SOUNDTRACK CLASSIFICATION BY TRANSIENT EVENTS

*Courtenay V. Cotton, Daniel P. W. Ellis**

LabROSA, Dept. of Electrical Engineering
Columbia University
{cvcotton,dpwe}@ee.columbia.edu

Alexander C. Loui

Kodak Research Laboratories
Eastman Kodak Company
alexander.loui@kodak.com

ABSTRACT

We present a method for video classification based on information in the soundtrack. Unlike previous approaches which describe the audio via statistics of mel-frequency cepstral coefficient (MFCC) features calculated on uniformly-spaced frames, we investigate an approach to focusing our representation on audio transients corresponding to soundtrack events. These event-related features can reflect the "foreground" of the soundtrack and capture its short-term temporal structure better than conventional frame-based statistics. We evaluate our method on a test set of 1873 YouTube videos labeled with 25 semantic concepts. Retrieval results based on transient features alone are comparable to an MFCC-based system, and fusing the two representations achieves a relative improvement of 7.5% in mean average precision (MAP).

***Index Terms*—** Acoustic signal processing, Multimedia databases, Video indexing

1. INTRODUCTION

The enormous volumes of video being captured by consumers, stored on computers, and uploaded to the Internet, presents an urgent need for automatic tools for video classification and retrieval – since they are often insufficiently labeled by their creators. While visual content is the most obvious basis for automatic analysis, the soundtrack of a video also contains important information about a clip's content, information that may be complementary to the video stream, and that may also be easier to process or recognize. We have been investigating the use of soundtracks in video classification for several years [1, 2].

A common approach to modeling audio is to extract features from uniformly-spaced short-time frames (e.g. 25 ms) extracted from the entire length of the soundtrack. A video's soundtrack, however, may have information that is very unevenly and sparsely distributed – such as an occasional dog barking, or other foreground sound event. Short, sparse events of this kind may have relatively little statistical impact when mixed in with all the frames in the clip, and their information may be lost.

To address this risk, we have developed a system for representing the soundtrack based on identifying and modeling the individual audio transients it contains. By analyzing only a subset of points in the soundtrack that are likely to contain distinct event onsets, our goal is to develop an approach that is complementary to the typical global background model. At each transient event time, we also model the local temporal structure over a relatively long window (e.g., 250 ms instead of 25 ms), which we hope will be able to further capture the temporal characteristics of these events.

Some related work oriented towards extracting a sparse subset of relevant points in an audio track for the purpose of classification can be found in [3, 4]. Our work differs in a number of ways, including our application which is based around a set of 25 labels derived from a study with actual users [5].

2. PROPOSED ALGORITHM

The following section details the processing stages of our algorithm. Figure 1 shows a block diagram of the system and example data.

2.1. Automatic Gain Control

A major problem in dealing with YouTube-style amateur video is the wide variation in background noise characteristics, recording equipment, and quality. Since our goal is to characterize individual transients according to their underlying cause, we would like to minimize the extent to which differences in recording conditions will result in irrelevant variability in the extracted features. We attempt to address this problem by applying automatic gain control (AGC) as a pre-processing step. In addition to reducing irrelevant variation, this stage can also make the subsequent transient detection more accurate.

Our automatic gain control equalizes the energy in both time and frequency by first converting the signal into an invertible short-time Fourier transform (STFT) using 32 ms

*This work was supported by the Eastman Kodak Company and by the National Geospatial Intelligence Agency.

