

# IMPROVING RHYTHMIC SIMILARITY COMPUTATION BY BEAT HISTOGRAM TRANSFORMATIONS

**Matthias Gruhne**  
Bach Technology AS  
ghe@bachtechnology.com

**Christian Dittmar**  
Fraunhofer IDMT  
dmr@idmt.fhg.de

**Daniel Gaertner**  
Fraunhofer IDMT  
gtr@idmt.fhg.de

## ABSTRACT

Rhythmic descriptors are often utilized for semantic music classification, such as genre recognition or tempo detection. Several algorithms dealing with the extraction of rhythmic information from music signals were proposed in literature. Most of them derive a so-called beat histogram by auto-correlating a representation of the temporal envelope of the music signal. To circumvent the problem of tempo dependency, post-processing via higher-order statistics has been reported. Tests concluded, that these statistics are still tempo dependent to a certain extent. This paper describes a method, which transforms the original auto-correlated envelope into a tempo-independent rhythmic feature vector by multiplying the lag-axis with a stretch factor. This factor is computed with a new correlation technique which works in the logarithmic domain. The proposed method is evaluated for rhythmic similarity, consisting of two tasks: One test with manually created rhythms as proof of concept and another test using a large real-world music archive.

## 1. INTRODUCTION

During the last years the need of new search and retrieval methods for digital music increased significantly due to the almost unlimited amount of digital music on users hard disks and in online stores. An important pre-requisite for these search methods is the semantic classification, which requires suitable low- and mid-level features. The major goal of many researchers is the computation of mid-level representations from audio signals, which are destined to capture the rhythmic gist from the music. A huge amount of work has been done in this field so far by developing techniques like beat histogram, inter-onset-interval histogram or rhythmic mid-level features, e.g., [1], [2], [3], [4], [5]. In general, the beat histogram technique very often used as feature basis for semantic classification. This histogram is computed by taking the audio spectrum envelope signal, which is differentiated and half/full-wave rectified. As a final step an auto-correlation function is ap-

plied, which estimates the periodicities within the modified envelope. The resulting feature vector is only limited usable for pattern recognition. Two similar rhythms are easily comparable with the beat histogram as feature, if their tempi are equal. A different tempo leads to a compression or expansion of the lag-axis, as depicted in Figure 1. This modification has a disadvantageous effect when performing a comparison of beat histograms via Euclidean distance measure. This issue has been raised by Foote [6]. A number of approaches tried to come up with solutions for that challenge. Paulus [7] presented a method, which could be considered reasonable for comparing beat histogram vectors containing different tempi by applying a dynamic time warping technique. A similar approach has been also proposed by Holzapfel [8]. These techniques require specialized classifiers and the beat histogram cannot be used as feature in conjunction with other low-level features. In order to solve that problem, Tzanetakis [1], Gouyon [2], and Burred [3] computed descriptive statistics, such as mean, variance, and kurtosis on the beat histogram. These statistics were used as feature vector for classification. To a certain degree, these are also tempo-dependent. This paper suggests a new post-processing method which performs a transformation of the beat histogram into the logarithmic lag domain. The transformation into the logarithmic domain has not been described for rhythm features, but for harmonic and chroma features in [9] and [10]. This transformation transfers the multiplicative factor of the tempo changes into an additive offset. Hence, the transformed rhythmic feature vector contains a tempo independent part located on the right-hand side of the vector. An approach for detection of this tempo independent rhythmic information is presented. A number of different features were extracted and evaluated for the task of rhythmic similarity.

The remainder of this paper will be organized as follows: Section 2 introduces the proposed algorithm, Section 3 describes the evaluation and discusses the results. Section 4 concludes and indicates further directions in this area.

