Locating Segments with Drums in Music Signals

Toni Heittola
Tampere University of Technology
P.O.Box 553
FIN-33101 Tampere, Finland
+358 3 365 4796
toni.heittola@tut.fi

Anssi Klapuri
Tampere University of Technology
P.O.Box 553
FIN-33101 Tampere, Finland
+358 3 365 2124
klap@cs.tut.fi

ABSTRACT

A system is described which segments musical signals according to the presence or absence of drum instruments. Two different yet approximately equally accurate approaches were taken to solve the problem. The first is based on periodicity detection in the amplitude envelopes of the signal at subbands. The band-wise periodicity estimates are aggregated into a summary autocorrelation function, the characteristics of which reveal the drums. The other mechanism applies straightforward acoustic pattern recognition approach with mel-frequency cepstrum coefficients as features and a Gaussian mixture model classifier. The integrated system achieves 88 % correct segmentation over a database of 28 hours of music from different musical genres. For the both methods, errors occur for borderline cases with soft percussive-like drum accompaniment, or transient-like instrumentation without drums.

1. INTRODUCTION

Segmentation and analysis of musical signals has gained increasing amounts of research interest in recent years [1,2,11,12]. The presence/absence of drum instruments is an important high-level descriptor for music classification and retrieval. In many cases, exactly expressible descriptors are more efficient for information retrieval than more ambiguous concepts such as musical genre. For example, someone might search for classical music by requesting a piece with string instruments and without drums. Information about the drums can also be used in audio editing, or in further analysis, e.g. in music transcription, metrical analysis, or rhythm recognition.

The aim of this paper is to present a drum detection system, which would be as generic as possible. The problem of drum detection in music is more difficult than what it seems at a first glance. For a major part of techno or rock/pop music, for example, detection is more or less trivial. However, a detection systems designed for these musical genres does not generalized to the others. Music contains a lot of cases that are much more ambiguous. Drums go easily undetected in jazz/big band music, where only hihat or cymbals are softly played at the background. On the other hand, erroneous detections may pop up for pieces with acoustic steel-stringed guitar, pizzicato strings, cembalo, or staccato piano accompaniment, to mention some examples.

Earlier work in the area of the automatic analysis of musical rhythms has mostly concentrated on metrical analysis [14], and in many cases for MIDI data. There are a few exceptions, however. Alghoniemy et al. used a narrowband filter at low frequencies to detect macro and micro scale periodicity [3]. Tzanetakis et al. have used the Discrete Wavelet Transform to decompose the signal into a number of bands and autocorrelation function to detect the various periodicities of the signal's envelope [1]. This structure was used to extract features for musical genre classification. Soltau et al. have used HMMs with Neural Networks to represent temporal structures and variations in musical signals [2].

2. METHODS

2.1 Preprocessing with Sinusoidal Modeling

Drum instruments in Western music typically have a clear stochastic noise component [4]. The spectral energy distribution of the noise component varies, being wide for the snare drum, and concentrated to high frequencies for cymbal sounds, for example. In addition to the stochastic component, some drums have strong harmonic vibration modes, and they have to be tuned. In the case of tom toms, for example, approximately half of the spectral energy is harmonic. Nevertheless, these sounds are still recognizable based on the stochastic component only. While most other musical instruments produce chiefly harmonic energy and we are interested in the drums, an attempt was made to separate the stochastic and harmonic signal components from each other.

A sinusoids plus noise spectrum model was used to extract the stochastic parts of acoustic musical signals. The model, described in [5,6], estimates the harmonic parts of the signal and subtracts them in time domain to obtain a noise residual. Although some harmonic components are not detected and beginning transients of other instruments leak through, the residual signal in general has significantly better "drums-vs-other" ratio than the input signal.

2.2 Periodicity Detection Approach

Periodicity is characteristic for musical rhythms. Drum events typically form a pattern which is repeated and varied over time. As a consequence, the time-varying power spectrum of the signal shows clear correlation with a time shift equal to the pattern length in the drum track. We propose that the presence of drums can be detected by measuring this correlation in musical signals. This evaluates a backgrounding hypothesis that periodicity of stochastic signal components is a universally characteristic of musical signals with drums. In order to alleviate the interference of other musical instruments, periodicity measurement is performed in the residual signal after preprocessing with a sinusoidal model.

