Video index and search services based on content identification features

Dr. Gregory Doumenis
Senior Research Associate,
ICCS/NTUA
Zografou, Athens, GREECE
greg@telecom.ntua.gr

Serafeim Papastefanos
Research Associate,
ICCS/NTUA
Zografou, Athens, GREECE
serafeim@telecom.ntua.gr

Victor Mateevitsi
Research Associate,
ICCS/NTUA
Zografou, Athens, GREECE
victoras@telecom.ntua.gr

Dr. Fotis Andritsopoulos
Senior Research Associate,
ICCS/NTUA
Zografou, Athens, GREECE
fandrit@telecom.ntua.gr

Nikos Achilleopoulos
Archetypon SA
Athens, GREECE
nax@archetypon.com

Anton V. Mikhalev
Elecard Ltd.
Tomsk, RUSSIA
anton.mikhalev@elecard.ru

Abstract
In this paper we examine the methodology, architectural issues and preliminary statistical results for identifying the presence and position of a given query clip within a massive collection of video content. This work is part of the European Union FP6 IST Programme project DIVAS (Direct Video & Audio Content Search Engine). The concept is applicable to a number of use cases ranging from video clip search into large repositories, to DRM and content policing in the internet.

Keywords
Video search, video indexing, segmentation.

INTRODUCTION
The concept of direct video search can be outlined with the following simple use case: A user somehow obtains a short video clip, possibly of low resolution and quality, and wishes to download its origin, i.e. the complete film or stored video broadcast to which this clip belongs to.

In the 'digital right management' use cases, characteristic parts of a video film are submitted to 'policing' nodes. These have the task to detect whether the video film is passed through the node. In either case, DIVAS project has set as its primary objective not to use metadata, nor to resort to additional watermarking and/or image based fingerprinting methods. Further, its goal is to prefer the encoded domain and to refrain as much as possible from solutions entailing transcoding. Figure 1 describes the interoperability between the system and the user for direct video searching.

A number of works have already been published about video search directly on the compressed domain [1]-[3]. Direct video search methods can be can be summarized into two main categories: (i) match through key frames, where the key frames of the query clip are extracted and compared with the key frames of the video to be searched and (ii) match through collective characteristics of sequences of successive, possibly sub sampled frames. In [1], which belongs to the first category, key frames are extracted using the cumulative directed divergence and then are compared through their modified Hausdorff distance. In [2], which belongs to the second category, an average of the colors of each I frame is extracted, thus creating a feature that is used for sequential search, audio features are then used to further enhance the results. [3] contains a mix of both approaches. The described work belongs to the second category, however scene changes (key frames) will be used as a first level search method to quickly exclude candidates from search operations.

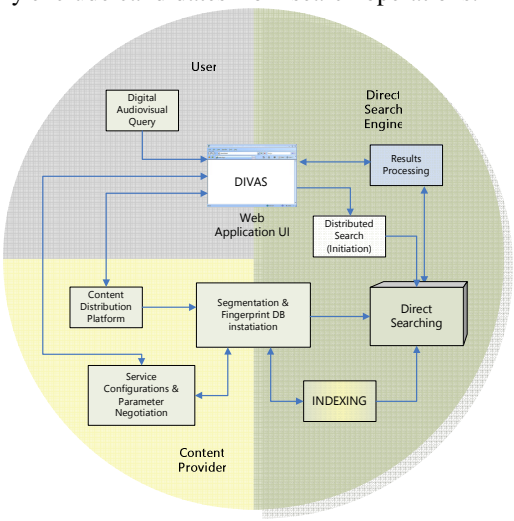

Figure 1. Interactivity of components between the user and the DIVAS system

DIVAS utilizes a set of features and search techniques to improve speed and detection accuracy. These features are extracted directly from the compressed domain. A survey of such features can be found in [4]. The correlation of extracted multimedia features (video and audio features, but also speech) is the cornerstone of the DIVAS multimedia search engine. As depicted in Figure 2, each submitted user clip for query is demultiplexed into video

and audio. Video and audio fingerprints are being processed separately and are correlated in a later stage in order to deliver an accurate result as a combined answer from both video and audio at the end of that process.

