BUILDING MUSICALLY-RELEVANT AUDIO FEATURES THROUGH MULTIPLE TIMESCALE REPRESENTATIONS

Philippe Hamel, Yoshua Bengio
DIRO, Université de Montréal
Montréal, Québec, Canada
{hamelphi,bengioy}@iro.umontreal.ca

Douglas Eck
Google Inc.
Mountain View, CA, USA
deck@google.com

ABSTRACT

Low-level aspects of music audio such as timbre, loudness and pitch, can be relatively well modelled by features extracted from short-time windows. Higher-level aspects such as melody, harmony, phrasing and rhythm, on the other hand, are salient only at larger timescales and require a better representation of time dynamics. For various music information retrieval tasks, one would benefit from modelling both low and high level aspects in a unified feature extraction framework. By combining adaptive features computed at different timescales, short-timescale events are put in context by detecting longer timescale features. In this paper, we describe a method to obtain such multi-scale features and evaluate its effectiveness for automatic tag annotation.

1. INTRODUCTION

Frame-level representations of music audio are omnipresent in the music information retrieval (MIR) field. Spectrograms, mel-frequency cepstral coefficients (MFCC), chromagrams and stabilized auditory images (SAI) are just a few examples of features that are typically computed over short frames. It has been shown that using frame-level features aggregated over time windows on the scale of a few seconds yields better results on various MIR tasks [2] than applying learning algorithms directly on frame-level features. However, the aggregation of frame-level features, also known as the bag-of-frames approach, does not model the temporal structure of the audio beyond the timescale of the frames. A simple method to get some information about short-time dynamics is to use the derivatives of the frame-level features. However, this method does not yield a representation that can model much longer temporal structure. Some alternative techniques to the bag-of-frames approach inspired by speech processing rely on the modelization of the temporal structure with models such as HMMs [12]. A representation that could jointly model the short-term spectral structure and long-term temporal structure of music audio would certainly improve MIR systems.

In this paper, we take a step to improve the bag-of-frames approach by combining a set of features computed over different timescales. The idea is that longer timescale features, by modelling temporal structure, will give some context to the shorter timescale features which model spectral structure. The combination of multiple timescales could yield a general representation of the music audio that would be useful to solve various MIR tasks relying on audio features. In particular, we will show that a simple classifier trained over a multi-scale spectral representation of music audio obtains state-of-the-art performance on the task of automatic tag annotation. The multi-timescale representation that we introduce in this paper has the advantage of being a general purpose scalable method that requires no prior knowledge of the spectral or temporal structure of music audio.

The paper is divided as follows. First, in Section 2, we describe the current state of the research on multi-scale representations. Then, in Section 3, we describe our experimental setup. In Section 4 we discuss our results. Finally, we conclude in Section 5.

2. MULTI-SCALE REPRESENTATIONS

Using representations at multiple scales allows much flexibility to model the structure of the data. Multi-scale representations offer a natural way to jointly model local and global aspects, without having prior knowledge about the local and global structures.

The idea of considering multiple scales is not new. It has been applied widely in the machine vision field. For example, pyramid representations [3] and convolutional networks [8] are just a few examples of multi-scale representations.

Recently, the MIR community as shown interest in taking advantage of multi-scale representations. Here are a few examples of recent work that has been done on multi-scale representation of music audio. Multi-scale spectro-temporal features inspired by the auditory cortex have been proposed in [11]. These features are used to discriminate speech from non-speech audio in a small dataset. In [10], structural change of harmonic, rhythmic and timbral features are computed at different timescales. This representation is used to build meaningful visualizations, although it has not been applied to music audio classifica-

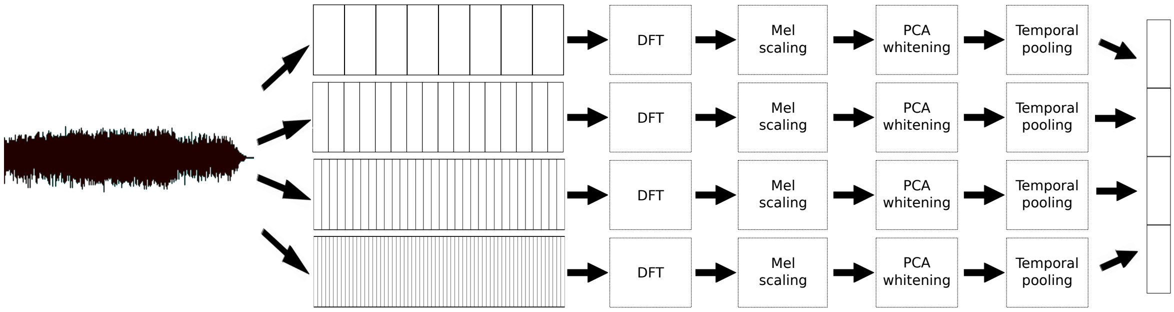


Figure 1: PMSCs are computed in parallel at different timescales.

