D5.3: Visual Search Tools – Report – Version B

Project ref. no.	FP7-ICT-610691
Project acronym	BRIDGET
Start date of project (duration)	2013-11-01 (36 months)
Document due Date:	2016-07-31
Actual date of delivery	2016-07-31
Leader of this document	Miroslaw Bober
Reply to	
Document status	Final (after QC)

Deliverable Identification Sheet

Project ref. no.	FP7-ICT-610691
Project acronym	BRIDGET
Project full title	BRIDging the Gap for Enhanced broadcasT
Document name	D5.3: Visual Search Tools – Report – Version B (First set of Visual Search Tools)
Security (distribution level)	PU
Contractual date of delivery	31/07/2016
Actual date of delivery	31/07/2016
Document number	
Type	Deliverable
Status & version	Version 1.0
Number of pages	27
WP / Task responsible	WP5
Other contributors (alphabetical order)	Massimo Balestri, Gianluca Francini, Sameed Husain, Skjalg Lepsoy, Simone Madeo, Alberto Messina, Stavros Paschalakis
Author(s)	Miroslaw Bober
Project Officer	Alberto Rabbachin
Abstract	This report describes version A of BRIDGET's Visual Search Tools.
Keywords	
Sent to peer reviewer	2016/07/27
Peer review completed	2016/07/29
Circulated to partners	2016/07/29
Read by partners	2016/07/29
Mgt. Board approval	N/A

Version	Date	Reason of change
0.1	2016-06-10	Miroslaw Bober – Initial draft
0.2	2016-07-21	Sameed Husain, Miroslaw Bober – expanded sections on RVD-W, improvements to structure, executive summary
0.3	2016-07-21	Massimo Balestri, Skjalg Lepsoy – added sections on Video DISTRAT and standardization activities
1.0	2016-07-26	Final integration, reference linking, sent for QC
1.1	2016-07-28	Skjalg Lepsoy - proofreading
1.2	2016-07-29	QC completed (Ingo Feldmann), comments implemented.

Table of Contents

Table of Figures

List of Tables

1 Executive summary

This deliverable presents the main technical developments and research achievements within WP5 – Visual Search Tools during the second phase of the project (Phase B: M18-M32). The key objective of WP5 is to develop advanced tools for automated Visual Search (VS) in image and video databases (Broadcast, Internet), enabling fast creation of content-based links for the Authoring Tools (ATs) and user-originated search capabilities for the Player. In the second phase (B), the main focus was on the development of visual-search engine capable of working with video libraries, and the development of BSOTA component tools for such engine.

This deliverable is structured based on active WP5 Tasks and focuses on main results of the research conducted between M18 and M32 of the project, and on the deployment within the BRIDGET pipeline.

The main achievement was a successful development of a complete video search engine, which has demonstrated BSOTA performance in the MPEG evaluation. The engine utilises the state-of-the art component tools developed in phase A, and tools designed or further extended in the phase B.

Robust Visual Search methods were advanced significantly in several areas. Firstly, our novel descriptor aggregation scheme called Robust Visual Descriptor (RVD) had been extended with local cluster whitening and a new normalisation scheme. We also redesigned the RVDW pipeline to work with deep CCN features, achieving further performance boost. Overall, our results advance significantly beyond SOTA in terms of recognition performance and speed, and formed the basis for a high-performance, scalable, binary global descriptor. Work also advanced on a new and fast approach to determine geometric consistency in video, called multi-frame DISTRAT. Finally, computationally efficient methods to analyse temporal variations in video and to derive a compact video-level descriptor, which minimise temporal redundancy have been developed. The tools were integrated into a complete search-engine pipeline, which was integrated into the BRIDGET authoring tool (AT), and formed the core base for consortium's response to MPEG CVDA Call for proposals.

On the standardisation front, the project supported successful finalisation of MPEG-CDVS standard and has been deeply engaged in the CDVA (Compact Descriptors for Visual Analysis) work. In the second phase, the project contributed video datasets and performed extensive annotation work, developed software evaluation framework for the CfP, managed the development of the first official experimental model, and presented two technical contributions to the CVDA pipeline. One proposal on reference frame selection has been accepted and is included in the current draft of the standard. 8 contributions to the MPEG CVDS/CDVA standardization work with significant impact were submitted, including (1) a response to the CfP with leading performance and (2) a successful proposal on temporal sampling in CE1 m38664.

2 Introduction

The objective of WP5 is to develop advanced tools for automated Visual Search (VS) in image and video databases (broadcast, Internet), enabling fast creation of content-based links for the Authoring Tools (ATs) and user-originated search capabilities for the player.

In the second part of the project the key objectives of WP5 were:

- complete development of global (aggregated) descriptors, matching algorithms and related tools for Visual Search and object recognition in large image databases, which are characterised by high true-positive detection rate at very low false-alarm rate and support ultra-fast matching;
- finalise development of binarisation strategies for local and global descriptors to enable fast matching;
- extend fast geometric consistency checking strategies to include video matching;
- develop techniques for Visual Search in Video with image-to-video and video-to-video search tools, including descriptor encoding methods, temporal aggregation, descriptor tracking and fast geometric consistency check in video;
- develop dedicated data structures and indexing schemes;
- participate and support completion of the ISO/MPEG standardisation work on Compact Descriptors for Visual Search (CDVS);
- continue participation in the Compact Descriptors for Visual Analysis (CDVA) group, leading efforts to design evaluation framework and annotated datasets for Visual Search in Video. Further, prepare a technical solution to Call for Proposal.

We first outline the current algorithmic flow for the Visual Search Engine develop in the project and then present the key innovation and achievements in the main elements of the processing pipeline.

As per project plan, in the reporting period the work concentrated around the following five areas:

1. Development of the extension to video of descriptors for images, encoding methods, temporal aggregation and effective comparison strategies for video (TI, UNIS, RAI and VA).
2. Further improvements to local descriptor aggregation RVD, including integration with deep features, in order to obtain a global descriptor with beyond the state-of-the art recognition performance (UNIS);
3. Data structures and fast indexing methods (UNIS, VA);
4. Extension of the DISTRAT tool for the geometric consistency check, developed in the 1 phase of the project, to enable processing multiple frames in video (TI), and
5. Contribution to the CDVS and CDVA standardisation activities (TI, VA, UNIS).

The following sections describe technical achievements in each of the above areas. Section 3 presents in greater detail the pipeline design of the BRIDGET visual search engine and explains the technical choices made. Section 4 discusses compact representations of video content and outlines two approaches to spatio-temporal sampling of video studied in the second phase of the project. The advances in robust aggregation methods, based on the extended RVD-W descriptor are presented in Section 5. This section also shows the latest results for aggregation of the deep descriptors. Section 6 present multi-frame DISTRAT for geometric verification, while Section 7 details standardisation activities and contributions. Conclusions are presented in Section 8.

3 The BRIDGET Visual Search Engine

3.1 Introduction

The objective of WP5 workpackage is to develop a low-complexity and high-performance visual search technology for video content. Please refer to [1] for the description of the visual search engine for images, developed in the first phase of the project (phase A). Work in the second phase, addressing search in video, is substantially based on the results of the first phase (A), which also contributed to the MPEG CDVS standard. There are, however, additional tools, extensions of the methods developed earlier, and a new video-search pipeline, which were developed in Phase B. The phase A solution served as the base to integrate further tools to enable work with video content (as opposed to image only content). The adopted approach is frame-based and exploits video redundancy along the time axis to optimise the way in which CDVS descriptors are extracted.

Before we introduce BRIDGET solution, we review the state-of-the art in video search.

3.2 State of the Art in Video Retrieval Systems

In the last decades many researchers investigated visual techniques, especially in the image domain, with notable results such as SIFT descriptors [2] We also witnessed the birth of new standards, such as the MPEG standards ISO/IEC 15938:13 [3] and ISO/IEC 15938:14 [4] which introduced CDVS descriptors. Then the focus has shifted to contents search in video and new challenges had to be addressed. In this section the state of the art of video matching and retrieval techniques will be described and a selection of the most recent papers will be presented, highlighting their relevant and innovative features.

Video retrieval systems aim to assist users to retrieve one or more video segments within a database starting from a query. Retrieved segments are usually the shots that are visually similar to the query, which can take a form of a text, an image (or a region of it), a video segment depicting object(s) of interest, or a combination of them. Video retrieval systems are typically divided into two categories: near duplicate video retrieval (NDVR) and content based video retrieval (CBVR).

3.2.1 Near duplicate video retrieval systems

In a near duplicate video retrieval system, a video segment (sometimes a frame) is submitted as a query, then the system searches in a database for the original video to which the query belongs. The system has to recognize the video source even if it differs from the query by format (encoding, frame rate, bit rates, etc.), photometric variations (colour or light changing) and editing operations (inserted logo or added borders).

This functionality is usually supported by suitably selected low-level features, which robustly represent underlying frames. NDVR systems mainly aim to solve two critical issues: detect copyright infringement and search for multiple copies of the content to reduce storage requirements. In [5] a NDVR system named ASVT (Adaptive Structure Video Tensor) is presented. It uses a tensor model to represent video as a series of 3D structure tensors. In the elaboration process, key-points are extracted from selected key-frames using SIFT based algorithm (PCA-SIFT), then each local descriptor is transformed into a probability density function (PDF). The set of PDF from each key-frame is represented by 3D structure tensors. Retrieval is performed using an R-tree indexing to compare query descriptors with the counterparts in the database.

In [6] a system based on spatio-temporal pattern programming is presented, which is capable to retrieve not only near duplicate video but also the precise temporal position of query segments in the reference videos. Firstly, low level features are extracted from frames and are encoded as sequences of symbols to obtain an index pattern, further an m-pattern is built from key-frames (subsampled frames). Then a spatio-temporal indexing structure, named Pattern-based Index Tree (PI-tree), is used to keep only partial near-duplicate videos. An m-Pattern-based Dynamic Programming (mPDP) method estimates the spatio-temporal similarity between video segments. Results are re-ranked in an additional post-filtering stage.

In [7] a NDVR system based on Earth Mover's Distance (EMD) is presented. The EMD is a transformation-based approach, which measures the cost of transforming one feature signature into another one, so authors exploit this property to compare local and global descriptors. To obtain signatures, key-frame features are extracted and then the feature space is clustered. The most representative features become the signature of the key-frame.

In [8] a Fisher vectors based NDVR system is proposed. A set of key-frame in selected in the video and SIFT descriptors are extracted. All descriptors in each shot are aggregated in a single Fisher vector used to compare and retrieve video segments.

In [9] a NDVR system based on Zernike moments is proposed. Zernike moments are selected as key-frame feature because they are scale invariant and resistant to rotation and noise. A Zernike moments similarity score is used to compare the query with the reference videos in the database.

3.2.2 Content based video retrieval systems

Content-based video retrieval systems take video queries as input and retrieve the most relevant videos from a database according to the query content. They can use low-level features (colour, texture, motion, speech [10] etc.) and high-level features (metadata, text, etc.). Given the specific nature of WP5 in BRIDGET, this section will address only systems based on low-level visual features.

CBVR systems have a wide range of applications such as quick browsing of video folders, analysis of visual electronic commerce, remote instruction, digital museums, news event analysis, intelligent management of web videos and video surveillance. In the recent years this problem has become more and more relevant with the increase of video contents stored on the Internet and in the archives. TRECVID, MPEG and other groups have promoted researches in this field.

In [11] a Bag of Word object retrieval architecture is proposed. After a video subsampling, local features are extracted from key-frames using SIFT descriptors. A vocabulary of visual words is created by performing K-means clustering of the descriptors, and each frame is represented by a visual word frequency histograms over the vocabulary. To perform retrieval, the user has to select a query region from an image which likely highlights an object. Then visual words are extracted from the query region and using their frequencies in the vocabulary the matching key-frames are returned. Finally, results are re-ranked using spatial consistency check. BoW-based architectures are not well suited to handle large-scale vocabulary because complexity grows linearly with the number of frames in the database. In response to this issue, many improvements to BoW have been provided in time, introducing:

- vocabulary tree [12] ;
- spatial pyramid kernel [13] ;
- Locality-constrained Linear Coding (LLC) [14] ;
- Vector of Locally Aggregated Descriptors (VLAD) [15] ;
- Fisher kernels [16] ;
- Genetic Programming [17] .

In [18] a CBVR system based on temporal pattern is proposed. After a process of video subsampling, shot identification and key-frame detection, colour, shape and motion are selected as features and are extracted and compressed using a DCT and a ZigZag scan. Shots are clustered using a K-means algorithm and a symbol is assigned to each cluster, where each key-frame receives the symbol of its cluster, so shots are now encoded as a sequence of symbols (temporal patterns). Fast Pattern Index (FPI) tree and Advanced Fast Pattern Index (AFPI) tree are built using temporal patterns. In Retrieval mode, the query video is indexed as described before, and a FPI (or AFPI) tree is redrawn while the database is updated by query data. The normalized pattern frequency represents the matching score.

In [19] a CBVR system based on salient regions is proposed. Salient regions are perceptually the most distinguishable parts in an image compared to their surroundings. In the first step, exploiting spatial local contrast, a saliency map in the frequency domain is computed for each frame. Then salient regions (objects) are detected and basic spatio-temporal (BST) feature is extracted. BST feature is the result of a pixel tracking operation that results in a 7D spatio-temporal features vector composed of 3 colour features, 2 motion angles, and 2 displacement components. The most stable features vectors are aggregated in normalized feature matrix. Retrieval is performed comparing the query and the database videos by the Hausdorff distance of their features vectors (similarity score).

In [20] a colour features based video retrieval system is presented. Firstly, the database has to be filled, so video content is temporarily subsampled at 1:20 ratio and selected frames become key-frames. Red, blue and green pixel components, extracted from each key-frame, are arranged in separate vectors. Based on two thresholds vectors, entries are divided by intensity in three groups, they are truncated and encoded using a Thepade's Sorted Ternary Block Truncation Coding level 2 (TSTBTC-2), which reduces the amount of data, keeping salient colour information. The results are merged into a single vector and stored in a database. In retrieval mode a video query is submitted and the same process described before is performed. Query features vector obtained is compared with database features vector using a user selected similarity measure. Most relevant matches (video reference) are retrieved.

The following CBVR systems use images as queries instead of video. In [21] a CBVR system based on Fisher Vectors (FV) is presented. Video is segmented into shots exploiting colour histograms and the median frame of each shot is selected as a key-frame. For each key-frame the Hessian-Affine regions are computed and represented by RootSIFT [22] descriptors. Then a local features tracking is performed to reduce the temporal redundancy. The evolution in time of a feature is named thread and using the Principal Component Analysis (PCA) the threads dimensionality is halved. Each thread is represented by a FV using descriptors as measurements and the Gaussian Mixture Model (GMM) as probabilistic model. A further PCA compression is performed to FV for footprint reduction. Comparison between the query frame and reference videos is done computing the inner product of the query FV with every reference FV. The result of the product is the matching score.

A Kirsch descriptor based CBVR system is presented in [23] Using Gabor Moments features the video is segmented in shots. For each shot a key-frame is selected as the most similar frame to a Temporally Maximum Occurrence Frame (TMOF), which is obtained by the analysis of the spatio-temporal distribution of the pixels in the shot. Then the FAST detector [24] detects the key-points in the key-frames. Kirsch descriptors extraction begins by placing a square window (16x16 pixel) centred in each key-point and rotating it in agreement with the key-point orientation. In the square windows domain, the Kirsch features are extracted that represent the directional edge features for horizontal, vertical, right-diagonal and left diagonal directions respectively. During retrieval each descriptor of one of the query frames is compared with the database clusters using the K-nearest neighbour searching algorithm.

The video segment retrieval system presented in [25] is based on shot affine hull, which is defined as the smallest affine subspace containing frames that represent the entire shot. Input video is segmented into shots, and for each frame colour histogram is selected as feature representation. Shot feature vector are collected in a matrix and using a Singular Value Decomposition (SVD) two vectors are obtained: the basis vector representing the affine subspace and the mean vector which represents the average of the features. Query frame and reference shots are compared by L2 distance (similarity score) of their affine hull representation.

Descriptors compression

One of the main problems with video content is the amount of data that has to be processed and stored, so visual feature compression is crucial. Some interesting compression techniques can be found in the algorithms described previously, [18] [20] [21] [25] but two more compression techniques should be presented.

In [26] the author focuses on visual descriptors coding in video, exploiting spatio-temporal frames redundancy. A GoP (Group of Picture) coding structure is proposed. Every GoP is made of an I-frame followed by a number of P-frames. I-frames are self-contained and managed as single images, so features extraction and coding techniques are the same as for images. However, for P-frames local features encoding is related to neighbouring frames. Within a GoP, features variation between consecutive frames is small, so a differential coding can be used. The proposed method extracts the local descriptors from a P-frame and searches the best matching in the neighbourhood (group of frames in a time window near current P-frame). When a descriptor matching is identified current local descriptor is substituted by a smaller reference descriptor. Furthermore, binary descriptors and entropy encoding are used to improve features compression.

In [27] an inter-frame visual descriptor coding is presented. Authors aim to use compressed representation of canonical patches as local descriptors. The video frames are divided into two categories: Detection frames (D-frames) and Forward Propagation frames or (FP-frames). Also patches are divided in two categories: (i) D-patches are associated to key-points found using SIFT detector, and (ii) FP-patches which are the result of tracking. Tracking is performed by searching in the current frame the patches found in the previous frame, so no key-point detection process is necessary. The use of D-patches and FP-patches is controlled by an adaptive algorithm. D-patches are intra coded using a trained Patch ENCoder (PENC), which uses Huffman table to associate canonical patches to the most similar gradient in the table. FP-patches are coded using only residual signal difference or can be a simple reference to obtain more compression. Also patch locations are encoded differentially.

