AUDIO EVENT DETECTION FROM ACOUSTIC UNIT OCCURRENCE PATTERNS

Anurag Kumar[1]*, Pranay Dighe*[1]*, Rita Singh*[2]*, Sourish Chaudhuri*[2]*, Bhiksha Raj*[2]

1. Indian Institute of Technology, Kanpur, India
2. Carnegie Mellon University, Pittsburgh, PA, USA
{anuragk,pranayd}@iitk.ac.in, {rsingh,sourishc,bhiksha}@cs.cmu.edu

ABSTRACT

In most real-world audio recordings, we encounter several types of audio events. In this paper, we develop a technique for detecting signature audio events, that is based on identifying patterns of occurrences of automatically learned atomic units of sound, which we call Acoustic Unit Descriptors or AUDs. Experiments show that the methodology works as well for detection of individual events and their boundaries in complex recordings.

Index Terms— Acoustic event detection, AUDs, Acoustic unit descriptors, Audio analysis, Audio retrieval

1. INTRODUCTION

This paper deals with the detection of audio event categories in real-life recordings.

Most real-world audio recordings are complex, in that they consist of sequences of many different sound events. We call a relatively short series of sounds an *event* if it can be distinguished as such by a human regardless of the acoustic context in which it occurs. For example the sound of water splashing, regardless of whether it is in the context of a stream or a kitchen sink or a party setting, is distinct to a human ear. However, the definition of an audio event is still likely to be inherently ambiguous, since audio events of any level are likely to be further divisible into smaller compositional events. For example, the sound of an ocean wave breaking may further be composed of the sound of the water rising, and then of dispersing on itself or on an obstacle like the shore land or rock. Taken out of context, these sub-events would no longer be semantically meaningful, except perhaps at a very coarse level, such as "whoosh", "crack", etc. To avoid some of this ambiguity, we set down the following definitions for this paper: (a) we call the set of naturally occurring sound events that can be given a semantic event label an *audio event*; (b) we call the lower-level basic sub-events "atomic units" of sound. These units form an alphabet for sounds in that they are much smaller in number, compared to the possibly infinite number of events in the real world that they can compose.

Given these definitions at the outset, we now address the problem of detecting audio events in recordings as the problem of detecting the patterns of atomic units that are likely to correspond to an audio event.

In recent work [1] we introduced an *unsupervised* mechanism for automatically discovering the atomic units of sound from unlabelled data. We call the discovered units *Acoustic Unit Descriptors*, or AUDs. Sequences or patterns of AUDs compose events. Since the AUDs themselves are automatically discovered, it is not possible to assign distinct labels such as "whoosh" or "crack" to them (although such associations may exist). Nevertheless, we demonstrated that by characterizing audio recordings in terms of their composition in terms of AUDs we can perform tasks such as retrieving audio recordings corresponding to a given semantic category (*e.g.* "baseball game") from a large corpus [2], or "summarize" generic audio recordings in a manner that only retains the distinctive portions of the recording while discarding irrelevant portions [3].

In this paper we analyze individual audio events in terms of the AUD patterns within them. In principle, these patterns may be characterized as "grammars", since the composition of audio events in terms of atomic sound units is often structured, however, we use a simpler unigram-based characterization that merely utilizes the relative occurrences of AUDs as signatures of the events. By searching any audio recording for the occurrences of these patterns using a simple discriminative classifier, we are able to accurately detect the occurrences of many varieties of sound events with a low rate of false positives, detecting over half of instances of desired events for no false positive at all.

A short note on the state-of-the-art in audio event detection is in order. The detection of specific audio events such as gunshots and screams have been of interest to the surveillance community, and a large number of techniques have been proposed. In general, these employ simple frame-level characterization of the audio and a variety of classifiers such as GMM based classifiers [4] and Bayes nets [5]. Commercial devices for indexing sports audio similarly depend on the detection of cheering and ball hits. More generally, several authors have also attempted to detect various audio concepts in generic multimedia recordings. Lee et. al. [6] use a simple GMM based characterization of the distribution of cepstral vectors in the audio to detect individual semantically tagged events. They impose an HMM-based characterization over these events to represent higher-level structure. In [7] they characterize segments of the audio through the Gaussian population histograms derived from a GMM. Audio event detection is directly performed as a classification task employing GMMs in [8]. In [9] individual events are modelled as HMMs, and a speech recognition framework is employed to detect them. In [10] the same authors use a GMM-supervector characterization combined with SVMs to specifically detect the sounds of falling objects in audio. SVMs are more directly employed over feature vectors derived from audio in [11].