tion. In [5], boosting is applied on features at different timescales to optimize music classification. Although the validity of this method is demonstrated, it does not obtain state-of-the-art results on the CAL500 dataset. Learning features jointly at different timescales obtains state-of-the-art performance for automatic tag annotation [6]. However this model still depends on a bag of short timescale frames to build the long timescale representation, limiting the potential to model temporal dynamics. Deep convolutional networks have been applied to genre recognition in [9]. The authors show that classification performance for genre recognition and artist identification can be improved by using an unsupervised deep convolutional representation instead of raw MFCC features. Unfortunately, the results presented in this work are not comparable to other work in the field. In [1], scattering representations of MFCCs have been shown to improve music genre classification. The performance reported are comparable to other results reported on the same dataset. A bag-of-system approach have been proposed in [4] to combine models at various time resolutions.

3. EXPERIMENTAL SETUP

We used the TagATune dataset [7] in our experiments. TagATune is the largest dataset for music annotation available for research that provides audio files. It contains over 20,000 30-second audio clips sampled at 22050 Hz, and 160 tag categories. Our train, valid and test datasets contained 14660, 1629 and 6499 clips respectively.

We used the area under the ROC curve (AUC) averaged over tags (AUC-tag) as our main performance measure. We also use the AUC averaged over clips (AUC-clip) and precision at k for comparison with other models. For more details on these performance measures, see [6].

3.1 Multi-scale Principal Mel-Spectrum Components

In our experiments, we used Principal Mel-Spectrum Components (PMSCs) [6] as base features. PMSCs are general purpose spectral features for audio. They are obtained by computing the principal components of the mel-spectrum. PMSCs have shown great potential for the task of music tag annotation.

Moreover, it is quite simple to compute PMSCs at different timescales. The time length of the frame used to compute the discrete Fourier transform (DFT) determines the timescale of the features. To obtain multi-timescale features, we simply need to compute a set of PMSCs over frames of different lengths (Figure 1). The smallest DFT window we used was 1024 samples (46.4 ms). The size of the timescales grew exponentially in powers of 2 (1024, 2048, 4096, etc.).

We keep the same number of mel coefficients for all timescales. Thus, longer frames are more compressed by the mel-scaling, since the dimensionality of the output from the DFT is proportional to the frame's length. However, mel-scaling is more important for high frequency bins, while low-frequency bins are barely compressed by the mel-scaling. Fortunately, these high frequencies are already represented in shorter timescales where they are less compressed. In our experiments, we used 200 mel energy bands.

In our experiments, we found that using the log amplitude of the mel-spectrum yields better performance than using the amplitude.

PCA whitening is computed and applied independently on each timescale. In order to circumvent memory problems when computing the PCA, we limit the number of frame examples by randomly sub-sampling frames in the training set. We typically used around 75 000 frames to compute the PCA. It is also worth noting that we preserve all the principal components since we don't use PCA for dimensionality reduction, but rather to obtain a feature space with an approximate diagonal covariance matrix. The PCA whitening step decorrelates the features, which allows a more efficient temporal aggregation.

The principal components obtained for different timescales are shown in Figure 2. For each timescale, the first few principal components (those that account for the most variance in the data) tend to model global spectral shape. Subsequent components then model harmonic structure in the lower part of the mel-spectrum, and as we go up in the coefficients (and lower in the accounted variance), the components model structure in higher frequencies. It is interesting to notice the periodic structure in the components which shows how the harmonics are captured by the components. Also, if we compare components between timescales, we can observe that components tend to model a larger part of the mel-spectrum and exhibit more structure in the lower frequencies as we go higher in the frame size.

