A scalable framework for joint clustering and synchronizing multi-camera videos

Ashish Bagri, Franck Thudor, Alexey Ozerov, Pierre Hellier

A SCALABLE FRAMEWORK FOR JOINT CLUSTERING AND SYNCHRONIZING MULTI-CAMERA VIDEOS

Ashish BAGRI, Franck THUDOR, Alexey OZEROV, Pierre HELLIER

Technicolor, Rennes, France
https://research.technicolor.com/rennes/

ABSTRACT

This paper describes a method to cluster and synchronize large scale audio-video sequences recorded by multiple users during an event. The proposed method is designed to jointly cluster audio content and synchronize sequences in each cluster to create a multi-view presentation of the event. The method is roughly based on cross-correlation of local audio features. In this paper, three main contributions are presented to obtain a scalable and accurate framework. First, a salient representation of features is used to reduce the computation complexity while maintaining high performance. Second, an intermediate clustering step is introduced to limit the number of comparisons required. Third, a voting approach is proposed to avoid tuning thresholds for cross-correlation. This framework was tested on 164 YouTube concert videos and results demonstrated the efficiency of the method with a correct clustering of 98.8% of the sequences.

Index Terms— Feature extraction, clustering methods, cross-correlation, synchronization, scalability.

1. INTRODUCTION

Growing popularity of portable devices, e.g., smart-phones, is leading to creation of a huge amount of multimedia recordings of the same or different events. For example, a concert of a popular music band may be filmed by hundreds of fans. Such collections could be exploited to enhance the corresponding audio-visual content, create summaries of a particular event, etc. The first challenge for any processing is to cluster sequences belonging to the same events and then synchronize them in time. Here we consider a collection-driven definition of event. Given a collection of video recordings, two recordings are considered belonging to the same *event* if and only if they are recorded in the same place and can be continuously linked in time via other recordings. This link can be understood as a path through successive overlapping recordings.

Synchronizing based solely on the video content is challenging because there could be no similar visual clue between videos (e.g., due to high variation in the point of views). However, the task becomes tractable if one relies on the audio content since there is more similarity to be found in the audio.

In this paper, the problem of joint clustering and synchronization of multimedia sequences is considered and some limitations of current state-of-the-art approaches are addressed.

The paper is organized as follows. The proposed framework is first overviewed and compared with the state-of-the-art in section 2, and then described in details in section 3. Experiments on a realistic database of YouTube videos are presented in section 4, and some conclusions are drawn in section 5.

2. METHOD OVERVIEW AND RELATED WORK

Several works on synchronization using audio content have already been performed [1, 2, 3, 4]. All these works are primarily based on computing cross-correlations between audio fingerprints. Shrestha *et al.* [1] propose a multimodal synchronization approach, using light flashes, audio fingerprints and onsets, which can be understood as "*audio flashes*". They do not address the clustering problem, assuming that all the videos already correspond to the same event. Kennedy *et al.* [2] list different applications that can be envisioned after video clips have been aligned, like clustering, ranking of clusters, selection of highest quality clips. Joint clustering and synchronization is considered in [3, 4]. Both are based on an implementation of audio fingerprints proposed by Wang [5].

In this paper, we propose a system for joint clustering and synchronizing multimedia sequences. The proposed system, schematized in Fig. 1, relies on the following main steps:

- Mel-frequency cepstral coefficients (MFCCs) [6] are first extracted for all recordings. So called *salient MFCC* representation is computed from the MFCCs using dimension-wise maxima over some predefined temporal window.
- Intermediate clustering and rough synchronization is performed on salient MFCCs using newly introduced *cluster representatives* and a novel *voting approach*.
- A precise time alignment is computed within each intermediate cluster and final clustering is performed.