3.3 The BRIDGET Visual Search Engine

The BRIDGET Visual Search Engine employs a classical processing flow and is based on the CDVS architecture. Figure 1 shows the current descriptor extraction pipeline. In order to achieve our objectives of significantly improved recognition and high processing speed, WP5 has developed several components with BSOTA performance, namely the ALP detector, RVD-W aggregation, multi-frame DISTRAT. A fast temporal key-frame selector was adopted from WP4.

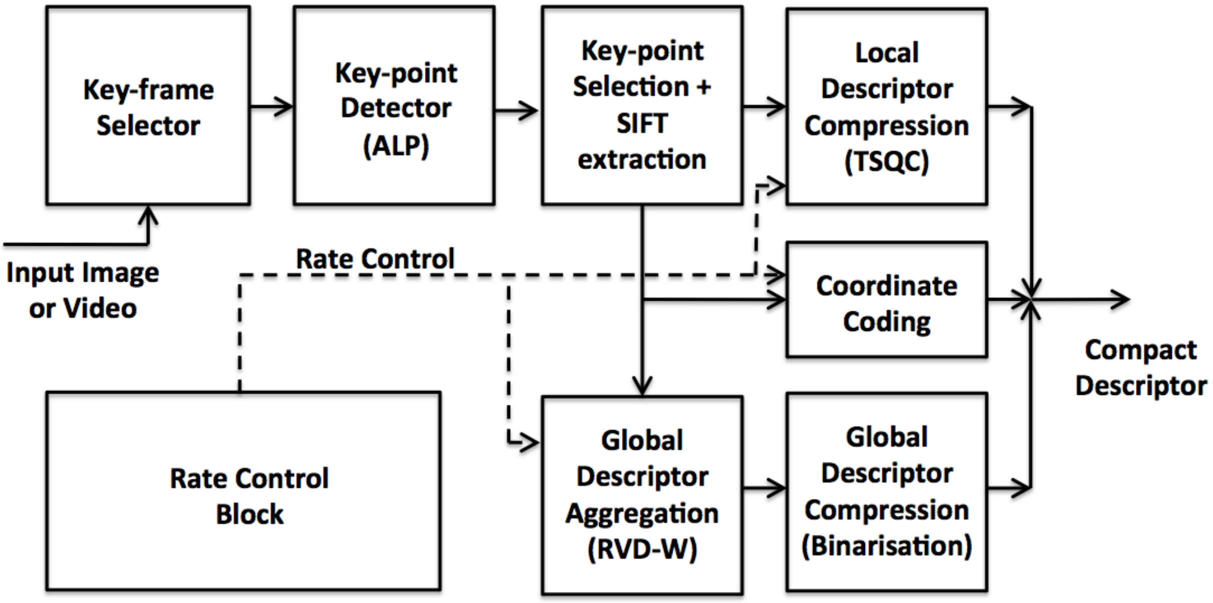

Key-frame Selector
Key-point Detector (ALP)
Key-point Selection + SIFT extraction
Local Descriptor Compression (TSQC)
Input Image or Video
Rate Control
Coordinate Coding
Compact Descriptor
Rate Control Block
Global Descriptor Aggregation (RVD-W)
Global Descriptor Compression (Binarisation)

Figure 1. BRIDGET Visual Search Engine descriptor extraction pipeline.

The current BRIDGET pipeline first selects a subset of reference frames, using a fast and simple change-detector, described in Section 4. Subsequently, descriptors for each key-frame are extracted, based on the CDVS-like approach, developed in the first phase of the project. For each key frame, feature-points are extracted using ALP detector. A subset of more robust features is selected, in the Key-point selection block, and a subset of the corresponding SIFT descriptors are selected. The descriptors are compressed using the Transform and Scalar Quantisation Coding method (TSQC), which offers a scalable, high-performance compression with ternary representation. Global descriptors are computed using the improved RVD-W aggregation method, extended in the second phase of the project, as described in Section 5. The ALP detector, key-point selector and TSQC compression were integrated in the first phase of the BRIDGET project, and now form a part of the MPEG CVDS specification. We also re-designed and extended DISTRAT geometric verification to video sequences, as detailed in Section 6. The design of BRIDGET video-search engine was submitted to the MPEG call for proposal, achieving leading performance. The work within MPEG, our response to CDVA CfP, and the experimental performance evaluation of our submission is described in Section 7.

4 Compact and efficient representation of video content

In the first phase of the project it was decided that in order to maintain interoperability with the BRIDGET image search engine, the video search engine should utilize as many components of the image search architecture as possible. We therefore focused our investigation on an efficient way to represent the content of a video, using frame-level based description. In this section we present two broad approaches we investigated. In the first one, the temporal sampling (i.e. key-frame selection) is based on temporal variations in the frames detected using either a global descriptor, or via fast frame comparisons based on histograms. The second approach is more sophisticated and investigates improvements to the efficiency of the video description by using motion-filed characteristics to select frame and frame-regions as a reference.

4.1 Change-based temporal sampling

In order to represent the video in a compact way, only a subset of the original frames are used to form the CDVA bit stream. We investigated two techniques for selecting these key frames; both detect change in frame content over time.

The techniques have been integrated into the CDVA experimentation model (CXM), tested with the CDVA Evaluation Framework, and submitted to MPEG in the context of Core Experiments in CDVA.

4.1.1 Temporal sampling based on RVWD descriptor

In the first technique, frame content is represented by the RVD-W global descriptor [28] The use of this descriptor for change detection was presented in [29] . Details of the procedure are found in Section 4.2.2, “Keyframe selection based on variation of the RVD descriptor” in the BRIDGET report [30]

This process requires every frame to be processed: local descriptors must be computed and aggregated into a global descriptor. The complexity of these operations causes the CDVA extraction to exceed the limits set forth in the Evaluation Framework of MPEG CDVA [31] and therefore we continued the development of an alternative, faster temporal sampling technique, based on histograms.

4.1.2 Temporal sampling based on colour histograms

In this faster technique, video frame content is represented by colour histograms. The measure of change is the sum of l2-norms of the differences between normalized histograms of the R,G,B components of the video frames. The technique is described in Chapter 7 “Standardisation activities: Proposal submitted to MPEG CDVA”. It was submitted to MPEG CDVA in [32] and accepted as a tool of the CDVA Experimentation Model.

4.2 Motion field based temporal and spatial sampling

The approach investigated here aims to exploit detailed motion information to determine optimal spatio-temporal sampling for a video. The objective is to remove redundancy and to optimise the way in which frame-level descriptors are extracted. This objective is achieved in two ways: by devising a key frame sampling algorithm based on camera movement detector, and by applying a motion attention filter to isolate spatiotemporal regions of the video. The results of these two approaches are respectively called shot motion adapted key frames and motion attention based masked images. Detailed description and results obtained are presented below.

4.2.1 Overall architecture

The architectural approach is summarised by Figure 2.

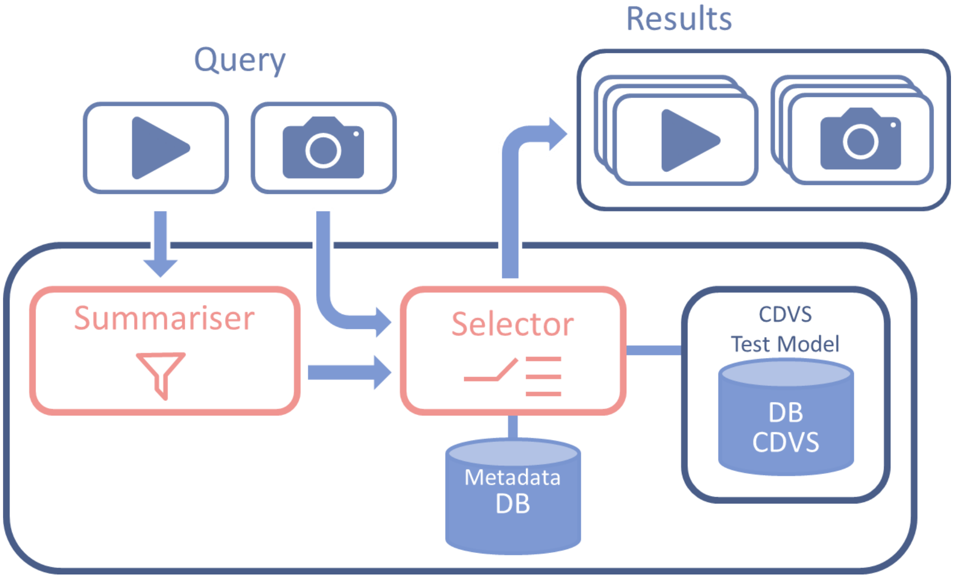


Figure 2. Proposed architecture for video summarisation and indexing.

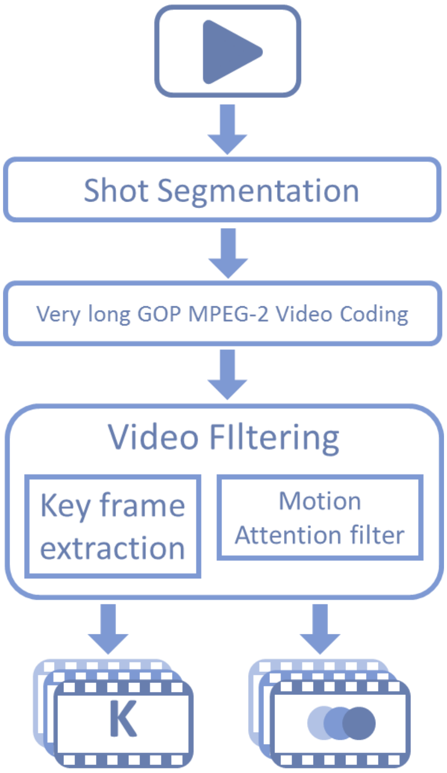


Figure 3. Video segmentation.

The query object can be either a still picture or a video. In case of a video, this goes through a video Summariser, which applies the processing steps illustrated in Figure 2. At first, a segmentation into shots is performed using a differential feature tracking approach based on a multiset of features, namely PSNR, luminance correlation and dominant colour distribution difference. Each resulting segment is processed by an MPEG-2 encoder using a very long GOP structure (IPPP…P)[1]. The resulting encoded stream is then processed by two parallel video filters, each providing a set of visual objects summarising the input video. The first filter extract key frames from the segment by considering the shot type (e.g., zoom, pan, static). Zoom and pan detection is implemented through a motion vector orientation analysis, downstream of a filtering, and accumulation step to remove noise; a Canny filter is also used to retain motion vectors only for textured and detailed regions of the image. The second filter applies a motion attention model extended from [33] . The objective of a motion attention filter is to extract from the video flow those spatiotemporal regions where we expect more attention from the viewer, under the heuristic assumption that focus of attention is attracted by "unusual" motion w.r.t. the global motion trend. Applying a motion attention filter on a video input results in the detection of a series of binary picture masks highlighting these spatiotemporal regions. Downstream the motion

[1] The method can also be applied if other more advanced coding techniques are used instead of MPEG-2 for motion compensation.

attention filter, a series of morphological operators is applied to the binary masks in order to regularise their shape and temporal evolution on the video timeline.

Finally, on the two series of visual objects, the classical CDVS extraction pipeline is applied and the results are compared by the Selector component to the previously saved CDVS database, built according to the standard, and a Metadata database retaining information about the nature of the indexed visual objects (i.e., if each indexed object is a key frame or a mask, from which portion of the reference video was extracted). Corresponding information about the query video are saved in the Selector memory in order to organise the results of the search.

4.2.2 Motion Vector Analysis and Filtering

In this Section some details are given about the performed MPEG-2 motion vector analysis used to extract shot motion adapted key frames from the video for subsequent CDVS indexing (left part of the processing pipeline of Figure 3). Due to the imprecision of the MPEG-2 motion compensation algorithm the non-zero motion vectors located in the picture border or in flat areas of the borders should be eliminated. This is done through the application of a Canny filter [34] Results are exemplified by Figure 4 and Figure 5.

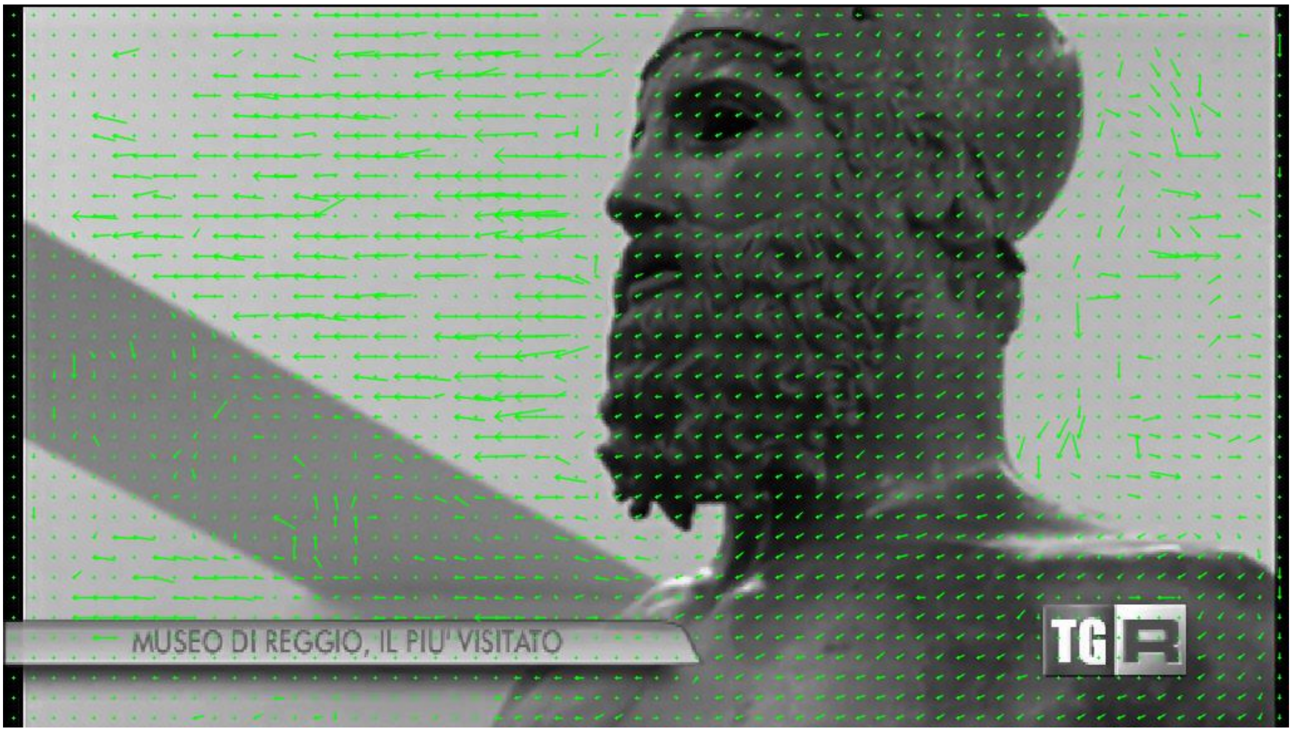


Figure 4. Original motion vector map on a P frame.

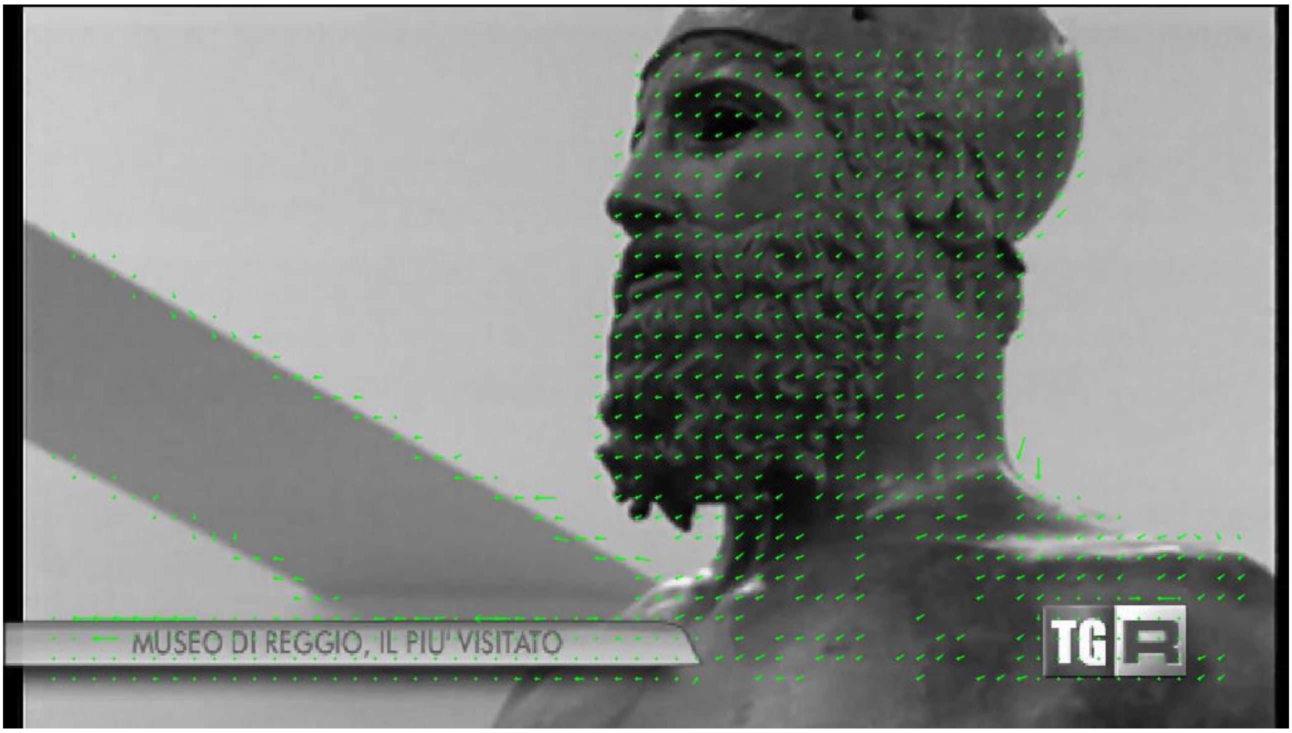


Figure 5. Filtered motion vector map.

