D:\data\mpeg\mpeg120\output\Text\_of\_ISO-IEC\_15938-15.docxWORKING DRAFT© ISO/IEC 2017 – All rights reservedISO/IEC WD 15938-15 63Part 15: Part 15: Compact Descriptors for Video AnalysisInformation Technology — Multimedia Content Description InterfaceÉlément introductif — Élément central — Partie 15: Titre de la partieInformation Technology — Multimedia Content Description Interface — Part 15: Part 15: Compact Descriptors for Video AnalysisE2017-12-01(20) PreparatoryISO/IECISO/IEC J   International Standard2017 ISO/IEC 15938‑ISO/IEC 15938‑15ISO/IEC WD 15938-15 JISC Information Technology — Multimedia Content Description Interface — Part 15: Compact Descriptors for Video Analysis11291 2Überschrift 2Überschrift 1    02 STD Version 2.1c210   4            ISO/IEC JTC 1/SC 29 N

Date:   2017-12-01

ISO/IEC WD 15938-15

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:   JISC

Information Technology — Multimedia Content Description Interface — Part 15: Part 15: Compact Descriptors for Video Analysis

*Élément introductif — Élément central — Partie 15: Titre de la partie*

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents Page

Foreword iv

Introduction v

1 Scope 1

2 Terms and definitions 1

3 Symbols and abbreviations 2

3.1 Symbols 2

3.2 Abbreviations 3

4 Overview of components 3

4.1 Global descriptor 3

4.1.1 Global descriptors of key frames 3

4.1.2 Temporal encoding of global descriptors 4

4.2 Local descriptor 4

4.2.1 Local feature descriptors of key frames 4

4.2.2 Temporal encoding of local feature descriptors 4

4.2.3 Encoding of local feature locations 5

4.3 Deep feature descriptor 5

4.3.1 Deep feature descriptors of key frames 5

4.3.2 Temporal encoding of deep feature descriptors 7

5 CDVA bitstream syntax 7

5.1 Header 7

5.2 Segment header 8

5.3 Global descriptor 8

5.4 Local descriptors 9

5.4.1 Local feature descriptor 9

5.4.2 Local descriptor locations 10

5.5 Deep feature descriptor 10

6 CDVA encoding 11

6.1 Overview of encoding procedure 11

6.2 Normative steps 13

6.2.1 Extract SCFV descriptor 13

6.2.2 Extract CDVS local descriptor 13

6.2.3 Extract deep feature descriptor 13

6.2.4 Determine encoding order 14

6.2.5 Determine global descriptor differences 14

6.2.6 Encode global descriptor 14

6.2.7 Collect and filter local descriptors 14

6.2.8 Encode local descriptors 14

6.2.9 Determine deep feature descriptor differences 14

6.2.10 Encode deep feature descriptor 14

6.2.11 Serialization 14

6.3 Informative steps 15

6.3.1 Sampling 15

6.3.2 Temporal segmentation 15

6.3.3 Verification of segmentation using SCFV 15

6.3.4 Determine representative frame 15

7 Bibliography 15

8 Annex 1: Proposed parameterization (informative) 16

9 Annex 2: Parameters of the extraction process 17

9.1.1 CNN for deep feature extraction 17

9.1.2 Mean NIP vector for binarization of deep feature descriptors 17

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.

ISO/IEC 15938‑15 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology — Multimedia Content Description Interface — Part 15: Compact Descriptors for Video Analysis*, Subcommittee SC 29, .

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

ISO/IEC 15938 consists of the following parts, under the general title *Information Technology — Multimedia Content Description Interface*:

* *Part 15: Part 15: Compact Descriptors for Video Analysis*
* *Part [n]:*
* *Part [n+1]:*

Introduction

The current media and entertainment domain is characterized by increasing volumes of content, large number of delivery channels and an ever-growing need for the relevant content to be accessed on demand or published quickly. This is very difficult without appropriate tools to manage content items, including search for object instances in video, categorization of scenes and content grouping. Thus, the ability to generate and exchange compact and standardized descriptors in an interoperable and efficient way is considered a key enabler in this domain.

An example application is scalable instance search to find a specific object instance in a very large video database (e.g. like a broadcaster might have). The industry needs video descriptors that enable performing this task with smaller descriptor size and shorter runtimes as compared to using CDVS (ISO/IEC 15938-13) in video domain. Rigid objects (e.g., textured buildings) are of highest relevance, however object instances with more challenging properties (deformable, reflective and inherently varying spatial views) are also of interest.

