Browsing Videos by Automatically Detected Audio Events

Virgínia Barbosa*, Thomas Pellegrini†, Miguel Bugalho† and Isabel Trancoso*†
*IST/UTL
Avenida Rovisco Pais, 1
Lisboa, 1049-001
Email: virginia.barbosa@ist.utl.pt
†INESC-ID
Rua Alves Redol, 9
Lisboa, 1000-029
Email: thomas@l2f.inesc-id.pt

***Abstract*—This paper focuses on Audio Event Detection (AED), a research area which aims to substantially enhance the access to audio in multimedia content. With the ever-growing quantity of multimedia documents uploaded on the Web, automatic description of the audio content of videos can provide very useful information, to index, archive and search multimedia documents. Preliminary experiments with a sound effects corpus showed good results for training models. However, the performance on the real data test set, where there are overlapping audio events and continuous background noise is lower. This paper describes the AED framework and methodologies used to build 6 Audio Event detectors, based on statistical machine learning tools (Support Vector Machines). The detectors showed some promising improvements achieved by adding background noises to the training data, comprised of clean sound effects that are quite different from the real audio events in *real life* videos and movies. A graphical interface prototype is also presented, that allows browsing a movie by its content and provides an audio event description with time codes.**

I. Introduction

More and more multimedia content is uploaded on the Internet. Take for instance video sharing websites like YouTube, they make it possible for anyone with an Internet connection to upload a video that could be watched worldwide within a few minutes. In fact, every minute, 24 hours of video is uploaded to YouTube[1]. Video has become one of the most popular forms of communication over the Web. Consequently, it has become increasingly difficult to make a simple content search. The grand majority of video search engines are based on textual tags inserted by the users. For example, if we wanted to see a live chicken on YouTube, we would insert the word "chicken" on the search field, but the first results would be food, cartoons, people impersonating a chicken, etc. Audio content retrieval could improve greatly the semantic video search. Yet, there are other application areas. With audio content retrieval it is possible to obtain an audio description of a movie. This description provides time segments with music and all kinds of sounds (e.g. gun shots, explosions, singing birds, cars passing-by, etc). This information can be very helpful to hearing impaired people, enabling them to follow the synopsis better or to be able to browse the movie in terms of audio events, through a graphical interface. As for blind people, an extended audio content description is of great value, because much more details from the scenes are available in audio form. With the ever-growing multimedia content available on the Web, manual annotation is unfeasible. With automatic audio content description, people will be able to access multimedia content more easily.

Audio Event Detection is a recent topic in our research group, it began with the European project VIDIVIDEO, which ended in January 2010. The goal of the project was to enhance the performance of video search engines, by providing true content-based search, as opposed to keyword-based access (e.g. YouTube textual tags) current search engines rely on. During the project more than 60 detectors were built, for different audio events. For a complete technical description of our systems and results achieved during the VIDIVIDEO project, the reader may refer to [1]–[3].

Audio Event Detection and Classification experiments reported in this work, concerns non-speech audio events only. A sound effects corpus was used to train the detectors. Sound effects are artificially created sounds, usually in optimal recording conditions, or sounds that are subject to some enhancing process, which make them very clean. The corpus showed good results on validation sets, comprised of sound effects, but this was not the case with real data test set [1]–[3]. The real data test set contains movies, documentaries, talk shows and broadcast news shows. The audio in these test files is likely to have background noise, many audio events occur simultaneously and recording conditions are not optimal. This gap between the train/validation sets and *real life* videos and movies test sets may explain the significant difference in performance. In order to improve the detection of audio events and increase the performance of the classifiers, one of the main objectives of this work was to overcome the data mismatch. The sound effects corpus was enriched with background noise, and 6 new detectors were trained and tested with the new modified corpus. The detectors were built using Support Vector

[1]http://www.youtube.com/t/fact_sheet

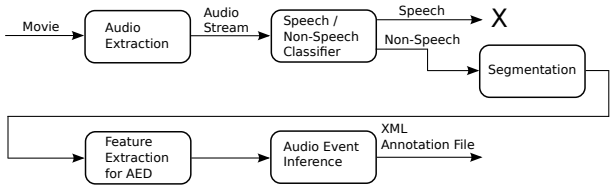

Fig. 1. Block diagram of the AED processing chain.

Machines (SVMs), a statistical machine learning tool.

This paper starts with a brief description of the Audio Event Detection (AED) framework, in Section II. Classification experiments are presented in Section III. Section IV describes a graphical interface prototype that allows browsing a movie by its content and presents a non-speech audio event description.

II. Audio Event Detection

Typical Audio Event Detection (AED) frameworks are characterized by two main modules: feature extraction and audio event inference. The feature extraction module extracts different types of features, some of these features are common in Automatic Speech Recognition (ASR), others borrowed from Music Information Retrieval (MIR). The audio inference module concerns the final detection of the audio events, which can be achieved using different machine learning methods [1], [2].

