Event retrieval in large video collections with circulant temporal encoding

Jérôme Revaud, Matthijs Douze, Cordelia Schmid, Hervé Jégou

Event retrieval in large video collections with circulant temporal encoding

Jérôme Revaud Matthijs Douze Cordelia Schmid Hervé Jégou
INRIA

Abstract

This paper presents an approach for large-scale event retrieval. Given a video clip of a specific event, e.g.*, the wedding of Prince William and Kate Middleton, the goal is to retrieve other videos representing the same event from a dataset of over 100k videos. Our approach encodes the frame descriptors of a video to jointly represent their appearance and temporal order. It exploits the properties of circulant matrices to compare the videos in the frequency domain. This offers a significant gain in complexity and accurately localizes the matching parts of videos.*

Furthermore, we extend product quantization to complex vectors in order to compress our descriptors, and to compare them in the compressed domain. Our method outperforms the state of the art both in search quality and query time on two large-scale video benchmarks for copy detection, TRECVID *and* CCWEB. *Finally, we introduce a challenging dataset for event retrieval, EVVE, and report the performance on this dataset.*

1. Introduction

This paper introduces an approach for specific event retrieval. Examples of events are news items such as the wedding of prince William and Kate, or re-occurring events such as the eruption of a geyser. Indexing this type of video material on-line and in archives will benefit to many. Home users will enhance their viewing experience via automatic linking of their digital library. Professional users will find video data in large archives, that are often indexed with irrelevant keywords and, thus, inaccessible.

Searching for specific events is related to video copy detection [13] and event category recognition [16], but there are substantial differences with both. The goal of video copy detection is to find deformed videos, *e.g.*, by compression, cam-cording or picture-in-picture combinations. Detecting event *categories* requires a classification approach that captures the large intra-class variability. The method introduced in this paper is tailored to specific event retrieval, as it is flexible enough to handle significant viewpoint change while still producing a precise alignment in time. Our first contribution is to encode the frame descriptors of a video into a temporal representation and to exploit the properties of circulant matrices to compare videos in the frequency domain. The second contribution is a dataset for specific event retrieval in large user-generated video content. This dataset, named EVVE, has been collected from Youtube and comprises a set of manually annotated videos of 13 events, as well as 100,000 distractor videos.

Many techniques for video retrieval represent a video as a set of descriptors extracted from frames or keyframes [4, 11, 20]. Searching in a collection is performed by comparing the query descriptors with those of the dataset. Then, temporal constraints are enforced on the matching descriptors, by e.g., partial alignment [22] or classic voting techniques, such as temporal Hough transform [4], which was popular in the TRECVID video copy detection task [19]. Such approaches are costly, since all frame descriptors of the query must be compared to those of the database before performing the temporal verification. Another possibility is to summarize a video in a "Seam image" [23]. This works for near-duplicate search but cannot handle severe transformations like large viewpoint changes.

In contrast, the technique proposed in this paper measures the similarity between two sequences for all possible alignments. Frame descriptors are jointly encoded in the frequency domain, where convolutions cast into efficient element-wise multiplications. This encoding is combined with frequency pruning to avoid the full computation of all cross-similarities between the frame descriptors. The comparison of sequences is improved by a regularization in the frequency domain. Computing a matching score between videos only requires component-wise operations and a single one-dimensional inverse Fourier transform, avoiding the reconstruction of the descriptor in the temporal domain. As a byproduct of the comparison, the approach precisely aligns the compared sequences. Similar techniques have been used in other contexts such as registration or watermark detection. However, they are usually applied to the raw signal such as image pixels [3, 6] or audio waveforms [10]. Recently, transforming a multi-dimensional signal to the Fourier domain to speed up detection was shown useful [5], but to our knowledge, it is new to analyze the temporal aspect of global image descriptors in this way.

The tradeoff between search quality, speed and memory usage is optimized with the product quantization technique [9], which is extended to complex vectors in order to compare our descriptors in the compressed Fourier domain.

The paper is organized as follows. Section 2 introduces the EVVE dataset and its evaluation protocol. Section 3 describes frame descriptors, Section 4 describes our temporal circulant encoding technique and Section 5 presents our indexing strategy. The experiments in Section 6 demonstrate the excellent results of our approach for event retrieval on the EVVE dataset. Our approach also significantly outperforms state-of-the-art systems for efficient video copy detection on the TRECVID and CCWEB benchmarks.

