Audio Event Detection in Movies using Multiple Audio Words and Contextual Bayesian Networks

Cédric Penet, Claire-Hélène Demarty, Guillaume Gravier, Patrick Gros

Audio Event Detection in Movies using Multiple Audio Words and Contextual Bayesian Networks

Cédric Penet*‡, Claire-Hélène Demarty*, Guillaume Gravier† and Patrick Gros‡
*Technicolor R&D France, 975, avenue des Champs Blancs, 35576 Cesson Sévigné, France
†IRISA/CNRS, Campus de Beaulieu, 35042 Rennes, France
‡INRIA, Campus de Beaulieu, 35042 Rennes, France

***Abstract*—This article investigates a novel use of the well-known audio words representations to detect specific audio events, namely gunshots and explosions, in order to get more robustness towards soundtrack variability in Hollywood movies. An audio stream is processed as a sequence of stationary segments. Each segment is described by one or several audio words obtained by applying product quantization to standard features. Such a representation using *multiple audio words* constructed via *product quantisation* is one of the novelties described in this work. Based on this representation, Bayesian networks are used to exploit the contextual information in order to detect audio events. Experiments are performed on a comprehensive set of 15 movies, made publicly available. Results are comparable to the state of the art results obtained on the same dataset but show increased robustness to decision thresholds, however limiting the range of possible operating points in some conditions. Late fusion provides a solution to this issue.**

I. Introduction

Audio event detection is a task with growing interest in the multimedia community, which consists in recognizing specific events within an audio stream. There exists a wide variety of audio events one may want to look for, such as door slams, speech, music, bird sounds or plane sounds. Furthermore, there exists a huge variety of audio content, e.g., TV news, radio broadcast, phone calls recordings, etc. Each type of content usually comes with its own particular conditions: YouTube videos, clear/noisy recordings, movies provide varying conditions in which one looks for a particular sound, or event. Different approaches are therefore required to take this variety into account.

In this context, sounds coming from movies have received very limited attention. Movies are a very particular type of edited audio streams, and there are numerous reasons why they have not been intensively studied yet. First, annotation is particularly difficult due to special effects and high variability. Second, audio is very complex: Several events usually happen at the same time. For example, speech is often mixed with environmental sounds and music, and this may change according to the action, or even during a same action. This makes the task very difficult. Finally, each audio stream is elaborated using special effects such that it corresponds to a message the director wants to send. We believe this brings a lot of variability between the different movies, although special audio effects usually come from common sound effects databases.

Only few works take the inter-movie variability into account. On the one hand, Giannakopoulos *et al.* [1] are interested in detecting events like music, speech, shots, fight sounds and screams. They obtain good results, but this is probably due to their cross validation technique. They split each film in parts and choose randomly which parts are used for training or for testing. As a consequence, most films appear both in the training and the testing sets and the results are biased. On the other hand, Schlüter *et al.* [2], referred as the ARF team in the following, provide results using a leave-one-movie-out cross-validation that confirm the inter movie variability for the detection of gunshots and explosions. In this work, ARF builds concept detectors for the MediaEval Affect Task [3], which provides annotations for some audio and video concepts on 15 movies. Trancoso *et. al.* [4] also tend to corroborate this hypothesis. They built concept detectors using a sound effects database, and tested them on a few movies. While their results on the sound effects database are good, comparable to the results reported by Giannakopoulos *et al.* [1], the results on movies correspond to the results obtained by the ARF team in terms of performance, therefore corroborating the high variability of movies content. Previous works are all based on a fixed segmentation of the audio stream, using 0.5 to 1 second segments. They aggregated classic audio features such as MFCC, or spectrum based features, on the audio segments using statistics and used static classifiers such as K nearest neighbors, support vector machines (SVM) or multilayer perceptrons. The use of classic low-level features, even aggregated on a fixed segmentation, may be one explanation for the lack of robustness towards variability that was observed.