After this step, a motion vector accumulation step is applied through a median filter, whose results are exemplified by Figure 6.

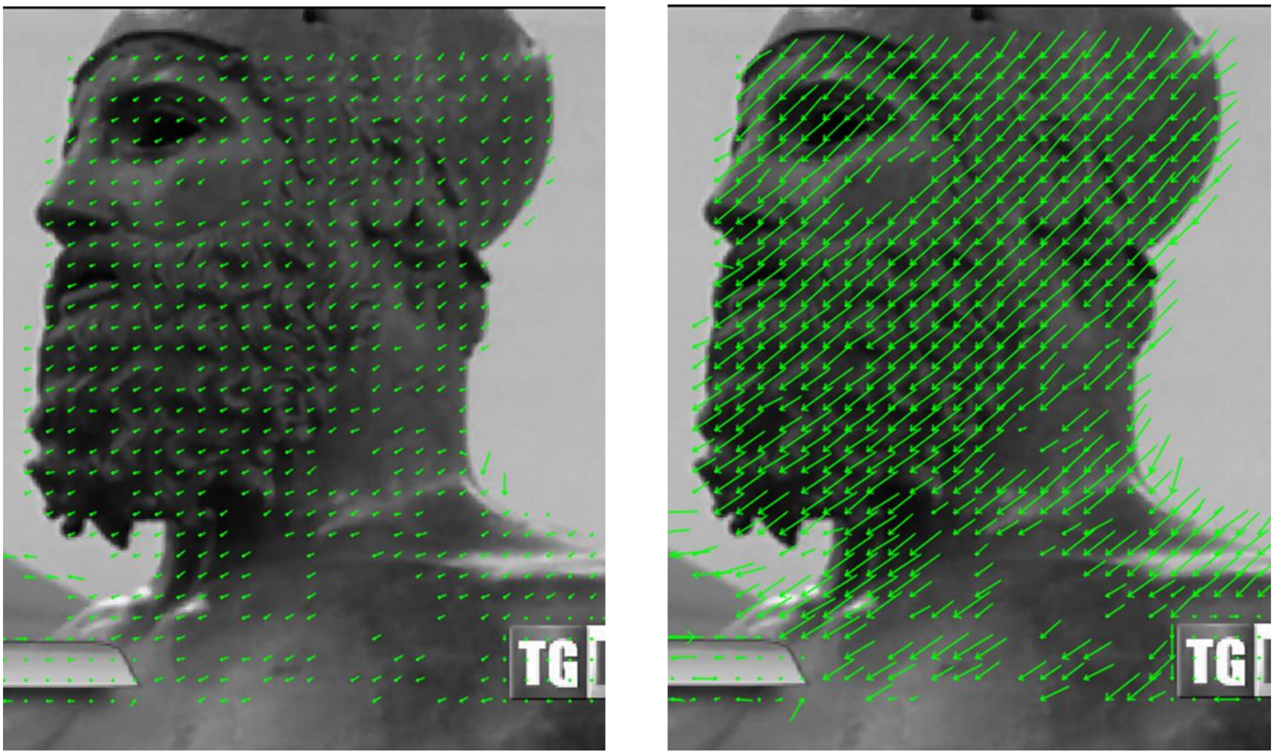


Figure 6. Results of the motion vector accumulation step.

The video content is now ready for the subsequent phase, consisting in camera zoom classification aimed at detecting the dominant zoom motion in the current shot and making the appropriate decisions about which keyframes of the shot to retain for visual indexing. This phase implemented using a two-step approach: 1) detection of the candidate zoom centre; 2) classification of the zoom type (in, out or undecided). The candidate zoom centre is detected in three steps:

1) Motion vector random subsampling, aimed at lowering the computation effort;
2) Calculation of intersection point for each couple of motion vectors in the subsampled motion vector set;
3) Calculation of the candidate zoom centre as the average intersection point.

Once the zoom centre is found, an estimated zoom map is calculated by associating to each macroblock of the picture a unitary vector whose direction points to the candidate zoom centre. Then, each actual motion vector is classified as contributing to “zoom in”, “zoom out” or “undecided” depending on its deviation from the local unitary vector. A shot is then respectively classified as “zoom in”, “zoom out” or “no zoom” depending on the majority of classified motion vectors. Figure 7 shows an example.

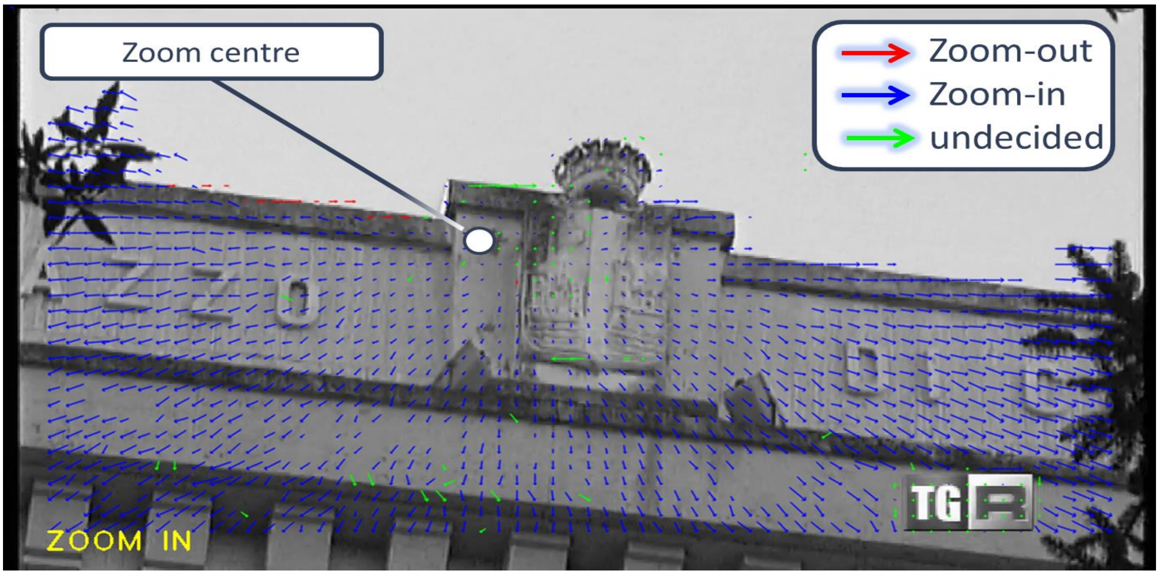


Figure 7. Example of shot zoom classification.

All of the described processing steps on the MPEG-2 motion vectors are aimed at the extraction of shot motion adapted key frames, i.e. key frames maximising the stability of the represented picture and coverage of the scene. The used heuristics to sample key frames from the current shot is as follows: 1) we extract the first frame of a shot classified as zoom-in; 2) we extract the last frame of a shot classified as zoom-out; 3) in addition we extract one extra frame every time a camera movement is detected to be equal to the height of the picture during the shooting. This latter parameter is calculated frame by frame by adding the average motion vector value of each frame to an accumulator which is reset at each shot.

4.2.3 Motion Attention Filtering

Motion attention filtering is aimed at extracting picture masks isolating moving regions in video shots which have some peculiar motion characteristics that make them more attractive to an observer. This is done in order to optimise frames and frame regions on which to extract CDVS descriptors, thus resulting in a more efficient system. This is done in steps as follows: 1) macroblock level median motion vector estimation; 2) motion attention global descriptor calculation; 3) target region search.

Macroblock level median motion vector estimation is performed by evaluating the average L2 distance between the motion vector of the macroblock and those of its 8 neighbour macroblocks, and subsequently selecting the motion vector minimising this distance among the neighbours. Figure 8 reports an example of application of this filter.

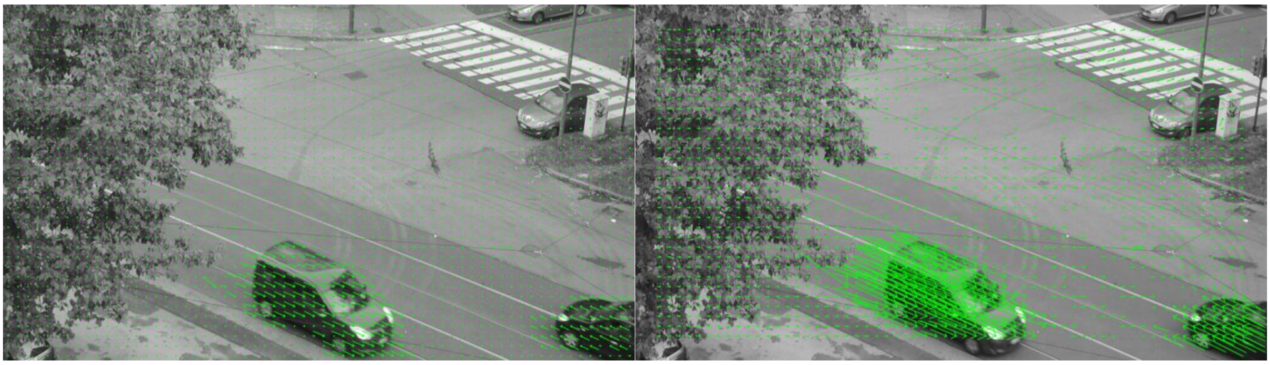


Figure 8. Original motion vector map and filtered motion vector map.

The macroblock level motion attention global descriptor $A_{ij}$is calculated as follows:

$$A_{ij} = \alpha L_{T_{ij}} + \beta L_{S_{ij}} + \frac{1}{2}\delta \max\left(L_{T_{ij}}, L_{S_{ij}}\right)\left|L_{T_{ij}} + L_{S_{ij}}\right|$$

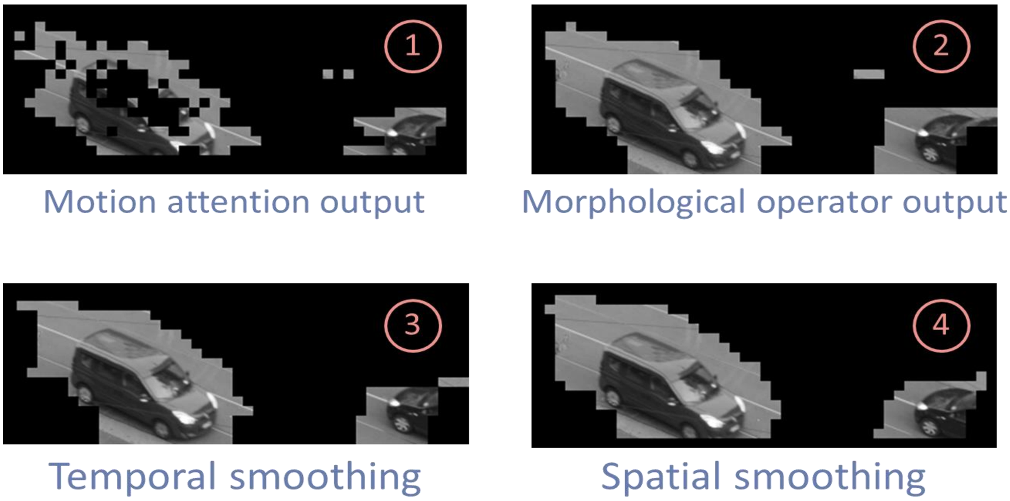
where $L_{S_{ij}}$ is the median motion vector spatial correlation, $L_{T_{ij}}$is the median motion vector temporal correlation, and α, β and δ are weighting parameters regulating the importance of each of the three terms of the above Equation.

The macroblock level median motion vector spatial correlation $L_{S_{ij}}$ is calculated as the absolute difference between the median motion vector of the macroblock and the average median motion vector of its neighbour's macroblocks.

The macroblock level median motion vector temporal correlation $L_{T_{ij}}$ is calculated as the absolute difference between its value at frame *i* and its value at frame *i-1*.

Once that the macroblock level motion attention global descriptor A_{ij} is calculated, the next step is that of identifying the target regions, i.e. macroblock connected agglomerations that retain a level of global motion attention level exceeding a certain threshold over a consecutive number of frames. In this case, instead of applying the originally proposed scheme of [33] with a fixed threshold, we opted for an adaptive threshold scheme in which the threshold is calculated as the average value over all macroblocks of the shot.

The last step of the algorithm consists in the application of a dilation morphological filter aimed at regularising the found target regions by removing holes and a subsequent spatiotemporal smoothing filter of the regularised regions. Figure 9 illustrates an example of the application of such filters.



Motion attention output

Morphological operator output

Temporal smoothing

Spatial smoothing

Figure 9. Example of morphological and spatiotemporal filter applied on a motion attention target region.

4.2.4 Evaluation

The tests conducted to evaluate the developed methodology were meant to analyse the impact of the extracted shot motion adapted key frames and motion attention based masked images on the scores obtained by applying the CDVS test model retrieval and compare the scores with the ones obtained when using a baseline frame subsampling approach to obtain key frames (baseline key frames). We used two datasets to perform this analysis: a dataset of 80 videos containing monuments, namely a subset of the "Monuments of Italy" dataset documented in [1] and 15 videos containing vehicles created on purpose. For each of the videos a set of query images has been collected matching the objects present in the videos. Therefore, the starting condition of the test was that each of the collected query images matched a subset of one of the two video datasets and that this information was available in advance. Thus the test only evaluated the change in the retrieval results depending on the employment of baseline key frames and/or shot motion adapted key frames w.r.t. when only baseline key frames were employed.

A preliminary measurement was done in order to estimate what level of compression of visual information was achieved by comparing the amount of data extracted (average on the data set) by the motion attention based masked images and the shot motion adapted key frames w.r.t. the baseline key frames. These latter were subsampled at a rate of 4 key frames per shot. Figure 10 shows the obtained numbers.

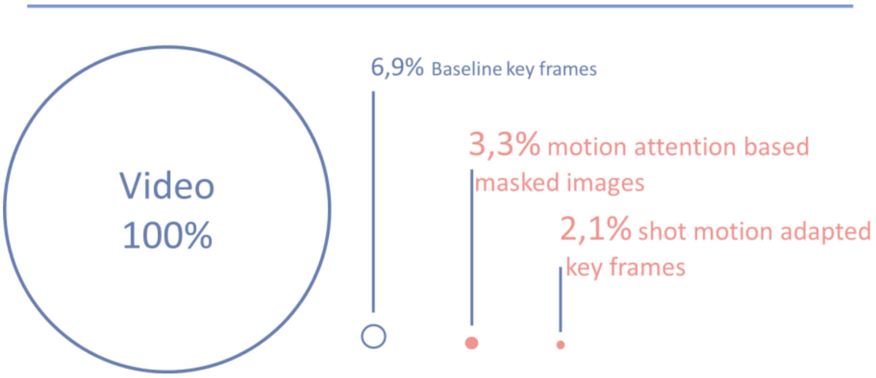


Figure 10. Size ratio between full video, baseline key frames at 4 key frames per shot, motion attention based masked images and shot motion adapted key frames.

We performed a total of three tests. The first one was aimed at evaluating how the three different types of extracted images behaved in terms of retrieval performance for a certain known query. We defined a quality score as the ratio between the CDVS retrieval score of the true matches and the number of corresponding retrieved images for the three categories of images. When this quality measure is higher it means that less images are ranked very high, i.e. the corresponding image type is more discriminative than the others in the average. This has a positive effect on the precision of the results in cases of low false positive system configurations (e.g., for higher CDVS retrieval score thresholds). Figure 11 reports the results for the monuments dataset, in which case the shot motion adapted key frames showed the best performance among the three types of images.

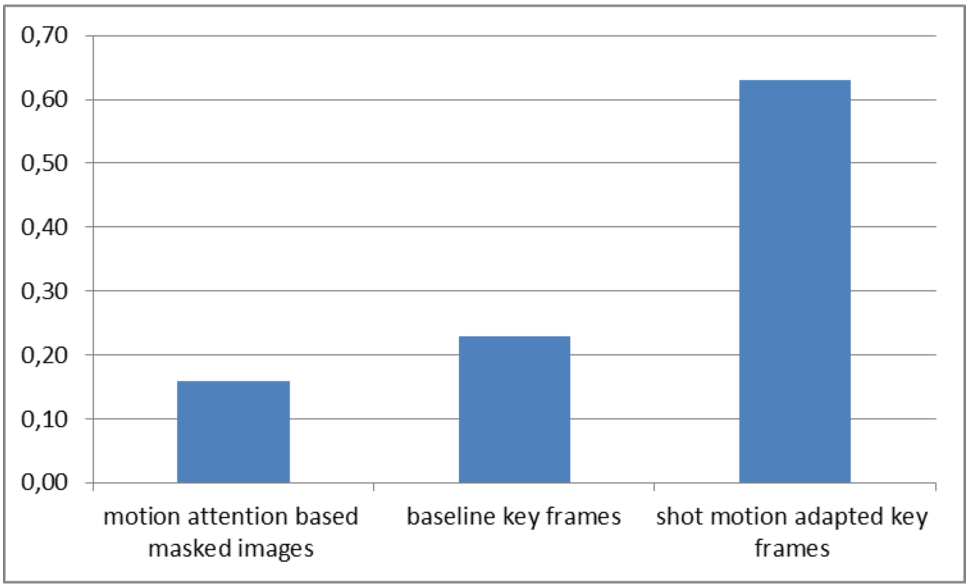


Figure 11. Quality of retrieved results vs. indexed image type (monuments dataset).