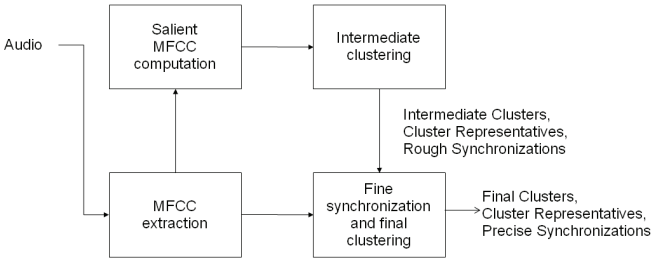

Fig. 1. Overall schema of the proposed framework. Computation of MFCCs and salient MFCCs are followed by an intermediate clustering and a rough synchronization which are then refined.

This time alignment is computed using cross-correlation on dense MFCC features over a reduced window (corresponding to salient MFCC computation window).

We differ from state-of-the-art approaches in the following main points.

Choice of the audio features: We investigate the applicability of MFCC features to this particular task of clustering audio-video clips. Since high accuracy is required for such application, MFCC features should at least be comparable to audio fingerprints. Moreover, as specified in details below, using MFCCs allows us to implement a robust clustering strategy that consists in voting and introducing a new representation, called salient MFCCs, that has a low temporal resolution and thus makes it possible to speed up the approach.

Clustering heuristics and threshold tuning: In state-of-the-art techniques [3, 4], the cross-correlation is computed jointly for all dimensions of the corresponding representation. A threshold on the final cross-correlation value needs tuning to decide whether the two recordings match or not. It also leads to some other heuristics to filter out the high number of false positives because of the thresholding. Bryan *et al.* [3] use various thresholds and tricks to perform final merge of video clips in clusters; this is due to the sparse distribution of fingerprints along with time. The two-step process proposed by Su *et al.* [4] first generates large clusters by using a low threshold for selecting matching clips; these clusters are further refined by maximizing the intra-cluster similarity to inter-cluster similarity ratio. We instead propose a different strategy based on *voting* which only needs a comparison to majority of alignments, i.e. we use a fixed threshold of 50% that is robust to data variation, thus does not need to be adaptive.

Scalability: Current state-of-the-art approaches do not scale well with the size of the dataset since they rely on pair-wise comparison of all the sequences. We overcome this by introducing an intermediate clustering step and cluster representatives which match to all the sequences in that cluster (see Fig. 2). To address large scale issues, it is possible to perform joint clustering and alignment in a bottom-up hierarchical manner by splitting the database in subsets at the lower stages and by comparing only clustering representatives at the higher stages. We demonstrate this in the experiment part, by showing how to iteratively add new recordings to an already processed database. The computational complexity is also reduced by comparing first *salient MFCCs* computed at a $1/10$ th of the normal sampling rate.

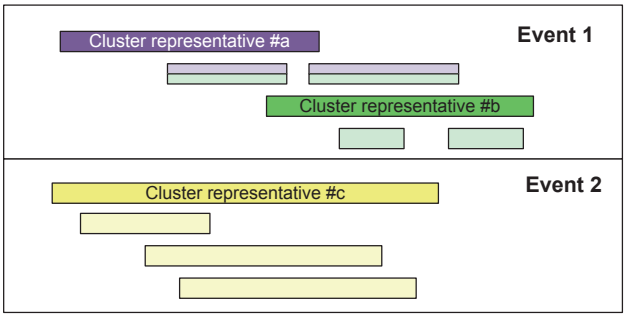

Fig. 2. Illustration of cluster representatives. One or several cluster representatives (that can include the same sequences) are computed for one event so as to link the longest sequences with others. Their use drastically limit the required number of comparisons.

3. METHOD

This section details the method presented in Figure 1.

3.1. Feature extraction

For each audio sequence, the first step is to extract MFCCs [6]. Each sequence is segmented into frames of 40 ms with an overlap of 50% and each frame is converted into 12 decorrelated MFCC coefficients.

3.2. Salient MFCCs

To reduce the number of features describing an audio sequence and limit the complexity, we extract *salient MFCC* values from the dense MFCC original vectors. It is a representation that has only a fraction (10% in this study) of the components of the original MFCC features and is still informative enough to be able to synchronize and match two audio files. To compute the salient MFCCs, we only retain dimension-wise MFCC maxima over a sliding window of W_s MFCC frames and an overlap of θ %. We discuss these values in section 4.2.

