INTERNATIONAL STANDARD

ISO/IEC 15938-15

First edition
2019-07

Information technology — Multimedia content description interface —

Part 15: **Compact descriptors for video analysis**

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

Partie 15: Descripteurs compacts pour analyse de vidéo

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

Reference number
ISO/IEC 15938-15:2019(E)

© ISO/IEC 2019

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

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.

The procedures used to develop this document and those intended for its further maintenance are specified in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 15938 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at http://www.iso.org/members.html.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

Introduction

ISO/IEC 15938 (all parts), also known as "Multimedia content description interface", provides a standardized set of technologies for describing multimedia content. It addresses a broad spectrum of multimedia applications and requirements by providing a metadata system for describing the features of multimedia content.

The following are specified in this ISO/IEC 15938 (all parts):

Description schemes (DS) describe entities or relationships pertaining to multimedia content. Description schemes specify the structure and semantics of their components, which may be description schemes, descriptors or datatypes.

Descriptors (D) describe features, attributes or groups of attributes of multimedia content.

Datatypes are the basic reusable datatypes employed by description schemes and descriptors.

Description definition language (DDL) defines description schemes, descriptors and datatypes by specifying their syntax, and allows their extension.

Systems tools support delivery of descriptions, multiplexing of descriptions with multimedia content, synchronization, file format, etc.

The ISO/IEC 15938 series is subdivided into 15 published parts with further parts in development:

— **Part 1: Systems**: specifies the tools for preparing descriptions for efficient transport and storage, compressing descriptions, and allowing synchronization between content and descriptions.

— **Part 2: Description definition language**: specifies the language for defining the series set of description tools (DSs, Ds and datatypes) and for defining new description tools.

— **Part 3: Visual**: specifies the description tools pertaining to visual content.

— **Part 4: Audio**: specifies the description tools pertaining to audio content.

— **Part 5: Multimedia description schemes**: specifies the generic description tools pertaining to multimedia including audio and visual content.

— **Part 6: Reference software**: provides a software implementation of the series.

— **Part 7: Conformance testing**: specifies the guidelines and procedures for testing conformance of implementations of the series.

— **Part 8: Extraction and use of MPEG-7 descriptions**: provides guidelines and examples of the extraction and use of descriptions.

— **Part 9: Profiles and levels**: provides guidelines and standard profiles.

— **Part 10: Schema definition**: specifies the schema using description definition language.

— **Part 11: MPEG-7 profile schemas**: listing of profile schemas using description definition language.

— **Part 12: Query format**: contains the tools of the MPEG query format (MPQF).

— **Part 13: Compact descriptors for visual search**: specifies an image description tool for visual search applications.

— **Part 14: Reference software, conformance and usage guidelines for compact descriptors for visual search**: provides the reference software and guidelines, specifies the conformance testing.

— **Part 15: Compact descriptors for video analysis (this document)**: specifies a video description tool designed to enable efficient and interoperable video analysis applications, allowing visual content matching in videos.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

The structure of this document is as follows:

— Clause 5 specifies the binary representation syntax and descriptor component semantics for a CDVA descriptor.

— Clause 6 specifies the extraction and encoding process for a CDVA descriptor.

— Annex A specifies recommended values for the parameters of the encoding process of Clause 6.

— Annex B specifies parameters and a neural network model of the deep feature extraction process.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right. The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from:

Joanneum Research Forschungagesellshaft mbH

Leonhardstrasse 59

8010 Graz, Austria

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

Information technology — Multimedia content description interface —

Part 15: Compact descriptors for video analysis

1 Scope

This document addresses descriptor technology for search and retrieval applications, i.e. for visual content matching in video. Visual content matching includes matching of views of large and small objects and scenes, with robustness to partial occlusions as well as changes in vantage point, camera parameters and lighting conditions. The objects of interest comprise planar or non-planar, rigid or partially rigid, textured or partially textured objects, but exclude the identification of people and faces. The databases can be large, for example broadcast archives or videos available on the internet. Such applications thus require video descriptors that enable matching with smaller descriptor sizes and shorter runtimes as compared to application enabled by single-frame (still image) descriptors (e.g. CVDS, ISO/IEC 15938-13) in the video domain.

Compact descriptors for video analysis for search and retrieval applications:

— enable design of interoperable object instance search applications;

— minimize the size of video descriptors;

— ensure high matching performances of objects (in terms of accuracy and complexity);

— enable efficient implementation of those functionalities on professional or embedded systems.

This document provides a complementary tool to the suite of existing standards, such as ISO/IEC 15938-13.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15938-13:2015, *Information technology — Multimedia content description interface — Part 13: Compact descriptors for visual search*

Neural Network Exchange Format, The Khronos Group, Version 1.0, Revision 3, 2018-06-13.

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax,* Jan. 2005.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

3.1
image descriptor
descriptor extracted from a single *key frame* (3.6) sampled from the *input video* (3.8), which contains *global descriptor* (3.2), *local feature descriptor* (3.3) and *deep feature descriptor* (3.4)

Note 1 to entry: Image descriptors are encoded as described in Clause 6.

3.2
global descriptor
aggregation of local feature descriptors into a compact representation of the *image* (3.5)

Note 1 to entry: The aggregation is as described in subclause 6.1.2.

3.3
local feature descriptor
descriptor of a local region, extracted around an interest point (a point in an *image* (3.5) showing detection stability under local and global perturbations in the image domain, including perspective transformations, changes in image scale, and illumination variations)

Note 1 to entry: The extraction is as described in subclause 6.1.3.

3.4
deep feature descriptor
feature descriptor extracted from a layer of a trained convolutional neural network

Note 1 to entry: The extraction is as described in subclause 6.1.4.

3.5
image
input *key frame* (3.6) to the *image descriptor* (3.1) encoder

Note 1 to entry: The image is as described in Clause 6.

3.6
key frame
frame extracted from the *input video segment* (3.7) by the frame difference process of colour histogram

Note 1 to entry: The extraction is as described in subclause 6.2.

3.7
input video segment
time range (temporal segment) of a video and from which a descriptor is extracted

3.8
input video
image sequence to be processed by the system containing a number of *input video segment(s)* (3.7) to CDVA extraction process

Note 1 to entry: Input video is as described in Clause 6.

3.9
segment descriptor
descriptor extracted from the sampled *key frames* (3.6) of an *input video segment* (3.7)

Note 1 to entry: Segment descriptors are encoded as described in Clause 6. They are contructed from the *image descriptors* (3.1) of the sampled key frames of the input video segment.

3.10
representative frame
frame of an *input video segment* (3.7) for which an uncompressed descriptor is represented and which is used as the basis for differential encoding

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

3.11
pixel
indexable element on an integer grid of the original image or the converted image, comprising spatial coordinates, a luminance value and (optional) chrominance values

4 Abbreviated terms, operators, mnemonics, functions and symbols

4.1 General

The mathematical symbols used in this document are similar to those used in the C programming language. However, integer divisions with truncation and rounding are specifically defined. Numbering and counting loops generally begin with zero.

