Searching for near-duplicate video sequences from a scalable sequence aligner

Leonardo S. de Oliveira, Zenilton Kleber G. Do Patrocínio Jr., Silvio Jamil F. Guimarães, Guillaume Gravier

Searching for Near-duplicate Video Sequences From a Scalable Sequence Aligner

Leonardo S. de Oliveira, Zenilton K. G. do Patrocínio Jr,
Silvio Jamil F. Guimarães
VIPLAB/ICEI – PUC Minas
{leo.oliveira, zenilton, sjamil}@pucminas.br

Guillaume Gravier
IRISA/TEXMEX - Bretagne Atlantique
guillaume.gravier@irisa.fr

***Abstract*—Near-duplicate video sequence identification consists in identifying real positions of a specific video clip in a video stream stored in a database. To address this problem, we propose a new approach based on a scalable sequence aligner borrowed from proteomics. Sequence alignment is performed on symbolic representations of features extracted from the input videos, based on an algorithm originally applied to bio-informatics. Experimental results demonstrate that our method performance achieved 94% recall with 100% precision, with an average searching time of about 1 second.**

***Keywords*-Videoclip localization; sequence aligner; multimedia retrieval**

I. Introduction

Search for quasi-invariant, near-duplicate, repeated sequences in multimedia data has received tremendous attention over the past 10 years. In particular, video streams, mostly coming from TV, have been under studies [1], [2], [3], [4] with the goal of detecting repeating ads, jingles or songs for structuring purposes.

Current methods for solving the video retrieval/localization problem can be grouped in two main groups with distinct approaches: (i) computation of video signatures after temporal video segmentation, as described in [5], [6]; and (ii) use of matching algorithms after transformation of the video frame content into a feature vector or symbolic representation, as described in [4], [7].

In order to propose a scalable approach, pattern identification methods from proteomics can be considered here. Building upon the fundamental algorithm, FASTA [8], dedicated to the alignment of symbolic sequences with specific distortions, several heuristics, such as BLAST [9] or SNAP [10], have been proposed. They efficiently find approximate repetitions in protein sequences by exploiting the peculiarities of DNA data, and as such, do not apply straightforwardly to multimedia data, even if a symbolic representation of the latter exists.

Our method transforms the video data into a symbolic representation, which can then be efficiently indexed and searched for, using an algorithm built upon a sequence aligner (SNAP) [10] used in bio-informatics.

This paper is organized as follows. In Section II, our near-duplicate video sequence identification method is described using a scalable sequence aligner applied to a symbolic representation which is computed from the multimedia data. In Section III, we discuss about the experiments and the setting of algorithm parameters. Finally, in Section IV, we draw some conclusions and suggest future works.

II. Scalable video aligner process – SVAP

The main goal of the near-duplicate video sequence identification problem is to identify occurrences of sub-sequences of the target video that are similar to a query video. Our proposal can be subdivided into two main tasks: (i) multimedia data representation, which transforms the video data into a symbolic representation; and (ii) alignment process, which is based on a scalable alignment process used in bio-informatics to efficiently find near-duplicate occurrences of protein sequences.

Once the multimedia data is converted into a symbolic representation, the index structure can be built. To this end, short subsequences (called video fragments) of a fixed size s are sequentially read from each video in the database, and used to build a hash table, where each key represents a video fragment.

Afterwards, given a query video to search for, SVAP draws multiple video fragments of the same length s from it and performs a look-up in the hash index to find locations in the database that contain the same sequence of frames (video fragment). It then computes the edit distance between the video query and each of these candidate locations to find the best matches.

These two steps are described in further detail in the following sections.

A. Multimedia data representation

Usually, the sub-sequence matching rely on symbolic representation of the data, like BLAST and SNAP, however this is clearly not the case of multimedia data which are continuous and high dimensional. Therefore, to apply a method based on symbolic representations, it is necessary to transform this complex data into a simpler one. To that end we propose transforming multimedia data into a character sequence, a methodology that can be summarized as follows:

1) The video is partitioned into video frames, which are represented by GIST and saliency maps [11].
2) The dimension of feature vectors is reduced by PCA, maintaining about 90% confidence. It is important to note that for computing the reduction matrix, the feature vectors are sub-sampled considering one frame per second.
3) Using K-means clustering, an alphabet is created, where each centroid of the clusters is a symbol of the alphabet. This step allows each video frame to be represented as a symbol of this alphabet.
4) Finally, each video fragment, which is a sequence of video frames, can be represented by a sequence of symbols (strings of an alphabet) and is stored into a hash table.

Considering that we are interested in studying the behavior of the alignment process applied to a multimedia data representation task, we decide to set the size of the alphabet and the size of video fragment to 256 symbols and 4 frames, respectively. The choice of these values is based on preliminary empirical studies.