The various techniques reported above have been shown to be reasonably successful at detecting audio events in audio recordings, given sufficient supervision in the training of the underlying classifiers. However, in all cases, the classification strategies employ models that work directly on feature vectors derived from audio, and the models employed are fundamentally unstructured: besides modeled as time series nature of the data using HMMs, they do not consider the finer-level characterization of these *macro* events in terms of their composing units. In a sister submission [2] we have shown that simple-frame level characterization of audio can result

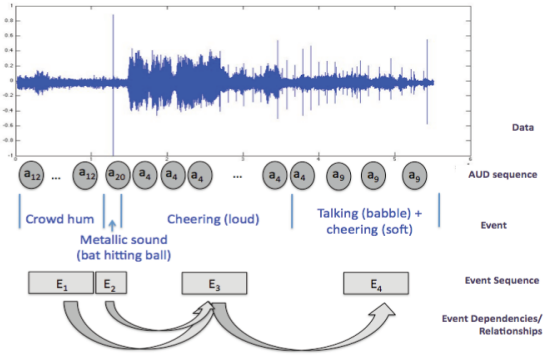

Fig. 1. Audio is modeled as being composed of a sequence of lower-level atomic sound units. Patterns over these units characterize higher-level phenomena.

in significantly inferior retrieval performance for audio, than when the audio is characterized in terms of its constituent units, namely the AUDs. The approach taken in this paper explicitly characterizes the audio events as patterns of AUDs. While we are unable to compare the performance of our approach on the optimized implementations by other authors on identical data, based on the results reported in this paper, where we show good detection performance for audio categories, and those in the sister submission [2] where we compare frame-based and AUD-based retrieval, we believe that the AUDs based characterization can result in superior performance and merits consideration and further investigation.

The rest of this paper is organized as follows: In Section 4 we describe our AUDs-based model and the formalism we employ to learn the AUDs. In Section 3 we describe the framework we employ for composing classifiers for individual audio events. In Section 4 we describe our detection framework. In Section 5 we describe our experiments and in Section 6 we finally present our conclusions.

2. ACOUSTIC UNIT DESCRIPTORS

The basic problem we address in this paper is that of automatically detecting audio events in a recording. We model all audio as being composed of a finite set of *atomic* sound units, such that every instant of a recording is part of a unit. The *patterns* over these units are descriptive of higher-level events. Figure 1 illustrates the model through an example of a clip of a recording from a local baseball game. The various audio events to be found in this clip are the crowd murmur, ball hit, cheering, and babble. Each of these individual events in turn are composed as sequences of smaller units which capture *atomic* events, which we term "Acoustic Unit Descriptors" or AUDs. The audio events themselves, namely the murmur, hit, etc., are characterized by the *patterns* that the AUDs form in composing them. If the AUDs and the patterns were known, then the problem of identifying the occurrence of any individual audio event would be reduced to detecting if the specific pattern of AUDs that represent that event has occurred in the audio.

The problems we address are therefore threefold:

1. Learning the set of AUDs that compose the audio,
2. Discovering the pattern over AUDs that characterize individual audio events, and
3. Detecting the occurrence of these patterns in novel audio, in order to detect the occurrence of the event.

Of these, we have previously described a solution for 1 in [1], and only briefly reprise it here for the benefit of the reader. Our solutions to 2 and 3 are in subsequent sections.

The first problem, learning the AUDs, is complicated by the fact that although the notion of the atomic unit is simple enough, we neither have a comprehensive list of all atomic units that can compose sounds in general, nor any data that is transcribed in terms of these units. We therefore treat it as a problem of unsupervised learning. In doing so, we make a critical assumption: we assume that we can model all audio using only a relatively small set of AUDs. The AUDs may hence no longer be assumed to have any semantic import of their own. Having done so, we can now employ an unsupervised formalism for learning basic sound units, that has previously been employed for unsupervised discovery of phones from speech signals [12, 13].

We represent the audio signal as a sequence of mel-frequency cepstral vectors, as is the norm in speech recognizers. We model each AUD by a Bakis-topology HMM with Gaussian-mixture state-output densities. "Learning" the AUDs now becomes identical to learning the parameters of these HMMS. Given a collection of audio recordings $\{\mathcal{A}\}$, the problem of learning the AUDs now becomes one of jointly estimating the parameters Λ of the HMMs modeling the AUDs and the *transcriptions* $T(\mathcal{A})$ of each of the recordings $\mathcal{A}$ in the training data in terms of these AUDs. The entire estimation procedure is an iterative maximum-likelihood estimator, *i.e.* we perform the estimation as $\mathrm{argmax}_{\Lambda,\{T(\mathcal{A})\}} P(\mathcal{A}|\Lambda, \{T(\mathcal{A})\})$. We refer the reader to [1] for additional details on the process.