4.2 Abbreviated terms

ABAC	adaptive binary arithmetic coding
CDVA	compact descriptors for visual analysis as defined by this document
CDVS	compact descriptors for visual search as defined by ISO/IEC 15938-13
CNN	convolutional neural network
MPEG-7	ISO/IEC 15938 (all parts)
NIP	nested invariance pooling
NN	neural network
NNEF	neural network exchange format as defined by the Khronos specification referenced in Clause 2
PCA	principal component analysis
RGB	red-green-blue colour space
ROI	region of interest
SCFV	scalable compressed fisher vector
URI	uniform resource identifier as defined by RFC 3986
XOR	binary exclusive OR operation

4.3 Arithmetic operators

+	addition
-	subtraction (as a binary operator) or negation (as a unary operator)
++	increment, i.e. x++ is equivalent to x = x+1
--	decrement, i.e. x-- is equivalent to x = x–1
*	multiplication
×	multiplication

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

^	power
/	integer division with truncation of the result towards zero For example, 7/4 and −7/−4 are truncated to 1, and −7/4 and 7/−4 are truncated to −1
//	integer division with rounding to the nearest integer; half-integer values are rounded away from zero unless otherwise specified For example, 3//2 is rounded to 2, and −3//2 is rounded to −2.
÷	indicates division in mathematical equations where no rounding is intended
%	modulus operator, defined only for positive numbers
ceil	minimum integer number greater than or equal to the given floating point number
sqrt	square root

4.4 Logical operators

||	logical OR
&&	logical AND
!	logical NOT
⊕	bit-wise difference (XOR) operator

4.5 Relational operators

>	greater than
>=	greater than or equal to
≥	greater than or equal to
<	less than
<=	less than or equal to
≤	less than or equal to
==	equal to
!=	not equal to

4.6 Bitwise operators

|	OR
&	AND

4.7 Interval specification

[a;b]	inclusive range from a to b

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

4.8 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bitstream.

bslbf	bit string, left bit first, where “left” is the order in which bits are written in this document Bit strings are generally written as a string of 1s and 0s within single quote marks, e.g. ‘1000 0001’. Blanks within a bit string are for ease of reading and have no significance. For convenience, large strings are occasionally written in hexadecimal, in which case conversion to a binary in the conventional manner will yield the value of the bit string. Thus, the left-most hexadecimal digit is first and in each hexadecimal digit the most significant of the four digits is first.
uimsbf	unsigned integer, most significant bit first
vlclbf	variable length code, left bit first, where “left” refers to the order in which the VLC codes are written in this document The byte order of multibyte words is most significant byte first.

4.9 Functions

$\text{argmax}_i()$	maximum value in argument list
$\text{argmin}_i()$	minimum value in argument list
δ_g	distance function for global descriptors
δ_l	distance function for local descriptors
$\sum_{i=a}^{i<b} f(i)$	summation of $f(i)$ with i taking integer values from a up to, but not including b
L0 norm	$\text{L0}(\text{x,y}) = \lVert x-y \rVert_0 = \sum_i \delta(x_i - y_i)$, where $\delta(\text{a}) = \begin{cases} 1(a!=0) \\ 0(a=0) \end{cases}$
L1 norm	$\text{L1}(\text{x,y}) = \lVert x-y \rVert_1 = \sum_i \lvert x_i - y_i \rvert$
L2 norm	$\text{L2}(\text{x,y}) = \lVert x-y \rVert_2 = \text{sqrt}\left(\sum_i (x_i - y_i)^2\right)$
Euclidean distance	$\text{D}(x,y) = \text{sqrt}\left(\sum_i (x_i - y_i)^2\right)$
hist(I,C)	histogram of image I for colour channel C

4.10 Symbols

β	selection priority of local feature
c	number of channels of feature map (dimension of descriptor extracted from CNN)
Δ_k	deep feature descriptor for frame k
D_k	binarized deep feature descriptor for frame k
f	feature vector of local descriptor

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

$\boldsymbol{G}$	set of global descriptors
G_k	global descriptor of frame k
γ_k	result of pooling operation for feature map for frame k
h	feature map height
I_k	RGB image of frame k
k	key frame index
m	feature map index
n_k	number of key frames in a video segment
n_l^k	number of local descriptors of frame k
n_r	number of rotation transformations
n_s	number of scale transformations
ρ	representative frame of video segment
p_r, p_s, p_t	statistical moment for pooling operation
P_s	scale invariance pooling
P_r	rotation invariance pooling
P_t	translation invariance pooling
q	quantization function
$\boldsymbol{L}^k$	set of local descriptors of frame k
L	list of local feature descriptors of a video segment
l_i^k	ith local descriptor of frame k
θ_l	threshold for local descriptor distance
w	feature map width
x	horizontal image coordinate
y	vertical image coordinate

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

5 CDVA bitstream syntax

5.1 CDVA descriptor

5.1.1 Binary representation syntax

CDVADescriptor {	**Number of bits**	**Mnemonics**
CDVAHeader	≥80	vlclbf
for (i=0; i<NrSegments; i++) {		

© ISO/IEC 2019 – All rights reserved

CDVADescriptor {	Number of bits	Mnemonics
SegmentHeader	136	bslbf
GlobalDescriptor	≥40	vlclbf
if (LocalFeatureDescriptorPresent) {		
LocalDescriptor	≥1	vlclbf
LocalDescriptorLocations	≥1	vlclbf
}		
if (DeepFeatureDescriptorPresent) {		
DeepFeatureDescriptor	≥1	vlclbf
}		
}		
}		

5.1.2 Descriptor component semantics

CDVAHeader

A bitstream header as defined in subclause 5.2. It also contains information whether the corresponding descriptior components are present, i.e. it defines LocalFeatureDescriptorPresent and DeepFeatureDescriptorPresent.

NrSegments

Number of the segments in the input video.

SegmentHeader

Header for each segment as defined in subclause 5.3. The segment header contains the size of the rest of the segment description. The segments header contains size fields.

GlobalDescriptor

As defined in subclause 5.4.

LocalDescriptor

As defined in subclause 5.5.

LocalDescriptorLocations

As defined in subclause 5.5.3.

DeepFeatureDescriptor

As defined in subclause 5.6.

5.2 CDVA header

5.2.1 Binary representation syntax

CDVAHeader {	Number of bits	Mnemonics
VersionID	4	bslbf
ExtractionParams {		
skip	4	uimsbf
kfTh	8	uimsbf
segTh	8	uimsbf

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

CDVAHeader {	Number of bits	Mnemonics
verTh	8	uimsbf
minLocalDiff	8	uimsbf
CDVSModeID	3	uimsbf
Reserved5Bits	5	
}		
LocalFeatureDescriptorPresent	1	bslbf
DeepFeatureDescriptorPresent	1	bslbf
IsDefaultNN	1	bslbf
IsDeepFeatureDescriptorBinarized	1	bslbf
Reserved4Bits	4	
if (DeepFeatureDescriptorPresent AND IsDefaultNN>0) {		
NNUri	>=8	vlclbf
NNFormat	8	bslbf
DeepFvDim	16	uimsbf
}		
OriginalPictureWidth	16	uimsbf
OriginalPictureHeight	16	uimsbf
if (LocalFeatureDescriptorPresent) {		
HistogramMapSizeX	16	uimsbf
HistogramMapSizeY	16	uimsbf
}		
}		

5.2.2 Descriptor component semantics

The header is included once for each file or stream, at the beginning of the CDVA descriptor bitstream.

