MOVIE SYNCHRONIZATION BY AUDIO LANDMARK MATCHING

Ngoc Q. K. Duong and Franck Thudor

Technicolor
975 avenue des Champs Blancs, CS 17616, 35576 Cesson Sévigné, France
{quang-khanh-ngoc.duong, franck.thudor}@technicolor.com

ABSTRACT

This paper addresses movie synchronization, *i.e.* synchronizing multiple versions of the same movie, with an objective of automatically transferring metadata available on a reference version to other ones. We first exploit audio tracks associated with two different versions and adapt an existing audio fingerprinting technique to find all temporal matching positions between them. We then propose additional steps to refine the match and eliminate outliers. The proposed approach can efficiently handle situations where temporal scene edits occur like scene addition, removal, and even the challenging scene re-ordering case. Experimental results over synthetic editorial data show the effectiveness of the proposed approach with respect to the state-of-the-art dynamic time warping (DTW) based solution.

Index Terms— Movie synchronization, landmark matching, scene addition, removal, and re-ordering.

1. INTRODUCTION

Nowadays, different versions of a given video content may co-exist and be accessed by end-users. A typical example is successive DVD versions of a blockbuster that can be found a couple of years after the theatrical one in extended version or in director's cut version. Another example is old movies that are brought up to date with new additional visual effects or in a colorized version. Due to local censure, some cleaned up versions may also appear in which violent, religious, sexual or political scenes are removed. Temporal edits that can occur between those versions include *scene addition* or *removal* and scene *re-ordering*. Thus, there is a real need for movie synchronization with an objective of automatically transferring some metadata available on the reference version to the second version where those metadata are absent. Such metadata may come from an artistic work, *e.g.* subtitles or chapters, but they may also be generated through a computational analysis of the audio-visual content itself, *e.g.* which characters are present, indoor/outdoor scene, etc. In both cases, transferring directly the metadata from one version to another avoids the time consuming task of metadata re-generation.

In this paper, we propose a synchronization algorithm grounded on a successful audio fingerprint technique [1] with significant extensions in order to address the considered scene addition, removal and re-ordering situations. The developed method is robust to distortions of the audio stream, *e.g.* under compression or format conversion, and computationally very efficient thank to the advantage of the fingerprinting technique.

The rest of the paper is organized as follows. We first discuss the related work in Section 2. We then present the proposed synchronization approach in Section 3 where, for clarity, each step in the global workflow is organized into a subsection. We illustrate the experimental results to validate the performance of the proposed algorithm in Section 4. Finally, we conclude in Section 5.

2. RELATED WORK

For such a content synchronization purpose, either image descriptors, represented in the video frames, or audio features, represented in the audio samples, are usually exploited for matching [2]. As examples, Thudor *et al.* addressed the video frame alignment for automatic chaptering of VoD content based on video feature matching in [3] while Bryan *et al.* [4] exploited the audio landmark feature for clustering and synchronizing multi-camera video. Concerning audio based approaches, audio fingerprinting is naturally considered due to its compactness, its discriminative power, and its robustness to various kinds of distortion [1, 5]. Originally, fingerprint techniques have been developed for content identification purpose, *e.g.* Shazam's music identification system. Recently, they have also been exploited for other applications such as identifying repeated sound events in personal recordings [6], clustering and synchronizing multi-camera video [4], synchronizing media components streamed over different networks for second screen TV applications [7], or retrieving large scale video for copy detection [8]. Another state-of-the-art approach for synchronizing two related contents is Dynamic Time Warping (DTW). As examples, Macrae *et al.* used it for music video alignment [9] and Anguera *et al.* used it for spoken word matching [10]. This technique efficiently tackles the problem of scene addition and removal by finding

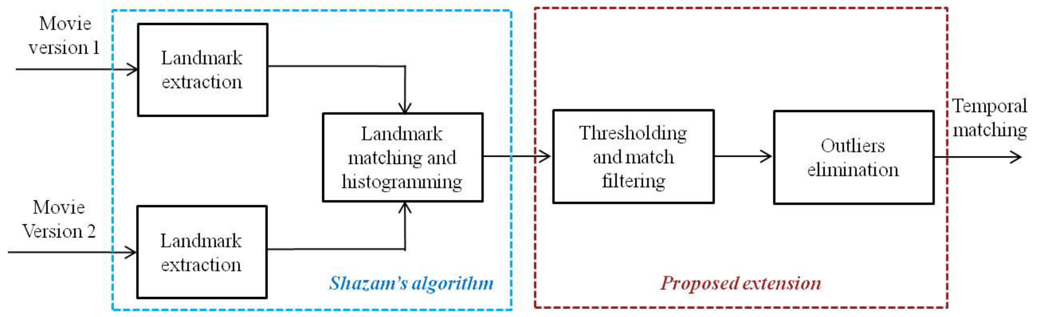

Fig. 1. Global workflow of the proposed algorithm.

an optimal path representing the highest similarity between two feature sequences extracted from those contents. However, DTW can not deal with the scene re-ordering situation due to the monotonicity condition. Moreover, its boundary condition requires prior knowledge of the start and end coordinates, which is not a trivial information, in order to compute an optimal path.

