Decompositions of 2D Feature Representations with Applications to Acoustic Event Detection

Alessia Cornaggia-Urrigshardt, Frank Kurth

Fraunhofer FKIE
Communication Systems
Fraunhoferstr. 20
53343 Wachtberg
alessia.cornaggia-urrigshardt@fkie.fraunhofer.de
frank.kurth@fkie.fraunhofer.de

Abstract: In this paper we present an automatic procedure for detecting suddenly occurring events in a given audio signal. In particular, we are interested in events which show a quasi-periodic behavior such as recurring hands-clapping or sounds of a person knocking on a resonating surface. We exploit the analogy that impulse-like events such as the ones we want to detect have with percussive components in music: they both appear as vertical lines in the spectrogram. This property allows us to adapt techniques from Music Information Retrieval to our scope. In particular we perform a vertical/horizontal decomposition of the spectrogram to emphasize these vertical lines. In a further step, we detect the positions of these lines and consequently of the events we are interested in with the help of a novelty curve. Finally, the periodicity plays an important role in the process of discarding peaks of the novelty curve coming from background noise or sounds which we do not want to detect.

1 Introduction

In the last years, acoustic event detection has attracted the attention of the scientific community. In particular, in the field of audio surveillance, methods have been implemented in order to detect gunshots in noisy environments ([CER05], [VGT$^+$07]), detect and localize screams [VGT$^+$07] and recognize shouts, knocks and footsteps [AMK06]. As acoustic event detection does not require a line-of-sight between the sensor and an emitter, it is ideally suited to complement other kinds of sensors such as cameras.

The task of robust event detection in real outdoor environments is usually hard, due to an often significant amount of noises. Common noises are for examples birds and wind, but also traffic noises and people talking. However, in some cases, the sounds to be detected show a particular structure, and this can be exploited in order to improve the detection performance.

In this paper, we focus on the detection of recurring burst-like events, i.e impulse-like events which happen more than once and are repeated in regular intervals. The recurring events which we want to consider are, for example, knocking sounds or clapping of hands.

Possible application scenarios could include rescue operations in case of earthquakes or similar natural disasters in order to help finding possible survivors under collapsed buildings, under the assumption that such people are trying to attract attention by creating repetitive sounds like knocking. In literature, a general technique for detecting repeated events has, e.g., been proposed by Damm et al. [DvZO$^+$12].

Here, we follow a different approach. In order to detect burst-like repeated events we can exploit techniques known from Music Information Retrieval (MIR). One of the related tasks in MIR is the estimation of rhythm and of notes onsets in music pieces. There is a strong analogy with our application scenario, given by the fact that percussive components – which also determine rhythm – and note onsets are characterized by burst-like events mostly of repetitive nature. The overall goal is to develop an automatic procedure which takes an audio signal as an input and returns a list of detected audio events along with their start time, end time, duration and number of repetitions.

As an introductory example of the application scenario we assume a person to beat *repeatedly* on a surface in more or less *regular intervals*. We also assume that this person will perform several sequences of beats having some breaks in between. We refer to these beat trains with *blocks*. An illustrative example of this scenario is provided in Figure 1.

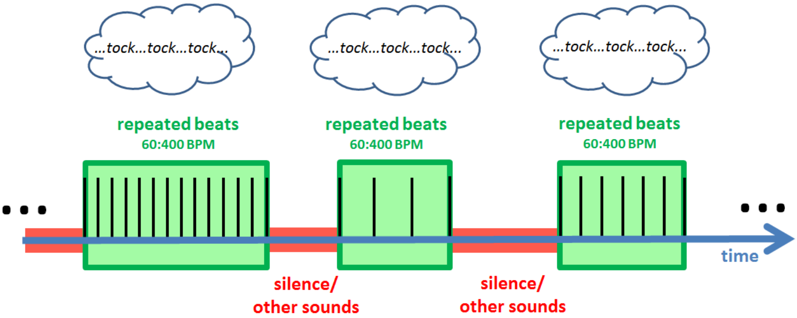


Figure 1: Illustration of a typical scenario faced by the algorithm.

The approach to detect recurring events followed in this work is to use a decomposition of a 2D-audio feature representation of the input signal. This is summarized in Figure 2. The signal is represented by its spectrogram, which, as a preprocessing step, is decomposed into two parts according to an algorithm described by FitzGerald [Fit10]. As a result, one of these parts contains emphasized vertical components. These vertical components represent the interesting events which have to be detected. The idea is then to use a novelty curve on this percussion-enhanced spectrogram to retrieve the time position of these events. A better quality of the detection can finally be achieved using methods to estimate the periodicity of the events.