Information Technology — Multimedia Content Description Interface — Part 15: Part 15: Compact Descriptors for Video Analysis

# Scope

This standard 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, that is robust 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. The industry thus needs video descriptors that enable performing this task 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.

There are a number of component technologies that are useful for visual search, including format of visual descriptors, descriptor extraction process, as well as indexing, and matching algorithms. The bitstream syntax of descriptors as well as parts of the extraction process and its parameters are defined to ensure interoperability.

A standard for compact descriptors for video analysis for search and retrieval applications:

* enables design of interoperable object instance search applications
* minimizes the size of video descriptors
* ensures high matching performances of objects (in terms of accuracy and complexity)
* enables efficient implementation of those functionalities on professional or embedded systems

This standard provides a complementary tool to the suite of existing MPEG standards, such as MPEG-7 Compact Descriptors for Visual Search (ISO/IEC 15938-13).

# Terms and definitions

**relevant video segments**

video segments that contain view(s) of object(s) or the scene present that match the query content

**identified video segments**

list of all video segments returned in response to a query request. This includes video material ID, and start/end frame for the segment.

**image descriptor**

descriptor extracted from one image

**segment descriptor**

descriptor extracted from the sampled key frames of a video segment

**input video segment**

a time range of a video from which a segment descriptor shall be extracted

**interest point**

point in an image showing detection stability under local and global perturbations in the image domain, including perspective transformations, changes in image scale, and illumination variations

**local feature descriptor**

descriptor of a local region, extracted around an interest point

**global descriptor**

aggregation of local feature descriptors into a compact representation of the image

**compressed local feature descriptor**

compressed representation of a local feature descriptor

**interest point coordinate**

horizontal and vertical pixel coordinates indicating the position of an interest point in the image resolution used for descriptor extraction, rounded to the nearest integer

**representative frame**

a frame of a video segment for which an uncompressed descriptor is represented, and which is used as the basis for differential encoding

**temporal localization information**

description of temporal location of relevant segments in a video

**deep feature descriptor**

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

**convolutional neural network (CNN)**

a class of deep, feed-forward artificial neural network

# Symbols and abbreviations

## Symbols

zero-norm distance (L0 norm)

bit-wise difference (XOR) operator

*c* number of channels of feature map (dimension of descriptor extracted from CNN)

*Δk* deep feature descriptor for frame *k*

*Dk* binarized deep feature descriptor for frame *k*

*δg* distance of global descriptors

*δl* distance of local descriptors

*f* feature vector of local descriptor

***G*** set of global descriptors

*Gk* global descriptor of frame *k*

*γk* result of pooling operation for feature map for frame *k*

*h* feature map height

*k* key frame index

*m* feature map index

*nk* number of key frames in a video segment

number of local descriptors of frame *k*

*ρ* representative frame of video segment

*pr, pr, pt* statistical moment for pooling operation

*Ps* scale invariance pooling

*Pr* rotation invariance pooling

*Pt* translation invariance pooling

selection priority of local feature

*q* quantization function

***L****k* set of local descriptors of frame *k*

*L* list of local feature descriptors of a video segment

*i*th local descriptor of frame *k*

*θl* thresholdfor local descriptor distance

*r* number of rotation transformations

*s* number of scale transformations

*w* feature map width

*x* horizontal image coordinate

*y* vertical image coordinate

## Abbreviations

**ABAC** Adaptive Binary Arithmetic Coding

**CDVA** Compact Descriptors for Visual Analysis

**CDVS** Compact Descriptors for Visual Search

**CNN** Convolutional neural network

**MPEG** Moving Picture Experts Group

**MPEG-7** ISO/IEC 15938

**NIP** Nested Invariance Pooling

**PCA** Principal Component Analysis

**SCFV** Scalable Compressed Fisher Vector

# Overview of components

The unit of the CDVA descriptor is a temporal video segment, for example, a shot.

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

A segment descriptor must 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.

## Global descriptor

### Global descriptors of key frames

A scalable compressed Fisher vector (SCFV) for each key frame of the segment as defined by CDVS

Extracted as specified in ISO/IEC 15938-13 clause 5.6. Different CDVS extractor modes may be used, however, the mode must be the same for all descriptors in one bitstream.

