LxTube - Processing of massive video archives in order to index and search information

Nuno Miguel Rosa
Instituto Superior Técnico, Lisboa, Portugal
Email: nuno.rosa94@gmail.com

***Abstract*—The rapid growth of video databases available online demands more efficient and robust search engines and indexing systems. The key challenges when developing such systems are: 1) Identification of the best features to represent the video; 2) Identification of the best way to index these features. For the former it is crucial that we come to a compromise between a perfect description of the video and the memory allocation and the time costs involved in the extraction process of the features. The algorithm adopted for the second challenge defines how the data will be organized in the database and, consequently, the speed of retrieval and the quality of the results. In this work I will present an approach to this type of problem, where four different types of features are used (Color, Texture, Semantic Information and Motion) and a hierarchical k-means clustering algorithm is applied to index the database information. This indexing step helps to identify the database's most similar frames to each of the frames of the query video in order to produce a graph used in the similarity evaluation process between the video of query and the videos of the database. The graph construction process can be adapted to produce different outputs according to the intentions of the user, which illustrates one of the great advantages of this algorithm, its enormous versatility. We also tested the scalability of the algorithm in a database with 43000 videos extracted from the Youtube-8M dataset.**

I. Introduction

According to 2016 Youtube's statistics almost 5 billion videos were watched everyday. In 2014 Cisco concluded that 64 % of all Internet traffic belonged to any form of video related activity and estimates that in 2020 this value may increase to 82 %. These studies show the impact video has on society nowadays with significant repercussions on people's habits. In 1951 the first live images were recorded and, since then, we watched an enormous transformation on society evidenced by the widespread access to the new technologies like smartphones which in current times represent the preferred method for the consumption and production of this type of content.The economy and marketing sectors also changed with this audiovisual revolution as it can be confirmed by the 2015 study by the Web Video Marketing Council where 96 % of the 350 businessman interviewed said that video is a crucial part of their sales and marketing strategies.

The main issues related to any search system (video or other type of information) can be segmented in three groups: (1) Speed of the search; (2) Precision of the results; (3) Subjectivity of the search. The first two issues are in conflict with each other many times because the results provided to the user in a split of second may not contain precise results and the time needed to achieve them may not suit the user needs. The third one appears because the criteria used on the search may come from a multitude of origins. Different users could want to see: (1) a person or event in particular; (2) images or shots similar to some other videos; (3) videos from a specific location or period of time; to give some examples.

This work intends to answer the question: "Given a database DB and a video A which videos of DB are more similar to A?" in order to produce a video search engine based on visual and semantic features that could work as a complement to the current video search systems that use only a set of labels more or less complex to characterize a video. The aforementioned subjectivity problem which will be tackled with a small study on the relative importance of the features used to describe the frames of the videos and ultimately to compare two audiovisual elements.

This paper is organized as follows. On section II is presented the related work and the main contributions from other authors to each one of the stages of the video search process. On section III are presented the methods and techniques developed. On section IV the main results are presented and analyzed in terms of efficiency and precision. Finally on section V the main conclusions from this work are presented as the objectives for future work.

II. Related work

A. Video Representation

The detection and retrieval of similar videos is intrinsically connected with the features chosen to represent them. The objective is to use more than one descriptor to achieve the most complete video representation so that the subjectivity problem associated with video search may be diminished. The pictorial information extracted from the videos should respect two criteria: (1) Accurate video description; (2) Fast extraction process.

Note that a feature with a fast extraction process but which does not describe correctly the video is ultimately useless and in the other extreme a perfect descriptor of the video frames which takes a lot of time to process is not desirable. These features are extracted for a collection of representative images of the video called key-frames in order to speed the pre-processing stage of the search engine. In the literature a multitude of features are used to characterize the key-frames such as color descriptors, texture features, motion descriptors and semantic content among others.

Color descriptors are the most used feature in image processing thanks to their ability to condense the hue of the images. An image is in fact a 3D matrix with dimensions $h \times w \times n$ where h represents the number of lines of the matrix (height of the image),w represents the number of columns of the matrxi (width of the image) and n represents the number of color channels used by the color model chosen to represent the image (RGB - 3 color channels; Gray-scale - 1 color channel).

The state of the art shows the application of several color models such as: (1) RGB [1], [2]; (2) HSV [3], [4]; (3) L*a*b [5]. The way the color information is organized is also dependent on the type of application the color feature is used in: (1) Histograms [6], [7], [2]; (2) Block Histograms [3]; (3) Joint Histograms [8]; (4) Correlogram [1]. The last three options try to solve the lost of spatial information associated with the histogram technique.

Texture in image processing corresponds to the features which describe the spatial arrangement of color or intensity of an image in certain regions. Examples of this kind of features are: (1) edge-detection features [9], [10]; (2) co-occurrence matrix [11].

Motion features are able to describe the video dynamic and identify the movement of the camera and/or the regions or points of interest and track them. Examples of these descriptors are: (1) Optical Flow [12]; (2) MPEG Motion Vectors [13], [14]; (3) Shot FFT [15].

Video search is done currently with a set of semantic concepts describing the video content. In recent years the search for a fast and automatic identification and location of these semantic concepts has increased and has already become a powerful tool to describe an image or video. These type of concepts can be retrieved using: (1) SIFT descriptors [16], [17], [6]; (2) SURF descriptors [6], [18]; (3) HOG descriptors [19]; (4) Neural Networks [20], [21].

B. Video Segmentation

Video segmentation is the process of reducing the video into smaller fragments which can be stationary images (key-frames) or a set of sequential images (video-skims). In the first case a collection of representative image is chosen whereas in the second one the video is divided into small clips separated by some kind of video effect. The method used in this work is based on the first option, like in [4] where the authors propose a video segmentation algorithm based on unsupervised clustering using a 64 bin HSV histogram as feature. [6] includes also a pre-segmentation of the video into shots is done based on 16-dimensional HSV histograms and an unsupervised clustering algorithm retrieves the key-frames from each shot, are selected as are presented innovative video segmentation algorithms. In [13] a user attention model is developed using motion *features* extracted from MPEG metadata. According to this model is frame is associated to a value and the highest value frames are selected as representative. The problem of gradual transition is tackled in [22] with Optical Flow features while [3] uses HSV histogram with 64 bins to detect and track objects defining a key-frame when one of these objects appear and disappear from the screen.

