSignatureSpatialRegion" minOccurs="0">
                <complexType>
                  <sequence>
                    <element name="Pixel" minOccurs="2" maxOccurs="2">
                      <simpleType>
                        <restriction base="mpeg7:integerVector">
                          <length value="2" />
                        </restriction>
                      </simpleType>
                    </element>
                  </sequence>
                </complexType>
              </element>
              <element name="StartFrameOfSpatialRegion" type="mpeg7:unsigned32"/>
              <element name="MediaTimeUnit" type="mpeg7:unsigned16"/>
              <element name="MediaTimeOfSpatialRegion" minOccurs="0">
                <complexType>
                  <sequence>
        <element name="StartMediaTimeOfSpatialRegion" type="mpeg7:unsigned32"/>
        <element name="EndMediaTimeOfSpatialRegion" type="mpeg7:unsigned32"/>
      </sequence>
    </complexType>
  </element>
  <element name="VSVideoSegment" maxOccurs="4294967295">
    <complexType>
      <sequence>
        <element name="StartFrameOfSegment" type="mpeg7:unsigned32"/>
        <element name="EndFrameOfSegment" type="mpeg7:unsigned32"/>
        <element name="MediaTimeOfSegment" minOccurs="0">
          <complexType>
            <sequence>
              <element name="StartMediaTimeOfSegment" type="mpeg7:unsigned32"/>
              <element name="EndMediaTimeOfSegment" type="mpeg7:unsigned32"/>
            </sequence>
          </complexType>
        </element>
        <element name="BagOfWords" minOccurs="5" maxOccurs="5">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned1"/>
              </simpleType>
              <length value="243"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </complexType>
  </element>
  <element name="VideoFrame" maxOccurs="4294967295">
    <complexType>
      <sequence>
        <element name="MediaTimeOfFrame" type="mpeg7:unsigned32" minOccurs="0"/>
        <element name="FrameConfidence" type="mpeg7:unsigned8"/>
        <element name="Word">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned8"/>
              </simpleType>
              <length value="5"/>
            </restriction>
          </simpleType>
        </element>
        <element name="FrameSignature">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned2"/>
              </simpleType>
              <length value="380"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </complexType>
  </element>
</sequence>
```

```
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Descriptor example:

```
<VideoSignature>
  <VideoSignatureRegion>
    <VideoSignatureSpatialRegion>
      <Pixel>0 0</Pixel>
      <Pixel>719 479</Pixel>
    </VideoSignatureSpatialRegion>
    <StartFrameOfSpatialRegion>0</StartFrameOfSpatialRegion>
    <MediaTimeUnit>1000</MediaTimeUnit>
    <MediaTimeOfSpatialRegion>
      <StartMediaTimeOfSpatialRegion>0</StartMediaTimeOfSpatialRegion>
      <EndMediaTimeOfSpatialRegion>5038</EndMediaTimeOfSpatialRegion>
    </MediaTimeOfSpatialRegion>
    <VSVideoSegment>
      <StartFrameOfSegment>0</StartFrameOfSegment>
      <EndFrameOfSegment>89</EndFrameOfSegment>
      <MediaTimeOfSegment>
        <StartMediaTimeOfSegment>0</StartMediaTimeOfSegment>
        <EndMediaTimeOfSegment>2969</EndMediaTimeOfSegment>
      </MediaTimeOfSegment>
      <BagOfWords>1 1 1 1 0 1 0 0 .....
        .............
      </BagOfWords>
        .............
    </VSVideoSegment>
<!-- 2 more VSVideoSegments -->
    <VideoFrame>
      <MediaTimeOfFrame>0</MediaTimeOfFrame>
      <FrameConfidence>100</FrameConfidence>
      <Word>1 2 3 4 5</Word>
      <FrameSignature>1 2 1 0 1 0 2 1 0 2 .....
        .............
      </FrameSignature>
    </VideoFrame>
<!-- 150 more VideoFrames -->
  </VideoSignatureRegion>
</VideoSignature>
```

11.4.3 Binary Representation Syntax

VideoSignature {	Number of bits	Mnemonics
NumOfSpatialRegions	32	uimsbf
for(r=0; r< NumOfSpatialRegions; r++) {		
SpatialLocationFlag	1	bslbf
if(SpatialLocationFlag == 1) {		
for(i=0;i<2;i++){		
PixelX	16	uimsbf
PixelY	16	uimsbf
}		
}		
StartFrameOfSpatialRegion	32	uimsbf
NumOfFrames	32	uimsbf
MediaTimeUnit	16	uimsbf
MediaTimeFlagOfSpatialRegion	1	bslbf
if(MediaTimeFlagOfSpatialRegion == 1) {		
StartMediaTimeOfSpatialRegion	32	uimsbf
EndMediaTimeOfSpatialRegion	32	uimsbf
}		
NumOfSegments	32	uimsbf
for(i=0; i< NumOfSegments; i++) {		
StartFrameOfSegment	32	uimsbf
EndFrameOfSegment	32	uimsbf
MediaTimeFlagOfSegment	1	bslbf
if(MediaTimeFlagOfSegment == 1) {		
StartMediaTimeOfSegment	32	uimsbf
EndMediaTimeOfSegment	32	uimsbf
}		
for(j=0; j< WordsPerFrame; j++) {		
BagOfWords[j]	243	bslbf
}		
}		
CompressionFlag	1	bslbf
if(CompressionFlag ==0) {		
for(i=0; i< NumOfFrames; i++) {		
MediaTimeFlagOfFrame	1	bslbf
if(MediaTimeFlagOfFrame == 1) {		
MediaTimeOfFrame	32	uimsbf
}		
FrameConfidence	8	uimsbf
for(j=0; j< WordsPerFrame; j++) {		
Word[j]	8	uimsbf
}		
FrameSignature	608	bslbf
}		
} else {		
for (i=0; i<NumOfFrames; i++) {		
MediaTimeFlagOfFrame	1	bslbf
if(MediaTimeFlagOfFrame == 1) {		
MediaTimeOfFrame	32	uimsbf
}		
FrameConfidence	8	uimsbf
for (j=0; j < WordsPerFrame; j++) {		
Word[j]	8	uimsbf
}		
}		

CompressedSegmentLength = 45		
n = 0		
for (i=0; i < NumOfSegments; i++) {		
if (i == NumOfSegments -1) {		
CompressedSegmentLength = NumOfFrames - n		
}		
CompressedSegment( )		bslbf
n += CompressedSegmentLength		
}		
}		
}		
}		

CompressedSegment {	Number of bits	Mnemonics
num_frames = 0		
while (num_frames < CompressedSegmentLength) {		
FrameSignature	608	bslbf
GOPLengthm1	ceil(ld(Segment Length))	uimsbf
PredictedPictures( )		bslbf
num_frames = num_frames + GOPLengthm1 + 1		
}		
}		

PredictedPictures {	Number of bits	Mnemonics
decoded_el = 0		
num_el = GOPLengthm1 × 380		
while (decoded_el < num_el) {		
ZeroRL	Variable (see Exp-Golomb coding in 11.4.9.1).	bslbf
decoded_el = decoded_el + ZeroRL		
if (decoded_el == num_el) then break		
NonZeroSymbol	1	
decoded_el = decoded_el + 1		
}		
}		

WordsPerFrame = 5

11.4.4 Descriptor Component Semantics

NumOfSpatialRegions

This field, which is only present in the binary syntax, specifies the number of spatial regions from the video.

