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 prior work on stabilized auditory image features, 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 15% improvement in mean Average Precision 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, automatic tagging and retrieval of multimedia content becomes increasingly important. Although much research has focused on the visual content of a video, modeling the audio content can also prove helpful [2, 3, 4, 5]. 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, and we have previously investigated a number of novel audio features for this task [6].

In [1, 7], 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 attempting to replicate their system as closely as possible to test it on a consumer video soundtrack retrieval task.

The next sections introduce our data/domain, and then describe our results using an available implementation of the auditory model front-end, and our modified, simplified features aiming to capture the same information. Sections 5 and 6 describe other experimentation with the original system, experimenting with replacing the original PAMIR retrieval model and with more common Support Vector Machine (SVM) classifiers, and with reduce the dimensionality of the representation. 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 [8]. This set of 9,317 video clips from YouTube comprises 210 hours of video. The clips are tagged with 20 semantic categories deemed relevant to consumers such as "beach" or "soccer". 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

Common audio features such as MFCCs start by calculating an average spectrum over a 20-30 ms window, immediately obliterating any variation at shorter timescales. Psychoacoustic results show, however, that listeners extract considerable information from this fine time structure, and we might expect that features reflecting this information will be a useful complement to MFCCs in statistical classification schemes. The system of [1, 7] employs an auditory model explicitly designed to capture this information in a stabilized image, constructed via a multi-step feature generation process. First

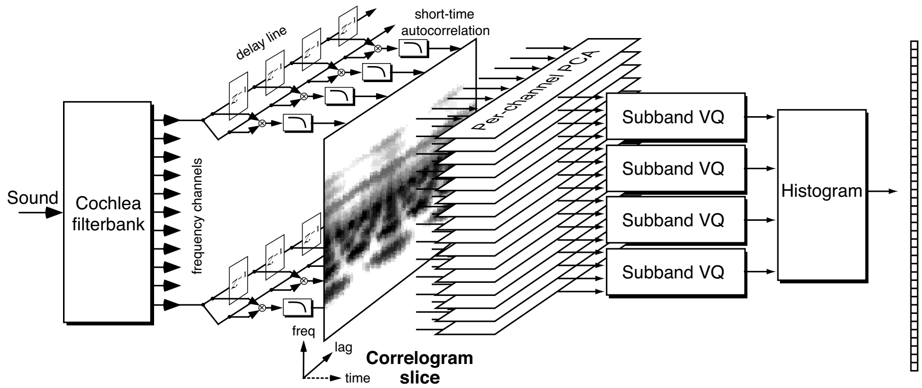

Fig. 1. Calculation of the SBPCA feature vectors.

the signal is passed through a time-varying filterbank modeling the cochlea, including local loudness adaptation through changes in individual filter resonance. The filterbank outputs are then integrated using so-called 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 (frequency channels $\times$ time lags) 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 4). In their experiments an SAI is generated every 20 ms to characterize the audio signal at that point. Each SAI is then overlaid with a set of rectangular patches of different sizes, each defining a local region of interest. The patches within each rectangle are collected over all data to build a separate vector quantization (VQ) codebook for each rectangle. A single SAI is then represented by a sparse code whose dimensionality is the number of rectangles times the size of each VQ codebook, and with just one nonzero element per rectangle. An audio clip is represented as the average of its SAI codes (essentially, a histogram).

To reproduce this system, we used a publicly-available C++ codebase, AIM-C [9], that computes stabilized auditory images that are closely related 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 the concatenation of the histograms of the VQ encodings over each of the 24 rectangles.

Table 1. Comparison of feature properties. Calculation times are over the 210 h CCV data set on a single CPU.

	MFCC	SAI (reduced)	SBPCA
Feature extraction	5.6 h	1087 h	310 h
Feature/patch dims	60	48	60
# patches/codebooks	1	24 (8)	4
Codebook size	3000	1000	1000
Histogram size	3000	24000 (8000)	4000

4. SUBBAND PCA FEATURES

As show in table 1 SAI feature calculation is almost 200$\times$ more expensive than for MFCCs, and around 5$\times$ slower than real time. Since our target application is for large multimedia archives (thousands of hours), we were strongly motivated to employ simpler processing. We reasoned that fine time structure could still be captured without the complexity of the adaptive auditory model and the strobing mechanism, so we explored features based on a fixed, linear-filter cochlear approximation, and conventional short-time autocorrelation, based on previous work in pitch tracking [12]. These features use Principal Component Analysis to reduce the dimensionality of normalized short-time autocorrelations calculated for a range of auditory-model subbands, so we call them subband PCA or SBPCA features. Figure 1 illustrates the entire calculation process for SBPCAs. First, a filterbank approximating the cochlea is used to divide the incoming audio into 24 subbands, spanning center frequencies from 100 Hz to 1600 Hz with six bands per octave, and with a quality factor $Q = 8$. In each subband, a normalized autocorrelation is calculated every 10 ms over a 25 ms window. The autocorrelation features are then run through principal component analysis (PCA) to yield 10 PCA coefficients per subband for every 10 ms of audio. Analogously to the SAI rectangle features, we then collect subbands into 4 groups of 6 bands each. For each 10 ms frame, we vector quantize each block of 6 (subbands)

$\times$ 10 (PCAs) into a 1000-entry codebook, yielding a 4000-dimension feature histogram.