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

### Temporal encoding of global descriptors

From the set of binarized global descriptors , the pairwise distances *δg*(*Gi*,*Gj*) are determined, and the representative frame is selected as the medoid of the descriptors

For the other key frames of the segment, 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.

Adaptive binary arithmetic coding (ABAC) [1] is applied to this output sequence, resulting in the encoded global descriptor block.

In order to meet a maximum size limit of the descriptor per segment, the descriptors of some key frames may be discarded.

## Local descriptor

### Local feature descriptors of key frames

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

The extraction is performed as described in ISO/IEC 15938-13 clause 5.5. Different CDVS extractor modes may be used, however, the mode must be the same for all descriptors in one bitstream.

### Temporal encoding of local feature descriptors

For each key frame *k*, we obtain a set of local descriptors ***L****k*, with each of its elements being a tuple of feature location (*x*,*y*), selection priority *π* and feature descriptor *f*. is a vector with ternary elements, and

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

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

,

where 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:

,

where *L't* is the current filtered sequence, *θl* is a distance threshold, and *L'0*=[]. *lt* is the *t*th element of *L*. If *lt* is not added to the sequence, a reference to the index *p* which minimizes *δl*(*lt*,*lp*) 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 *θl*=0 corresponds to lossless compression, higher values of *θl* will result in lossy compression of local feature descriptors.

Adaptive binary arithmetic coding (ABAC) [1] is applied to the sequence *L'*, resulting in the encoded local descriptor block.

In order to meet a maximum size limit of the descriptor per segment, the descriptors of some key frames may be discarded.

### Encoding of local feature locations

Encoding of local feature locations is done independently for each key frame of a segment, as defined in ISO/IEC 15938-13 clause 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.

## Deep feature descriptor

### Deep feature descriptors of key frames

Figure 1 shows the extraction pipeline of our compact deep invariant global descriptors with Nested Invariance Pooling (NIP). Given a video key frame *k* we rotate it by *r* times. By forwarding each rotated image through a pre-trained deep CNN, the convolutional feature maps output by intermediate layer (e.g., convolutional layer) are represented by a cube *w* × *h* × *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, we extract a set of ROIs from the cube using an overlapping sliding window, with window size *w' ≤ w* and *h'≤ h*. The window size is adjusted to incorporate ROIs with different scales (e.g., 5 × 5, 10 × 10). Here, we denote the number of scales as *s*. Finally, a 5-D data structure *γk(Pt,Ps,Pr,m) ∈* ℝ*w'×h'×s×r×c* is derived, which encodes translations *Pt* (i.e., spatial locations *w'* × *h'* ), scales *Ps* and rotations *Pr* of input key frame *k*.

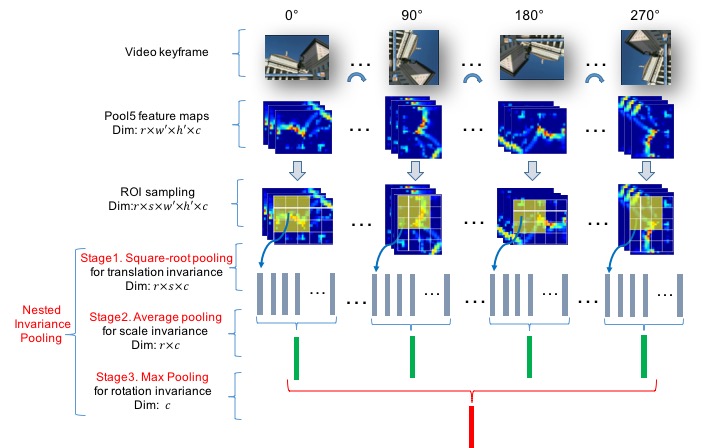


Figure 1: Nested Invariance Pooling on feature maps from pool5 layer of VGG-16 network.