2. EVVE: an event retrieval dataset

This section introduces the EVVE (EVent VidEo) dataset which is dedicated to the retrieval of particular events. This differs from recognizing event *categories* such as "birthday party" or "grooming an animal", as in the TRECVID Multimedia event detection task [16]. Figure 1 presents the 13 events. Several of them are localized precisely in time and space as professional reporters and spectators have captured the same event simultaneously. An example is the event "Concert of Madonna in Rome 2012". In this case, the videos overlap visually and can be aligned. EVVE also includes events for which relevant videos might not correspond to the same instance in place or time. For instance, the event "The major autumn flood in Thailand in 2011" is covered by videos of the flood in different places, and "Austerity riots in Barcelona" includes shots of riots at different places and moments. Finally, there are re-occurring events, which are well localized but re-occur temporally, such as "Eruption of Strokkur geyser in Iceland" and "Jurassic Park ride in Universal Studios theme park". All videos have been collected from Youtube. Each event was annotated by one annotator, who first produced a precise definition of the event. For example, the event "The wedding of Prince William and Kate Middleton" is defined as:

> Images of Kate & William together on the wedding day in an official setting (either in the church, in the car or waving at the crowd from the balcony). A single image eg. in a slideshow is counted as positive. It is positive even if the main topic of the video is something else (eg. another wedding). Spoken text without a relevant image is annotated as negative.

The human annotators have marked the videos as either positive or negative. Ambiguous videos were removed.

Distractors. In addition to the videos collected for the specific events, we have also retrieved a set of 100,000 "distractor" videos by querying Youtube with unrelated terms. These videos have all been collected *before* September 2008, which ensures that the distractor set does not contain any of the relevant events of EVVE, since all events are temporally localized *after* September 2008 (except the

Figure 1. Illustration of the 13 events in our EVVE dataset. The number of queries (#q) and number of positives (#pos) are given for each event. The dataset is available at `http://lear.inrialpes.fr/data`.

re-occurring events #11 and #12). The distractor videos representing a similar but distinct event, such as videos of other bomb attacks for Event #9, are counted as negatives.

EVVE: Evaluation protocol. Evaluation is performed in a standard retrieval scenario, where we submit one video query at a time and the algorithm returns a list of videos ranked by similarity scores. We do not use audio or metadata in this paper, but they are provided along with the dataset. We evaluate the average precision (AP) for each query. The mean AP [18] (mAP) is computed per event, by averaging the individual APs for this event. As a synthetic measure of the overall performance, we compute the average of the mAPs over the 13 different events (avg-mAP measure).

3. Frame description

We represent a video by a sequence of high-dimensional frame descriptors, as described in this section.

Pre-processing. All videos are mapped to a common format, by sampling them at a fixed rate of 15 fps and resizing them to a maximum of 120k pixels, while keeping the aspect ratio.

Local description. Local SIFT descriptors [14] are extracted for each frame on a dense grid [15], every 4 pixels and for 5 scale levels. We square root the SIFT components and reduce the descriptor to 32 dimensions with principal component analysis (PCA) [1, 7]. We chose to use dense sampling rather than interest points, as this increases the accuracy without impacting the storage size after they are aggregated.

Descriptor aggregation. The SIFT descriptors of a frame are encoded using MultiVLAD [8], a variant of the Fisher vector [17]. Two VLAD descriptors are obtained from two different codebooks of size 128, and concatenated. Power-law normalization is applied to the vector and it is reduced by PCA to dimension d (a parameter of our approach). The vector is normalized using the PCA's covariance matrix and L2-normalized.

Our implementation performs the entire description step in real time (15 fps) on a single processor core.

4. Circulant temporal aggregation

The method introduced in this section aims at comparing two sequences of frame descriptors $\boldsymbol{q} = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_m] \in \mathbb{R}^{d\times m}$ and $\boldsymbol{b} = [\boldsymbol{b}_1, \dots, \boldsymbol{b}_n] \in \mathbb{R}^{d\times n}$. We first consider the metric

$$s_\delta(\boldsymbol{q}, \boldsymbol{b}) = \sum_{t=-\infty}^{\infty} \langle \boldsymbol{q}_t, \boldsymbol{b}_{t-\delta} \rangle , \qquad (1)$$

where the vectors $\boldsymbol{q}_t$ (resp., $\boldsymbol{b}_t$) are zero when $t < 1$ and $t > m$ (resp., $t > n$). This is an extension of the *correlation* used for pattern detection in scalar signals [12]. The metric $s_\delta(\boldsymbol{q}, \boldsymbol{b})$ reaches a maximum in δ when the $\boldsymbol{q}$ and $\boldsymbol{b}$ are aligned if the following assumptions are satisfied:

Assumption 1: There is no (or limited) temporal acceleration. This hypothesis is assumed by the "temporal Hough transform" [4] when only the shift parameter is estimated.

Assumption 2: The inner product is a good similarity between individual frames. This is the case for Fisher and our Multi-VLAD descriptors (Section 3), but not for other type of descriptors to be compared with complex kernels.

Assumption 3: The sum of similarities between the frame descriptors reflects the similarity of the sequences. In practice, this assumption is not well satisfied, because the videos are very self-similar in time, so the similarity proposed in Eqn. 1 is suboptimal. In the case of the temporal Hough transform, this problem is avoided by considering only the per-frame nearest neighbors.