The outcome of the learning process is the set of HMM parameters Λ for the AUDs. Also obtained is a *language model* Θ over the AUDs derived from $\{T(\mathcal{A})\}$, that represents typical N-gram patterns over the AUDs. For the work in this paper we will use a simple unigram model, as we have found this to result in the most robust performance.

The *key* step of our technique follows the learning of these models. We employ these *acoustic* models Λ and the language model Θ to *decode* all audio recordings, both training and test data, into sequences of AUDs in a manner similar to that employed by automatic speech recognition systems, *i.e.* for each recording $\mathcal{A}$ we derive $T(\mathcal{A}) = \mathrm{argmax}_T P(T|\mathcal{A}; \Lambda, \Theta)$. Thereafter we use $T(\mathcal{A})$ as a proxy for $\mathcal{A}$. This procedure converts all audio into sequences of symbols, each symbol representing an AUD. All subsequent processing is performed solely over these symbol sequences, although we also do retain information about the precise time instants where each symbol has been hypothesized in the audio. In the rest of this paper we will assume that each audio recording is represented by a AUD sequence (with corresponding time stamps where required).

3. CHARACTERIZING AUD PATTERNS FOR AUDIO EVENTS

The basic premise of our model is that it is the *pattern* over AUDs that characterizes individual audio events. To illustrate, a running sound is characterized by a rather rhythmic repetition of impact sounds, while water gurgling down a drain is a fairly complicated, yet distinctive pattern of tinkles, swishes and other such units (we use these semantically meaningful units in our description only for illustrative purposes; automatically learned AUDs themselves may not have associated semantics as we have mentioned earlier). Given the rather unconstrained complexity of these patterns, and also given the fact that the decoding of audio recordings into AUD sequences can by no means be considered to be perfect and will almost certainly be noisy in itself, we do not attempt to capture the detailed

structure of these patterns.

Instead we use a simple unigram characterization to represent these patterns. Given a set of K AUDs, we characterize any segment $\mathcal{S}$ of audio by a K-dimensional *AUD count* vector $V_{\mathcal{S}}$. The j^{th} component of the vector represents the number of times the j^{th} AUD occurred within the segment $\mathcal{S}$. In order to eliminate variations arising from the lengths of the segments we normalize the vectors such that the components of the vector sum to 1, *i.e.* $\ell_1(V_{\mathcal{S}}) = 1.0$.

4. DETECTING AUDIO EVENTS

We treat the problem of detecting any audio event in a recording as one of binary classification. To do so, we require a set of *training* audio segments $\{\mathcal{S}_A\}$ for the audio event we aim to detect. From each segment $\mathcal{S}_A$ we compute an AUD-count vector V_{S_A} to obtain set of AUD-count vectors $\{V_{S_A}\}$. Audio segments that were longer than 10 seconds are partitioned into multiple 10-second-long segments. These complete collection of these segments represent *positive* instances of the event.

We also obtain a large set of audio segments $\{\mathcal{S}_{\overline{\mathcal{A}}}\}$ that do *not* contain the audio event. The segments $\mathcal{S}_{\overline{\mathcal{A}}}$ are randomly chosen from non-audio-event-containing segments of recordings, to have a length equal to $min(10sec, L_{max})$, where L_{max} is the length of the longest audio-event-containing segment from $\{\mathcal{S}_A\}$. In practice, we round the boundaries to lie at the boundaries of AUDs, since the individual AUDs can span 30ms to several seconds. We derive a set of AUD-count vectors $\{V_{\mathcal{S}_{\overline{\mathcal{A}}}}\}$ from these recordings. These represent *negative* instances of the event.

From $\{\{V_{S_A}\}, \{V_{\mathcal{S}_{\overline{\mathcal{A}}}}\}\}$, the set of positive and negative training examples, we train a discriminative binary classifier C_{event}. Although any discriminative classification mechanism could be used in principle, in this paper we have used a random forest classifier [14]. A key aspect of the random forest classifier is that in addition to the classification outcome for any test instance, it also provides a score which is, in essence, a measure of the distance of the instance from the classification boundary.