NIP aims to aggregate the 5-D data into a compact deep invariant global descriptor. In particular, it firstly performs pooling over translations (*w'× h'*), then scales (*s*) and finally rotations (*r*) in a nested way, resulting in a *c*-dimensional global descriptor. Formally, feature map *m*, *pt* -norm pooling over translations *Pt* is given by

(4.3.1)

where the pooling operation has a parameter *pt* defining the statistical moments, e.g., *pt* = 1 is first-order (i.e., average pooling), *pt* → +∞ on the other extreme is infinite order (i.e., max pooling), and *pt* = 2 is second order (i.e., square-root pooling). Equation (4.3.1) leads to a 3D vector *γk*(*Ps,Pr,m*) ∈ ℝ*s×r×c* . Analogously, *ps*-norm pooling over scale transformations *Ps* and the subsequent *pr* -norm pooling over rotation transformations *Pr* are

(4.3.2)

The corresponding deep feature descriptor is obtained by concatenating *γk*(*m*) for all feature maps.

(4.3.3)

The resulting descriptors are binarized. A one-bit scalar quantizer is to binarize the NIP descriptor, so that retrieval performance with much less memory footprint and runtime cost can be achieved. The goal is to encode the NIP descriptor using a binary vector . Each element of is projected into 1 if , otherwise 0, with being a threshold. The quantizer is supposed to minimize the quantization error:

. (4.3.4)

Expanding the above formulation results in:

(4.3.5)

Thus minimizing equation (4.3.5) can be solved by maximizing . The NIP descriptor is translated in feature space by subtracting the mean value NIP descriptor determined on a data set (specified in Annex 2), i.e. To maximize , the binarized code is set to whenever .; otherwise = 0. Accordingly, the quantizer in the form as:

(4.3.6)

Therefore, for a NIP descriptor, the results are obtained by performing element-wise binarization operation in equation (4.3.6).

### Temporal encoding of deep feature descriptors

The deep feature descriptor of the representative frame *ρ* (determined as described in Clause 4.1.2) is included as is. For the other key frames of the segment, 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 described in clause 4.1.2).

Adaptive binary arithmetic coding (ABAC) [1] is applied to this output sequence, resulting in the encoded deep features descriptor block.

# CDVA bitstream syntax

## Header

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

| **Element** | **Description** | **Size (bits)** |
| --- | --- | --- |
| Version | The version number of the CDVA descriptor (1 for the version described in this document. | 4 |
| Extraction parameters |  | 36 |
| skip | Number of frames to be skipped before and after a sampled frame. | 4 |
| th | Threshold (color histogram) for selecting key frames. [0;3] | 8 |
| seg\_th | Threshold (color histogram) for segment candidates. [0;3] | 8 |
| ver\_th | Threshold (SCFV) for verifying segment candidates [0;255] | 8 |
| min\_local\_diff | The minimum local difference (in terms of elements) between local descriptors to be encoded. [0;127] | 7 |
| (unused) |  | 1 |
| Original picture width | pixels | 16 |
| Original picture height | pixels | 16 |
| Histogram map size X | Spatial resolution of the histogram for coordinate coding (see clause 5.4.2) | 16 |
| Histogram map size Y | Spatial resolution of the histogram for coordinate coding (see clause 5.4.2) | 16 |

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.

## Segment header

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

| **Element** | **Description** | **Size (bits)** |
| --- | --- | --- |
| Flags | 0 start time is valid  1 end time is valid | 2 |
| (unused) |  | 6 |
| Start time | Segment start time in milliseconds. | 24 |
| End time | Segment end time in milliseconds. | 24 |
| Global descriptor buffer size | Encoded size in bytes | 24 |
| Local descriptor buffer size | Encoded size in bytes. Set to 0, if the local descriptor is not included. | 24 |
| Deep feature descriptor buffer size | Encoded size in bytes. Set to 0, if the deep feature descriptor is not included. | 24 |
| Number of frames encoded in segment | Number of key frames of the segment retained for encoding | 8 |

## Global descriptor

| **Element** | **Description** | **Size (bits)** |
| --- | --- | --- |
| Flags | 0 uses bit selection  1 includes variance | 2 |
| (unused) |  | 6 |
| Uncompressed global descriptor block size | Bytes | 24 |
| Encoded global descriptor block |  | 8 x compressed size in bytes as specified in segment header |

## Local descriptors

The local descriptor is optional.

