Temporal Aggregation of Visual Features for Large-Scale Image-to-Video Retrieval

Noa Garcia
Aston University
Birmingham, UK
garciadn@aston.ac.uk

ABSTRACT

In this research we study the specific task of image-to-video retrieval, in which static pictures are used to find a specific timestamp or frame within a collection of videos. The inner temporal structure of video data consists of a sequence of highly correlated images or frames, commonly reproduced at rates of 24 to 30 frames per second. To perform large-scale retrieval, it is necessary to reduce the amount of data to be processed by exploiting the redundancy between these highly correlated images. In this work, we explore several techniques to aggregate visual temporal information from video data based on both standard local features and deep learning representations with the focus on the image-to-video retrieval task.

CCS CONCEPTS

• **Information systems** → **Image search**; **Video search**;

KEYWORDS

Image Retrieval, Video Retrieval, Temporal Aggregation

ACM Reference Format:
Noa Garcia. 2018. Temporal Aggregation of Visual Features for Large-Scale Image-to-Video Retrieval. In *ICMR '18: 2018 International Conference on Multimedia Retrieval, June 11–14, 2018, Yokohama, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3206025.3206083

1 INTRODUCTION

Billions of images and videos are generated online every day. With this amount of data publicly available, systems to access and manage multimedia content efficiently are crucial. In this context, visual search and retrieval techniques play an important role in the management of multimedia datasets. Visual multimedia retrieval systems index and find visual content in a collection of images or videos by using a query input. These systems may perform several retrieval tasks, depending on the type of data to be processed.

There exists many different kinds of multimedia retrieval tasks. For example, the well-known image retrieval task, in which a query image is used to find a picture within a collection, has been an active field of research in the computer vision community for a long time [24] and only lately, with the introduction of deep learning techniques [6] its performance has been boosted considerably [11]. Other methods for retrieving images from datasets include text-to-image retrieval [23], in which a detailed text description is provided to find the image of interest, and audio-to-image retrieval [12], in which images are retrieved based on descriptive audio tracks. Regarding video, video-to-video retrieval [18], in which video clips are found by providing a relevant video, is widely used for video copy detection, and image-to-video retrieval [4] aims to find a specific frame within a video collection from a static image.

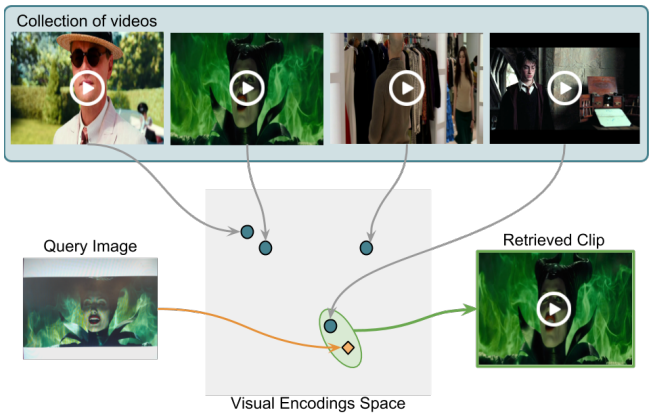

Figure 1: Image-to-video retrieval task. Using visual encodings, videos are indexed in a collection and static images are used as queries to find a specific video clip or timestamp.

In this work we study the specific task of image-to-video retrieval, which has not been explored as much as other multimedia retrieval tasks. Identifying a certain frame from a collection of videos (Figure 1) is a task with many applications, such as video search [2], video content augmentation [10] or video bookmark [9], among others. In contrast to image-to-image retrieval, image-to-video retrieval is an asymmetric task in which dataset items and query images require different processing algorithms. When only considering their visual content (i.e. ignoring audio tracks and metadata), videos are a sequence of consecutive images, which usually are presented at rates between 24 to 30 frames per second to fake the human brain and simulate temporal movement. This temporal structure implies that frames that are close in time are highly correlated to each other. As a consequence, most of the visual information in a video turns out to be redundant or duplicated. Large-scale image-to-video retrieval systems cannot afford to process and index all the visual data available in videos and thus, summarization methods to reduce the amount of data are required. Temporal aggregation of visual

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICMR '18, June 11–14, 2018, Yokohama, Japan
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5046-4/18/06…$15.00
https://doi.org/10.1145/3206025.3206083

features in videos is the technique to summarize redundant visual data in video frames into more compact representations (Figure 2). In this work, we study several temporal aggregation methods to exploit visual redundancy in video data whereas at the same time, meaningful representations for image-to-video retrieval are obtained.