In the second test we compared the score obtained by a certain frame and of its corresponding motion attention based masked images in the true match cases, using the vehicle dataset. The results are shown in Figure 12. The picture reports the percentage of cases in which the whole frame (left column) or the corresponding motion attention based masked images (right column) had the higher score. The result seems to show that the motion attention based masked images are better focused in representing the shot content than the corresponding whole frame, and thus have the effect of increasing the recall.

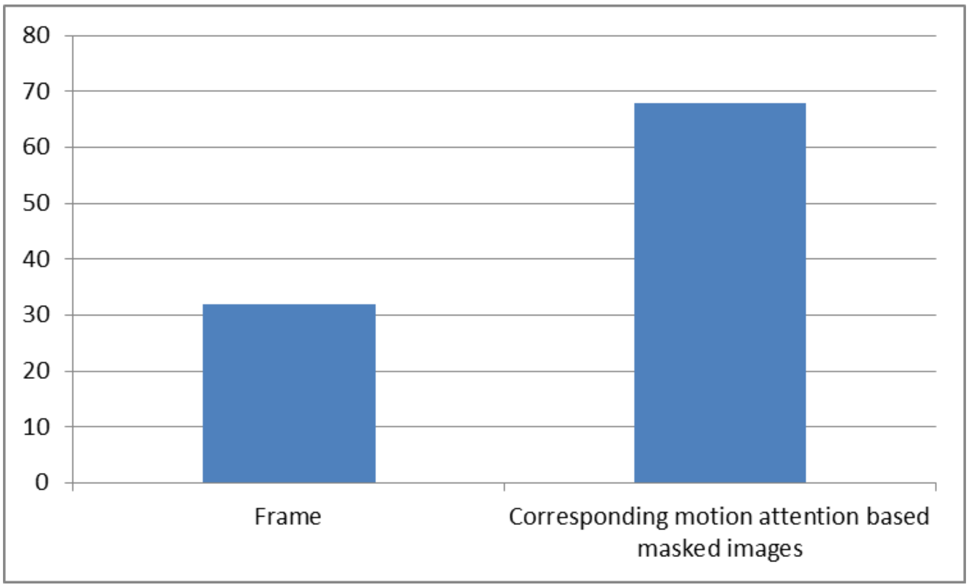


Figure 12. Distribution of winning image type (vehicle dataset).

The third test consisted in evaluating the obtained average CDVS retrieval score of baseline key frames and of motion attention based masked images in true match cases, using the vehicle data set. The results are shown in Figure 13. The average increase of about 1.4 score value w.r.t. baseline key frames obtained by the motion attention based masked images indicates that the usage of these masked images has a positive effect on the recall since it increases the probability to retrieve true matches at a given CDVS score threshold.

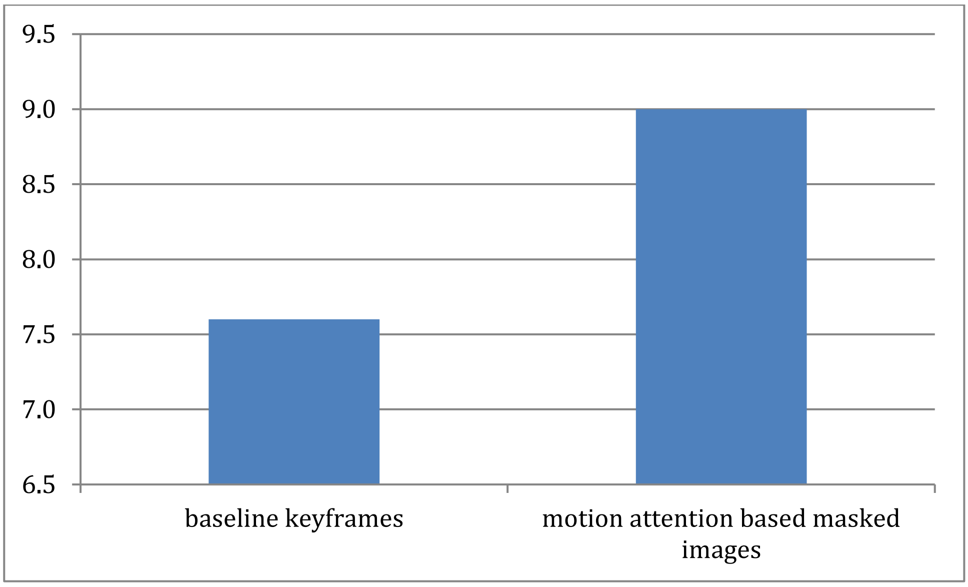


Figure 13. Average CDVS retrieval score vs. image type (vehicle dataset).

4.2.5 Conclusion on the motion-based sampling

We developed a novel approach to temporal and spatial sampling of video content, introducing two processing pipelines. They extract motion-attention based masked images and shot motion adapted key frames downstream of a basic video segmentation algorithm based on a low level feature tracking approach. These images, once extracted, are indexed by the standard CDVS pipeline and used in the reference retrieval architecture of the standard. We made some preliminary tests showing that collectively the usage of these two kind of images in a classical CDVS retrieval system are expected to increase the quality of the retrieval, having positive effects both on the precision and on the recall. Although promising results were obtained, the method has high computational complexity and exceeds the extraction time limits specified by the CDVA. It was therefore decided that for the BRIDGET engine and our CDVA proposals the consortium should focus on a simpler design presented in Subsection 4.1.

4.3 Conclusions

We have investigated two approaches to extraction of frame-level descriptions for the video content. While the motion-flow based approach extracts more detailed information from the sequence and has the potential to better remove redundancy, it also has substantially higher computational complexity. It was therefore decided that a faster approach, based on histograms, should be employed in the BRIDGET search engine and in our submission to CDVA. As the available computational resources increase with time, the more sophisticated solution, employing the spatio-temporal analysis based on motion may become practicable.

5 RVDW – Extended Robust Aggregation of Local features

5.1 Extended aggregation mechanism

In the first stage of the project we have developed a novel aggregation method, Robust Visual Descriptor, with BSOTA performance. In the second stage we have further extended our solution and designed a new descriptor – called Robust Visual Descriptor with Whitening (RVD-W). It combines rank-based multi-assignment with robust aggregation framework and cluster-wise whitening. Compact codes can be obtained by product quantisation approach or by sign-based binarisation. Furthermore, to enable descriptor size scalability new cluster-level and bit-level element selection mechanisms were developed. RVD-W pipeline is shown in Figure 14.

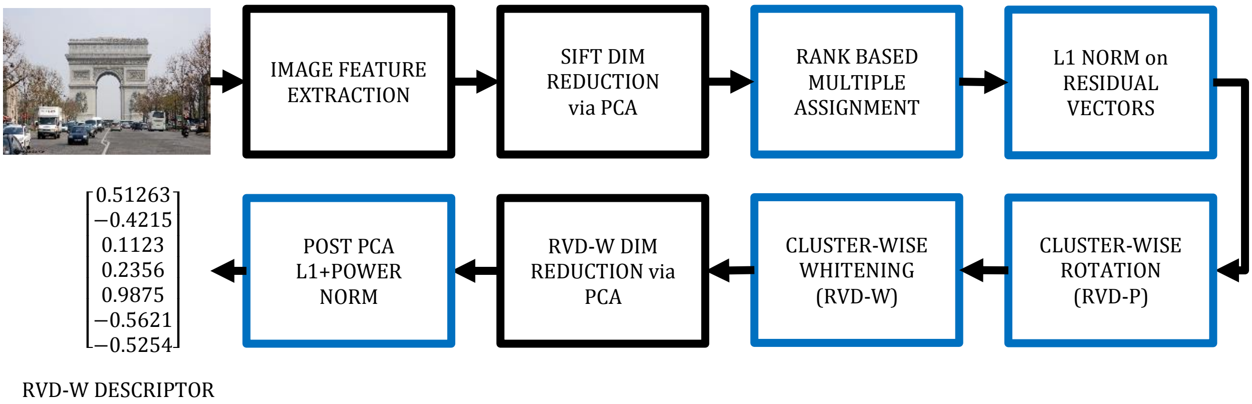


IMAGE FEATURE EXTRACTION
SIFT DIM REDUCTION via PCA
RANK BASED MULTIPLE ASSIGNMENT
L1 NORM on RESIDUAL VECTORS
CLUSTER-WISE ROTATION (RVD-P)
CLUSTER-WISE WHITENING (RVD-W)
RVD-W DIM REDUCTION via PCA
POST PCA L1+POWER NORM
0.51263
−0.4215
0.1123
0.2356
0.9875
−0.5621
−0.5254
RVD-W DESCRIPTOR

Figure 14. RVDW processing pipeline with novel processing blocks marked in blue.

The novel elements developed by the project are marked in blue. In particular, in the second phase of the project, 3 novel processing blocks were developed and incorporated in the pipeline: (1) cluster wise rotation, (2) cluster-wise whitening, and (3) novel post-processing involving L1 normalisation combined with the power norm. These new developments and extensions resulted in further and significant performance improvements over the already BSOTA results at the end phase A, as illustrated in Table 1 below. Using the conventional SIFT descriptors, the mAP for Oxford 5k database was improved from 59.5% to 66.8%, while for the Holidays dataset from 73.2% to 76.5%. These results represent the state-of-the art performance, compared to competitors also using SIFT local features. Furthermore, we also extended the RVDW approach to work with deep features, resulting in BSOTA performance (Subsection 5.5.3).

Table 1. Comparison of mAP performance achieved at the end of Phase A and Phase B of the project (based on SIFT features).

Method	Dimension	Size	Oxford 5k	Holidays	UKB
RVD (version A)	8k	32kB	59.5	73.2	3.53
RVD-W (version B)	8k	32kB	66.8	76.5	3.59

5.2 RVD Local Whitening (RVD-W)

We introduced additional whitening on the cluster level, in order to “normalise” the probability distribution of residual vectors is each cluster. This has led to a marked increase in performance.

In the RVD-W aggregation scheme each local descriptor x_t is defined by its position with respect to the K nearest cluster centres (typically K=3) in the high dimensional space. More precisely, K-means clustering is performed to learn a codebook of $\{u_1, \dots, u_n\}$ of n cluster centres typically. Each local descriptor x_t is quantized to K nearest cluster centres thus increasing the number of descriptors assigned to each centre, resulting in more populous cluster-level representations, which are more robust. For each cluster, the residual vectors $x_t - u_j$ are computed and subsequently L1-normalized.

Each normalized residual vector is weighted for each neighbourhood rank (N) before aggregation, to yield vector $r_{tj} = \Psi_N(x_t - u_j/\|x_t - u_j\|_1)$. The neighbourhood weights Ψ_N are computed as the empirical probability that two descriptors forming a matching pair (inliers) with specific neighbourhood rank are assigned to the same cluster.

The variance in each dimension of weighted residual vector r_{tj} is different which affects the discriminability of the RVD-W representation. We solve this problem by applying cluster level PCA and a whitening operation on r_{tj} vectors before aggregation. More precisely, given a set of m weighted residual vectors $\{r_{1j}, r_{2j}, \dots, r_{mj}\}$ extracted from training images, we compute the mean vector $\eta_j = \mathrm{E}[r_{tj}]$ and the covariance matrix Σ_j for each cluster j. We then learn a PCA matrix P_j whose columns consists of the orthonormal eigenvectors of Σ_j corresponding to the d largest eigenvalues $\lambda_1 \geq \lambda_2 \dots \geq \lambda_d$. Finally, the cluster level whitening matrix P_j^w is computed as $P_j^w = P_j \Lambda_j$ where $\Lambda_j = diag(\lambda_1, \lambda_2 \dots, \lambda_d)$.

Given an image I, the vectors r_{tj} are computed for each cluster j. The mean subtracted r_{tj} vectors are then projected using P_j and subsequently whitened before aggregation into representation ζ_j.

$$\zeta_j = \sum_{N=1}^{K} \sum_{x_t \in \text{ rank } N \; of \, u_j} P_j^{w\top} \left(\Psi_N \frac{x_t - u_j}{\left\| x_t - u_j \right\|_1} - \eta_j \right)$$

The L2-normalized ζ_j vectors are stacked to form the final RVD-W representation R^w.

5.3 PCA transformation and L1+Power normalization

In order to improve the separability between matching and non-matching representations, we proposed a new normalization approach applied after transforming the RVD-W vectors via PCA. Our normalization involves an L1-norm followed by a power-norm creating L1-P normalization. More precisely, the mean-centred R^w vector is first transformed using a $D' \times D$ PCA matrix P and then the resultant vector is L1-normalized to form $R^{w\rho}$.

$$R^{w\rho} = \frac{P^\top(R^w - R_0)}{\|P^\top(R^w - R_0)\|_1}$$

Finally, the vector $\boldsymbol{R^{w\rho}} = (\boldsymbol{R_1^{w\rho}}, .., \boldsymbol{R_{D'}^{w\rho}})$ is processed using power-normalization with factor $\boldsymbol{\beta}$. **Table 2** below shows that our L1-P norm brings significant improvement in terms of mAP on all datasets.

Table 2. Comparison of Post PCA normalization methods.

Method	Oxford 5k [mAP]	Oxford 105k [mAP]	Holidays [mAP]	UKB [Top 4]
No norm	62.7	59.7	72.2	3.54
L1-P norm	66.8	64.0	76.5	3.59

5.4 Compact RVD-W code

The descriptor size, expressed as bytes per image, has a major impact on the performance of an image retrieval system; ideally the descriptors for the entire dataset should fit in the RAM memory of the server for fast processing. Aggregating a 128-dimensional local descriptor (e.g. SIFT) using a small codebook of 64 cluster centres results in 8k-dimensional global descriptor. This size is too large for efficient retrieval in very large databases.

We followed [35] to compress RVD-W vectors into small codes for large-scale retrieval. More precisely, the dimensions of vector $R^{w\rho}$ are permuted using the Eigenvalue Allocation method. The transformed vector is divided into g sub-vectors or groups of equal length D'/g. Each sub-vector is quantized using a separate K-means quantizer with n centroids (256) and encoded using $k = log_2(n)$ bits. The storage requirement of the embedded vector is $B = g \times k$ bits. The distance between the query vector and database vectors is computed using Asymmetric Distance Computation (ADC).

5.5 RVD-W based on Convolutional Neural Networks (CNN)

Recent research has shown that image descriptors computed using deep CNNs achieve state-of-the-art performance for image retrieval and classification tasks. Babenko et al. [36] aggregated deep convolutional descriptors to form global image representations: Fisher Vectors, Triangulation embedding and SPoC. The SPoC signature is obtained by sum-pooling of the deep features. Razavian et al. [37] compute an image representation by the max pooling aggregation of the last convolutional layer.

We propose to encode CNN-based descriptors into the RVD-W representation. More precisely, an RGB image is first warped into a $c \times c$ square and a mean RGB value is subtracted from each pixel. The image is then passed through a pre-trained network comprising of L convolutional layers. The output of a l-th layer L^l is a $c^l \times c^l \times d^l$ feature map, where d^l is the number of filters corresponding to L^l. A set $X^l = x_{1,1}^l, x_{1,2}^l, .., x_{c^l,c^l}^l$ of d^l- dimensional feature vectors is obtained at each location (a, b), $1 \leq a \leq c^l$ and $1 \leq b \leq c^l$, in the feature map. As in the SIFT-based approaches, a codebook of n cluster centres is learned using a set of training images. For each centroid, the residual vectors $x_{a,b}^l - u_j^l$ are computed, normalized and whitened to form vector ζ_j^l, regarding layer L^l. The RVD-W representation is obtained by concatenating all aggregated vectors ζ_j^l for all n visual words.

Experiments and Extensive performance evaluation has been performed, including comparisons to the state-of-the art pipeline developed by the MPEG group standardising Compact Descriptors for Visual Search (CVDS), and with the latest published work. The evaluation used standard reference databases including Holidays, Oxford and UKB, and also data sets extended by us with 10M distractor images. The evaluation has clearly demonstrated that in large-scale retrieval RVD-W descriptors significantly and consistently outperform all known techniques at comparable bitrate, including the BoW, VLAD, Fisher Vector and the recent Triangulation with Democratic Aggregation (Temb+Demo).

5.5.1 Compact RVD-W representation based on SIFT descriptors

In this experiment, key-points are detected using the Hessian affine detector and local regions are encoded in a 128-dimensional SIFT descriptor. The dimensionality of the SIFT descriptors is reduced from 128 to 64 dimensions using PCA matrix. The size of codebook is fixed at 128 resulting in 8k dimensional RVD-W descriptor.

Table 3 summarizes the results for medium footprint signatures. In practical applications, the use of medium footprint representations is prohibitive due to search time and memory requirements, however, the results are helpful in understanding the capabilities of each representation, and also serve as an upper bound on the expected performance of compact descriptors derived from them. It can be seen that the proposed RVD-W representation outperforms most of the prior-art methods.