In order to reduce the variability carried out by the standard features, we propose to use the well-known concept of audio words in a novel scheme. This concept comes from the text theory: The idea is to find canonical words between documents in order to reduce the variability between the different documents. For example, the words "called" and "calling" may both be represented by the canonical word "call". Applied to audio theory, the idea is to group together the audio samples from different audio files that are similar, and to represent the audio stream as a sequence of symbols called words by analogy. Although the concept of audio words is not novel, the novelty of our approach lies in the dictionary learning method used and in the use we make of the audio words sequence. On top of that, we propose to use a data-driven segmentation to be as precise as possible in the detection of the events we are

This work was partly achieved as part of the Quaero Program, funded by OSEO, French State agency for innovation.

We would like to acknowledge the MediaEval Multimedia Benchmark http://www.multimediaeval.org/ and in particular the Affect Task 2011 for providing the data used in this research.

interested in, and to use contextual Bayesian networks (BN) to classify audio samples according to their context as in [5]. If two samples have the same representation but belong to different classes, the use of their context is bound to help the classifier disambiguate classes.

We propose to apply such a scheme to the detection of "violent audio events" such as gunshots or explosions. This type of events has been shown to be of particular interest in the literature for various practical applications such as violence detection in movies for automatic ratings or parental control [1] or automatic alarms raising in surveillance videos [6].

In the following, we present our novel approach for audio event detection in movies. Experiments are performed on the set of annotated movies available in the framework of the MediaEval 2012 Affect Task and results are compared to those of [2] for the event classes gunshots and explosions. The paper is organized as follows. In Section II, we present some background on audio words and the works that inspired us. The theoretical description of the approach proposed is detailed in Section III. Experiments and results are described in Section IV.

II. Background on audio words

Audio words were recently introduced but, to the best of our knowledge, their use in the context of audio event detection in movie soundtracks has never been reported. We describe seminal works on audio words that serve as a basis to our system.

The main difficulty in text, video or audio words techniques is to find the canonical words that are to be used, i.e., to build a dictionary or codebook of canonical words. This usually consists in clustering the feature space. For instance, in [7], [8], Chin and Burred propose a system for discovering audio patterns in an audio stream. They extract MFCC features and explore three ways to learn a dictionary: non-negative matrix factorization (NMF), PCA and k-means clustering. As a result of quantization, the signal is represented as a sequence of indexes in the dictionary from which patterns are derived. Traditionally, each audio segment is assigned one dictionary element, however it has been proposed in [8] to associate the k-nearest dictionary elements to each audio segment. The underlying idea is that two samples can be represented by the same audio word without being equivalent. Therefore, considering the second and the third closest words adds some complementary information. However, the work presented in [8] remains preliminary with an evaluation limited to short synthetic sounds or simple recordings.

The approach followed in [9] is the closest to ours. In this article, the 2011 TRECVID Multimedia Event Detection (MED) data [10] is used to characterize user generated video excerpts coming from Internet and to detect audio events for which annotations are provided. The dictionary is built based on an iterative HMM and N-gram process, which takes temporality into account. Random forest classifiers are used on histograms of audio words from $\simeq$10 seconds audio segments with 75 % overlap. Good results are obtained in terms of recall. However, one would expect the precision value to be rather low, due to unbalanced classes in the data.

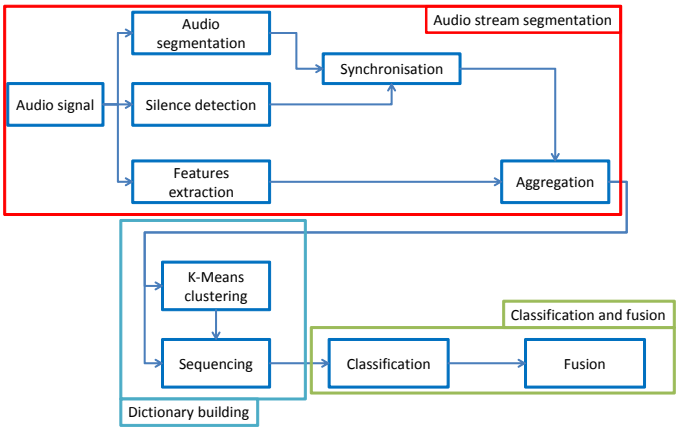

Fig. 1. System overview diagram.