More specifically, the contribution of our paper to the detection of pseudo-periodic burst-like events consists in the following: We combine several techniques (mentioned above) which in the literature have been mostly applied to music audio signals and we adapt them

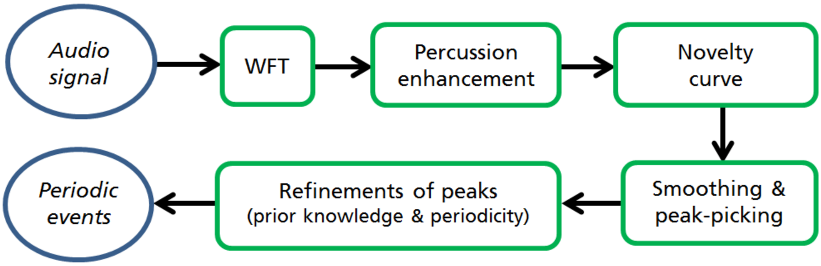


Figure 2: Overview of the algorithm presented in this paper. The input is an audio signal and the output a list of periodic events with their positions.

to our target scenario. Furthermore we have tested several novelty curves and developed a method to extract the events we are interested in from it. Finally, we have created a Graphical User Interface to enable a user to have an overview of the parameters, choose them and visualize the result of the algorithm.

The rest of this paper is organized as follows. Sections 2 to 5 describe each step in more detail: Section 2 explains how we make a percussion enhancement using a separation of horizontal and vertical components in the spectrogram. Section 3 gives an overview of the different kinds of novelty curves we have implemented together with the smoothing of the novelty curve, the peak-picking and the refinement of the peaks. Section 4 explains how the periodicity of the events has been estimated and exploited for our scope. Furthermore, Section 5 reports the results of some of the test we have conducted to evaluate the implemented algorithm and finally some conclusions and further steps are included in Section 6.

2 Percussion Enhancement

In order to enhance percussive elements, which corresponds to enhancing the events we are interested in in the spectrogram, we apply the method presented by FitzGerald [Fit10]. This method uses a rather easy and fast technique to separate harmonic and percussive elements in a music audio signal.

The main idea is that, thanks to their structure in the spectrogram, percussive events can be regarded as “outliers across time in a given frequency bin” [Fit10]. For this reason, the author applies two median filters on the spectrogram of the audio signal: a horizontal one to eliminate the spikes given by percussive events and get the harmonic part of the signal and a vertical one to extract the percussive components. In this paper, we are only interested in the percussion-enhanced spectrogram. The steps of FitzGerald’s algorithm are summarized in Figure 3. After the percussion-enhanced spectrogram has been calculated via a Windowed Fourier Transform (WFT), this is used to build a mask – binary or based on Wiener filtering – that is multiplied with the original complex-valued spectrogram. The resulting complex-valued spectrogram is the one we use in the following steps of our algorithm.

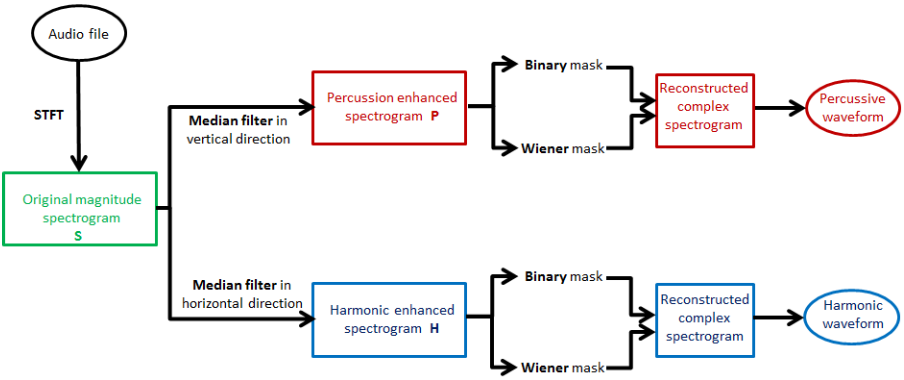

Audio file
STFT
Original magnitude spectrogram S
Median filter in vertical direction
Median filter in horizontal direction
Percussion enhanced spectrogram P
Harmonic enhanced spectrogram H
Binary mask
Wiener mask
Binary mask
Wiener mask
Reconstructed complex spectrogram
Reconstructed complex spectrogram
Percussive waveform
Harmonic waveform

Figure 3: Summary of the algorithm presented by FitzGerald in ([Fit10]). The circles indicate the time-domain, while rectangles indicate the time-frequency domain.