2.2.1 Feature Stream

A signal model was employed which discards the fine structure of signals, but preserves their rough spectral energy distribution. Band energy ratio (BER) is defined as the ratio of the energy at a certain frequency band to the total energy. Thus the BER for the i^{th} subband in time frame k is:

$$C(i,k) = \frac{\sum_{n \in S_i} \left|X_k(n)\right|^2}{\sum_{n=0}^{M} \left|X_k(n)\right|^2} \qquad (1)$$

where S_i is the set of Fourier transform coefficients belonging to the i^{th} subband [7]. Feature vectors are extracted from the preprocessed signal.

Human auditory perception does not operate on a linear frequency scale. Therefore we apply a filter bank consisting of triangular filters spaced uniformly on the mel-scale. An approximation between a frequency value in Hertz and in mel is given as: [8]

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \qquad (2)$$

Features were extracted in 10 ms analysis windows (Hanning windowing) and with 50% overlap. Short window length was preferred to achieve a better time resolution in the autocorrelation calculations later on. The amount of 16 frequency bands was found to give sufficient resolution in frequency domain. Obtained feature vectors form a feature stream $C(i,k)$, which is subject for autocorrelation function calculations.

2.2.2 Summary Autocorrelation Function

At each frequency band, an autocorrelation function (ACF) is calculated over the BER values within a sliding analysis window. Analysis window length of three seconds was chosen to capture a few patterns of even the slowest rhythms. Autocorrelation function of a K-length excerpt of $C(i,k)$ at band i is given by:

$$r_i(\tau) = \frac{1}{K} \sum_{k=0}^{K-\tau-1} C(i,k) \cdot C(i, k + |\tau|) \qquad (3)$$

where τ is the lag. Peaks in the autocorrelation function correspond to the lags where the time-domain signal has stronger periodicity.

Despite the preprocessing, also other instruments cause peaks to the bandwise autocorrelation functions. Fortunately, however, the spectrum of the other instruments tends to concentrate to the mid-bands, whereas drums are more prominent at the low or high bands (there are exceptions from this rule, e.g. the violin or the snare drum). On the basis of this observation we will weight bands differently before forming the summary autocorrelation function (SACF). Lower and higher bands are assigned equal weights and mid-bands have are steeply attenuated. Autocorrelation functions are weighted and then summed up in order to form SACF

$$s(\tau) = \sum_{i=1}^{Bands} W_i \cdot r_i(\tau) \qquad (4)$$

This overall structure bears a close resemblance to the mechanisms of human pitch perception, as modeled in [15]. A major difference here is that processing is done for subband amplitude envelopes instead of the signal fine structure. The SACF was then mean-normalized to get real peaks step out better from the SACF. Mean normalization was done with the following equation [9]:

$$\begin{cases} \hat{s}(0) = 1 \\ \hat{s}(\tau) = \dfrac{s(\tau)}{\dfrac{1}{\tau} \sum_{j=1}^{\tau} s(j)} \end{cases} \qquad (5)$$

Overview of whole system is shown in Figure 1.

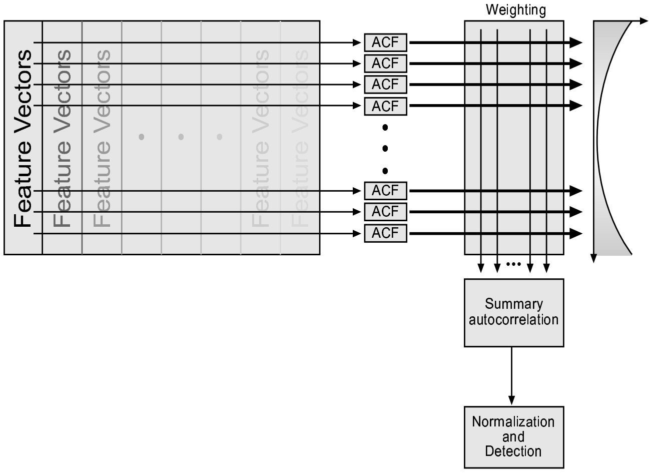

Figure 1. System overview.

2.2.3 Detection

Since a quite short analysis frame (10ms) was used in extracting the feature stream, the lowest frequency components cause slight framing artefacts. These appear as a low-amplitude and high-frequency ripple in the SACF, which is easily removed using moving averaging. Also, a long-term trend caused by differences in signal levels within the ACF analysis window will be detrended from SACF using high pass filtering. Thus obtained SACFs for different type of music are shown in Figure 2.

