A TEMPO-INSENSITIVE REPRESENTATION OF RHYTHMIC PATTERNS

Jesper Højvang Jensen, Mads Græsbøll Christensen and Søren Holdt Jensen

Department of Electronic Systems
Aalborg University, Denmark
email: {jhj, mgc, shj}@es.aau.dk

ABSTRACT

We introduce a representation for rhythmic patterns that is insensitive to minor tempo deviations and that has well-defined behavior for larger changes in tempo. We have combined the representation with an Euclidean distance measure and compared it to other systems in a classification task of ballroom music. Compared to the other systems, the proposed representation shows much better generalization behavior when we limit the training data to songs with different tempi than the query. When both test and training data contain songs with similar tempo, the proposed representation has comparable performance to other systems.

1. INTRODUCTION

Together with timbre and melody, rhythm is one of the basic properties of Western music. Nevertheless, it has been somewhat overlooked in the music information retrieval community, perhaps because rhythm is a quite abstract concept that is difficult to describe verbally. A manifestation of this is that in an online music tagging game, Mandel noted that except for the occasional use of the word “beat”, hardly any tags were describing rhythm [1]. This suggests that a computational measure of rhythmic distance could supplement a word-based music search engine quite well. An indication that rhythmic similarity has been largely neglected is the audio description contests that were held in conjunction with the International Conference on Music Information Retrieval (ISMIR) in 2004 to compare the performance of different algorithms [2]. Among these evaluations was an automated rhythm classification task, where [3] was the *only* participant. While other tasks such as genre classification were quite popular and have recurred in the Music Information Retrieval Evaluation eXchange (MIREX) as a direct continuation of the ISMIR 2004 evaluation, the rhythm classification task has to date not been repeated. Fortunately, the ballroom music used for the evaluation has been released (see Table 1 and Figure 1).

Some of the first systems for rhythm matching were described by Foote et al. [4], who used a self similarity matrix to obtain a beat spectrum that estimates the periodicity of songs at different lags; Paulus and Klapuri [5] who among others use dynamic time warping to match different rhythms; and Tzanetakis and Cook [6] who used an enhanced autocorrelation function of the temporal envelope and a peak picking algorithm to compute a beat histogram as part of a more general genre classification framework. More recent systems include [3, 7–9]. Seyerlehner et al. also use a measure of distance between rhythmic patterns, although with the purpose of tempo estimation [10]. For a review of rhythm description systems, see e.g. [11].

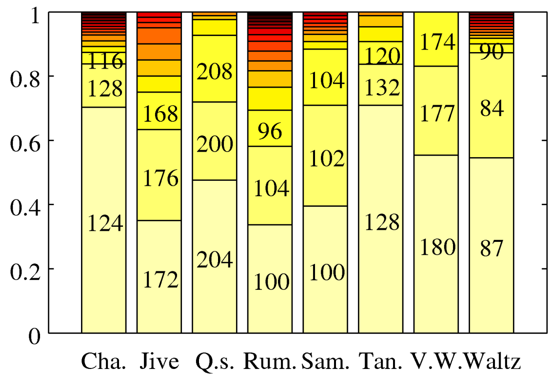

Figure 1: Distribution of tempi for the different rhythmic styles in the ballroom dataset. For the three most common number of beat per minutes (BPMs), the value is shown.

Several authors have observed that tempo is an important aspect of matching songs by rhythm [8, 12, 13]. Using the Ballroom dataset (see Table 1 and Figure 1), Gouyon reports a classification accuracy of 82% from the annotated tempi alone, although the accuracy decreases to 53% when using estimated tempi [14]. Peeters reports that combining rhythmic features with the annotated tempi typically increases classification accuracy by around 15% [8]. Seyerlehner et al. have gone even further and have shown that a nearest neighbor classifier that matches the autocorrelation function of the

Table 1: Distribution of rhythmic styles and training/test split for the music used in the ISMIR 2004 rhythm classification contest. The set consists of 698 clips of ballroom music from `http://ballroomdancers.com/`.