FitzGerald also reports that the length of the median filter is optimally restricted to odd numbers in the range from 15 to 30. If we consider that in his experiments he uses a FFT size of 4096 and a hop size of 1024 for a file with a sampling frequency of 44100 Hz, this corresponds to a length of the horizontal filter in the interval from 350 ms to 700 ms and a length of the vertical filter in the range from 161.5 Hz to 323 Hz. We mainly follow his suggestion, mostly using a length of 17, which for us corresponds to a horizontal filter of 400 ms and a vertical filter of approximately 366 Hz – taking for example a window of 2048 samples and a step size of 1024. A longer median filter could, in some cases, lead to a better distinction between harmonic and percussive components but at the cost of a longer running time. This is not acceptable if the algorithm is supposed to be used also for real-time processing. For more mathematical details we refer to [Fit10].

3 Reduction to Novelty Curve and Post-processing

Relating the application considered in this work to problems addressed in literature, a very close task is the one of *onset detection* [BDA+05]. If we consider the so-called ADSR curve, which divides the envelope of a note into four intervals (attack, decay, sustain and release), then the onset can be described as the point in which the attack phase starts.

In the literature, onsets are often detected by so called novelty curves [Foo00], also called detection functions [BDA+05]. They capture changes of a certain property of a signal, for example its energy, in order to facilitate the detection of these changes. A novelty curve is mostly under-sampled with respect to the original signal, which makes it possible to analyze it faster.

We calculate the novelty curve, as it is often done in literature, based on the spectrogram. A main difference of our approach to what has been proposed until now is that instead of the regular spectrogram, the percussion-enhanced spectrogram is used. For our algorithm, several types of spectral-based novelty curves have been studied and tested in order to find the one that best fits our scope.

We focus partially on novelty curves that emphasize the higher frequencies of the spectrum. Here, the changes caused by the transients are more noticeable, while the lower frequencies usually contain the energy of the signal [RJ01], [BDA$^+$05]. The following novelty curves have been tested in our experiments:

1. *High Frequency Component* (HFC): Four variations of this approach ([BDA$^+$05], [Mas96] [MB96],[Col05] [JA03]) are included in the comparison.
2. *Spectral Difference* (SD), also called Spectral Flux: Three modifications of this approach ([Col05] [JA03], [BDA$^+$05] [DSD02], [Mas96] [Dix06]) are considered.
3. *Measure of Transient* (MoT) [Mas96].

For the rest of this section, the following notation, taken from [GM11], is used: $X = (X(k,t))_{k\in[1:K],t\in[1:T]}$ indicates the spectrogram of the audio signal calculated using the WFT, where $X(k,t)$ is the kth Fourier coefficient for time frame t. k can be seen as the frequency bin index of the FFT (Fast Fourier Transform)-array. Note that if a window of even length N is used for the WFT, then $K = N/2$.

3.1 High Frequency Component

High Frequency Component (HFC) is a widely used novelty curve. The general formula to calculate it can be expressed as follows:

$$\text{HFC}(t) := \sum_{k=1}^{K} W_k \left|X(k,t)\right|^2 . \tag{1}$$

As can be observed from the above formula, the contribution of each frequency bin is weighted by a scalar value W_k. Three variations of (1) are obtained by giving different values to the weights:

a. $\forall k$: $W_k = 1$ ([BDA$^+$05]): In this case, as reported by Bello et al., the novelty curve becomes a measure of the local energy; the emphasis of the higher frequencies is indeed lost.

b. $W_k = k$, $\forall k \in [1 : K]$ ([Mas96], [MB96]): Masri et al. introduce a weight k in proportion to the frequency to give more importance to the higher coefficients and, as a consequence, to the higher frequencies. Masri also discards the lowest two bins in order to “avoid unwanted bias from DC component”.

c. $W_k = k^2, \forall k \in [1 : K]$ ([BDA$^+$05]): The energy of the local derivative of the signal is considered here.

A last modification of the HFC novelty curve has been proposed by Jensen et al. [JA03] as well as by Collins [Col05]. In this case the magnitude of the spectrogram is not squared, but only the weight k is squared:

$$\mathrm{HFC2}(t) := \sum_{k=1}^{K} k^2 \left|X(k,t)\right|. \tag{2}$$

According to the literature already mentioned and as observed in the experiments conducted for this work, HFC performs well in the presence of percussive sounds, presenting strong peaks at the onsets of these sounds, which, if the level of background noise is not too high, can be well distinguished and detected via adequate peak-picking techniques. Finally, note that any probability density function (pdf) might be used as well as a weighting function. In fact, depending on the shape of the pdf, certain frequency areas can be emphasized.