3.3. Comparison of two sequences

The comparison between two sequences is carried out by computing the cross-correlation on the feature values. The

features used in sections 3.4 and 3.5 below are, respectively, the salient and the dense MFCCs. Cross-correlation is performed individually on each coefficient. Cross-correlation is an effective way to find the time offset between two signals. The peak of the cross-correlation indicates the time offset between the two signals but a threshold is required to determine whether these two signals match. Threshold tuning is a non trivial task and an inappropriate threshold choice can alter the performance drastically [3, 4].

To avoid the issue of threshold tuning, a novel voting approach is used instead. With our method, each of the 12 dimensions of MFCC leads to an estimated time offset. A match is thus considered as true if a majority (> 6) of estimated time offsets are in agreement. Therefore, no threshold tuning is needed any more, which increases the reliability in selecting matching sequences.

3.4. Intermediate clustering

A first level clustering is performed to group the set of sequences which have a common overlapping segment. When dealing with large datasets, it becomes quickly infeasible to compare all the sequences pair-wise. Therefore, we propose to create intermediate clusters by means of cluster representatives. Cluster representatives are created by matching the longest sequences with others, as described below.

To form these intermediate clusters, all sequences are sorted according to their lengths. The longest video is made the cluster representative of the first cluster. Subsequent sequences are compared to the existing cluster representatives. If a sequence has an overlapping segment with an existing cluster representative, the sequence is added to that cluster. A new cluster is formed if a sequence does not match with any existing representative. Note that one sequence can be correlated to several cluster representatives, hence can be part of many intermediate clusters. The above comparisons are performed in the salient MFCC domain and are based on cross-correlations as described in section 3.3.

3.5. Fine synchronization and final clustering

A pair wise comparison is done between all sequences belonging to the same intermediate cluster to find precise alignment between them. It is done using the dense MFCC features over a small range of window which was given by the salient comparison.

Each sequence in a cluster is only compared to all other sequences of the same cluster, as the non overlapping sequences have already been separated in the step described in section 3.4. This reduces drastically the number of dense comparisons to be performed. A complete match-list with time offset in seconds between the matching sequences is generated. The final clustering consists in categorizing sequences into events (Figure 2). Sequences which have an overlapping segment form part of the same event. Sequences which do not overlap but are connected via a common sequence also form part of the same event. Hence one event can be represented by several representatives.

3.6. Scalability

In sections 3.4 and 3.5, a method was proposed to reduce the number of comparisons required for clustering. Our framework benefits from this feature even when new videos are added to the database of events that have already been clustered and synchronized. Normally, new videos would have to be compared to all existing videos to determine the potential matches, within our framework this can be avoided as described in section 3.4.

Apart from storing final alignments as discussed in 3.5 as the final output, intermediate clusters created in the step 3.4 are stored together with the corresponding representatives. Hence, a new video needs only be compared to the stored cluster representatives. After the initial clustering, the same steps of section 3.5 are performed to get the precise alignment and classify them into events. This method can handle efficiently incremental addition to the database, paving the way for a scalable and distributed system. It can also handle large amounts of data as the number of comparisons are reduced to a bare minimum and the use of salient features speeds up the comparison.

3.7. Complexity analysis

To create a match list for K sequences, normally the number of comparisons needed is $K(K-1)/2$. Each comparison using FFT based cross-correlation is of order $O(NLogN)$, leading to a complexity $C_{baseline}$:

$$C_{baseline} = (K(K-1)/2)N\log(N)$$

where N is the average number of MFCCs per sequence. The same process based on salient MFCCs would exhibit a complexity $C_{salient}$:

$$C_{salient} = (K(K-1)/2)N_s\log(N_s)$$

where N_s is the average number of salient MFCCs per sequence. The reduction in the complexity is of the order N_s/N ($1/10^{th}$ in our case). The reduction is even more as we do not do all the comparisons as discussed in sections 3.4 and 3.5. So we separate the complexity formula into two parts. The first part deals with the salient MFCCs used for clustering while the second part deals with dense MFCCs dealing with the fine synchronization

$$C_{proposed} = L_{salient}N_s\log(N_s) + L_{fine}N\log(W_s)$$

where $L_{salient}$ and L_{fine} are respectively the number of comparisons performed by our algorithm at the salient and fine level. These quantities are difficult to express by formulas,