VersionID

The version number of the CDVA descriptor (1 for the version specified in this document).

ExtractionParams

Extraction parameters include all free parameters for extraction that are not predefined to ensure interoperability. This set includes the parameters for all descriptors supported by CDVA.

skipNum	Number of frames to be skipped after a sampled frame.
kfTh	Threshold (colour histogram) for selecting key frames. [0.00;2.55], represented as unsigned integer *kfTh**100.
segTh	Threshold (colour histogram) for segment candidates. [0.0;2.55], represented as unsigned integer *segTh**100.
verTh	Threshold (SCFV) for verifying segment candidates [0;255]
minLocalDiff	The minimum local difference (in terms of elements) between local descriptors to be encoded. [0;255]
CDVSModeID	Identifier of the CDVS mode used for extracting the local descriptors [0;6]
Reserved5Bits	To be skipped by the parser.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

LocalFeatureDescriptorPresent

Indication of whether local feature descriptors are present for each of the segments. If LocalFeatureDescriptorPresent is set to 1, local feature descriptors are present for each segment, if LocalFeatureDescriptorPresent is set to 0, no local feature descriptors are present.

DeepFeatureDescriptorPresent

Indication of whether deep feature descriptors are present for each of the segments. If DeepFeatureDescriptorPresent is set to 1, deep feature descriptors are present for each segment, if DeepFeatureDescriptorPresent is set to 0, no deep feature descriptors are present.

IsDefaultNN

Indication of whether custom NN identification is present as follows:

IsDefaultNN == 0	The neural network specified in Annex B.
IsDefaultNN == 1	Another neural network, e.g., a variant of Annex B with different quantization of layers, or alternative network. In this case, the three components NNUri, NNFormat and DeepFvDim are provided.

IsDeepFeatureDescriptorBinarized

Indication of whether deep feature descriptors are binarized (IsDeepFeatureDescriptorBinarized set to 1) or not (IsDeepFeatureDescriptorBinarized set to 0).

Reserved4Bits

To be skipped by the parser.

NNUri

Identifier (URI) of the trained neural network used for deep feature descriptor extraction. This is to be used for quantized versions of the default network, as well as for any other network. The URI is provided as a 0 terminated string with 8 bit characters. The URI serves as an identifier for the particular network instance, and it is recommended to provide the network definition at this location. The format of the network definition shall conform to the format specified in NNFormat.

NNFormat

The format used for representing the neural network referenced by NNUri.

NNFormat == 1 ... NNEF

NNFormat == 0 ... custom format

other values reserved for future use

DeepFvDim

Dimension of deep feature vector, i.e. size of the binarized feature vector resulting from the neural network evaluation (512 for the neural network defined in Annex B).

OriginalPictureWidth

Input frame width in pixels.

OriginalPictureHeight

Input frame height in pixels.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

LocalFeatureDescriptorPresent

This descriptor component specifies whether a relevance bit for each compressed local feature descriptor is present in the bitstream. If LocalFeatureDescriptorPresent is equal to 1 then the relevance bits are present in the bitstream, and if LocalFeatureDescriptorPresent is equal to 0 then the relevance bits are not present in the bitstream. More details are provided in subclause 5.5.

HistogramMapSizeX

Spatial resolution of the histogram for coordinate coding (see subclause 5.5.3).

HistogramMapSizeY

Spatial resolution of the histogram for coordinate coding (see subclause 5.5.3).

5.3 Segment header

5.3.1 General

The segment header is included once per segment, preceding the encoded descriptors of the segment. At least one segment shall be present in the bitstream.

5.3.2 Binary representation syntax

SegmentHeader {	**Number of bits**	**Mnemonics**
IsValidStartTime	1	bslbf
IsValidEndTime	1	bslbf
IsBufferSizeValid	1	bslbf
Reserved5Bits	5	
StartTime	24	uimsbf
EndTime	24	uimsbf
GlobalDescSize	24	uimsbf
LocalDescSize	24	uimsbf
DeepFeatureDescSize	24	uimsbf
NrFrames	8	uimsbf
}		

5.3.3 Descriptor component semantics

IsValidStartTime

This descriptor component specifies whether the start time is valid in the bit stream. If IsValidStartTime is equal to 1 then the start time is valid. If IsValidStartTime is equal to 0 then the start time is not valid in the bitstream.

IsValidEndTime

This descriptor component specifies whether the end time is valid in the bit stream. If IsValidEndTime is equal to 1 then the end time is valid. If IsValidEndTime is equal to 0 then the end time is not valid in the bitstream.

IsBufferSizeValid

This descriptor component specifies whether the buffer size is valid in the bit stream. If IsBufferSizeValid is equal to 1 then the buffer sizes are valid. If IsBufferSizeValid is equal to 0 then the buffer size is not valid in the bitstream.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

Reserved5Bits

To be skipped by the parser.

StartTime

Segment start time in the input video in milliseconds.

EndTime

Segment end time in the input video in milliseconds.

GlobalDescSize

Encoded size in bytes of the global descriptor buffer, including the descriptor of the representative frame and the difference descriptors.

LocalDescSize

Encoded size in bytes of the local descriptor buffer, including the descriptor of the representative frame and the difference descriptors. Set to 0, if the local descriptor is not included.

DeepFeatureDescSize

Encoded size in bytes of the deep feature descriptor buffer. Set to 0, if the deep feature descriptor is not included.

NrFrames

Number of key frames of the segment retained for encoding [1;255].

5.4 Global descriptor

5.4.1 Binary representation syntax

GlobalDescriptor {	**Number of bits**	**Mnemonics**
GlobalHasBitSelection	1	bslbf
GlobalHasVariance	1	bslbf
Reserved6Bits	6	
GlobalRefDescSize	16	uimsbf
EncodedRefDescriptor	8*GlobalRefDescSize	vlclbf
RawGlobalDiffDescSize	24	uimsbf
EncodedDiffDescriptor	8*(GlobalDiffDesc-Size – GlobalRef-DescSize)	vlclbf
}		

5.4.2 Descriptor component semantics

GlobalHasBitSelection

This descriptor component specifies whether the bit selection is present in the bit stream. If GlobalHasBitSelection is equal to 1 then it uses bit selection. If GlobalHasBitSelection is eqaual to 0 then the bit selection is not present in the bitstream.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

GlobalHasVariance

This descriptor component specifies whether the variance is present in the bit stream. If GlobalHasVariance is equal to 1 then it includes variance. If GlobalHasVariance is equal to 0 then the variance is not present in the bitstream.

Reserved6Bits

To be skipped by the parser.

RawGlobalRefDescSize

The uncompressed size (in bytes) of the binarized global descriptor of the representative frame of the segment, required for the termination of ABAC decoding.

EncodedRefDescriptor

The ABAC encoded block formed from the sequence of the binarized global descriptor of the representative frame of the segment. Bit stuffing is used to fill the block to the next byte boundary with bits having the value 0.

RawGlobalDiffDescSize

The uncompressed size (in bytes) of the block formed from the sequence of binarized differences of the global descriptors of the segment for the frames other than the representative frame, required for the termination of ABAC decoding.

EncodedDiffDescriptor

The ABAC encoded block formed from the sequence of binarized differences of the global descriptors of the segment for the frames other than the representative frame. Bit stuffing is used to fill the block to the next byte boundary with bits having the value 0.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

5.5 Local descriptor

5.5.1 General

The local descriptor is optional.