3.2 Spectral Difference

Novelty curves known by the names of Spectral Difference (SD) or Spectral Flux take into consideration the changes of the spectrogram over time. This kind of novelty curve is calculated considering distances between two successive FFT-vectors in the spectrogram. Also in this case several detection functions have been analyzed, the difference lying mainly in the way in which the distance is defined.

a. Collins [Col05] describes a novelty curve based on the HFC2 (equation 2), described above. This particular function has been introduced by Jensen et al. [JA03] and is defined as follows:

$$\mathrm{SD}(t) := \mathrm{HFC2}(t) - \mathrm{HFC2}(t-1). \tag{3}$$

b. In the works of Duxbury [DSD02], also studied by Bello et al. [BDA$^+$05], the measure of distance is the L_2-norm, used on the rectified difference:

$$\mathrm{SD}_2(t) := \sum_{k=1}^{K} \{H\left(|X(k,t)| - |X(k,t-1)|\right)\}^2 \tag{4}$$

where

$$H(x) := \frac{x+|x|}{2}. \tag{5}$$

c. Similarly to b., Dixon [Dix06] as well as Masri [Mas96] report the use of the L_1-norm instead of the L_2-norm, which results in:

$$\mathrm{SD}_1(t) := \sum_{k=1}^{K} H\left(|X(k,t)| - |X(k,t-1)|\right). \tag{6}$$

3.3 Measure of Transient

In his PhD thesis, Masri [Mas96] describes the Measure of Transient (MoT) detection function: a novelty curve based on the HFC (equation 1). Here, after the calculation of the HFC novelty curve (discarding the first two bins), a further step is introduced: two consecutive frames are analyzed and combined in the following way:

$$\mathrm{MoT}(t) := \frac{\mathrm{HFC}(t)}{\mathrm{HFC}(t-1)} \cdot \frac{\mathrm{HFC}(t)}{E(t)}. \tag{7}$$

In the formula above, t indicates the current frame, $t-1$ the previous one and $E(t)$ the energy function (evaluated at t) defined as:

$$E(t) := \sum_{k=1}^{K} |X(k,t)|^2. \tag{8}$$

3.4 Smoothing and Thresholding

As a first step, the novelty curve is smoothed via a median filter, then a threshold function is calculated. This threshold is later on subtracted from the smoothed novelty curve. Finally, to get the thresholded smoothed novelty curve, the maximum between this distance and 0 is taken.

In more detail, $m(t)$ is defined to be the *median filter* of length l applied to the absolute value of the normalized novelty curve:

$$m(t) := \mathrm{median}\{|\mathrm{NC}(j)| \mid j \in \mathbb{N},\ t-r \le j \le t+r\},\quad r = \begin{cases} \frac{(l-1)}{2} & \text{if } l \text{ odd,} \\ \frac{l}{2} & \text{if } l \text{ even.} \end{cases} \tag{9}$$

A *threshold curve* $\delta(t)$ is then defined as

$$\delta(t) := m(t) + \tau_1 \cdot \overline{m}, \tag{10}$$

where $\tau_1 \in [0,1]$ and $\overline{m}$ denotes the *mean* defined as

$$\overline{m} := \frac{1}{T} \sum_{i=1}^{T} m(i). \tag{11}$$

The threshold τ_1 has to be chosen carefully according to the application: The larger its value is, the more peaks with small amplitudes will be cut away.

The *thresholded smoothed novelty curve* $\mathrm{TNC}(t)$ is finally calculated as

$$\mathrm{TNC}(t) := \max\{0, \mathrm{NC}(t) - \delta(t)\}. \tag{12}$$

3.5 Peak-picking and Peak Refinement Steps

Once the novelty curve is calculated and smoothed, we need to extract its maxima. In this work we have decided not to develop a new peak-picking technique, but to apply a simple algorithm and then refine the detected peaks using other techniques. Firstly, all the peaks of the smoothed novelty curve are extracted and saved in a list, which is subsequently refined in several steps. These steps are schematically represented in Figure 4. At the beginning, all the peaks of the smoothed novelty curve are extracted. For the elimination of the peaks with a low amplitude, a threshold is used. The threshold is based on the mean amplitude of the peaks. Mathematically, let p be a vector containing the amplitudes of all the *extracted peaks*. Then p is a vector

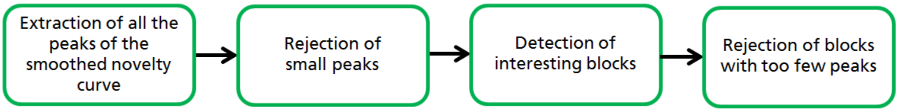

Extraction of all the peaks of the smoothed novelty curve
Rejection of small peaks
Detection of interesting blocks
Rejection of blocks with too few peaks

Figure 4: Overview of the peak refinement process.

$$p = (p_1, ..., p_h), \tag{13}$$

where h is the number of extracted peaks.

A peak p_i is *accepted* if

$$p_i > \tau_2 \cdot \overline{p}, \qquad \text{with} \qquad 1 \leq i \leq h, \tag{14}$$

where $\tau_2 \in [0, 1]$ and $\overline{p}$, in analogy to $\overline{m}$, is the mean $\overline{p} := \frac{1}{h} \sum_{i=1}^{h} p_i$. If $p_i \leq \tau_2 \cdot \overline{p}$, the candidate peak p_i is rejected. The choice of τ_2 has an impact on the amount of accepted peaks: If τ_2 grows, more peaks will be rejected and vice versa.