Different techniques are used and in the literature there is no optimal solution, because many times they focus on specific cases. They work with small domains, or with a limited set of events and background sounds. Also they have a controlled quality of audio. CLEAR Evaluation focused on event detection during meetings [4]. The European CUIDADO project concerns musical instruments [5]. Other studies concern only sports game [6]. The SOLAR project is a system that focuses on a problem similar to ours [7]. Our study deals with a more complex audio domain, such as movies or any kind of video. The domain is very wide, several Audio Events occur simultaneously, background noise often blurs the Audio Events of interest, etc.

The Audio Event Detection can be viewed as a multi-class classification. Our approach consisted in using binary Support Vector Machines (SVMs) classifiers, one per Audio Event. The SVMs are one-against-all detectors, and for each target audio event a classifier is built that distinguishes the event from the other events. This approach has the advantage that adding a new Audio Event is straightforward. A new detector can be trained without having to retrain the other existing detectors.

A. Audio Event Detection Module Structure

This section provides an overview of the AED framework. From the movie video file to the annotation file with the final classification. The AED module structure is illustrated in Figure 1. The diagram represents the processing chain of a movie, starting with any video or movie. Five major modules comprise the chain.

First the audio signal is retrieved from the video file using the FFMPEG[2] software. Therefore any format supported by FFMPEG is a valid input, such as MPEG2 (.mpeg, .mpg, .avi).

In the Speech/Non-speech (SNS) Classifier module, an in-house SNS detector is applied to separate speech from other events.

The segmentation module comes next, where an Acoustic Change Detector (ACD) determines the audio event limits in time. The ACD is used to segment the audio stream into acoustically homogeneous segments. The segmentation results are used later in the process, after the audio event inference module. An average confidence of all frames in the segment is used to compute the segment classification.

There is a great number of features that can be extracted from an audio signal. Using large numbers of features would be computationally expensive and would slow down the convergence of the classification algorithms. We used some features commonly used in Automatic Speech Recognition (Mel-Frequency Cepstral Coefficients), and many other low-level audio descriptors, such as Zero Crossing Rate (ZCR) and Audio Spectrum Envelope (ASE) for example [3]. In total, vectors of 150 features were extracted every 20ms for 40ms windows.

In the audio event inference module, various machine learning methods can be used to provide a final classification of the audio events. The Audio Event Detection is frame-based, but the ACD module provides time segments where only one Audio Event is supposed to happen.

Finally, the information from each frame is combined to generate a final XML annotation file for each concept. This file follows the format described in the MPEG7 [8] standard schema, and contains information about time intervals of the segments provided by ACD module. Confidence measures are estimated considering the number of frames a segment contains.

III. Classification Experiments

A. Corpora

In this study, the corpora available was comprised of movies and documentaries, talk shows and news telecasts. To train statistical detectors, one needs manually annotated multimedia files at event-level. Manual labeling audio events is a very expensive and time-consuming task. More so if the task demands the identification of a large amount of audio events. Also, audio events often occur simultaneously, or with speech or music, and sometimes it is difficult to perceive an event.

To overcome these issues, a large corpus of sound effects was adopted, for both AED train and validation sets [2]. Conventionally, each sound effect file normally has only one type of sound, hence being individually labeled.

The results obtained with the sound effects corpus on the validation sets were good. However on the test set (*real life* videos, such as movies, talk shows, etc), the results decreased. *Real life* videos and movies often have background noise and

[2]http://www.ffmpeg.org/

TABLE I
LIST OF BACKGROUND NOISE AUDIO FILES.

Noise
Beach
City Public Space
Classroom
Eating Place
Entry-Room
Factory
Forest Water
Forest Wind
Library
Office
Outdoor
Shopping Mall
Traffic
Transit-Station

different audio events occur simultaneously, as opposed to sound effects that are very clear and distinct sounds. The difference in performance on the train/validation sets and *real life* videos lies in the gap between sets. In order to leverage the data mismatch and improve the results, background noise was added to the sound effects corpus. A few sound effects concepts were selected as background noise. Table I lists the background noise concepts. A compatibility analysis (event-noise) was performed, to guarantee that only appropriate sounds were merged. It would not make much sense to have Elephant sounds in a Shopping Mall ambience. This kind of situations were accounted for in the compatibility analysis. A random segment from a random background file was chosen to be mixed. This procedure was executed for every file on the data set.

In this work, experiments for 6 Audio Event Detectors are reported: Birds, Crowd Applause, Dog Barking, Sirens, Telephone Ringing Bell and Water. These events were selected among the set of about 60 events for which detectors were trained, conforming the events supposed to happen in our test set.