Style	Num. clips	Training	# Test
Cha-cha-cha	111	78	33
Jive	60	42	18
Quickstep	82	57	25
Rumba	98	69	29
Samba	86	60	26
Tango	86	60	26
Viennese Waltz	65	45	20
Waltz	110	77	33

This research was supported by the Intelligent Sound project, Danish Technical Research Council grant no. 26–04–0092, and the Parametric Audio Processing project, Danish Research Council for Technology and Production Sciences grant no. 274–06–0521.

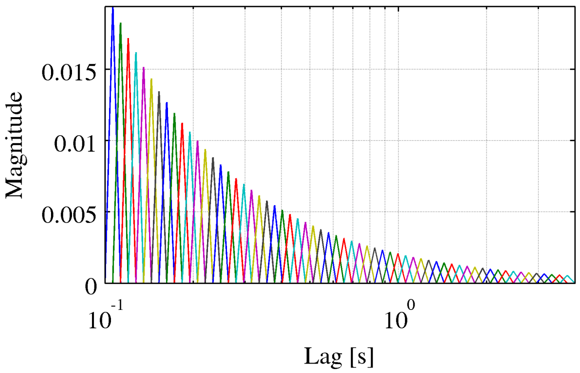

Figure 2: The 60 exponentially distributed bands the autocorrelation values are merged into.

envelope performed on par with state of the art tempo induction systems [10], suggesting that tempo estimation can be considered a special case of rhythmic pattern matching. Davies and Plumbley [15] take the opposite approach and use a rhythmic style classifier to improve tempo estimates by letting the prior probabilities of different tempi be a function of the estimated style.

Since rhythm and tempo are so critically linked, we propose a representation of rhythmic patterns that is insensitive to small tempo variations, and where the effect of large variations is very explicit. The representation is based on the melodic distance measures we presented in [16, 17], which were designed to find cover songs, i.e. different renditions of the same song. To make features insensitive to the tempo variations that are inevitable when artists interpret songs differently, we averaged intensities over exponentially spaced bands, which effectively changes a time scaling into a translation. In this paper, we apply the same idea to a measure of rhythmic distance. In Section 2, we describe the proposed representation of rhythmic patterns. In Section 3, we use a nearest neighbor classifier based on the Euclidean distance between the proposed feature to evaluate the performance of the representation on the ballroom dataset. In Section 4, we discuss the results.

2. A TEMPO-INSENSITIVE RHYTHMIC DISTANCE MEASURE

Our proposed rhythmic distance measure is inspired by [10], which again is based on [18]. The first steps proceed as in [10, 18]:

1. For each song, resample it to 8 kHz and split it into 32 ms windows with a hop size of 4 ms.
2. For each window, compute the energy in 40 frequency bands distributed according to the mel-scale.
3. For each mel-band, compute the difference along the temporal dimension and truncate negative values to zero to obtain an onset function.
4. Sum the onset functions from all mel-bands into a single, combined onset function. If $P_b(k)$ is the energy of the b'th mel-band in the k'th window, the combined onset function is given by $\sum_b \max(0, P_b(k) - P_b(k-1))$.
5. High-pass filter the combined onset function.

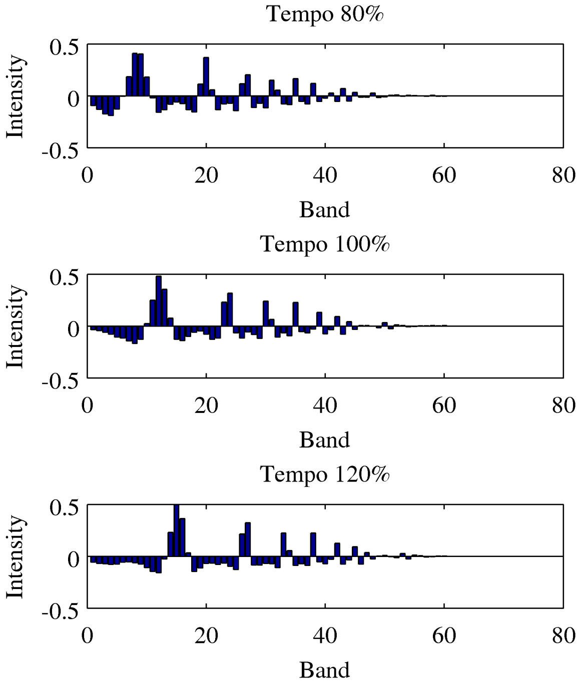

Figure 3: The resulting feature vector from a synthesized MIDI file with duration 80%, 100% and 120% of the original length, respectively. Note that the feature vectors are merely shifted versions of each other.