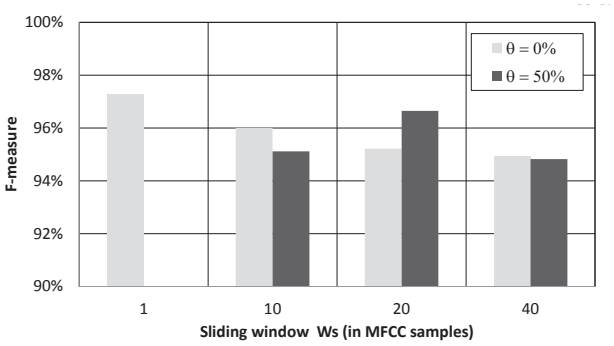

Fig. 3. Influence of parameters on salient representation performance.

since they strongly depend on the database structure. A comparison on complexity based on the number of actual computations for our dataset is drawn in table 2.

4. EXPERIMENTS

4.1. Dataset

The dataset consists of user contributed videos posted on YouTube. 164 videos from 6 different artists/bands, captured during 6 different concerts, and having a cumulative duration of 17.56 hours were used. The longest sequence was 21 mins long while the shortest was 44 seconds long. Table 1 includes more details on this database. A ground truth of 36 clusters was manually created. The smallest cluster contains 1 sequence, the biggest 29 sequences, and the average cluster size is 4.5 sequences. From this ground truth, a binary matrix of size 164×164 is generated, where one/zero codes indicate respectively matching/non-matching sequences. This matrix is called GT matching matrix in the rest of the document.

4.2. Evaluation

First, the robustness of salient representation is evaluated for comparing sequences. An exhaustive comparison of all sequences in the dataset was performed and results assessed with the GT matching matrix using the F-measure [7]. Results shown in Figure 3 demonstrate the accuracy of the method with a mean F-measure above 95% and about the same as for dense MFCCs ($W_s = 1$). In addition, it can be noticed that results do not vary much with parameters W_s and θ. In the rest of the experiments, the setup $W_s = 20$ and $\theta = 50\%$ was chosen.

Secondly, the obtained clustering is compared to the 36 clusters of the ground truth. All but one are found to be correctly clustered. The missed one is a cluster composed of two songs, which is wrongly merged with a five-song cluster captured during the same concert. Interestingly, the two songs are correctly synchronized together, but the analysis of the audio files showed that one of them exhibits a very low SNR, leading to a mismatch with one of the representatives of the other cluster. Considering the sequence clustering rate as the number of sequences correctly clustered with respect to the whole database, the achieved clustering performance is 98.8% $(= 162/164)$.

	Baseline	Salient	Proposed
Salient MFCC	No	Yes	Yes
Representatives	No	No	Yes
Complexity	100%	3.8%	2.6%

Table 2. Complexity comparison w.r.t baseline ($C_{baseline}$) on our dataset of 164 videos.

Fine synchronization performance was not measured in absence of ground truth. However, a manual check was performed "a-posteriori" for each cluster by listening to all cluster's elements synchronized. All the sequences of all ground truth were found to be correctly synchronized. Moreover, we believe that for this task it is in the most cases simply impossible to define a precise synchronization ground truth. Indeed, audio recordings are often mixtures of different sources that, being located in different places, are recorded with different delays by different devices [8]. Thus, a "perfect synchronization" is not unique, since it depends on the source one would like to focus on.

The number of computations performed by our algorithm at the salient and fine level were also gathered, and the formulas of section 3.7 applied to compute the complexity. Results shown in table 2 show that our approach allows a huge decrease in complexity.