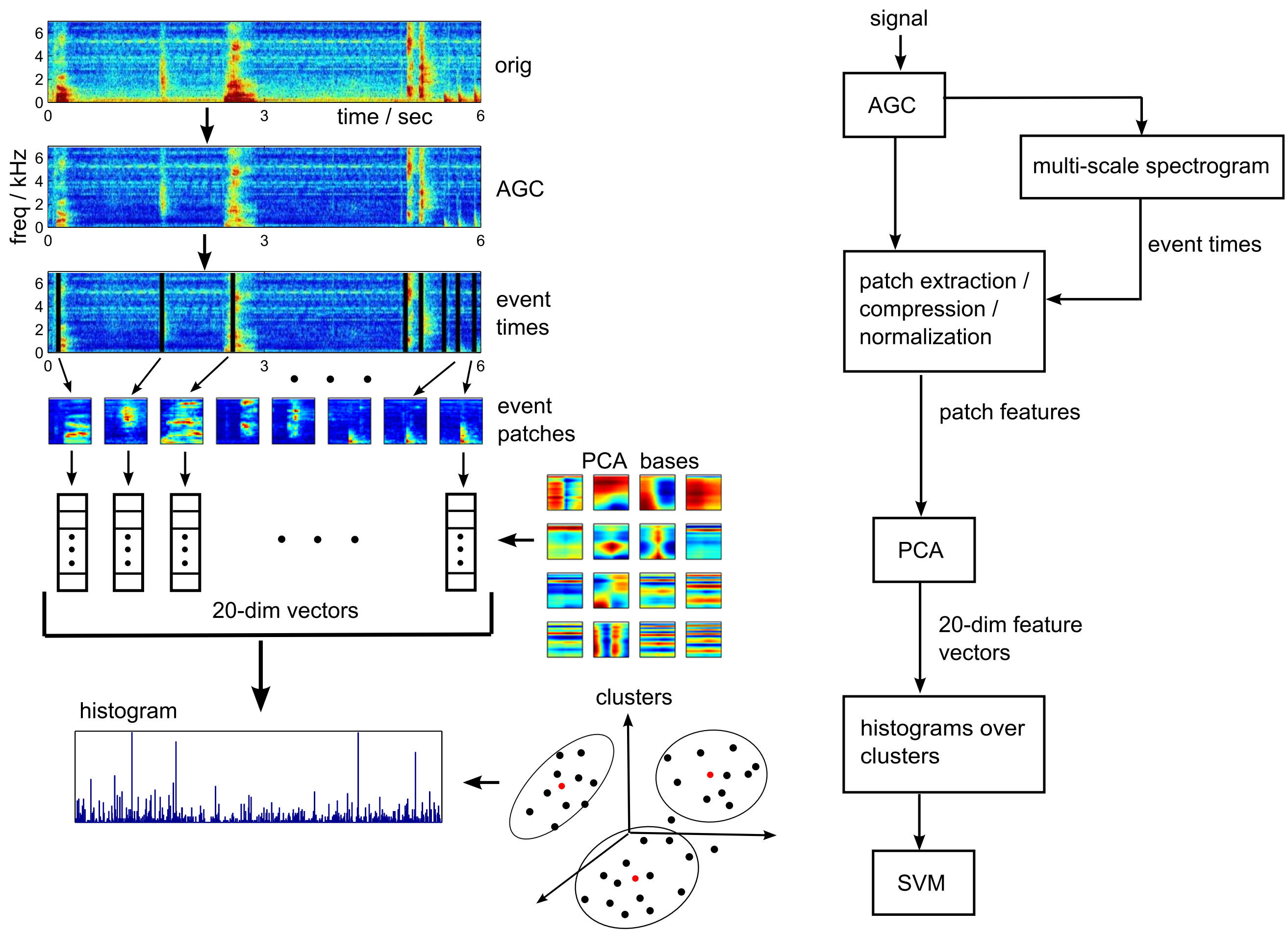

Fig. 1. Block diagram of the system, with examples of data.

windows. The magnitude of this representation is then smoothed across time and frequency, using a fixed time window, and a frequency window defined in terms of an auditory frequency axis, leading to wider integration windows (in Hertz) for higher center frequencies. We use a mel frequency mapping. The local average energy obtained by this smoothing of the energy surface is then divided out of the STFT magnitude prior to inverting back to an audio waveform using overlap-add synthesis. The code for this AGC is available[1].

The AGC parameters were tuned for our task. We used symmetric non-causal smoothing with frequency integration on the order of 1 mel and temporal smoothing on the order of 4 seconds.

2.2. Transient Detection and Feature Extraction

After applying the AGC, the STFT (or spectrogram) of the signal is taken for a number of different time-frequency tradeoffs, corresponding to window lengths between 2 and 80 ms. We use multiple scales to be able to locate events of different durations. High-magnitude bins in any spectrogram result in a candidate transient event at the corresponding time. A threshold is set to some amount above the local (temporal) mean in each frequency band, and bins with values higher than this threshold are recorded. Additionally, a limit is set on the minimum distance between successive events. In this case, the overall system was tuned to produce an average of around 4 events per second.

For each event time, a short window of the signal is extracted centered on the event time. This window is 250 ms long in order to capture the temporal structure of the transient. For this short snippet, we again take the STFT, this time at a single scale of 25 ms with 10 ms hops, and integrate the frequency dimension into 40 mel-frequency bands. The result is an event patch consisting of 23 successive time frames, each consisting of 40 frequency bins. We restrict the spectrum to 7 kHz to compensate for differences in the high-frequency cutoff characteristics of different recording equipment, which would otherwise affect the comparisons between event patches.