C. Video Indexing

In the case of video search a good indexing algorithm allows us to go through the database video key-frames and find the best matches to the query data without having to compute all similarities. Examples of indexing algorithms are: (1) Hashing Tables [23], [24] [25]; (2) Decision Trees [26], [5], [2]; (3) Graphs [27], [28], [29].

In [23] and [24] the indexing algorithm is based on Discrete Cosine Transforms. Each video is normalized spatially and temporally and then the DCT coefficients are extracted, filtered and finally binarized according to the median value while in [25] a compact descriptor of the video is created using the luminosity block histograms and consecutive difference between frames.

A decision tree is an indexing structure organized like a flow chart where each node represents a test and each branch the possible results. In the case of videos each node represents a video descriptor and each branch is established between similar nodes. A query descriptor (retrieved from a key-frame of a query video) is then compared sequentially throughout the tree against the nodes. In [2] the concept of Video Triplets is introduced as an indexing algorithm (based on the B+trees). These entities are obtained from a clustering algorithm that uses HSV histograms with 64 bins as the key-frames representation. A video is therefore represented by a collection of Video Triplets that can be used in the comparison with another videos.

A graph is an indexing algorithm very similar to the decision trees in terms of structure. Each node represents a frame descriptor and a branch specifies some form of relationship between them. In these type of structure the direction of the connections may be taken into account. Typically we try to find the smallest path between the start node and end node in a significantly optimized process. In [28] an iterative graph construction algorithm is proposed. Starting from a query image and a small set of correspondences an initial graph is constructed and adjusted according to the similarities of the initial set.

D. Video Search

Video similarity is a somewhat a vague concept so authors tested throughout the years multiple features to describe the videos. The algorithms proposed tend to follow the same concept: (1) A pre-processing step where the video is segmented into key-frames; (2) Retrieval of pictorial and/or semantic data from these images; (3) Data Indexing; (4) Video retrieval. In [7] features of motion and color and conjugated to produce a metric to assess the video similarity while in [30] the authors propose an algorithm for full video similarity search and similar segment retrieval using a decision tree and a graph in the search step respectively. In [26] the key-frames are clustered and assigned to the cluster ID used to develop a code for the video. The comparison between two videos compare

the sequences produced using a KNN algorithm. An algorithm for video comparison based on pixel analysis is proposed on [31] while [32] shows an improvement to a image retrieval system in order to include video search. Finally [33] shows a video search engine with negative feedback that improves the results and [34] proposes an algorithm for video search using object detection.

Another option on video search is based on the classification of video content in order to extract semantic concepts to be used in the video comparison process [5], [35] and [27].

In [5] the concept of Semantic Texton Forest are introduced for video classification using the notion of Random Trees. A color histogram in the model L*a*b* is used to describe the key-frames and each pixel and its neighborhood is used as input for the Semantic Texton Forest. The descriptor used in the classification step is obtained from a bag of words model trained with the outputs of the semantic texton forests (collection of leaf nodes) for all pixels of the images on the database. The semantic information present in MPEG metadata is expanded in [35] using the concepts of belonging, particularity or relative position in order to increase the precision in video search by semantic concepts. Finally in [27] the search by semantic concepts is adapted to the autonomous driving environment where a classifier is trained to detect objects and identify their relative position and motion and a graph search is done to match the data retrieved from the video with the textual query.

Video search may also consist on the temporal alignment of videos which corresponds to the matching between sequences of frames from two videos. It has several real life applications such as copy detection, copyright infringement or advertisement management. This problem surpasses the video similarity because in addition to that we have to consider the temporal information present on the videos. Works like [17], [36] and [37]. In [17] and [36] the concept of Circular temporal Encoding is presented as a way to video retrieval and temporal alignment using SIFT features and Fast Fourier Transform to speed-up the alignment process. In [37] the authors used a neural network to process video shots and retrieve the orientation of the object used in a sampling algorithm to estimate the matches between the two videos.

III. Implementation

A. Algorithm

The algorithm developed in this work has a general structure similar to the ones listed before however it presents an innovative and flexible search and indexing method based on graph search. The developed search engine is modular so each stages is independent of the remaining ones which means that the graph construction stage, for example, depends only on the desired application ignoring the type of features extracted beforehand.

Two types of segmentation were tested: (1) Video Sampling (VS); (2) MPEG metadata analysis (MPEG). In the first one a fixed time step τ is defined and a key-frame is selected at every interval (see figure 1), whereas in the second case the information about the location of key-frames is included already in the metadata and can be accessed directly. In the later the user can choose the location of key-frames or a consecutive frame difference process could be applied to identify the most representative images of the video (see figure 2).

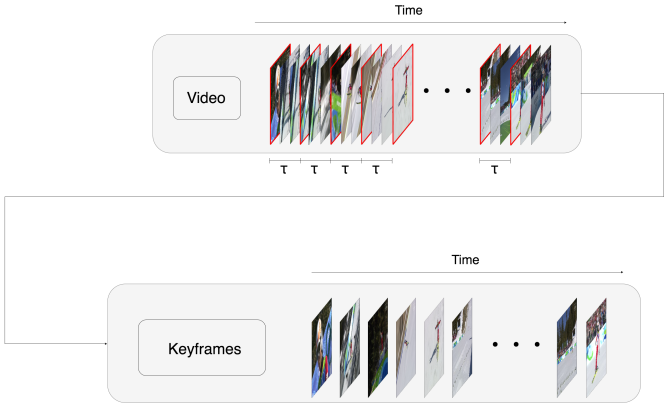

Fig. 1. Example of the application of the sampling algorithm in video segmentation

The features used in this work are: (1) a HSV color histogram with 256 bins (16 for the Hue channel, 4 for the Saturation channel and 4 for the Value channel) (C); (2) an 1000 bin histogram collected from a bag of words model constructed from SIFT descriptors extracted on a dense grid (dense-SIFT) (T); (3) a 2000 bin histogram of movement retrieved from a 3D Fourier transform applied in a shot and the normalization of each cube obtained from the segmentation of the output matrix in a grid with $20 \times 20 \times 5$ elements (M); (4) an 1000 bin histogram containing probabilities of occurrence of real-life objects in an image collected from the output of a pre-trained neural network (SC) ([38]).