SpatialLocationFlag

This field, which is only present in the binary syntax, indicates the presence of the PixelX, PixelY, elements, which specify the location of the spatial region. If the flag is set to 1 these elements are present, if the flag is set to 0 these elements are not present.

VideoSignatureSpatialRegion, Pixel, PixelX, PixelY

These attributes specify a rectangular region, by locating the position of the top-left and bottom-right points. The origin of the coordinate system (0,0) is in the top-left corner of the frame. In the binary representation the position of the top-left corner of the region is specified by the first occurrence of (PixelX, PixelY) and the position of the bottom-right corner is specified by the second occurrence of (PixelX, PixelY). Where, PixelX and PixelY specify respectively the X and Y coordinates of a pixel. In the DDL representation the position of the top-left corner of the region is specified by the first occurrence of Pixel, and the position of the bottom-right corner is specified by the second occurrence of Pixel. Where the first element of the Pixel vector corresponds to the X coordinate and the second element corresponds to the Y coordinate.

StartFrameOfSpatialRegion

This element specifies the frame index of the first frame represented in the spatial region representation.

NumOfFrames

This field, which is only present in the binary representation, specifies the number of frame signatures represented in the spatial region representation.

MediaTimeUnit

This element specifies the unit of the media time as a fraction of a second. For example, if MediaTimeUnit is 1000, the unit of the media time is 1/1000 second. The time unit specified by this element is used as a time unit for the elements StartMediaTimeOfSpatialRegion, EndMediaTimeOfSpatialRegion, StartMediaTimeOfSegement, EndMediaTimeOfSegment, and MediaTimeOfFrame.

MediaTimeFlagOfSpatialRegion

This field, which is only present in the binary representation, indicates the presence of the fields specifying the start and end media time of the spatial region representation, i.e. the elements StartMediaTimeOfSpatialRegion and EndMediaTimeOfSpatialRegion. If the flag is set to 1 the elements are present if the flag is set to 0 the elements are not present.

MediaTimeOfSpatialRegion, StartMediaTimeOfSpatialRegion, EndMediaTimeOfSpatialRegion

These elements specify the media time of the first (StartMediaTimeOfSpatialRegion) and last (EndMediaTimeOfSpatialRegion) frame represented in the spatial region.

NumOfSegments

This field, which is only present in the binary representation, specifies the number of temporal segments represented, these are referred to as bag-of-words.

StartFrameOfSegment, EndFrameOfSegment

These elements specify the frame index of the first (StartFrameOfSegment) and last (EndFrameOfSegment) frames represented in the temporal segment.

MediaTimeFlagOfSegment

This field, which is only present in the binary representation, indicates the presence of the fields specifying the start and end media time of the segment, i.e. the elements StartMediaTimeOfSegment and

EndMediaTimeOfSegment. If the flag is set to 1 the elements are present, if the flag is set to 0 the elements are not present.

MediaTimeOfSegment, StartMediaTimeOfSegment, EndMediaTimeOfSegment

These elements specify the media time of the first (StartMediaTimeOfSegment) and last (EndMediaTimeOfSegment) frame represented in the temporal segment.

WordsPerFrame

This element specifies the number of Word elements extracted from each frame.

VSVideoSegment

This element specifies a representation of a temporal segment in a video.

BagOfWords

This element specifies the bag-of-words representation for a temporal segment of 90 frames, for all but the last segments. The extraction of bag-of-words is described in 11.4.8.

MediaTimeFlagOfFrame

This field, which is only present in the binary representation, indicates the presence of the element MediaTimeOfFrame. If the flag is set to 1, the element is present, if the flag is set to 0 the element is not present.

MediaTimeOfFrame

This element specifies the media time of the frame.

VideoFrame

This element specifies a representation of a single video frame.

FrameConfidence

This element specifies the confidence value attributed to a frame signature. Low frame confidence represents a flat image where there is little or no luminance difference between sub-regions, for example a black screen. The extraction of confidence is described in 11.4.6.

Word

This element specifies a compact representation of the frame signature. The extraction of Word elements is described in 11.4.7.

FrameSignature

This element specifies the full video signature representation of the frame. It is a 380 dimensional vector of base-3 ternary values $\{0,1,2\}$. The extraction of frame signature is described in 11.4.5.

In the binary representation the extracted ternary elements $\mathbf{x} = \{x_1, x_2, \cdots, x_{380}\}$ shall be encoded in 5-dimensional units, where each unit is encoded into 8 bits. Therefore, a total of 380/5=76 units (608 bits) are used to encode the frame signature.

Let $b_j,\ j=1,\ldots,76$ denote the encoded value of each 5-dimensional unit which shall be calculated using the following equation:

$$b_j = 81\times x_{5j-4} + 27\times x_{5j-3} + 9\times x_{5j-2} + 3\times x_{5j-1} + x_{5j}\ ,$$

the encoding is shown explicitly in Table E.1.

In the CompressedSegment field the FrameSignature element is a key picture.

CompressionFlag

This field, only present in the binary syntax, indicates whether the frame signatures are compressed. When CompressionFlag is set to 1 the corresponding frame signatures are compressed, when the field is set to 0 the frame signatures are not compressed.

CompressedSegment

This field, only present in the binary syntax, specifies the frame signatures in a compressed form. The ordering of compressed segments corresponds to the ordering of the temporal segments (bag-of-words) and the start frame of each compressed segment corresponds to the StartFrameOfSegment element. The decoding of the compressed segment is specified in 11.4.9.

CompressedSegmentLength

This field, only present in the binary syntax, specifies the length of a compressed segment, in number of frames. It shall be equal to 45 for all but the last segment. In the last segment it shall be NumOfFrames - 45 * (NumOfSegments - 1).

GOPLengthm1

This field, only present in the binary syntax, specifies the GOP length, in number of frames minus one. For instance, GOP consisting of one KP and one PP has the value of this field equal to 1.

PredictedPictures

This field, only present in the binary syntax, specifies part of the bit-stream with encoded predicted pictures. The decoding of the PredictedPictures field is specified in 11.4.9.1.

ZeroRL

This field, only present in the binary syntax, specifies the length of run of ternary 0 encoded using the Exp-Golomb coding.

NonZeroSymbol

This field, only present in the binary syntax, specifies an encoded symbol. When this field is equal to 0 the symbol is ternary 1. When this field is equal to 1 the symbol is ternary 2.

11.4.5 Frame Signature Extraction

This Subclause specifies the extraction process for the FrameSignature element of the Video Signature descriptor.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351
352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383
384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447
448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479
480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511
512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543
544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607
608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639
640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671
672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703
704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735
736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767
768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799
800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831
832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863
864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895
896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927
928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959
960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991
992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

Figure Amd4-1 — Block partitioning and indices.

The extraction of frame signature shall be carried out for each spatial region independently. Specifically, only pixels within the spatial region shall be processed to extract the frame signature, as if the spatial region formed the entire frame region. Hereafter, the term “frame” refers to the entire frame region or a spatial region thereof.

The 380 dimensional ternary vector is extracted from predefined sub-regions associated with each dimension. The luminance values of a frame shall be spatially resampled to 32x32 pixel resolution, where the width and the height of the frame are equally split into 32. Figure Amd4-1 shows the block partitioning and the indices given to each resampled pixel. These resampled pixels are used to construct the sub-regions. Here, luminance shall be the Y component of the YCbCr colour space defined in 6.2 Color Space.

