SUBBAND AUTOCORRELATION FEATURES FOR VIDEO SOUNDTRACK CLASSIFICATION

Courtenay V. Cotton, Daniel P. W. Ellis

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

ABSTRACT

Inspired by the system presented in [1], we have developed novel auditory-model-based features that preserve the fine time structure lost in conventional frame-based features. While the original auditory model is computationally intense, we present a simpler system that runs about ten times faster but achieves equivalent performance. We use these features for video soundtrack classification with the Columbia Consumer Video dataset, showing that the new features alone are roughly comparable to traditional MFCCs, but combining classifiers based on both features achieves a substantial mean Average Precision improvement of 15% over the MFCC baseline.

Index Terms— Acoustic signal processing, Multimedia databases, Video indexing, Auditory models

1. INTRODUCTION

As the means to collect and share video and audio become increasingly ubiquitous and cheap, tagging and retrieval of multimedia content become increasingly important. Although much of this work focuses on the visual content of a video, modeling the audio content can also prove helpful for the purpose of search and indexing. A standard approach to characterizing audio content uses mel-frequency cepstral coefficients (MFCCs), which are short-time spectral features. There are on-going efforts to identify other useful features in this domain and novel methods for employing them in retrieval tasks. The authors of this work have previously investigated a number of novel features for this task, for example in [2].

In [1, 3], features based on an auditory model were presented for use in audio recognition. In contrast to traditional features which average the signal spectrum over 20-30 ms windows, the auditory model features attempt to preserve the fine temporal structure of the sound via a "stabilized image" of the waveform. These features were used in conjunction with the "passive-aggressive" model for image retrieval (PAMIR) as the learning mechanism. The authors showed that these features performed as well as or better than traditional MFCC features for retrieval tasks, and that they are particularly useful for the identification of sounds in mixtures. Since we are working with broadly similar problems of classifying unconstrained environmental audio, we investigated this system. We began by attempting to replicate their system as closely as possible and test it on a retrieval task on a corpus of tagged consumer video soundtracks.

The next sections introduce our data/domain, and then describe our results using an available implementation of the auditory model front end, both with the original PAMIR retrieval model, and with more conventional Support Vector Machine (SVM) classifications. Sections 5 and 6 describe our modifications to the original system to reduce the dimensionality of the representation, and to simplify the overall calculation to reduce its computational burden. Section 7 describes the further improvements we obtained by fusing these novel features with the existing baseline MFCCs.

2. DATASET AND TASK

We performed all evaluations on the Columbia Consumer Video (CCV) dataset described in [4]. This is a set of 9,317 video clips from YouTube, comprising 210 hours of video. The clips are tagged with 20 semantic categories. For all our experiments, the metric used was average precision of retrieval results for each category, with the mean average precision (mAP) over all categories serving as the main objective index of performance.

3. STABILIZED AUDITORY IMAGE FEATURES

The system of [1, 3] has a multi-step feature generation process. First the signal is passed through a time-varying filterbank intended to model the cochlea, including its local loudness adaptation (through changes in individual filter resonance). The filterbank outputs are then integrated using what the authors call strobed temporal integration. Strobe (peak) points are identified, and the signal is cross-correlated with a sparse function that is zero except at these strobe points. This

is done separately in each filter channel, resulting in a two-dimensional (number of channels $\times$ time lag) image, termed the stabilized auditory image (SAI). (In lieu of a more detailed description, please see the presentation of our simplified auditory model features in section 6). In their experiments an SAI is generated every 20 ms to characterize the audio signal at that point. Then a sequence of SAIs is converted into features using a sparse code representation as follows: Each SAI is overlaid with a set of rectangular patches of different sizes. Each of these rectangles defines a local region of interest on the SAI. The set of rectangle features is collected over all data, and each rectangle region is vector quantized (VQ) separately. A single SAI is then represented by a sparse code whose dimensionality is the number of rectangles times the size of each VQ codebook. An audio clip is represented as the sum of its SAI codes (essentially, a set of histograms).