5.5.2 Local feature descriptor

5.5.2.1 Binary representation syntax

LocalDescriptor {	**Number of bits**	**Mnemonics**
HasRelevanceBits	1	bslbf
Reserved7Bits	7	
for (i=0; i<NrFrames; i++) {		
NrLocalDesc[i]	16	uimsbf
}		
TotalNrLocalDesc	16	uimsbf
for (i=0; i<TotalNrLocalDesc; i++) {		
if (i<NrLocalDesc[0]) {		
Constant1	1	bslbf
AbsFeatureIdx	14	bslbf
}		
else {		

© ISO/IEC 2019 – All rights reserved

LocalDescriptor {	**Number of bits**	**Mnemonics**
if (FeatureSkipped[i]) {		
Constant1	1	bslbf
AbsFeatureIdx	14	bslbf
}		
else		
Constant0	1	bslbf
}		
}		
}		
RawLocalDescSize	24	uimsbf
EncodedDescriptor	8*LocalDescSize	vlclbf
}		

The ABAC encoded block is formed from the sequence of binarized local descriptors of the segment. In particular, the following syntax elements defined in ISO/IEC 15938-13:2015 subclause 4.1 are included for each descriptor:

EncodedDescriptor {	**Number of bits**	**Mnemonics**
for(k=0; k<NumberOfLocalDescriptors; k++) {		
for(n=0; n<(4*NumberOfElementGroups); n++) {		
LocalDescriptorElements[k][n]	1-2	vlclbf
}		
}		
if(HasRelevanceBits) {		
for(k=0; k<NumberOfLocalDescriptors; k++)		
RelevanceBits[k]	1	bslbf
}		
}		
}		

Bit stuffing is used to fill the block to the next byte boundary with bits having the value 0.

5.5.2.2 Descriptor component semantics

The number and order of frames shall be consistent between the specification of the number of descriptors, the descriptor index and the location histograms.

HasRelevanceBits

This descriptor component specifies whether relevance bits are present in the bit stream. If HasRelevanceBits is equal to 1, then relevance bits are present (relevance information of local descriptors as defined by CDVS).

Reserved7Bits

To be skipped by the parser.

NrFrames

Number of key frames in the input video.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

NrLocalDesc[i]

Number of local descriptors in frame *i*, starting with the representative frame.

TotalNrLocalDesc

Total number of independently encoded local descriptors in the input video, $n_T \leq \sum_{i=0}^{n_k} n_l^k[i]$, where n_T is the value of TotalNrLocalDesc, $n_l^k[i]$ is the value of NrLocalDesc[i], and n_k is the number of key frames of the segment.

Constant1

Bit set to 1 to indicate that absolute feature index will follow.

AbsFeatureIdx

Absolute local feature index.

FeatureSkipped[i]

The condition is true if the *i*th local feature is skipped because it is too similar to an already encoded one, and false otherwise.

Constant0

Bit set to 0 to indicate that the feature index will be incremented by 1. This is the case if no features have been skipped at this position, and the feature list is encoded incrementally.

RawLocalDescSize

The uncompressed size (in bytes) of the block formed from the sequence of binarized local descriptors of the segment, required for the termination of ABAC decoding.

5.5.3 Local descriptor locations

5.5.3.1 General

The local descriptor block contains the encoded coordinates of the local feature locations in each of the frames of the segment.

5.5.3.2 Binary representation syntax

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

LocalDescriptorLocations {	**Number of bits**	**Mnemonics**
HistogramBufferSize	24	uimsbf
for (i=0; i<NrFrames; i++) {		
CoordHisto[i]	8*FrameHistogramBufferSize[i]	bslbf
}		
}		

Encoded coordinate histogram for frame *i* as defined by CDVS. FrameHistogramBufferSize[i] is the size of the encoded histogram buffer for frame *i*. The following syntax elements defined in ISO/IEC 15938-13:2015, subclause 4.1 are included for each frame:

HistogramCount (arithmetically coded block)	≥1	vlclbf
HistogramMap (arithmetically coded block)	≥1	vlclbf

© ISO/IEC 2019 – All rights reserved

5.5.3.3 Descriptor component semantics

HistogramBufferSize

Size of the encoded histogram buffer in bytes.

NrFrames

Number of key frames in the input video.

5.6 Deep feature descriptor

5.6.1 Binary representation syntax

DeepFeatureDescriptor {	**Number of bits**	**Mnemonics**
DeepRefDescSize	16	uimsbf
EncodedRefDescriptor	8*DeepRefDescSize	vlclbf
RawDeepDiffDescSize	24	uimsbf
EncodedDiffDescriptor	8*(DeepFeatureDesc-Size – DeepRefDescSize)	vlclbf
}		

5.6.2 Descriptor component semantics

RawDeepRefDescSize

The uncompressed size (in bytes) of the binarized deep feature descriptor of the representative frame of the segment, required for the termination of ABAC decoding.

EncodedRefDescriptor

The ABAC encoded block formed from the sequence of the binarized deep feature descriptor of the representative frame of the segment. Bit stuffing is used to fill the block to the next byte boundary with bits having the value 0.

RawDeepDiffDescSize

The uncompressed size (in bytes) of the block formed from the sequence of binarized differences of the deep feature descriptors of the segment for the frames other than the representative frame, required for the termination of ABAC decoding.

EncodedDiffDescriptor

The ABAC encoded block formed from the sequence of binarized differences of the deep feature descriptors of the segment for the frames other than the representative frame. Bit stuffing is used to fill the block to the next byte boundary with bits having the value 0.

6 CDVA descriptor

6.1 Components

6.1.1 General

The unit described by a CDVA descriptor is a temporal video segment, for example a shot.

A segment descriptor shall contain at least an image descriptor for one key frame of the segment (i.e., the representative frame). Depending on the visual heterogeneity of the content, additional key frames may be described as well.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

A segment descriptor shall always include a global descriptor, and it may optionally include a local feature descriptor and/or a deep feature descriptor.

Global, local and deep feature descriptors may use lossless or lossy encoding along the timeline (i.e., for predictive coding between the descriptors of key frames of one segment).

6.1.2 Global descriptor

6.1.2.1 Global descriptors of key frames

The global descriptor consists of a scalable compressed Fisher vector (SCFV) for each key frame of the segment as defined by CDVS.

It shall be extracted in accordance with ISO/IEC 15938-13:2015, subclause 5.6. Different CDVS extractor modes may be used; however, the mode shall be the same for all descriptors in one bitstream.

At least one SCVF descriptor per segment shall be included in the bitstream.