3. PROPOSED APPROACH

This section describes the details of the proposed algorithm starting from audio landmark extraction and matching steps inherited from [1]. Following steps are proposed for the considered movie synchronization application. The global workflow of the algorithm is shown in Fig. 1.

3.1. Landmark extraction

The audio landmark feature was originally presented in the well-known Shazam music recognition system [1]. This feature is derived from the spectrogram, a time-frequency (T-F) representation of an audio signal by means of the short time Fourier transform, where only local energy peaks, which have higher energy than all their neighboring T-F points, are indexed by their T-F coordinates. A fixed number of peaks are chosen along each T-F region in order to assure a reasonably uniform distribution and each peak, known as anchor point, is paired with nearby ones within its target zone to form the landmarks. These landmarks are then quantized and packed into the L-bit unsigned integers and stored in the database together with the associated time offset from the beginning of the file for hashing.

Note that by choosing spectral peaks as anchor points, the derived feature is intrinsically robust to noise, and by using pairs of T-F points rather than a single one, the deriving landmark exploits additionally discriminative spectral structure of the audio signal. Although the implementation corresponding to the original paper [1] has not been revealed, we use the Matlab implementation by Dan Ellis [11] for the development of our system.

3.2. Landmark matching and histogramming

At this step, each landmark extracted from the audio track associated with the second version of the movie is used to search in the database containing all landmarks extracted from the audio track associated with the first version to find the match. The detail of this process was described in [1] where deriving hash representation offers a great improvement in search efficiency compared to a linear search. For each matching landmark found, the corresponding time offsets from the beginning of each audio file and their difference are stored. After all landmarks have been scanned, the scatterplot of time offsets corresponding to all matching landmarks is drawn together with the histogram representing the number of matches as a function of the time differences between the two versions. As an example, these pictures are shown by Fig. 2 and Fig. 3, respectively, for two versions of a 3 minute sequence of a movie.

What can be observed from these figures is that when a significant number of matching time pairs appears on a diagonal in Fig. 2, there is a corresponding histogram peak counting for the same difference of time offsets in Fig. 3.

3.3. Thresholding and match filtering

By scanning landmarks for all possible hits, accidental matches usually co-exist together with expected matches. The latter form a diagonal in the scatterplot of time offsets and a corresponding peak in the histogram. In order to filter out these accidental matches, local maximum peaks in the histogram are first specified by comparing the numbers of matching landmarks (values in y-axis in Fig. 3) to a predefined threshold. The threshold value is generally chosen depending on the landmark density, *i.e.* the approximate number of extracted landmarks per second, and the durations of the matching periods between two versions. Then the differences of time offsets between two versions corresponding to these maximum peaks (values in x-axis in Fig. 3) are used to reproduce the corresponding time offsets in two versions. Note that this reverse process is possible because

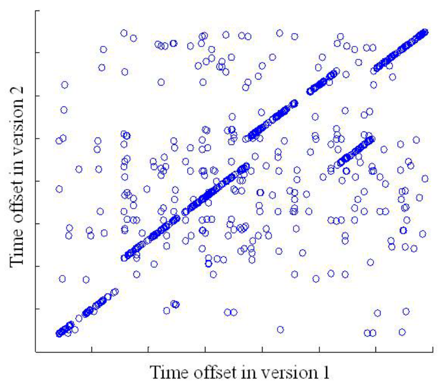

Fig. 2. Scatterplot of all matching landmarks.

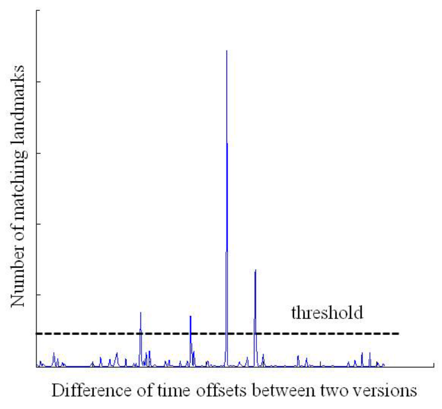

Fig. 3. Histogram of the difference of time offsets.