Table 3. Comparison with the-state-of-the-art using full dimensional vectors on Oxford5k, Oxford105k, Holidays and UKB datasets. The representation 8k→1k denotes that the global descriptor dimensionality is reduced from 8192 to 1024 via PCA

Method	Dim	Oxf5k	Oxf105k	Hol	UKB
VLAD Intra [38]	32k	55.8	-	65.3	-
HiVLAD [44]	32k	63.8	-	72.1	3.56
VLAD+SURF [42]	12k	-	-	71.7	3.52
CPVLAT [45]	9k	-	-	70.0	-
VLAD* [38]	8k	50.0	44.5	62.2	-
VLAD (LCS+RN) [38]	8k	51.7	45.6	65.8	-
HVLAD [43]	8k	47.2	-	69.1	-
HiVLAD [44]	8k	57.6	-	66.6	3.48
Temb+Demo [39]	16k	66.5	-	76.8	-
Temb+Demo [39]	8k	67.6	61.1	77.1	-
Temb+Sum [39]	8k	63.3	55.5	74.5	-
FAemb [46]	16k	**70.9**	-	78.7	-
FAemb [46]	8k	66.7	-	76.2	-
RVD-W	16k	68.9	**66.0**	**78.8**	**3.60**

RVD-W	8k	66.8	64.0	76.5	3.59
VLAD [15]	4k	37.8	-	55.6	3.28
FV [15]	4k	41.8	-	60.5	3.35
VLAD+SURF [42]	4k	32.8	-	64.9	3.20
Temb+Demo [39]	8k→1k	56.2	50.2	72.0	-
RVD-W	8k→1k	**59.0**	**56.1**	**73.2**	**3.56**

We now focus on a comparison of compact representations, which are practicable in large-scale retrieval, as presented in Table 4. The dimensionality of the RVD-W descriptor is reduced from 8192 to 128 via PCA. The results show that our method outperforms all presented methods by a large margin. On the ultra large dataset of Holidays10M, RVD-W significantly outperforms the best published results (VLAD+SURF).

Table 4. Comparison with the state of the art using 96/128 dimensional vectors

Method	Dim	Oxf5k	Oxf105k	Hol	Hol1M	Hol10M
VLAD [15]	128	28.7		55.7	-	-
FV [15]	96	-	-	56.0	31.8	28.0
FV [15]	128	30.1	-	56.5	-	-
VLAD* [38]	128	32.5	26.6	-	33.5	-
VLAD (LCS) [38]	128	32.2	26.2	-	39.2	-
CPVLAT [45]	256	-	-	60.6	38.0	-
VLAD+SURF [42]	96	-	-	65.5	42.5	34.0
HiVLAD [44]	128	-	-	64.0	43.0	-
Temb [39]	8k→128	40.0	33.9	61.5	–	–
Temb [39]	2k→128	43.3	35.3	61.7	38.7	-
RVD-W	128	**46.1**	**42.5**	**66.9**	**45.1**	**40.5**

Table 4 shows the performance of our method using compact codes obtained by product quantization. Compared to VLAD (LCS), the gain remains very significant on Oxford5k (+14%), Oxford105k (+16%) and Holidays1M (+5%). The RVD-W provides a gain of 9.4% on largest Holidays10M over VLAD+SURF.

Table 5. Comparison with the state of the art with compact codes via PQ

Method	Size	Oxf5k	Oxf105k	Hol	Hol1M	Hol10M
VLAD [15]	40 B	-	-	49.5	-	-
FV [15]	16 B	-	-	50.6	28.7	21.0
VLAD* [38]	16 B	28.9	22.2	-	29.9	-
VLAD (LCS) [38]	16 B	27.0	21.0	-	32.3	-
VLAD+SURF [42]	10 B	-	-	58.0	30.2	22.1
RVD-W	16 B	**41.2**	**37.1**	**61.4**	**37.3**	**31.5**

5.5.2 Compact RVD-W representation based on CNN features

This section compares the performance of CNN-based representations suitable for large-scale retrieval. We extract deep convolutional descriptors using the state-of-the-art CNN, OxfordNet [40] Each image is resized to the size 586 × 586 before passing through the network. The output of the last layer is a 37 ×

37 × 512 feature map, forming a set of 1369 512-dimensional descriptors. We compare RVD-W to the state-of-the art methods successfully used with CNN features: Max-pooling[37] , SPoC [36] and FV.

The retrieval performance of the CNN-based representations is presented in Table 6. It can be seen that 256-dim RVD-W improves over FV, delivering a gain of +7.8% on Oxford and 3.3% on Holidays. Compared to Max-pooling, RVD-W provides an improvement of +6.7% and 5.5% on Oxford and Holidays datasets. On large scale datasets Hol1M and Oxf1M, RVDW offers a gain of +1.3% and +3.7% compared to the best performing state-of-the-art SPoC signature. The 2048-dimensional RVD-W outperforms all CNN based approaches. It should be noted that the performance of SPoC deteriorates when a 512 dimensional signature is used (79.6% on Holidays and 55% on Oxford).

Table 6. Comparison with the state of the art with CNN-based compact codes

Method	**Size**	**Oxf5k**	**Hol**	**Hol1M**	**Oxf1M**
MOP-CNN [47]	512	-	78.4	-	-
Max-pooling [37]	256	53.3	74.2	-	-
SPoC [36]	256	58.9	78.5	62.2	41.1
FV	256	52.2	76.4	58.1	35.5
RVD-W	256	**60.0**	**79.7**	**63.5**	**44.8**
MOP-CNN [47]	2048	-	80.2	-	-
Max-pooling [37]	2048	58.0	70.7	-	-
FV	2048	64.1	81.9	-	-
RVD-W	2048	**67.5**	**84.5**	-	-

5.5.3 Compact RVD-W representations based on binary local features

In this section we present a pipeline to aggregate local binary descriptors into RVD-W framework for large-scale image retrieval in mobile scenarios. Binary descriptors are becoming increasingly popular, especially in mobile applications, as they deliver high matching speed, have a small memory footprint and are fast to extract.

For our experiments, we select three local binary descriptors: two intensity-based (BRISK and FREAK) and one gradient-based (BRIGHT). Our choice is motivated by their high level of performance in retrieval using descriptor-by-descriptor matching with bi-directional ratio test, but without geometric verification. In all cases we employ BRISK key-point detection, as it is fast, delivers good performance and is the de-facto standard in mobile applications.

Given an image, a set of binary local descriptors are extracted. The descriptors are compressed to $d = 128$ dimensions using Principal Component Analysis. The compressed descriptors are rank-assigned to multiple clusters and a robust representation of residual vectors in each cluster is derived forming the RVD-W global descriptor. The size of codebook is fixed at 64 resulting in 8k dimensional RVD-W descriptor. The high-dimensional global descriptor is converted into a compact signature by application of global PCA.

Table 7 compares the performance of RVD-W representation with Fisher Vector and VLAD. The upper section of Table 7 lists the performance of binary descriptor aggregation schemes with the fast BRISK detector and BRISK/FREAK descriptors. It can be seen that RVD-W significantly outperforms the state of the art global descriptors by +3.5% on average.

Although this section is about aggregation of binary descriptors, we also compare our framework with global descriptors that use Hessian-affine or DoG detector with SIFT descriptor, which is five times slower (lower part of Table). It can be clearly observed that the FREAK+RVD-W is better or comparable to

HA+SIFT and DoG+SIFT combined with BoW, VLAD and FV. On large-scale dataset of Holidays1Million, FREAK+RVD-W achieves 35.1% compared to SIFT+FV 31.8%.

Table 7. Comparison with the state of the art with binary local descriptors

METHOD	Detector	Descriptor	Dimension	Oxford	Holidays
FV	BRISK	BRISK	8192	35.3	58.1
FV	BRISK	FREAK	8192	36.7	59.8
VLAD	BRISK	BRISK	8192	33.5	55.3
VLAD	BRISK	FREAK	8192	34.8	56.7
RVD-W	BRISK	BRISK	8192	38.5	60.9
RVD-W	BRISK	FREAK	8192	**40.9**	**63.3**
BOW [15]	HESSIAN AFFINE	SIFT	20000	35.4	43.7
FV [15]	HESSIAN AFFINE	SIFT	8192	41.8	60.5
VLAD [15]	HESSIAN AFFINE	SIFT	8192	37.8	55.6
VLAD [42]	DoG	SIFT	8192	24.3	56.1

Table 8 compares the average time required to compute RVD-W signature, for different combinations of detectors and descriptors. The total time comprises local descriptors extraction (first column) and encoding of the global representation (second column). It can be observed that use of binary local descriptors reduced computational complexity by factor of 5, as compared to working with SIFT.

Table 8. Average time required to compute B-RVDW signature using different detectors and descriptors combinations (DoG: Difference of Gaussian, HA: Hessian-Affine).

Detector	Descriptor	Local descriptor extraction time (ms)	Global descriptor extraction time (ms)	Total time
BRISK	BRISK	85	200	285
BRISK	FREAK	85	200	285
DoG	SIFT	900	190	1090
HESSIAN AFFINE	SIFT	1230	190	1420

5.6 Conclusions on the family of extended RVD-W representations

This section presents a novel method for extraction of a robust and highly discriminative global descriptor called RVD-W. The key ideas include a novel robust aggregation approach with rank-based multi-assignment, direction based accumulation, and mid-stage de-correlation and whitening of the residual vectors. The proposed aggregation is also combined and shown to be effective with CNN based and binary features, outperforming the latest global descriptors. A detailed evaluation on de-facto standard benchmarks demonstrates that in large-scale retrieval our scheme outperforms state-of-the art methods.

5.7 Indexing RVDW features for fast search

The previous sections provided a method for extracting an efficient representation for images based on RVDW. However, there remains the problem of efficiently searching for nearby examples to a given query. Even for compact binary descriptors, the exhaustive search through possibly billions of binary strings can take seconds or even minutes. In this section, we describe a method for efficiently searching within Hamming space using multiple hash tables, each with different length hash-keys. We assume that the image representation is in the form of a binary vector.

A related hashtables-based retrieval approach was proposed by Norouzi et. al [48] called Multi-Index Hashing (MIH). Here, a binary code is divided into equally sized substrings and separate hashtables are built from them. The configuration of substring lengths and their number is selected such that a superset of relevant examples (i.e. within some r-neighbourhood in Hamming space) are returned. Examples that are above distance r are then removed using linear scan. The resulting search speeds were significantly faster than linear scan. However, this speedup is possible only for small Hamming thresholds r. When r increases, the time spent on removing inaccurate retrievals increases very quickly and eventually, becomes very similar to exhaustive linear scans. This is due to the constraint of equal length strings. Our work removes this constraint and we show how this improves in the retrieval time compared to the MIH approach, across a large range of Hamming thresholds. Simultaneously, our approach also provides the option for a faster approximate search. Henceforth, we shall denote our improved method as: *variable length hashtable retrieval*.

The retrieval mechanism consists of M separate hashtables, which we denote as a hashtable set, each associated with a substring of length $m(i)$. Each hashtable takes a substring as the hashkey and returns a set of example indices with a similar key (i.e. colliding examples). The full set of retrieved examples is the union of the retrieved examples across all the M hashtables. In this work, we aim to configure the keys such that there is high probability of collisions in the hashtable for examples that have Hamming distances less than θ, whilst minimising the collisions for examples with distances greater than θ. Formally, we can think of each hashtable as a function, $H_i : \{0, 1\}^{m(i)} \rightarrow 2^N$. Associated with the i^{th} hashtable H_i, is a set of key vector dimensions $\mathbf{D}_i = \{ d_{i,j} \}^{m(i)}_{j=1}$. The “dimension index set” $\mathbf{D}_i$ can then be used to extract the hashtable key from the query binary vector. Then, suppose we are given an input query $\mathbf{q} \in \{0, 1\}^D$ .We first extract the substring keys (of length m_i) for each of the hashtables using their respective dimension index set $\mathbf{D}_i$ as follows: $\mathbf{q}^i = (q_{d(i,j)})^{m(i)}_{j=1}$. The final set of retrieved examples, $\mathbf{R}$, is then given as: $\mathbf{R} = H_1(\mathbf{q}^1) \cup H_2(\mathbf{q}^2) \cup \ldots \cup H_M(\mathbf{q}^m)$.

An illustration of an example of the hashtable retrieval mechanism can be seen in Figure 15. Example of the hashtable retrieval mechanism.. Here, the feature vector is split into 3 hashkeys of lengths *m1, m2, m3* respectively. Each substring is a hashkey to the corresponding hashtables *H1, H2, H3* respectively. Given a query vector **q**, its substrings are used to retrieve relevant example indices before a final union to give the final retrieved indices.

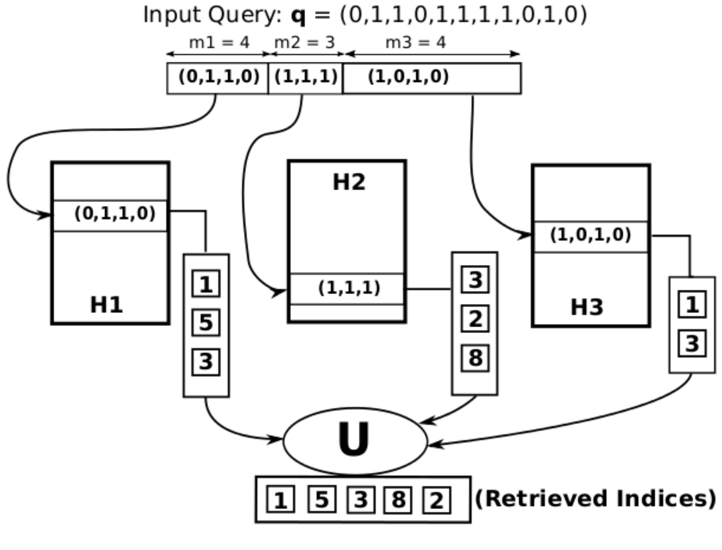


Figure 15. Example of the hashtable retrieval mechanism.

However, since the retrieval from the hashtables is a superset of the required examples, that is, there exists some examples returned that have Hamming distance greater than the required threshold, θ. To tackle this issue, we perform linear scan on the examples in **R**, and only return those that have Hamming distance less than θ. This is still significantly faster than performing linear scan on the entire dataset, since the size of **R** is usually much smaller than the size of the dataset.

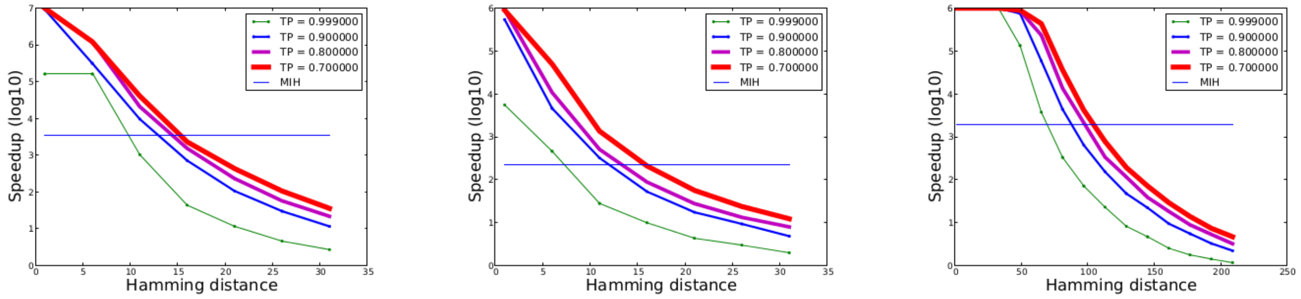
A method of determining the retrieval probability given a set of hashtables with a particular hashkey length configuration was proposed. Following this, a novel algorithm searching for an optimal hashkey length set allow for retrieval satisfying a pre-defined minimal recall rate, whilst minimising the recall time was proposed. The details can be found in the paper itself [49]

The variable hashkey hashtable retrieval method was evaluated on 3 different large-scale databases: 1-Billion dataset (128-D SIFT), 1-Million ANN (SIFT), 1-Million Flickr Images (512-D RVD vector). The speed up results over the original MIH method can be seen in Table 9: Speedup of retrieval time of proposed method over MIH..

Table 9: Speedup of retrieval time of proposed method over MIH.

Min. Recall Rate	**0.999**	**0.9**	**0.8**	**0.7**
1-Billion ANN	30	875	784	785
1 Million 128	13	818	1359	1064
1 Million Flickr	18	86	98	103

As can be seen, the proposed method achieves on average a factor of 10-1000 times speedup in retrieval times when compared with the MIH method. The speedup over linear scan is even more considerable, as can be seen in Figure 16. Speed-up of proposed retrieval method over linear scan.



1 Billion ANN 1 Million 128D SIFT Flickr 1 Million (512-bit)

Figure 16. Speed-up of proposed retrieval method over linear scan.

Here, we find that when only retrieval of examples that have small Hamming distance from the query are required, the speedup over linear scanning can be up to 100K. This is because the retrieval sets from hashtables are considerably smaller in size at lower Hamming distance thresholds. This, in turn, leads to a much faster retrieval time, even for large datasets with 1 Billion examples.

5.8 Conclusions

In summary, in the second phase of the project, we further improved our Robust Visual Descriptor, creating RVDW. We have also integrated RVDW with deep features, demonstrating world leading performance, beyond state-of-the-art. Further, we have also shown that RVW-D can aggregate binary descriptors directly, which is useful for systems with limited computational resources. Binary local descriptors require less processing resources to extract, compared to floating point or deep descriptors.

Finally, we designed a fast indexing scheme for our RVD-W descriptors, based on sub-string matching, to reduce search speed for very large databases.

