ON THE USE OF ZERO-CROSSING RATE FOR AN APPLICATION OF CLASSIFICATION OF PERCUSSIVE SOUNDS

Fabien Gouyon ⊥

Audiovisual Institute, Pompeu Fabra University, Barcelona

fabien.gouyon@iua.upf.es

François Pachet, Olivier Delerue

Sony Computer Science Laboratory, Paris

{pachet,delerue}@csl.sony.fr

ABSTRACT

We address the issue of automatically extracting rhythm descriptors from audio signals, to be eventually used in content-based musical applications such as in the context of MPEG7. Our aim is to approach the comprehension of auditory scenes in raw polyphonic audio signals without preliminary source separation.

As a first step towards the automatic extraction of rhythmic structures out of signals taken from the popular music repertoire, we propose an approach for automatically extracting time indexes of occurrences of different percussive timbres in an audio signal. Within this framework, we found that a particular issue lies in the classification of percussive sounds. In this paper, we report on the method currently used to deal with this problem.

1. INTRODUCTION

Most of the work on automatic audio descriptors focuses on 1) low-level or mid-level descriptors (see e.g. [4]) and 2) small or middle sized audio data, typically sounds (see e.g. [7] or [12]). In our ongoing project, we focus on high-level descriptors that describe music titles in a global fashion. Such global musical descriptors typically include tempo (see e.g. [9]), type of instruments, but also musical genre, rhythm type, etc.

Rhythm is acknowledged to be a fundamental dimension of music perception, a wide field of investigation in computer music concerns with transcription and understanding of rhythm (e.g. works by Peter Desain and Henkjan Honing); however, it is still a poorly understood phenomenon. Designing cognitive models on rhythm perception (e.g. [3]), producing acceptable notations from a list of onset times such as MIDI notes (e.g. [2]), and deriving from it musicological abstractions, like tempo and meter (e.g. [1]), are still unsolved problems and are clearly out of the scope of this paper.

Some works regarding the automatic transcription of percussive music exist (e.g. [11]), nonetheless, there exists no reference representation of rhythm that can be used for classification purposes. To produce such a representation, we believe that we need to extract from the audio signal occurrences of percussive timbres; the reason being that we target applications dealing with popular music, in which rhythm is a predominant feature that is mainly given by a particular set of timbres: the drum sounds.

The problem we address here is precisely the classification of percussive timbres into two classes: snare-like and bass drum-like sounds, as found in popular music titles. We now give a short overview of the detection scheme used to provide time indexes of occurrences of percussive timbres. The classification task discussed in this paper is successive to this detection and prior to the design of higher-level representations of rhythm. The integration of the results presented here in a complete system for automatically extracting rhythmic structures from audio is the object of a forthcoming paper.

For a given musical excerpt input, we intend to determine two time series of temporal indexes at which two different, perceptively important, rhythmic events take place; in our framework: the snare-like and bass drum-like sounds. The percussive sound detection scheme currently used applies correlation techniques with percussive sounds templates. These templates may be synthesized so that they would permit to deal with the greatest musical database possible; so they may not be too specific (e.g. an actual snare drum sound would be efficient for only a very few number of titles). Wanting to use generic sounds templates in our detection scheme, we must consider the fact that some artefacts are present in the detected occurrences. These artefacts are precisely occurrences of the second percussive

⊥ Fabien Gouyon was working at Sony CSL Paris when participating to the work described in this paper.

sound that is important in the perception of the rhythm (see Figure 1).

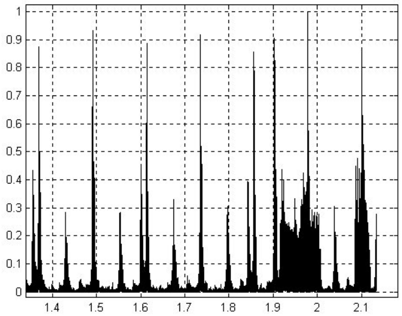

Figure 1: Correlation between a one-second excerpt and a template sound. Here, artefacts in the detection scheme correspond to the peaks with amplitude around .3.

Eventually, the classification task addressed here is very specific. Given any musical title from large popular music databases, and making the assumption that the stream of events to classify is made up of only two families of events, snare-like sounds and bass drum-like sounds – what yields our detection method – we address here the issue of their discrimination.

The purpose of this article is to introduce the choice of a precise physical parameter for this classification task.