2 RELATED WORK

Early work in image-to-video retrieval [16, 21] was based on applying image retreival techniques to video data. Sivic and Zisserman presented in [21] the well-known bag-of-words (BoW) algorithm by indexing video frames from two movies. Similarly, other work [9, 16] proposed to use BoW with vocabulary trees to index video frames. These methods, however, processed each video frame independently, without considering any temporal correlation between similar frames.

However, to perform large-scale image-to-video retrieval and to exploit temporal redundancy within highly correlated frames, temporal aggregation techniques are needed. Temporal aggregation techniques for image-to-video retrieval can be classified into two different groups: local features-based and global features-based. Local features-based methods [1, 5], extract a set of local features from each frame, typically a few hundred. Each visual feature is tracked along similar frames. Features in the same track are aggregated into a single feature, so the total number of visual features representing each video segment is reduced. For example, in [1] aggregated vectors are obtained by averaging SIFT [15] descriptors within the same track, whereas authors in [5] explore other methods, such as keeping one feature or computing the minimum distance.

On the other hand, global features-based temporal aggregation methods for image-to-video retrieval [4, 25] aim to encode the visual information of a video segment into a single compact vector. Zhu and Satoh [25] aggregate all the SIFT local features in a video clip into a high-dimensional BoW vector. Similarly, Araujo et al. [4] compute compact Fisher Vectors [19] per frame and aggregate them into a single binarized vector per clip.

Previous work in image-to-video retrieval is mostly based on the aggregation of hand-crafted local SIFT features. Considering the outstanding results of deep learning in other retrieval tasks (such as image-to-image [11] or text-to-image retrieval [23]), in this work we explore temporal aggregation methods based on the aggregation of other kind of features, such as local binary features [8] and deep learning visual features [6, 22].

3 PROBLEM FORMULATION

In this section, we formalize the problem of image-to-video retrieval. Considering the inner visual structure of videos, which consist on a set of standard static images known as frames, a shot is defined as a set of consecutive frames that have been captured with the same camera without interruptions.

Let's consider a set of videos $\mathbf{V} = \{\mathbf{V}_i\}_{i \in (0,..,N)}$ of size N, where each video $\mathbf{V}_i$ is at the same time a set of shots $\mathbf{V}_i = \{\mathbf{S}_{i,j}\}_{j \in (0,..,N_{V_i})}$ of size N_{V_i}, and where each shot $\mathbf{S}_{i,j}$ is at the same time a set of frames $\mathbf{S}_{i,j} = \{f_{i,j,k}\}_{k \in (0,..,N_{S_{i,j}})}$ of size $N_{S_{i,j}}$. Note that $f_{i,j,k}$ corresponds to the k-th frame of the j-th shot of the i-th video in the collection. Given a query image q, the goal is to find the most similar frame $\hat{f}$, belonging to the shot $\hat{\mathbf{S}}$, according to a specific metric distance d, such as:

$$\hat{f} = \underset{f_{i,j,k} \in \mathbf{V}}{\arg\min}\, d(\phi(q), \phi(f_{i,j,k})) \tag{1}$$

where $\phi(q)$ and $\phi(f_{i,j,k})$ are the visual representations of q and $f_{i,j,k}$, respectively.

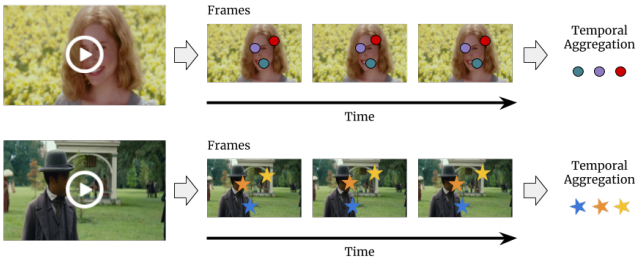

Figure 2: Visual data in video frames is encoded into visual vectors. When frames are very similar, many visual vectors are redundant. By using temporal aggregation methods, compact and meaningful representations are obtained.