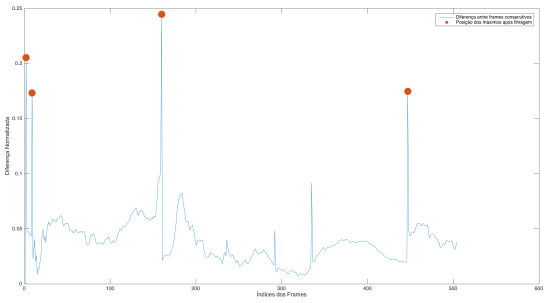

Fig. 2. Key-frame detection algorithm based on the difference between consecutive key-frames

After the segmentation process these features are independently extracted for each key-frame of the videos and indexed in a decision tree like structure obtained through a hierarchical clustering algorithm where all the descriptors collected for the key-frames of the database are organized in ”n” groups, which are then divided in another ”n” groups until each descriptor

is isolated (leaf-node) (see figure 3). The key-frame match search algorithm is based on a KNN methodology where the descriptor for a query key-frame is compared sequentially along the decision tree in order to retrieve the "K" best matches

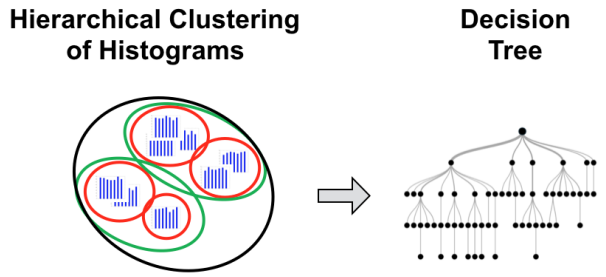

Fig. 3. Example of the adopted indexing algorithm.

Finally there were tested two video search algorithms: (1) weighted sum of key-frame matches (WS); (2) Graph Search Algorithm (GSA). In the first one the distances between a query key-frame and its neighbors are converted into similarity values through equation 1 and then all the matches belonging to a certain video are multiplied by a weight factor according to the feature used and summed according to the equation 2. In the end we will have a similarity value between each database video and the query video. The videos with the higher values are chosen as the matches.

$$s_{ik} = e^{\frac{d_{ik}}{\sigma}} \tag{1}$$

$$D_{v_q v_{db}} = \sum_{i=1}^{N} \sum_{k \in S} \alpha \cdot s_{ik}^{c} + \beta \cdot s_{ik}^{w} + \gamma \cdot s_{ik}^{s} + \theta \cdot s_{ik}^{m} \tag{2}$$

Note that σ in equation 1 and α, β, γ and θ in equation 2 (whose sum must be always one) are adjustable parameters for each feature. In the latter $D_{v_q v_{db}}$ is the distance between the query video and a video of the database; N corresponds to the number of key-frames of the query video; S is the set of neighbors of the query key-frame belonging to the database video under consideration; s_{ik}^{x} is the similarity values for each feature.

This method allows us to compare two videos but does not take into account that: (1) a video is a ordered sequence of frames; (2) a small section of a video very similar to other could be enough to be considered similar; (3) a query frame must have only one valid match per database video. With the previous version a database video with all the neighbors for a small set of query key-frames could be considered wrongfully the best match impairing the results for other possible better correspondences. The sum was also done for every possible neighbor ignoring the temporal sequence of a video.

In order to work around this issues it is proposed an innovative and flexible algorithm of video search based on graphs. In the base model of construction N+2 artificial nodes are created corresponding to the "StartNode" from which every search begins, the "EndNode" where every search finishes and N V_x nodes, one for each video in the database with key-frames as neighbors of the query key-frames. Every node V_x connects with "StartNode" with weight 0 and to each neighbor belonging to it (NK_x) with a weight given by the equation 2 multiplied by a time parameter (see equation 3) that takes into account the temporal distance between neighbors evaluated through the query key-frame index, in other words, the connection between two neighbors of consecutive query key-frames will be privileged when compared to a connection between a neighbor of the first and last query with the same base weight.

$$w_t = \frac{N + 1 - (K_i - K_j)}{N} \tag{3}$$

In equation 3 w_t corresponds to the time parameter, while N is the number of key-frames and K_i and K_j are two different key-frame indexes.

Each NK_x will connect also with the neighbors from following query *key-frames* using the same time parameter as before. Finally every node in the graph will connect with the "EndNode" with a weight of 0. An example of this type of construction can be seen in figure 4.

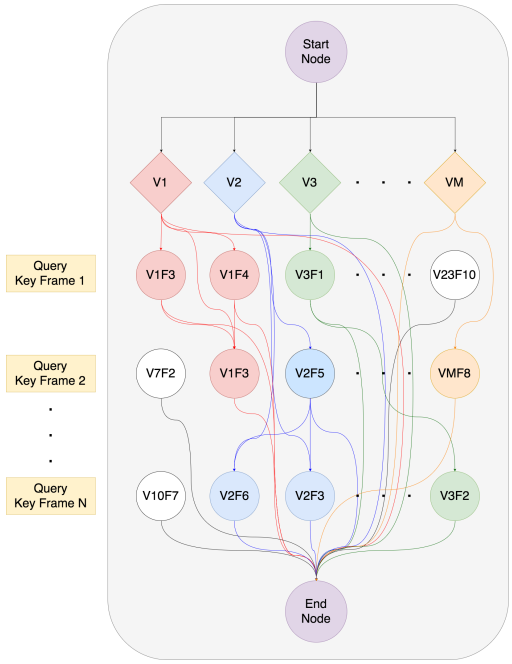

Fig. 4. Example of a graph and the connections established between the nodes for the base construction model

Since we are using similarity values with non negative values in which a perfect match corresponds to one and a complete opposite is 0 the search procedure applied in this kind of graph will retrieve the longest path. Given this conditions is easy to conclude that the best result will have to go through the maximum number of nodes avoiding the introduction of so many connections. A new methodology

achieves the same results connecting only the neighbors of a query key-frame with the neighbors of the following key-frame with matches from the same video. The same graph example with the new methodology can be seen in figure 5.