The encoding technique for sequences of vector descriptors presented in this section, is referred to as *Circulant Temporal Encoding* (CTE). It strongly relies on Fourier-domain processing and includes regularization techniques that address the limitations mentioned in Assumption 3 (see Subsection 4.2).

4.1. Circulant encoding of vector sequences

Eqn. 1 can be decomposed along the dimensions of the descriptor. Using the column notation $\boldsymbol{q} = [\boldsymbol{q}_{\bullet 1}^\top, \dots, \boldsymbol{q}_{\bullet d}^\top]^\top$ and $\boldsymbol{b} = [\boldsymbol{b}_{\bullet 1}^\top, \dots, \boldsymbol{b}_{\bullet d}^\top]^\top$, the vector of scores for all possible values of δ is given by

$$\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b}) = [\dots \boldsymbol{s}_0(\boldsymbol{q}, \boldsymbol{b}), \boldsymbol{s}_1(\boldsymbol{q}, \boldsymbol{b}) \dots] = \sum_{i=1}^{d} \boldsymbol{q}_{\bullet i} \otimes \boldsymbol{b}_{\bullet i} \quad (2)$$

where $\otimes$ is the convolution operator. Assuming sequences of equal lengths ($n = m$), $\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b})$ can be computed in the Fourier domain [12]. Denoting by $\mathcal{F}$ the 1D-Discrete Fourier transform and $\mathcal{F}^{-1}$ its inverse, the convolution theorem states that:

$$\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b}) = \sum_{i=1}^{d} \mathcal{F}^{-1} \left(\mathcal{F}(\boldsymbol{q}_{\bullet i})^* \odot \mathcal{F}(\boldsymbol{b}_{\bullet i}) \right) \qquad (3)$$

where $\odot$ is the element-wise multiplication of 2 vectors. Denoting $\mathcal{Q}_i = \mathcal{F}(\boldsymbol{q}_{\bullet i}) \in \mathbb{C}^m$ and $\mathcal{B}_i = \mathcal{F}(\boldsymbol{b}_{\bullet i}) \in \mathbb{C}^n$, the linearity of the Fourier operator gives:

$$\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b}) = \mathcal{F}^{-1} \left(\sum_{i=1}^{d} \mathcal{Q}_i^* \odot \mathcal{B}_i \right) , \qquad (4)$$

which is more efficient to compute than Eqn. 3 because it requires a single inverse FFT instead of d, while performing the same number of component-wise multiplications.

In practice, we rely on the Fast Fourier Transform (FFT) and its inverse, which are very efficient, especially for sequences whose length is power of two. As a common practice, the descriptor sequences are padded with zeros to reach the next power of two [12]. Unless stated otherwise, we consider hereafter that the sequences have been preprocessed to have the same length $m = n = 2^\ell$.

4.2. Regularized comparison metric

As mentioned above, due to the temporal consistency and more generally the self-similarity of frames in videos, the values of the score vector $\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b})$ are noisy and its peak over δ is not precisely localized. This is shown by comparing the query to itself. Ideally, one would expect a Dirac-like response: $s_\delta(\boldsymbol{q}, \boldsymbol{q}) = 0$ for $\delta \neq 0$, and $s_0(\boldsymbol{q}, \boldsymbol{q}) = 1$. This behavior can be achieved through an additional filtering stage in the Fourier domain. Formally, we search a set of filters $W = \{W_1, \dots, W_d\}, W_i \in \mathbb{R}^n$ satisfying

$$\begin{aligned} \boldsymbol{s}_W(\boldsymbol{q}, \boldsymbol{q}) &= \mathcal{F}^{-1}\left(\sum_{i=1}^{d} W_i \odot \mathcal{Q}_i^* \odot \mathcal{Q}_i\right) \\ &= [1, 0, \dots 0] = \boldsymbol{e}_1. \end{aligned} \tag{5}$$

For the sake of simplicity, we compute W_i assuming that the contributions are shared equally across dimensions:

$$\mathcal{F}^{-1}\left(W_i \odot \mathcal{Q}_i^* \odot \mathcal{Q}_i\right) = \frac{1}{d}\boldsymbol{e}_1 \quad \forall i = 1..d \tag{6}$$

$$W_i \odot \mathcal{Q}_i^* \odot \mathcal{Q}_i = \frac{1}{d}\mathcal{F}\left(\boldsymbol{e}_1\right) = \frac{1}{d}[1, 1, \dots 1], \tag{7}$$

$$W_i = \frac{1}{d}\frac{1}{\mathcal{Q}_i^* \odot \mathcal{Q}_i}, \tag{8}$$

where all operations are performed element-wise. The filter W can be interpreted as a peak detector in $\boldsymbol{s}(\boldsymbol{q}, \boldsymbol{b})$. In practice, its spectrum resembles that of a Laplacian filter.