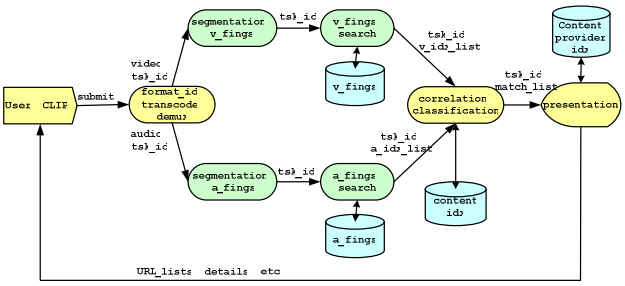

Figure 2: Search architecture using combination of video and audio identifiers

The present paper focuses exclusively on the video part of DIVAS.

VIDEO CHARACTERIZATION & SEARCH

The search operations applied on video can be performed more efficient and with increased accuracy if the video files itself are characterized by several video identifiers. The term "video identifier" stands for a specific set of video characteristics and it is extracted directly from the compressed domain.

For video characterization, features of several classes are used. In the first class there are features that make some sort of segmentation. Segmentation means logical dividing long video sequence into several smaller subsequences. The most common (and understandable for human) is segmentation based on scene changes or shot detection. Shot detection can be defined as the process of detecting transitions between two consecutive shots, so that a sequence of frames belonging to a shot will be grouped together [5].

A shot is the basic unit in video production. From the general point of view, a video film is a series of edited video shots. The transition between shots usually corresponds to a change of subject, scene, camera angle, or view. Therefore, it is very natural to use shots as the unit for video indexing and analysis, and the first step in such applications is to split long video sequence into video shots [4].

Different algorithms can be used as a basis for scene changes detection as described in [6] and namely the most popular are the brightness histogram comparing, image segmentation and motion amount calculation.

The second class contains features that characterize every video frame (not sequences of frames). The examples of such features might be brightness histogram, average brightness of vertical or horizontal slices, or amount of certain objects (faces, cars, etc), detected on the frame.

Features of the second class are more complex to extract and thus the processing power required is increased. In addition, this extracted information is characterized by increased memory footprint and thus more storage is needed. However, these methods have been found to be tolerant to considerable distortions of the image plane (e.g. images captured with side-cuts), so they are widely adoptable.

Each identifier is assigned by priority factors characterizing its importance which is used for the hierarchical search. For the current DIVAS system, a set of features has been selected to compose a robust and efficient approach on video searching:

- Segmentation features:
  - scene change by brightness histogram
  - scene change by image segmentation
  - scene change by motion amount
- Per-frame features:
  - brightness histogram
  - average brightness of vertical and horizontal slices with normalization
  - image segmentation based on pixel brightness

Assuming a set of -non-uniquely characterizing- extracted features, accurate video matching can be performed by an hierarchical and recursive search of state space decimation.

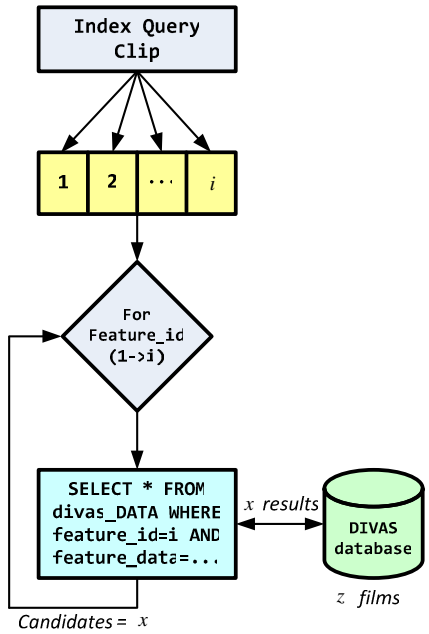

Figure 3. Iterative process for searching using different video identifiers

The system uses at first the feature identifier with the highest impact and builds an SQL query as depicted in Figure 3. The database will respond with a subset of the stored films. Assuming that the repository holds N films, the database responds to the query returning K potential matching candidates. The next search (based on the second higher impact identifier) will be performed on the subset of

K candidates and so on. This iterative process will last until one of the following occurs:

- The subset K (standing for candidates) reduces to a singleton, which means that the DIVAS system concluded that the film is found.
- All the feature identifiers have been used, which means that there are no more criteria to be applied for searching. The subset K might contain more than one findings in this case.