As one can see from Figure 2, periodic drum events produce also a periodic SACF. In order to robustly detect this, SACF has to be enhanced in a manner illustrated in Figure 3. The original SACF curve is time-scaled by a factor of two and three and these two stretched curves are added to the original, resulting in the enhanced summary autocorrelation function (ESACF). Thus peaks at integer multiples of a fundamental tempo are used to enhance the peaks of a slower tempo. If the original SACF is periodic in nature, this technique produces clearer peaks. This technique has been earlier applied in [10].

The region of interest in the ESACF is determined by reasonable tempo limits. Lower limit was fixed to 35 beats per minute, and higher to 120 beats per minute. Whereas the upper limit may seem too tight, it should be noted that due to the above describe enhancement procedure, these limits actually corresponds to 35 and 360 in SACF. This wide tempo range is essential because the rate of playing certain drum instruments (e.g. the hihat) is typically an integer multiple of tempo, and causes a clear peak in the SACF.

Final detection is carried out by measuring the absolute maximum value within the given tempo limits. Maximum value distributions for segments with drums and without are presented in Figure 4. Distributions overlap to some extent, but nevertheless enable robust classification.

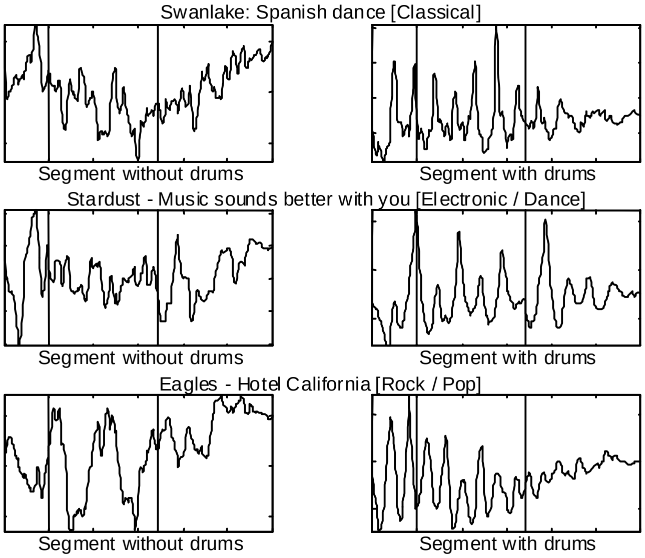

Figure 2. Representative summary autocorrelations from different type of music (Tempo limits marked in the plots).

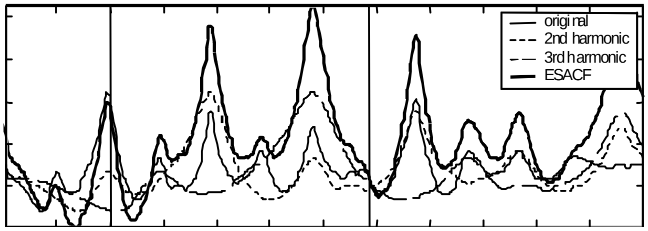

Figure 3. Enhancing the summary autocorrelation function.

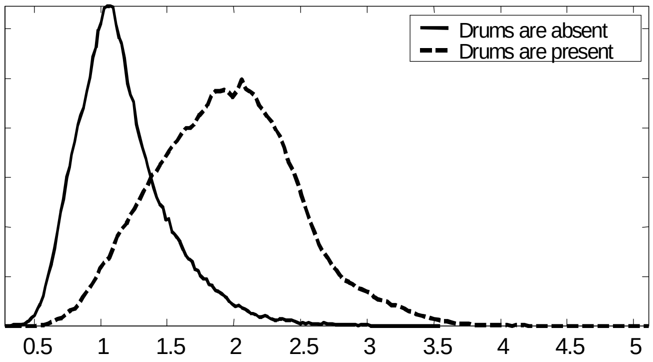

Figure 4. Unit area normalized feature value distributions for both classes.