One major drawback is that the denominator in Eqn. 8 may be close to zero, magnifying the noise and introducing instability in the solution. To tackle this issue, Bolme et al. [2] proposed to average the filters obtained from independent samples, which helps when some frequencies have small energy for some dimensions. In our case, we could average the filters W_i, since they are decorrelated by the PCA (Section 4). Unfortunately, averaging does not always suffice, as many videos contain only one shot composed of a single frame: the components associated with high frequencies are almost 0 for all dimensions. Therefore, we propose instead to incorporate a regularization term into Eqn. 5 and to minimize over W_i:

$$\lambda \left\|W_i\right\|^2 + \left\|\mathcal{F}^{-1}\left(W_i \odot \mathcal{Q}_i^* \odot \mathcal{Q}_i\right) - \frac{1}{d}\boldsymbol{e}_1\right\|^2, \tag{9}$$

where the regularization coefficient λ ensures the stability of the filter. Notice that setting $\lambda = 0$ amounts to solving Eqn. 7 and leads to the solution proposed in Eqn. 8. A closed-form solution to this minimization problem in the Fourier domain, obtained by leveraging properties of circulant matrices, consists of adding λ to the denominator in Eqn. 8 [6]. This leads to a regularized score between two video sequences $\boldsymbol{q}$ and $\boldsymbol{b}$:

$$\boldsymbol{s}^\lambda(\boldsymbol{q}, \boldsymbol{b}) = \frac{1}{d}\mathcal{F}^{-1}\left(\sum_{i=1}^{d} \frac{\mathcal{Q}_i^* \odot \mathcal{B}_i}{\mathcal{Q}_i^* \odot \mathcal{Q}_i + \lambda}\right). \tag{10}$$

Both regularization techniques, *i.e.*, averaging the filters and using a regularization term, are complementary and hence combined. The choice of λ is discussed in Section 6.

4.3. Boundary detection

The strategy presented above produces a set of scores $\boldsymbol{s}^\lambda(\boldsymbol{q}, \boldsymbol{b}) = [\dots, s_\delta^\lambda(\boldsymbol{q}, \boldsymbol{b}), \dots]$ between two videos sequences $\boldsymbol{q}$ and $\boldsymbol{b}$ for all possible temporal shifts. The time shift $\delta^* = \arg\max_{\delta \in \mathbb{Z}} s_\delta^\lambda(\boldsymbol{q}, \boldsymbol{b})$ gives the optimal alignment of the videos, and $s_{\delta^*}^\lambda(\boldsymbol{q}, \boldsymbol{b})$ is their similarity score.

In some applications such as video alignment (see Section 6), we also need the boundaries of the matching segments. For this purpose, the database descriptors are reconstructed in the temporal domain from $\mathcal{F}^{-1}(\boldsymbol{b}_{\bullet i})$. A frame-per-frame similarity is then computed with the estimated shift δ^*:

$$S_t = \left\langle \boldsymbol{q}_t, \boldsymbol{b}_{t-\delta^*} \right\rangle.$$

The matching sequence is defined as a set of contiguous t for which the scores S_t are high enough.

Note that, unlike the computation of $\boldsymbol{s}_{\delta^*}^\lambda(\boldsymbol{q}, \boldsymbol{b})$, this processing requires d distinct 1D inverse FFT, one per component. Yet, on large datasets this does not impact the overall efficiency, since it is only applied to a short-list of videos with the highest scores.

5. Indexing strategy and complexity

This section discusses the steps used to efficiently encode the descriptors in the Fourier domain. The goal is to implement the method presented in Section 4 in an approximate manner. Beyond the complexity gain already obtained from our Fourier-domain processing, this considerably improves the efficiency of the method while reducing its memory footprint by orders of magnitude. As shown in Section 6, this gain is achieved without significantly impacting the retrieval quality.

5.1. Frequency-domain representation

A database video $\boldsymbol{b}$ of length n is represented in the Fourier domain by a complex matrix $\mathcal{B} =$

$[\mathcal{B}_1^\top, \ldots, \mathcal{B}_d^\top]^\top = [\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{n-1}] \in \mathbb{C}^{d\times n}$. Our input descriptors are real-valued, so only half of the components are stored, as $\boldsymbol{f}_{n-i}$ is the complex conjugate of $\boldsymbol{f}_i$.

Frequency pruning is applied to reduce the video representation by keeping only a fraction $\beta \ll 1$ of the low-frequency vectors $\boldsymbol{f}_i$, $i = 0 \ldots \beta n - 1$ (in practice, β is an inverse power of 2). We keep a fraction rather than a fixed number of frequencies for all videos, as this would make the localization accuracy dependent on the sequence length.

Descriptor sizes. If $m \leq n$, we precompute a Fourier descriptor for different zero-padded versions of the query, *i.e.*, for all sizes 2^ℓ such that $m \leq 2^\ell \leq n_{\max}$, where $n_{\max}$ is the size of the longest database video.