B. Alignment process

The video alignment process proposed in this paper (illustrated in Algorithm 1) is based on the alignment process presented in [10]. The main difference between both alignments is the fact that our method can return an arbitrary number or multiple hits (in descending order of match quality) for any query (*line 18*).

The first step in the alignment process is the creation of an index of all videos to be considered when searching for the query video. The size of each video fragment s represents the size of the hash table's keys, which corresponds to the number of frames used when comparing the video query against the database. This number ultimately affects directly the size of the hash table (greater values of s means a larger hash table) and the computational costs. For the purposes of this study, we adopted a video fragment size of 4, meaning that every segment of 4 frames from the video database represents one key in the hash table.

The second and final step of the alignment process is the actual search for a given video query in the database, which, aside from the differences previously discussed, is similar to SNAP. Our method takes a parameter confidence threshold c that represents the minimum difference in edit distance scores between the best and all other alignments. However, since we are in fact returning multiple hits (instead of only the best one), this parameter allows us to control the error between the best video query alignment and all other identifications. The maximum error is limited to d. It is important to note that the number of video fragments to be tested (n) plays an important role in this process since it helps us control the accuracy of our process.

In Table I, it is described the parameters of our alignment step, namely, video fragment size (s), video fragment samples (n), maximum video distance (d), confidence threshold (c) and max hits (h).

Table I
PARAMETERS OF THE ALIGNMENT STEP

Parameter	Meaning
Video fragment size (s)	Number of frames for each video fragment.
Video fragment samples (n)	Number of video fragments which is randomly sampled for each query.
Maximum video distance (d)	Maximum edit video distance from reference sequence to allow for an alignment.
Confidence threshold (c)	Difference in edit distances between a video fragment's best and second-best alignments needed to report it as confidently aligned.
Max hits (h)	Maximum number of index locations to be checked for a video fragment. Some video fragments contain frames that are far too common (e.g., black frames between scenes) and may be present in a large number of videos, so they are just ignored.

Algorithm 1: The alignment using SVAP.

```
   input : Video index (index)
           Video query (vQuery)
           Video fragment samples (n)
           Maximum video distance (d)
           Confidence threshold (c)
           Max hits (h)
   output: Locations of the hits (locations)
1  d_best ← ∞;
2  d_second ← ∞;
3  for i = 1 to n do
4      S ← i^th video fragment sample of vQuery with size s;
5      if number of entries for S ≤ h then
6          for l ∈ locations of S in index do
7              p ← l-offset of video frag i from start of vQuery;
8              VidFragsHit[p] ← VidFragsHit[p] + 1;
9          p ← unscored location with the most video fragment
           samples hitting;
10         if d_best > d then
11             d_limit ← d + c − 1;
12         else if d_second ≥ d_best + c then
13             d_limit ← d_best + c − 1;
14         else
15             d_limit ← d_best − 1;
16         e ← EditDistance(VideoFrag, Reference[p]; d_limit) ;
17         update d_best and d_second based on newly scored e;
18         FillHitsFound(locations) ;
19         if d_best < c and d_second < d_best + c then
               /* multiple hits (we have two hits
                  within distance c and no better hit
                  can be confident)                  */
20             return locations;
21         else if |non-overlapping video frags tested| ≥ d_best + c then
22             score remaining locations and break (any unscored
               location will have too high a distance);
23 if d_best ≤ d and d_second ≥ d_best + c then
       // single hit at location with best score
24     return locations;
25 else if d_best ≤ d then
       // multiple hits
26     return locations;
27 else
       // not found
28     return NULL;
```

III. EXPERIMENTS

In this section, we present some experiments to show the effectiveness of our method when compared to three other approaches: Shen's [7], BGM and 2BGM [4].

The purposes of our experiments are: (i) to show the effectiveness of our method; (ii) to compare it to other methods; (iii) to understand the tuning of parameters; and (iv) to understand the performance issues in terms of computational time cost.

A. Experimental setup

Our video corpora, or ground-truth (GT), consists of TV broadcast, recorded directly and continuously from a Brazilian cable TV channel, and Internet retrieved videos. It is composed by videos of different categories, like cartoon, news, advertisement and series. Table II shows some information about the dataset (including video length and the number of queries) The experiments searched for 92 occurrences of video clips (46 unique videos) in our dataset (54 for TV broadcast and 38 for Internet retrieved videos), with lengths varying from 9 to 61 seconds. It is important to note that we extracted some video queries from the original videos to create our query video dataset.