For large-scale datasets, performing a search over all frames within the collection $\mathbf{V}$ is prohibitive. To alleviate the search, two techniques are used. Firstly, the amount of visual features $\phi(f_{i,j,k})$ within a shot $\mathbf{S}_{i,j}$ is reduced by using a temporal aggregation method $\Theta(\cdot)$ over each shot:

$$\Theta(\mathbf{S}_{i,j}) = \Theta(\{\phi(f_{i,j,k})\}) \tag{2}$$

Secondly, taking advantage of the inner visual structure of videos, the search is performed in two stages: shot-level search and frame-level search. In the shot-level search, the shot of interest $\hat{\mathbf{S}}$ is found:

$$\hat{\mathbf{S}} = \underset{\mathbf{S}_{i,j} \in \mathbf{V}}{\arg\min}\, d(\phi(q), \Theta(S_{i,j})) \tag{3}$$

Finally, in the frame-level search, $\hat{f}$ is retrieved from the frames contained in $\hat{S}$:

$$\hat{f} = \underset{f_{i,j,k} \in \hat{\mathbf{S}}}{\arg\min}\, d(\phi(q), \phi(f_{i,j,k})) \tag{4}$$

In the next section, we present two different aggregation methods $\Theta(\cdot)$ to perform image-to-video retrieval for large-scale datasets.

4 OUR APPROACHES

We propose two different models to aggregate temporal redundant visual information from videos:

- Local Binary Temporal Tracking (LBTT)
- Deep Features Temporal Aggregation (DFTA)

4.1 Local Binary Temporal Tracking

LBTT method, which is detailed in [10], is based on the summarization of hand-crafted local binary features.

4.1.1 Image Representation. In LBTT, images are represented by a set of BRIEF [8] features, which encode the pixel intensity value of small image regions into 256-dimensional binary vectors.

4.1.2 Temporal Encoding. Binary features are tracked along time by matching descriptors in consecutive frames. Two descriptors are a match when the Hamming distance between them is below a threshold. Matches that are not spatially close in the pixel space are filtered. The tracking is performed in a bidirectional way so features within a track are unique (i.e. each feature can only be matched with up to two features: one in the previous frame and one in the following frame). To avoid adding noisy features, only stable tracks in time are considered.

The motivation of using binary features as the local descriptor is two-fold. Firstly, Hamming distance for matching binary features is faster to compute than Euclidean distance for matching SIFT vectors. Secondly, we find that binary features are more stable over time than SIFT (Figure 3) and hence, easier to aggregate. We define a *key feature* as the aggregation of all the features in the same track into a single vector. For each track, its key feature is computed by using majorities.

4.1.3 Shot Boundary Detection. Consecutive frames that share visual similarities are grouped into shots. The boundaries of different shots are detected when two consecutive frames have no common tracks. Subsequently, each shot is then represented by a set of key features, similarly to how frames are represented by a set of features.

4.1.4 Retrieval. At query time, BRIEF features are extracted from the query image. To find the most similar shot to the query image, for each query feature a set of nearest neighbour (NN) key features is obtained. Key features are indexed in a kd-tree so that the NN search can be performed faster. Each key feature votes for the shot it belongs to. The set of frames contained in the most voted shot are compared against the input image by brute force, i.e. distances between descriptors in the query image and descriptors in the candidate frames are computed. The frame with minimum distance is retrieved.

4.2 Deep Features Temporal Aggregation

DLTA method is based on the temporal aggregation of deep learning visual features.

4.2.1 Image Representation. In DLTA, the visual content of each frame is encoded in a RMAC [22] image vector. RMAC is a deep global image representation obtained from the last convolutional layer of a convolutional neural network (CNN). When an image is fed into the CNN, the response of each filter of the last convolutional layer is represented by a feature map. RMAC computes local features by max-pooling the activations of different regions in the feature map. Each of these local vectors is independently post-processed with $\ell 2$-normalization, PCA-whitening and $\ell 2$-normalization. Post-processed vectors are summed together and $\ell 2$-normalized one last time to obtain a single vector per image.

4.2.2 Temporal Encoding. By encoding each video segment into a single visual vector, we aim to capture as much visual information as possible whereas at the same time we reduce the amount of redundant data. So far, we study two different approaches to aggregate global RMAC vectors into a video shot representation:

- DLTA-Max: for each dimension, the shot encoding is obtained by computing the frame RMAC maximum value.

$$\Theta(\mathbf{S}_{i,j}) = maxpool(\phi(f_{i,j,k})) \tag{5}$$

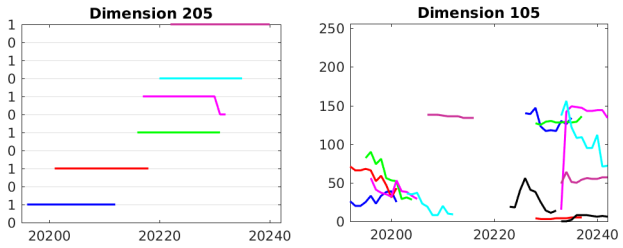

Figure 3: Trajectories of sample tracks along a sequence of frames. Left: Binary features. Right: SIFT features. Binary features are more constant over time than SIFT features.