6.1.2.2 Temporal encoding of global descriptors

6.1.2.2.1 General

From the set of binarized global descriptors $\boldsymbol{G}=\left\{G_1,\ldots,G_{n_k}\right\}$, their pairwise distances $\delta_g(G_i,G_j)$ and the representative frame ρ are determined, e.g. as the medoid of the binarized global descriptors.

$$\rho=\operatorname{argmin}_j \sum_i \delta_g\left(G_i,G_j\right) \tag{1}$$

For the other key frames of the segment, $\widehat{G_i}=G_\rho \oplus G_i, i=1,\ldots n_k, i\neq\rho$ is determined, i.e., the bit-wise differences of the binarized global descriptors are calculated. The rationale is to obtain descriptors of the same size, but with a lower number of bits set. The descriptors are output starting with the descriptor of the representative frame, followed by difference descriptors in order of descending distance to the previously encoded descriptors, i.e.

$$\left[G_\rho,\widehat{G_{s_0}},\ldots,\widehat{G_{s_{n_k-1}}}\right], \text{where} \tag{2}$$

$$s_k=\begin{cases}\arg\max_i \delta_g\left(G_i,G_\rho\right), k=0\\ \arg\max_i \delta_g\left(G_i,G_{s_{k-1}}\right), k>0\end{cases} \tag{3}$$

The number of key frames to be encoded is limited to 255 per segment. If the limit is reached, the remaining descriptors in the sorted list are discarded.

Adaptive binary arithmetic coding (ABAC) is applied to this output sequence, resulting in the encoded global descriptor block. The ABAC encoding and decoding procedure used in CDVA is defined in the following subclauses.

6.1.2.2.2 ABAC encoding

The probability model is initialized by resetting the counters $hist_0$ = $hist_1$ = 0. The encoding parameters are initialized as: min = 0, max = 2^{16} -1, mid = 2^{15} - 1.

A sequence of binary symbols $v[1]$, ..., $v[n_b]$ is encoded as specified by the following pseudocode. The function `output()` writes one bit to the output data stream.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

```
e3count = 0;
for (k=1;k≤$n_b$;k++) {
  update the model
  mid = low + (high − low) ($hist_0$ / ($hist_0$ + $hist_1$));
  adapt the range boundaries and increase the counter
  if (v==0) {
    low = mid;
    $hist_0$ = $hist_0$ + 1;
  }
  else {
    mid = high;
    hist1 = hist1 + 1;
  }
  check/adjustment of boundaries and reading the next symbol
  while (1 == 1) {
    if ((high & ($2^{16}$ - 1)) == (low & ($2^{16}$ - 1))) {
      msb = ((high & ($2^{16}$-1) >> 15;
      d = (($2^{15}$ - 1) * msb) + msb;
      low = low - d;
      high = high - d;
      output(msb);

      for (i=0;i<e3count;i++) output(!msb);
      e3count = 0;
    }
    else if ((high <= 3*($2^{14}$ - 1)) and (low > ($2^{14}$ - 1))){
      d = $2^{14}$;
      low = low - d;
      high = high - d;
      e3count = e3count +1;
    }
    else break;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

```
    high =  ((high << 0x1) & (2^16 -1)) | 0x1;
    low =  ((low << 0x1) & (2^16 -1)) | 0x1;
  }
}
```

6.1.2.2.3 ABAC decoding

The decoding procedure requires the target number of decoded symbol n_{sym} as input parameter (determine from the size specifications in the segment header of the respective encoded elements). The probability model is initialized by resetting the counters $hist_0 = hist_1 = 0$. The encoding parameters are initialized as: $min = 0$, $max = 2^{16} -1$, $mid = 2^{15} - 1$.

An output sequence of bits *outval*[1], ..., *outval*[n_{sym}] is decoded as specified by the following pseudocode. The function `readbit()` reads one bit from the input data stream.

```
v = 0
for (k=1;k≤$n_{sym}$;k++) {
  update the model
  mid = low + (high – low) (hist$_0$ / (hist$_0$ + hist$_1$));
  obtain the output value outval is obtained as
  if (low <= v <= mid) outval = 0;
  else outval = 1;
  if (outval==0) {
    high = mid;
    hist$_0$ = hist$_0$ + 1;
  }
  else {
    low = mid + 1;
    hist$_1$ = hist$_1$ + 1;
  }
  bit = 0;
  while (1 == 1) {
    if (high <= ($2^{15}$ - 1)) {}
    else if (low > ($2^{15}$ - 1)) {
      high -= (($2^{15}$ - 1) + 1);
      low -= (($2^{15}$ - 1) + 1);
      v -= $2^{15}$;
    }
    else if (high <= 3*($2^{14}$ - 1)) and (low > ($2^{14}$ - 1))){
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

```
          d = 2^14;

          low = low - d;

          high = high - d;

          v = v - d;

        }

        else break;



        bit = readbit();


        high = ((high << 0x1) & (2^16 -1)) | 0x1;

        low = ((low << 0x1) & (2^16 -1)) | 0x0;

        v = ((v << 0x1) & (2^16 -1)) | bit;

      }

    }
```

6.1.3 Local descriptor

6.1.3.1 Local feature descriptors of key frames

A set of local feature descriptors for each key frame of the segment as defined by CDVS.

They shall be extracted in accordance with ISO/IEC 15938-13:2015, subclause 5.5. Any CDVS extractor mode may be used; however, the mode shall be the same for all descriptors in one bitstream.

6.1.3.2 Temporal encoding of local feature descriptors

For each key frame *k*, a set of local descriptors $\boldsymbol{L}^k$ is obtained, with each of its elements $l_i^k = \{x, y, \beta, f\}$ being a tuple of feature location (*x*,*y*), selection priority *β* and feature descriptor *f*. l_i^k is a vector with ternary elements, and

$$\delta_l\left(l_i^p, l_j^q\right) = \left\| l_i^p - l_j^q \right\|_0 \tag{4}$$

is the zero-norm distance function that determines the number of different elements between the two vectors.

Starting from the representative frame, a list *L* of descriptors in the segment is collected:

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

$$L=\left[l_1^{\rho},\ldots,l_{n_l^{\rho}}^{\rho},l_1^{s_0},\ldots,l_{n_l^{s_0}}^{s_0},\ldots\right], \tag{5}$$

where n_l^i is the number of local feature descriptors in key frame *i*.

The list is then filtered to remove descriptors that are similar to already encoded descriptors.

$$L'_t=\begin{cases}\left[L'_{t-1},l_t\right],\delta_l\left(l_t,l_p\right)>\theta_l,\forall l_p\in L'_{t-1}\\ L'_{t-1},\text{otherwise}\end{cases}, \tag{6}$$

where

L'_t is the current filtered sequence;

θ_l is a distance threshold;

L'_0 =[];

l_t is the tth element of L.

If l_t is not added to the sequence, a reference to the index p which minimizes $\delta_l(l_t,l_p)$ is kept. The list of features and the references are used to build the local descriptor index. This index is needed to reconstruct the sequence of descriptors per frame, in order to establish the correspondence to encoded local feature locations.

A parameterization of θ_l = 0 corresponds to lossless compression. Higher values of θ_l result in lossy compression of local feature descriptors.

Adaptive binary arithmetic coding (ABAC, see subclause 6.1.2.2) is applied to the sequence L', resulting in the encoded local descriptor block.

The index of local descriptors per segment is limited to 14 bits. Encoders shall thus truncate the list L'_t to ensure that the number of descriptors per segment does not exceed 2^{14} and discard the additional descriptors.

6.1.3.3 Encoding of local feature locations

Encoding of local feature locations shall be done independently for each key frame of a segment, in accordance with ISO/IEC 15938-13:2015, subclause 5.8. The local feature locations block contains the sequence of encoded coordinate histograms for the key frames of the segment, in the same order as used to encode the global and local descriptors.