B. Support Vector Machines Classifiers

Support Vector Machines (SVMs) were built using the LIB-SVM toolkit to create a baseline[3]. SVMs are discriminative classifiers that need both positive and negative training examples. Each audio event has a minimum amount of examples [9], [10], needed to properly train a model. For each Audio Event, sound effects representative of the Audio Events to be modeled are used as positive examples. Some examples of other Audio Events were used as negative sets.

C. Audio Event Detection Results

The results obtained in the *real life* test set for the presented training methods are shown in Table II. For 6 Audio Events, two models were trained. One with the clean sound effects corpus, and another with the new extended corpus with background noise. The results presented are frame-based. The

[3]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

TABLE II
RESULTS FOR MODELS TRAINED WITH AND WITHOUT BACKGROUND NOISE.

Concept	Test file	Precision		Recall		F-measure	
		Clean	Noise	Clean	Noise	Clean	Noise
Birds	Castelli	0.15	0.09	1.00	1.00	0.26	0.16
	Kosovosodo	0.01	0.01	0.01	0.01	0.01	0.01
	Populonia	0.55	0.44	0.95	0.96	0.70	0.60
	Total	0.29	0.18	0.36	0.37	0.32	0.25
Applause	Portugal-Coracao	0.93	0.96	0.23	0.12	0.37	0.33
	Preco-Certo	0.74	0.92	0.35	0.23	0.47	0.37
	Total	0.78	0.94	0.31	0.22	0.44	0.35
Dog	Corazon Batida	0.88	0.86	0.16	0.43	0.27	0.57
	Monterias	0.69	0.67	0.10	0.29	0.18	0.40
	Total	0.76	0.75	0.12	0.34	0.21	0.47
Sirens	007	0.00	0.04	0.00	0.12	0.00	0.06
	Die Hard 4	0.00	0.03	0.01	0.09	0.00	0.04
	Telejornal	0.65	0.63	0.37	0.34	0.47	0.44
	Total	0.23	0.18	0.24	0.26	0.24	0.22
Telephone	The Aviator	0.32	0.39	0.76	0.79	0.45	0.53
	The Matrix	0.27	0.32	0.84	0.70	0.40	0.44
	Total	0.28	0.34	0.82	0.72	0.41	0.46
Water	Paesaggio	0.09	0.09	0.91	1.00	0.17	0.16
Total	Total	0.39	0.41	0.25	0.32	0.31	0.36

TABLE III
AED COMPUTATIONAL TIME. RT: REAL TIME.

	Audio Extraction	Feature Extraction	Audio Event Detection	Total
Computational time	0.02 RT	0.05 RT	4.25 RT	4.32 RT

performance of the classifiers was assessed in terms of the F-measure, a measure that combines Precision and Recall:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

The higher the F-measure, the better the classification.

For the Dog Barking audio event, for example, adding background noise led to gain in F-measure of about a factor two. Dog Barking videos are those with more background noise content (wind and other animal sounds, step sounds, people talking, gun shots). This may explain the significant gain for this particular audio event. For a given concept, some detectors showed good or low performance depending on the movie; this may be explained by the diversity of the sounds in these movies. On the whole, results show a slight improvement when background noise is added to the training set.

In order to know how long the whole AED process lasts, the computational time was measured. Audio Extraction, Feature Extraction and Audio Event Detection (for 6 Audio Event models) are the three main tasks of the process. Table III shows the computational time for each task, and the total time. The Audio Event Detection stage is the longest. It can take more or less time depending on how many Audio Events are to be detected. It is possible to conclude that for one minute of video, the whole process, from the extraction of audio to the final classification, lasts 4.32 minutes. Usually a movie lasts 90/100 minutes in general, so it takes 1 hour per Audio Event on average to perform AED.

IV. Graphical Interface Prototype

The semantic video search is possible due to the information obtained via the Audio Event Detection techniques described in the above sections. It would be interesting to see the results accomplished so far in a practical and user-friendly manner.

To achieve this goal, a graphical interface prototype was implemented, specifically designed to browse videos semantically using its audio track. Figure 2 shows the screen capture of the interface. The interface presented here is a web application which provides an audio content description of a movie. The user can choose from a catalog of movies available and semantically browse the movie in terms of non-speech Audio Events. By choosing a specific event, the user can move forward or backward through the different segments of the film, accordingly to the classification. For each movie, a list of existing Audio Events and respective time codes is available. These audio events are indexed by the confidence measures also, thus allowing the user to browse best-detected Audio Events. When the user selects one of the time code buttons on the right, the video jumps to that specific time.

There is a preprocessing stage of each movie file, before it can be loaded on the application. It consists of running the Audio Event Detection software on the movie file, so there is information about the audio events on the database. The XML file outputted by the AED module indexes each audio event with the time codes of the movie. It also includes the confidence measures of each audio event.