Our system differs from [9] in several ways. First, the dictionary building mechanism is rather different. There is also a difference on how temporality is accounted for and in the type of audio material audio word description is applied to. We believe that Hollywood movies generate different challenges than MED data given that a lot more editing effort is made on movie soundtracks, resulting in high inter-movie variability.

III. System Overview

Our approach can be decomposed into three main parts, as illustrated in Figure 1. First, the audio stream is transformed into stationary segments. Second, audio words are computed for each audio segment. Finally, machine learning is used to classify the sequence of audio words. This section presents in turn these three steps.

A. Audio stream segmentation

The audio stream is segmented into stationary segments of variable length using the forward-backward divergence algorithm [11]. This algorithm produces segments whose length varies from 10ms up to several seconds, with an average of 20-30ms.

Independently, we also detect silent segments using the energy profile of the signal. The histogram of frame energies is approximated using two Gaussians from which a threshold is obtained to find low-energy segments corresponding to silence[1]. The reason for silence segmentation is that gunshots and explosions are highly susceptible to be non silent. The stationary segments extracted in the first step are said to be silent if the overlap with a silence segment is larger than 50 %. The number of audio segments detected as silent corresponds to about 50 % of the total number of segments.

Finally, short term audio features are extracted using 20 ms windows with 10 ms overlap. We consider three different feature types, namely

- **MFCC**: 12 Mel frequency cepstral coefficients
- **Energies**: 24 energies extracted from a uniform Mel-filterbank

[1]Silence segmentation is performed using the AudioSeg software, gforge.inria.fr/projects/audioseg.

- **Flatness**: 24 flatness features extracted from a uniform Mel-filterbank, where flatness is defined as the ratio between the geometric and arithmetic means of the spectrum in each filter.

For each type of feature, first and second order derivatives are also added. All features are normalized to zero mean and unit variance on a per movie basis. For each audio segment, short term features are aggregated by averaging over all frames within the segment. As a result, we end up with a 36 or 72 dimensions feature vector for each audio segment and with a binary label indicating which segments are silent.

B. Dictionary learning and segment quantization

Now, we want to replace the audio features of each segment by one or several symbols corresponding to audio words. The quantization dictionary learning phase implements a k-means algorithm using product quantization [12][2]. The main drawback of k-means is that the clustering time might become prohibitive if one wants a large number of clusters on a large feature set $\mathcal{F} := \{\mathbf{X} \in \mathbb{R}^D\}$, with D a large dimension. Product quantization consists in learning subquantizers on smaller dimensions, i.e., quantizers operating on a small part of the input feature vectors, and in combining the output, thus artificially increasing the number of centroids. Assuming N subquantizers with C centroids and dividing the input feature vectors as follows

$$\mathbf{X} := \{\underbrace{X_1, \ldots, X_{\frac{D}{N}}}_{1^{\text{st}} \text{ quantizer}}, \ldots, \underbrace{X_{D-\frac{D}{N}+1}, \ldots, X_D}_{N^{\text{th}} \text{ quantizer}}\} \ , \qquad (1)$$

where X_i, $i \in [1, D]$, correspond to the indexed dimensions of the input vectors $\mathbf{X} \in \mathcal{F}$, then the number of centroids is equal to C^N [12].

Subquantization also allows to consider different parts of a feature vector. As an example, consider three quantizers learned on the MFCC. The first one will correspond to static MFCC, the second one to the first derivatives and the last one to the second derivatives, thus resulting in one or several words per quantizer.