Significantly, calculating and training the system with SBPCA features is much faster than using the SAI features. As shown in the final column of 1, SBPCA feature calculation is more than 3.5 $\times$ faster than SAI, even with some of the calculation still in Matlab (versus the all-C implementation of SAI we used). Not included in the table is the time spent training the SVM classifiers, since raw feature calculation dominated computation time. However, we note that the time it takes to learn k-means codebooks and compute histograms over them is a function of the number and size of the codebooks and (to a lesser extent) the dimensionality of the data points; these factors are listed for each feature set. Finally, the SVM training time is primarily a function of the dimensionality of the histogram feature used to characterize each video, since a distance matrix must be computed to create the kernel for the SVM; this size is the last row in the table. The distance matrix calculation takes a non-trivial amount of time, especially when using the chi-square distance as we do here (chi-square typically works well for computing distances between histograms but is much slower to compute than euclidean distance). Cumulatively, considering the raw feature extraction time as well as the larger set of codebooks involved, the SAI system can end up being an order of magnitude slower than the SBPCA system.

5. PAMIR VERSUS SVM LEARNING

Following [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 [10]. PAMIR is especially efficient to use on sparse feature vectors (such as the high dimensional histograms described above).

Although PAMIR is attractive for learning from very large datasets, our experiments consisted only of thousands, not millions, of data items, allowing us to use more expensive classifiers that achieved superior performance. Specifically, we used a support vector machines (SVM) with a radial basis function (RBF) kernel [11]. To compare PAMIR and SVM classifiers, we used a single split of the data: 40% training data, 50% test data; in the SVM case, we used the remaining 10% for parameter tuning.

Table 2 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 along with deltas and double deltas, for 60 dimensions. For consistency with the SAI features, MFCC frames were vector quantized and collected into a single 3000-codeword histogram representation for each video clip. The table 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 comparably, although MFCCs perform slightly better under both learning methods. Because of the clear superiority of SVM classification, we did not try the SBPCA features with PAMIR.

Table 2. Baseline system comparisons: MFCC and SAI features, in conjunction with both PAMIR and SVM learning methods. Results are averaged over the 20 categories of the CCV dataset.

Feature	Classifier	mAP%
MFCC	PAMIR	18.3
	SVM	34.9
SAI	PAMIR	14.7
	SVM	32.7

6. REDUCTION OF FEATURE DIMENSIONALITY

We were interested in investigating how the set of rectangle features selected influenced the final results. 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 likely redundant information. Our goal was to minimize the number of rectangles while maintaining high performance.

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 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. Table 3 compares the SAI 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 table also includes the performance of the SBPCA feature set, which despite a slight drop in performance, have performance very close to SAIs. All results use the SVM classifier.

7. IMPROVEMENT WITH CLASSIFIER FUSION

At this point we have developed two sets of features that perform nearly 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

Table 3. Comparison of SVM systems using the full-size SAI representation, a reduced-dimensionality version of SAI, and our computationally-simpler SBPCA features.

Feature	Rectangles	Histogram	mAP%
SAI	24	24000	32.7
SAI reduced	8	8000	32.7
SBPCA	4	4000	31.6

about the data will combine in a complementary way to produce a noticeable performance improvement. We therefore tried the same approach here, and used late fusion [13] (in this case, adding together the output decision value of each SVM classifier) to create classifiers based on different feature sets. We combined each of the three single feature classifiers (MFCC, SAI, SBPCA) with each other and also tried the combination of all three classifiers. Figure 2 shows the performance of the three individual systems and the four combinations. Adding either SBPCA or SAI features to MFCCs gives a substantial increase in mAP, with SBPCA features slightly more useful than SAIs. The baseline mAP performance of 0.35 for MFCCs alone improves to 0.40 in combination with SBPCAs, a relative improvement of around 15%. Combining SAI with SBPCA features performs better than either individually, but not as well as the combinations with MFCCs. Combining the margins of all three classifiers performs the best, with an mAP of 0.42, a 20% relative improvement over the MFCC baseline, but incurring the large computational cost of calculating SAI features.