Tables F.1 and F.2 show the indices of the pixels which construct the sub-regions associated with each dimension. Dimensions 1-32 (Table F.2), which are called the average element dimensions, have one associated sub-region. Dimensions 33-380 (Table F.2), which are called the difference element dimensions, have two associated sub-regions. The notation “m-n” in Tables F.1 and F.2 represents a rectangular region with the index of its top-left pixel being m, and the index of its bottom-right pixel being n. Symbol “|” represents that the sub-region is composed of multiple rectangular regions separated by this symbol. For example “m-n | o-p | q-r” represents a region that is composed of three rectangular regions of “m-n”, “o-p”, and “q-r”. Dimensions 1-32 (average element dimensions) are further categorized by their patterns of sub-regions into 2 different pattern-types (pattern-type A1 and A2), specified in the second column of Table F.1. Dimensions 33-380 (difference element dimensions) are further categorized by their patterns of sub-regions into 8 different pattern-types (pattern-type D1 – D8), specified in the second column of Table F.2. Figures G.1 to G.8 show samples of sub-regions for dimensions 21, 33, 149, 174, 210, 240, 302, and 311, respectively.

Ternary values of dimensions 1-32 (average element dimensions) shall be calculated by quantizing the average luminance of the associated sub-region. Ternary values of dimensions 33-380 (difference element dimensions) shall be calculated by quantizing the differences between the average luminances of the associated two sub-regions. Thus, the frame signature is composed of a ternary value vector with 32 average element dimensions, and 348 difference element dimensions.

The extraction of the frame signature from a video frame is carried out in 3 steps.

Step 1

The frame luminance shall be spatially resampled to 32x32 pixel resolution. The luminance corresponds to the Y component of the YCbCr colour space, ranged [0,255]. The spatial resampling shall be done by partitioning the frame into 32x32 blocks, where the width and the height of the frame shall both be split into 32 equal regions, as shown in Figure Amd4-1. Then, the mean luminance shall be calculated for each block. The mean luminance shall be calculated by summing the luminance value of the pixels in the block, and dividing the sum by the number of pixels, where no rounding is intended.

The following pseudo-code shows how a frame shall be resampled.

```
original_image[image_width*image_height];
resampled_image[32*32];
sum_luminance[32*32];
num_pixel[32*32];

for( n = 0; n < 32*32; n++ ){
    sum[n] = 0;
    num_pixel[n] = 0;
}
for( y = 0; y < image_height; y++ ){
    j = (y*32)/image_height;
    for( x = 0; x < image_width; x++ ){
        i = (x*32)/image_width;
        sum[j*32+i] += original_image[y*image_width+x];
        num_pixel[j*32+i]++;
    }
}
for( n = 0; n < 32*32; n++ ){
    resampled_image[n] = sum_luminance[n] ÷ num_pixel[n];
}
```

Step 2

For each dimension, the average luminance of the associated sub-region(s) shall be calculated using the resampled 32x32 pixel resolution frame. The average luminance of a sub-region shall be calculated as the average luminance of the pixels which construct the sub-region. The average luminance of the sub-regions shall be calculated by summing the luminance of the pixels which construct the sub-region, and dividing the sum by the number of pixels, where no rounding is intended. See Tables F.1 and F.2 for the indices of the pixels which construct the sub-regions.

Step 3

For each dimension, the average luminance of its sub-region(s) shall be used to calculate the ternary value.

For dimensions 1-32 (average element dimensions), let $v1_i$ denote the average luminance of the sub-region 1, where $i = 1, \ldots, 32$ denotes the dimension index. The average luminance shall be quantized into a ternary value x_i using the following equation:

$$x_i = \begin{cases} 2 & (\text{if} \quad v1_i - 128 \ > \ ThA_{type}) \\ 1 & (\text{if} \quad \text{Abs}(v1_i - 128) \ \leq \ ThA_{type}) \\ 0 & (\text{if} \quad v1_i - 128 \ < \ -ThA_{type}) \end{cases}$$

The threshold ThA_{type} is a threshold defined independently for each of the 2 pattern-types (type A1 and type A2) of the average element dimensions, where $type = A1, A2$ denotes the pattern-type. The pattern-type of the average element dimensions is specified in the second column of Table F.1. Thresholds ThA_{type} shall be adaptively determined for each frame, which are defined as the 33.3% percentile rank of the absolute values $\mathrm{Abs}(v1_i - 128)$ of the dimensions categorized in the same pattern-type. More specifically, threshold ThA_{type} for each pattern-type $type$ shall be calculated by:

1. For the N dimensions composing pattern-type $type$, calculate the absolute values $\mathrm{Abs}(v1_i - 128)$ and sort them in the ascending order. Let $a_{type,k} = \{a_{type,0}, a_{type,1}, \ldots, a_{type,N-1}\}$ denote the values sorted in the ascending order. Note that $k = 0, \ldots, N-1$ is an index representing the order of the sorted values, and is different from dimension index i.

2. Threshold ThA_{type} shall be the element $a_{type,k}$ where $k = \mathrm{floor}(0.333 \times N)$.

For example, for pattern-type A1 ($type = A1$, dimensions 1-20) composed of 20 dimensions ($N = 20$), let $a_{A1,k} = \{a_{A1,0}, a_{A1,1}, \ldots, a_{A1,19}\}$ denote the absolute values $\mathrm{Abs}(v1_i - 128), i = 1, \ldots, 20$ of the 20 dimensions composing pattern-type A1 sorted in the ascending order. Threshold ThA_{A1} shall be the element $a_{A1,k}$ where $k = \mathrm{floor}(0.333 \times 20) = \mathrm{floor}(6.66) = 6$. The threshold for each pattern-type is explicitly shown in Table Amd4-1.

For dimensions 33-380 (difference element dimensions), let $v1_i$ and $v2_i$ denote the average luminance of the two sub-regions (sub-region 1 and sub-region 2), where $i = 33, \ldots, 380$ denotes the dimension index. The difference of these values $v1_i - v2_i$ shall be quantized into a ternary value x_i using the following equation:

$$x_i = \begin{cases} 2 & (\text{if} \quad v1_i - v2_i \; > \; ThD_{type}) \\ 1 & (\text{if} \quad \mathrm{Abs}(v1_i - v2_i) \; \leq \; ThD_{type}) \\ 0 & (\text{if} \quad v1_i - v2_i \; < \; -ThD_{type}) \end{cases}$$

The threshold ThD_{type} is a threshold defined independently for each of the 8 pattern-types (type D1 – type D8) of the difference element dimensions, where $type = D1, \ldots, D8$ denotes the pattern-type. The pattern-type of the difference element dimensions is specified in the second column of Table F.2. Thresholds ThD_{type} shall be adaptively determined for each frame, which are defined as the 33.3% percentile rank of the absolute differences $\mathrm{Abs}(v1_i - v2_i)$ of the dimensions categorized in the same pattern-type. More specifically, threshold ThD_{type} for each pattern-type $type$ shall be calculated by:

1. For the N dimensions composing pattern-type $type$, calculate the absolute differences $\mathrm{Abs}(v1_i - v2_i)$ and sort them in the ascending order. Let $d_{type,k} = \{d_{type,0}, d_{type,1}, \ldots, d_{type,N-1}\}$ denote the values sorted in the ascending order. Note that $k = 0, \ldots, N-1$ is an index representing the order of the sorted values, and is different from dimension index i.