6 Development of a new tool for the geometric consistency check in video

Most modern methods for determining whether two images display the same object involve detection and matching of interest points followed by a test to see whether some of the matches may correspond to a common image transformation. This latter operation is often called a *geometric consistency* test.

We present a geometric consistency test for video *shots*, understood as

> [...] one or more frames generated and recorded contiguously and representing a continuous action in time and space [50]

The method compares a query shot to a reference shot, and the goal is to establish whether the same object can be seen in both. A reference shot that displays the same object as that in a query will be called a *matching* shot, and a shot that does not will be called *non-matching*.

Traditionally, in a retrieval setting, a shot is represented by a single frame (key frame). The goal is usually to detect whether a certain object is present in the shot by comparing the keyframe to a query that depicts the desired object. This may cause some difficulties, notably when the object depicted in the query image is visible in parts of the shot but not in the keyframe. Another difficulty arises when the object is small, blurred, or is poor in detail, such that a simple comparison between two images does not reveal the presence of the object.

We propose to overcome these problems by letting a shot be represented by several frames, obtained by sampling the shot over its length (uniformly in time or by some other rule). The proposed test thus accumulates evidence from interest point matching over several image pairs (a frame in the query, a frame in the reference). The method is robust, in the sense that the presence of common objects is detected also when the ratio of correct to incorrect matches of interest points is low across all image pairs.

In the first sub-section, we introduce the proposed method and provide arguments for why it should be successful. Thereafter we report on an experiment comparing the method to some extensions to the state of the art for single image pair matching.

6.1 The method: DISTRAT for multiple image pairs

The DISTRAT method considers all line segments between interest points and computes the logarithm of the ratio of the lengths of corresponding segments in the two images, see [51] The statistics of the logarithmic distance ratio (LDR) for the image pair is expressed by a histogram. This histogram is compared to the shape it would have if all matches were wrong – this shape is expressed by a probability mass function (pmf). The image pair is declared as a match if the histogram differs sufficiently from the pmf (the method uses Pearson's chi-squared test).

The proposed technique extends this procedure to video shots. The method performs interest point matches independently for all pairs of key frames from the query and reference shots. These matches are processed according to the DISTRAT method, producing a sequence of histograms and a sequence of probability mass functions. This part will be called the *preliminary operation*. The method then proceeds to the *decision operation*, which merges both the histograms and the probability mass functions and then performs a hypothesis test to decide whether the shot matches the query.

When values from different sources are measured

Suppose that M image pairs are available for comparison, and that the first step in the preliminary operation produces the individual histograms

$$\begin{gathered}[g_1(1) \quad \cdots \quad g_1(K)] \\ \vdots \\ [g_M(1) \quad \cdots \quad g_M(K)]\end{gathered}$$

Each of these represents the frequencies of LDR values for one image pair. Also, the numbers of LDR values taken for the individual image pairs are $N_1, \dots, N_M$, and the total number of LDR values is their sum

$$N = N_1 + \cdots + N_M.$$

According to the DISTRAT method, the hypothetical probability mass function (representing the case where all matches are outliers) is specific for an image pair, since it depends on the actual configuration of interest points in the two images. It follows that M different pmfs are needed to model the outlier behaviour in the M image pairs

$$\begin{gathered}p_1 = [p_1(1) \quad \cdots \quad p_1(K)] \\ \vdots \\ p_M = [p_M(1) \quad \cdots \quad p_M(K)]\end{gathered}$$

We shall call these the *individual outlier pmfs*.

The histogram that represents frequencies over the whole set of image pairs is obtained by summing the individual histograms

$$g = g_1 + \cdots + g_M.$$

This histogram will be called the accumulated histogram.

To arrive at an outlier pmf for the total set of image pairs, we must take into account the changing pmfs. The appropriate model in this case is a *mixture*, such that the pmf is a linear combination of the individual outlier pmfs,

$$p(k) = \sum_{m=1}^{M} P(m) \cdot p(k|m)$$

where $P(m)$ is the probability (i.e. relative frequency) that an LDR value is taken from image pair m, and $p(k|m)$ is the probability that the LDR value is within bin ς_k provided that the LDR is taken from image pair m. Since the relative frequency of values in image pair m is the fraction of the number of LDR values to the total,

$$P(m) = \frac{N_m}{N}, m = 1, \dots, M.$$

The conditional probability mass functions are simply the individual pmfs

$p(k|m) = p_m(k), m = 1, \dots, M.$

Therefore, the outlier pmf for the total set of image pairs is

$$p(k) = \frac{1}{N} \sum_{m=1}^{M} N_m \cdot p_m(k).$$

We may now introduce the method.

How the method works

The method consists of a preliminary operation (known from previous techniques) and a hypothesis test that uses accumulated information produced in the preliminary operation. It can be summarized as follows.

Procedure 1 [Multi-frame DISTRAT]

Let a query shot and a reference shot be given. Select key frames from each of the shots, creating pairs consisting of one key frame from the query and one key frame from the reference.

PRELIMINARY OPERATION

The following iterations coincide with the initial processing in DISTRAT, see [51] The output is a set of histograms and probability density functions.

1. For each image in (query and reference), detect local features.
2. For each image pair, match local features to produce lists of coordinate matches.
3. For each list of coordinate matches, compute a histogram of logarithmic distance ratios.
4. For each list of coordinate matches, compute a probability mass function for outliers (incorrect matches).

DECISION OPERATION

The following steps prepare and carry out a hypothesis test for establishing whether the query and reference shots depict the same object.

1. Sum the previously computed LDR histograms.
2. Compute a mixture of the previously computed outlier pmfs.
3. Compute Pearson's test statistic [53]
4. If Pearson's test statistic is below a threshold, then the query most probably does not match the reference. Stop.
   Otherwise, the query and reference shots may match, so proceed with the next step.
5. Estimate the correctly matched key points (inliers), through the eigenvalue problem. If the sum of weights for the inliers exceeds a given threshold, then declare that the query and reference shots match.

6.2 Why it works

There are two mechanisms at work ensuring that the proposed works well. The first is that it effectively provides a large number of observations that increase the reliability of the hypothesis test. The second mechanism is that the LDR statistics show small variation across all the considered image pairs – such that similar histograms are accumulated.

We discuss these two mechanisms in the following paragraphs.

Robustness increases with shot length

We examine the expected value of Pearsons's test statistic for two cases: in the first case, the histogram is exclusively due to outliers; in the second case, there are some inliers.

A histogram **due to outliers** is modelled by a multinomial random variable H with parameters $N, p(1), \ldots, p(K)$. The expected value of the test statistic C in this case is,

$E(C) = K - 1,$

which depends on the number of bins and not on the number of samples used for making the histogram.

In the second case with **some inliers** present, the histogram is modelled by a multinomial random variable G with parameters $N, q(1), \ldots, q(K)$, such that the probabilities $q(k)$ are different from the probabilities $p(1), \ldots, p(K)$ that characterize the LDR for outliers. If the two mass functions p and q are held fixed, then the expected value of the test statistic C grows as a function of N,

$$E(C) = f(N) = a \cdot N + b$$

where the two constants are positive. A proof is straightforward.

Thus, by just letting N become large enough, the expected value of C in the case of inliers may exceed any threshold. Since the constants are functions of p and q, of course the rate at which this happens will vary: if q is close to the outlier pmf p (meaning that there are few inliers), then the number N of matches will have to be large -- but eventually, $E(C|q)$ will become larger than any fixed threshold. Therefore, if the pmfs p and q remain unchanged, then increasing the number N of samples will help separating the values of the test statistic for matching images (some inliers) from the values for non-matching images (no inliers).

The peak in the pmf often moves little

Inliers give rise to local maxima in the probability mass functions (and therefore peaks in the LDR histograms). For all pairs (query frame, reference frame) that contain the same object, these local maxima will be found over roughly the same bins. As a consequence, the accumulated probability mass function in q will also display such a local maximum.

This is rooted in the nature of video shots, as stated by Cotsaces et al [52] :

> [...] video content within a shot tends to be continuous, due to the continuity of both the physical scene and the parameters (motion, zoom, focus) of the camera that images it.

This continuity is manifested as a small and gradual change in the individual LDR histograms across the sequence. Since the logarithm compresses the range of the LDR, this will most often lead to narrow peaks over the same bins for all histograms over the sequence.

6.3 Correct interest point matches

The matched interest points in the various pairs of video frames provide evidence that can be extracted thanks to multi-frame DISTRAT. As mentioned, in a single pair of frames, the number of correctly matched interest points (inliers) may in many cases be relatively low (say, below 25% of the total number of matches). Such low ratios of inliers to total number effectively prevent robust methods like RANSAC or single-frame DISTRAT from correctly identifying the inliers.

Single-frame DISTRAT suffers this breakdown mostly because the number of segments between matched interest points is low, causing the LDR histogram to become irregular or jagged. Video DISTRAT can improve on this situation, since it accumulates LDR histograms (and pmfs) from several pairs of frames. In this case, the histograms tend to become smoother, as their shape approaches that of the probability mass function q(k) (see the section "Robustness increases with shot length" above in this chapter).

The LDR histogram and outlier pmf computed for the union of all frame pairs will, for each single pair of frames, be used to define the matrix that yields the estimate of inliers.

Let g(k) denote the accumulated histogram and let p(k) denote the outlier pmf. Then the inliers in a single pair of frames are then estimated as follows.

- Compute the factor β,

$$\beta = \frac{\sum_{k=1}^{K} g(k)p(k)}{\sum_{k=1}^{K} (p(k))^2}$$

- Create the *outlier normal* of the histogram

$$d(k) = g(k) - \beta p(k).$$

- Let q be the quantizer that assigns a bin to any LDR value,

$$z \in \zeta_k \Rightarrow z \xrightarrow{q} k.$$

Construct the matrix D

$$D_{ij} = \begin{cases} d_q\left(z_{ij}\right) & i \neq j \\ 0 & i = j \end{cases}$$

where $d_q = d \circ q$, and z_{ij} are the LDR values of the single pair of frames. That is, z_{ij} is the logarithm of the ratio lengths of the segments between the ith and jth interest points in either of the two frames.

- Find the dominant eigenvector r of D with eigenvalue μ,

$$Dr = \mu r.$$

- Estimate the number of inliers,

$$\widehat{m} = 1 + \frac{\mu}{\max\limits_{k=1,\ldots,K} d(k)}.$$

- The inliers correspond to (the indices of) the $\widehat{m}$ largest elements in the eigenvector r.

This procedure is almost identical to the one adopted for geometric verification in MPEG-7 CDVS, see “Text of ISO/IEC CD 15938-14 Reference software, conformance and usage guidelines for compact descriptors for visual search” [56] The only difference is that the histogram and outlier pmf are computed over the whole set of frame pairs instead of a single frame pair.

The local feature matches all have weights, representing the degree of uniqueness of the match [68]. Now, the weights of all the estimated inliers across all the pairs of frames may be summed in order to get a score of how well the query and reference shots match.

6.4 Experiment: test on video sequences

The method has been tested on a large set of video shots, extracted from 180 video sequences contained in the CTurin180 dataset, used in the development of the CDVS standard, by dividing each sequence into 10 shots. For the test, 16200 matching pairs (query, reference) and 72000 nonmatching pairs have been identified. The method, as well as two alternative methods for comparison, have been applied to this material.

All frames are represented by CDVS descriptors of length 512 bytes, that is, the shortest of the alternatives of the standard.

The alternative methods: late fusion strategies

The test framework for CDVS represents the state of the art for matching pairs of single images. The score used for declaring match in CDVS is a sum of distinctness scores for those interest point matches that are estimated to be correct. This method therefore involves more algorithmic steps than the basic DISTRAT method.

We consider extensions of this technique to multiple image pairs by using *late fusion* methods, commonly used for combining scores in information retrieval [54] [55] The methods are combMAX and combSUM applied to the CDVS scores for all pairs (query, reference). The first method produces a score as the maximum over the individual scores in a set of (query, reference). The second method produces a score as the sum of the individual scores.

The outcome

The experiment regards cases in which the shots are represented by 1, 2,...,10 frames. Results are presented in terms of True Positive Rate for thresholds such that the False Positive Rate is 0.01. (The True Positive Rate is the fraction of positive outcomes across the matching pairs. The False Positive Rate is the fraction of positive outcomes – wrongly declared matches – across the nonmatching pairs.). Figure 17 summarizes the results.

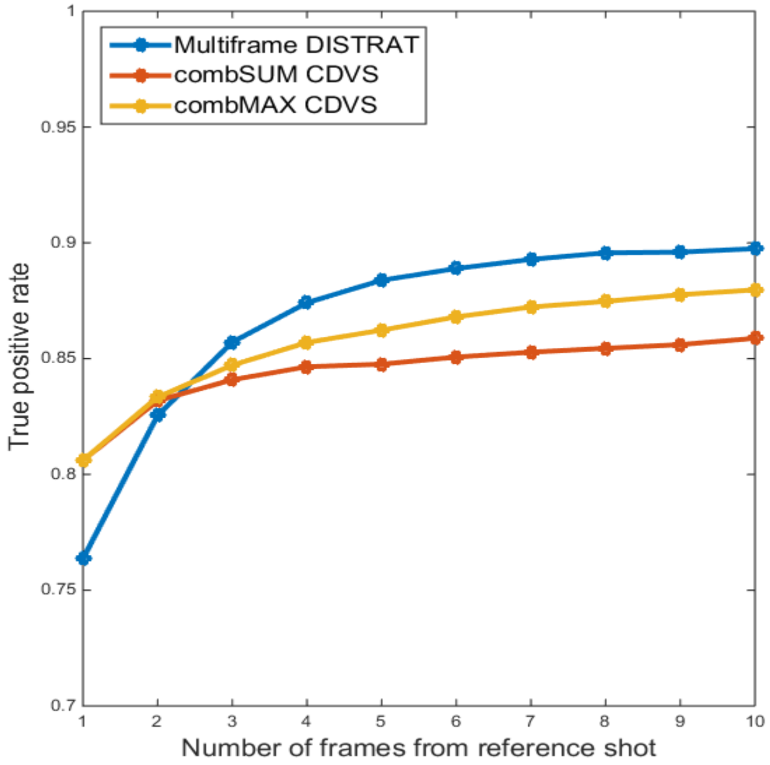


Figure 17. True positive rates at 1% false positive rate.

It is seen that the proposed method significantly outperforms late fusion of the CDVS scores when shots are represented by more than 3 frames. The advantage in TPR is approximately 2% when shots are represented by more than 5 frames.

6.5 Conclusion

We proposed a method to identify match between a query shot and a reference shot, meaning that the two display the same object. The method consists in comparing a histogram for logarithmic ratio distances to a certain probability mass function. The proposed method outperforms late fusion of CDVS scores over a large dataset.

7 Standardisation activities: proposal submitted to MPEG CDVA

In this section we present the results of the experiments carried out on MPEG Compact Descriptors for Video Analysis (CDVA) using the CDVA Experimental Model (CXM) v0.1 [57] based on the BRIDGET proposal submitted at the 114th MPEG meeting [60]

The document addresses Core Experiment 1 (CE1) "Temporal Sampling" as described in [58]

7.1 Technical Description

The proposed technology for search and retrieval of videos is composed by three parts:

1. the extraction of binary descriptors from source videos;
2. the matching of descriptors to detect if any part of a query video matches any part of a reference video;
3. the retrieval of relevant videos from a large video archive using the binary descriptor of a video as the query information and a set of binary descriptors of reference videos as the database of the archive.

The proposed CDVA search and retrieval solution is based on the CDVA Experimental Model (CXM) v0.1, with some modifications:

1. the use of the RVD-W global descriptor, proposed by the University of Surrey and Visual Atoms [65] ;
2. the use of the Coordinate Coding algorithm that was originally proposed by Telecom Italia in its response to the CDVS Call for Proposal [66] ;
3. the different binary descriptor format adopted to carry some extra information needed for temporal localization of key frames (i.e. the start and end time in ms of each key frame), and to speed up the serialization of descriptors into binary streams.

The RVD descriptor has been described in detail in [65] the RVD-W is a successor and adds cluster-level whitening of the residual vectors. In this extension, the variances of residual vector directions are balanced, in order to maximize the discriminatory power of the aggregated vectors. This is achieved by a novel intra-stage pre-processing of the residual directions using cluster-wise PCA with a whitening operation. We call this representation RVD-W and apply it in the proposed CDVA solution.

The Coordinate Coding algorithm encodes the coordinates of the key points in an image by quantization and arithmetic coding. The method is similar to that of Tsai *et al.* [67] inasmuch as we form a histogram of the coordinates over the quantization grid. We have used a grid of 3x3 pixels in each bin, such that the maximum error is of 1.5 pixels in both dimensions. The positions of the nonzero bins (histogram map) is are encoded by forming binary words through scanning columns and compressing the words by arithmetic coding. The number of coordinates in the nonzero bins (histogram count) is encoded by specifying which bins contain more than 1,…,B key points (B is the largest count). Depending on the number of key points that are selected in the image, the bit rate for the coordinates of one key point is in the range from 2 to 7.5 bits.

All other CXM technologies (e.g. the ALP detector, the SIFT descriptor binary encoding, the matching algorithms, etc.) are used without modifications (using the "mode 0" parameter set); however, some parameters have been modified from the default to achieve better results.

7.2 Extraction

Extraction is performed in a slightly different way compared to the CXM. In the following figure we show the modified CXM extractor.