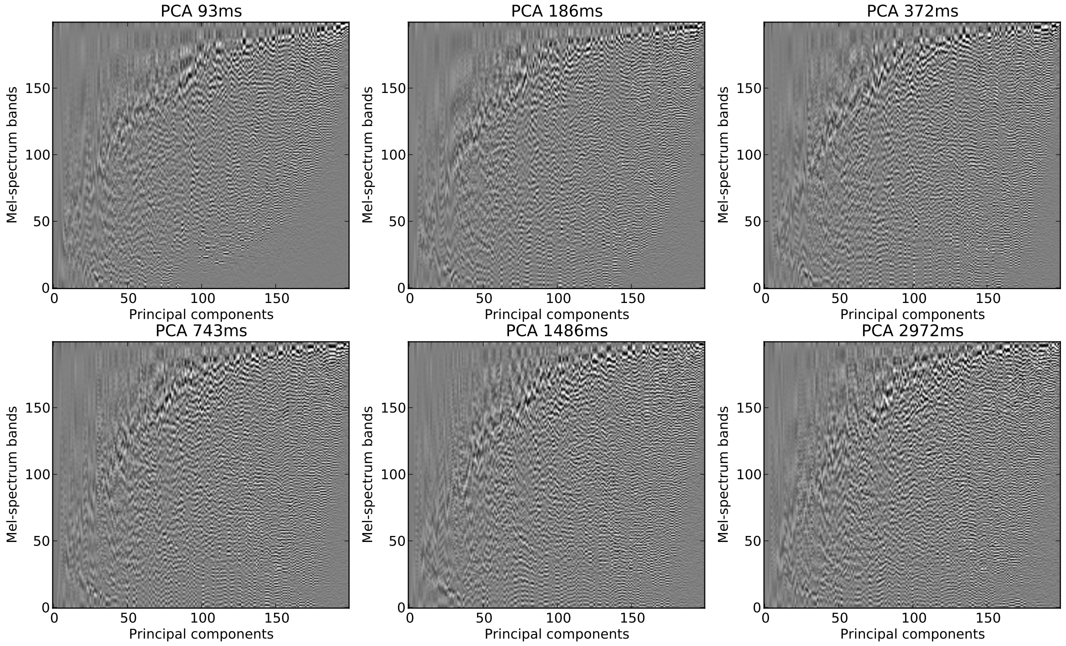


Figure 2: PCA Whitening matrices for different timescales. The first few principal components tend to model global spectral shape. Subsequent components then model harmonic structure in the lower part of the mel-spectrum, and as we go up in the coefficients, the components model structure in higher frequencies.

The next step consists of summarizing the features over a given time window by computing meaningful statistics. We refer to this step as temporal pooling. Following results from [6], we combined four pooling functions: mean, variance, maximum and minimum. These statistics are applied independently to each principal component through time and concatenated into a single feature vector for a given time window. In consequence, for each timescale we obtain a feature vector having four times the dimension of a single frame. Again, following results from [6], we fixed the pooling window at approximately 3 seconds for all experiments. Although, depending on how the frames were overlapped, this window length might vary for different timescales (see Section 4). The choice of the window length can be justified by the fact that 3 seconds would be enough for a human listener to label audio examples, but longer windows would give us less meaningful statistics for shorter timescales.

By concatenating the pooled features from each timescale, we obtain multi-timescale PMSCs (Figure 1).

3.2 Multi-Layer Perceptron

The classifier we used is similar as the pooled feature classifier (PFC) model presented in [6]. However, in our case, the input pooled feature vector will tend to be larger, since it is obtained by concatenating many timescales.

We used a one-hidden layer artificial neural network, also known as multi-layer perceptron (MLP), as the classifier for all experiments. We kept the size of the network constant at 1000 hidden units for all experiments. The number of parameters (weights) in the system varies depending on the dimensionality of the input.

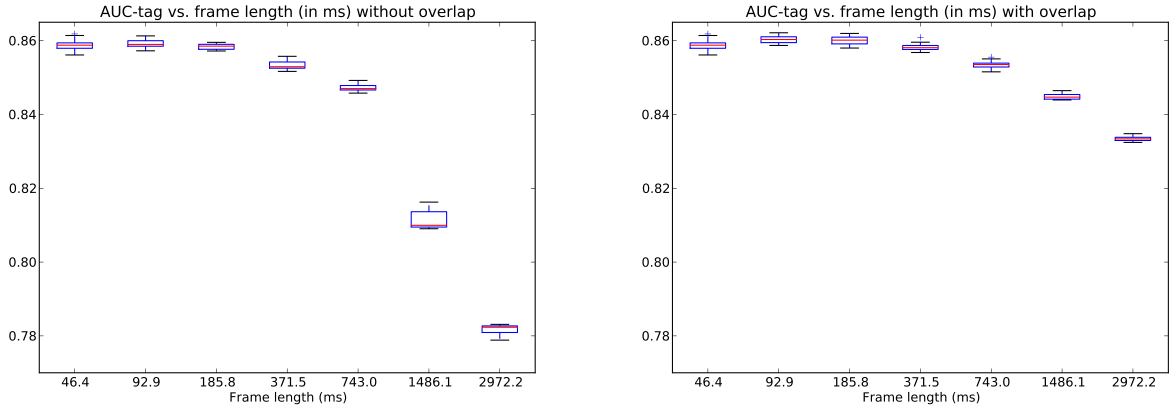
The input to the MLP is a multi-timescale PMSC representation a window of approximately 3 seconds of audio. In order to obtain tags for a full song in the test and validation phases, we simply average the MLP outputs over all windows from that song.