To detect the occurrences of the audio event in a test recording, although we could, in principle, use a dynamic programming strategy that optimizes detection performance directly, we employ a simple scanning heuristic instead. We segment the recording into segments of length equal to $min(10sec, L_{max})$, *i.e.* the lower of 10 seconds, or the length of the longest positive training instance for the event. Segmentation is performed such that adjacent segments overlap by 75% of their length. From the t^{th} such segment, $\mathcal{S}_t$, we derive an AUD-count vector $V_{\mathcal{S}_t}$. We classify each of the T vectors $V_{\mathcal{S}_t}$, $t = 0, \ldots, T$ derived from the recording with the classifier C_{event} to derive a sequence of scores $s_0, \ldots, s_T$.

Since our goal is detection of events rather than classification of individual segments, we perform an additional heuristic: we median filter the sequence of scores $s_0, \ldots, s_T$ to obtain and smoothed score sequence $\hat{s}_0, \ldots, \hat{s}_T$. This score sequence is compared to a threshold. Contiguous segments of audio that have a score that lies above the threshold are classified as a single detection of the event. If, however, the length of a single detection exceeds $2L_{max}$, twice the length of the longest training instance of the event, we segment the duration of the detected event into shorter segments of length $2L_{max}$ and consider each to be an individual detection of the event. Figure 2 illustrates this procedure.

When evaluating the performance of the detector, we consider any detection to be a true positive if (a) more than 50% of the detected segment coincides with a true instance of the event or (b) more than 50% of a true instance of the event falls within the detected segment. Moreover, a single detected segment is only permitted to match a single true occurrence.

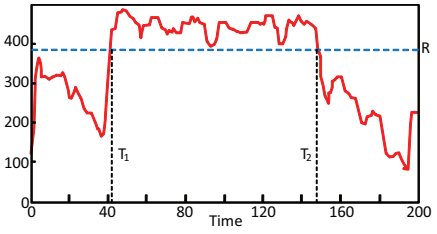

Fig. 2. Trajectory of classifier score vs. time. Each point on the trajectory represents the median filtered value of the the score computed by the classifier for a corresponding segment of audio. Here the score is being compared to a threshold R, resulting the in the segment (T_1, T_2) being detected as an instance of the event.

5. EXPERIMENTS

We ran experiments on the TRECVID, 2011 [15] corpus. As specified for the Multimedia Event Detection (MED) track of TRECVID, the data comprises a large number of multimedia recordings. The clips are tagged as belonging to one of a small number of broad-category events, such as *attempting a board trick*, *feeding an animal*, *landing a fish*, *wedding ceremony*, *working on a woodworking project*, etc. Correspondingly, the recordings also contain various audio events within them.

Of these, we have audio-event-level labels for several audio events such as laughter, wedding vows, singing, engine noises, etc. for a total of 860 files, provided to us by SRI Sarnoff labs. The occurrence of most events in this data, however, is very sparse – there are only 5 instances of *footsteps* in the entire set of labelled files, for instance. For our experiments, we chose 10 categories of audio events for which a sufficient number of instances were available to enable partitioning into both training and test data: "wedding audio", "board hitting surface", "cheering", "children's voices", "clapping", "crowd noise", "engine noise", "music", "scraping" and "singing".

The 860 files were partitioned into a training set of 638 files for training and 222 files for testing. Consequently the number of segments of positive instances of each event in the training data (recalling that segments were limited to be no longer than 10 seconds in length) was as follows: wedding audio: 95, board hitting surface: 111, cheering: 360, children's voices: 208, clapping: 241, crowd noise: 789, engine noise: 301, music: 3223, scraping: 288 and singing: 266. For negative training instances for each event, we derived random segments of the recordings of length equal to the longest positive instance. The number of negative instances was set to be approximately equal to the number of positive instances to balance the training data.

The number of instances of each of these sounds in the test audio were as follows – wedding audio: 33, board hitting surface: 38, cheering: 122, children's voices: 70, clapping: 81, crowd noise: 267, engine noise: 100, music: 1069, scraping: 102 and singing: 90.

