Towards coarse-scale event detection in music

Anna M. Kruspe
Fraunhofer IDMT
Ehrenbergstr. 31
98693 Ilmenau, Germany
kpe@idmt.fraunhofer.de

Jakob Abeßer
Fraunhofer IDMT
Ehrenbergstr. 31
98693 Ilmenau, Germany
abr@idmt.fraunhofer.de

Christian Dittmar
Fraunhofer IDMT
Ehrenbergstr. 31
98693 Ilmenau, Germany
dmr@idmt.fraunhofer.de

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract
Over the past years, the detection of onset times of acoustic events has been investigated in various publications. However, to our knowledge, there is no research on event detection on a broader scale. In this paper, we introduce a method to automatically detect "big" events in music pieces in order to match them with events in videos. Furthermore, we discuss different application scenarios for this audio-visual matching.

Keywords
Event detection, Musical events, Coarse-scale events

ACM Classification Keywords
H.5.5 Sound and Music Computing; I.5.4 Pattern Recognition - Applications

Introduction

In our recent research project *SyncGlobal*[1], we were faced with the task of automatically finding musical pieces to match user-provided videos. This is done in a number of semantic categories, such as mood, brightness, speed, etc. One of those categories is time structure. We aim to detect events both in the audio and the video material in order to find a proper temporal alignment. For videos, these events can be annotated by the user or extracted via detection [4]. For music, we aim to find these events automatically.

The first question here, of course, is "what is a musical event?". To our knowledge, there is no clear definition. For the "onset detection" tasks which are part of automatic music transcription systems, musical notes played by different instruments are thought of as the main events. For our purposes, we assume that an event is a big and fast perceptual change of the musical properties within an audio recording, either in a positive or in a negative direction. Such events could be a sudden change of loudness, a single very loud tone, the onset of a new, very present instrument, a change in style or genre, and similar properties.

We have developed a first prototype of such an event detection system. It is based on the band-wise log-loudness of the signal. We assume that big musical changes will be represented by a large change in at least one spectral band.

In the following, we first describe related work, then present our event detection system, and finally provide an outlook on the next steps for coarse-scale event detection.

Related work

As mentioned above, note onset detection can be interpreted as event detection on a much finer scale and has been the subject of many publications of the past year (e.g. [8][2][6]). It is also part of the ongoing *MIREX* competition[2]. Our proposed system follows some of the general ideas of [8], but applies them to a coarser temporal level. Many newer approaches are built on the assumption that onsets occur periodically over time. This is the opposite of what we require for our event detection system, since we specifically search for unexpected, unstructured musical events.

For non-music audio material, event detection of auditory scenes has also been a topic of research. The main application scenario are surveillance systems. Along with visual analysis, auditory analysis systems are used to detect potentially dangerous situations. Detected audio events often include breaking glass, gunshots, screaming, etc. Some examples can be found in [10], [3], and [1]. A competition for this kind of task has also been recently created [7]. We believe that this task is somewhat comparable to ours, but different in the sense that the differentiation between events and their surrounding audio material is not as clear-cut in our objective. Additionally, the expected types of events are often known in advance for event detection in auditory scenes. This knowledge can be used to train classifiers to specifically detect these kinds of events. We, however, make no prior assumptions about the types of musical events we are looking for.

[1] http://www.syncglobal.de/, Latest check: 05/13/13

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME, Last check: 05/13/13

Proposed system

Our system consists of a number of consecutive steps performed on the audio material. A broad overview is shown in Figure 1.

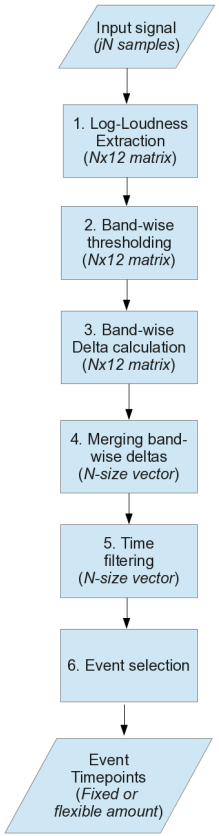

Figure 1: Overview over the event extraction steps

1. Log-loudness extraction First of all, we perform a Short-Term Fourier Transform (STFT) with a time resolution of 10ms on our signal. For the resulting spectrogram, the log-loudness is calculated and the results are summed up to represent twelve critical bands. The time resolution remains at 10ms. Band-wise processing provides a psycho-acoustic component [11], as does the logarithmic magnitude representation [9].