2. Threshold ThD_{type} shall be the element $d_{type,k}$ where $k = \text{floor}(0.333 \times N)$.

For example, for pattern-type D1 ($type = D1$, dimensions 33-148) composed of 116 dimensions ($N = 116$), let $d_{D1,k} = \{d_{D1,0}, d_{D1,1}, \ldots, d_{D1,115}\}$ denote the absolute differences $\text{Abs}(v1_i - v2_i), i = 33, \ldots, 148$ of the 116 dimensions composing type D1 sorted in ascending order. Threshold ThD_{D1} shall be the element $d_{D1,k}$ where $k = \text{floor}(0.333 \times 116) = \text{floor}(38.628) = 38$. The threshold for each pattern-type is explicitly shown in Table Amd4-1.

The vector of ternary values $\mathbf{x} = \{x_1, x_2, \cdots, x_{380}\}$ shall form the frame signature.

Table Amd4-1 — Thresholds for each pattern-type.

pattern-type $type$	dimensions	number of dimensions N	threshold
A1	1-20	20	$ThA_{A1} = a_{A1,6}$
A2	21-32	12	$ThA_{A2} = a_{A2,3}$
D1	33-148	116	$ThD_{D1} = d_{D1,38}$
D2	149-173	25	$ThD_{D2} = d_{D2,8}$
D3	174-209	36	$ThD_{D3} = d_{D3,11}$
D4	210-239	30	$ThD_{D4} = d_{D4,9}$
D5	240-301	62	$ThD_{D5} = d_{D5,20}$
D6	302-310	9	$ThD_{D6} = d_{D6,2}$
D7	311-360	50	$ThD_{D7} = d_{D7,16}$
D8	361-380	20	$ThD_{D8} = d_{D8,6}$

11.4.6 Confidence Extraction

This Subclause specifies the extraction process for the FrameConfidence element of the video signature descriptor.

The confidence measure of the frame shall be calculated based on the median value of the absolute differences $\text{Abs}(v1_i - v2_i)$ of the 348 difference element dimensions. More specifically, the confidence measure shall be calculated by:

1. For dimensions 33-380 (difference element dimensions), calculate the absolute differences $\text{Abs}(v1_i - v2_i)$ and sort them in the ascending order. Let $d_k = \{d_0, d_1, \ldots, d_{347}\}$ denote the values sorted in the ascending order, where k is an index representing the order of the sorted values.

2. Obtain the median value by taking the element d_k where $k = \text{floor}(0.5 \times 348) = 174$.

3. Calculate the confidence measure by converting the median value d_{174} into an integer value ranged [0,255] by the following:

$$\min(\mathrm{floor}(d_{174} \times 8), 255)$$

11.4.7 Word Extraction

This Subclause specifies the extraction process for the Word elements of the video signature descriptor.

A Word shall be used to refer to a compact representation of the complete frame signature. The set of all permissible Words are collectively referred to as the vocabulary. For two video frames, the distance between two corresponding Words is an approximation of the distance between the full frame descriptors.

To form each Word, an ordered subset of dimensions from the frame signature $\mathbf{x}$ shall be concatenated. The 380 dimensional ternary vector $\mathbf{x} = \{x_1, x_2, \cdots, x_{380}\}$ shall be used to form Q Ψ-dimensional vocabularies, with $Q = 5$ and $\Psi = 5$. The process of word formation can be seen as a projection from a 380-dimensional space to a 5-dimensional space.

Table Amd4-2 shows the ordered subset of dimensions from the descriptor $\mathbf{x}$ selected for each vocabulary. Each of these subsets of 5 dimensions shall be stored in 8 bits, according to the ternary packing scheme of Table E.1, to form a Word.

More specifically, each Word shall be calculated using the following equations.

$$\mathrm{Word}[0] = 81 \times x_{211} + 27 \times x_{218} + 9 \times x_{220} + 3 \times x_{275} + x_{335}$$

$$\mathrm{Word}[1] = 81 \times x_{45} + 27 \times x_{176} + 9 \times x_{234} + 3 \times x_{271} + x_{274}$$

$$\mathrm{Word}[2] = 81 \times x_{58} + 27 \times x_{71} + 9 \times x_{104} + 3 \times x_{238} + x_{270}$$

$$\mathrm{Word}[3] = 81 \times x_{101} + 27 \times x_{286} + 9 \times x_{296} + 3 \times x_{338} + x_{355}$$

$$\mathrm{Word}[4] = 81 \times x_{102} + 27 \times x_{103} + 9 \times x_{112} + 3 \times x_{276} + x_{297}$$

Table Amd4-2 — Ordered subset of dimensions for each word.

Word[j]	ordered subset of dimensions (from MSB (left) to LSB (right))
Word[0]	211, 218, 220, 275, 335
Word[1]	45, 176, 234, 271, 274
Word[2]	58, 71, 104, 238, 270
Word[3]	101, 286, 296, 338, 355
Word[4]	102, 103, 112, 276, 297

11.4.8 BagOfWords Extraction

This Subclause specifies the extraction process for the BagOfWords element of the video signature descriptor.

A BagOfWords representation shall be extracted for a temporal segment of 90 consecutive frames. For the temporal segment the frequency of occurrence of the words in each of the 5 vocabularies shall be calculated to obtain 5 BagOfWords. For each vocabulary, $j = 0,\ldots,Q-1$, a histogram $h[j]$ of the words found in the frame sequence shall be created. Such a histogram shows the frequency with which words appear in a frame sequence and shall be referred to as a bag-of-words. Then, each bag-of-words (histogram) $h[j]$ shall be binarised according to

$$BagOfWords[j][a] = \begin{cases} 1 & \text{if } h[j][a] > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $a = 0,\ldots,242$ refers to the bin indices of the histogram. The sets of consecutive frames in a BagOfWords shall come from intervals overlapping by 45 frames with the previous set. Therefore the first BagOfWords shall represent frames 0-89, the second frames 45-134, the third frames 90-179 and so on until the end of the sequence is reached. Note that the last BagOfWords will not, in general, represent 90 frames.

11.4.9 Decoding of Compressed Frame Signature Components

This Subclause describes the decoding process for the optional compressed frame signature field of the video signature descriptor. These fields are only present in the binary syntax. An example encoder is provided for informative purposes in 11.4.9.3.

The compressed temporal segments are aligned with the temporal segments of the BagOfWords. The first frame of a compressed segment is the first frame of a temporal segment represented by the BagOfWords. The relationship between these two elements is illustrated in Figure Amd4-2.

The CompressedSegment( ) part of binary representation is decoded into a set of FrameSignature as outlined below.

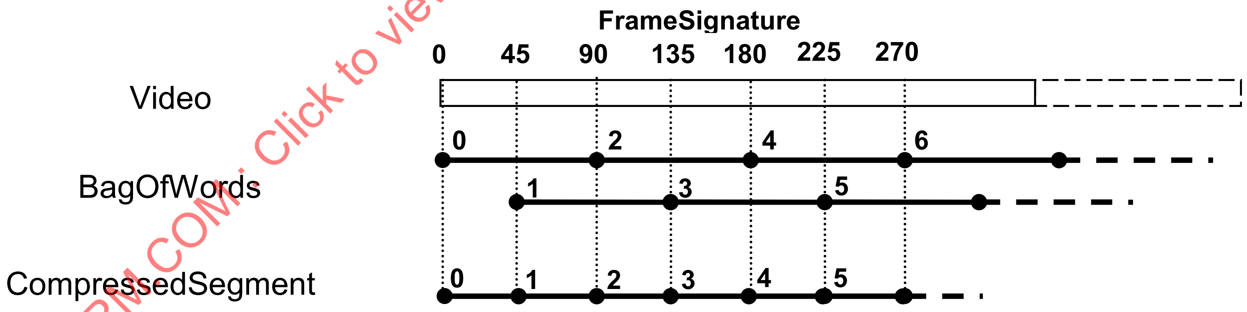

Figure Amd4-2 — Relationship between the BagOfWords and CompressedSegment temporal segments.