To sum up, in this step, N codebooks of size C are learned on the set $\mathcal{F}$ that comprises the non-silent samples of all the training movies, and each of these samples is assigned K words for each codebook. The K words are chosen as the K closest centroids according to the Euclidean distance. The silent segments are assigned to an additional word in each codebook. At the end, each segment is assigned KN words, K for each sequence, as presented in Figure 2.

C. Classification and fusion

After the quantization step, a classifier is learned on audio words. Bayesian networks are used to define a probability distribution over the features. The structure of the BNs is a sensitive issue, but such a structure can be efficiently learned from the data [13]. Moreover, the huge advantage of BNs over the popular SVMs is to have a very low parameter learning cost, and no hyperparameters to tune, this yielding better generalization capabilities. Contrarily to SVMs, the number of parameters in BN is only dependent on the structure of the graph and, in the absence of latent variables, the parameters are learned by counting in the learning database. Nevertheless, the BN inference complexity grows very fast with the number of variables used.

[2]via the Yael & LibPQ libraries.

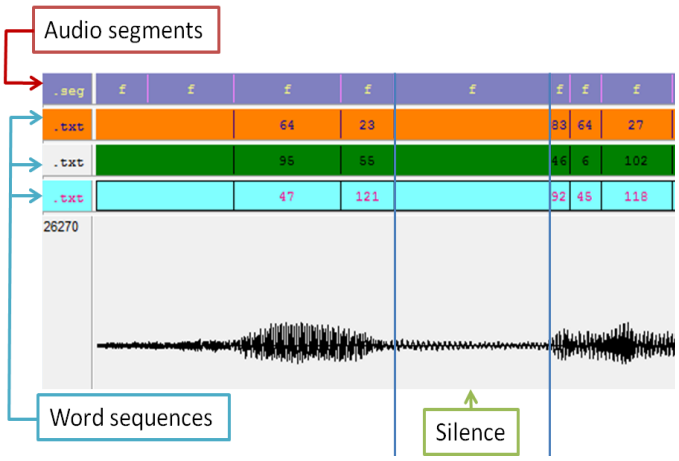

Fig. 2. Example of an audio sequence after the sequencing stage ($K = 1$).

Two types of BN have been tested: Naive BN and forest augmented naive BN (FAN). The naive BN makes the assumption that all features are conditionally independent with respect to the class node, i.e., the decision node which indicates the class of the sample. It is usually used as a baseline. The FAN relaxes the independence assumption by adding dependence links between the feature nodes. This more elaborated structure was shown to outperform the naive BN in a classification setup [14].

We also implemented BN in a contextual fashion as suggested in [5]: The samples are represented using their context over a sliding window. Therefore, in our setup, each segment will be represented by its own words and the words of n segments before and n segments after.

At the end, we obtain for each contextual sample and for each class, namely gunshots, explosions and others (everything not gunshots or explosions), the probability that the sample belongs to this class. Combining the results of gunshots and explosions also allows to verify whether the BN confuses them together or with others.

Finally, we analyze the impact of learning with all the feature types together as input, or fusing classifiers built on each feature type using average fusion or weighted sum fusion.

IV. Experiments

After presenting the dataset used, this section first focuses on an experimental study of the various choices for our system (feature type, codebook size, etc.) before providing comparative results on a test set.

A. Dataset

Experiments are performed on the MediaEval 2012 Affect Task dataset [3], composed of 15 movies in which gunshots

Movie	Duration (s)	G (s)	E (s)
Training data			
Billy Elliot	6,349.4	-	-
Eragon	5,985.4	-	25.42
Harry Potter 5	7,953.48	-	139.63
I Am Legend	5,779.88	41.00	27.72
Leon	6,344.52	84.33	13.66
Midnight Express	6,961	15.82	-
Pirates	8,239.36	153.41	63.49
Reservoir Dogs	5,712.92	43.34	-
The Sixth Sense	6,178	2.18	-
The Wicker Man	5,870.4	11.84	14.82
Total	**65,374.36**	**351.92**	**284.74**
Test data			
Armageddon	8,680.12	31.83	496.26
Kill Bill 1	6,370.44	23.36	2.00
Saving Private Ryan	9,750.96	2,501.22	1,229.39
The Bourne Identity	6,816	27.67	5.53
The Wizard of Oz	5,859.2	-	61.95
Total	**37,476.72**	**2,584.08**	**1,795.13**