To approximate this system, we used a publicly-available C++ codebase, AIM-C [5], that computes stabilized auditory images that are similar though not completely identical to those described in [1]. The audio data is first downsampled to 16 kHz and processed with AIM-C to produce a series of SAIs. The SAIs were then cut into 24 rectangles, using the box-cutting method described in [1], where the smallest boxes were 32 frequency channels by 16 time steps. Each dimension was then doubled systematically until the edge of the SAI was reached. We then downsampled and quantized each of the 24 rectangles with a 1000-codeword dictionary learned by k-means on the training set. This leads to a representation of each video clip as a sparse 24,000-element vector which is essentially the concatenation of the histograms over each of the 24 rectangles.

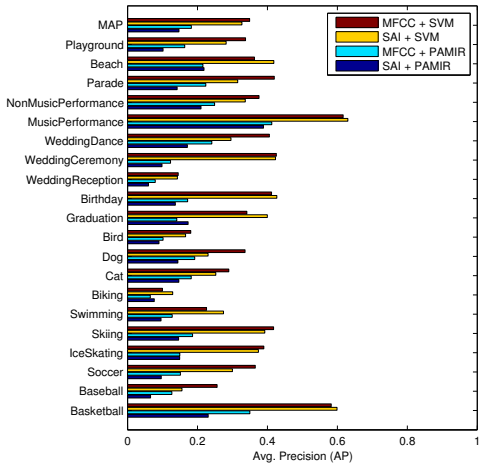

Fig. 1. Baseline system comparisons: MFCC and SAI features, in conjunction with both PAMIR and SVM learning methods.

4. PAMIR VERSUS SVM LEARNING

As in [1], we initially used PAMIR as the learning method in our system. PAMIR is an algorithm for learning a linear mapping between input feature vectors and output classes or tags. PAMIR is especially efficient to use on sparse feature vectors (such as the extremely high dimensional histograms described above), which is one reason the authors chose it.

We were unable to get particularly good performance from PAMIR. PAMIR is theoretically useful for learning associations reasonably quickly when the scale of the data is very large. However, in reality our experiments consisting of thousands, but not millions, of data items, were not large scale enough to necessitate the use of PAMIR. We realized that we could obtain better results by combining SAI features with more standard learning techniques such as support vector machines (SVMs).

Figure 1 compares the performance of SAI features using PAMIR and SVM learning techniques. As in [1], we compare the novel SAI features with a baseline system using standard MFCC features. Here, we used 20 MFCC coefficients and also added deltas and double deltas, for 60 dimensional feature vectors. For consistency with the SAI features, MFCC frames were vector quantized and collected into a single 3000-codeword histogram representation for each video clip. Figure 1 also shows results using these MFCC features with both learning methods. In our experiments, SVM learning significantly outperforms PAMIR on both feature sets. SAI and MFCC features perform roughly comparably to each other although MFCCs perform slightly better under both learning methods.

5. REDUCTION OF FEATURE SET SIZE

We were interested in investigating how the set of rectangle features selected influenced the final results. Specifically we wondered to what extent we could minimize the number of rectangles (in order to reduce feature vector size) while retaining a similar level of performance. The authors of [1] experimented with numerous rectangle cutting strategies but did not offer strong conclusions about the extent to which larger numbers of rectangles can lead to improved performance. Since their cutting method results in rectangles that overlap, there is presumably some duplicate information. Our goal was to see if we could minimize the number of rectangles while maintaining high performance.

We experimented with reducing the set of rectangles in various ways. The original set of 24 rectangles consists of rectangles covering four different frequency ranges (low frequency, high frequency, mid frequency overlapping both low and high, and all frequency bands together), at each of six timescales (where each timescale is twice as long as the previous one). We were able to achieve performance very close to the full set using only eight rectangles. Specifically, we

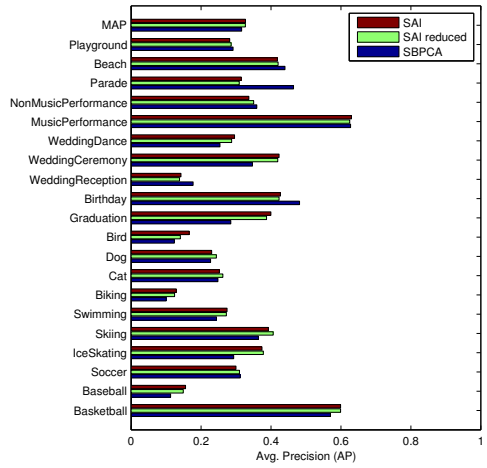

Fig. 2. Comparison of SVM systems using: full set of 24 SAI rectangles (24,000 dimensions), a reduced set of 8 SAI rectangles (8,000 dimensions), and the simpler SBPCA features (4,000 dimensions).