We handle the case $m > n$ by noticing that the Fourier descriptor of the concatenation of a signal with itself is $[\boldsymbol{f}_0, 0, \boldsymbol{f}_1, 0, \boldsymbol{f}_2, 0, \ldots]$. Therefore, expanded versions of database descriptors can be generated on the fly and at no cost. This asymmetric processing of the videos was chosen for efficiency reasons. Unfortunately, this introduces an uncertainty on the alignment of the query and database videos: δ^* can be determined modulo n only.

5.2. Complex PQ-codes and metric optimization

In order to further compress the descriptors and to efficiently compute Eqn. 10, we propose two extensions of the product quantization technique [9], which is a compression technique that enables efficient compressed-domain comparison and search. The original technique proceeds as follows. A given database vector $y \in \mathbb{R}^d$ is split into p sub-vectors y_j, $i = 1 \ldots p$, of length d/p. The sub-vectors are separately quantized using k-means quantizers $q_i(.)$, $i = 1 \ldots p$. This produces a vector of indexes $[q_1(y_1), \ldots, q_p(y_p)]$. Typically, $q_i(y_i) \in [1, \ldots, 2^8]$.

The comparison between a query descriptor x and the database vectors is performed in two stages. First, the squared distances between each sub-vector x_j and all the possible centroids are computed and stored in a table $T = [t_{j,i}] \in \mathbb{R}^{p\times 256}$. This step is independent of the database size. Second, the squared distance between x and y is approximated as

$$d(x, y)^2 \approx \sum_{j=1}^{p} t_{j,q_j(y_j)}, \tag{11}$$

which only requires p look-ups and additions.

We adapt this technique to our context in two ways. First, it is extended to complex vectors in a straightforward manner. We learn the k-means centroids for complex vectors by considering a d-dimensional complex vector to be a $2d$-dimensional real vector, and this for all the frequency vectors that we keep: $\mathbb{C}^d \equiv \mathbb{R}^{2d}$ and $\boldsymbol{f}_j \equiv y_j$. At query time, the table T stores complex values.

As a second extension, we use product quantization to compute more structured quantities than distances. Instead of storing partial squared distances or Hermitian products, we directly pre-compute the partial sums involved in Eqn. 10 to further improve the efficiency. This is possible because Eqn. 11 only requires that the metric is separable (such as a sum, a product or a max).

As a result, our table T directly stores the partial sums for all possible centroids, which in our case includes the processing associated with the regularization filter. As with the regular product quantization technique, a single comparison only requires p look-ups and additions of complex numbers. The memory used for T is twice that of the original technique ($2 \times 256 \times p$) because of the complex values. This is a constant that does not depend on the database size.

Interestingly, the product quantization vocabularies do not need to be learned on representative training data: they can be trained on random Gaussian vectors in $\mathbb{R}^{(2d/p)}$. This is because the PCA whitening applied to generate $\boldsymbol{b}_j$ and the Fourier transform applied on $\boldsymbol{b}_{\cdot i}$ decorrelate the signal, which is close to Gaussian when it is encoded by PQ.

5.3. Summary of search procedure and complexity

Each database video is processed offline as follows:

1. The video is pre-processed and each frame is described as a d-dimensional Multi-VLAD descriptor.
2. This vector is padded with zeros to the next power of two, and mapped to the Fourier domain using d independent 1-dimensional FFTs.
3. High frequencies are pruned: Only $n' = \beta \times n$ frequency vectors are kept. After this step, the video is represented by $n' \times d$-dimensional complex vectors.
4. These vectors are separately encoded with a complex product quantizer, producing a compressed representation of $p \times n'$ bytes for the whole video.

At query time, the submitted video is described in the same manner. The complexity at query time depends on the number N of database videos, the dimensionality d of the frame descriptor and the video length, that we assume for readability to be constant (n frames):

1. $\mathcal{O}(d \times n \log n)$ – The query frame descriptors are mapped to the frequency domain by d FFTs.
2. $\mathcal{O}(256 \times p \times n')$ – The PQ table T associated with the query is pre-computed ($n' = n\beta \ll n$).
3. $\mathcal{O}(N \times p \times n')$ – Eqn. 10 is evaluated for all database vectors using the approximation of Eqn. 11, directly in the compressed domain using $n'p$ look-ups from T and additions. This produces a n'-dimensional vector for each database video.

dataset	query	database		
	videos	videos	hours	frames
CCWEB	24	13129	551	29.7M
CCWEB + 100k	24	113129	5921	320M
TRECVID CCD 08	2010	438	208	11.2M
EVVE	620	2375	166	8.9M
EVVE + 100k	620	102375	5536	299M

Table 1. Statistics on the datasets used in this paper.

4. $\mathcal{O}(N \times n' \log n')$ – This vector is mapped to the temporal domain using a single inverse FFT. Its maximum gives the time shift δ^* and the score $s^{\lambda}_{\delta^*}$.