6.1.4 Deep feature descriptor

6.1.4.1 Deep feature descriptors of key frames

Figure 1 shows the extraction pipeline of compact deep invariant global descriptors with nested invariance pooling (NIP). n_r is the variation of rotated image angles, and a video key frame k is rotated $(n_r - 1)$ times. By forwarding each rotated image through a pre-trained deep CNN, the convolutional feature maps output by the intermediate layer (e.g., convolutional layer) are represented by a cube $w \times h \times c$, where w and h denote width and height of each feature map respectively, and c is the number of channels in the feature maps. Subsequently, a set of ROIs are extracted from the cube using an overlapping sliding window, with window size $w' \leq w$ and $h' \leq h$. The window size is adjusted to incorporate ROIs with different scales (e.g., 5 × 5, 10 × 10). The number of scales is denoted as n_s. Finally, a 5-D data structure $\gamma_k(P_t,P_s,P_r,m) \in \mathbb{R}^{w'\times h'\times n_s\times n_r\times c}$ is derived, which encodes translations P_t (i.e., spatial locations $w' \times h'$), scales P_s and rotations P_r of input key frame k.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

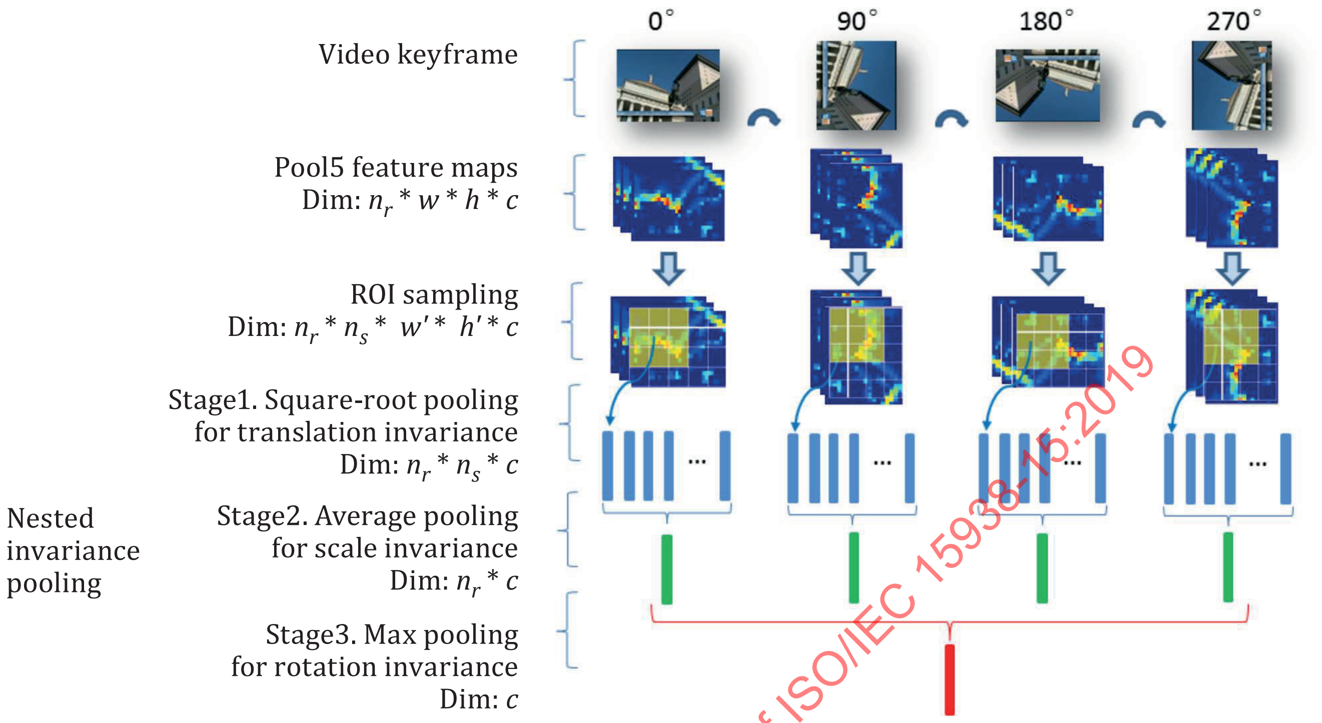

0°
90°
180°
270°
Video keyframe
Pool5 feature maps
Dim: $n_r * w * h * c$
ROI sampling
Dim: $n_r * n_s * w' * h' * c$
Stage1. Square-root pooling for translation invariance
Dim: $n_r * n_s * c$
Nested invariance pooling
Stage2. Average pooling for scale invariance
Dim: $n_r * c$
Stage3. Max pooling for rotation invariance
Dim: c

Figure 1 — Nested invariance pooling on feature maps from pool5 layer of VGG-16 network (n_r = 4)

NIP aims to aggregate the 5-D data into a compact deep invariant global descriptor. In particular, it first performs pooling over translations ($w' \times h'$), then scales (n_s) and finally rotations (n_r) in a nested way, resulting in a c-dimensional global descriptor. Formally, feature map m, p_t -norm pooling over translations P_t is given by

$$\gamma_k\left(P_s, P_r, m\right) = \left(\frac{1}{w' \times h'} \sum_{j=1}^{w' \times h'} \gamma_k\left(j, P_s, P_r, m\right)^{p_t}\right)^{\frac{1}{p_t}} \tag{7}$$

where the pooling operation has a parameter p_t defining the statistical moments, e.g., $p_t = 1$ is first-order (i.e., average pooling), $p_t \rightarrow +\infty$ on the other extreme is infinite order (i.e., max pooling), and $p_t = 2$ is second order (i.e., square-root pooling). Formula (7) leads to a 3D vector $\gamma_k(P_s, P_r, m) \in \mathbb{R}^{n_s \times n_r \times c}$. Analogously, p_s-norm pooling over scale transformations P_s and the subsequent p_r -norm pooling over rotation transformations P_r are

$$\gamma_k\left(P_r, m\right) = \left(\frac{1}{n_s} \sum_{j=1}^{n_s} \gamma_k\left(j, P_r, m\right)^{p_s}\right)^{\frac{1}{p_s}}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

$$\Delta_k^m = \left(\frac{1}{n_r} \sum_{j=1}^{n_r} \gamma_k(j,m)^{p_r} \right)^{\frac{1}{p_r}} \quad (8)$$

The corresponding deep feature descriptor is obtained by concatenating $\gamma_k(m)$ for all feature maps.

$$\Delta_k = \left\{ \Delta_k^m \right\}, 0 \le m < c \quad (9)$$

The resulting descriptors are binarized. A one-bit scalar quantizer q used to binarize the NIP descriptor, so that similar retrieval performance with much less memory footprint and runtime cost can be achieved. The goal is to encode the NIP descriptor $\Delta_k \in \mathbb{R}^c$ as the binary NIP descriptor $D_k \in \{0,1\}^c$. Each element of Δ_k is projected into 1 if $q(\Delta_k) > \tau$, otherwise 0, with τ being a threshold. The quantizer q is supposed to minimize the quantization error:

$$\left\| \Delta_k - D_k \right\|^2 \quad (10)$$

Expanding the above formulation results in:

$$\left\| \Delta_k - D_k \right\|^2 = \left\| \Delta_k \right\|^2 + c - 2\Delta_k{}' D_k . \quad (11)$$

Thus, minimizing Formula (11) can be solved by maximizing $\Delta_k{}' D_k$. The NIP descriptor is translated in feature space by subtracting the mean value NIP descriptor determined on a data set (specified in Annex B), i.e. $\sum_{i=1}^{c} \Delta_{ki} \approx 0$. To maximize $\Delta_k{}' D_k$, the binarized code is set to $D_{ki} = 1$ whenever $\Delta_{ki} > 0$; otherwise $D_{ki} = 0$. Accordingly, the quantizer $q(g)$ is defined as:

$$\mathrm{q}(\Delta_{ki}) = \begin{cases} 1 & \mathit{if}\ \Delta_{ki} > 0; \\ 0 & \mathit{otherwise}. \end{cases} \quad (12)$$

Therefore, for a NIP descriptor, the results are obtained by performing element-wise binarization operation in Formula (12).