removed all rectangles from the mid frequency and full frequency ranges, keeping only high and low frequency rectangles. We also removed the largest two timescales, keeping only the shortest four. Figure 2 compares the SAI and SVM system using all 24 rectangle features (SAI) with only these eight rectangle features (SAI reduced), demonstrating that performance remains very similar between the two. The figure also includes our reformulated auditory model features, described below.

6. SUBBAND PCA FEATURES

Even using a reduced number of rectangles in the final quantization, the calculation of SAI features is a relatively slow process. Since our target application is for very large multimedia archive (up to thousands of hours), we wanted to see if we could retain the performance of this type of feature but with simpler processing. The goal was to identify a simpler feature set that could capture information similar to the cross-correlations of the strobed temporal integration process used to produce SAIs. We decided to try a set of features based on subband autocorrelations. These features were based on earlier work in pitch tracking in our group [6], and consist of the first ten coefficients obtained from principal component analysis (PCA) on the normalized autocorrelations in each of 24 frequency subbands spanning center frequencies from 100 Hz to 1600 Hz with six bands per octave, and a Q of 8. Like the SAI features, we hoped these would capture some of the fine temporal structure not typically captured in traditional MFCC features. Unlike SAIs, the filterbank is time-invariant, and the correlation does not depend on any strobe instant selection. Analogously to the SAI rectangle features, we divided the 24 subbands into 4 (non-overlapping) frequency ranges, and vector quantized each of these 10×6 subband coefficient feature sets into 1000 codewords, for a total of 4000 dimensions. Figure 3 illustrates the entire calculation process for these features.

Figure 2 also includes the performance of this subband PCA (SBPCA) feature set compared to the full- and reduced-dimensionality SAI features. Although the SBPCA features show a slight drop in performance, they perform nearly as well as the SAI features. Significantly, calculation of these SBPCA features is about an order of magnitude faster than the SAI features: Both features are computed using reasonably optimized compiled C++ code, but SAI features can take, on average, $8\times$ longer than real time to calculate. In contrast, SBPCA features can be calculated in around $0.6\times$ real time. Especially when working with large amounts of data, this difference is enormous.

7. IMPROVEMENT WITH CLASSIFIER FUSION

At this point we have developed two sets of features that perform relatively comparably with traditional MFCC features, but are based on very different processing chains. In the past we have observed that feature sets capturing diverse information about the data will combine in a complementary way to produce a noticeable performance improvement. We therefore tried the same approach here, and used margin fusion (adding together the output decision value of each SVM classifier) to create classifiers based on different feature sets. We combined MFCCs with both the SAI and SBPCA features in this way. Figure 4 shows the performance of the three individual systems and the two combinations. Adding either auditory model feature to MFCCs gives a very substantial increase in mAP, with SBPCA features slightly better than SAIs. The baseline mAP performance of 0.34 for MFCCs alone improves to 0.40 in combination with SBPCAs, a relative improvement of around 18%.

8. DISCUSSION AND CONCLUSIONS

In the course of these experiments, we investigated a number of different approaches to video soundtrack classification. We draw several conclusions. Initially, we verified that SAI features perform well for audio classification, although they did not actually outperform traditional MFCC features in our scenario (which is significantly different from the isolated sounds used by [1, 3]). We observed that a standard machine learning technique (SVMs) significantly outperformed the PAMIR approach (although PAMIR may prove more useful on very large amounts of data where SVMs are infeasible). We demonstrated that the SAI feature dimensionality can be reduced significantly without significantly lowering performance. We discovered that a novel feature set, SBPCA, com-