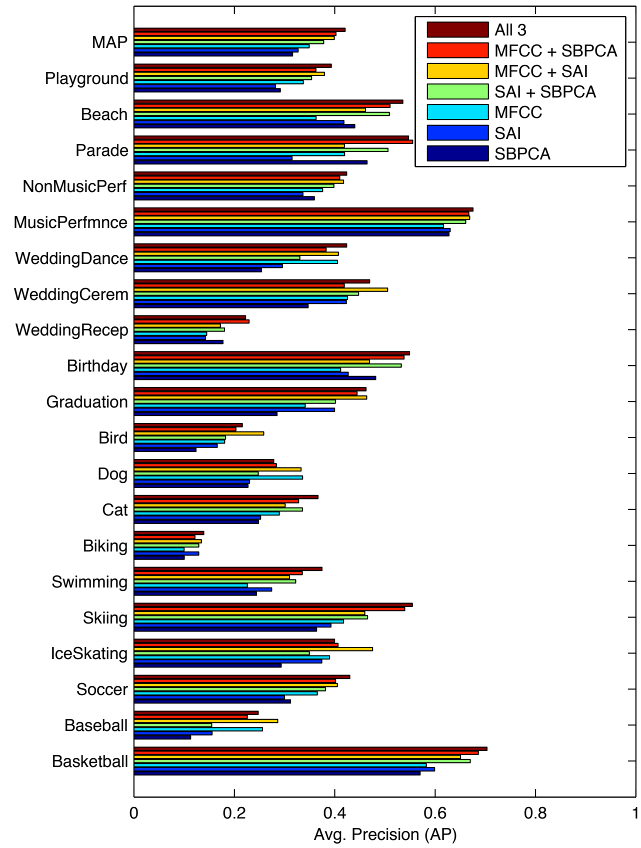

Fig. 2. SVM results with each individual feature set (MFCC, SAI, SBPCA), fusion of each pair, and fusion of all three.

8. DISCUSSION AND CONCLUSIONS

We investigated the use of fine-time-structure information in audio, such as that captured by the auditory model features of [1, 7], to the task of classifying real-world noisy environmental recordings. This task is significantly more demanding than the classification of individual sound effects used in earlier work. We verified that SAI features perform well for this task, equaling if not outperforming traditional MFCC features in our scenario. We observed that a standard machine learning technique (SVMs) greatly outperformed the PAMIR approach – although PAMIR may prove more useful on very large amounts of data where SVMs are infeasible. We also found that the SAI feature dimensionality can be reduced substantially without significantly lowering performance.

We have proposed a novel feature, SBPCA, that can capture fine-time information similar to that present in the SAI, and we show that its performance compares favorably with SAI but with significantly less processing cost. While the SAI has a closer correspondence to the processing of the auditory system, this fidelity was apparently not critical to the classification of video soundtracks.

Finally, we demonstrated that both SAI and SBPCA features can be combined with MFCCs for a substantial overall performance improvement. We note that the 15% relative improvement in mean Average Precision is double what we have achieved through combinations with novel features on similar tasks in the past [6]. Since SBPCA features are reasonably fast to calculate (at least relative to SAIs), they are a useful tool for capturing information from fine temporal structure that is excluded from traditional features, and can significantly improve the performance of future audio classifier systems when used in conjunction with traditional features.

9. ACKNOWLEDGEMENT

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20070. The U.S. Government is authorized to reproduce and distribute reprints for Government 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 expressed 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, pp. 2390–2416, Sept. 2010.

[2] L. Lu, H.J. Zhang, and S.Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia systems*, vol. 8, no. 6, pp. 482–492, 2003.

[3] K. Lee and D.P.W. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 18, no. 6, pp. 1406–1416, Aug 2010.

[4] H.K. Ekenel, T. Semela, and R. Stiefelhagen, "Content-based video genre classification using multiple cues," in *Proceedings of the 3rd international workshop on Automated information extraction in media production*. ACM, 2010, pp. 21–26.

[5] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li, "Youtubecat: Learning to categorize wild web videos," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 879–886.

[6] C. Cotton, D. Ellis, and A. Loui, "Soundtrack classification by transient events," in *Proc. IEEE ICASSP*, May 2011, pp. 473–476.

[7] R.F. Lyon, J. Ponte, and G. Chechik, "Sparse coding of auditory features for machine hearing in interference," in *Proc. IEEE ICASSP*, May 2011, pp. 5876–5879.

[8] 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, p. 29.

[9] Tom Walters, "AIM-C, a C++ implementation of the auditory image model," *http://code.google.com/p/aimc/*.

[10] D. Grangier, F. Monay, and S. Bengio, "A discriminative approach for the retrieval of images from text queries," *Machine Learning: ECML 2006*, pp. 162–173, 2006.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

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

[13] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 399–402.