Table II
VIDEO CORPORA

Video dataset	Time length	Video queries	Frame rate	Category
TV Broadcast 1	1h 00m 04s	8	30 fps	News
TV Broadcast 2	35m 02s	2	30 fps	Cartoon
TV Broadcast 3	31m 50s	3	30 fps	Series
TV Broadcast 4	33m 13s	5	30 fps	Series
TV Broadcast 5	30m 27s	7	30 fps	General
Internet Video	19m 52s	21	25 fps	Advertisement
Total	3h 30m 29s	46	-	

B. Precision-Recall analysis

Precision-Recall (PR) curves give a more informative picture of the algorithm performance, since they group information about hits, miss, false positives and false negatives [12]. An optimal algorithm should have a precision-recall value of (1,1) (which means 100% of recall with 100% of precision), i.e. it managed to identify all video clip occurrences with no false positives. Moreover, in order to compute a trade-off between the precision ($\mathcal{P}$) and recall ($\mathcal{R}$) rates, we consider the F measure, as defined in Eq. 1.

Definition 1 (F measure) *The F measure is a weight harmonic mean of precision and recall rates, and it is given by*

$$F = \frac{2 \times \mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}}. \quad (1)$$

C. Comparative analysis

In order to provide a quantitative analysis, two different experiments were done. In this first one, we use the same strategy presented in [4], in which the video datasets are divided into categories and it is created one index for each category. Regarding the query process, we look for the video queries only in the index related to their category. For the second experiment, we create only one index for all video datasets.

In Fig. 1, we present the results related to the first experiment in order to compare some methods (Shen's [7], BGM and 2BGM [4]) to our proposed approach, so-called SVAP. It is important to note that in this experiment, we sum all hits, misses and falses obtained for all categories to compute the precision and recall rates.

Regarding our method, we observed that the misses were caused by short query videos, in terms of length, and sometimes by a problem on the border of the query videos. In order to prove these assumptions, we present three different results when we applied our method: (i) searching full query applied to reduced GT (F-R); (ii) searching full query applied to full GT (F-F); and (iii) searching a time-reduced query (1s) applied to full GT (TR-F). For reduced GT, we eliminate all video queries with size smaller than or equal to 10 seconds and for time-reduced query, we crop 1s (at the beginning and at the end) from the video query.

The best result obtained by SVAP, considering full GT (several hash index) and full query, is obtained by the following set of parameters: $s = 16$, $n = 200$, $d = 100$, $c = 350$ and $h = 100$. The average time for searching a specific query video is about 1 second. Moreover, the precision, recall and F measure rates, are 100%, 94% and 97%, respectively.

If we reduce, for example, the number of video fragment samples d from 100 to 25, the average time for searching a specific query video is about 0.6 second obtaining a precision, recall and F measure rates of 100%, 90% and 95%, respectively, which is still higher than other compared methods. Considering now full GT with only one hash index and full query, the average time for searching a specific query video is about 0.2 second obtaining a precision, recall and F measure rates of 100%, 89% and 94%, respectively,

As it is easy to see in the Fig. 1, and as expected, the results obtained by SVAP applied to full GT are worse than SVAP applied to the reduced GT. Thus, this experiment prove that short query videos are not well identified by this methodology. Moreover, the results for time-reduced and full query are quite similar, and consequently, show us that our method is quite robust to border discretization. Furthermore, our method is much better when compared to precision and recall rates achieved by other assessed methods.

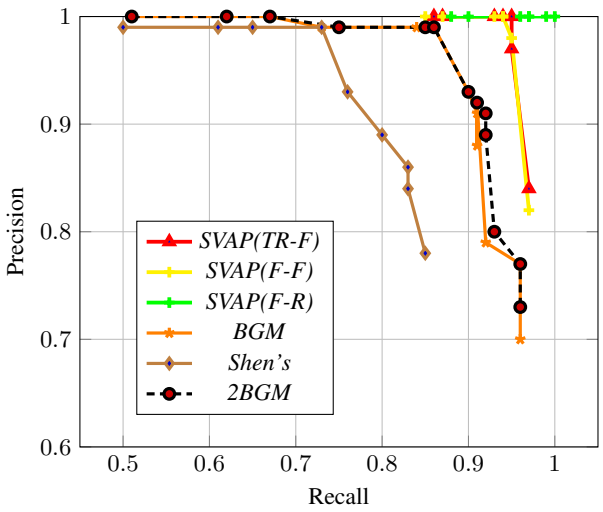

Figure 1. Precision-recall curves comparing the methods Shen's [7], BGM and 2BGM [4] and SVAP. For SVAP, we present different settings (Full query and reduced GT (F-R), Full query and full GT (F-F), Time-reduced query (1s) and full GT (TR-F).

D. Tuning of parameters

The parameters of our alignment step are described in Table I. According to our experiments, the video fragment size (s) and max hits (h) can be fixed to 4 and 100 respectively, since other values do not influence much the quality of results. The video fragment samples (n) and maximum video distance (d) have influenced the quality of results in the number of video fragment to be tested, and consequently, in the quality of results. It is important to note that confidence threshold (c) is directly related to maximum video distance (d).