[1]http://labrosa.ee.columbia.edu/matlab/tf_agc/

Rather than take the log of a patch's magnitude values (as we would do if we were producing MFCCs), we raise the magnitude to a fractional exponent to compress larger values. This was determined empirically to perform better than taking the log. The specific exponent (0.2) was arrived at through tuning.

We finally normalize the patches by scaling the maximum value to be 1.

2.3. Clustering

The resulting feature patches have 920 dimensions, which is too large to efficiently compare. We perform principal component analysis (PCA) on the training data, and use the top 20 bases to reduce the dimensionality down to 20 elements. We then perform k-means clustering on the 20-dimensional training data to arrive at a set of K clusters. Here, K is selected to be 1000. We store the means and covariances of these clusters.

2.4. Video Feature Representation

For each test video, we again extract patches around all event times detected by our algorithm. We then characterize the video as a histogram of its events as they are distributed over the K learned clusters. Initially, we took histograms using hard assignment of each event descriptor to a single cluster. However, we improved performance by distributing an event's weight proportionally amongst all clusters. Specifically, we assign weight to the histogram bins according to the posterior probability that each patch comes from each cluster according to a Gaussian distribution given the cluster's mean and covariance. Each video's histogram is normalized by the total number of events extracted from that clip.

2.5. Concept Detection

We use support vector machines (SVMs) to compare videos using their histogram features. We train one SVM per concept. The SVM's gram matrix is computed as the Mahalanobis distance between histogram vectors (with covariance estimated from the entire training set), and SVM parameters are tuned on a validation data set. To produce retrieval results for a given concept, test videos are ranked according to their decision value (margin) under that concept's SVM.

3. EXPERIMENTAL RESULTS

The evaluation task is to retrieve videos in the order most relevant to each of the 25 concepts. We report average precision (AP) as the performance metric. Average precision is defined as the average over precision values evaluated at the depth of each true result in the ranked list.

The data used is a set of 1873 consumer videos downloaded from YouTube, and labeled with one or more of 25 semantic concepts, as described in [2]. We use five-fold cross-validation in our experiments, where each fold of the data is divided randomly into 40% training, 20% validation, and 40% testing data.

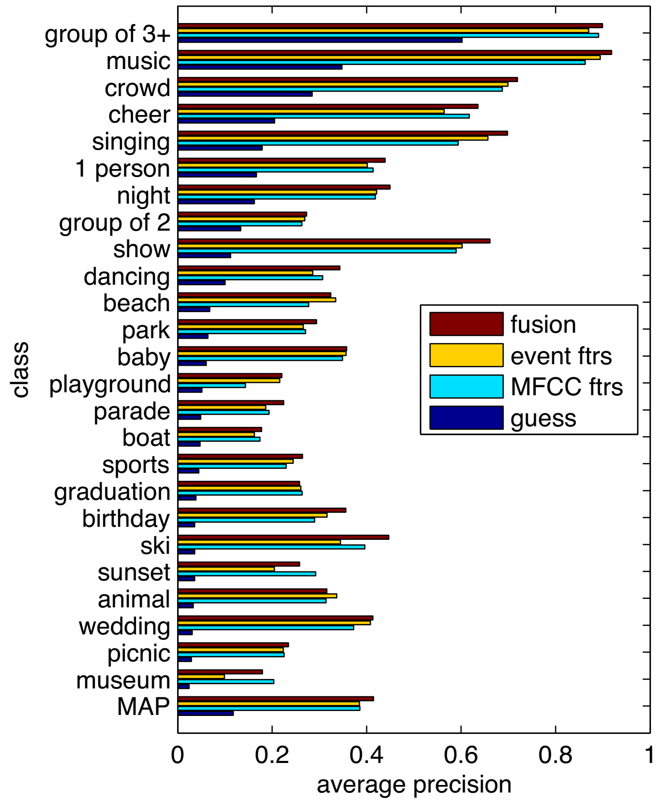

Fig. 2. Average precision results for each class, and mean AP.

We compare our results with a baseline approach in which MFCC features are extracted from every frame of the audio clip. The parameters used to extract MFCC's mirror those of our patch extraction stage: 25 ms frames with 10 ms hops, and 40 mel-frequency bands covering up to 7 kHz. Twenty-five coefficients are retained. Each clip's frames are modeled as a single Gaussian, where the clip's feature representation is the mean and (unique) covariance values of that Gaussian. A set of SVM's are then trained on this feature set, again using the Mahalanobis distance between these statistical parameters to characterize the distance between clips. This follows the single Gaussian modeling procedure of [2].