11.4.9.1 Decoding of PredictedPictures()

PredictedPictures() part of binary representation is a combination of run-length and entropy coding.

Zero run-length (ZeroRL) is represented with a variable number of bits corresponding to second-order Exp-Golomb codes, as in example for the first eight run-lengths in the Table Amd4-3.

Table Amd4-3 — Second-order Exp-Golomb codewords and the corresponding bit-lengths

Run length (n)	Code(number of bits)
0	000(3)
1	001(3)
2	010(3)
3	011(3)
4	10000(5)
5	10001(5)
6	10010(5)
7	10011(5)

The assignment of second order Exp-Golomb codes to integer values n is defined with the following steps (encoding):

1. add 4 to n, take binary representation of the result and remove the most significant bit. The binary length of such code is k,

2. write k-2 ones concluded with a zero, prefix that to the code generated in the previous step.

The assignment of integer values to second order Exp-Golomb codes are defined with the following steps (decoding):

1. read m ones concluded with a zero,

2. read an m+2 bit integer number, add 2^{m+2}-4 to obtain n.

The coded data is transformed into raw ternary data, where vb refers to a zero run-length represented with a variable number of bits (Table Amd4-3), while 1b refers to one-bit NonZeroSymbol, as depicted in Figure Amd4-3.

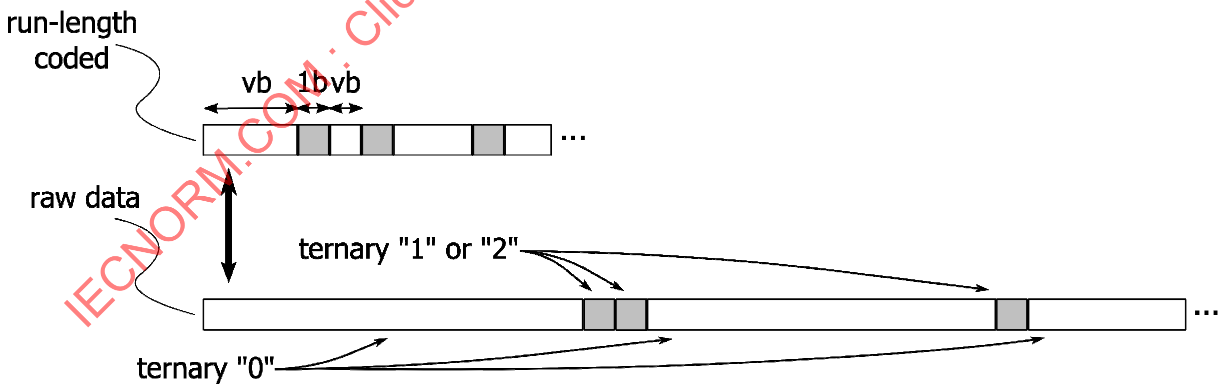

Figure Amd4-3 — Run-length and entropy coding

Such raw ternary data represents a vectorised prediction difference matrix of length GOPLengthm1 × 380.

11.4.9.2 Decoding of prediction difference matrix

The vectorised prediction difference matrix is transformed into a matrix form of GOPLengthm1 rows and 380 columns. To do this the first element of the vectorised prediction difference is placed in the top-left position of the matrix, and the matrix is populated column-wise in the left-to-right, top-to-bottom scan-order.

A key frame signature is a frame signature that is coded in a non-predictive way, with no reference to the temporally neighbouring frames, and is denoted as KP (key psicture). In the CompressedSegment field a KP is coded as a FrameSignature. Predicted frame signatures are coded in a predictive way using the temporally preceding frames, and are denoted as PP (predicted picture). The i-th element of a FrameSignature at time instance t is denoted as $x_{i,t}$, and corresponds to a prediction difference (PP) element $\hat{x}_{i,t}$.

The prediction difference matrix is transformed into GOPLengthm1 number of FrameSignature by computing modulo 3 subtraction between elements in the current PP ($\hat{x}_{i,t}$) and the previously reconstructed frame signature elements $x_{i,t-1}$. For the first row of the prediction difference matrix, the first preceding KP is used as a reference.

The operation of the modulo 3 subtraction at the decoder is defined as:

$$x_{i,t} = (x_{i,t-1} - \hat{x}_{i,t})\%3 .$$

It reverses the following operation of modulo 3 subtraction, performed at the encoder:

$$\hat{x}_{i,t} = (x_{i,t-1} - x_{i,t})\%3 .$$

A group-of-pictures (GOP) is defined to be a set of frame signatures between two KPs, including the temporally first KP and all PPs before the next KP. A GOP therefore consists of a key frame and zero or more predicted frames.

The structure of a GOP as coded, composed of a KP FrameSignature and GOPLengthm1 number of PP FrameSignatures, transformed into a prediction difference matrix, is depicted in Figure Amd4-4.

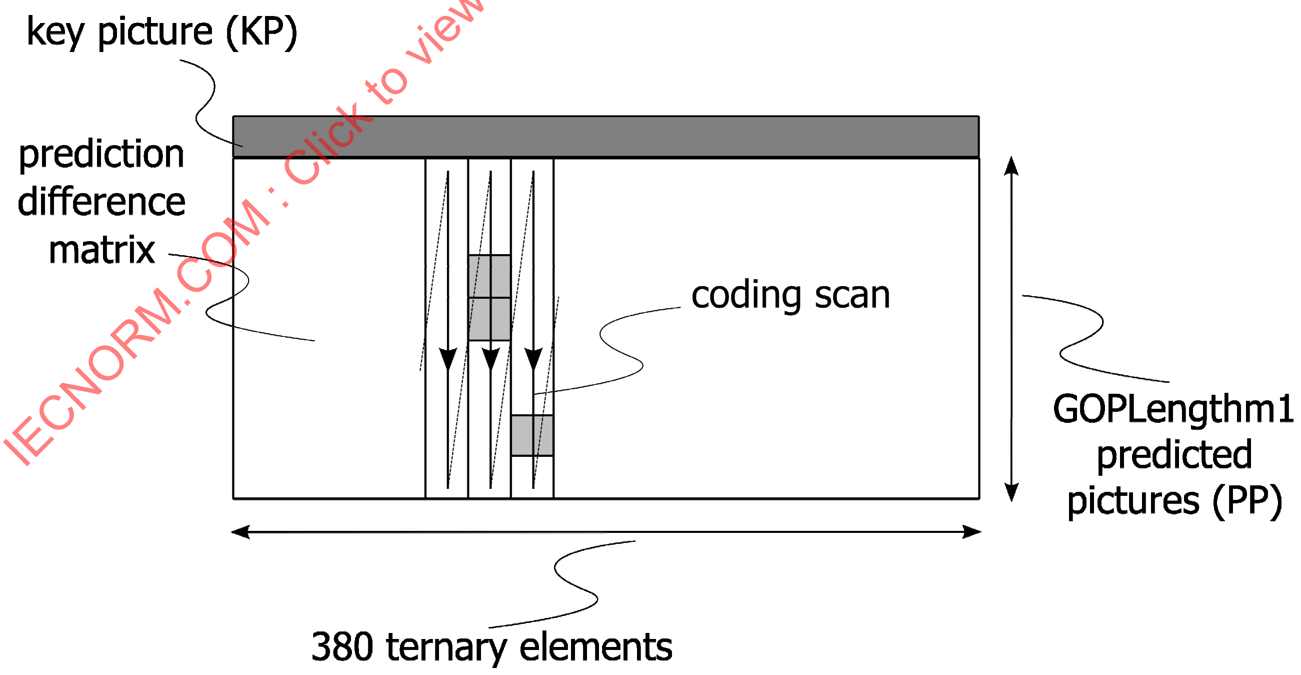
key picture (KP)
prediction difference matrix
coding scan
GOPLengthm1 predicted pictures (PP)
380 ternary elements