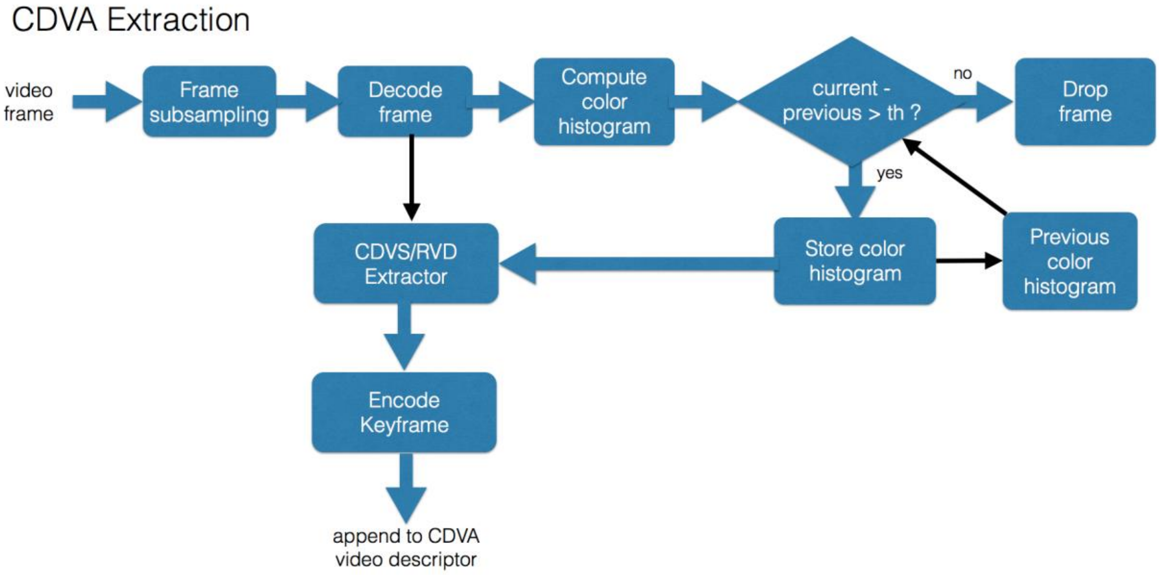


Figure 18. The modified CXM extractor.

Figure 18 illustrates how the modified CXM produces 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 output descriptor is updated by appending the output of the process to a single CDVA descriptor.

1. **Frame subsampling:** Performs temporal frame subsampling.
2. **Decode frame**: Decode a frame present in the video.
3. **Compute colour 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 colour histograms**: if the difference is greater than a given threshold, the frame is selected as *keyframe* 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. **CDVS/RVD Extractor**: Extracts the CDVS/RVD descriptors from individual frames, using mode 0 of the CDVS standard [69] This step is composed by the following operations:

    a. the candidate keyframe image is converted to grey scale and resampled at VGA resolution;
    b. the relevant features of the image are detected by the CDVS keypoint detector;
    c. the feature selection algorithm of CDVS is applied to select the most important 300 features;
    d. the SIFT descriptor information of each selected feature is extracted and stored in both normalized uncompressed mode (as in CDVS) and root SIFT mode (RVD specific);
    e. root SIFT information is used to produce the RVD-W global descriptor;
    f. normalized uncompressed mode SIFT information is used to produce a binary encoded local descriptor;
    g. the coordinates of all selected features are encoded using the Coordinate Coding algorithm;

8. **Encode keyframe**: data structures extracted from the current frame by the CDVS/RVD Extractor according to step 7 are encoded in binary format and written into a CDVA descriptor concatenating each keyframe to the following.

7.3 Pairwise matching

The matching algorithm decodes all keyframes stored in the CDVA **reference** descriptor and builds a DB using them. Then, it decodes all keyframes stored in the CDVA **query** descriptor and for each of them executes a retrieval operation on the reference DB. A score is computed as the product of the local and global score, after having subtracted the base threshold for each of the two values:

score = (local_score - local_thr) * (global_score - global_thr)

If the score is above a given threshold, the keyframe is marked as matching for temporal localization. Finally, the maximum score value of the top matching results of all retrieval operations is returned as the matching score.

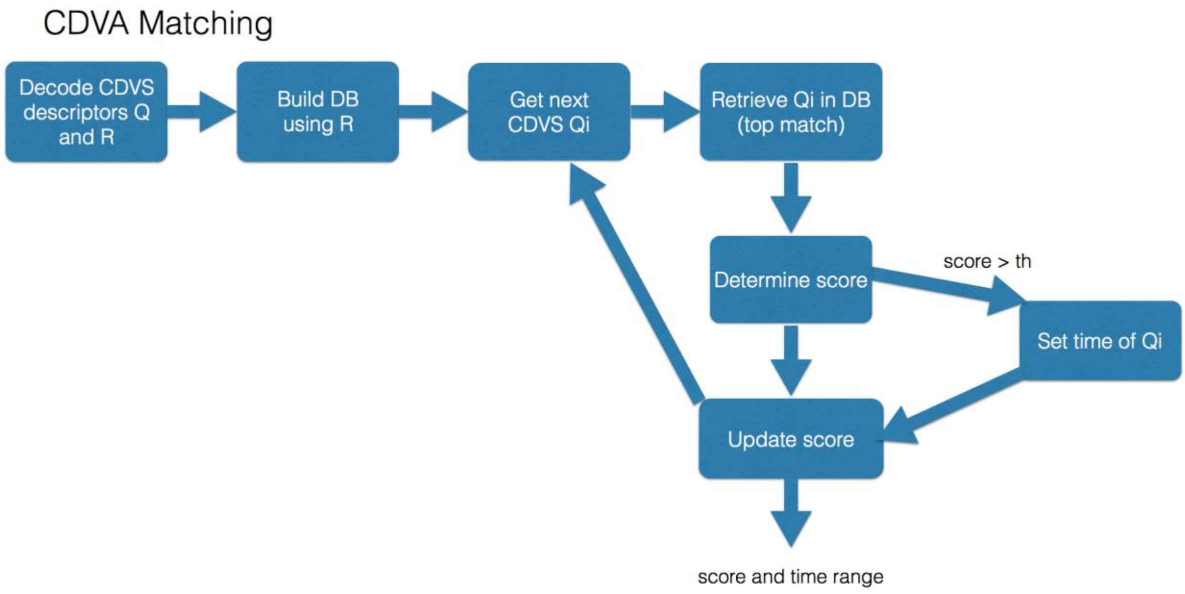


Figure 19. The modified CXM pairwise matching.

Figure 2 illustrates in detail the CDVA pairwise matching operation.

1. **Decode:** Decode all CDVS descriptors stored in the Query CDVA Descriptor (Q) and in the Reference CDVA Descriptor (R).
2. **Build DB using R**: use all Reference CDVS descriptors to build a Database (DB).
3. **Get next CDVS Qi**: for each Query descriptor, execute the following steps.
4. **Retrieve Qi in DB**: perform a CDVS retrieval operation using the Query Descriptor Qi obtaining the top match result.
5. **Determine score**: determine the current score as a combination of the local and global score of the top match result.
6. **Set time of Qi**: if the current score is greater than a given threshold, store the time of Qi. When all Qi descriptors have been retrieved, find the maximum interval that satisfies the matching strategy.
7. **Update score**: set the total score as the maximum of all current scores.

7.4 Retrieval

In retrieval, all CDVA **reference** and **distractor** descriptors are decoded and all keyframes contained therein are used to build a single DB. Then, the CDVA **query** descriptor is decoded and each decoded keyframe is used to perform a retrieval operation on the DB. The results of all queries are merged (removing duplicates) and sorted again, then clipped to a short value (50). The list of 50 results is returned to the CDVA Evaluation Framework.

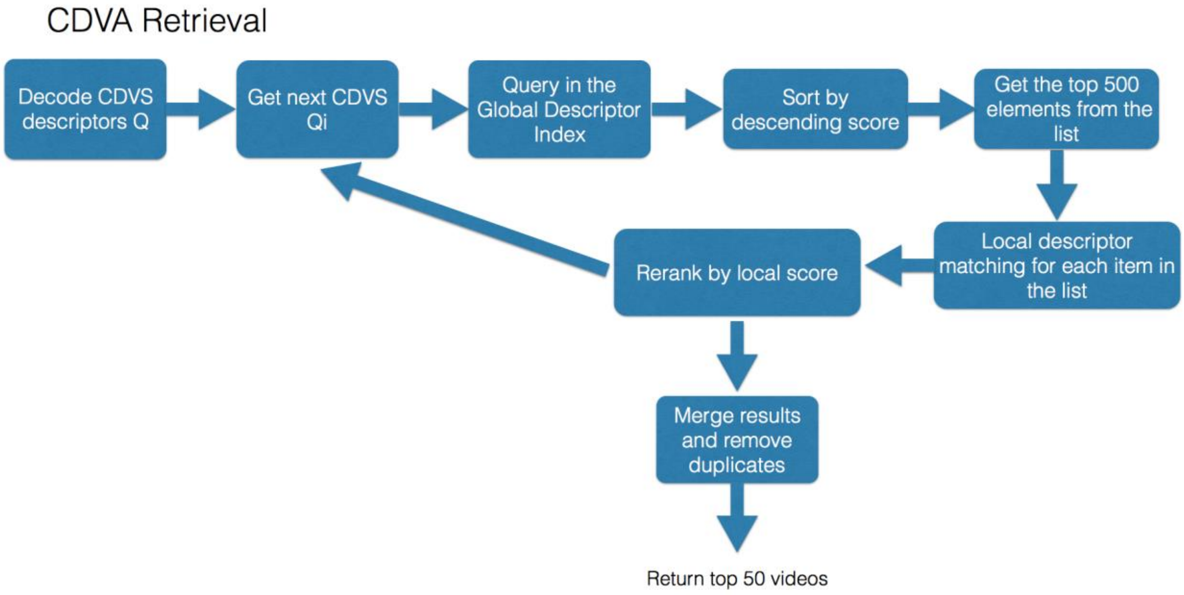

CDVA Retrieval
Decode CDVS descriptors Q
Get next CDVS Qi
Query in the Global Descriptor Index
Sort by descending score
Get the top 500 elements from the list
Local descriptor matching for each item in the list
Rerank by local score
Merge results and remove duplicates
Return top 50 videos

Figure 20. The CDVA retrieval pipeline.

Figure 20 illustrates the CDVA retrieval operation. This figure assumes that a database index has been built and is in place.

1. **Decode**: decode all CDVS descriptors stored in the Query CDVA descriptor.
2. **Get next**: for all CDVS descriptors in the Query CDVA descriptor, execute the following steps.
3. **Query in the GD Index**: perform a CDVS retrieval operation using query Qi on the Global descriptor database index.
4. **Sort**: sort the results by descending score.
5. **Local descriptor matching**: perform CDVS local descriptor matching on the top 500 results.
6. **Rerank**: rerank by local score.
7. **Merge results**: merge results removing duplicates.
8. Return the top 50 videos.

7.5 Results

The following results have been computed running the modified CXM on the CDVA Dataset and using the evaluation framework contained in the CDVA Experimentation Model (CXM) version 0.1. The CDVA Dataset is described in detail in [62] however in the following tables we report the main figures for the convenience of the reader:

Type	Items of interest	Instances	
		Videos	Images
Unmodified (Direct & Partial)	796	5029	260
Modified (Direct & Partial)	123	4686	0
Total	796	9715	260

Table 10. Query set.

Type	Items of interest	Instances	
		Videos	Images
All	796	5128	0

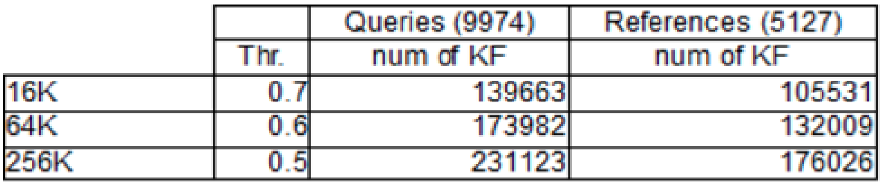
Table 11. Reference set.

Source	Videos	Type
MediaEval Blip	4701	UGC
OpenImages.eu (various collections)	3789	Broadcast, archival, education

Table 12. Distractor set.

The total size of the CDVA Dataset is 926 GB of compressed videos. One file had to be removed during the experiments from both the query and reference set because it was difficult to decode (it had been encoded using an obsolete proprietary video encoder).

The experiments were performed by building and running the proposed C++ source code on a 64-bit Linux server. In the Table 13, we show the variation of the number of extracted key frames when using different thresholds in the colour histogram comparison performed by the key frame selector.



		Queries (9974)	References (5127)
	Thr.	num of KF	num of KF
16K	0.7	139663	105531
64K	0.6	173982	132009
256K	0.5	231123	176026

Table 13. Number of keyframes.

Table 14 contains the measurements required by the MPEG Evaluation Framework for Compact Descriptors for Video Analysis [62] It comprises the average and maximum lengths (in bytes per second of video content) of the compressed descriptors, the retrieval and pairwise matching performances for each of the three operating points (16K, 64K, 256K). Please note that the operating points are just upper bounds that apply to the average query and reference descriptor lengths (Bps).

```
                    Descriptor lengths (Bps):        16K         64K        256K
                       Query average lengths:    4295.35     5234.96     6792.56
                           Query max lengths:  271916.92   300389.86   376271.27
                   Reference average lengths:    5621.77     6894.29     9037.02
                       Reference max lengths:  271916.92   300389.86   376271.27


                    Retrieval performance at:        16K         64K        256K
                      Mean average precision:      0.736       0.744       0.747
                                 r-Precision:      0.729       0.736       0.738

            Pairwise matching performance at:        16K        64K       256K   16K_256K
True positive rate at 1% false positive rate:      0.837      0.844      0.848      0.838
Mean Jaccard index for temporal localisation:      0.547      0.572      0.586      0.548
```

Table 14. Descriptor lengths and results.

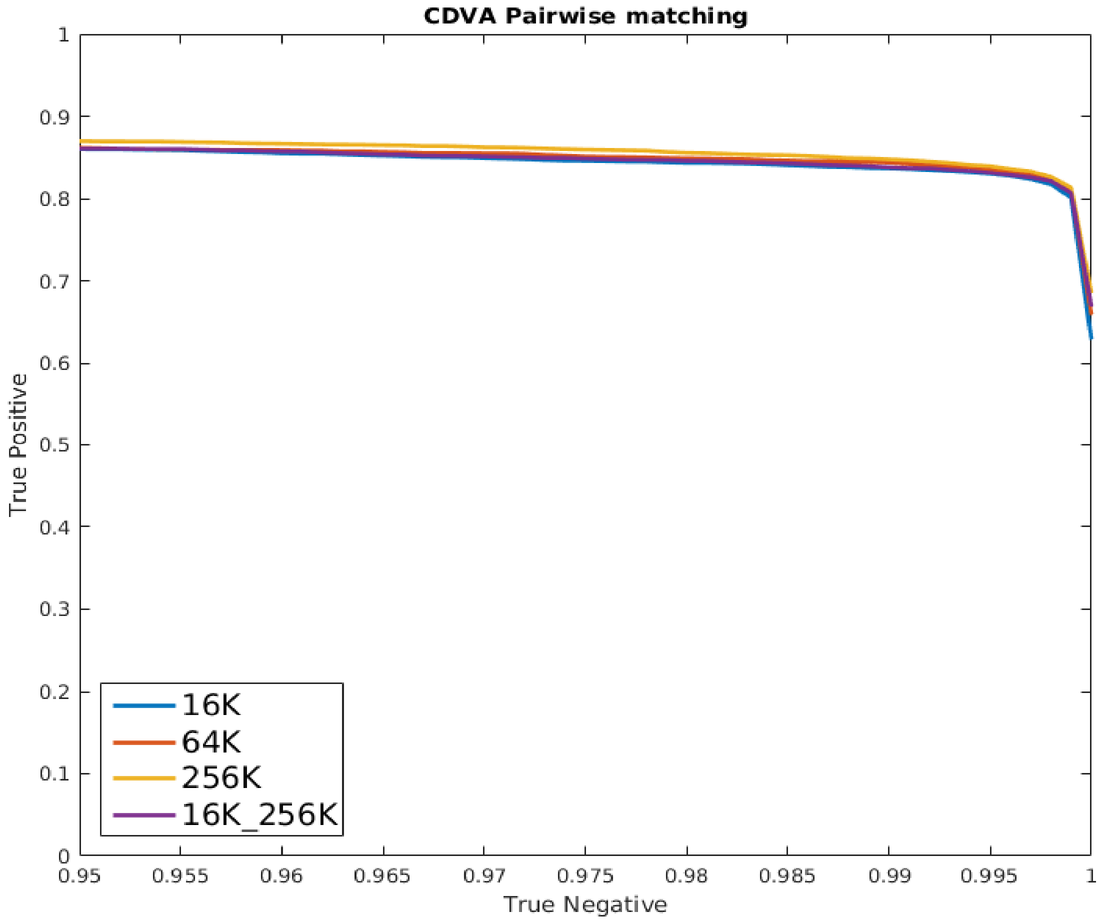


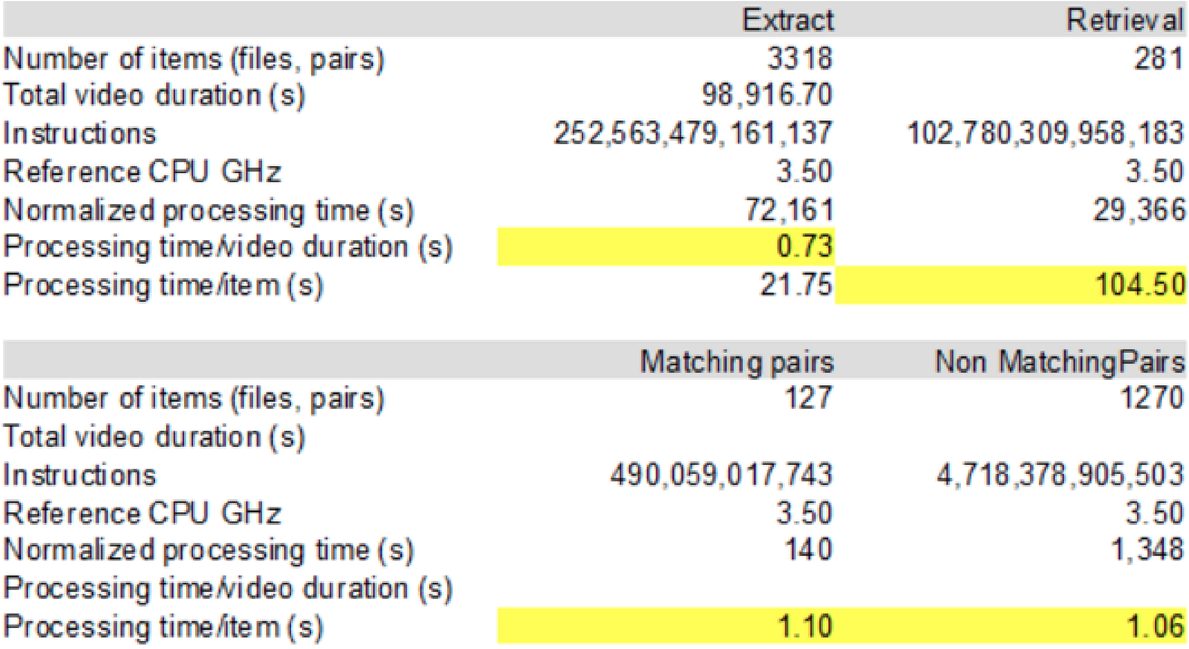
Figure 21. Pairwise matching ROC graph.