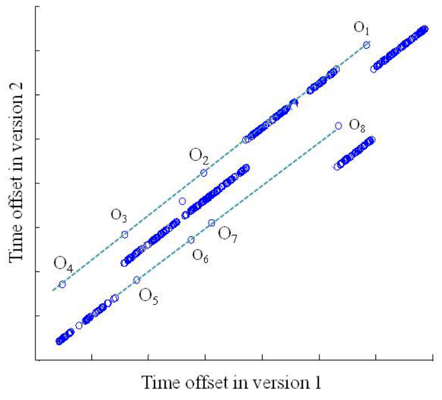

Fig. 4. Scatterplot of filtered matching landmarks.

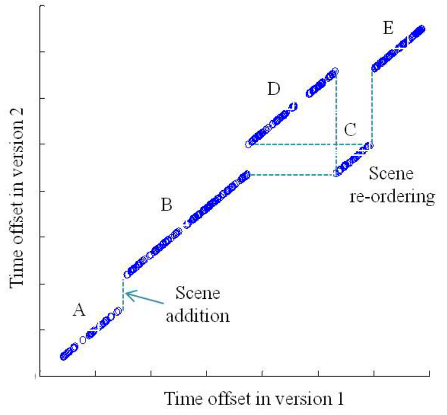

Fig. 5. Scatterplot of final matching landmarks after outlier elimination.

a time offset was stored together with the time difference in the landmark extraction step. Finally a scatterplot accounting for only matching positions lying on considered diagonals is produced. This scaterplot, as shown in Fig. 4, is actually a filtered version of the one shown in Fig. 2 where accidental matches, which do not form a significant diagonal, are eliminated.

It is important to note that at this step Shazam [1] searches for only one maximum peak in the histogram to declare if two signals are similar or not. In our proposed approach, several maximum peaks are normally identified due to the multiple edits along the second version of the movie.

3.4. Outlier elimination

The resulting scatterplot from the previous step may contain outliers, *i.e.* points accidently lying on a diagonal but do not represent the actual temporal match between two versions. These outliers are shown, as an example, by points $O_1, O_2, ..., O_8$ in Fig. 4. Thus, at this stage, they need to be eliminated in order to specify the correct boundary of each resulting temporal matching segment. Additionally, this step will help to specify the valid segments, if there is more than one, in each diagonal.

We propose to use a hierarchical clustering-based approach [12] for this non-trivial work where each point in a diagonal of the filtered scatterplot is first considered as a cluster containing a single item. Euclidean distance between each pair of clusters is computed and the clusters for which their distance is smaller than a pre-defined threshold are merged. This ”bottom up” process is repeated until either the distance between any pair of clusters is larger than the threshold or only one cluster remains. Note that, in our implementation we define the distance between clusters as the minimum distance between their boundary points, *i.e.* two points having the lowest and the highest time offsets in each cluster. Finally, clusters with a small number of points are considered to be outliers and are eliminated. The remaining clusters for every diagonal, *e.g.* as visualized by Fig. 5, will represent the estimated temporal matching between two versions.

In the example of Fig. 5, fives consecutive matching segments A, B, C, D, and E are identified. The gap between A and B along the y-axis indicates that some scenes do not exist in version 1, but are present in version 2. In the same manner, the gaps between B and D, and D and E indicate that some scenes are added to version 2 and removed from version 1, respectively. However these scenes are considered to be matched by segment C meaning that the scenes in version 1 appear at different time offset in version 2. This situation corresponds to the scene re-ordering edits.

4. EXPERIMENTAL RESULTS

In order to evaluate the synchronization performance, due to the lack of groundtruth for real edited versions of the movies, we generated a synthetic dataset from 11 movies on DVD by first extracting an audio track associated with each of them. Then for each movie sound track, we selected 6 sequences of approximately 180 second duration with different genres, *i.e.* action, speech, and music, resulting in a total of 66 reference audio files. For each of these files, we performed 4 different edits: removing a 5, 10, and 15 second slot in the middle of the file, and re-ordering a 10 second slot from a position at a quarter of the file length to a position at three quarters of the file length, resulting in a total of 264 edited audio versions.

Since DTW is a well-known technique for sequence alignment, we use it as a baseline for the comparison with our proposed approach. Our DTW implementation took a vector of 12 Mel-Frequency Cepstral Coefficients (MFCCs), the well-known feature widely used in speech recognition [13], as an audio descriptor. Each MFCC sample is obtained over a 40 ms window with an overlap of 50% between two consecutive ones. In the implementation of the proposed algorithm, we extracted about 30 landmarks per second for the matching and the threshold for the number of matching landmarks described in Fig. 3 was set to 50. At the clustering step, clusters with less than 5 matching points are considered to be outliers and are eliminated.