The previous technique operates well, under the assumption that the search algorithm *always* returns a subset with all true matches and some false ones.

In this paper, we focus on the identification of (relatively rare) "events" in the video content, as a means to characterize the content. Further, we explore the possible uniqueness of such characterization based on metrics related to the robustness of the event detection mechanism, the number of detected events and the size of the video repository.

Top Level Candidate Identification

The detection of scene changes is the key for finding a signature index for stored video steams and for initiating a top level search. The concept, characteristics and performance of this top level search are key parameters for the success of DIVAS. It entails all scalability and reliability issues. For the rough estimation, the top level indexing will create an index that leads to 85-90% probability of correct finding. Means are inherently provided, at theoretical, algorithmic and operational level, for maximizing *recall* at the expense of *precision*. Thus matching candidates will not be lost, while wrong findings will be eliminated at the next for in-depth evaluation.

Once this top level search has identified *an initial set of candidates*, a whole palette of deeper search possibilities, resembling more and more direct comparisons on a frame-by frame level are performed. Data for these are stored in a comprehensive xml repository along the lines of the MPEG 7 standards.

For the top level discussed in this paragraph, scene changes are seen as 'events'. We consider

- a sequence R of experiments, each resulting into the (rare) event TRUE with probability *e*, otherwise the event FALSE. Experiments are in the context of video frames and outcomes are represented with the presence of a scene change. This outcome is rather rare, i.e. very few and widely dispersed (in time) frames will represent scene changes. Ideally scene changes are not influenced by the type and settings of the encoder, however see below.
- a sequence P of *perceived outcomes* of the elements of R. During the processing of video (in whatever form) in order to arrive at the corresponding index, an event is not detected (missed detection) or a non events taken as an event (false alarm).
- a subsequence Q of R, consisting of consecutive elements of R. This is the 'query' clip video. This might be of low quality, however it is supposed, that here (due to its limited length) detailed analysis leads to an almost perfectly reliable detection of scene changes, contrary to sequence P.

Sequence Q can be compared with P, but not with the original R. A criterion for detection is defined, examining whether or not Q is synchronised with R. The criterion is based on the count of the numbers of positions where the presence of an event in Q and P do not coincide (count of mismatches).

When sequence Q is synchronized with P (meaning that Q is extracted from point of P under examination) then there is a different mismatch probability than when Q is not synchronized with P. We will represent these probabilities with e_s and e_n correspondingly.

The total sum of mismatches is minimised when Q is synchronized to P, i.e. the query clip has been found within the content base, and so synchronized mismatch probability in that case will be smaller than non synchronized mismatch probability. In operational terms this count can be achieved by direct algorithmic methods or by resorting to normal data base sql queries, edited dynamically at run time.

Temporal event detection under uncertainties

According to previous sections, the search procedure follows a scalable approach for video searching. First it attempts to find candidates by using a sequence of scene changes.

We use the event positions of P *and* Q as starting points (anchors) for the comparison. If we have a match, then at least one event of P and Q have to coincide, otherwise we have a maximal mismatch number equal to the sum of events in P and events in Q. Usage of events as anchors can be seen in Figure 4.