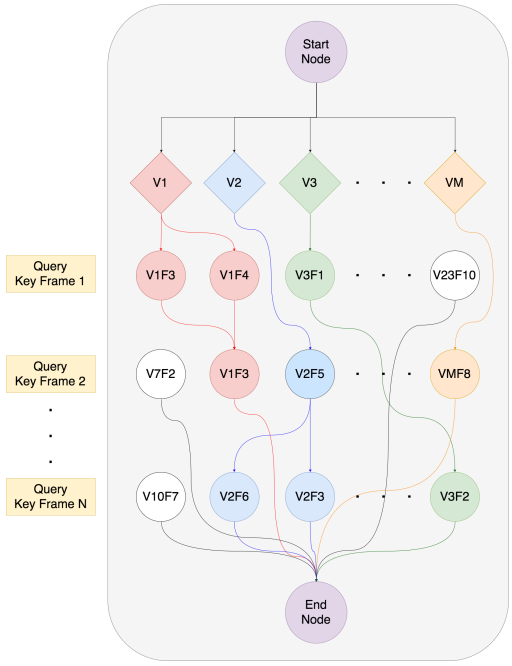

Fig. 5. Example of a graph constructed with the restriction to connect only to the next available neighbor

Finally two other graph construction methodologies were tested to show the flexibility of this algorithm, (1) A mixing between video nodes is introduced in order to construct a compilation of the query video using the best segments of the database; (2) Connect only sequential neighbors from sequential key-frames to detect very similar video segments.

The graph search algorithm applied for the first three graph construction methodologies was the Bellman-Ford algorithm. In order to get the longest path from the "StartNode" to the "EndNode" all connections are converted to negative values and we look for the path with the smallest value. For each video an independent graph is constructed and the search results are retrieved and sorted in order to collect the best matches. In the segment search methodology a recursive algorithm is developed to identify the sequences with the correct time interval.

In terms of complexity this algorithm will be O(FNMDKI log(NM)/log(K)) for the indexing stage (according to the authors [39]), where N corresponds to average number of key-frames of a database video, M is the number of videos on the database, F the number of features used, D size of each descriptor, K is the branching factor and I corresponds to the maximum number of iterations allowed. The key-frame similarity search will be O(FLD log(NM)/log(K)) where L is the maximum number of points checked. Considering we have Q nearest neighbors per query key-frame and P query images we will have a complexity of O(FQP) for the first search algorithm and O((FQP)^2/M) in the graph construction algorithm and O(MVE) on the video search algorithm, where V and E are the number of vertices and nodes on the graph.

B. Tests

The tests developed in this work evaluated the precision of a collection of algorithms in order to identify the best and for that one the scalability was tested using the dataset YT-8M with 8 million pre-processed videos. In addition to these we also included a small study about the preferences of the users when using a video search engine that uses the 4 descriptors listed before, in other words, which feature has more influence in the search process in order to provide the results the users want.

For the first one we consider all the combinations of segmentation options (2), features (4 + 1) and video search algorithm (2) to create 20 video search engines (see table III). In this test we used a collection of 132 videos of which 35 were divided in 6 classes according to their content. The remaining were considered as a control group. Each engine received as input one of the videos of these classes and provided at the output the 5 best matches from the 132 videos, used in the evaluation of precision and recall. A match was considered good if it belonged to the same class. In the scalability analysis, the best algorithm from the previous test is used to evaluate a collection of 42576 videos. The algorithm used the features provided in the dataset (video-level features) to reduce the number of videos to use in the search procedure (frame-level features). The labels associated with each video are used to evaluate the precision of the engine. Finally the study on the labels was divided in two parts, one in which 60 combinations of weights were pre-established and the users would have to check if these were adequate or not and a second one were we tested a machine learning where the weights were progressively adapted with feedback from the user.

IV. Results

A. Video search engine precision and recall evaluation

From the 132 videos used in this test 35 were divided into 6 classes the following way: (1) 7 videos of Ski Jumping - SKJ; (2) 7 videos of Ski Slalom - SKL; (3) 6 videos of cycling - CYC; (4) - 6 videos of 100 meters athletics - ATL; (5) 5 videos of football highlights - FBL; (6) 4 videos of canoeing. The remaining videos were chosen randomly as a control group. All the results presented were extracted using a late 2013 MacBook Pro, with an Intel i5 with 2.4 GHz processor and a graphics card Intel Iris 1536 MB.

The first step of each engine is the segmentation process whose results in terms of number of key-frames (KF) and processing time are presented in the table II for all the classes. In the sampling approach the parameter τ used was 1 frame per second.

The video sampling segmentation algorithm consistently provided larger number of key-frames but this factor by itself cannot be considered in the evaluation of the quality of the search engine. In fact the second method checks the difference

VS + C + WS	E_1
VS + T + WS	E_2
VS + SC + WS	E_3
VS + M + WS	E_4
VS + C + T + SC + M + WS	E_5
VS + C + GSA	E_6
VS + T + GSA	E_7
VS + SC + GSA	E_8
VS + M + GSA	E_9
VS + C + T + SC + M + GSA	E_{10}
MPEG + C + WS	E_{11}
MPEG + T + WS	E_{12}
MPEG + SC + WS	E_{13}
MPEG + M + WS	E_{14}
MPEG + C + T + SC + M + WS	A_{15}
MPEG + C + GSA	E_{16}
MPEG + T + GSA	E_{17}
MPEG + SC + GSA	E_{18}
MPEG + M + GSA	E_{19}
MPEG + C + T + SC + M + GSA	E_{20}

TABLE I
LIST OF TESTED ALGORITHMS

	VS		MPEG	
	# *KF*	Δ_t (s)	# *KF*	Δ_t (s)
ATL	226	31,45	49	32,04
SKL	109	11,44	40	11,08
FBL	147	19,05	49	18,35
SKJ	19	2,39	6	2,51
CAN	210	28,81	89	26,94
CYC	76	10,32	22	10,26

TABLE II
RESULTS OBTAINED FOR THE TWO SEGMENTATION TECHNIQUES

between consecutive frames, so if the video has no significant changes or cuts no new key-frames are identified.