The web page was coded in Hypertext Markup Language (HTML). The interface has dynamic content that depends on the user's choices (e.g. choosing the audio event to listen, choosing the movie). The dynamic content is be generated using PHP and JavaScript. PHP was used to retrieve information from a database where XML annotation files are stored. To create a dynamic website jQuery[4], a JavaScript library, was used. Flowplayer[5] is an an Open Source project, a video player used to embed video streams into the web page. Flowplayer also allows to build a customized player, as it includes high level of customization possibilities, through a JavaScript API.

V. Conclusion & Future Work

The AED experiments conducted with the SVMs classifiers show that the results for the clean sound effects data set are, much better than for the *real life* data set where recording conditions are different, multiple audio events may be combined, background noise is often present in audio. The gap between sound effects data and real life data was somehow reduced with the addition of background noise to the sound effects corpus. Thus the advantage of using implicitly labeled corpora still holds, sparing the time-consuming task of manually labeling, but with the addition of background noise the classifiers became more robust.

A graphical interface prototype was presented, to allow users to search videos in terms of automatically detected audio events. It is also a user-friendly manner to show the results of the experiments developed.

As future work, further functionalities will be added to the application. The user will be able to upload short videos, and to select which Audio Events he is interested in. The AED software will run remotely, and the video will be integrated into the video database of the application, thus allowing the user to view the results and browse his own video.

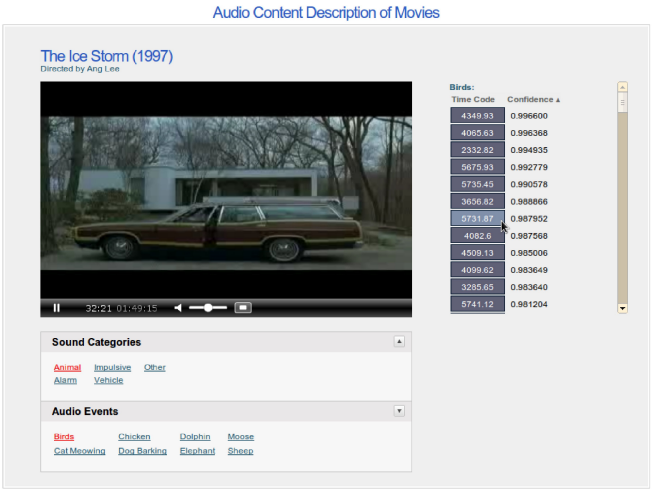

Fig. 2. Graphical Interface Prototype.

[4]http://jquery.com/
[5]http://flowplayer.org/

References

[1] J. Portêlo, and M. Bugalho, and I. Trancoso, and J. Neto, and A. Abad, and A. Serralheiro, *Non-speech audio event detection*, Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing-Volume 00, 1973–1976, IEEE Computer Society, 2009.

[2] I. Trancoso, and J. Portêlo, and M. Bugalho, and J. Neto, and A. Serralheiro, *Training audio events detectors with a sound effects corpus*, Brisbane, September 2008.

[3] M. Bugalho, and J. Portêlo, and I. Trancoso, and T. Pellegrini, and A. Abad, *Detecting audio events for semantic video search*, InterSpeech, Brighton, 2009.

[4] A. Temko, and R. Malkin, and C. Zieger, and D. Macho, and C. Nadeu, and M. Omologo, *CLEAR evaluation of acoustic event detection and classification systems* Multimodal Technologies for Perception of Humans, pages 311–322, Springer, 2007

[5] H. Vinet, and P. Herrera, and F. Pachet, *The CUIDADO Project*, Proc. International Symposium on Music Information Retrieval, Citeseer, 2002

[6] Q. Huang, and S. Cox, *Hierarchical language modeling for audio events detection in a sports game*, Acoustics Speech and Signal Processing (ICASSP) IEEE International Conference, pages 2286–2289, 2010

[7] D. Hoiem, and Y. Ke, and R. Sukthankar, *SOLAR: sound object localization and retrieval in complex audio environments*, Acoustics, Speech, and Signal Processing, Proceedings.(ICASSP'05) IEEE International Conference on Volume 5, 2005

[8] J.M. Martínez, and R. Koenen, and F. Pereira, *MPEG-7: the generic multimedia content description standard, part 1*, IEEE multimedia, pages 78–87, IEEE Computer Society, 2002.

[9] W.-T. Chu, and W.-H. Cheng, and J.-L. Wu and J. Yung-jen Hsu, *A study of semantic context detection by using SVM and GMM approaches*, Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume 3, pages 1591–1594, June, 2004.

[10] G. Guo, and S.Z. Li, *Content-based audio classification and retrieval by support vector machines*, IEEE Transactions on Neural Networks, Volume 14, pages 209–215, 2003.