### Local feature descriptor

|  |  |  |
| --- | --- | --- |
| **Element** | **Description** | **Size (bits)** |
| CDVS Descriptor mode | Identifier of the CDVS mode used for extracting the local descriptors [0,6] | 3 |
| (unused) |  | 1 |
| Flags | 0 relevance bits present (relevance information of local descriptors as defined by CDVS) | 1 |
| (unused) |  | 3 |
| Number of local descriptors per frame | Starting with the reference frame | nFrames \* 16 |
| Local descriptor index | Mapping between the local feature in aggregated list *L*' and the local feature index in the original frame, encoded as   * bit set to 1, followed by absolute feature index (13 bits), for first feature of a frame, or if features were skipped * bit set to 0 to increment feature index | 2\*total nr. local descriptors\*8 |
| Uncompressed local descriptor block size | Bytes | 24 |
| Encoded local descriptor block |  | 8 x compressed size in bytes as specified in segment header |

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

### Local descriptor locations

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

|  |  |  |
| --- | --- | --- |
| **Element** | **Description** | **Size (bits)** |
| Histogram buffer size | Bytes | 24 |
| Sequence of encoded coordinate histograms as defined in CDVS |  | 8 x compressed size in bytes as specified |

## Deep feature descriptor

|  |  |  |
| --- | --- | --- |
| **Item** |  | **Size (bits)** |
| Deep feature descriptor flags | 0 … neural network used for descriptor extraction (0 = neural network specified in Annex 2, 1 = other neural network) | 1 |
| (unused) |  | 7 |
| Identifier of neural network | Application specific identifier | 0 (if flags indicate network from Annex 2 is used)  24 otherwise |
| Dimension of deep feature vector | Size of the binarized feature vector resulting from the neural network evaluation (512 for the neural network defined in Annex 2) | 16 |
| Uncompressed deep feature descriptor block size | Bytes | 24 |
| Encoded deep feature descriptor block |  | 8 x compressed size in bytes as specified in the segment header |

# CDVA encoding

## Overview of encoding procedure



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 Extraction into a single CDVA descriptor.

1. **Frame subsampling:** Performs temporal frame subsampling (typically by a factor of 2-10).
2. **Decode frame**: Decode a frame present in the video.
3. **Compute color histogram**: a histogram of the R,G,B planes is computed, using 32 bins for each plane.
4. **Check the difference between current and previous color histograms**: if the difference is greater than a given threshold (*th*), 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 color histogram**: the color histogram is stored in memory, to be used as “previous histogram” in the next iteration.
7. **Extract SCFV descriptor**: Extracts the CDVS descriptors from individual frames, using the CDVS standard (ISO/IEC 15938-13).
8. **Extract CDVS local descriptor:** Extracts the CDVS local descriptors from individual frames, using of the CDVS standard (ISO/IEC 15938-13).
9. **Extract deep feature descriptor:** Extract the deep feature descriptor using a pretrained VGG16 network, and binaries the resulting descriptor (as described in clause 4.3).
10. **Check the difference of color 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 (*gth*), the frame is confirmed to be a segment boundary.
12. **Store color 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 reference 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 frames (alternating in forward and backward direction). The descriptors are filter as follows: if a local descriptor does not differ in more than *min\_local\_diff* 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. In case descriptors have been replaced by references, the map will have 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 reference 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.

## Normative steps

### Extract SCFV descriptor

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

### Extract CDVS local descriptor

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

### Extract deep feature descriptor

[NOTE: normative only for strict conformance, for conformance only on the level of encoded bitstream, this step may be exchanged]

The VGG16 network [2] is used for this step, using the output of the pool5 layer. In our implementation, the input image is scaled to 640x480 which means w=20, h=15 for feature maps. c=512 denotes 512 channels and s=20 denotes 20 regions. In practice, we rotate input image by 90, 180, 270 degrees (r=4 rotations).

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.

During NIP descriptor extraction procedures, a series of regions needs to be sampled. Regions are sampled from three different scales. Under different scale, the stride and region sizes are different. The below is the configuration we used in NIP implementation. The input image size is 640x480. Then after feedforward operations in VGG16 model, we can get 512 feature maps of size 20x15. In total, there are 20 regions, and the top-left points of the regions are listed.

| **Scale** | **Region size** | **Number of sampled regions** | **Top-left point coordinates** |
| --- | --- | --- | --- |
| 1 | 15x15 | 2 | { (0,0), 0,5) } |
| 2 | 10x10 | 6 | { (0,0), (0,5), (0,10), (5,0), (5,5), (5,10) } |
| 3 | 7x7 | 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) } |

### Determine encoding order

As specified in clause 4.1.2.

### Determine global descriptor differences

As specified in clause 4.1.2.