As described in Section 5.1, the operations 1 and 2 are repeated for all sizes $n = 2^{\ell}$ found in the dataset. This doubles the runtime of the operations applied to $n = n_{\max}$. Only the steps 3 and 4 depend on the database size. They dominate the complexity for large databases.

6. Experiments

In this section we evaluate our approach, both for video copy detection and event retrieval. To compare the contributions of the frame descriptors and of the temporal matching, we introduce an additional descriptor obtained by averaging the frame descriptors (see section 3) over the entire video. This static descriptor is compared using the dot product and denoted by Mean-MultiVLAD (MMV).

6.1. Video copy detection

This task is evaluated on two public benchmarks, the CCWEB dataset [21] and the TRECVID 2008 content based copy detection dataset (CCD) [19], see Table 1. CCWEB contains 24 query videos, mostly focusing on near-duplicate detection. The transformed versions in the database correspond to user re-posts on video sharing sites. Large-scale performance is evaluated on CCWEB+100K obtained by adding the distractors from the EVVE dataset. Performance is reported as the mAP over all queries.

The 2008 campaign of the TRECVID CCD task is the last for which video-only results were evaluated. We present results on the camcording subtask, which is most relevant to our context of event retrieval in the presence of significant viewpoint changes. We report results with the official NDCR measure.

Compression parameters. The spatial and temporal compression is parametrized by the dimensionality d after PCA, the number p of PQ sub-quantizers and the frame description rate β, which defines the ratio between the number of frequency vectors and the number of video frames. As a general observation across all datasets and experiments, we notice that higher values of d yield better performance, for all values of p. Yet d should be kept reasonably small to avoid increasing the cost of the PCA projection. We thus fix the PCA output dimension to $d = 512$ in all our experiments and vary the number of sub-quantizers and the rate β.

method	PQ	β	perf.	memory usage	search time
CCWEB					
HIRACH [21]			0.952	-	-
MFH [20]			0.954	0.5 MB	-
MMV	no	-	0.971	26.9 MB	1.5 ms
MMV	64	-	0.969	0.8 MB	0.7 ms
MMV	16	-	0.962	0.2 MB	0.5 ms
CTE	no	1/64	**0.996**	2,960 MB	66.1 ms
CTE	no	1/1024	0.995	207 MB	4.8 ms
CTE	64	1/1024	0.994	3.6 MB	1.0 ms
CTE	16	1/1024	0.992	0.9 MB	0.5 ms
CCWEB + 100,000 distractors					
MFH [20]			0.866	5.3 MB	533 ms
MMV	16	-	0.887	1.8 MB	23 ms
CTE	16	1/1024	**0.960**	9.6 MB	75 ms
TRECVID CCD 08 – Camcording					
Best official result			0.079	10,000 MB	16 min
Douze & al. [4]			0.224	300 MB	191 s
MMV	no	-	0.967	0.9 MB	4 ms
CTE	no	1/8	**0.049**	8,600 MB	9.4 s
CTE	no	1/32	0.077	2,150 MB	2.2 s
CTE	64	1/8	**0.049**	134 MB	8.9 s

Table 2. Results for video copy detection. For CCWEB, the performance is measured with mAP (higher = better). For TRECVID the measure is NDCR (lower = better). Search times are given for one core and are averaged across queries.

Impact of the regularization parameter. The choice of λ depends on the task and the evaluation metric. For near-duplicate retrieval as well as for event retrieval, Figure 2 shows that intermediate values of λ yield the best performance. In contrast, we observe that small values of λ produce the best NDCR performance for the TRECVID copy detection task. This is probably due to the fact that the NDCR measure strongly favors precision over recall, whereas any matching tolerance obtained by a larger λ also produces more false positives. In all our experiments, we set λ=0.1 for the near-duplicate and event retrieval tasks, and λ=0.001 for the TV08 benchmark.

Comparison with the state of the art. Table 2 reports our results for near-duplicate and copy-detection for different compression trade-offs and compares our results to the state of the art. On CCWEB, both the temporal and non-temporal versions of our method outperform the state of the art for comparable memory footprints. The good performance of MMV assesses the quality of the image descriptors. CTE compresses the vector sequence by a factor 1024 along the temporal axis and by a factor 128 in the visual axis, which amounts to storing *4 bits per second of video*. The results

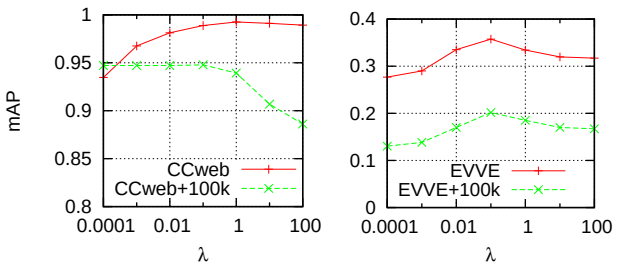

Figure 2. Impact of the parameter λ on the performance

for the large-scale version of the dataset are not strictly comparable with those of the original paper [20] because the distractor videos are different (they do not provide theirs).