2.3 Acoustic Pattern Recognition Approach

As discussed above, drums have characteristic spectral energy distributions. The spectral energy of a bass drum is concentrated to lower frequencies. Cymbals and hihats occupy a wide frequency band, mainly concentrated to the treble end. The highest frequencies of the cymbals and hihats are so high that there are only a few other instruments, which have prominent frequency components in the same range (e.g. strings). Therefore drums make a significant contribution to the overall tone colour of musical signals. Based on this, we studied the ability of certain acoustic features to indicate the presence of drums in musical signals.

2.3.1 Mel-frequency cepstral coefficients (MFCC)

Mel-frequency cepstral coefficients have been widely used to model speech and music signals. Foote has used cepstral representation in his content-based retrieval system [11]. Also Li et al. found MFCC to be generally the best way to model audio signals [12]. We used 16 MFCC coefficients, calculated in 20ms frames with ¼ overlap, as features for a classifier.

MFCC is a short-term spectral feature and is able to represent the rough shape of the magnitude spectrum in a compact way [13]. First step of a MFCC feature extractor is preprocessing, which consists of pre-emphasizing, frame blocking and time domain windowing. After this, a discrete Fourier transform is calculated and the power spectrum is transformed to a mel-frequency scale. This is done by using a filter bank consisting of triangular filters, spaced uniformly on the mel-scale. An approximation between frequency in hertz and in mel was given in Equation 2.

Both static (MFCC) and delta coefficients (ΔMFCC) were used. Applying sinusoidal modeling as a preprocessing step before feature extraction was tried out, but it did not affect the overall performance as shown in Section 3.3.

2.3.2 Gaussian Mixture Models (GMM)

A Gaussian mixture density is able to approximate an arbitrary probability distribution function (pdf) with a weighted sum of *M* multivariate Gaussian pdf's [14,13]. The Gaussian mixture density with a model order M is given by

$$p(x|\lambda)=\sum_{i=1}^{M} w_i p_i(x) \qquad (6)$$

where x is a *d*-dimensional random vector, $p_i(x)$ are the *M* Gaussian pdf's and w_i are the *M* mixture weights. The sum of the mixture weights is one and the pdf of the i^{th} *d*-variate normal distribution is given with $p_i(x)$. A GMM is completely represented with three parameters: the mean vectors, the covariance matrices, and the mixture weights. These three parameters are collectively represented with λ. The parameters are estimated using the Expectation Maximization (EM) algorithm so that the likelihood of the data is maximized.

The algorithm guarantees a monotonically non-decreasing likelihood and it converges at least to a local maximum of the underlying likelihood function. For a sequence of *T* data vectors, $X=(x_1,...,x_T)$, the GMM likelihood is given as follows

$$p(X|\lambda)=\prod_{t=1}^{T} p(x_t|\lambda) \qquad (7)$$

In order to use GMM as a classifier, GMM parameters for each class must first be estimated from a training data set. In a classification phase, the probability of each class for a given observation is evaluated and the class that gives the highest probability is chosen as the classification result.

2.3.3 k-Nearest Neighbour Classifier (k-NN)

The *k-NN* classifier places the feature vectors of the training set in a feature space, and makes the classification decision by "voting" among the nearest *k* neighbors of the data vector to be classified. The voting is done by picking the *k* points nearest to the current test point, and the chosen class is the class that is most often picked.

In this paper, Mahalanobis distance was used in determining the nearest neighbours. Also, the extracted features were processed before using them with this classifier. The mean and standard deviation of each feature are calculated within frames of 0.5

seconds, and the mean and standard deviation are used in place of the original features. This doubles the amount of features, but significantly reduces the amount of feature vectors over time.

3. Simulations

3.1 Database

A database of 397 entire musical pieces from different genres was used to evaluate the two drum detection schemes presented in this article. For each piece in the database, time segments with and without drums were manually annotated. Annotation was done with a precision of one second, and only stable segments of more than five seconds were used in simulations.

“Presence of drums” was defined to include the requirement that the drum is played in a rhythmic role. Special care had to be taken with classical music. Kettledrum is used in many classical pieces, but not always in a rhythmic role. Kettledrum has to play a clear repeating pattern, not just to be used to emphasize a certain part of the piece, in order to accepted as a “drum” instrument. With breaks in modern electronic dance music, where drum track’s amplitude increases gradually, the boundary was chosen based on when a human listener perceived the presence of the drums. Detailed statistics of the database are shown in Table 1.

Table 1. Statistics of the evaluation database.