6.1.4.2 Temporal encoding of deep feature descriptors

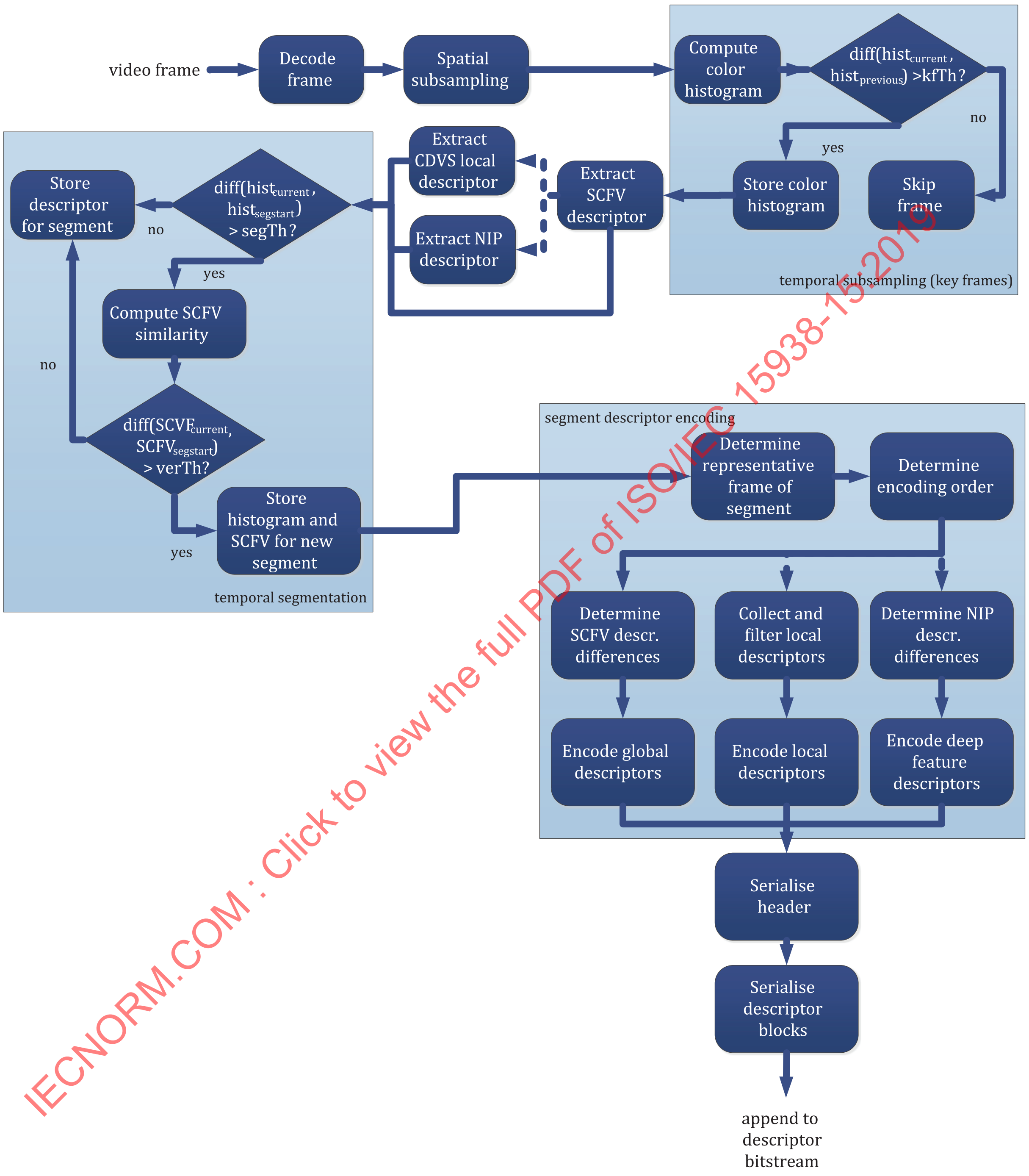
The deep feature descriptor of the representative frame ρ (determined as specified in subclause 6.1.2.2) is included as is. For the other key frames of the segment, $\widehat{D_i} = D_\rho \oplus D_i, i = 1,\ldots n_k, i \ne \rho$ is determined, i.e., the bit-wise differences of the binarized global descriptors are calculated. The rationale is to obtain descriptors of the same size, but with a lower number of bits set. The descriptors are output in the order determined for the global descriptors (as specified in subclause 6.1.2.2), applying the same limit of encoding at most 255 key frames per segment.

Adaptive binary arithmetic coding (ABAC, see subclause 6.1.2.2) is applied to this output sequence, resulting in the encoded deep features descriptor block.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

6.2 Encoding procedure

6.2.1 General



video frame
Decode frame
Spatial subsampling
Compute color histogram
diff(hist_current, hist_previous) >kfTh?
no
yes
Store color histogram
Skip frame
temporal subsampling (key frames)
Extract SCFV descriptor
Extract CDVS local descriptor
Extract NIP descriptor
Store descriptor for segment
diff(hist_current, hist_segstart) > segTh?
no
yes
Compute SCFV similarity
no
diff(SCVF_current, SCFV_segstart) > verTh?
yes
Store histogram and SCFV for new segment
temporal segmentation
segment descriptor encoding
Determine representative frame of segment
Determine encoding order
Determine SCFV descr. differences
Collect and filter local descriptors
Determine NIP descr. differences
Encode global descriptors
Encode local descriptors
Encode deep feature descriptors
Serialise header
Serialise descriptor blocks
append to descriptor bitstream

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

NOTE Dashed lines indicate optional steps in the process.

Figure 2 — CDVA extraction process

Figure 2 illustrates the extraction of a compact descriptor of a video segment in a series of processing steps, when a video frame is given in input to the system. The process is repeated for all frames in the input video segment. The resulting descriptors are grouped by segments. Segments are represented with their first key frame being identified as segment boundary. All following frames until the next segment boundary or the end of the file belong to the same segment. The output descriptor is updated by appending the output of the CDVA segment descriptor extraction into a single CDVA descriptor.

This process contains both normative steps (impacting interoperability of descriptors) and informative steps. Those are specified in subclauses 6.2.2 and 6.2.3, respectively.