The next step consists of detecting blocks inside the audio signal (technically in the list of peaks of the novelty curve). Let $\tilde{p}$ be a vector $\tilde{p} = (\tilde{p}_1, ..., \tilde{p}_h)$ of length h containing the *time positions* of the peaks of vector p. The entry p_i in p corresponds to the entry $\tilde{p}_i$ in $\tilde{p}$, for $1 \leq i \leq h$, such that there is a peak with amplitude p_i and position $\tilde{p}_i$. At this point a vector d of length $h - 1$ is defined by

$$d_j := \tilde{p}_{j+1} - \tilde{p}_j, \qquad \text{for} \qquad 1 \leq j \leq h - 1, \tag{15}$$

which means that every entry of this vector denotes the *distance* (i.e. time lag) between two consecutive peaks. In order to find interesting blocks – which are supposed to contain relevant events – we consider the vector of distances d. Before running the algorithm, one has to decide which is the minimum and maximum distance between two events to be considered relevant, and which is the minimum length between two consecutive blocks. Let this minimum distance be called γ. To have an *interesting block*, the condition $d_j \geq \gamma$ has to be fulfilled. In this case, $\tilde{p}_j$ is the time position at which the previous block finishes and $\tilde{p}_{j+1}$ is the position in which the new block starts. The first block is supposed to start with the first peak detected and the last block is supposed to end with the last peak detected. Isolated peaks can be seen as a borderline case, i.e. a block with only one peak.

The last step consists of eliminating those blocks which contain too few peaks to be interesting. The minimum amount of peaks in one block is also part of the prior knowledge, which varies according to the application and is set automatically according to the range of BPM desired before running the algorithm.

What remains at the end of the peaks refinement process is a modified list in form of a vector p, in which the amplitudes of the relevant peaks are contained, as well as a modified vector $\tilde{p}$, where the positions of these peaks are stored.

4 Period Estimation

The estimation of the repetition rates (equivalently, the periods) of the different series of interesting events is used to improve the detection of these events. Here the list containing event positions is improved by rejecting all the peaks coming from events that do not fit the predominant period of each block.

To estimate the periodicity of the events in the different blocks, we use a technique presented by Grosche and Müller [GM11]. They describe a method, which they apply in music analysis, that works well also with changes of tempo in the same piece of music. We use their method not in substitution of a peak-picking technique as in the original work, but to support and improve the quality of our list of extracted peaks.

The method described in [GM11] starts from the novelty curve and, through the use of a tempogram – a time-pulse representation – derives a Predominant Local Pulse (PLP) curve, which indicates the main local periodicity of the novelty curve. A significant advantage of the PLP curve is that it is invariant to outliers, which means that peaks given by external, non-interesting and non-periodic noise do not affect the extraction of the local periodicity. An example of how the tempogram and the PLP curve of an audio signal typical for our scenario look like is show in Figure 5.

For more details on how the tempogram and the PLP curve are calculated we refer to [GM11].

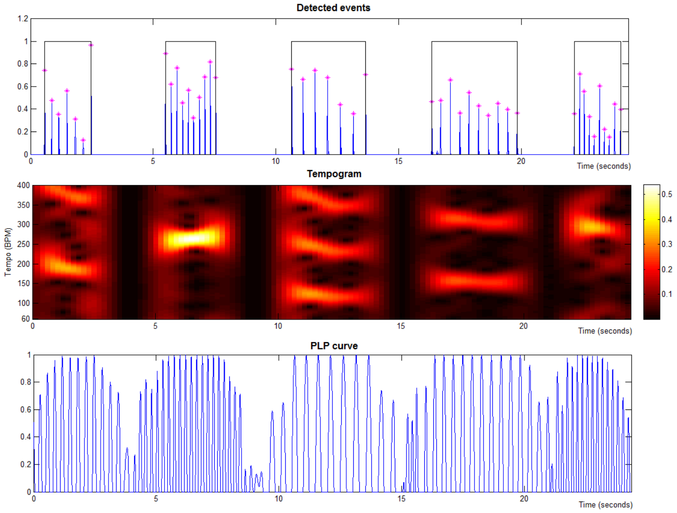


Figure 5: Example of detected events, the tempogram and the PLP curve applied on a segment of a real recording.

5 Tests and Evaluation

In order to test the algorithm presented in this work and its performance in different scenarios, several experiments have been conducted. The methods and the tests have been implemented in Matlab. We have performed different kinds of tests, firstly on synthetic audio files disturbed by noise and secondly on realistic outdoors recordings.