Genre	%	# of songs	Drums absent	Drums present
Classical	27%	107	89%	11%
Electronic / Dance	7%	27	18%	82%
Hip Hop / Rap	3%	12	5%	95%
Jazz / Blues	16%	64	10%	90%
Rock / Pop	29%	115	11%	89%
Soul / RnB / Funk	11%	45	8%	92%
World / Folk	7%	27	56%	44%
Total (over 28h)		**397**	**32%**	**68%**

3.2 Test Setup

As can be seen in Table 1, the evaluation database is not nicely balanced from the point of view of the amount of material with and without drums in each individual genre. Since drums are a basic element in many Western genres, this was expected. In order to assure that we have as balanced as possible train and test sets, following scheme was used:

1. Pieces were divided into the seven main genres shown in Table 1.
2. The seven main genres were further divided into three sub-categories: pieces containing only segments where drums are present, pieces containing only segments where drums are absent, and pieces containing both segments
3. Fifty percent of pieces in each sub-category were randomly selected to the training set, and the rest to the test set.
4. An individual piece may appear only in the test or in training set, but not in both.

3.3 Results

3.3.1 Periodicity Detection Approach

First an optimal weight vector to be used in the SACF formulation was determined (see Eq. 4). For this sake, a smaller test carried out. Test set was formed using scheme described in Section 3.2 but only 30% of pieces were chosen. Results are presented in Figure 5. Performance difference between the flat line (78.7%) and steep parabola (80.6 %) was quite small. However, the best performance is reached with equally weighted lower and higher band and attenuation for center bands. So we fixed unit weight for both the highest and the lowest band, and 1/100 weight for center band to be used in final simulations.

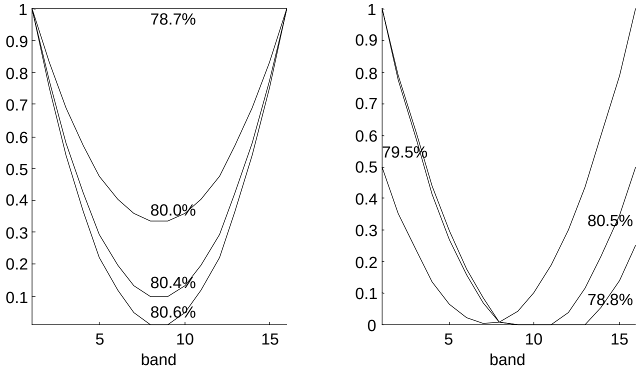

Figure 5. Effect of weighting before SACF.

Fifty percent of the pieces were used to estimate feature value distributions for intervals with drums and without. Division between this distribution estimation set and final test set was done as described in Section 3.3. Obtained feature value distributions were presented earlier in Figure 4. Based on these distributions a threshold value for maximum value within periodicity limits was defined. Detection results obtained with this threshold value are shown in Table 2.

Table 2. Results using periodicity detection.

Genre	Performance	Drums absent	Drums present
Classical	83.2%	83.9%	78.1%
Electronic / Dance	91.0%	61.4%	95.6%
Hip Hop / Rap	87.3%	69.5%	88.0%
Jazz / Blues	75.2%	38.2%	79.2%
Rock / Pop	83.0%	81.6%	83.2%
Soul / RnB / Funk	78.2%	79.5%	78.1%
World / Folk	69.2%	51.9%	92.3%
Total	**81.3%**	**76.5%**	**83.4%**

Overall performance was 81.3%. The reason why the distrubutions of the two classes overlap rather much is that the stochastic residual contains harmonic components and beginning transients from other instruments, too, and in some cases these show very much drum-like periodicity. Thus the starting hypothesis that periodic stochastic components reveal drum events was still mainly right. More attention should be paid for the preprocessing system in order to make concluding remarks.

3.3.2 Acoustic Pattern Recognition Approach

In order to perform classification with Gaussian Mixture Models, training set feature vectors were used to estimate model parameters for the two classes, one model for music with drums and another for music without drums.