1) **Frame subsampling:** perform temporal frame subsampling (typically by a factor of 2-10).
2) **Decode frame**: decode a frame present in the video.
3) **Compute colour histogram**: a histogram in RGB colour space is computed, using 32 bins for each plane.
4) **Check the difference between current and previous colour histograms**: if the difference is greater than a given threshold (*kfTh)*, the frame is selected as key frame and further processed. If not, the frame is dropped.
5) **Frame drop module:** if, according to step 4, the current frame is similar to the previously encoded one, the current frame is dropped.
6) **Store colour histogram**: the colour histogram is stored in memory, to be used as "previous histogram" in the next iteration.
7) **Extract SCFV descriptor**: extract the SCFV descriptors from individual frames, using ISO/IEC 15938-13.
8) **Extract CDVS local descriptor:** extract the CDVS local descriptors from individual frames, using ISO/IEC 15938-13.
9) **Extract deep feature descriptor:** extract the deep feature descriptor using a trained neural network, and binarizes the resulting descriptor.
10) **Check the difference of colour histograms between current frame and segment start:** if the difference is greater than a given threshold (*segTh*), the frame is selected as candidate of a segment boundary (first frame of a new segment).
11) **Compute global SCFV similarity between current descriptor and descriptor of first frame of segment:** if the similarity is below a given threshold (*verTh*), the frame is confirmed to be a segment boundary.
12) **Store colour histogram and SCFV descriptor:** for frames identified as segment boundaries, store the descriptors for comparisons with the subsequent frames.
13) **Store descriptors for segment:** store all extracted descriptors for the current segment together with their time index.
14) **Determine representative frame:** select a representative frame for the segment, selected as the frame with the medoid SCFV descriptor. This frame is used as the reference for encoding the global and local descriptor of the segment.
15) **Determine encoding order:** the encoding order is determined from the SCFV descriptors of the key frames in the segment. The first is the representative frame, followed by frames with decreasing distance to any of the frames encoded so far.
16) **Determine global descriptor differences:** for each key frame other than the representative frame, determine a difference descriptor as XOR between its SCFV descriptor and the SCFV descriptor of the representative frame.
17) **Encode global descriptor:** encode the block formed from the SCFV descriptor of the representative frame and the difference descriptors using adaptive binary arithmetic coding.
18) **Collect and filter local descriptors:** collect and order the local descriptors used in the segment, starting from the local descriptors of the representative frame, and continuing to the temporally adjacent key frame (alternating in forward and backward direction). The descriptors are filtered

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved

as follows: if a local descriptor does not differ in more than *minLocalDiff* elements from one already collected, it is discarded and replaced by a reference to the already encoded descriptor.

19) **Encode local descriptors:** generate a map of descriptor indices between the original per frame indices and the filtered list of descriptors. If descriptors have been replaced by references, the map has multiple pointers to the same index. The set of local descriptors remaining after filtering is encoded using adaptive binary arithmetic coding.

20) **Determine deep feature descriptor differences:** for each key frame other than the representative frame, determine a difference descriptor as XOR between its deep feature descriptor and the deep feature descriptor of the representative frame.

21) **Encode deep feature descriptor:** encode the block formed from the deep feature descriptor of the representative frame and the difference descriptors using adaptive binary arithmetic coding.

22) **Write header:** write a header structure, containing the start and end time of the segments, parameters needed for decoding, the number of frames, local descriptors and the sizes of the blocks containing the different types of descriptors.

23) **Write descriptor blocks:** write the encoded global, local and deep feature descriptor blocks.

6.2.2 Normative steps

6.2.2.1 Extract SCFV descriptor

As specified in the CDVS specification and implemented in the CDVS reference software.

6.2.2.2 Extract CDVS local descriptor

As specified in the CDVS specification and implemented in the CDVS reference software.

6.2.2.3 Extract deep feature descriptor

The original image is a luminance raster image containing values in the interval [0, 255] where increasing values correspond to increasing luminance. The exact mapping of luminance values within this interval is beyond the scope of this document. The original image shall be spatially resampled to a resolution of 640 × 480 pixels to obtain a converted image $J(x,y)$, in which $x \in \{0, \mathrm{K}, X-1\}$ and $y \in \{0, \mathrm{K}, Y-1\}$ are the horizontal and vertical pixel coordinates respectively, X and Y the pixel horizontal and vertical image dimensions respectively, and with coordinates located at the top left corner of the image. For this resampling operation, a Lanczos filter with $a = 3$ should be used.

The default deep feature descriptor is extracted with the VGG16 network, using the output of the pool5 layer. The input resolution of 640 × 480 determines dimensions of $w = 20$, $h = 15$ for feature maps. $c = 512$ denotes 512 channels and $n_s = 20$ denotes 20 regions. In practice, the input image is rotated by 90, 180, 270 degrees ($n_r = 4$).

NIP descriptors are further improved by post-processing techniques such as PCA whitening. In particular, NIP is firstly L2 normalized, followed by PCA projection and whitening with a pre-trained PCA matrix. The whitened vectors are L2 normalized and compared with cosine similarity.

Other neural networks may be used for the deep feature descriptor. In this case, the metadata identifying the neural network shall be provided in the CDVA header.

During NIP descriptor extraction procedures, a series of regions needs to be sampled. Regions are sampled from three different scales. Under different scales, the stride and region sizes are different. The following is the configuration used in the NIP implementation. The input image size is 640 × 480. Then after feedforward operations in VGG16 model, 512 feature maps of size 20 × 15 are obtained. In total, there are 20 regions, and the top-left points of the regions are listed in Table 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

Table 1 — Parameters of regions sampled at different scales.

Scale	Region size	Number of sampled regions	Top-left point coordinates
1	15 × 15	2	{ (0,0), 0,5) }
2	10 × 10	6	{ (0,0), (0,5), (0,10), (5,0), (5,5), (5,10) }
3	7 × 7	12	{ (0,0), (0,4), (0,8), (0,13), (4,0), (4,4), (4,8), (4,13), (8,0), (8,4), (8,8), (8,13) }

6.2.2.4 Determine encoding order

As specified in subclause 6.1.2.2.

6.2.2.5 Determine global descriptor differences

As specified in subclause 6.1.2.2.

6.2.2.6 Encode global descriptor

As specified in subclause 6.1.2.2.

6.2.2.7 Collect and filter local descriptors

As specified in subclause 6.1.3.2.

6.2.2.8 Encode local descriptors

As specified in subclause 6.1.3.2.

6.2.2.9 Determine deep feature descriptor differences

As specified in subclause 6.1.4.2.

6.2.2.10 Encode deep feature descriptor

As specified in subclause 6.1.4.2.

6.2.2.11 Serialization

The serialized representation consists of one instance of the header for the bitstream, followed by one or more encoded segment descriptors. A segment descriptor consists of at least a segment header and the buffer for the global descriptor. When indicated in the segment header, the buffers for local and/or deep feature descriptors may be included.

The format of the serialized structure shall conform to the bitstream syntax specified in Clause 5.

6.2.3 Informative steps

6.2.3.1 Sampling

Temporal subsampling may be performed to reduce the computational effort. Frames of the input sequence are skipped after sampling a frame, as specified by parameter. The subsampling factor is thus given as (*skipNum* + 1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 15938-15:2019

© ISO/IEC 2019 – All rights reserved