4.3. Scalability

Experiments were carried out to test the capability of the method to deal efficiently with additions of new videos into an existing database as discussed in section 3.6. For this purpose, the entire database of 164 video was split into two parts: a primary part and a secondary part. The primary part is clustered and aligned, while the secondary part is incrementally added to the clusters created using the update mechanism presented in 3.6. Several split configurations were tested: $\{120 + 44; 100 + 64; 90 + 74; 84 + 80\}$. For each configuration, multiple iterations were run with random initialization leading to a total of 175 tests. Precision, recall and F-measure [7] of the final match list for each of the 175 tests were computed relying on the GT matching matrix as in section 4.2. Results summarized in table 3 indicate clearly that the method handles efficiently the addition of new videos. The clustering performance is not affected by the split configurations.

It must be stressed out that these tests were done to demonstrate the ability of our framework to incrementally add sequences to an existing clustered database while maintaining the same performance as we would obtain by using our algorithm on all the sequences together. This property

Artist / Band	Concert	Nb. of videos	Nb. of songs (events)
Muse	Paris, Stade de France - June 2010	84	19
Adele	London, Royal Albert Hall - September 2011	22	1
Arcade Fire	Berkeley - June 2007	15	7
Iron Maiden	Singapore - Feburary 2011	31	6
Lady Gaga	NewYork - New Year's eve 2012	7	1
Hubert-Félix Thiéfaine	Paris, Bercy - October 2006	5	2
	total	164	36

Table 1. Database description.

makes our approach scalable and provides us with a platform to carry out the clustering and synchronization in a distributed manner. This is particularly important if the large number of existing videos (for example from YouTube) has to be clustered in a seamless way.

5. CONCLUSIONS

We have introduced a framework for joint clustering and synchronizing multi-camera videos. It is based on a light but still discriminative fraction of the dense MFCCs that we named "salient MFCCs", allowing a reduction of time when comparing sequences. We also introduced the notion of "cluster representative" which also limit the number of comparison to be performed. A novel voting approach based on the individual result of each MFCC coefficient's cross-correlation avoids the definition of any threshold for sequence matching decision. The rough synchronization obtained on the salient MFCCs at the first stage is refined at the second stage; sequences belonging to the same cluster are precisely synchronized using the dense MFCC features. We have shown that the proposed framework drastically reduces the complexity needed to cluster and synchronize sequences, while maintaining a high level of accuracy. We have also shown that the process is scalable and that the algorithm retains its benefits even when the clustering is performed in a distributed manner. Finally, to our best knowledge, this is the first time the MFCC features were used for joint clustering and synchronizing multi-camera videos by audio. Given the high clustering performance achieved (only two out of 164 sequences were miss-clustered), we can conclude that the performance of MFCCs within such an application should be at least comparable to that of audio fingerprints.

Configuration	164	120 +44	100 +64	90 +74	84 +80
Precision (%)	99.87	99.21	98.82	98.80	98.67
Recall (%)	93.63	93.16	93.51	93.81	94.13
F-measure (%)	96.65	96.09	96.09	96.23	96.34

Table 3. Average precision, recall and F-measure over the 175 tests with different split configuration.

6. ACKNOWLEDGEMENTS

This work was partially supported by European projects AXES and Quaero.

7. REFERENCES

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

[2] L. Kennedy and M. Naaman, "Less talk, more rock: automated organization of community-contributed collections of concert videos," in *Proceedings of the 18th international conference on World wide web (WWW)*, 2009, pp. 311–320.

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

[4] K. Su, M. Naaman, A. Gurjar, M. Patel, and D. P. W. Ellis, "Making a scene: alignment of complete sets of clips based on pairwise audio match," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR)*, 2012, pp. 26:1–26:8.

[5] A. Wang, "An industrial strength audio search algorithm," in *International Conference on Music Information Retrieval (ISMIR)*, 2003, vol. 2.

[6] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[7] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.

[8] C. Blandin, A. Ozerov, and E. Vincent, "Multi-source TDOA estimation in reverberant audio using angular spectra and clustering," *Signal Processing*, vol. 92, pp. 1950–1960, 2012.