On the TRECVID 2008 dataset, our approach significantly outperforms that of Douze & al. [4] in performance, speed and memory usage. MMV cannot be realistically evaluated on this dataset because it can not output boundaries for the matching segments. To compute its NDCR score, we disregard the boundaries, which are normally used to assess the correct localization of the matching segment within a video clip. Despite this advantage, MMV performs poorly (NDCR close to 1), due to the small overlap between queries and database videos (typically 1%), which dilutes the matching segment in the video descriptor.

Remark: The performance of CTE mainly depends on the length of the subsequence shared by the query and retrieved videos: Pairs with subsequences shorter than 5 s are correctly found with 62% accuracy, subsequences between 5s and 10s with 80% accuracy and longer subsequences with 93% accuracy.

Timings. Even for the largest dataset, *i.e.*, CCWEB with 100k distractors, the bottleneck remains the descriptor computation, which is performed faster than real-time on one processor core (1-2 minute per query on TRECVID and CCWEB). Table 2 shows that the search itself takes 23 ms and 75 ms on average for MMV and CTE, respectively, which is orders of magnitude faster than other methods with comparable accuracies.

6.2. Event detection

The evaluation is carried out on the EVVE dataset, see Section 2 for details about the experimental protocol. The parameters are fixed to $p = 64$, $\lambda = 0.1$ and $\beta = 1/16$. On EVVE+100k, this generates a database size of 943 MB and an average query time of 11 s. The detailed results are presented per event in Table 3 for both the temporal and non-temporal versions of our algorithm. Interestingly, MMV performs similarly to CTE on average, at a much lower memory and computational cost, which means that some events are better captured by using a global descriptor of visual appearance. For instance, videos from the Shakira concert always feature the crowd in the foreground and the same concert scene behind, so averaging the frame descriptors provides a robust visual summary of the event.

Event number	EVVE			EVVE+100,000 distractors		
	MMV	CTE	MMV+CTE	MMV	CTE	MMV+CTE
#1	0.531	**0.803**	0.694	0.411	**0.637**	0.566
#2	0.338	**0.413**	0.394	0.195	0.177	**0.229**
#3	0.087	**0.128**	0.111	0.050	**0.069**	0.068
#4	0.455	0.409	**0.486**	0.413	0.335	**0.449**
#5	0.234	**0.262**	0.260	0.148	0.102	**0.164**
#6	0.254	0.257	**0.281**	0.193	0.118	**0.210**
#7	0.199	0.166	**0.202**	0.156	0.086	**0.160**
#8	0.126	0.108	**0.132**	0.056	0.025	**0.058**
#9	0.124	**0.252**	0.212	0.115	**0.174**	0.159
#10	0.366	0.297	**0.371**	**0.158**	0.043	0.157
#11	0.239	0.139	**0.246**	0.174	0.062	**0.174**
#12	0.773	0.714	**0.774**	0.282	0.219	**0.300**
#13	0.604	0.693	**0.719**	0.499	0.569	**0.600**
avg-mAP	0.334	0.352	**0.376**	0.220	0.202	**0.254**

Table 3. EVVE dataset: Retrieval performance (mAP) per event

MMV and CTE are complementary. We therefore combine both methods to capture events that are characterized by exactly repeatable small sequences such as the victory speech of Obama—event #1 (best retrieved with CTE) as well as events that are visually consistent, but not temporally, such as major autumn flood in Thailand in 2011—event #10 (best recognized by MMV). This is done by adding the normalized scores obtained from MMV and CTE for each database video and for each query. This combination achieves a significant improvement in performance (column MMV+CTE in Table 3) and is obtained at no cost, since the computation of MMV is a byproduct of our CTE scheme (*i.e.*, only using $\boldsymbol{f}_0$, see Section 5.1). Note that CTE also outputs the matching video parts, which is important for the video alignment described in the next section.

6.3. Automatic video alignment

For some events from EVVE, many people have filmed the same scene, *e.g.*, for concerts or for re-occuring events. We use the CTE method to automatically align the videos on a common timeline. We match all possible videos pairs (including all query and database videos), which results in a time shift δ^* for all pairs (see Section 4.3).

Aligning the videos consists in estimating the starting time of each video on the common timeline, so that the time shifts are satisfied. Because of mis-matches, edited input videos, etc., the estimation needs to be robust to outliers. We solve the problem by iterating a linear-least squares estimation that identifies the outliers, which are then removed. During this process, groups of independent videos emerge, where each group corresponds to a distinct scene. We use this to display different viewpoints of an event on a shared timeline, as depicted in Figure 3.