TABLE I. DATASET USED FOR OUR EXPERIMENTS: G (RESP. E) INDICATES THE AMOUNT OF GUNSHOTS (RESP. EXPLOSIONS) IN SECONDS. PIRATES CORRESPONDS TO "PIRATES OF THE CARIBBEAN 1: THE CURSE OF THE BLACK PEARL".

and explosions are annotated[3]. Among these 15 movies, 10 movies were used for training and 5 for tests. The list of movies in the training and test sets, together with statistics on the events annotated, are given in Table I. Events of interest are scarce (about 1 % of the training data and 10 % of the test data) with a large variability between movies, depending mostly on the genre. This setting of rare event detection also makes the task more challenging.

B. Study on the parameters

The influence of the various parameters in the system is studied on the training data in a cross-validation setting, leaving one movie out for each fold.

Due to the large number of parameters in the system, reporting results for each of them is out of the scope of this paper and we limit ourselves to a discussion based on the results that were obtained. When studying one parameter, the others are kept constant to the following default values: MFCC, naive BN classifier, $C = 128$, $N = 3$, $K = 1$ and $n = 5$. Results are finally presented for the best setting.

BN structure: While FAN is supposed to work better than naive BN for classification, we observe that structure learning does not work in our case. Indeed, with the FAN network, almost no samples are detected as either gunshots or explosions (recall rate < 1 % for both classes). The BN structure do not influence the precision value.

Feature type: The different feature types are shown to be complementary. MFCC work much better for gunshots (recall > 70 %) than for explosions (recall < 9 %), while flatness or energies work much better for explosions (recall > 50 %) than for gunshots (recall < 20 %). With all features together, we achieve reasonable recall rates for each of the classes. The different feature types do not influence the precision value.

Codebook size: Codebook size C is chosen among $64, 128, 512, 2048$. Unsurprisingly, the bigger C, the lower the recall and the higher the precision. With the number of word combination growing fast with the codebooks sizes, the probability of finding an unknown combination grows, thus decreasing recall. Conversely, the probability that only a few combinations belong to one class grows, increasing precision.

[3]The dataset and its annotations are publicly available at: https://research.technicolor.com/rennes/vsd/.

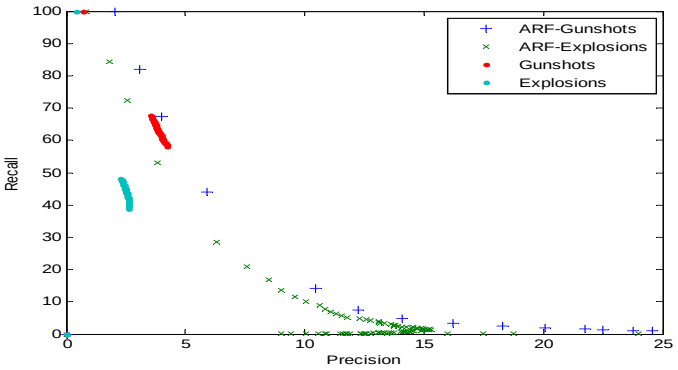

Fig. 3. Cross-validation results. Comparison with ARF cross-validation results.

Number of codebooks and words per codebook: Although increasing the number of codebooks ($N \in [1, 3, 9]$) or the number of words extracted per codebooks ($K \in [1, 3]$) increases the number of combinations and should have the same effect as increasing the size of the codebooks, we observe that recall tends to increase, and precision slightly dropps. We believe that using several words per codebook, i.e., $K = 3$, drastically improves the description capabilities of audio words. However, the number of variables in the Bayesian network increases and the inference complexity might become prohibitive with too many codebooks or words per codebook.