Taking into account the target scenario described in the introduction, we consider sequences containing from 60 to 300 beats per minute (BPM) to be valid blocks. This means, to be considered as valid, the maximum distance from one acoustic event to another is 1 second. Furthermore, the distance from one block to another one has to be larger than 1 second. These are parameters that can be considered to be prior knowledge. These can be changed by setting a different interval for the BPM, according to the application scenario faced in the particular occasion before running the program.

5.1 Results on Synthetic Audio Files

As an initial test scenario we considered synthetic audio files to which we have added white, pink and brown noise with increasing amplitude. It turns out that in this case noise is not misclassified by the algorithm as an interesting event. This means that the algorithm does not find any false positives (FP), which can be explained by the fact that noise is rather stationary. The detection performance for the case of pink noise is shown in Figure 6.

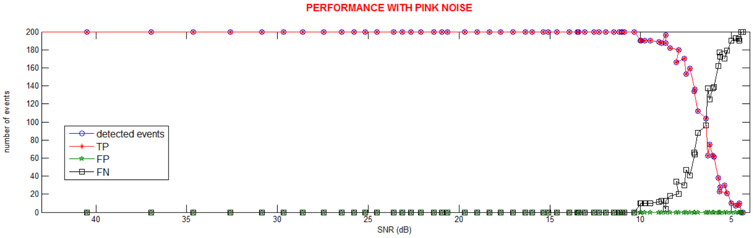


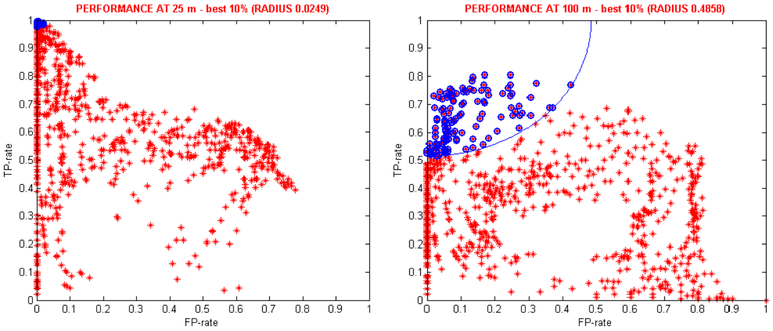
Figure 6: Overview of the performance of the algorithm on a synthetic audio file with pink noise.

5.2 Results on Realistic Recordings

After looking at the case of synthetic audio files, it is important to verify the performance of the algorithm on realistic recordings, i.e., audio files recorded in outdoor scenarios with background noises like wind, birds etc. We assume a situation like the one described in the introduction: A person is alternatingly beating a stone on a metallic object and screaming to attract attention. The microphone has been put at four different distances from the sound source (25, 50, 75 and 100 meters). The recordings contain about 200 beats and their length is mostly between the 90 and 120 seconds. It is important to see how the performance changes with increasing distance and whether the trains of beats can be distinguished from the screaming.

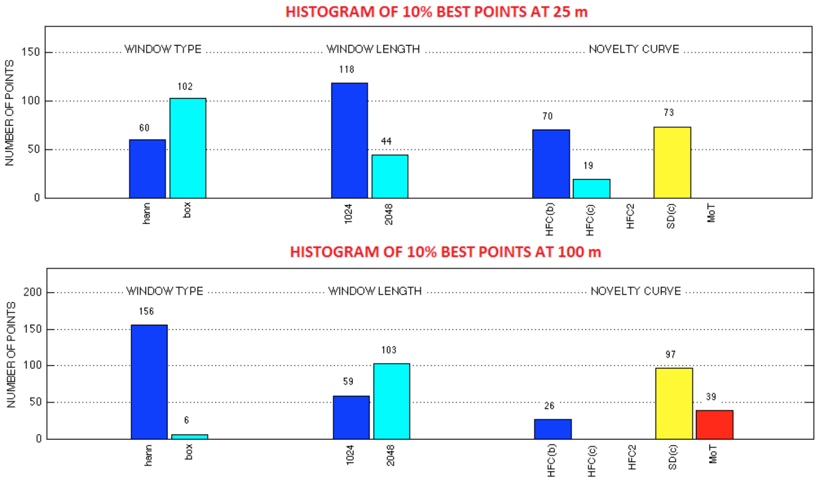
In order to evaluate the results of our algorithm, we have to compare the detected events with a ground truth. In this case the ground truth has been obtained by annotating the audio files manually using Sound Onset Labellizer (SOL), a Matlab annotation tool implemented by Leveau et al. [LDR04]. We have run the algorithm with several combinations of parameters and each time the TP-rate and FP-rate are calculated, which can be then interpreted as the coordinates of a point in the ROC plane. As a result, for each distance at which the microphone has been positioned, we have looked at the best 10% of these points to see where they lie in the Receiver Operating Characteristic (ROC) plane and how the combinations of parameters of these points look like. For each of the four cases the best 10% of points lie in the upper left triangle of the plane (within a maximal radius from the optimal point $p_{\text{opt}} = (\text{FP-rate}, \text{TP-rate}) = (0, 1)$ of 0.4858 in the case of 100 meters – where TP stands for rue positives). This indicates a good performance. In Figure 7 one can see, for the case of 25 and 100 meters, how the distribution of points looks like in the plane. The points indicated by a blue circle are the best 10%. The combinations of parameters leading to such points are also summarized in the form of a histogram. Considering all of the four distances (25, 50, 75, and 100 m), and observing the histograms, one may conclude, for example, that the Hann window performs mostly better and that the novelty curve HFC2 (eq. 2) never appears in the best combinations of parameters.