- DLTA-Mean: the shot encoding is computed as the average of the RMAC features within the shot.

$$\Theta(\mathbf{S}_{i,j}) = \frac{1}{N_{S_{i,j}}} \sum_{k=1}^{N_{S_{i,j}}} \phi(f_{i,j,k}) \tag{6}$$

Shot boundaries are detected using the same algorithm as in §4.1.3.

4.2.3 Retrieval. For each query image, its RMAC vector is obtained. Query vector is matched against the aggregated shot encodings using cosine similarity. The shot encoding with minimum distance to the query is retrieved. Then, frame RMACs within the retrieved shot are matched against the query vector. The most similar frame according to its cosine similarity is retrieved.

5 EXPERIMENTS

5.1 Experimental Details

Dataset. We have created the MoviesDB dataset [10] to evaluate different image-to-video methods. This dataset is a collection of 40 movies with more than 80 hours of video and up to 7 million frames. Query images for retrieval are captured by a webcam while movies are being played on a screen in front of it. The frame number of each captured image is saved in a text file as a ground truth. Movies and queries have different resolutions and aspect ratios. For evaluation, as long as the retrieved frame shares strong similarities with the ground truth frame by matching SURF [7] features, it is consider a *Visual Match*. Performance is measured in terms of accuracy:

$$\text{Acc} = \frac{\text{No. Visual Matches}}{\text{Total No. Queries}} \tag{7}$$

LBTT. All the frames and images are scaled down to 720 pixels in width. When tracking binary features, only matches with a Hamming distance less than 20 and a spatial distance less than 100 pixels are considered. For computing key features, only stable tracks longer than 7 frames are used.

DLTA. Frames are resized to 1024 pixels width. Query images are resized to 960 pixels width. RMAC features are extracted using the end-to-end architecture proposed in [11] with pretrained VGG16 weights [20]. Frames are subsampled and processed at 5 fps. PCA whitening is computed using the movie *American Beauty*, which is not in the MoviesDB dataset.

Table 1: Results in *The Devil Wears Prada* from MovieDB.

	Method	Dim	Memory	N.Features	Acc
Local	IR-BRIEF	256	2.53 GB	85M	0.93
	LBTT	256	61 MB	2M	0.93
Global	IR-FC1	4096	614 MB	39,324	0.63
	IR-FC2	4096	614 MB	39,324	0.42
	IR-RMAC	512	76.8 MB	39,324	0.91
	DLTA-Max	512	3.13 MB	1,602	0.22
	DLTA-Mean	512	3.13 MB	1,602	0.69

5.2 Experimental Results

We evaluate our temporal aggregation methods in a small subset of the MoviesDB dataset. Table 1 shows the results for *The Devil Wears Prada* movie. Our aggregation methods LBTT and DLTA are compared to their corresponding image retrieval versions, IR-BRIEF and IR-RMAC, respectively (i.e. no temporal aggregation is used). IR-FC1 and IR-FC2 are image-retrieval algorithms on top of features from VGG16 network's first and second fully connected layer, respectively, which obtain significantly worst accuracy than IR-RMAC by using 8 times more memory. This indicates that features from fully connected layers may not be good enough for image-to-video retrieval, as suggested by other authors [4]. With respect to our temporal aggregation methods, LBTT achieves comparable accuracy to both IR-BRIEF and IR-RMAC by using significantly less memory. With only 3.13MB of memory, DLTA-Mean performance is slightly worst then IR-RMAC. However, DLTA-Max is clearly not capturing the visual video data according to the needs of the image-to-video retrieval task. Our temporal aggregation methods, are further compared in fours different movies in Table 1, where again, DLTA methods are not able to capture the visual video data as well as LBTT. This may suggest that either representing each shot into a single vector is not enough for image-to-video retrieval or that more complex DLTA methods are needed.