Lastly, we fuse the results from the two approaches. We do late fusion, wherein we add the (normalized) decision values from each SVM, using a weighting factor to trade off between the two decision values. The weight factor is optimized for each class, based on our expectation that the event-based system will be more effective at detecting some concepts and the global system will be more effective on others. The fusion weight used for each concept is tuned over the validation data.

Figure 2 shows average precision results over the 25 con-

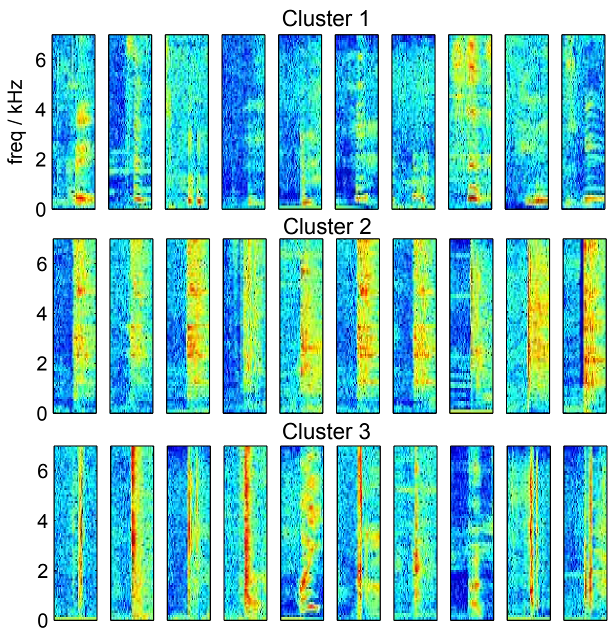

Fig. 3. Examples of event patches from three clusters that are well-correlated with the 'baby', 'birthday', and 'playground' concepts, respectively.

cepts and mean AP for all concepts for our algorithm, the MFCC model, and the fusion of the two. The AP that would result from guessing randomly is included for reference.

Figure 3 shows spectrograms of example event patches from three different clusters that are well-correlated with the labels 'baby', 'birthday', and 'playground', respectively. For example, listening to examples from the 'baby' cluster reveals that they generally correspond to similar-sounding instances of people laughing.

Table 1 shows mean AP values for the system as described above (original), and for some alternate parameter settings: without the AGC, with the event threshold adjusted to give an average of 2 events per second rather than 4, for larger and smaller values of K, and for 2 different settings of the patch compression exponent.

4. DISCUSSION AND CONCLUSIONS

We demonstrate that focusing a soundtrack representation specifically on the subset of the signal indicated by transient events can improve concept retrieval performance over simply modeling MFCC frames globally for an entire audio clip. Our fusion of these two models achieves a 7.5% relative improvement in the mean AP, from 0.38 to 0.41, over the global model alone. The event model is especially helpful for predicting some semantic concepts (such as "playground" and "animal") and less useful for others (such as "cheer" and "ski"). This is reasonable since some of the concepts in this test set would be expected to have more distinct types of

Parameter settings	MAP
original (AGC, 4 events/sec, K = 1000, comp. exp. = 0.2)	**0.385**
without AGC	0.332
2 events/sec	0.284
K = 500	0.378
K = 2000	0.375
patch compression exponent 0.125	0.375
patch compression exponent 0.4	0.335

Table 1. Mean AP results for some alternate parameter settings

events associated with them than others.

In addition to improving overall retrieval performance with our fusion method, we achieve comparable performance to a global model with our method alone. This is promising because it allows the possibility of building a classification framework around these type of sound events. The concept labels used in this task are a proxy for attempting to determine what is happening in a video. By building a concept detection system around events that also have some semantic meaning themselves, we can learn more about what is happening at an event level in the video. This has the potential to enhance search and retrieval capabilities for video based on the occurrence of specific audio events.

5. REFERENCES

[1] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A.C. Loui, and J. Luo, "Large-scale multimodal semantic concept detection for consumer video," in *Proc. ACM Multimedia, Information Retrieval Workshop*, Sept 2007.

[2] K. Lee and D.P.W. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE TASLP*, vol. 18, no. 6, pp. 1406–1416, Aug 2010.

[3] O. Kalinli, S. Sundaram, and S. Narayanan, "Saliency-driven unstructured acoustic scene classification using latent perceptual indexing," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Oct 2009.

[4] S. Chu, S. Narayanan, and C.C.J. Kuo, "Environmental sound recognition using MP-based features," in *Proc. IEEE ICASSP*, 2008.

[5] A.C. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, K. Lee, L. Kennedy, and A. Yanagawa, "Kodak's consumer video benchmark data set: Concept definition and annotation," in *Proc. ACM Multimedia, Information Retrieval Workshop*, Sept 2007.