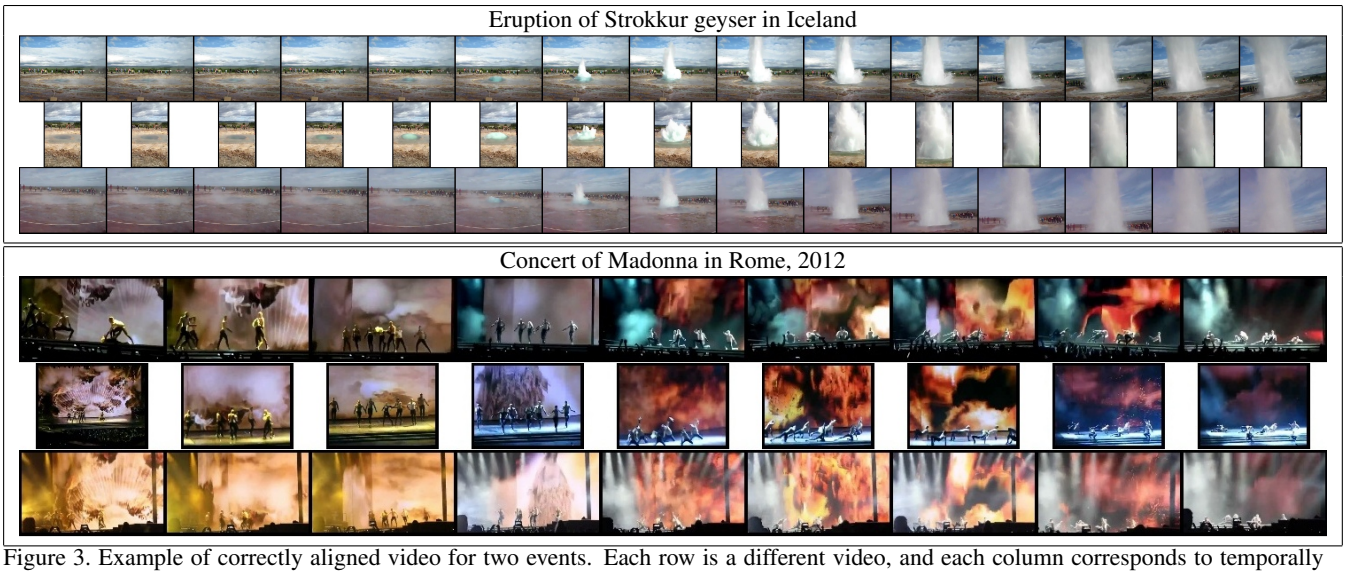

Figure 3. Example of correctly aligned video for two events. Each row is a different video, and each column corresponds to temporally aligned frames from the videos. Note the strong variability across matching videos.

7. Conclusion

This paper proposed a method to jointly encode in a single vector the appearance information at the image level and the temporal sequence of frames. This video representation provides an efficient search scheme that avoids the exhaustive comparison of frames, which is commonly performed when estimating the temporal Hough transform.

Extensive experiments on two video copy detection benchmarks show that our approach improves over the state of the art with respect to accuracy, search time and memory usage. Moving towards the more challenging task of event retrieval, our approach efficiently retrieves instances of events in a large collection of videos, as shown for the EVVE event retrieval dataset introduced in this paper.

Acknowledgments. This work was partially funded by Quaero, (supported by OSEO, French State agency for innovation), and by the European integrated project AXES. We thank Jonathan Delhumeau for helping with the annotation of EVVE.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[2] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[3] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, Dec. 1992.

[4] M. Douze, H. Jégou, C. Schmid, and P. Pérez. Compact video description for copy detection with precise temporal alignment. In *ECCV*, 2010.

[5] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. In *ECCV*, 2012.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[7] M. Jain, R. Benmokhtar, P. Gros, and H. Jégou. Hamming embedding similarity-based image classification. In *ICMR*, 2012.

[8] H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, 2012.

[9] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *Trans. PAMI*, 33(1):117–128, Jan. 2011.

[10] T. Kalker, G. Depovere, J. Haitsma, and M. Maes. A video watermarking system for broadcast monitoring. In *SPIE Conference on Security and watermarking of multimedia contents*, 1999.

[11] A. Karpenko and P. Aarabi. Tiny Videos: A large data set for nonparametric video retrieval and frame classification. *Trans. PAMI*, 33(3):618–630, 2011.

[12] B. V. K. V. Kumar, A. Mahalanobis, and R. D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005.

[13] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *CIVR*, 2007.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[15] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.

[16] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. TRECVID*, 2012.

[17] F. Perronnin, J.Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[19] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and Trecvid. In *MIR*, 2006.

[20] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, 2011.

[21] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM Multimedia*, 2007.

[22] M.-C. Yeh and K.-T. Cheng. Video copy detection by fast sequence matching. In *CIVR*, 2009.

[23] X. Zhang, G. Hua, L. Zhang, and H.-Y. Shum. Interest seam image. In *CVPR*, 2010.