6 CONCLUSIONS AND FUTURE WORK

This research proposes temporal aggregation models of visual features for image-to-video retrieval, in which static pictures are used to find a specific frame in a video a collection. The models presented in this work are based on the aggregation of local binary features and deep learning features. The experiments conducted so far show that methods based on binary features outperform deep learning methods. As a future work, we aim to explore more complex methods to aggregate deep learning features for image-to-video retrieval task. According to other work [4], which successfully aggregates shots into single vectors using SIFT features, we believe that there is still room for improvement in the aggregation of deep learning features. So far, DLTA methods are based on the aggregation of deep features from pre-trained CNN architectures. Re-training and fine-tunning these architectures for the specific retrieval task should lead to better results [11]. As part of our work in progress, we are studying DLTA based in LSTMs [13]. However, it is still an open question whether LSTMs are able to properly encode data for image-to-video retrieval. Other methods to be considered in the future are 3D CNN [14] and temporal autoencoders [17]. We also aim to conduct experiments in the full MoviesDB dataset as well as in other image-to-video retrieval collections (SI2V [2] and VB [3]).

Table 2: Accuracy in The Devil Wears Prada, Groundhog Day, Her and Pirates of the Caribbean movies from MoviesDB.

Method	Movie1	Movie2	Movie3	Movie4
LBTT	0.93	0.97	0.76	0.80
DLTA-Max	0.22	0.16	0.18	0.12
DLTA-Mean	0.69	0.56	0.53	0.47

REFERENCES

[1] Arasanathan Anjulan and Nishan Canagarajah. 2007. Object based video retrieval with local region tracking. *Signal Processing: Image Communication* 22, 7 (2007).

[2] André Araujo, Jason Chaves, David Chen, Roland Angst, and Bernd Girod. 2015. Stanford I2V: a news video dataset for query-by-image experiments. In *ACM Multimedia Systems*.

[3] André Araujo, Jason Chaves, Haricharan Lakshman, Roland Angst, and Bernd Girod. 2016. Large-scale query-by-image video retrieval using bloom filters. *arXiv preprint arXiv:1604.07939* (2016).

[4] Andre Araujo and Bernd Girod. 2017. Large-Scale Video Retrieval Using Image Queries. *IEEE Transactions on Circuits and Systems for Video Technology* (2017).

[5] A Araujo, Mina Makar, Vijay Chandrasekhar, D Chen, S Tsai, Huizhong Chen, Roland Angst, and Bernd Girod. 2014. Efficient video search using image queries. In *ICIP*.

[6] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural codes for image retrieval. In *ECCV*.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF : Speeded Up Robust Features. *ECCV* (2006), 404–417.

[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. Brief: Binary robust independent elementary features. *ECCV* (2010).

[9] David Chen, Ngai-Man Cheung, Sam Tsai, Vijay Chandrasekhar, Gabriel Takacs, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod. 2010. Dynamic selection of a feature-rich query frame for mobile video retrieval. In *ICIP*.

[10] Noa Garcia and George Vogiatzis. 2017. Dress Like a Star: Retrieving Fashion Products From Videos. In *ICCV Workshops*.

[11] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2017. End-to-end learning of deep visual representations for image retrieval. *IJCV* 124, 2 (2017).

[12] David Harwath, Antonio Torralba, and James Glass. 2016. Unsupervised learning of spoken language with visual context. In *NIPS*.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2013), 221–231.

[15] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60, 2 (2004).

[16] David Nistér and Henrik Stewénius. 2006. Scalable recognition with a vocabulary tree. *CVPR* (2006).

[17] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. 2015. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309* (2015).

[18] Sébastien Poullot, Shunsuke Tsukatani, Anh Phuong Nguyen, Hervé Jégou, and Shin'Ichi Satoh. 2015. Temporal matching kernel with explicit feature maps. In *ACM Multimedia*.

[19] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. 2013. Image classification with the fisher vector: Theory and practice. *IJCV* 105, 3 (2013).

[20] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* (2014).

[21] Josef Sivic and Andrew Zisserman. 2003. Video google: A text retrieval approach to object matching in videos. In *ICCV*.

[22] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*.

[23] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*.

[24] Liang Zheng, Yi Yang, and Qi Tian. 2017. SIFT meets CNN: A decade survey of instance retrieval. *TPAMI* (2017).

[25] Cai-Zhi Zhu and Shin'ichi Satoh. 2012. Large vocabulary quantization for searching instances from videos. In *ACM Multimedia Retrieval*.