### Encode global descriptor

As specified in clause 4.1.2.

### Collect and filter local descriptors

As specified in clause 4.2.2

### Encode local descriptors

As specified in clause 4.2.2

### Determine deep feature descriptor differences

As specified in clause 4.3.2

### Encode deep feature descriptor

As specified in clause 4.3.2

### 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 must conform to the bitstream syntax specified in clause 5.

## Informative steps

### Sampling

### Temporal segmentation

### Verification of segmentation using SCFV

### Determine representative frame

# Bibliography

[1] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon & R. B. Arps: “An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder”, IBM J. Res. Develop. 32, pp. 717  726, 1988

[2] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

# Annex 1: Proposed parameterization (informative)

No fixed working points, informative section with proposed parameter ranges

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Process** | **Description** | **Default** |
| skip\_before | Extraction | Number of frames skipped before a sampled frame | 4 |
| skip\_after | Extraction | Number of frames skipped after a sampled frame | 4 |
| th | Extraction | Threshold (color histogram) for selecting key frames | 0.7 |
| seg\_th | Extraction | Threshold (color histogram) for segment candidates | 1.98 |
| ver\_th | Extraction | Threshold (CDVS global) for verifying segment candidates | 18 |
| min\_local\_diff | Extraction | The minimum local difference (in terms of elements) between local descriptors to be encoded (corresponding to lossless compression). | 1 |
| CDVS Descriptor mode | Extraction | The CDVS descriptor mode used. | 0 |

# Annex 2: Parameters of the extraction process

### CNN for deep feature extraction

describing network, requires capturing network topology and weights

### Mean NIP vector for binarization of deep feature descriptors

The mean NIP vector used for binarization is listed below.