## 2. PROPOSED APPROACH

In this work, the beat histogram is extracted from MPEG-7 AudioSpectrumEnvelope (ASE) features [11]. Different variants of the basic feature extraction algorithm have been reported in literature. Tzanetakis' [1] work was based on a wavelet transform, Scheirer [12] used a filter bank. Nev-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2009 International Society for Music Information Retrieval.

ertheless, both authors extracted an envelope signal from non-linearly spaced frequency bands, as is the case with ASE. In the proposed implementation, the different ASE bands are smoothed in time. Subsequently, the bands are weighted by enhancing the lower and higher frequency bands and decreasing the center frequencies. All bands are accumulated, differentiated in time, and full-wave-rectified. This results into a so-called detection function, containing the the most salient rhythmic information of the music signal. The detection function is subdivided into snippets of  $N$  successive frames. The auto-correlation inside such a frame yields the beat histogram, also called rhythmic mid-level feature, beat spectrum, etc.

The beat histogram may be used in a different number of applications, such as beat tracking or tempo detection. As already mentioned in Chapter 1, this vector should not be directly utilized for classification. If two similar rhythms are played in different rhythms and there beat histograms are compared, the vectors would look similar, but one would be a more stretched or compressed (in terms of the lag-axis) version from the other. Hence, a direct comparison of these vectors using common distance measures (e.g., Euclidean distance) results in large distances. Thus, it is state of the art to compute descriptive statistics from the beat histogram and use these measures as features for classification. Unfortunately, these statistics are also prone to tempo changes.

In order to create a tempo independent beat histogram, Foote [6] proposed to stretch or compress the original vector based on the tempo of the rhythm. The compression of the beat histogram can be considered as multiplication of a time-stretching factor  $f$  with the argument  $\tau$  of the underlying pattern signal  $c(\tau')$ . This pattern signal can be the mentioned auto-correlation signal. The observed feature vector can therefore be described with  $c(\tau) = c(\tau' * f)$ . In order to obtain the tempo invariant beat histogram  $c(\tau')$ , the stretch factor  $f$  needs to be known, but its automatic computation might be unreliable. One option for solving this issue is to use a logarithm function. By applying the logarithm on an arbitrary function, multiplicative terms are transformed to additive terms. Transferring this theorem to the lag-axis of the beat histogram  $c(\tau)$  leads to the equation (1):

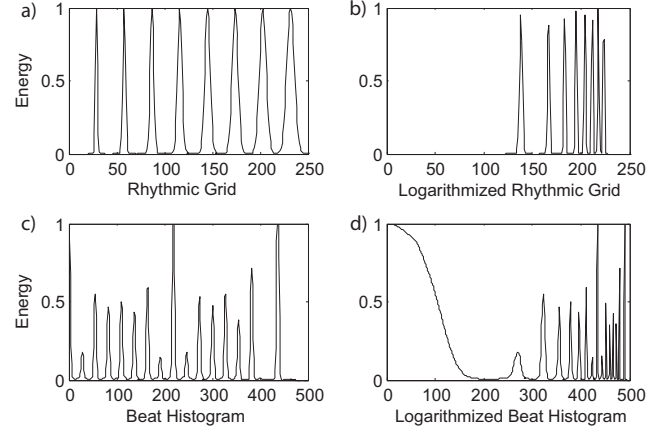
$$c(\log(\tau' * f)) = c(\log(f) + \log(\tau')) \quad (1)$$

For the logarithmic processing step, a new argument is estimated by (2):

$$\tau_{log} = \frac{\log(\tau) * \max(\tau)}{\log(\max(\tau))} \quad (2)$$

Resampling the original beat histogram  $c(\tau)$  in such a way, that the values in  $\tau$  are available on places of  $\tau_{log}$  results in a new beat histogram feature with logarithmized lag-axis (Figure 2 d).

Since  $\tau_{log}$  consists of non-integer values, the practical implementation of this variable requires an interpolation. For this task, a bicubic interpolation method as described in [13] has been applied.