Figure Amd4-4 — Coding of a GOP

11.4.9.3 Encode

An example implementation of an encoder is:

1. Encode the frame signatures of each 45 frame temporal segment independently of other temporal segments.

2. Encode the first frame signature of a segment as a KP.

3. Move forward one frame. If the current frame is the last frame in the segment, go to step 6.

4. Using a selected criteria decide whether to encode the current frame signature as a KP. These criteria could, for instance, be correlation between the current frame's frame signature and the frame signature of the previous frame. A correlation test could consist of comparing the number of zero elements in a prediction difference vector to a pre-defined threshold, and if this number is lower than the threshold then this points to low correlation between the current and the previous frame, and KP is inserted instead of using prediction. This part of the encoder offers flexibility in the number of KP that are used in a segment.

5. If in step 4 a KP was not inserted then store the prediction difference computed between the current frame signature and the previous in the prediction difference matrix, and go back to step 3, otherwise go to step 6. The prediction difference is obtained by a modulo 3 subtraction between the temporally preceding and the current frame signatures. The modulo 3 difference operation between ternary elements of the signatures results in the three ternary symbols: 0,1 and 2.

6. Convert the prediction difference matrix to a vector (vectorisation step) by concatenating the columns of the prediction difference matrix.

7. The prediction difference vector is thus composed of ternary symbols, which are coded differently:

   a. ternary 1 and 2 - coded with one bit each, binary 0 for ternary 1 and binary 1 for ternary 2. After each ternary symbol, a run-length codeword for a run of zeros is inserted. If no zeros follow, a codeword for zero length is used.

   b. ternary 0 - the ternary symbol is implicitly encoded as after each non-zero ternary symbol a zero run must follow, unless the non-zero ternary symbol is last in the prediction difference matrix. The zero-run length is entropy coded using Exp-Golomb codewords.

8. Encode all signatures from the previous KP to the last frame before the current frame as one GOP.

9. At the beginning of a GOP insert the length of a GOP.

Add the following Annexes:

Annex E
(normative)

Ternary value packing

Table E.1 — Packing ternary values.

Ternary Value					Representation	
x_{5j-4}	x_{5j-3}	x_{5j-2}	x_{5j-1}	x_{5j}	b_j	Binary
0	0	0	0	0	0	00000000
0	0	0	0	1	1	00000001
0	0	0	0	2	2	00000010
0	0	0	1	0	3	00000011
0	0	0	1	1	4	00000100
0	0	0	1	2	5	00000101
0	0	0	2	0	6	00000110
0	0	0	2	1	7	00000111
0	0	0	2	2	8	00001000
0	0	1	0	0	9	00001001
0	0	1	0	1	10	00001010
0	0	1	0	2	11	00001011
0	0	1	1	0	12	00001100
0	0	1	1	1	13	00001101
0	0	1	1	2	14	00001110
0	0	1	2	0	15	00001111
0	0	1	2	1	16	00010000
0	0	1	2	2	17	00010001
0	0	2	0	0	18	00010010
0	0	2	0	1	19	00010011
0	0	2	0	2	20	00010100
0	0	2	1	0	21	00010101
0	0	2	1	1	22	00010110
0	0	2	1	2	23	00010111
0	0	2	2	0	24	00011000
0	0	2	2	1	25	00011001
0	0	2	2	2	26	00011010
0	1	0	0	0	27	00011011
0	1	0	0	1	28	00011100
0	1	0	0	2	29	00011101
0	1	0	1	0	30	00011110
0	1	0	1	1	31	00011111
0	1	0	1	2	32	00100000
0	1	0	2	0	33	00100001
0	1	0	2	1	34	00100010
0	1	0	2	2	35	00100011
0	1	1	0	0	36	00100100
0	1	1	0	1	37	00100101
0	1	1	0	2	38	00100110
0	1	1	1	0	39	00100111

0	1	1	1	1	40	00101000
0	1	1	1	2	41	00101001
0	1	1	2	0	42	00101010
0	1	1	2	1	43	00101011
0	1	1	2	2	44	00101100
0	1	2	0	0	45	00101101
0	1	2	0	1	46	00101110
0	1	2	0	2	47	00101111
0	1	2	1	0	48	00110000
0	1	2	1	1	49	00110001
0	1	2	1	2	50	00110010
0	1	2	2	0	51	00110011
0	1	2	2	1	52	00110100
0	1	2	2	2	53	00110101
0	2	0	0	0	54	00110110
0	2	0	0	1	55	00110111
0	2	0	0	2	56	00111000
0	2	0	1	0	57	00111001
0	2	0	1	1	58	00111010
0	2	0	1	2	59	00111011
0	2	0	2	0	60	00111100
0	2	0	2	1	61	00111101
0	2	0	2	2	62	00111110
0	2	1	0	0	63	00111111
0	2	1	0	1	64	01000000
0	2	1	0	2	65	01000001
0	2	1	1	0	66	01000010
0	2	1	1	1	67	01000011
0	2	1	1	2	68	01000100
0	2	1	2	0	69	01000101
0	2	1	2	1	70	01000110
0	2	1	2	2	71	01000111
0	2	2	0	0	72	01001000
0	2	2	0	1	73	01001001
0	2	2	0	2	74	01001010
0	2	2	1	0	75	01001011
0	2	2	1	1	76	01001100
0	2	2	1	2	77	01001101
0	2	2	2	0	78	01001110
0	2	2	2	1	79	01001111
0	2	2	2	2	80	01010000
1	0	0	0	0	81	01010001
1	0	0	0	1	82	01010010
1	0	0	0	2	83	01010011
1	0	0	1	0	84	01010100
1	0	0	1	1	85	01010101
1	0	0	1	2	86	01010110
1	0	0	2	0	87	01010111
1	0	0	2	1	88	01011000
1	0	0	2	2	89	01011001
1	0	1	0	0	90	01011010
1	0	1	0	1	91	01011011
1	0	1	0	2	92	01011100
1	0	1	1	0	93	01011101