1) Tuning of video fragment samples: The recall and precision rates are directly influenced by the number of video fragment samples. Recall rates increase together with high video fragment samples. However, the alignment time also increases when the number of samples increase. This should be expected since the number of searches may imply in finding a greater number of query video fragments. In the same way, precision rate also increases for high values of video fragment samples since less false positives are identified. Moreover, the precision rates for full GT are higher than

the precision rates for reduced GT for the same number of video fragment samples since the number of video fragment samples to cover the query video is smaller for short query videos. Thus, this parameter is dependent on the query video size.

2) Tuning of maximum distance: The recall and precision rates, and consequently, the F measure, are also directly influenced by the tuning of maximum distance value. To better understand the behavior of our method, we fixed the parameters video fragment size (s), video fragment samples (n) and max hits (h) to 4, 25 and 100, respectively. Furthermore, all video datasets are stored into one hash index. The F measure increases when the maximum distance increase. Moreover, the computational time also increases when the confidence threshold increases. This behavior is expected since the number of permitted alignment increases and consequently time and quality also increase.

One should notice that the F measure rates are better when we remove short query videos if we analyze the same confidence threshold. As our method works better for longer query videos, when we remove the small queries, the number of hits increases maintaining the same number of falses, and consequently, the F measure also increases.

IV. Conclusions and further works

In this paper, we propose a method, called SVAP, which identifies the locations (and the number of hits) of query videos in a video database. To cope with this identification, our alignment process borrowed from SNAP is applied to a symbolic representation computed from the multimedia data. Thus, firstly, we transform the multimedia data into a symbolic representation using several approaches for reducing the information dimensional.

In the second step, short video sub-sequences, represented by video symbols computed from the video database, are stored into a hash index video database, and for a query video to align, our method draws multiple video fragments from it and performs a look-up in the hash index to find locations in the database that contain the same video fragment followed by a computation of the edit distance between the video query and each of these candidate locations to find the best alignments, and consequently the identification of video clips.

According to experimental results, our method performance (94% recall with 100% precision) are quite better than the one proposed by [4], [7].

However, sub-sequence identification results can be highly dependent on the testing material, which is usually scarce and not especially representative. Moreover, choosing an appropriate multimedia data representation is not a trivial task, and will be studied in more details in a future work. Finally, we also intend to adapt the work presented here to unsupervised multimedia motif discovery in huge datasets, while also improving the overall performance of the algorithm.

Acknowledgment

The authors are grateful to PUC Minas – Pontifícia Universidade Católica de Minas Gerais, CNPq, CAPES and FAPEMIG for the financial support of this work.

References

[1] S.-A. Berrani, G. Manson, and P. Lechat, "A non-supervised approach for repeated sequence detection in TV broadcast streams," *Signal Processing: Image Communication*, vol. 23, no. 7, pp. 525–537, 2008.

[2] X. Wu and S. Satoh, "Temporal recurrence hashing algorithm for mining commercials from multimedia streams," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 2324–2327.

[3] J. Yuan, G. Gravier, S. Campion, X. Li, and H. Jégou, "Efficient mining of repetitions in large-scale TV streams with product quantization hashing," in *ECCV Workshop on Web-scale Vision and Social Media*, 2012.

[4] S. J. F. Guimarães and Z. K. G. Patrocinio, "A two-step video subsequence identification based on bipartite graph matching," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2012, pp. 2330–2335.

[5] A. Joly, C. Frelicot, and O. Buisson, "Content-based video copy detection in large databases: A local fingerprints statistical similarity search approach," in *Proc. of International Conference on Image Processing*, 2005, pp. 505–508.

[6] X. Naturel and P. Gros, "A fast shot matching strategy for detecting duplicate sequences in a television stream," in *Proc. of the 2nd ACM SIGMOD International Workshop on Computer Vision meets DataBases*, 2005.

[7] H. T. Shen, J. Shao, Z. Huang, and X. Zhou, "Effective and efficient query processing for video subsequence identification," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 3, pp. 321–334, 2009.

[8] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proc. of the National Academy of Sciences of the United States of America*, vol. 85, no. 8, 1988.

[9] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[10] M. Zaharia, W. J. Bolosky, K. Curtis, A. Fox, D. A. Patterson, S. Shenker, I. Stoica, R. M. Karp, and T. Sittler, "Faster and more accurate sequence alignment with snap," *CoRR*, vol. abs/1111.5572, 2011.

[11] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, Feb 2007. [Online]. Available: http://ilab.usc.edu/publications/doc/Siagian_Itti07pami.pdf

[12] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proc. of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, June 2006, pp. on–line.