Then the 4 features described earlier are extracted for each key-frame. In the tables III and IV is possible to see the average time of each feature per key-frame (FP), the average time of search for similarity for all key-frames (ST) and the size of the descriptor (DS)

	C	T	SC	M
FP (s/KF)	0,009	0,700	0,141	12,925
ST (s)	0,294	1,427	1,614	3,654
DS (bins)	256	1000	1000	2000

TABLE III
RESULTS OF THE EXTRACTION AND SEARCH PROCESSES FOR THE 4 FEATURES USED IN THE ENGINE WHEN THE SEGMENTATION ALGORITHM IS THE VIDEO SAMPLING

In both tables III and IV the motion descriptor takes too much time to process which makes it impractical to use in an online search engine where the user provides its own video. Resizing the images of a shot before the fourier transform or the using another good motion descriptor are some of the options available to improve this feature. On the other hand the elapsed time in the search step is relatively quick for all

	C	T	SC	M
FP (s/KF)	0,017	0,835	0,326	19,562
ST (s)	0,360	0,708	1,016	2,464
DS (bins)	256	1000	1000	2000

TABLE IV
RESULTS OF THE EXTRACTION AND SEARCH PROCESSES FOR THEE 4 FEATURES USED IN THE ENGINE WHEN THE SEGMENTATION ALGORITHM IS THE ANALYSIS OF FFMPEG METADATA

features, validating the indexing algorithm used.

The tests were developed for each video in the classes and the process starts by selecting a query video, remove it from the database, segment it and extract the features. The decision tree is constructed with the remaining videos on the database and is used to get the nearest neighbors for each query key-frame. Finally the 5 most similar videos are extracted and the corresponding precision and recall results are collected. The type of search algorithm used in the engine changes the elapsed time on this stage of the process as can be noted in tables Vand VI for the weighted sum and the graph search algorithm respectively. In the latter both the graph construction (GC) and graph search (GS) times are presented

Classes	MPEG	VS
	Search (s)	
ATL	0,015	0,067
SKL	0,016	0,031
FBL	0,020	0,044
SKJ	0,005	0,015
CAN	0,031	0,058
CYC	0,014	0,028

TABLE V
AVERAGE ELAPSED TIME IN THE SEARCH STAGE OF THE ALGORITHM WHEN CONSIDERING THE WEIGHTED SUM VIDEO SEARCH ALGORITHM

Classes	MPEG		VS	
	GC (s)	GS (s)	GC (s)	GS (s)
ATL	0,45	0,16	5,23	1,40
SKL	0,41	0,13	1,78	0,42
FBL	0,48	0,19	3,39	0,78
SKJ	0,28	0,02	0,34	0,05
CAN	1,67	0,62	5,27	2,04
CYC	0,43	0,06	0,88	0,28

TABLE VI
AVERAGE ELAPSED TIME PER CLASS IN THE GRAPH CONSTRUCTION AND GRAPH SEARCH STAGES OF THE ALGORITHM

The time expended in both cases is relatively small and perfectly adjusted for a video search application. Given the 5 best matches for every video the precision and recall of the engine was evaluated according to: (1) type of segmentation algorithm; (2) type of feature; (3) type of search algorithm.

For the first case (see figure 6) is clear that on average the results obtained with the video sampling algorithm are better than the ones collected with the analysis of ffmpeg metadata. A justification for the worst results obtained in general for the latter is the fact that the consecutive frame

comparison algorithm applied may be only sensible to cuts so the moment when those occur may not be the same for every video which lead to different images under analysis and ultimately divergences in the results that may not occur with the sampling algorithm for a sufficient small τ.

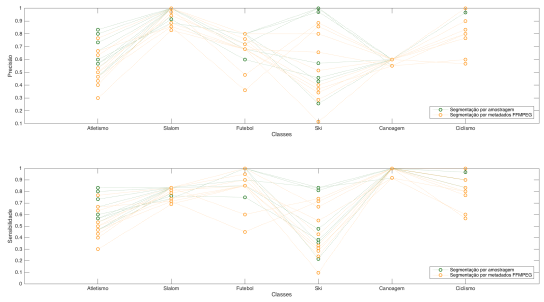

Fig. 6. Variation of precision and recall according to the segmentation algorithm

In the second case (see figure 7) as expected the mixture of features produces a better result in general. Each feature is capable of describing an image and a video by itself but adding more than one descriptor allows the characterization of the image to be more complete and therefore produce better result.

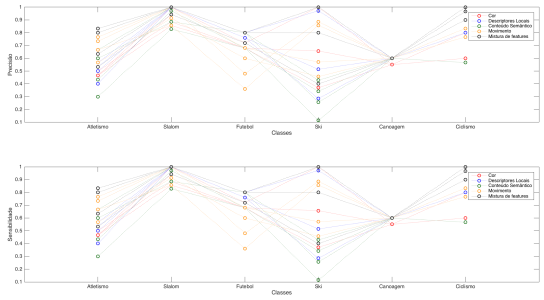

Fig. 7. Variation of precision and recall according to the feature used for comparison

In the last case (see figure 7) the graph search algorithm produces the best results in general which was also expected. The weighted sum algorithm allows multiple correspondence between the query key-frames and the database frames which produces a lot of false positives lowering the quality of the results.