1	0	1	1	1	94	01011110
1	0	1	1	2	95	01011111
1	0	1	2	0	96	01100000
1	0	1	2	1	97	01100001
1	0	1	2	2	98	01100010
1	0	2	0	0	99	01100011
1	0	2	0	1	100	01100100
1	0	2	0	2	101	01100101
1	0	2	1	0	102	01100110
1	0	2	1	1	103	01100111
1	0	2	1	2	104	01101000
1	0	2	2	0	105	01101001
1	0	2	2	1	106	01101010
1	0	2	2	2	107	01101011
1	1	0	0	0	108	01101100
1	1	0	0	1	109	01101101
1	1	0	0	2	110	01101110
1	1	0	1	0	111	01101111
1	1	0	1	1	112	01110000
1	1	0	1	2	113	01110001
1	1	0	2	0	114	01110010
1	1	0	2	1	115	01110011
1	1	0	2	2	116	01110100
1	1	1	0	0	117	01110101
1	1	1	0	1	118	01110110
1	1	1	0	2	119	01110111
1	1	1	1	0	120	01111000
1	1	1	1	1	121	01111001
1	1	1	1	2	122	01111010
1	1	1	2	0	123	01111011
1	1	1	2	1	124	01111100
1	1	1	2	2	125	01111101
1	1	2	0	0	126	01111110
1	1	2	0	1	127	01111111
1	1	2	0	2	128	10000000
1	1	2	1	0	129	10000001
1	1	2	1	1	130	10000010
1	1	2	1	2	131	10000011
1	1	2	2	0	132	10000100
1	1	2	2	1	133	10000101
1	1	2	2	2	134	10000110
1	2	0	0	0	135	10000111
1	2	0	0	1	136	10001000
1	2	0	0	2	137	10001001
1	2	0	1	0	138	10001010
1	2	0	1	1	139	10001011
1	2	0	1	2	140	10001100
1	2	0	2	0	141	10001101
1	2	0	2	1	142	10001110
1	2	0	2	2	143	10001111
1	2	1	0	0	144	10010000
1	2	1	0	1	145	10010001
1	2	1	0	2	146	10010010
1	2	1	1	0	147	10010011

1	2	1	1	1	148	10010100
1	2	1	1	2	149	10010101
1	2	1	2	0	150	10010110
1	2	1	2	1	151	10010111
1	2	1	2	2	152	10011000
1	2	2	0	0	153	10011001
1	2	2	0	1	154	10011010
1	2	2	0	2	155	10011011
1	2	2	1	0	156	10011100
1	2	2	1	1	157	10011101
1	2	2	1	2	158	10011110
1	2	2	2	0	159	10011111
1	2	2	2	1	160	10100000
1	2	2	2	2	161	10100001
2	0	0	0	0	162	10100010
2	0	0	0	1	163	10100011
2	0	0	0	2	164	10100100
2	0	0	1	0	165	10100101
2	0	0	1	1	166	10100110
2	0	0	1	2	167	10100111
2	0	0	2	0	168	10101000
2	0	0	2	1	169	10101001
2	0	0	2	2	170	10101010
2	0	1	0	0	171	10101011
2	0	1	0	1	172	10101100
2	0	1	0	2	173	10101101
2	0	1	1	0	174	10101110
2	0	1	1	1	175	10101111
2	0	1	1	2	176	10110000
2	0	1	2	0	177	10110001
2	0	1	2	1	178	10110010
2	0	1	2	2	179	10110011
2	0	2	0	0	180	10110100
2	0	2	0	1	181	10110101
2	0	2	0	2	182	10110110
2	0	2	1	0	183	10110111
2	0	2	1	1	184	10111000
2	0	2	1	2	185	10111001
2	0	2	2	0	186	10111010
2	0	2	2	1	187	10111011
2	0	2	2	2	188	10111100
2	1	0	0	0	189	10111101
2	1	0	0	1	190	10111110
2	1	0	0	2	191	10111111
2	1	0	1	0	192	11000000
2	1	0	1	1	193	11000001
2	1	0	1	2	194	11000010
2	1	0	2	0	195	11000011
2	1	0	2	1	196	11000100
2	1	0	2	2	197	11000101
2	1	1	0	0	198	11000110
2	1	1	0	1	199	11000111
2	1	1	0	2	200	11001000
2	1	1	1	0	201	11001001

2	1	1	1	1	202	11001010
2	1	1	1	2	203	11001011
2	1	1	2	0	204	11001100
2	1	1	2	1	205	11001101
2	1	1	2	2	206	11001110
2	1	2	0	0	207	11001111
2	1	2	0	1	208	11010000
2	1	2	0	2	209	11010001
2	1	2	1	0	210	11010010
2	1	2	1	1	211	11010011
2	1	2	1	2	212	11010100
2	1	2	2	0	213	11010101
2	1	2	2	1	214	11010110
2	1	2	2	2	215	11010111
2	2	0	0	0	216	11011000
2	2	0	0	1	217	11011001
2	2	0	0	2	218	11011010
2	2	0	1	0	219	11011011
2	2	0	1	1	220	11011100
2	2	0	1	2	221	11011101
2	2	0	2	0	222	11011110
2	2	0	2	1	223	11011111
2	2	0	2	2	224	11100000
2	2	1	0	0	225	11100001
2	2	1	0	1	226	11100010
2	2	1	0	2	227	11100011
2	2	1	1	0	228	11100100
2	2	1	1	1	229	11100101
2	2	1	1	2	230	11100110
2	2	1	2	0	231	11100111
2	2	1	2	1	232	11101000
2	2	1	2	2	233	11101001
2	2	2	0	0	234	11101010
2	2	2	0	1	235	11101011
2	2	2	0	2	236	11101100
2	2	2	1	0	237	11101101
2	2	2	1	1	238	11101110
2	2	2	1	2	239	11101111
2	2	2	2	0	240	11110000
2	2	2	2	1	241	11110001
2	2	2	2	2	242	11110010
Reserved					243-255	11110011 - 11111111

Annex F
(normative)

Descriptor dimension definitions

Table F.1 — Descriptor dimension definitions of the average element dimensions.

Dimension	Pattern-type	Indices of sub-region 1
1	A1	0-231
2	A1	8-239
3	A1	256-487
4	A1	264-495
5	A1	16-247
6	A1	24-255
7	A1	272-503
8	A1	280-511
9	A1	512-743
10	A1	520-751
11	A1	768-999
12	A1	776-1007
13	A1	528-759
14	A1	536-767
15	A1	784-1015
16	A1	792-1023
17	A1	0-495
18	A1	16-511
19	A1	512-1007
20	A1	528-1023
21	A2	66-297
22	A2	76-307
23	A2	86-317
24	A2	386-617
25	A2	396-627
26	A2	406-637
27	A2	706-937
28	A2	716-947
29	A2	726-957
30	A2	297-726
31	A2	198-825
32	A2	99-924

Table F.2 — Descriptor dimension definitions the difference element dimensions.

Dimension	Pattern-type	Indices of sub-region 1	Indices of sub-region 2
33	D1	0-97	2-99
34	D1	4-39	68-103
35	D1	192-227	128-163
36	D1	134-231	132-229
37	D1	8-105	10-107
38	D1	12-47	76-111
39	D1	200-235	136-171
40	D1	142-239	140-237
41	D1	256-353	258-355
42	D1	260-295	324-359
43	D1	448-483	384-419
44	D1	390-487	388-485
45	D1	264-361	266-363
46	D1	268-303	332-367
47	D1	456-491	392-427
48	D1	398-495	396-493
49	D1	16-51	80-115
50	D1	22-119	20-117
51	D1	144-241	146-243
52	D1	212-247	148-183
53	D1	24-59	88-123
54	D1	30-127	28-125
55	D1	152-249	154-251
56	D1	220-255	156-191
57	D1	272-307	336-371
58	D1	278-375	276-373
59	D1	400-497	402-499
60	D1	468-503	404-439
61	D1	280-315	344-379
62	D1	286-383	284-381
63	D1	408-505	410-507
64	D1	476-511	412-447
65	D1	512-547	576-611
66	D1	518-615	516-613
67	D1	640-737	642-739
68	D1	708-743	644-679
69	D1	520-555	584-619
70	D1	526-623	524-621
71	D1	648-745	650-747
72	D1	716-751	652-687
73	D1	768-803	832-867
74	D1	774-871	772-869
75	D1	896-993	898-995
76	D1	964-999	900-935
77	D1	776-811	840-875
78	D1	782-879	780-877
79	D1	904-1001	906-1003
80	D1	972-1007	908-943
81	D1	528-625	530-627
82	D1	532-567	596-631
83	D1	720-755	656-691
84	D1	662-759	660-757
85	D1	536-633	538-635
86	D1	540-575	604-639