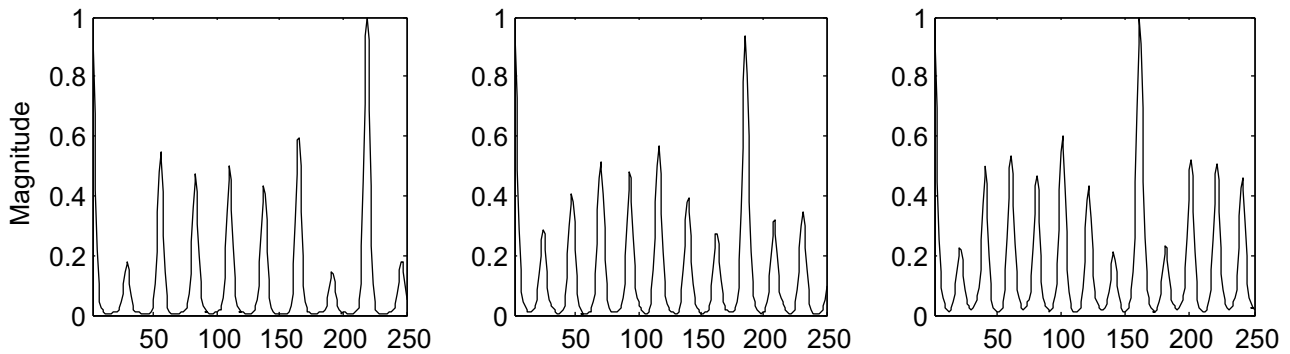


**Figure 2.** This figures shows an example beat histogram (c) and a rhythmic grid (a) and their logarithmic counterparts (d,a, respectively).

Figure 2 c,d shows an example beat histogram and its transformation into the log-lag domain.

By inspecting a large number of such logarithmized vectors it can be observed, that all vectors consist of a large decaying slope towards a first local minimum, whose absolute position depends on the tempo of the music. That slope represents the first maximum lobe of the auto-correlation function. Due to the fact, that a time-varying signal is always most similar to itself for small lags, the first lobe is always the highest and does not carry any significant rhythmic information. However, the successive minimum appears to be the point from where on the logarithmized beat histogram shows similar tempo-independent characteristics if the rhythm is similar. These characteristics are similar, but they are moved further right or further left, depending on the tempo. The goal is to find the starting point of these tempo-independent characteristics and to use the tempo-independent excerpt of the feature vector for classification. In the original beat histogram the first local minimum (or maximum) could be used as starting point for stretching or compressing the vector in order to receive a tempo-independent version. Unfortunately, this procedure is only applicable on a minority of rhythms, since often the first local minimum is misleading and the stretched vector results in octave errors. In the log-lag domain the result would be similar, if only the first minimum is used. The proposal in this publication is to find the point more reliably by taking the evolution of the vector into account. Therefore, the authors use an artificial rhythmic grid featuring eight successive Gaussian pulses as depicted in Figure 2 a. The Gaussian pulses are computed as described in the following Matlab code snippet (Code 1) with the block-size `blksize` as functional parameter and `tmp_acf` as result vector.

This rhythmic grid is transformed into the logarithmic domain with the same method as described above. In order to find the tempo-independent characteristics of the logarithmized beat histogram, both vectors, the logarithmized rhythmic grid and the logarithmized beat histogram are



**Figure 1.** These figures depict a beat histogram excerpt for the same rhythm with tempos of 90 Bpm (left), 110 Bpm (middle), 130 Bpm (right).

---

**Code 1** Example Matlab code for the creation of Gaussian pulses

---

```
mu = [29:29:blksize]; sd = 2;
tmp_acf = zeros(1,blksize); lobe=[];
for k = 1:length(mu)
t_exp=-0.5*((1:blksize)-mu(k))/sd).^2;
lobe(k,:) = exp(t_exp)/(sd*sqrt(2*pi));
lobe(k,:) = lobe(k,)/max(lobe(k,:));
tmp_acf = tmp_acf + lobe(k,:);
end
```