The MLP is well suited for multi-label classification like the music annotation task. The hidden layer acts as a latent representation that can model correlation between inputs as well as shared statistical structure between the conditional distributions associated with different targets (tags). This gives the MLP an advantage over other models such as the multi-class SVM, for which one would have to train a separate model for each tag. Also, the MLP scales sub-linearly in the number of examples, so it scales well to large datasets.

4. RESULTS

In our experiments, we evaluated the performance of different timescales individually, and their combination for the task of automatic tag annotation.

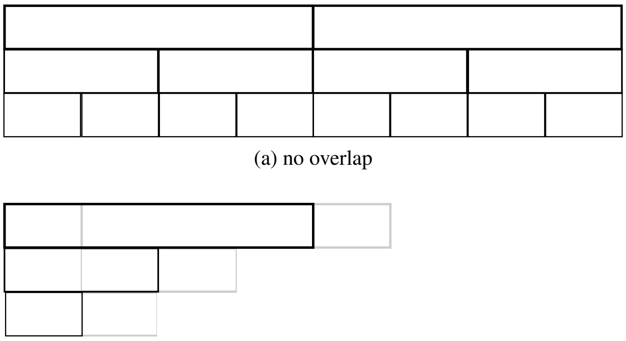
In our first experiment, for a given timescale, we did not overlap frames. In consequence, longer timescales have fewer frame examples. In the extreme case, the longest timescale is the size of the pooling window, meaning that the max, mean and min are all equal, and variance is zero. Obviously, this is not ideal. As we can see in Figure 3a, longer timescale perform worse than short timescales. However, we still see a significant advantage to using a com-



(a)

(b)

Figure 3: AUC-tag for single timescale features without overlap (a) and with overlap (b). Shorter timescales tend to perform better than longer timescales, and performance generally improve when using overlapped frames.



(a) no overlap

(b) overlap

Figure 4: Illustration of frames without overlap (a) and with overlap (b).

bination of timescales. In Figure 5a, we show the performance of multi-timescale features. We combined timescales incrementally, starting from the shortest one to the longest one. For example, the representation with two timescales combines 46.4ms and 92.9ms frames, the one with three timescales combines 46.4ms, 92.9ms and 185.8ms frames, etc.

In order to obtain more examples for higher timescales, and yield more meaningful statistics for the temporal pooling, we considered using more overlapping between windows. In our second experiment, we used the same frame step for all timescales, corresponding to the smallest frame length, in this case, 46ms (Figure 4). We include all frames that start within the pooling window in the temporal pooling. This means that the longest timescale frames will overflow beyond the pooling window length up to almost twice the window length. Even though this method will give us the same number of frames to aggregate for each timescale, the longer timescales will still have much more redundancy than shorter timescales. Longer timescales perform significantly better with more overlap than without overlap, as we can see by comparing Figure 3a and 3b. The overlap also gives a boost of performance when combining timescales (Figure 5b).

	Multi PMSCs	PMSCs	PMSCs + MTSL	MUSLSE
AUC-Tag	**0.870**	0.858	0.868	-
AUC-Clip	**0.949**	0.944	0.947	-
Precision at 3	**0.481**	0.467	0.470	0.476
Precision at 6	**0.339**	0.330	0.333	0.334
Precision at 9	**0.263**	0.257	0.260	0.259
Precision at 12	**0.216**	0.210	0.214	0.212
Precision at 15	**0.184**	0.179	0.182	0.181