We tested MFCCs alone as well MFCCs catenated with ΔMFCC as a feature vectors. In order to avoid numerical problems, the features were normalized to have zero mean and unity variance, given by:

$$\hat{x}_{ik} = \frac{x_{ik} - m_k}{\sigma_k} \qquad (7)$$

The results obtained with GMM-classifier are shown in Table 3. As one can see, the overall performance was slightly better than with the system periodicity detection approach. The performance difference between preprocessed signals and original signals was marginal. However, if we take a closer look to the results in Table 4, we will see that performance is not evenly distributed within different musical styles. Although a high performance is obtained for one class (e.g. drums present), the other fails within the individual musical style. In other words, the system starts to recognize the musical style rather than the drums. This is clearly seen for classical music, for example. Due to the small amount of training material for e.g. classical music with drums, GMM was unable to model it effectively with one generic model for all genres with drums present.

In order to prevent the above described problem, the number of GMM models was increased. For each musical style, two models were estimated: for intervals with drums and without. Since we are not interested in the musical style, genre was ignored in classification stage. So we were only interested in the set of models (drums are present or absent) from which we got the highest likelihood. The results are much better balanced than those obtained with just two models, as shown in Table 5.

Table 3. Classification results with GMM.

GMM model order	MFCC with preprocessing	MFCC+ΔMFCC with preprocessing	MFCC+ΔMFCC without preprocessing
4	81.9%	86.4%	86.0%
8	82.9%	86.4%	86.9%
12	83.5%	86.4%	86.1%
16	83.4%	86.4%	86.1%
24	83.7%	86.7%	86.9%

Table 4. GMM results with two models. (MFCC + ΔMFCC, model order 24, 3-second test excerpts.)

Genre	Performance	Drums absent	Drums present
Classical	89.9%	97.4%	38.6%
Electronic / Dance	88.5%	48.9%	96.0%
Hip Hop / Rap	93.8%	25.7%	98.3%
Jazz / Blues	73.9%	58.0%	75.7%
Rock / Pop	92.1%	67.7%	94.8%
Soul / RnB / Funk	90.9%	76.9%	92.5%
World / Folk	68.3%	47.9%	95.0%
Total	**86.7%**	**83.5%**	**88.2%**

Table 5. GMM results with two models for each genre. (MFCC + ΔMFCC, model order 24, 3-second test excerpts.)

Genre	Performance	Drums absent	Drums present
Classical	85.6%	89.2%	61.0%
Electronic / Dance	89.28	63.4%	94.7%
Hip Hop / Rap	89.9%	25.7%	94.2%
Jazz / Blues	70.9%	67.4%	71.3%
Rock / Pop	89.0%	76.5%	90.4%
Soul / RnB / Funk	91.8%	85.4%	92.5%
World / Folk	66.0%	45.7%	92.7%
Total	**84.2%**	**80.2%**	**86.1%**

Figure 5 shows the overall performance of GMM as a function of the length of the signal excerpt used for classification. A reasonable performance (80 %) was achieved already with a 100 ms test excerpt.

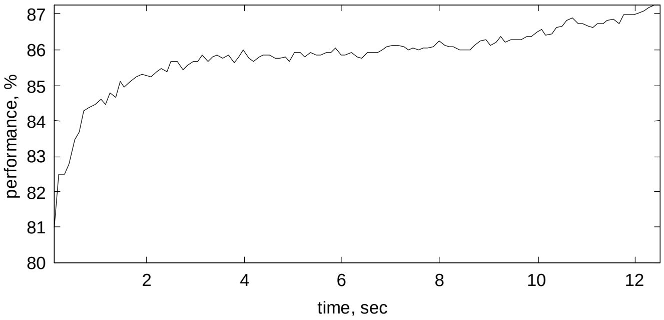

Figure 5. Effects of test sequence length with GMM-classification.

In addition to GMM-classifier, a k-NN classifier was also used to evaluate differences between classifiers. Train data and test was processed as described in Section 2.3.3. The results are presented in Table 5. Performance k-NN is between GMM and periodicity system. Performance is unbalanced like it was with GMM. The performance is slightly improved by increasing the number of “voting” points, *k*.

Table 5. Classification results for k-NN classifier.

k	Overall performance	Drums absent	Drums present
1	80.0%	69.5%	84.8%
5	83.4%	70.9%	89.2%

3.3.3 Combination of the Two Approaches

The two drum detection systems are based on different information, one on periodicity and the other on spectral features. One would thus guess that the combination of the two systems would perform more reliably than either of them alone. Fusion of the two systems was realized by combining their output likelihoods. For periodicity detection, the likelihood is obtained from the feature value distributions presented in Figure 3. For GMM, the likelihoods are obtained as described in Section 2.3.2.. The results are presented in Table 6. Only a minor improvement (1-2 %) was achieved, as can be seen. This is due to the fact that both of the systems typically misclassify within the same intervals. For example, jazz pieces where drums are played quite softly with brush, or ride cymbal is continually tapped are likely to be misclassified with both systems. In some cases, the misclassification might be acceptable, since the drums are difficult to detect even for a human listener.