---

cross-correlated. Best results could be achieved by only evaluating only the first slope (histogram points 200-300 in 2 d). The maximum of the correlation function equals the point in the vector, where the tempo-independent characteristic starts. A faster tempo results in a shift of the tempo-independent part to the left, and thus additional peaks appearing at the right border. In order to process almost identical beat histograms, regardless of the tempo, the length of the tempo independent characteristics has to be suitably restricted. This tempo independent vector could be theoretically used as feature vector for rhythmic similarity. Due to the interpolation for the logarithmic processing, small variations lead sometimes to a small movement either to the right or to the left side of the axis. These small variations affect the rhythmic similarity negatively. In order to reduce this effect, statistical measures as proposed by the other authors have been applied in the tests for this paper. The following statistics as described by Tzanetakis [1], Gouyon [4], and Burred [3] were computed from the tempo independent vector. All statistics from these authors were appended and formed the final feature vector for the experiments:

- Tzanetakis: Relative amplitude (divided by the sum of amplitudes) of the first, and second histogram peak; ration of the amplitude of the second peak divided by the amplitude of the first peak; period of the first, second peak in bpm; overall sum of the histogram
- Gouyon: Mean of magnitude distribution; geometric mean of magnitude distribution; total energy; cen-

triod; flatness; skewness; high-frequency content

- Burred: Mean; standard deviation; mean of the derivative; standard deviation of the derivative; skewness; kurtosis and entropy.

Since some statistics from Gouyon and Burred partly overlapped the final feature size consisted of 18 dimensions. For the practical implementation, excerpts of 500 ASE frames were chosen, which corresponds to 5 seconds in music, given a low-level hop-size of 10 milliseconds. This size constitutes a trade-off between the length of at least two repeating patterns and the ability to track abrupt tempo changes sometimes encountered in real-world music. A correlation size of 5 seconds has been also used in previous approaches (e.g., [14]). Since the test songs contain more than five seconds of audio content, one of such a feature vector is computed every 0.5 seconds. In order to compute the Gaussian pulses, a default standard deviation of 2 has been chosen and only eight successive pulses were used in the evaluation. Another standard deviation could also be chosen, which increases/decreases the width of the pulses.

For the tests in this paper, the following 4 feature vectors were created:

- Statistics of original beat histogram: The beat histogram has been extracted as described in this paper. Based on that histogram, a feature vector containing all statistics by Tzanetakis [1], Gouyon [4], and Burred [3] as described above was extracted.
- Statistics of logarithmized beat histogram: The statistics by Tzanetakis, Gouyon, and Burred were computed from the logarithmized beat histogram technique as described above.
- Statistics of beat histogram with stretch factor: Based on the logarithmized beat histogram, a point has been estimated, where the tempo-independent rhythmic characteristic begins. This point has been transformed into the non-logarithmic domain and a stretch factor (as proposed by Foote) has been computed. The original beat histogram has been stretched by the

stretch factor and the statistics from Tzanetakis, Gouyon, and Burred were computed from that vector.

- Beat histogram with stretch factor: The original beat histogram has been stretched as suggested by Foote with the stretch factor derived from the logarithmic post-processing.

### 3. EVALUATION

#### 3.1 Evaluation Procedure

In order to test the logarithmic post-processing of the beat histogram, two different evaluation strategies were implemented. The first test evaluated a number of manually created rhythms in order to prove the theoretic improvement of the results. The second test evaluates rhythmic similarity based on beat histograms with a large real-world music set.