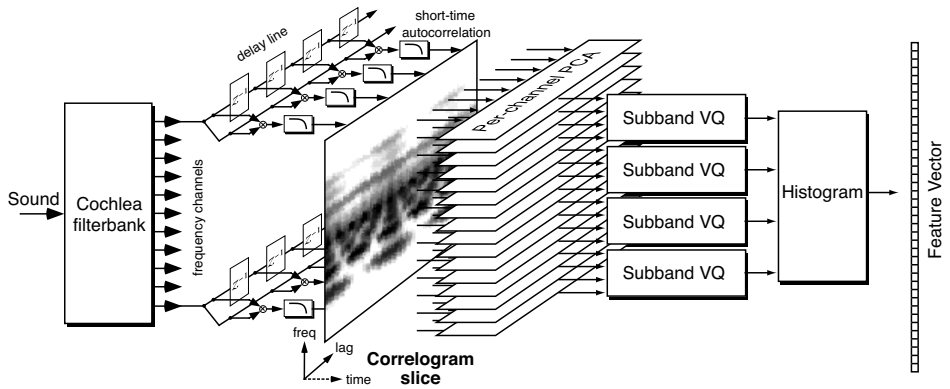

Fig. 3. Block diagram of the calculation of the subband autocorrelation PCA feature vectors.

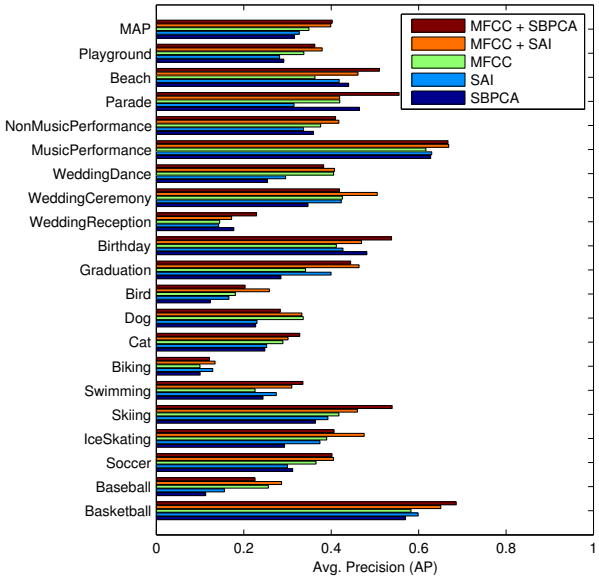

Fig. 4. SVM results with each individual feature set (MFCC, SAI, SBPCA) and margin fusion of MFCC with each of the 2 novel features.

pares favorably with SAI features but with significantly less processing overhead. Finally, we demonstrated that both SAI and SBPCA features can be combined with MFCC features for an overall performance improvement that is considerably better than our previous MFCC baseline. Since SBPCA features are reasonably fast to calculate (at least relative to SAIs), we believe that they are a promising direction to investigate for capturing information from fine temporal structure that is excluded from traditional feature. We believe this can significantly improve the performance of future audio classifier systems, especially when used in conjunction with more traditional features.

9. ACKNOWLEDGEMENT

Supported by the Intelligence Advanced Research Projects Activity (IAPRA) via Department of Interior National Business Center contract number D11PC20070. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of IARPA, DoI/NBC, or the U.S. Government.

10. REFERENCES

[1] R.F. Lyon, M. Rehn, S. Bengio, T.C. Walters, and G. Chechik, "Sound retrieval and ranking using sparse auditory representations," *Neural Computation*, vol. 22, no. 9, Sept. 2010.

[2] C. Cotton, D. Ellis, and A. Loui, "Soundtrack classification by transient events," in *Proc. IEEE ICASSP*, May 2011.

[3] R.F. Lyon, J. Ponte, and G. Chechik, "Sparse coding of auditory features for machine hearing in interference," in *Proc. IEEE ICASSP*, May 2011.

[4] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A.C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. ACM International Conference on Multimedia Retrieval (ICMR)*, Apr. 2011.

[5] Tom Walters, "Aim-c, a c++ implementation of the auditory image model," http://code.google.com/p/aimc/.

[6] B.-S. Lee and D. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proc. Interspeech-12*, Sept. 2012.