The synchronization performance is evaluated in terms of *precision*, *i.e.* the fraction of detected synchronization parts that are correct, *recall*, *i.e.* the fraction of correct synchronization parts that are detected, and *F-measure*, *i.e.* the harmonic mean of the precision and recall. We observed that the proposed algorithm offers similar precision for all three edits of scene removal while the DTW algorithm results in lower performance with respect to larger scene removal, *i.e.* its precision is 0.97, 0.94, and 0.92 for 5 s, 10 s, and 15 s edits, respectively. This is because DTW continuity condition brings more false positive at the gap between two diagonals. At another point, both algorithms offer similar result on action, speech, and music data, meaning that audio genres do not significantly affect the synchronization performance. The overall performance, averaged for all genres and all scene removal edits, is shown in Table 1.

Edits	Algorithm	Precision	Recall	F-measure
Scene removal	Proposed	0.98	0.96	0.97
	DTW	0.94	1.0	0.97
Scene re-ordering	Proposed	0.95	0.90	0.92
	DTW	non applicable		

Table 1. Average synchronization performance.

It is not surprising that for the scene removal edits the proposed algorithm offers higher precision, but lower recall than that of the DTW algorithm. This can be explained by the fact that landmark matching offers very low false positive, but it may results in high false negative when landmarks are not extracted for a certain period. On the contrary, DTW finds the matching path frame by frame so it does not bring significant false negative but more false positive usually appears at discontinuity edges. Overall, F-measure are comparable for two methods. For the scene re-ordering edit, the proposed approach results in the average precision, recall, and F-measure of 0.95, 0.90, and 0.92, respectively, while DTW algorithm is non-applicable due to the monotonicity condition. Finally, it is worth mentioning that the standard deviations of all the measures are very small, *e.g.* that of the F-measure are 0.01 and 0.02 for DTW and the proposed approach, respectively, meaning that the result achieved for each edited audio version is very similar to the averaged value presented in Table 1.

5. CONCLUSION

In this paper, we presented an efficient method for frame synchronization of different versions of a media content for a practical application of automatically transferring metadata from one version to the others. The proposed approach relies on audio tracks associated with the videos such that an existing audio landmark matching is successfully applied with substantial extension. Our experimental results indicate that a high level of synchronization accuracy can be achieved when considering practical edits such as scene addition, removal and even re-ordering. Future work will validate the performance of the proposed algorithms over real-world data.

6. ACKNOWLEDGMENT

The authors would like to thank Ashish Bagri for sharing the code of DTW algorithm.

7. REFERENCES

[1] A. L-C. Wang, "An industrial-strength audio search algorithm," in *Proc. Int. Sym. on Music Information Retrieval (ISMIR)*, 2003, pp. 1–4.

[2] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski, "Synchronization of multiple camera videos using audio-visual features," *IEEE Trans. on Multimedia*, vol. 12, no. 1, pp. 79–92, 2010.

[3] F. Thudor, I. Autier, B. Chupeau, F. Lefebvre, and L. Oisel, "Automatic chaptering of VoD content based on DVD content," in *Proc. Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, 2012, pp. 1–6.

[4] N. J. Bryan, P. Smaragdis, and G. J. Mysore, "Clustering and synchronizing multi-camera video via landmark cross-correlation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 2389–2392.

[5] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *Journal of VLSI Signal Processing*, vol. 41, no. 3, pp. 271–284, 2005.

[6] J. Ogle and D. Ellis, "Fingerprinting to identify repeated sound events in long-duration personal audio recordings," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 233–236.

[7] N. Q. K. Duong, C. Howson, and Y. Legallais, "Fast second screen TV synchronization combining audio fingerprint technique and generalized cross correlation," in *Proc. IEEE Int. Conf. on Consumer Electronics - Berlin (ICCE-Berlin)*, 2012, pp. 241–244.

[8] H. Jégou, J. Delhumeau, J. Yuan, G. Gravier, and P. Gros, "Babaz: a large scale audio search system for video copy detection," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 2369–2372.

[9] R. Macrae, X. Anguera, and N. Oliver, "Muvisync: Realtime music video alignment," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2010, pp. 534–539.

[10] X. Anguera, R. Macrae, and N. Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 3582–3585.

[11] D. P. W. Ellis, "Robust landmark-based audio fingerprinting," 2009, *http://labrosa.ee.columbia.edu/matlab/fingerprint/*.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics. Springer, 2009.

[13] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Sym. on Music Information Retrieval (ISMIR)*, 2000.