##### 3.1.1 Tests based on manually created rhythms

The first test scenario examined the tempo dependence of the described feature sets based on different rhythms. A number of 18 different base rhythms were established, which can be divided into 9 rhythm genres, e.g., electro, drum'n'hop or hip hop. The rhythms were played without any additional instruments in order to test the tempo dependence of only the base rhythms. Each of these rhythms was played in six different tempo variations ranging from 90 Bpm to 190 Bpm in 20 Bpm steps. Each base rhythm was repeated a number of times, whereby the duration of one single rhythm pattern was less than 5 seconds. A total of 108 rhythms were collected and the low-level ASE features as well all four versions of the described mid-level features were extracted. Since the window length of the described mid-level features consisted of 5 seconds, the base rhythm of every rhythm class is contained in every frame of the feature matrix. Therefore, an arbitrary frame from the feature matrix can be chosen for comparison. In the evaluation for this paper, the second consecutive vector was used as mid-level feature. Prior to the classification, a mean and a variance normalization step over all data was applied. A simple k-nearest neighbor classifier with Euclidean distance was set up using the features and the rhythm class information as ground-truth.  $k$  for the k-nearest neighbor classifier has been chosen to be one. Subsequently, all features were consecutively used as query to the classifier, whereby it has been ensured, that the query item was not contained in the reference set. The evaluation method returned the distance and the closest class to each of the 108 rhythms. The average accuracy has been estimated per class. The minimum, maximum and average of the overall test set has been estimated by using the class-dependent accuracy. Based on the results of this simple classifier a base-line assumption can be made about the accuracy of the tempo independent rhythmic classification. One might raise concerns that the comparison of base rhythms is not very practice relevant, since popular music contains additional polyphonic properties in the signal, which may interfere with the beat histogram. In order to prevent this

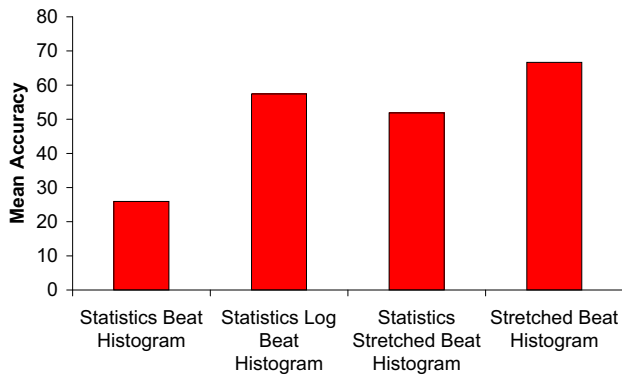
”distortion” it has been shown, e.g. in [15], that drum transcription algorithms as preprocessing steps have a positive effect on beat histogram.

##### 3.1.2 Tests based on a large test set

To evaluate the performance on real world data instead of the rather artificial data, a diverse set of 753 songs from 60 different genres and sub-genres was compiled. Rhythmic similarity measures are hard to evaluate by using real world data. An option for testing rhythmic similarity measures can be based on the assumption, that songs from the same genre have similar rhythms, while songs from different genres have different rhythms. But similar rhythms might be also available across genres and the results would not directly predicate rhythmic similarity. To cope with that, another approach was chosen. A rhythm similarity ground truth was manually created for the used dataset. First, for each song, a representative rhythm pattern was annotated by hand, then a similarity matrix from all pairs of rhythms was calculated.

Representative rhythm pattern: For each song, one representative rhythm pattern was manually annotated. Five different classes of rhythmical events were differentiated: base drum, snare drum, hi hats, further percussive events, and non-percussive events. A quantization could be freely chosen, but in general, events have been quantized onto 1/16 bar length in case of a 4/4 bar and 1/12 bar length in case of a 3/4 bar. Similarity between patterns: The distance between two characteristic patterns was calculated by performing the following steps. First, both of the patterns have been stretched onto the same length. Then, all the simultaneous occurrences of an event of a certain class in both patterns were summed up. Finally, the resulting value was normalized by the length of the pattern. For each of the mentioned percussion classes, the 753x753 distance matrix was computed. Afterwards, the mean distance matrix was estimated by equally weighting all distances of the distance matrices from each percussion class.

Also, for each song in the database the features described above were extracted whereby the mean value for all feature frames of a song was calculated. Using Euclidean distance, the 5 closest songs to each song excluding the query itself were determined. The list of the 5 closest songs to the query song  $C$  are denoted  $L_C$ . Incorporating both the ground-truth rhythm similarity matrix and the list of the 5 closest songs for each of the 753 queries, the different feature sets were compared using the following procedure: For each query song  $C$ , a list  $T_C$  of all the other songs, was generated. This list was sorted in ascending order of the distances derived from the manually annotated rhythm patterns. Then, for each song  $c$  in  $L_C$  the number of songs in  $T_C$  have been counted, which were closer to  $C$  than  $c$ . By averaging over these numbers, a value  $r$  is calculated. This value describes the mean number of songs in  $T_C$  that are closer to the query song than the retrieved songs. In order to obtain a statement about the accuracy of the system in such a way, that higher numbers refer to better results, a score has been computed by  $S_i = |S - 1|$ .