At that point, let's stress the fact that the issue is not to design a universal timbre space over which one would project any drum sounds and would be able to identify them (as in e.g. [6]). We look for a sound feature that would permit to differentiate snares and bass drums in any musical excerpt, nonetheless, the difference between the automatic source identification topic and ours lies in the fact that the actual classes boundaries corresponding to classification features may differ from one given title to another. Indeed, physical attributes of snare and bass drum sounds differ greatly when considering large databases of titles. Besides, we believe that this is a good reason to investigate towards non-supervised classification methods (where it is assumed that no database of labelled data is available prior to the classification). We look for a parameter that can characterize differences between timbres that would be relative to musical excerpts rather than absolute.

The following part describes the multiple features examined that were computed over synthesized as well as real percussive sounds, introducing the sound segmentation scheme used. Examining two classification methods (Discriminant Factor Analysis and Agglomerative Clustering), we describe experimentations permitting to identify the best feature fitting our demands. We then propose conclusions and discussions.

2. EXTRACTION OF PERCUSSIVE SOUNDS FEATURES

In the search for a sound feature that would be relevant in respect to our classification purpose, we developed signal processing algorithms to extract several physical parameters. The target is to extract parameters from a relatively short signal, assumed to be percussive, but also mixed with important levels of noise (i.e. actual noise or any other instrument overlapping).

We first segment the sounds in several regions: an "attack" region (before the onset), a "decay" region (after the onset), and two "noise" regions at the extremities.

2.1. Segmentation

In order to avoid thresholds-based algorithms, whose goodness is too dependent on the level of noise, the segmentation scheme used here is based on the detection of temporal envelopes over absolute values of the signals (as in [11]).

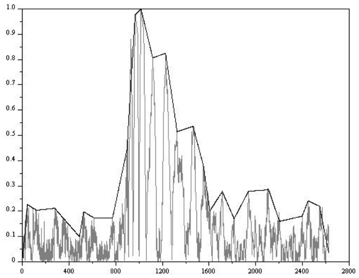

Figure 2: Rough envelope of an overall percussive sound extracted from a musical title.

Since the sounds we deal with may be mixed with noise, a very accurate envelope of the signal around the onset is hard to determine. However the goodness of the determination of the attack and decay times is directly linked to the accuracy of the envelope. We determine envelopes by finding the maximum of the waveform in windowed portions of the signal. Because the attack time is typically very short, and the sound is non harmonic, we cannot use the FFT to determine the size of the window (as done in e.g. [11]). Instead, we determine the window size by a local search algorithm which stops when it finds a window size which is stable (with respect to attack time estimation).

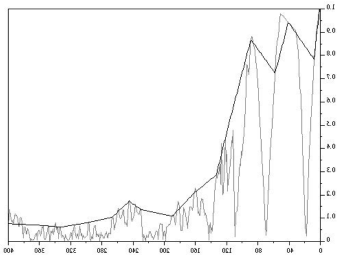

Figure 3: Absolute value of the signal (bass drum) before the onset and its envelope. Bad window size (too small), we enter the intraperiod of the sound.

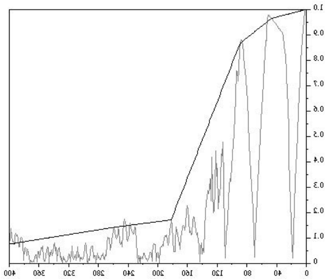

Figure 4: Absolute value of the signal (bass drum) before the onset and its envelope. Satisfying window size.

The attack and decay time are computed by a scheme focusing on the slopes of successive portions of the envelope (e.g. maximum-slope detection for the attack, relative flatness for the decay), achieved over the envelope in the regions respectively preceding and following the onset. With a sampling frequency equal to 11025 Hz, this method works with an uncertainty of approximately 30 samples, which corresponds to 3 ms.

2.2. Features

Labelling percussive sounds regions as attack and decay segments, we consider temporal descriptors such as attack and decay time; energy parameters, as well as frequencial descriptors computed over different regions. Since some sound regions can be relatively small (about 30 milliseconds), we also performed Prony modelling over attack and decay regions in order to account for a better frequencial precision. From this modelling, we keep the first 2 coefficients (damping factor and frequency) of the frequencial component with the highest magnitude, as well as features such as the number of sinusoids found in each regions. Eventually, zero-crossing rates have also been computed in the regions defined. The exhaustive list of the parameters is the following:

1. Attack time
2. Decay time
3. Focusing on the envelope of the attack region, the time difference between the index of maximum slope and the onset ; that gives an indication of the sharpness or the smoothness of the attack.
4. Number of sinusoids in the Prony modelling of the reversed attack region.
5. Number of sinusoids in the Prony modelling of the decay region.
6. Maximum magnitude component in the Prony modelling of the reversed attack region.
7. Maximum magnitude component in the Prony modelling of the decay region.
8. Exponential decay factor of the maximum magnitude component in the Prony modelling of the reversed attack region.
9. Exponential decay factor of the maximum magnitude component in the Prony modelling of the decay region.
10. Maximum magnitude component in the Fourier Transform of the attack region
11. Maximum magnitude component in the Fourier Transform of the decay region – below the *StrongestPartialFFT_Decay*
12. Maximum magnitude component in the Fourier Transform of the whole percussive sound
13. Local mean energy of the attack region
14. Local mean energy of the decay region
15. Local mean energy of the whole percussive sound
16. Proportion between local mean energy of the attack and the decay regions
17. Zero-Crossing Rate (ZCR) of the attack region – below the *ZCR_Attack*
18. ZCR of the decay region – below the *ZCR_Decay*
19. ZCR of the whole percussive sound

2.2.1. ZCR computation

It is defined as the number of time-domain zero-crossings within a defined region of signal, divided by the number of samples of that region.

The ZCR algorithm was implemented with a concern for the handling of two additive noises. The signal we are dealing with are very short (typically <100 ms), thus, a very low frequency note (w.r.t. the inverse of the duration of the signal), played by an overlapping instrument (e.g. a bass), acts as a disruptive element over the average level. The second type of noise we want to be able to deal with concerns the other instruments' high frequency components (again w.r.t. the inverse of the duration of the signal), which amplitudes can be considered inferior to the percussive sound's amplitude around the onset (e.g. voices, cymbals). These two characteristics of signals are considered as noise in the determination of the ZCR of the percussive sounds. Thus, in order to avoid artefacts, prior to the actual computation of the ZCR, the signal is transformed as followed :

1. A DC offset is computed and subtracted.
2. The signal is passed through a 30 dB noise gate.

The ZCR is then computed, focusing on the changes of sign of the signal, using a sample-by-sample sequential algorithm:

```
n = 0;
temoin = 1;
for i=2:N
          if (sign(x(i))==sign(x(i-temoin))) | (sign(x(i))==0),
    temoin = temoin+1;
  else n = n+1;
    temoin = 1;
  end;
end;  ZCR=n/N ;
```

3. CLASSIFICATION OF PERCUSSIVE SOUNDS

Within the framework of the classification of percussive sounds into a snare-like class and bass drum-like one, we present here the experimentations achieved over the set of features previously described.

To achieve the classification, we first identify the dimensions of percussive sounds that seem the most relevant to our problem. We then validate the choice of one of these dimensions on larger data sets.

3.1. Identification of a relevant dimension

To test the relevance for classification tasks of several features of percussive sounds, we consider the framework of supervised analysis (i.e. considering that a database of labelled data is available for a pre-processing phase). As a starting point, we describe percussive sounds in a large and redundant representation space (as in e.g. [8]), consisting of the 19 parameters described above.

Amongst the extracted features of all the sounds, we look for the most relevant for our discrimination task. This pre-processing is done by applying a Discriminant Factor Analysis that uses the Fisher criterion (see e.g. [13]).

3.1.1. Discriminant Factor Analysis

The scheme consists in determining the axis, over which projection of the data is achieved, permitting to best separate the classes. In the Fisher criterion's framework, an axis, labelled u, permits a good projection if the distances between the averages of the classes are important, and if the variances within each class are small.

The following is defined:
B: the *interclass* dispersion matrix
S: the *intraclass* dispersion matrix

Fisher criterion: $J(u)=\frac{u'Bu}{u'Su}$

One can also define
T: the covariance matrix

and the criterion $I(u)=\frac{u'Bu}{u'Tu}$

and verify that $u'Tu=u'Bu+u'Su$

Determining $J(u)$ for each axis u, the more this criterion is important, the more the associated axis is discriminant.

3.1.2. Pre-processing database

We consider a database consisting of samples taken from the Korg 05RW's General MIDI drum kit. These sounds are classified into two categories by hand: bass drum sounds (15 sounds) and snare sounds (6 sounds).

This analysis indicates that:

- there are two dimensions which, taken alone, allows to differentiate the sounds: *ZCR_Decay* (zero-crossing rate computed over the decay region) and the *StrongestPartialFFT_Decay*. These two parameters are approximately as discriminant.
- *ZCR_Decay* is 1.3 times more discriminant than *ZCR_Attack*.