Context: Context refers to the length of the sliding context window used. We experimented with $n \in [2, 5, 10]$. The bigger n, the higher the recall but the lower the precision, and the higher the inference complexity.

Overall, precision rates are low (<5-6 %) due to the class imbalance in the data. Indeed, accepting 10 % of others as gunshots yields more samples than actual samples of gunshots as others correspond to more than 99 % of the data.

In the light of these experiments, a trade-off is made between results and complexity. Results of a naive BN classifier on all features with $C = 128$, $N = 3$, $K = 1$, $n = 5$ are compared in Figure 3 with those of ARF[4], which can be considered as a state of the art baseline. Besides using the same dataset, the choice to compare our results with the results of the ARF team was driven by the use of a proper leave one movie out cross-validation protocol. We believe such a protocol to report realistic performance values. For gunshots, we obtain results comparable with those of ARF with the difference that the recall/precision curves are concentrated in a higher recall zone. This limited range of operating points indicates an almost binary decision which demonstrates the robustness to variability of our system. The detection of explosions is significantly lower than for ARF in terms of precision,

[4]Though on the same dataset, results of the ARF team are obtained by cross-validation on the 15 movies, leaving one movie out, while ours are limited to movies in the training set.

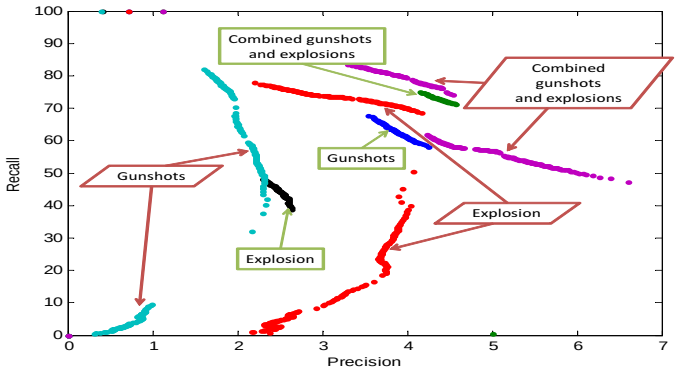

Fig. 4. All features VS weighted sum fusion for cross validation. Green boxes correspond to all features and red parallelograms correspond to weighted sum fusion.

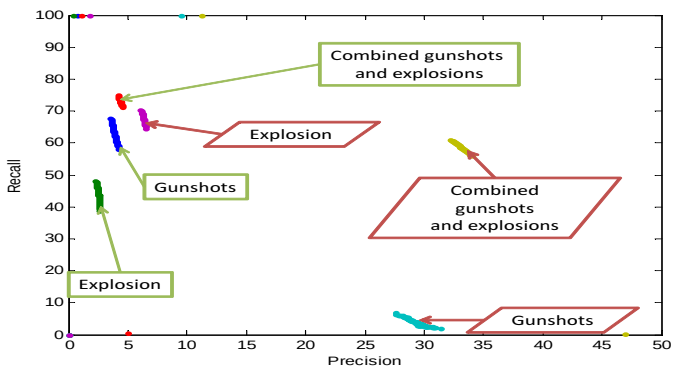

Fig. 5. Test VS cross-validation results for all features. Green boxes correspond to cross-validation and red parallelograms correspond to test.