**Figure 3.** Average accuracy for rhythmic classification of the first test based on different feature vectors in percent.

	Mean	Min	Max
Stat. Original Beat Hist.	25.93	0.00	83.33
Stat. Logarithm. Beat Hist.	57.41	0.00	100.00
Stat. Stretched Beat Hist.	51.85	16.67	66.67
Stretched Beat Hist.	66.67	33.33	100.00

**Table 1.** Accuracy measures (first test) for rhythm classification based on different feature vectors in percent.

This score is referred to the term similarity index. For significance purposes a random score has been established by generating a random result list for each of the 753 songs. This result list has been evaluated in a similar procedure as the described mid-level features.

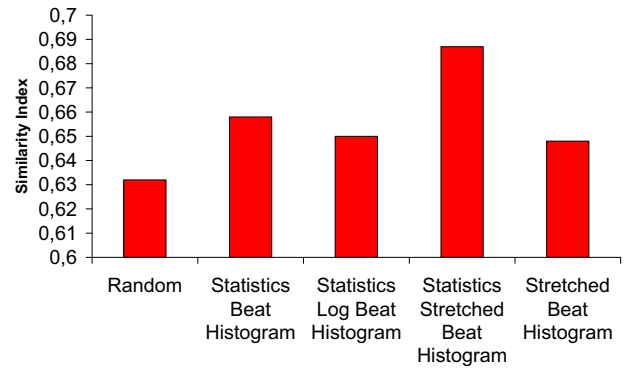
Other rhythmic similarity measures were described in literature by Hofman-Engl [16] and Toussiant [17]. These measures are established when it comes to the comparison of actual rhythmic descriptions. In this paper features based on rhythms are to be compared. Therefore, these methodologies could not be applied.

## 3.2 Results and Discussion

### 3.2.1 Test based on manually created rhythms

The following table (Table 3.2.1) shows the results for the first test containing the manually created rhythms. This table shows minimum, maximum and mean accuracy. In order to get a quick overview about the results in general, the mean is also plotted in Figure 3.

The state of the art methodology by computing statistics over the beat histogram achieves an average accuracy of approx. 26%. This is based on the fact, that the statistic measures are by far not tempo independent. Better results could be obtained by the logarithmic post-processing step. The statistics computed on the logarithmized beat histogram and over the stretched beat histogram performed reasonably well with 57.4% and 51.9%, respectively. The best results could be obtained by the stretched beat histogram with the stretch factor computed from the logarithmized beat histogram. This methodology leads to an average accuracy of 66.7%. An intuitive guess would be, that identical rhythms in different tempos should always return an accuracy of 100%. In practice, the results look differ-



**Figure 4.** Similarity index (second test) expressing the rhythmic similarity for different feature vectors.

ent due to windowing effects. The minimum accuracy of the algorithms ranges from 0% to 33.3%. This is based on the fact, that the separability between some of the 18 base rhythms is strongly restricted. The highest accuracy is obtained by the stretched beat histogram also in case of the minimum. This might imply that postprocessed beat histogram performs better as feature than the statistics over postprocessed beat histograms. A similar statement can be also made by evaluating the maxima of the four feature vectors. These tests prove, that the tempo independent version of the beat histogram (stretched beat histogram) outperforms the statistics over the beat histogram.

### 3.2.2 Test based on a large real-world music set

The following figure (Figure 4) shows the accuracy for the test with real-world music. Additionally, these numbers are depicted in Table 3.2.2. The similarities between manually annotated base rhythms and the beat histogram features are expressed by a similarity index. The higher the index is, the better is the similarity between the manually annotated rhythms and the automatically extracted rhythms. The figure shows, that a random generation of similarities results with a similarity index of 0.632. Most of the observed feature vectors obtained a similarity index around 0.65, including the statistics over the beat histogram, the statistics over the logarithmized beat histogram and the stretched version of the beat histogram. The statistics computed from the stretched beat histogram outperform all other results by a similarity index of 0.03.