The average precision (P) and recall (R) values for each algorithm can be seen in the table VII.

According to the images 6, 7 and 8 and the table VII we conclude the E_{10} is the one that provides the best result and will be use along the remaining of this work.

In the next step we check the ability of the algorithm to reconstruct the query video by compiling the database key-frames. A simple approach would be to consider only the best match for every key-frame however in this step we prioritize

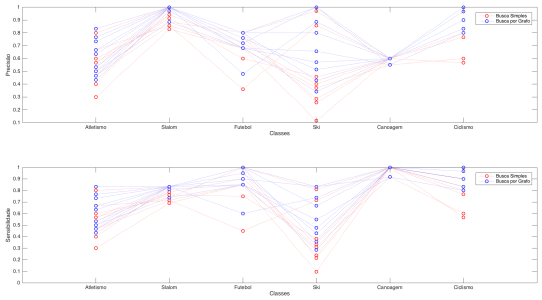

Fig. 8. Variation of precision and recall according to the search algorithm

	P	R		P	R
E_1	0,75	0,79	E_{11}	0,59	0,64
E_2	0,79	0,83	E_{12}	0,64	0,70
E_3	0,66	0,73	E_{13}	0,51	0,58
E_4	0,67	0,73	E_{14}	0,68	0,72
E_5	0,85	0,89	E_{15}	0,68	0,74
E_6	0,76	0,79	E_{16}	0,68	0,73
E_7	0,81	0,85	E_{17}	0,70	0,76
E_8	0,73	0,79	E_{18}	0,65	0,71
E_9	0,75	0,80	E_{19}	0,76	0,80
E_{10}	**0,87**	**0,91**	E_{20}	0,81	0,86

TABLE VII
PRECISION AND RECALL VALUES FOR EACH TESTED ALGORITHM

the sequences with more than one frame from the same video, in other words, using sequences of frames is preferred to jump around the videos so that a transition only occurs when the video correspondent to the last match does not have a valid pair for the current one. A small section of the results obtained for a video of the ski jumping class can be seen in figure 9.

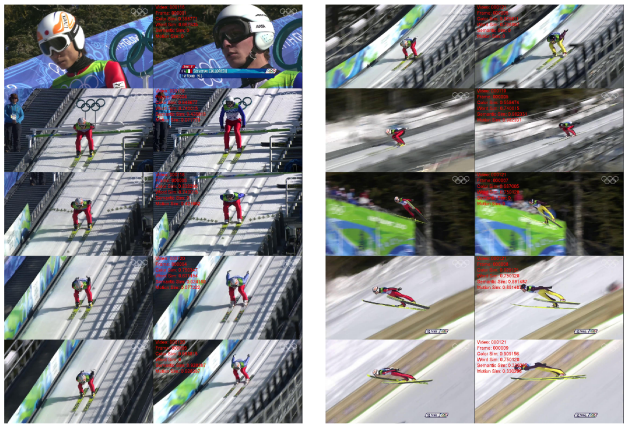

Fig. 9. Part of the results obtained for the reconstruction of the query video using database data

In figure 9 we have two columns where on the left of each one are presented the query key-frames and on the right the match from the database. The similarities obtained for each feature, the video index on the database and the key-frame

index are shown in red at the top left corner of each match. As it can be seen for each query key-frame an almost identical image was identified.

In the last type of search we want to retrieve similar sequences. The user defines a value correspondent to the number of seconds a similar database sequence must have to be considered similar to any query sequence. The algorithm will construct a graph and search for connections between consecutive key-frames with consecutive matches with the amount of time introduced. Finally all the sequences with that amount of time (if any) are presented to the user. The results obtained for a video of the ATL class with sequences of 17 seconds can be seen in figure 10.

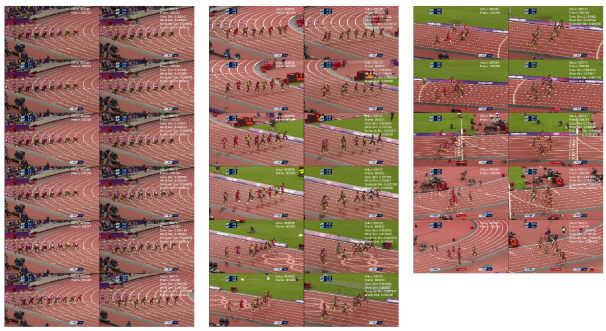

Fig. 10. Results for the segment search for a video of the class ATL

As we can see in figure 10 the images of the two sequences are almost copies of each other proving that the algorithm developed is capable of retrieving similar videos based on the sequence search. The definition of "sequence" used in this step is very strict, since the frames must all be consecutive. A relaxation parameter may be introduced to expand the set of results.

B. Video search engine scalability evaluation

In this stage we tested the scalability of the video search engine in a dataset of 42576 videos collected from the YT-8M dataset provided by Google. The videos were already processed and we have access to video level features (1024 histogram with pictorial information about the video), frame-level features (1024 histogram with pictorial information about the frames), a set of labels associated with each video and a link to the Youtube website. The methodology used in this process starts with a segmentation of the dataset into a group of 30 videos using the video-level features. Then the decision tree to collect the correspondences to the query key-frames and the graph are constructed and the search procedure is equal to the one described before. Five videos (V_x) were chosen randomly producing the results in tables VIII and IX that show the average elapsed time on each stage of the algorithm and the labels associated with the query video and the matches (M_y), respectively.

According to the table VIII we confirm that the developed engine was able to produce a result in under 10 second for a database with considerable amount of videos however the

	V_1	V_2	V_3	V_4	V_5
T1 (s)	0,26	0,65	0,51	0,29	0,63
T2 (s)	5,66	9,00	7,35	5,44	6,21
T3 (s)	0,26	0,13	0,41	0,20	0,31
T4 (s)	0,16	0,25	4,37	0,15	0,24

TABLE VIII
PROCESSING TIMES FOR THE ALGORITHM SCALABILITY TEST. T1 - QUERY VIDEO FEATURE EXTRACTION; T2 - KEY-FRAME SIMILARITY SEARCH; T3 - GRAPH CONSTRUCTION; T4 - VIDEO SIMILARITY SEARCH.