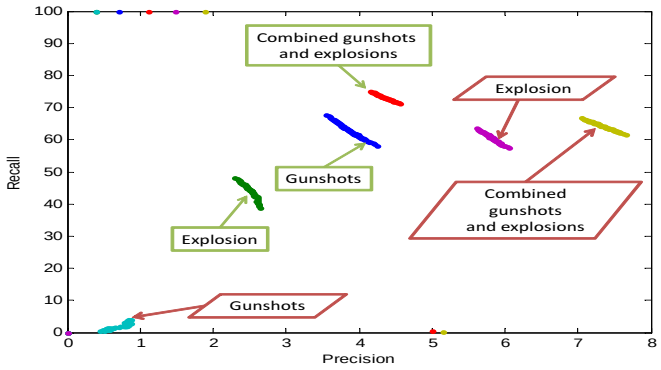

Fig. 6. Test VS cross-validation results for all features without Saving Private Ryan test results. Green boxes correspond to cross-validation and red parallelograms correspond to test.

although the recall/precision curve is concentrated in a higher recall zone.

We also experiment fusion of classifiers built on top of the different feature types. Experiments show that there is almost no difference between average and weighted sum fusion, therefore we only compare the results of early fusion and weighted sum fusion on Figure 4. We report results for gunshots, explosions and for the combination of both, i.e., merging the two classes. Late fusion clearly offers a larger range of operating points than early fusion: while robustness is decreased, a larger choice of compromises is available. The better results obtained when merging gunshots and explosions reveal a high confusion rate between the two classes. It is interesting to see that, despite the imbalance in the data, explosions are confused with gunshots more than explosions or gunshots with others.

C. Experimental results analysis

The best setting obtained in cross-validation is then applied on the test data for which we compare performance of early and late fusion. Results obtained for each feature type are not reported in the section but are nevertheless interesting as they contradict what was observed on the training set. On the test set, MFCC perform as well as flatness and filter-bank features for explosions. For gunshots, while recall is high in cross-validation with MFCC ($\simeq$70 %), it falls down to almost 0 on the test data. We believe that these results are due to the high variability between movies.

Figure 5 presents the results with all the features together and compares them with cross-validation results on the training data. The first thing that emerges is the bad recall rate of gunshots, which is due to the use of MFCC. Indeed, when building a model using only energies and flatness (not shown), gunshots recall is better. However, the combination results state that gunshots are actually classified as explosions, and not as others. The second thing that surfaces is the fact that the recall of explosions is much better for the test movies than in cross-validation. Finally, the last thing worth mentioning is that precision rates on the test movies are multiplied by 3 for explosions and almost 6 for gunshots compared to cross-validation. This is mostly due to one movie in the test set, Saving Private Ryan, which contains $\simeq$96 % of the gunshots and $\simeq$70 % of the explosions in the test database. Therefore, the results presented are mostly due to this movie which works surprisingly well compared to the other movies. Results not taking into account Saving Private Ryan are reported in Figure 6. The same conclusions hold for the confusion of gunshots as explosions, however precision rate for the combination is closer to cross-validation results, achieving up to 7.8 %.

All the previous conclusions indicate that, while audio words can handle more variability than low-level features, robustness to variability between movies is still an issue. This is shown in particular through the results obtained on Saving Private Ryan and the difference between test and cross-validation results. The low precision rates indicate that there is still a huge number of other segments confused as gunshots or explosions. Yet, this huge number only corresponds to only 10 to 15 % of the other segments. Despite the fact that the precision is still low and that the variability is still present, the high recall rate is quite encouraging for the continuation of our work. Moreover, our cross-validation experiments confirmed that using contextual features improves the results over static classification. We also confirmed that the use of multiple audio words, i.e., $K > 1$, to describe a segment improves the description capabilities of audio words.

V. Conclusions

This article presents a novel and simple system for audio event detection in movies yet comparable to the state-of-the-

art. The variability between the different movies is reduced using similarity representation between audio segments based on the use of k-means for representing audio segments with words. We investigate the use of multiple words for segment description, using subquantizers and multiple codeword assignments. Finally, BN are used with contextual features in order to detect gunshots and explosions in movies. This work is novel in the way the audio words are used, i.e., combined with subquantizers and contextual features. The system has been evaluated on a publicly available dataset, the MediaEval 2012 Affect Task dataset, that comprises some audio and video concepts annotations for 15 Hollywood movies [3]. A comparison with the state-of-the-art system of the ARF team is provided.