0.0058197395745, 0.0226217675917, 0.0202749371303, 0.0155842523303, 0.00206882785025, 0.029584621966, 0.038005899671, 0.016528418024, 0.0247320122345, 0.0122353927363, 0.017080113834, 0.00712884363225, 0.0123059255535, -0.0105145122705, 0.0207673153715,   
-0.00909547014575, 0.00916992669575, 0.0273888260937, 0.018752814925, 0.0152868328593, 0.0445336729452, 0.0215145660928, 0.0203954865802, 0.0234313924643, 0.00524717686175, 0.0242731346265, 0.0289761905763, 0.0280460966355, 0.017615927364, 0.0243089468142, 0.0193036371632, 0.013409030424, 0.0309236881018, 0.0257856419995, 0.011399816407, 0.019129428682, 0.0259679771572, 0.0284168086223, 0.01345578691, 0.0175459276602, 0.027395239278, 0.00666205869725, 0.0195553116088, 0.0392152485792, 0.0136351882353, 0.0409940035042, 0.0167323242787, 0.028913315617, 0.0257141238363, 0.02690220789, 0.0227256735683, 0.022336113465, 0.021211028819, 0.021772930042, 0.021908772172, 0.0261280332655, 0.0253713006843, 0.0239380601443, 0.03415758131, 0.0211405949132, 0.0279264624117, 0.0261920855632, 0.02128137101, 0.0303210906008, 0.0310672161878, 0.0365590380875, 0.0236870791805, 0.0243907555218, 0.026322812217, 0.0305406238485, 0.0361602068683, 0.0238308424665, 0.0232999506227, 0.0314031987985, 0.0219974163363, 0.0156449749333, 0.0286064100997, 0.0294873809845, 0.0235161836865, 0.03447762474, 0.0277428287925, 0.0293508453585, 0.0278911456642, 0.0294239725642, 0.0312639482457, 0.0233654262662, 0.0169140082432, 0.0279709921002, 0.0283202204552, 0.0225416976585, 0.0215537471933, 0.029058299679, 0.0232663183397, 0.016614706213, 0.0211547221025, 0.025384580921, 0.019664851772, 0.024960806945, 0.0206738486005, 0.0197707366675, 0.0279514846577, 0.0294827706572, 0.023431541614, 0.0216941109597, 0.0305681944643, 0.0315824753132, 0.020248133739, 0.0236163262458, 0.0317520210487, 0.0327342338805, 0.0297010453865, 0.030222793652, 0.0224698529018, 0.0263406126527, 0.0271875863347, 0.030348754238, 0.0260163466505, 0.0200897973242, 0.0285137565478, 0.027136599387, 0.0296097338805, 0.0287653057302, 0.0282526544912, 0.0288727058652, 0.033271722726, 0.025932215883, 0.0263048944515, 0.024224776357, 0.024565864631, 0.0279205320288, 0.0309231297297, 0.0347142209062, 0.0403212456945, 0.031239713809, 0.0238175790327, 0.0287571898685, 0.0342754993582, 0.0193019639645, 0.0276222059255, 0.0202216318653, 0.0295162987598, 0.0252226728543, 0.029154281835, 0.0314661849217, 0.0192040843237, 0.0314981175073, 0.0254065509248, 0.0230813488198, 0.024692731679, 0.0256251256093, 0.0326177089335, 0.0319256394952, 0.0250807688187, 0.0291628143008, 0.036706077866, 0.0331032075195, 0.022590679736, 0.030768290391, 0.0243820557292, 0.0272579297092, 0.0308361629515, 0.020727241318, 0.0308387033335, 0.029924132708, 0.0227707651483, 0.0327331945035, 0.023580599136, 0.0342618573207, 0.0233942059808, 0.03093170719, 0.0296530524385, 0.0298438369802, 0.0245936033625, 0.023605879956, 0.0273949869565, 0.0238446511835, 0.0242326018003, 0.0281564644735, 0.0281974365738, 0.030071122657, 0.031864731148, 0.0250918541315, 0.0321710143938, 0.0283538821483, 0.029706207414, 0.0239366809277, 0.0263788457187, 0.0270092568242, 0.032302620426, 0.0267683291743, 0.0289038030315, 0.0250336385095, 0.0351151765717, 0.0234271303583, 0.035444160666, 0.0294192925318, 0.0250971470065, 0.0282549192462, 0.026746021514, 0.0309562638975, 0.0349411097165, 0.0279854703858, 0.0273719979258, 0.028850290678, 0.024677842984, 0.0307281782003, 0.03287051178, 0.0289088738375, 0.0267455129325, 0.0293861691037, 0.028082594747, 0.028874946859, 0.0330696857035, 0.0280922629012, 0.0280832393907, 0.0277765193855, 0.0293994957767, 0.0308071271557, 0.0245425149145, 0.0325668884262, 0.0315901276765, 0.0344813526713, 0.0293978046748, 0.0325455867145, 0.0292055251928, 0.0286491233553, 0.0259101732933, 0.0305542679653, 0.0335555267605, 0.0267651571593, 0.0276011314415, 0.0274300069327, 0.026625609054, 0.0277532715705, 0.033398343493, 0.0239298614925, 0.033372478853, 0.03397257458, 0.0348946650035, 0.033581786472,   