	Labels	M_1	
		S	L
V_1	Concert	118.62	Performance
V_2	Game	85.94	Game
V_3	Orchestra	105.22	Concert
V_4	Trailer	58.53	Football
V_5	Animation	72.63	Video-Game

TABLE IX
RESULTS OBTAINED IN THE SCALABILITY TEST FOR THE ALGORITHM SCALABILITY TEST. S - SIMILARITY VALUE BETWEEN TWO VIDEOS; L - LABELS OF THE MATCH VIDEO

results in table IX are not the best, since a match between labels is not obtained in most of the cases. This could be explained by the type of *features* used. This dataset and the features provided have great application in deep learning systems as a training set where a map between them and the labels is created contributing to an automatic video classification system, in other words, the video search engine developed is not capable of dealing with these raw features and output a valid result.

C. User preference evaluation

The algorithm presented in IV-A uses 0.25 for every weight parameter related to each feature, however this may not be the best combination. A perfect combination that works for every user may not even exist. The objective with this small study was to understand the relative importance of the 4 features presented and the general preferences of the users. A set of 9 videos were chosen and the most similar videos in the database of 132 videos were collected for 60 different weight combinations. Fifteen independent users were asked to classify each result (Top-3 matches) as valid or invalid. In order retrieve this results a small prototype of a video search application was developed (see figure 11). Ten labels corresponding to the semantic concepts most seen among the query video key-frames were also shown in order to segment the set of videos studied.

One of the results obtained can be observed in figure 12 where the combinations are shown in a 3D space with the motion values represented in the color of the points (grayscale) with white being 1 and black being 0. Each point has also a circle around it corresponding to the number of users that considered it as a combination that produces valid results. A yellow circle means that a larger number of users considered that combination as valid and the color red means no counts.

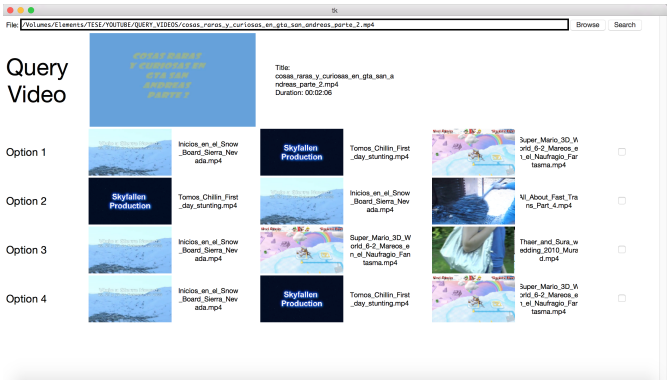

Fig. 11. Screenshot of the application used in the users preferences study

The thumbnail of the video and the matches for 5 specific points are also shown.

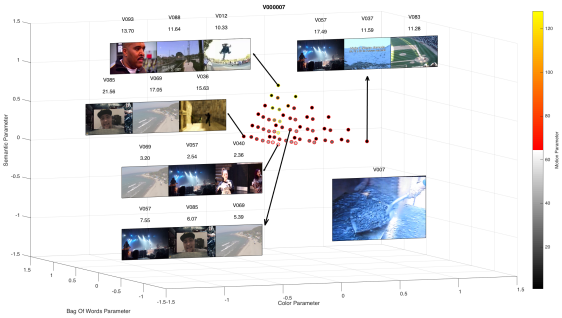

Fig. 12. Example of a result obtained for a database video

In the great majority of cases the semantic content parameter and/or the SIFT features were considered as the most important features when deciding if the matches were good or not which confirms the expected because a user has a higher tendency to use high-level features like semantic concepts when searching for similar videos instead of color or motion. Studying the results obtained for the 9 videos it was confirmed that the perfect combination does not exist, since each video had its own set of good combinations. On the other hand the users tend to agree in the set of good combinations for a specific video.

A relevance feedback system was also tested with the same prototype. In this case the first results presented considered the extremes of the weight combinations, in other words, a feature has a weight almost equal to 1 and the other to 0. From this first step the user chooses which ones are the best and a new solution is calculated using the mean from the good matches. In the next iteration new results are shown and the user repeats the process until the variance of the good combinations is small enough. This test was developed for the same 9 videos as before with only one user and the results can be seen in figure 13.

The results in figure 13 match the expected, since the semantic content and dense-SIFT model features have the highest values while the color and motion are smaller. The justification for this is the same provided for the results of the previous study, in other words, a user tends to utilize high-level features to evaluate the results more often than features like color or motion considered low-level features.

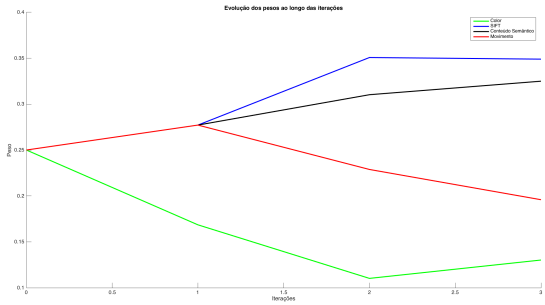

Fig. 13. Feature parameters weight evolution analysis using user feedback

V. Conclusions

This thesis focused on the problem of indexing and retrieval of videos in large databases. The main contributions of this work are the innovative, fast and flexible algorithm for video search based on graphs. After the analysis of multiple options the best engine involved the use of sampling segmentation, mixture of features and graph search. Besides the flexibility in relation to the features used (Color, Texture, Motion, Semantic Content) the algorithm is also able to produce different results according to the type of search desired by the user. The scalability was also tested with very good results in terms of time but no so great in terms of results precision because the algorithm was not well adapted to the features used. The studies conducted in this thesis try to identify the preferences of the users when considering the similarity between two videos. Although the results seemed promising, with the identification of two features with higher preference in relation to the others the reduced number of videos, features and ultimately users does not allow us to extract any further conclusions on this matter.