Furthermore, we have taken the outdoor recordings and disturbed them with non-periodic impulse-like events (events similar to the ones we want to detect). The performance of the algorithm, as expected, is not significantly affected by these non-periodic events. Isolated peaks are in fact not detected as interesting events. The only problem might arise in the case in which they occur much closer to the beginning or the end of a block: In this case the algorithm might confuse them as being part of the block.


PERFORMANCE AT 25 m - best 10% (RADIUS 0.0249)

PERFORMANCE AT 100 m - best 10% (RADIUS 0.4858)

Figure 7: Summary of the performance of the algorithm with different combinations of parameters at the distances of 25 and 100 meters.


HISTOGRAM OF 10% BEST POINTS AT 25 m

HISTOGRAM OF 10% BEST POINTS AT 100 m

Figure 8: Histograms indicating the distribution of parameters *window type*, *window length* and *novelty curve* for the best 10 % of the points illustrated in Figure 7 (in blue), at the distances of 25 and 100 meters.

The results in the case in which the knocking sounds and speech occur at the same time is also similar: The performance of the algorithm does not significantly decrease, or not at all. This is not particularly surprising, because speech, even if simultaneous with the knocking, does not have the characteristic periodic vertical shape in the spectrogram that we want to detect.

5.3 Utility of the Percussion Enhancement

Finally we want verify that percussion enhancement improves the performance of the algorithm. This can be seen in Table 1 which shows for each distance how far the point indicating the performance is from the point p_{opt}. In this case for each distance the best performing combination of parameters has been used.

Table 1: Performance of the algorithm with and without percussion enhancement step. This table shows, for each distance (25, 50, 75 and 100 meters), the distance of the point reflecting the performance of the algorithm in the ROC plane and the optimal point p_{opt}, i.e. values close to zero are close to the optimum.

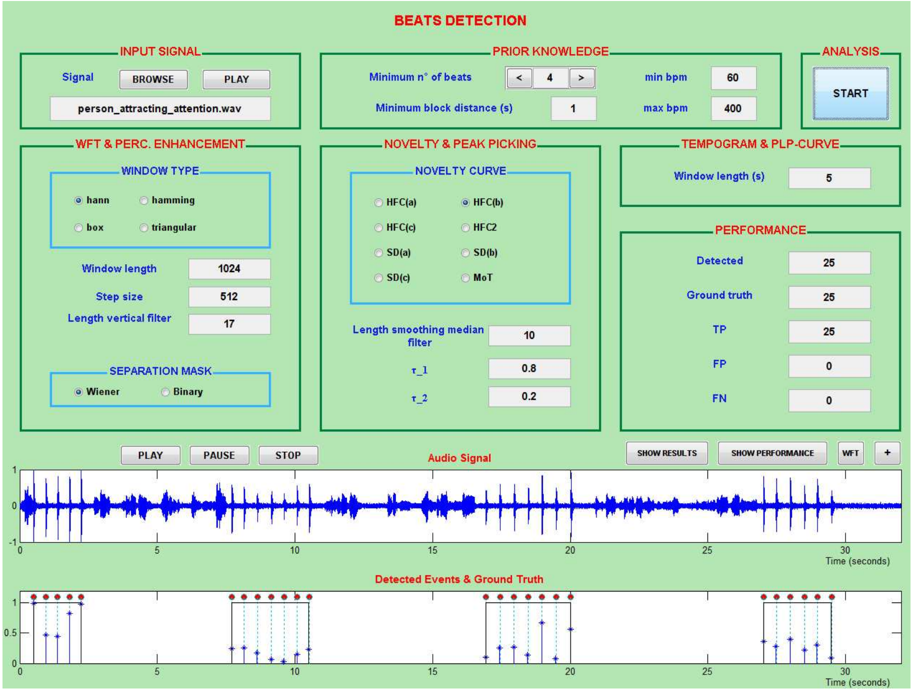
	WITH	WITHOUT
25 meters	0.0050	0.0508
50 meters	0.0111	0.0222
75 meters	0.1320	0.2521
100 meters	0.2672	0.8512

5.4 Graphical User Interface

For the algorithm presented we have created a Graphical User Interface (GUI) which is shown is Figure 9. One can select all the different possible parameters and, in the lower part of the GUI, the waveform of the selected audio signal is shown together with the result of the detection algorithm. The detected events are plotted as stems with a star, while in the same plot, the ground truth (if available) is indicated by stems with circles. One can also plot the WFT of the audio signal, which is then shown instead of the waveform. It is possible to separately play back the different sections of the file simply clicking on them (on a block or on the area between one block and the other one).