87	D1	728-763	664-699
88	D1	670-767	668-765
89	D1	784-881	786-883
90	D1	788-823	852-887
91	D1	976-1011	912-947
92	D1	918-1015	916-1013
93	D1	792-889	794-891
94	D1	796-831	860-895
95	D1	984-1019	920-955
96	D1	926-1023	924-1021
97	D1	66-163	68-165
98	D1	70-105	134-169
99	D1	258-293	194-229
100	D1	200-297	198-295
101	D1	76-173	78-175
102	D1	80-115	144-179
103	D1	268-303	204-239
104	D1	210-307	208-305
105	D1	86-183	88-185
106	D1	90-125	154-189
107	D1	278-313	214-249
108	D1	220-317	218-315
109	D1	386-483	388-485
110	D1	390-425	454-489
111	D1	578-613	514-549
112	D1	520-617	518-615
113	D1	396-431	460-495
114	D1	400-435	464-499
115	D1	588-623	524-559
116	D1	592-627	528-563
117	D1	406-503	408-505
118	D1	410-445	474-509
119	D1	598-633	534-569
120	D1	540-637	538-635
121	D1	706-803	708-805
122	D1	710-745	774-809
123	D1	898-933	834-869
124	D1	840-937	838-935
125	D1	716-813	718-815
126	D1	720-755	784-819
127	D1	908-943	844-879
128	D1	850-947	848-945
129	D1	726-823	728-825
130	D1	730-765	794-829
131	D1	918-953	854-889
132	D1	860-957	858-955
133	D1	231-266	295-330
134	D1	235-332	237-334
135	D1	359-456	361-458
136	D1	363-398	427-462
137	D1	241-276	305-340
138	D1	245-342	247-344
139	D1	369-466	371-468
140	D1	373-408	437-472
141	D1	551-586	615-650
142	D1	555-652	557-654

143	D1	679-776	681-778
144	D1	683-718	747-782
145	D1	561-596	625-660
146	D1	565-662	567-664
147	D1	689-786	691-788
148	D1	693-728	757-792
149	D2	0-99 | 132-231	4-103 | 128-227
150	D2	8-107 | 140-239	12-111 | 136-235
151	D2	16-115 | 148-247	20-119 | 144-243
152	D2	24-123 | 156-255	28-127 | 152-251
153	D2	256-355 | 388-487	260-359 | 384-483
154	D2	264-363 | 396-495	268-367 | 392-491
155	D2	272-371 | 404-503	276-375 | 400-499
156	D2	280-379 | 412-511	284-383 | 408-507
157	D2	512-611 | 644-743	516-615 | 640-739
158	D2	520-619 | 652-751	524-623 | 648-747
159	D2	528-627 | 660-759	532-631 | 656-755
160	D2	536-635 | 668-767	540-639 | 664-763
161	D2	768-867 | 900-999	772-871 | 896-995
162	D2	776-875 | 908-1007	780-879 | 904-1003
163	D2	784-883 | 916-1015	788-887 | 912-1011
164	D2	792-891 | 924-1023	796-895 | 920-1019
165	D2	132-231 | 264-363	136-235 | 260-359
166	D2	140-239 | 272-371	144-243 | 268-367
167	D2	148-247 | 280-379	152-251 | 276-375
168	D2	388-487 | 520-619	392-491 | 516-615
169	D2	396-495 | 528-627	400-499 | 524-623
170	D2	404-503 | 536-635	408-507 | 532-631
171	D2	644-743 | 776-875	648-747 | 772-871
172	D2	652-751 | 784-883	656-755 | 780-879
173	D2	660-759 | 792-891	664-763 | 788-887
174	D3	33-330	43-340
175	D3	33-330	53-350
176	D3	33-330	353-650
177	D3	33-330	363-660
178	D3	33-330	373-670
179	D3	33-330	673-970
180	D3	33-330	683-980
181	D3	33-330	693-990
182	D3	43-340	53-350
183	D3	43-340	353-650
184	D3	43-340	363-660
185	D3	43-340	373-670
186	D3	43-340	673-970
187	D3	43-340	683-980
188	D3	43-340	693-990
189	D3	53-350	353-650
190	D3	53-350	363-660
191	D3	53-350	373-670
192	D3	53-350	673-970
193	D3	53-350	683-980
194	D3	53-350	693-990
195	D3	353-650	363-660
196	D3	353-650	373-670
197	D3	353-650	673-970
198	D3	353-650	683-980

199	D3	353-650	693-990
200	D3	363-660	373-670
201	D3	363-660	673-970
202	D3	363-660	683-980
203	D3	363-660	693-990
204	D3	373-670	673-970
205	D3	373-670	683-980
206	D3	373-670	693-990
207	D3	673-970	683-980
208	D3	673-970	693-990
209	D3	683-980	693-990
210	D4	423-588	435-600
211	D4	237-402	621-786
212	D4	231-396	627-792
213	D4	243-408	615-780
214	D4	237-402	435-600
215	D4	435-600	621-786
216	D4	621-786	423-588
217	D4	423-588	237-402
218	D4	231-396	243-408
219	D4	243-408	627-792
220	D4	627-792	615-780
221	D4	615-780	231-396
222	D4	429-594	45-210
223	D4	429-594	441-606
224	D4	429-594	813-978
225	D4	429-594	417-582
226	D4	45-210	813-978
227	D4	417-582	441-606
228	D4	39-204	51-216
229	D4	807-972	819-984
230	D4	225-390	609-774
231	D4	249-414	633-798
232	D4	39-204	225-390
233	D4	51-216	249-414
234	D4	633-798	819-984
235	D4	609-774	807-972
236	D4	33-198	57-222
237	D4	57-222	825-990
238	D4	825-990	801-966
239	D4	801-966	33-198
240	D5	33-106 | 129-227 | 136-234 | 257-330	132-231
241	D5	43-116 | 139-237 | 146-244 | 267-340	142-241
242	D5	53-126 | 149-247 | 156-254 | 277-350	152-251
243	D5	353-426 | 449-547 | 456-554 | 577-650	452-551
244	D5	363-436 | 459-557 | 466-564 | 587-660	462-561
245	D5	373-446 | 469-567 | 476-574 | 597-670	472-571
246	D5	673-746 | 769-867 | 776-874 | 897-970	772-871
247	D5	683-756 | 779-877 | 786-884 | 907-980	782-881
248	D5	693-766 | 789-887 | 796-894 | 917-990	792-891
249	D5	198-271 | 294-392 | 301-399 | 422-495	297-396
250	D5	208-281 | 304-402 | 311-409 | 432-505	307-406
251	D5	518-591 | 614-712 | 621-719 | 742-815	617-716
252	D5	528-601 | 624-722 | 631-729 | 752-825	627-726
253	D5	38-111 | 134-232 | 141-239 | 262-335	137-236
254	D5	48-121 | 144-242 | 151-249 | 272-345	147-246