In the future it is intended to improve the algorithm speed, study the possible implementation of new features and fundamentally speed up the motion descriptor extraction process. The scalability did not produced the intended results and a more profound test is needed to confirm if the origin of the issues is due to the features provided by the Youtube or the algorithm. Finally the expansion of the study to a broader set of videos and users is needed to extract a valid set of conclusions from them.

References

[1] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, 1997.

[2] H. T. Shen, B. C. Ooi, and X. Zhou, "Towards effective indexing for very large video sequence database," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 730–741.

[3] W. Barhoumi and E. Zagrouba, "On-the-fly Extraction of Key Frames for Efficient Video Summarization," *AASRI Procedia*, vol. 4, pp. 78–84, 2013.

[4] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1. IEEE, 1998, pp. 866–870.

[5] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[6] E. J. Y. C. Cahuina and G. C. Chavez, "A New Method for Static Video Summarization Using Local Descriptors and Video Temporal Segmentation," in *XXVI Conference on Graphics, Patterns and Images, SIBGRAPI*, 2013.

[7] L.-H. Chen, K.-H. Chin, and H.-Y. Liao, "An integrated approach to video retrieval," in *Proceedings of the nineteenth conference on Australasian database-Volume 75*. Australian Computer Society, Inc., 2008, pp. 49–55.

[8] G. Pass and R. Zabih, "Comparing images using joint histograms," *Multimedia systems*, vol. 7, no. 3, pp. 234–240, 1999.

[9] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. M. Rao, and G. Seetharaman, "Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video," in *Information Fusion, 2010. FUSION 2008. 13th IEEE International Conference on*. IEEE, 2010, pp. 1–8.

[10] Y. Zhang, C. Xu, Y. Rui, J. Wang, and H. Lu, "Semantic event extraction from basketball games using multi-modal analysis," in *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007, pp. 2190–2193.

[11] D. B. Russakoff, C. Tomasi, T. Rohlfing, and C. R. M. Jr, "Image Similarity Using Mutual Information of Regions," in *Proceedings of the 8th European Conference on Computer Vision*, 2004.

[12] H. Zhang, J. Y. Wang, and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," in *Image Processing, 1997. Proceedings., International Conference on*, vol. 1. IEEE, 1997, pp. 13–16.

[13] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *Proceedings of the tenth ACM international conference on Multimedia*. ACM, 2002, pp. 533–542.

[14] V. Kantorov and I. Laptev, "Efficient Feature Extraction, Encoding, and Classification for Action Recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 2593–2600.

[15] A. Kojima, N. Sakurai, and J. I. Kishigami, "Motion detection using 3d-FFT spectrum," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 5. IEEE, 1993, pp. 213–216.

[16] H. Jegou, F. Perronnin, M. Douze, J. Snchez, 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, 2012.

[17] M. Douze, J. Revaud, J. Verbeek, H. Jgou, and C. Schmid, "Circulant temporal encoding for video retrieval and temporal alignment," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 291–306, 2016.

[18] W. Liu, T. Mei, and Y. Zhang, "Instant Mobile Video Search With Layered Audio-Video Indexing and Progressive Transmission," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2242–2255, Dec. 2014.

[19] S. Lee, N. Maisonneuve, D. J. Crandall, A. A. Efros, and J. Sivic, "Linking Past to Present: Discovering Style in Two Centuries of Architecture," in *ICCP*. IEEE Computer Society, 2015, pp. 1–10.

[20] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[21] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[22] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-Motion Video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, Jan. 1993.

[23] B. Coskun and B. Sankur, "Robust video hash extraction," in *Signal Processing Conference, 2004 12th European*. IEEE, 2004, pp. 2295–2298.

[24] B. Coskun, B. Sankur, and N. Memon, "Spatiotemporal transform based video hashing," *ieee Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 2006.

[25] X. Zhou, M. Schmucker, and C. L. Brown, "Video Perceptual Hashing Using Interframe Similarity," in *Sicherheit*, ser. LNI, vol. 77. GI, 2006, pp. 107–110.

[26] X. Zhou, X. Zhou, and H. T. Shen, "Efficient similarity search by summarization in large video database," in *Proceedings of the eighteenth conference on Australasian database-Volume 63*. Australian Computer Society, Inc., 2007, pp. 161–167.

[27] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual Semantic Search: Retrieving Videos via Complex Textual Queries," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[28] K. I. Kim, J. Tompkin, M. Theobald, J. Kautz, and C. Theobalt, "Match graph construction for large image databases," in *European Conference on Computer Vision*. Springer, 2012, pp. 272–285.

[29] M. Tapaswi, M. Bauml, and R. Stiefelhagen, "Book2movie: Aligning video scenes with book chapters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1827–1835.

[30] J. Shao, H. T. Shen, and X. Zhou, "Challenges and techniques for effective and efficient similarity search in large video databases," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1598–1603, 2008.

[31] F. Lee, K. Kotani, Q. Chen, and T. Ohmi, "Fast Search Algorithm for Short Video Clips from Large Video Database Using a Novel Histogram Feature." IEEE, 2008, pp. 1223–1227.

[32] G. de Oliveira Barra, M. Lux, and X. Giro-i Nieto, "Large scale content-based video retrieval with LIvRE," in *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*. IEEE, 2016, pp. 1–4.

[33] R. Yan, A. G. Hauptmann, and R. Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 343–346.

[34] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," in *Toward category-level object recognition*. Springer, 2006, pp. 127–144.

[35] E. Spyrou, P. Mylonas, and Y. Avrithis, "Using region semantics and visual context for scene classification," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 53–56.

[36] J. Revaud, M. Douze, C. Schmid, and H. Jegou, "Event Retrieval in Large Video Collections with Circulant Temporal Encoding." IEEE, Jun. 2013, pp. 2459–2466.

[37] A. Papazoglou, L. Del Pero, and V. Ferrari, "Video Temporal Alignment for Object Viewpoint," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 273–288.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[39] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, 2014.