6 Conclusion

In this paper we proposed a method for detecting repeated (knocking) sounds. To achieve this goal we combined techniques from MIR, such as onset detection, with techniques for separating harmonic and percussive components in a spectrogram, in order to emphasize the events we want to detect. Later on we exploited the periodicity of these events to improve the detection performance. Leaving out the step of period estimation, this method could be also used to detect any kind of percussive sounds, such as slammed doors, mouse clicks or, more generally, every sound which is visible as a vertical line in a spectrogram. From our evaluations we conclude that the proposed algorithm detects the events we are interested in rather well.



Signal: person_attracting_attention.wav

Minimum n° of beats: 4

min bpm: 60

max bpm: 400

Minimum block distance (s): 1

● hann

Window length: 1024

Step size: 512

Length vertical filter: 17

● Wiener

● HFC(b)

Length smoothing median filter: 10

τ_1: 0.8

τ_2: 0.2

Window length (s): 5

Detected: 25

Ground truth: 25

TP: 25

FP: 0

FN: 0

Figure 9: Graphical user interface of the evaluation environment for the proposed algorithm.

Future work might focus on an extensive series of experiments considering more audio files to find out the best combination of parameters. Furthermore, other 2D-representations might be considered complementary to the WFT. Finally, we have based our detection on the concepts of blocks. Everything that is outside these blocks is not analyzed any more. In some cases this might lead to a loss of interesting events due to a loss of a complete block. A next step could consist of analyzing what we have gained in our experiments by including prior knowledge like the blocks.

References

[AMK06] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli. Audio based event detection for multimedia surveillance. In *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 813–816, 2006.

[BDA+05] J. P. Bello, L. Daudet, S. Abdallah, M. Davies C. Duxbury, and M. B. Sandler. A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13 (5):10351047, 2005.

[CER05] C. Clavel, T. Ehrette, and G. Richard. Events Detection for an Audio-Based Surveillance System. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Con-*

ference on, pages 1306 –1309, july 2005.

[Col05] N. Collins. A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions. In *In AES Convention 118*, pages 28–31, 2005.

[Dix06] S. Dixon. Onset Detection Revisited. In *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada, 2006.

[DSD02] C. Duxbury, M. B. Sandler, and M. Davies. A Hybrid Approach to Musical Note Onset Detection. In *Proc. Digital Audio Effects Conf. (DAFX,'02)*, pages pp. 33–38, Hamburg, Germany, 2002.

[DvZO$^+$12] D. Damm, D. von Zeddelmann, M. Oispuu, M. Häge, and F. Kurth. A system for audio summarization in acoustic monitoring scenarios. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1279 –1283, aug. 2012.

[Fit10] D. FitzGerald. Harmonic/Percussive Separation using Median Filtering. In *13th International Conference on Digital Audio Effects (DAFX10)*, Graz, Austria, 2010.

[Foo00] J. Foote. Automatic Audio Segmentation using a Measure of Audio Novelty. In *IEEE International Conference on Multimedia and Expo (I)*, pages 452–, 2000.

[GM11] P. Grosche and M. Müller. Extracting Predominant Local Pulse Information from Music Recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.

[JA03] K. Jensen and T. H. Andersen. Real-time beat estimation using feature extraction. In *Proc. Computer Music Modeling and Retrieval Symposium, Lecture Notes in Computer Science*. Springer Verlag, 2003.

[LDR04] P. Leveau, L. Daudet, and G. Richard. Methodology and Tools for the evaluation of automatic onset detection algorithms in music. In *In Proc. Int. Symp. on Music Information Retrieval*, pages 72–75, 2004.

[Mas96] P. Masri. *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, University of Bristol, December 1996.

[MB96] P. Masri and A. Bateman. Improved Modelling of Attack Transients in Music Analysis-Resynthesis. In *Proc. Int. Computer Music Conference*, 1996.

[RJ01] X. Rodet and F. Jaillet. Detection and modeling of fast attack transients. In *Detection and modeling of fast attack transients*, page pp. 3033, Havana, Cuba, 2001.

[VGT$^+$07] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti. Scream and gunshot detection and localization for audio-surveillance systems. In *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 21–26, Washington, DC, USA, 2007. IEEE Computer Society.