Figure 3 shows the ROC curves showing the tradeoff between missed detection and false alarms for the detectors for each of the events. False alarms are reported in terms of false alarms per unit time, since this is a more relevant measure for analysis of continuous audio. Table 1 shows detection recall obtained when the operating

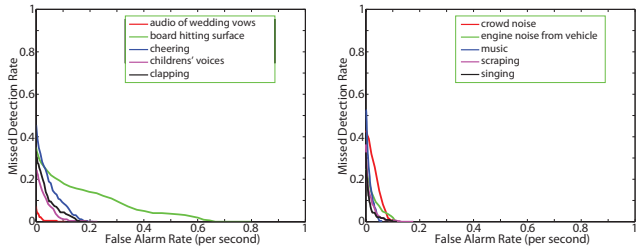

Fig. 3. ROCs for a number of different events, reporting missed detections as a function of false alarms per second

Event	Recall	AUC
wedding audio	99.5	0.0009
board hitting surface	77.4	0.066
cheering	81.5	0.02
children's voices	92.4	0.009
clapping	91.5	0.01
crowd noise	85.5	0.01
engine noise	92.8	0.009
music	99.7	0.007
scraping	97.6	0.005
singing	97.9	0.004

Table 1. Performance of detectors.

point was chosen to achieve a false alarm rate of one false alarm every 20 seconds. The AUC column shows the area under the curve in Figure 3. Ideally this number would be 0.

6. DISCUSSION AND CONCLUSION

It is clear from our results that the proposed approach is able to accurately detect several categories of audio event in a recording. In nearly all categories, we are able to detect over 50% of all events for no false alarms at all on our tests. At the rather more noisy operating point of one false alarm every 20 seconds, we manage to detect nearly all instances of most events.

The most difficult to detect category is the sound of a board hitting a surface. This is a short event which is even audibly hard to distinguish from other impact sounds. We note that the task being performed here is not multi-class classification but that of detection. In other words, other similar impact sounds can well confuse the detector. Nevertheless we are able to detect over half of all instances of the sound for no false alarm at all.

Particularly interesting is the fact that the events that are best detected are those that have complex structure that translate to complex AUD patterns. Yet, the AUD patterns for simpler sounds such as the board hitting a surface also comprise more than just one AUD, and this enables us to distinguish the sounds from other impact sounds.

In our experiments we have not run comparable detection experiments using techniques reported by other authors elsewhere, since no comparable experiments have been reported on the MED-11 dataset, but based on the results reported by other authors on other data sets, we may claim that the proposed method minimally provides comparable results with the best of other reported techniques.

The AUDs based characterization does also provide additional benefits. We currently only employ unigram characterizations of AUD patterns: by increasing the detail of the characterization of patterns we may expect to get better performance. Also, the mechanism of converting audio to symbol sequences enables us to deal with detection as a text-based retrieval problem. The AUDs used in these experiments were trained on data unrelated to the task at hand. Improving these are also expected to improve performance better.

7. ACKNOWLEDGEMENTS

We thank SRI Sarnoff labs for providing us with the annotations for the data. We also thank Ajay Diwakaran of SRI Sarnoff for discussions and suggestions.

8. REFERENCES

[1] S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification," in *Interspeech*, 2011, pp. 717–720.

[2] S. Chaudhuri, R. Singh, and B. Raj, "Data driven acoustic units for audio classification," in *Submitted to ICASSP*, 2011.

[3] S. Chaudhuri and B. Raj, "Learning contextual relevance of audio segments using discriminative models over aud sequences," in *WASPAA*, 2011.

[4] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE Conf. on advanced Video and Signal Based Surveillance*, 2008, pp. 21–26.

[5] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis, "Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2008.

[6] K. Lee, D. Ellis, and A. Loui, "Detecting local semantic concepts in environmental sounds using markov model based clustering," in *ICASSP*, 2010.

[7] K. Lee and D. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18:6, pp. 1406–1416, 2010.

[8] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *IEEE Intl. Conf. on Acoustics Speech and Signal Processing*, 2006.

[9] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson, "Feature analysis and selection for acoustic event detection," in *IEEE Intl. Conf. on Acoustics, Speech and Sig. Proc.*, 2008.

[10] X. Zhuang, J. Juang, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using gaussian mixture models and gmm supervectors," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2009.

[11] L. Lu, F. Ge, Q. Zhao, and Y. Yan, "A svm-based audio event detection system," in *Intl. Conf. on Electrical and Control Engineering*, 2010.

[12] M. Bacchiani, "Speech recognition system design based on automatically derived units," *PhD Thesis*, 1999.

[13] R. Singh, B. Raj, and R. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Trans. on Speech and Audio Processing*, vol. 10:2, pp. 89–99, 2002.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[15] "Trecvid 2011," http://www.nist.gov/itl/iad/mig/med11.cfm.