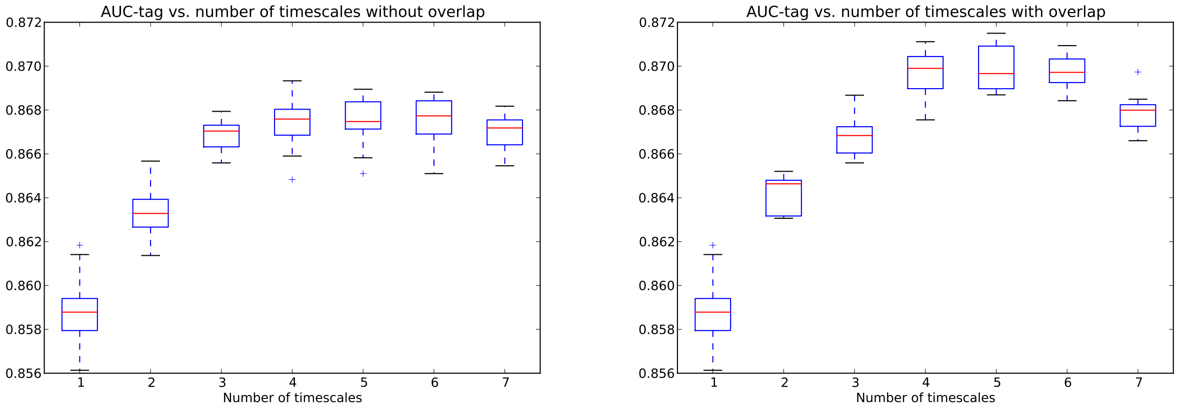
Table 1: Performance of different automatic annotation models on the TagATune dataset

In Table 1, we show the test performance of the model that obtained the best AUC-tag on the validation set. We compare with two other state-of-the-art models: Multi-timescale learning model (MTSL) [6] and Music Understanding by Semantic Large Scale Embedding MUSLSE [13]. The multi-timescale PMSCs trained with the MLP obtains the best performance on all measures. Moreover, this model is a lot faster to train than the MTSL. For the TagATune dataset, the training time would typically be a few hours for the MLP compared to a few days for the MTSL.

5. CONCLUSION

Multi-timescale PMSCs are general purpose features that aim at jointly modelling aspects salient at multiple timescales. We showed that, for the task of automatic tag annotation, using multi-timescale features gives an important boost in performance compared to using features computed over a single timescale. Moreover, with a simple classifier, we obtain state-of-the-art performance on the TagATune dataset.

Multi-timescale PMSCs could potentially improve the performance of more complex learning models such as MTSL or MUSLSE. They could most likely be useful for other music information retrieval tasks such as genre recognition, instrument recognition or music similarity as well.



(a)

(b)

Figure 5: AUC-tag in function of number of timescales used without overlap (a) and with overlap (b). The combination of timescales always include the shorter timescales. For example, the representation with 2 timescales combines 46.4ms and 92.9ms frames, the one with 3 timescales combines 46.4ms, 92.9ms and 185.8ms frames, etc.

Although the timescales used in these experiments are not long enough to model many aspects of the temporal structure of music, the combination of multiple timescales of analysis allows to model some mid-level temporal dynamics that are useful for music classification. It is also a improvement on the typical bag-of-frames approach. Even though we are still using frame level features, the concatenation of longer timescale representations puts short-time features in context.

In future work, it would be interesting to optimize the pooling window lengths independently for each timescale. This would allow longer timescale features to be aggregated over less redundant information and provide more relevant and stable statistics. It would also allow us to compute PMSCs over even larger timescales.

6. REFERENCES

[1] J. Andén and S. Mallat. Multiscale scattering for audio classification. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR'11)*, 2011.

[2] J. Bergstra. Algorithms for Classifying Recorded Music by Genre. Masters thesis, Université de Montréal, 2006.

[3] P. Burt and T. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Communications*, 9:4(532–540), 1983.

[4] K. Ellis, E. Coviello, and G.R.G. Lanckriet. Semantic annotation and retrieval of music using a bag of systems representation. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR'11)*, 2011.

[5] R. Foucard, S. Essid, Lagrange M., and Richard G. Multi-scale temporal fusion by boosting for music classification. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR'11)*, 2011.

[6] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR'11)*, 2011.

[7] E. Law, K. West, M. Mandel, M. Bay, and J. S. Downie. Evaluation of algorithms using games: the case of music tagging. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR'09)*, 2009.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1:541–551, December 1989.

[9] H. Lee, Y. Largman, P. Pham, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems (NIPS) 22.*, 2009.

[10] M. Mauch and M. Levy. Structural change on multiple time scales as a correlate of musical complexity. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR'11)*, 2011.

[11] N. Mesgarani, M. Slaney, and S. Shamma. Content-based audio classification based on multiscale spectro-temporal features. *IEEE Transaction on Speech and Audio Processing*, 2006.

[12] J. Reed and C.-H. Lee. On the importance of modeling temporal information in music tag annotation.

In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, pages 1873–1876, 2009.

[13] J. Weston, S. Bengio, and P. Hamel. Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval. *Journal of New Music Research*, 2011.