We now investigate the goodness of the reduction of the 19-dimensional space onto a single dimension.

3.2. Validation

We apply an Agglomerative Clustering method (i.e. non-supervised analysis) over a set of sounds projected in the representation space yielded by the pre-processing phase, to check the relevance of the parameter used for measuring distances between sounds.

3.2.1. Agglomerative Clustering

Computing the values of the chosen parameter for a set of sounds, we actually project these sounds over a vector space (in this particular case: 1-dimensional). These measures, representative of each sound, allow to compute distances – relative to this dimension – between sounds.

Given n sounds, initially, n groups of single elements are constructed. Regarding the distances between groups, the two closest ones are agglomerated in a single one. This scheme is iterated (n – 1) times, so that the clustering method eventually yields only two clusters.

Given a measure over a dimension, determining the distances between singletons is achieved easily, the problem resides in determining a distance between a singleton and a group, or between two groups. Solutions to this problem differ whether one considers solely the relative distances between elements, or the absolute projection values of each element over the vector space (in some problems, these values are not directly accessed). In the latter case, one generally computes centres of gravity of groups and uses it as a distance measure between groups. In the former case, the literature is wide, some solutions are the average distance (average distance between each couple of elements), the minimal radius (smallest distance between all the couples of elements of both groups), or the maximal radius.

Let $G_1 = \{I_{1,1},...,I_{1,n}\}$ and $G_2 = \{I_{2,1},...,I_{2,m}\}$ be two groups of sounds, the following proximity measures between groups are defined:

- Average distance

$$\text{Prox}_{\,Average}(G_1,G_2) = \frac{1}{n*m}\sum_{i=1}^{n}\sum_{j=1}^{m}\text{Prox}(I_{1,i},I_{2,j})$$

- Minimal radius

$$\text{Prox}_{R\min}(G_1,G_2) = \text{Min}\{\text{Prox}(I_{1,i},I_{2,j}), with\, i \in [1,n] and\ j \in [1,m]\}$$

- Maximal radius

$$\text{Prox}_{R\max}(G_1,G_2) = \text{Max}\{\text{Prox}(I_{1,i},I_{2,j}), with\, i \in [1,n] and\ j \in [1,m]\}$$

Where $\text{Prox}(I_1,I_2)$ is the proximity measure between instrument sounds.

Other possibilities are thinkable, for instance, one can use the distances to the square.

The maximal radius and average distances have been implemented.

Different types of sounds have been used for the validation tests.

3.2.2. *Monophonic/clean sounds*

The first data set used for validation is made up of thirty-six bass and snare drums sounds taken from other Korg 05RW drum kits − Jazz Kit, Brush Kit, Dance Kit and Power Kit. It is important to notice that these sounds are monophonic and very clean. We feed these sounds as input to the agglomerative cluster analysis, and compute distances between sounds according to *ZCR_Decay* and the *StrongestPartialFFT_Decay* dimension. In both cases the clustering in two classes yields a snare drum class and a bass drum class with 94.5 % accuracy: *ZCR_Decay* and *StrongestPartialFFT_Decay* are as efficient.

3.2.3. *Polyphonic/noisy sounds*

Things change when dealing with real sounds. We used data sets made up of real sounds taken from excerpts of popular music titles. To build a data set of “real sounds”, we use a percussive sound detection scheme currently developed in our project and briefly described in the introduction.

In real music, percussive sounds are not as “pure” as in synthesizers sound banks: occurrences of bass drum or snares are often mixed with the rest of the music (voices, parts of the electric bass, and more generally “noise”). Our rationale here is not to attempt to separate the “pure” percussive sound from the rest of the music, but rather to try to classify the short music segment as a whole.

For each twenty seconds excerpt (20 excerpts from a popular music database have been used), the number of percussive sounds as given by our detection scheme varies from 19 to 63.

Using the *StrongestPartialFFT_Decay* dimension, the clustering of these sounds in two classes yields a snare drum-like class and a bass drum-like class, with an accuracy varying from to 78% to 89% depending on the actual music excerpt. The results are approximately the same for the *ZCR_Attack* dimension.

However, using the *ZCR_Decay* dimension, the clustering of these sounds yields an accuracy varying from 87.5 to 96%, which is better.