7.5.1 Complexity measurements

Complexity measurements are reported in the following tables. The measured operations are: extraction of the "*TimingExtract.txt*" list, matching of the "*TimingMatchingPairs.txt*" list, and matching of the "*TimingNonMatchingPairs.txt*" list of the CDVA Dataset. Retrieval measurements were taken using the "*TimingRetrieval.txt"* list.

7.5.2 Processing times

All reported times are normalized using the reference platform CPU characteristics (Intel Core I7-5930K), assuming that the CPU is operating at a base frequency of 3.5 GHz, in single thread execution, at an average rate of one instruction per cycle, without GPU acceleration. The cells marked yellow are values that are compared against the corresponding CDVA thresholds.



	Extract	Retrieval
Number of items (files, pairs)	3318	281
Total video duration (s)	98,916.70	
Instructions	252,563,479,161,137	102,780,309,958,183
Reference CPU GHz	3.50	3.50
Normalized processing time (s)	72,161	29,366
Processing time/video duration (s)	0.73	
Processing time/item (s)	21.75	104.50

	Matching pairs	Non MatchingPairs
Number of items (files, pairs)	127	1270
Total video duration (s)		
Instructions	490,059,017,743	4,718,378,905,503
Reference CPU GHz	3.50	3.50
Normalized processing time (s)	140	1,348
Processing time/video duration (s)		
Processing time/item (s)	1.10	1.06

Table 15. Processing times.

The Evaluation Framework document states that the following limits for average time numbers on the reference platform must be met:

1. extraction time: must not exceed 10 seconds per second of decoded video content.
2. pairwise matching time: must not exceed 1 second per pair.
3. retrieval time: must not exceed 60 seconds per query.

The conditions are partially met. In particular, condition 2 and 3 are not fully met.

7.5.3 Peak memory usage

The peak memory usage was detected using the *run-memory-test.pl* script of the CXM, based on the Linux "time" tool.

Operation	Maximum resident set size (kbytes)
cdva extract	354,328
cdva match (matching pair)	36,764
cdva match (non-matching pair)	31,204

cdva retrieve	17,709,228

Table 16. Peak memory usage.

7.6 Implementation

The code implementing extraction, pairwise matching and retrieval is written in C++ using the structure proposed in the CDVA Evaluation Framework; in particular, all the high level implementation details are contained in the "*CdvaImpl*" C++ class. The code is compliant to both C++98 and C++11, and can be compiled and run on both Linux and Windows.

At a lower level of detail, the "*cdvscore"* subdirectory contains other C++ classes that provide a modified version of the CDVS library which has been used as a low level library to perform keypoint detection, extraction, binary encoding of descriptors and coordinate coding. A C++ class for RVD encoding and matching is also provided here. Moreover, the proposed implementation uses the following external libraries: opencv, vlfeat, rescaler, eigen.

7.7 Standardization outcome

The proposal was partially accepted as part of the CXM 0.2 at the 115th MPEG meeting. The adopted part is the key frame selection mechanism. The decision about the other elements has been postponed until the next meeting in October 2016.

7.8 Future proposal to be submitted to MPEG CDVA

Video DISTRAT has been implemented in C++ and integrated with the CXM 0.2 (CDVA Experimentation Model). It will be used in the extensive experiments defined by the CDVA Evaluation Framework, to verify the performance of the proposed solution.

The goal is to make a proposal to MPEG at the 116th meeting that will be held in October 2016.

8 Conclusions

In the second part of the project, WP5 has made a significant progress in the development of the component technologies with BSOTA performance and successfully integrated them into the second version of the BRIDGET VS engine.

The most significant results include:

- ❖ Successful development of a complete Visual Search pipeline for video content (mark 2), and its integration into the first release of the BRIDGET AT.
- ❖ Extensions to our descriptor aggregation scheme – Robust Visual Descriptor (RVDW), which advances significantly beyond SOTA in terms of recognition performance and speed.
- ❖ Integration of RVDW with deep features, demonstrating world-class performance.
- ❖ Extended our geometric consistency check to video: the multi-frame DISTRAT, with unique features and outstanding performance.
- ❖ Research papers published at leading conferences (CVPR 2016, ICME 2016), including the "Best paper award" for the work entitled ”On Aggregation of Local Binary Descriptors", presented at the 3rd IEEE International Mobile Multimedia Computing workshop (ICME). The work on RVDW and local binary descriptors has been also submitted to IEEE Transactions PAMI and IEEE Transaction on Multimedia.
- ❖ 8 contributions to the MPEG CVDS/CDVA standardization work with significant impact, including (1) a response to the CfP with leading performance and (2) a successful proposal on temporal sampling in CE1 m38664.

9 References

[1] BRIDGET Deliverable D8.3, “Material Library and Ground Truth – Version A”, June 2015.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, January 2004.

[3] MPEG, "ISO/IEC 15938:13 - Information technology - Multimedia content description interface - Part 13: Compact descriptors for visual search," August 2015.

[4] MPEG, "ISO/IEC 15938:14 - Information technology - Multimedia content description interface - Part 14: Reference software, conformance and usage guidelines for compact descriptors for visual search," October 2015.

[5] Lei Chen, Xiangmin Zhou, "ASVTDECTOR: A Practical Near Duplicate Video Retrieval System," 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 1348 - 1351, April 2013.

[6] Hua-Tsung Chen, Suh-Yin Lee, Chien-Li Chou, "Pattern-Based Near-Duplicate Video Retrieval and Localization on Web-Scale Videos," IEEE Transactions on Multimedia, vol. 17, no. 3, pp. 382-395, March 2015.

[7] Christian Beecks, Thomas Seidl, Merih Seran Uysal, "On Efficient Content-based Near-duplicate Video Detection," 2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1-6, June 2015.

[8] Frederic Lefebvre, Alexey Ozerov, Vignesh Srinivasan, "Shot aggregating strategy for near-duplicate video retrieval," 2015 23rd European Signal Processing Conference (EUSIPCO), pp. 1825 - 1829, August 2015.

[9] Shen-Chuan Tai, Guo-Shiang Lin, Tang-You Chang, "A Near-Duplicate Video Retrieval Method Based on Zernike Moments," Proceedings of APSIPA Annual Summit and Conference 2015, pp. 860 - 864, December 2015.

[10] M. M. Waghmare, Amrit Priyadarshi, Laxmikant S. Kate, "An approach for automated video indexing and video search in large lecture video archives," 2015 International Conference on Pervasive Computing (ICPC), pp. 1-5, January 2015.

[11] A. Zisserman, J. Sivic, "Video google: A text retrieval approach to object matching in videos," International Conference on Computer Vision, pp. 1470-1477, 2003.

[12] H. Stewenius, D. Nister, "Scalable recognition with a vocabulary tree," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2161-2168, 2006.

[13] C. Schmid, J. Ponce, S. Lazebnik, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169-2178, 2006.

[14] Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, Yihong Gong, Jinjun Wang, "Locality-constrained linear coding for image classification," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360-3367, June 2010.

[15] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, “Aggregating local image descriptors into compact codes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp.1704–1716, Sept. 2012.

[16] Christopher Dance, Florent Perronnin, "Fisher kernels on visual vocabularies for image categorization," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, June 2007.

[17] Jose Martifnez-Carraza, Sergio Escalerat, Victor Ponce-Lopezt, Xavier Baro, Hugo Jair Escalante, "Improving bag of visual words representations with genetic programming", International Joint Conference on Neural Networks, 2015

[18] Bhagyashri Patil, Bela Joglekar, Parag Kulkarni, "An Effective Content Based Video Analysis and Retrieval Using Pattern Indexing Techniques," 2015 International Conference on Industrial Instrumentation and Control, 2015.

[19] Wided Souidene, Azeddine Beghdadi, Sameh Megrhi, "Spatio-temporal salient feature extraction for perceptual content based video retrieval," Colour and Visual Computing Symposium, September 2013.

[20] Nalini B. Yadav, Sudeep D. Thepade, "Assessment of Similarity Measurement Criteria in Thepade's Sorted Ternary Block Truncation Coding (TSTBTC) for Content Based Video Retrieval," in International Conference on Communication, Information & Computing Technology, 2015.

[21] Attilio Fiandrotti, Enrico Magli, Luca Bertinetto, "Shot-based object retrieval from video with compressed Fisher Vectors," 2014 22nd European Signal Processing Conference (EUSIPCO), pp. 2210 - 2214, September 2014.

[22] A. Zisserman, R. Arandjelovic, "Three things everyone should know to improve object retrieval," IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), pp. 2911–2918, 2012.

[23] K Raghurama Holla, M Sharmila Kumari, B H Shekar, "Video retrieval: An accurate approach based on Kirsch descriptor," 2014 International Conference on Contemporary Computing and Informatics, pp. 1203-1207, November 2014.

[24] R. Porter, T. Drummond, E. Rosten, "Faster and better: A machine learning approach to corner detection," Pattern Recognition and Machine Intelligence, vol. 8251, pp. 327–334, 2013.

[25] Cheng Cai, Yahui Li, "Video segment retrieval based on affine hulls," 2015 10th Asian Control Conference (ASCC), pp. 1-6, May 2015.

[26] Baroffio Luca, "Visual analysis tools for energy aware heterogeneous networks," Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Tesi di dottorato 2015.

[27] Vijay Chandrasekhar, S Tsai, David Chen, Bernd Girod, Mina Makar, "Interframe Coding of Feature Descriptors for Mobile Augmented Reality," IEEE Transactions on Image Processing, no. 23, pp. 3352-3367, 2014.

[28] ISO/IEC JTC1/SC29/WG11 MPEG2014/M32330, "Improved RVD in TM8 - CE 2 Response from University of Surrey and Visual Atoms", January 2014, San Jose, USA

[29] ISO/IEC JTC1/SC29/WG11 MPEG2016/M37880, "BRIDGET Response to the MPEG CfP for Compact Descriptors for Video Analysis (CDVA) - Search and Retrieval", February 2016, San Diego, USA

[30] BRIDGET Deliverable "D4.3: Media Analysis Tools - Report - Version B", 2016

[31] ISO/IEC JTC1/SC29/WG11 MPEG2015/N15729, "Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval – Version 2.0", October 2015, Geneva, CH

[32] ISO/IEC JTC1/SC29/WG11 MPEG2016/M38664, "BRIDGET Report on CDVA Core Experiment 1 (CE1)", May2016, Geneva, CH

[33] Liu L., Fan B., Zhao J., Research on Motion Attention Fusion Model-Based Video Target Detection and Extraction of Global Motion Scene, Journal of Signal and Information Processing, Scientific Research, IEEE, v 4, pp. 30–35, April 2013.

[34] Canny, J., A Computational Approach to Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[35] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 744–755, 2014.

[36] A. Babenko and V. S. Lempitsky, "Aggregating deep convolutional features for image retrieval," CoRR, 2015.

[37] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual instance retrieval with deep convolutional networks," CoRR, 2014.

[38] J. Delhumeau, P. Gosselin, H. Jegou, and P. Perez, "Revisiting the VLAD image representation," in ACM Multimedia, Barcelona, Spain, Oct. 2013.

[39] H. Jégou and A. Zisserman, "Triangulation embedding and democratic aggregation for image search," in IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, 2014.

[41] R. Arandjelović and A. Zisserman, "All about VLAD," in IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[42] E. S. Xioufis, S. Papadopoulos, Y. Kompatsiaris, G. Tsoumakas, and I. P. Vlahavas, “A comprehensive study over VLAD and product quantization in large-scale image retrieval,” IEEE Transactions on Multimedia, pp. 1713–1728, 2014.
[43] C. Eggert, S. Romberg, and R. Lienhart, “Improving VLAD: hierarchical coding and a refined local coordinate system,” in IEEE International Conference on Image Processing, 2014, pp. 3018–3022.
[44] Z. Liu, H. Li, W. Zhou, T. Rui, and Q. Tian, “Uniforming residual vector distribution for distinctive image representation,” Circuits and Systems for Video Technology, IEEE Transactions on, 2015.
[45] R. Negrel, D. Picard, and P.-H. Gosselin, “Web scale image retrieval using compact tensor aggregation of visual descriptors,” IEEE Transactions on Multimedia, pp. 24–33, Mar 2013.
[46] T.-T. Do, Q. D. Tran, and N.-M. Cheung, “Faemb: A function approximation-based embedding method for image retrieval,” in The IEEE Conference on Computer Vision and Pattern Recognition, June 2015.
[47] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in European Conference on Computer Vision, 2014.
[48] M. Norouzi, A. Punjanio, and D. Fleet. Fast exact search in hamming space with multi-index hashing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(6):1107–1119, 2014.
[49] E. Ong and M. Bober, Improved Hamming Distance Search using Variable Length Hashing, CVPR 2016, July 2016
[50] Davenport, G., Smith, T. A. & Pincever, N. Cinematic primitives for multimedia. IEEE Computer Graphics and Applications, 11(4), 1991, pp. 67-74.
[51] S. Lepsoy, G. Francini, G. Cordara, P.P. de Gusmao, “Statistical modelling of outliers for fast visual search”, IEEE International Conference on Multimedia and Expo, 2011.
[52] COTSACES, Costas; NIKOLAIDIS, Nikos; PITAS, Ioannis. Video shot detection and condensed representation. A review. Signal Processing Magazine, IEEE, 2006.
[53] R.J. Larsen and M.L. Marx. An Introduction to Mathematical Statistics and its Applications. Prentice-Hall, 1986.
[54] FOX, Edward A.; SHAW, Joseph A. Combination of multiple searches. NIST SPECIAL PUBLICATION SP, 1994, 243-243.
[55] ZHOU, Xin; DEPEURSINGE, Adrien; MULLER, Henning. Information fusion for combining visual and textual image retrieval. In: Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010. p. 1590-1593.
[56] ISO/IEC JTC 1/SC 29 (MPEG): Text of ISO/IEC CD 15938-14 Reference software, conformance and usage guidelines for compact descriptors for visual search. Output document N15371 of the 112th MPEG meeting, Warsaw 2015.
[57] N16064, “CDVA Experimentation Model (CXM) 0.1”, February 2016, San Diego, US
[58] N16065, “Description of Core Experiments in CDVA”, February 2016, San Diego, US
[59] N15938, “Results of the Call for Proposals on CDVA”, February 2016, San Diego, US
[60] m37880, “BRIDGET Response to the MPEG CfP for Compact Descriptors for Video Analysis (CDVA) - Search and Retrieval”, February 2016, San Diego, US
[61] N15339, “Call for Proposals for Compact Descriptors for Video Analysis (CDVA) - Search and Retrieval”, June 2015, Warsaw, PL
[62] N15729, Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval – Version 2.0, October 2015, Geneva, CH
[63] N15040, Compact Descriptors for Video Analysis: Requirements for Search Applications, Oct. 2015, Strasbourg, FR.
[64] N14507, Compact Descriptors for Video Analysis: Objectives, Applications and Use Cases, Apr. 2014, Valencia, ES.
[65] m32330, “Improved RVD in TM8 - CE 2 Response from University of Surrey and Visual Atoms”, January 2014, San Jose, USA
[66] m22672, “Telecom Italia's response to the MPEG CfP for Compact Descriptors for Visual Search”, November 2011, Geneva, Switzerland
[67] Tsai, S. S., Chen, D., Takacs, G., Chandrasekhar, V., Singh, J. P., & Girod, B. (2009). Location Coding for Mobile Image Retrieval. Proceedings of the 5th International ICST Mobile Multimedia Com-

munications Conference (Mobimedia'09). London, UK: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[68] N15765, "Text of ISO/IEC DIS 15938-14 of CDVS Reference Software and Conformance Testing", Oct. 2015, Geneva, CH.

[69] ISO/IEC 15938-13:2015(en) "Information technology - Multimedia content description interface - Part 13: Compact descriptors for visual search"