The first test, which was based on the manually created rhythms, showed the best results on the stretched beat histogram. In this second test, these results cannot be validated in every case. This may be based on the fact that the point in the logarithmic domain, which separates the tempo dependent and tempo independent parts is inaccurate in a few cases. These inaccuracies have influence on the stretched beat histogram and may result in octave errors, which affect the rhythmic similarity. However, computing the descriptive statistics over the resulting vectors improves the results. These statistics seem to neglect the slight deviations significantly. This test on real world data might be not optimal, since rhythms in real songs might

Feature Name	Similarity Index
Random	0.632
Stat. Original Beat Hist.	0.658
Stat. Logarithm. Beat Hist.	0.650
Stat. Stretched Beat Hist.	0.687
Stretched Beat Hist.	0.648

**Table 2.** Similarity index of the second test expressing the rhythmic similarity for different feature vectors.

change and the evaluation was performed on one representative rhythm of the song. But this methodology gives a rough indication of the performance of the logarithmic processing.

#### 4. CONCLUSIONS AND FUTURE WORK

The rhythmic information from music is captured by the commonly used beat histogram. This paper presented a post-processing technique for the beat histogram, which is based on logarithmic re-sampling of the lag axis and cross-correlation with an artificial rhythmic grid. This technique seems to improve the applicability of the beat histogram technique as feature for music information retrieval tasks. The practical tests on a large music archive were based on a mean feature vector per song. In order to be more accurate, future tests should perform a rhythmic segmentation and analyze the segments individually. The logarithmic processing methodology as described in this paper may be also beneficial for beat tracking and tempo detection. Future tests will provide an evaluation, if the tempo estimation results can be improved when using the proposed algorithm.

#### 5. ACKNOWLEDGMENT

This work has been partly supported by the PHAROS Integrated Project (IST-2005-2.6.3), funded under the EC IST 6th Framework Program. Additionally, this project has been funded by the MetaMoses project (nr. 183217) from the Norwegian research council.

#### 6. REFERENCES

- [1] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech, Audio, and Language Processing*, 10(5):293–302, 2002.
- [2] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1), 2005.
- [3] J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, 2003.
- [4] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the 25th AES International Conference*, 2004.
- [5] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the 25th AES International Conference*, 2004.
- [6] J. Foote and S. Uchihashi. The beat spectrum: A new approach to rhythm analysis. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, 2001.
- [7] J. Paulus and A. Klapuri. Measuring the similarity of rhythmic patterns. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR)*, 2002.
- [8] A. Holzapfel and Y. Stylianou. A scale transform based method for rhythmic similarity of music. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.
- [9] S. Saito, H. Kameoka, T. Nishimoto, and S. Sagayama. Specmurt analysis of multi-pitch music signals with adaptive estimation of common harmonic structure. In *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005.
- [10] J. Jensen, M. Christensen, D.P.W. Ellis, and S. Jensen. A tempo-insensitive distance measure for cover song identification based on chroma features. In *Proceedings of the IEEE International Conference on Audio, Acoustics, and Signal Processing (ICASSP)*, 2008.
- [11] M. Casey. Mpeg-7 sound recognition. *IEEE Transaction on Circuits and Systems Video Technology, special issue on MPEG-7*, 11:737–747, 2001.
- [12] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.
- [13] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.
- [14] S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR)*, 2003.
- [15] M. Gruhne and C. Dittmar. Improving rhythmic pattern features based on logarithmic preprocessing. In *Proceedings of the 126th Audio Engineering Society (AES) Convention*, 2009.
- [16] L. Hofmann-Engl. Rhythmic similarity: A theoretical and empirical approach. In *Proceedings of the Seventh International Conference on Music Perception and Cognition*, 2002.
- [17] G.T. Toussaint. A comparison of rhythmic similarity measures. In *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004.