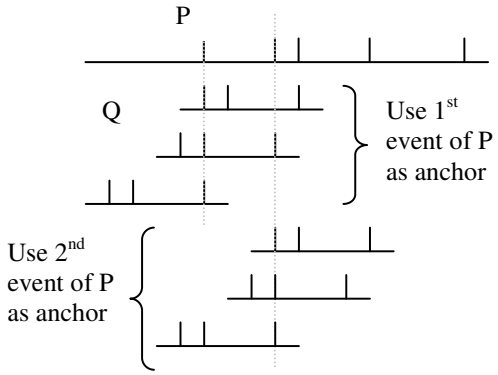

Figure 4: Usage of events as anchors

For each different position, the number of mismatches between Q and P is counted and, if that number does not exceed a certain threshold then Q has been found in that position in P.

Experimental results

Some initial results can be extracted by calculating the mismatch probability between a synchronized and a non synchronized clip. The mismatch probability was calculated by counting the number of mismatches between P and Q in a sample video, and was found equal to 0,0045 for e_s and 0,028 for e_n. When a query clip with 1000 frames is used to search, then, for each case we will have 1000 repeated Bernoulli trials that follow the binomial distribution. The distribution of both e_s and e_n can be seen in Figure 5.

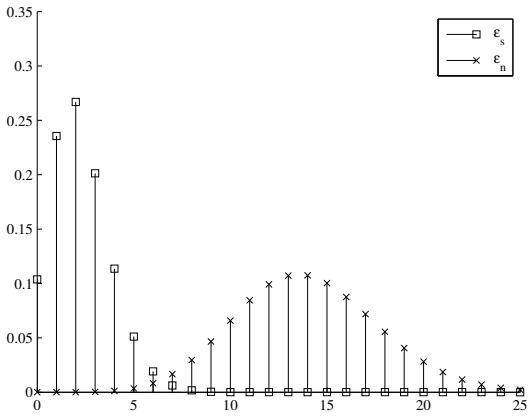

Figure 5: Mismatch distribution

As can be seen, the peak of the distribution for the synchronized cases is around 2 mismatches, while the peak of the distribution for the non-synchronized case is around 14 mismatches. We can see that the gap between the synchronized and non-synchronized case is enough to make a decision; however there will be some cases that will fall in-between the two distributions. Whenever that happens then there will be a match decision that will decrease the precision of the top level search but will leave the recall to high levels; the next level searches will be enough to improve the precision of the framework to the expected levels.

CONCLUSION

This paper presented a method for direct searching of video content, based on the content itself without using metadata information. The presented approach applies directly to the compressed domain without fully decoding the stream. Its video file is characterized by several identifiers and the searching is being applied based on these tags. Following an hierarchical approach, the method behaves ideally in terms of computational complexity while the accuracy of the system is very high. Some preliminary results are also shown, based on the first level of that hierarchical searching.

ACKNOWLEDGMENTS

The work described in this paper has been funded in part by the *European Union FP6 IST Programme project DIVAS (IST-2-045081-STP)* [8].

REFERENCES

[1] S. H. Kim and R.-H. Park, “An efficient algorithm for video sequence matching using the modified hausdorff distance and the directed divergence,” IEEE Trans. Circuits Syst. Video Technol., no. 7, Jul. 2002.

[2] J. Yuan, Q. Tian, and S. Ranganath, “Fast and robust search method for short video clips from large video collection,” in ICPR ’04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04) Volume 3, 2004, pp. 866–869.

[3] A. Jain, A. Vailaya, and X. Wei, “Query by video clip”, Multimedia syst., vol. 7, no. 5, 1999.

[4] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, and H. Sun, “Survey of compressed-domain features used in audio-visual indexing and analysis,” J. Vis. Commun. Image R., vol. 14, no. 2, Jun. 2003.

[5] U. Gargi, R. Kasturi, and S. H. Strayer, “Performance characterization of video-shot-change detection methods,” IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 1, Feb. 2000.

[6] G. Ahanger, and T.D.C Little, “A Survey of Technologies for Parsing and Indexing Digital Video, JVCIR(7), No. 1, March 1996, pp. 28-43.

[7] Janco Calic, “Highly efficient low-level feature extraction for video representation and retrieval”, 2004.

[8] DIVAS Web Site, [Online]. Available: http://www.ist-divas.eu