The high recall rates and the confusion between gunshots and explosions show that a first step has been taken towards solving the generalization problem in movies, which is an encouraging result for future research. Learning with all the features provides a robust algorithm with respect to decision threshold setting but limits the choice for an operating point, while the fusion of classifiers using either average or weighted sum fusion provides slightly better results and more operating points for the user. The low precision values are due to the strong imbalance between gunshots and explosions samples on the one hand, and the rest on the other hand.

While we believe that the use of multiple audio words increases the robustness to variability between movies, the difference between the results of Saving Private Ryan and the other movies states that the variability is still a problem that needs to be resolved. Modeling the variability, e.g., using factor analysis [15], is possibly complementary to the audio word representation. The problem of reducing the false alarms causing the low precision rates is also an axis for future research.

VI. Acknowledgement

We would like to thank Jan Schlüter, Bogdan Ionescu, Ionuţ Mironică and Markus Schedl from the ARF team for providing us with their recall-precision curves results, and allowing comparison.

References

[1] T. Giannakopoulos, D. I. Kosmopoulos, A. Aristidou, and S. Theodoridis, "A multi-class audio classification method with respect to violent content in movies using bayesian networks," in *IEEE Workshop on MSP*, 2007.

[2] J. Schlüter, B. Ionescu, I. Mironică, and M. Schedl, "ARF @ MediaEval 2012: An Uninformed Approach to Violence Detection in Hollywood Movies ," in *MediaEval 2012 Workshop*. ceur-ws.org, 2012.

[3] C.-H. Demarty, C. Penet, G. Gravier, and M. Soleymani, "A benchmarking campaign for the multimodal detection of violent scenes in movies." in *ECCV 2012 Workshop on IFCVCR*, springer, Ed., 2012.

[4] I. Trancoso, T. Pellegrini, J. Portelo, H. Meinedo, M. Bugalho, A. Abad, and J. Neto, "Audio contributions to semantic video search," in *ICME*, 2009.

[5] C. Penet, C.-H. Demarty, G. Gravier, and P. Gros, "Multimodal Information Fusion and Temporal Integration for Violence Detection in Movies," in *ICASSP*, 2012.

[6] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection in noisy environments," in *EUSIPCO*, 2007.

[7] J. J. Burred, "Genetic Motif Discovery Applied To Audio Analysis," in *ICASSP*, 2012.

[8] M. L. Chin and J. J. Burred, "Audio Event Detection Based on Layered Symbolic Sequence Representations," in *ICASSP*, 2012.

[9] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio Event Detection from Acoustic Unit Occurence Patterns," in *ICASSP*, 2012.

[10] P. Over, G. Awad, , B. A. M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quénot, "TRECVID 2011 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisms, and Metrics," in *Proceedings of TRECVID 2011*. NIST, USA, 2011.

[11] R. Andre-Obrecht, "A new statistical approach for the automatic segmentation of continuous speech signals," *ASSP, IEEE Trans. on*, vol. 36, no. 1, pp. 29 –40, jan 1988.

[12] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. on PAMI*, vol. 33, no. 1, pp. 117–128, jan 2011.

[13] G. Gravier, C.-H. Demarty, S. Baghdadi, and P. Gros, "Classification-oriented structure learning in bayesian networks for multimodal event detection in videos," *Multimedia Tools and Applications*, pp. 1–17, 2012.

[14] P. Lucas, "Restricted Bayesian Network Structure Learning," in *Advances in Bayesian Networks, Studies in Fuzziness and Soft Computing*, 2002, pp. 217–232.

[15] D. Matrouf, F. Verdet, M. Rouvier, J. francois Bonastre, and G. Linarès, "Modeling nuisance variabilities with factor analysis for gmm-based audio pattern classification," *Computer Speech & Language*, vol. 25, no. 3, pp. 481 – 498, 2011.