4. CONCLUSION

In classification tasks, it is generally understood that issues one is likely to address are the following: the type of features to use, the actual classification method to use and, in the case of supervised analysis, the size of the data set for the pre-processing phase. This paper doesn’t claim to present a review of instrument classification techniques (as done in [5]). The precise classification task we address here is different from the identification task the reader may be used too (training of the system with a large data set of labelled data, and then actual classification). As the physical attributes of snare and bass drum sounds differ greatly when considering large databases of titles, and so does the type of noise surrounding, the classification scheme used should permit the classes boundaries to differ from one given title to another. This justifies the use of a non-supervised classification technique: Agglomerative Clustering. However, to compute distances, this technique must be fed an input parameter: the dimension over which sounds are projected. In order to get clues regarding relevant dimensions, we believe that defining a large number of percussive sounds’ features, and achieving a Discriminant Factor Analysis over a small set of sounds is justified.

As introduced in the first paragraph, the issue addressed here is not the design of a universal timbre space over which one would achieve instrument identifications. This research topic assumes that monophonic and clean sounds are provided. In the framework of the content-based description of audio achieved directly from commercial CD samples, this implies that efficient source separation techniques are applied as a preliminary step. Instead of restricting the applicability of classification algorithms to such a preliminary stage, we believe that we must devote our efforts to an approach of *signal understanding without separation* (see [10]). Our way to approach this theme is to deal directly with real polyphonic signals and to set simpler, but more specific objectives as the one addressed in this paper's abstract. We believe that the extraction of time indexes of percussive timbres occurrences in audio signals is a first step to be taken towards the automatic extraction of rhythmic structures, and that it can be seen as a symbolic intermediate between very low level representation of the information (the signal itself) and higher level of abstraction of the information (the rhythm structure).

The classification of percussive sounds holds a central position in the process of extracting these time indexes, and we believe that the *ZCR_Decay* dimension (as introduced before, the ZCR algorithm was designed with a concern for the handling of additive noises) is the feature that best fits our specific demands. It can be used over large data sets to achieve satisfying discrimination between two important classes of percussive sounds. We observe that, in the case of real sounds, such a simple parameter is more appropriate than more complicated and computationally time-consuming ones. Instead of using only *ZCR_Decay*, we could project sounds over a higher dimensional space, adding a second dimension being the zero-crossing rate computed over the attack regions. We could also use the criterion $I(u)$ and look for the best linear mixture of parameters weighted by their scores in the Discriminant Factor Analysis. However, the gain in accuracy probably would not balance the extra computational cost.

5. REFERENCES

[1] Brown, J.C.: "Determination of the meter of musical scores by autocorrelation", J. Acoust. Soc. Am. 94, 1993

[2] Cemgil A., Desain P., Kappen B.: “Rhythm quantization for transcription” Computer Music Journal, vol. 24 nº2, 2000

[3] Gabrielsson A.: "Similarity ratings and dimension analyses of auditory rhythm patterns", Parts I & II, Scandanavian Journal of Psychology 14, 1973

[4] Herrera P., Serra X., Peeters G.: "Audio Descriptors and Descriptors Schemes in the Context of MPEG-7", Proc. ICMC 1999

[5] Herrera P., Amatriain X., Batlle E., Serra X.: “Towards instrument segmentation for music content description: a critical review of instrument classification techniques”, ISMIR 2000

[6] Kaminskyj I. and Materka A.: “Automatic source identification of monophonic musical instrument sounds”, Proc. of the IEEE International Conference On Neural Networks. 1, 1995

[7] Rossignol S. & al: "Features extraction and temporal segmentation of acoustic signals", Proc. ICMC 1998

[8] Scheirer E., Slaney M.: "Construction and evaluation of a robust multifeature speech/music discriminator", Proc. IEEE ICASSP 1997

[9] Scheirer E.: "Tempo and beat analysis of acoustic signals", JASA, 103(1) 1998

[10] Scheirer E., "Music-Listening Systems." Ph.D. thesis. MIT. Cambridge, MA. 2000.

[11] Schloss A.: "On the automatic transcription of percussive music – From acoustic signals to high-level analysis", CCRMA internal report, Stanford Univesity 1985

[12] Serra X., Bonada J.: "Sounds transformations based on the SMS high level attributes", Proceedings of the DAFX98, Barcelona 1998

[13] Tourneret J.Y.: “Classification et Reconnaissance des formes” ed. ENSEEIHT (Ecole Nationale Supérieure d’Electrotechnique, d´Electronique, d’Informatique et d’Hydraulique de Toulouse), 1997