2. Thresholding In a next step, we threshold each band separately. In order to do this, we first generate a dynamic threshold curve. This is done by compressing the band signal, smoothing it slightly, and then decompressing it again. The compression is performed in a non-linear way.

Once we have obtained the dynamic thresholds for each band, we retain only those values of each spectral band that are above this band's threshold. We perform this step in order to minimize the influence of slower dynamic changes in the signal, since we are only looking for relatively fast loudness changes.

In addition to the compression step, we also tested smoothing the original band signals, and applying a global amplitude threshold. However, these steps degraded the final results. Signal smoothing might discard some of the important rapid changes in the band signals, while the offset value might cover up some big changes in loudness that still remain below the threshold.

3. Calculating deltas After thresholding the band signals, we calculate delta values between consecutive values for each band. However, some events are not represented by a very sudden change in loudness. A fast crescendo, for example, will show as quickly ramping up values in a succession of time frames. This is the biggest difference between our approach and onset de-

tection methods.
For this reason, we calculate delta values for different time contexts around each frame according to:

$$\Delta\hat{v}_k(t) = \frac{1}{2W}\Big|\sum_{\tau=0}^{\min(W,N-t)} v_k(t+\tau) - \sum_{\tau=\min(1,t)}^{\min(W,t)} v_k(t-\tau)\Big| \quad (1)$$

where W is the window size of the time context, v_k is the thresholded loudness vector for the kth band, and $t \in [0, N]$ is the current center frame index.
We perform this calculation for the time contexts 2 to 10 at every frame and keep only the highest resulting delta value for further processing.

4. Merging band-wise deltas The previous step produces a matrix of delta values for each time frame and for each spectral band. We compared different strategies for merging the band-wise values: First, we tried a simple mean and median calculation. Second, we used a voting strategy where a certain number of bands were required to be above a certain (magnitude) threshold. Finally, we looked at the maximum over all bands. We found that the latter approach was the one that was most perceptually salient. After all, users will probably perceive a strong event even if it only occurs in a part of the spectrum.

5. Time filtering Strong events might show some fluctuation over time. Therefore, we implemented a fixed timespan within which at most one event may occur. Using an overlapping moving window, all delta values within this timespan except the maximum delta are set to 0.

6. Selection of events After the previous step, we obtain a vector which describes how strongly the signal has changed at each timeframe and may therefore be interpreted as the likelihood of an event occuring at this point. We implemented different strategies to choose the final event timepoints:

- **Fixed number of events** In some scenarios, the same number of events for each musical piece might be required. We can then simply choose the timeframes of the highest values in our delta vector. For our purposes, a number of 10 to 20 events per musical piece often yields useful results.

- **Thresholding** A threshold between 0 and 1 can be used to only allow timepoints at which the delta vector is above this value. This is somewhat more flexible, as musical pieces with no perceptual events will not produce any events here, while the algorithm will also find more events for pieces with lots of changes.

Results

The algorithm was evaluated in a qualitative manner using a selection of commercial movie soundtrack excerpts, since we expect these pieces to have a number of present musical changes [5]. We annotated 26 excerpts which were between $30s$ and $60s$ in duration. We then performed our algorithm on these pieces and compared the results. We allowed for a tolerance of $1.5s$ between annotated and recognized events. The results are presented in Figure 2. For testing, we used the thresholding option explained above. Different thresholds were tested.
As the results show, about 47% of the detected events correlate with the annotated events. The best threshold value seems to lie between .55 and .75. At these

threshold values, we obtain Recall values of approximately 57%. Of course, this just presents a rough idea of the quality of our algorithm. As mentioned above, event annotations and expectations are highly subjective and may also depend on the application scenario.
For strongly percussive pieces, the algorithm sometimes found superfluous events as percussive onsets stand out spectrally. We tried to mediate this with the compression step, but it still happens for some pieces.
For events that ramp up or down slowly, events will often be found at some point during the progression, but not necessarily at the beginning or end of it.

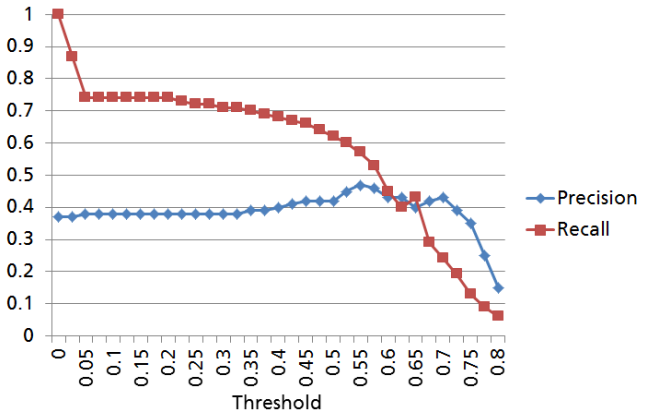

Figure 2: Average Precision and Recall results of our algorithm on 26 soundtrack excerpts for different thresholds. A tolerance of 1.5s was allowed.