0.030023363268, 0.028403478765, 0.0341631561863, 0.0286766831203, 0.030297623174, 0.0284927249617, 0.0312253405323, 0.0293969683573, 0.0331701725335, 0.0290916984317, 0.0254170781148, 0.03322021924, 0.0317817605157, 0.030556624708, 0.0285278324833, 0.029663595497, 0.0286140484585, 0.0288106080895, 0.0300296191353, 0.0311715805277, 0.0305311768073, 0.0311608192052, 0.0300077470905, 0.0271409012165, 0.030077297791, 0.0282007574495, 0.033863719907, 0.033050423517, 0.0307423222473, 0.030509903522, 0.024979511976, 0.0279493741193, 0.0312742280747, 0.0226712091697, 0.0310798099133, 0.03118675818, 0.0299971871467, 0.027517609126, 0.0383080313662, 0.0274216835873, 0.0317109951135, 0.0281019089283, 0.033225526883, 0.0291657199305, 0.0306566293905, 0.0327373989065, 0.0348325292493, 0.0315469191795, 0.0348671191425, 0.0325448879755, 0.0298935860895, 0.0304467333765, 0.0284371759772, 0.0309737486685, 0.029729577894, 0.0301795539012, 0.029291424279, 0.0325247677492, 0.033622334954, 0.0297019012447, 0.0318481543717, 0.0313265857413, 0.0308693020323, 0.0323915585797, 0.0289017611357, 0.0321577830197, 0.0278262627123, 0.0305114449582, 0.0285186108945, 0.0293513839542, 0.031830889438, 0.031509032763, 0.0274477761142, 0.0271028366462, 0.0297565335242, 0.03248279573, 0.0304176390092, 0.0264395444605, 0.033106075518, 0.026113148902,  
0.032340056978, 0.029008171609, 0.0290341098982, 0.0303408750697, 0.0293805775922, 0.03171021293, 0.028744070727, 0.0276803829028, 0.0297883307477, 0.032145950338, 0.0289025024525, 0.030998750046, 0.0334560689193, 0.0327860582643, 0.0302882766793, 0.0300137764893, 0.030663118822, 0.0319853755132, 0.0351638289335, 0.0270146325495, 0.0324935145527, 0.0319067984605, 0.0278537655425, 0.0326348501755, 0.032373126405, 0.033141360277, 0.0332600789488, 0.0327914714263, 0.0298386243935, 0.0319611096777, 0.0343059085407, 0.0291653066707, 0.0344409566642, 0.0284622517933, 0.033585483755, 0.030371093276, 0.0322821096108, 0.0313927747075, 0.0333132238382, 0.0357923030505, 0.0332683289612, 0.0290572761143, 0.0356361236632, 0.027398539265, 0.0296664251753, 0.0272112828033, 0.030505058675, 0.031102794707, 0.031322856817, 0.0324160159408, 0.0335092602798, 0.0316599949625, 0.03529991054, 0.0282665238225, 0.0290026461145, 0.0314603444123, 0.032760574038, 0.0350207335343, 0.032949076276, 0.0354640723603, 0.0300394970485, 0.0297808299208, 0.0325614509385, 0.0281335580978, 0.0322394541912, 0.0297348795437, 0.0309115560482, 0.0291725268557, 0.0353826798258, 0.032777055423, 0.0339779006458, 0.03014045436, 0.0314676520333, 0.0323782267255, 0.0328870326697, 0.0311912002307, 0.0302912921458, 0.0323622743742, 0.0281228400622, 0.03531938135, 0.0322376515525, 0.0357475325968, 0.0319859960165, 0.037339830645, 0.0331519394538, 0.0325339431705, 0.037164277891, 0.0350746122, 0.0322576838023, 0.0339861455745, 0.0325381883075, 0.0329361355193, 0.0330218384898, 0.0320929891762, 0.0297956223645, 0.0317536955013, 0.032711785481, 0.0351765440788, 0.0311181387485, 0.0304094122015, 0.0301847255843, 0.031356710874, 0.036950248211, 0.033181302933, 0.0332838975887, 0.0344636813258, 0.0338261594642, 0.0302204413117, 0.0313193721813, 0.0332455537127, 0.0322315294542, 0.031945385737, 0.0337329308372, 0.0342213323042, 0.0312708777038, 0.033943299625, 0.03073641449, 0.0307447231745, 0.0336413385355, 0.0303286971585, 0.0344690595065, 0.0318857248505, 0.0310391254222, 0.030658445829, 0.0326751368657, 0.0362719545308, 0.031811516975, 0.02978882974, 0.0326402262085, 0.035492505239, 0.0289806391647, 0.0318026274928, 0.0298476063215, 0.0322472316877, 0.029932783173, 0.035769738265, 0.0373262528155, 0.0351015913355, 0.037895683226, 0.0384093755475, 0.0305572324475, 0.0289847845458, 0.0345930166623, 0.032200357355, 0.0334854762938, 0.0369566264907, 0.0323732668237, 0.0315310560275, 0.0335895147888, 0.0351606318207, 0.0370052677673, 0.0321469696422, 0.0334854544248, 0.033393222942, 0.0399975915448, 0.0298192896072, 0.037174031878, 0.0334316686445, 0.0331546963555, 0.0345708023935, 0.0336152165883, 0.0370166253925, 0.0318259720198, 0.0341467024015, 0.0326495332505, 0.035245534894, 0.0320742759435, 0.0324082248125, 0.0335827383688, 0.0374319952045, 0.0399748010387, 0.0367520331035, 0.032553910633, 0.0347369760783, 0.0364552726025, 0.0335421970585, 0.0371101686233, 0.0391068565412, 0.037543774814, 0.0390381380732, 0.036348081589, 0.037692411655, 0.036145483583, 0.036753937319, 0.0339558965473, 0.0383995669475, 0.0329661873607, 0.0391447281473, 0.037238973784, 0.032926773255, 0.0382053145303, 0.0297602867073