Table 6. Comparison of results obtained earlier and by combining GMM (MFCC + ΔMFCC with GMM model order 24) and periodicity detection.

Detection system	Overall performance	Drums absent	Drums present
Periodicity detection	81.3%	76.5%	83.4%
GMM	86.7%	83.5%	88.2%
Combined detection	**88.0%**	**83.9%**	**90.1%**

4. SUMMARY AND CONCLUSIONS

Two different drum detection schemes were described and evaluated. The obtained results are rather close to each other and, somewhat surprisingly, the combination performs only slightly better. This highlights a fact which was also validated by listening: both system fail in borderline cases that are difficult, not just due to algorithmic artefacts. Achieved segmentation accuracy of the integrated system was 88 % over a database of varying musical genres. The misclassified intervals are more or less ambiguous by nature and in many cases might be tolerated by a user. In order to construct a substantially more accurate system, it seems that more complicated sound separation and recognition mechanism would be required. In non-causal applications, longer analysis excerpts and the global context can be used to improve the performance.

5. ACKNOWLEDGMENTS

The feature extractors and classifiers were developed with Antti Eronen. Our MFCC analysis was based on Slaney’s implementation (http://rvl4.ecn.purdue.edu/~malcolm/interval/1998-010/). Sinusoidals plus noise spectrum modeling tools were kindly provided by Tuomas Virtanen.

6. REFERENCES

[1] Tzanetakis,G., Essl, G. and Cook, P. Automatic musical genre classification of audio signals, In International Symposium on Musical Information Retrieval, 2001

[2] Soltau,H., Schultz,T., Westphal, M. and Waibel, A. Recognition of music types, In Proc. International Conference on Acoustic, Speech, and Signal Processing, Lansdowne, Virginia, February 1998

[3] Alghoniemy, M., Tewfik, A.H., Rhythm and periodicity detection in polyphonic music, In Proc. IEEE Third Workshop on Multimedia Signal Processing, Denmak, September 1999, 185-190.

[4] Fletcher, N. H. and Rossing, T. D., The Physics of Musical Instruments. Springer-Verlag, New York, 1991.

[5] Virtanen, T., Audio signal modeling with sinusoids plus noise. Master’s thesis, Department of Information Technology, Tampere University Of Technology, 2000

[6] Serra and Xavier, Musical Sound Modeling with Sinusoids plus Noise. Roads C. & Pope S. & Picialli G. & De Poli G. (eds). Musical Signal Processing. Swets & Zeitlinger Publishers.

[7] Peltonen,V., Computational Auditory Scene Recognition. In Proc. International Conference on Acoustic, Speech, and Signal Processing, Orlando, Florida, May 2002.

[8] Moore, B.C.J. (Ed.), Hearing – Handbook of Perception and Cognition, New York: Academic Press, 1995.

[9] Cheveigne and Kawahara. YIN a fundamental estimator for speech and music. JASA.

[10] Tolonen, T. and Karjalainen, M., A computationally efficient multipitch analysis model, IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 6, Nov. 2000.

[11] Foote, J. , Content-based retrieval of music and audio, Proceedings of SPIE, 138-147, 1997

[12] Li,D. , Sethi,I.K. , Dimitrova,N., and McGee,T. Classification of general audio data for content-based retrieval. Pattern Recognition Letters, 25(5):533-544, April 2001.

[13] Rabiner, L. and Juang B-H. , Fundamentals of Speech Recognition, Englewood Cliffs/NJ, Prentice-hall, 1993.

[14] Reynolds,D. And Rose,R. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Transactions on Speech and Audio Processing, 3(1):72-83, 1995.

[15] Scheirer, E.D., Tempo and beat analysis of acoustic musical signals, J. Acoust. Soc. Am. 103 (1), 1998, 558-601.

[16] Meddis, R., Hewitt M.J., Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification, J. Acoust. Soc. Am. 89 (6), June 1991.