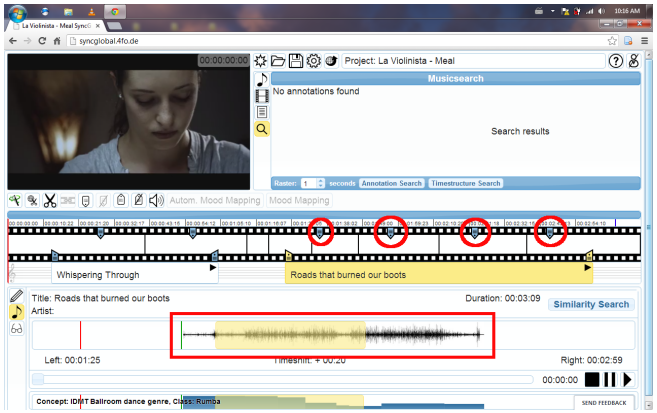

Figure 3: The *SyncGlobal* user interface. The user marks events on the video timeline (red circles). The algorithm then searches for musical pieces with matching events (red rectangle).

Conclusion

The developed event detection algorithm is currently being used in the *SyncGlobal* system for synch search. Users are presented with automatically detected events and can choose to match them up with video events, as shown in Figure 3. They can also search for songs with a similar event structure as their video.
As mentioned above, there is no clear definition yet as to what constitutes a coarse-scale musical event. We plan to conduct a listening test to answer this question empirically.
In this context, we consider our algorithm a first approach to automatically detect those events. After having gathered information on the definition of events, another listening test would be very helpful to determine the efficiency of our algorithm at finding such events. We aim to create a dataset with event annotations provided

by music experts in order to perform a quantitative evaluation of our algorithm. To our best knowledge, no such dataset exists at the moment.

Acknowledgments

This research work is a part of the *SyncGlobal* project. It is a 2-year collaborative research project between piranha womex AG from Berlin and Bach Technology GmbH, 4FriendsOnly AG, and Fraunhofer IDMT in Ilmenau, Germany. The project is co-financed by the German Ministry of Education and Research in the frame of an SME innovation program (FKZ 01/S11007).

References

[1] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli. Audio based event detection for multimedia surveillance. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 813–816, Toulouse, France, 2006.

[2] J. Bello, L. Daudet, S. Abdullah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, September 2005.

[3] C. Clavel, T. Ehrette, and G. Richard. Events detection for an audio-based surveillance system. In *International Conference on Multimedia and Expo (ICME)*, pages 1306–1309, Amsterdam, The Netherlands, 2005. IEEE.

[4] C. Cotsaces, N. Nikolaidis, and I. Pitas. Shot detection and condensed representation – a review. *IEEE Signal Processing Magazine*, 23(2):28–37, 2006.

[5] V. Dhandhania, J. Abesser, A. Kruspe, and H. Grossmann. Automatic and manual annotation of time-varying perceptual properties in movie soundtracks. In *Proceedings of the 9th Sound and Music Computing Conference (SMC)*, pages 461–466, Copenhagen, Denmark, 2012.

[6] S. Gao and Y. Zhu. A hmm-embedded unsupervised learning to musical event detection. In *International Conference on Multimedia and Expo (ICME)*, pages 334–337, Amsterdam, The Netherlands, 2005. IEEE.

[7] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. Plumbley. Detection and classification of acoustic scenes and events. Technical report, IEEE AASP Challenge, 2012.

[8] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 115–118, Phoenix, AZ, USA, 1999.

[9] B. Moore, B. Glasberg, and T. Baer. A model for the prediction of thresholds, loudness, and partial loudness. *J. Audio Eng. Soc.*, 45(4):224–240, 1997.

[10] J. Portelo, M. Bugalho, I. Trancoso, J. P. Neto, A. Abad, and A. Serralheiro. Non-speech audio event detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1973–1976, Taipei, Taiwan, 2009. IEEE.

[11] E. Scheirer. Tempo and beat analysis of acoustic musical signals. Technical report, Machine Listening Group, MIT Media Laboratory, 1996.