6. Compute the autocorrelation function of the high-pass filtered onset signal up to a lag of 4 seconds.

The autocorrelation function is independent of temporal onset, and it does not change if silence is added to the beginning or end of a song. However, as argued by Peeters [8] it still captures relative phase. While some different rhythmic patterns will share the same autocorrelation function, this is not generally the case. In particular, two rhythmic patterns build from the same durations, (e.g. two $\frac{1}{4}$ notes followed by two $\frac{1}{8}$ notes compared to the sequence $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{8}$) do not in general result in identical autocorrelation functions.

Unlike [10], who smoothes the autocorrelation function on a linear time scale, we use a logarithmic scale. That is, we split the autocorrelation function into the 60 exponentially spaced bands with lags from 0.1 s to 4 s that are shown in Figure 2. Viewing the energy of the bands on a linear scale corresponds to viewing the autocorrelation function on a logarithmic scale. Changing the tempo of a song would result in a scaling of the autocorrelation function along the x axis by a constant, but on a logarithmic scale, this would be a simple translation. This trick is used in e.g. [19] for fundamental frequency estimation to obtain a representation where the distances between the fundamental frequency and its harmonics are independent of the fundamental frequency. With the exponentially spaced bands, a small change of tempo does not significantly change the distribution of energy between the bands, while larger changes will cause the energy to shift a few bands up or down. We collect the band outputs in a 60-dimensional feature vector $\mathbf{x}$ that has the energy of the n'th

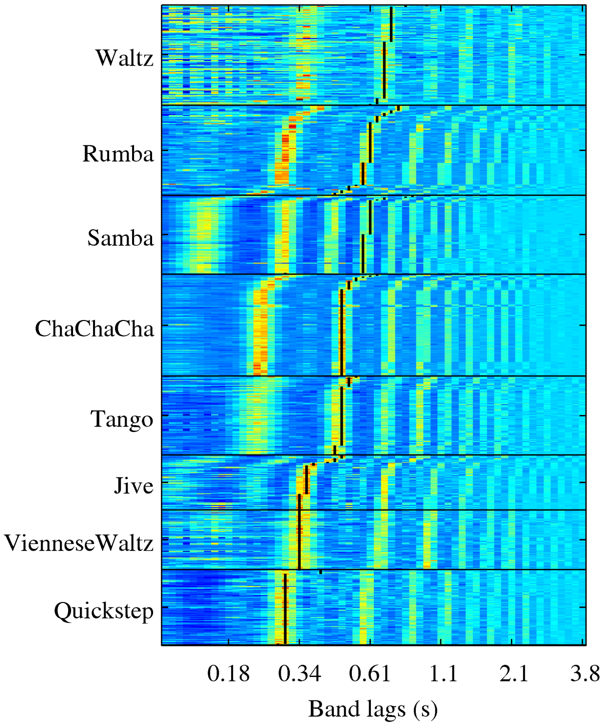

Figure 4: Features from the ballroom dataset. Within each style, the features are sorted by the annotated tempo. The band with a lag that corresponds to the annotated tempo (i.e., 120 bpm corresponds to 0.5 s) is indicated by the black, vertical lines. The 60 bands along the x axis are denoted by lag time rather than index.

band as its n'th component, $(\mathbf{x})_n$. As the final step in the feature extraction process, we normalize the vector to have unit Euclidean norm. In Figure 3 and 4, we have shown the proposed feature extracted from the same MIDI file synthesized at three different tempi and from the ballroom dataset, respectively.

With 60 bands, the effective bandwidth of each band extends ±3% from the center frequency. Since a 3% change of tempo is hardly noticeable, in the evaluation we extend the permissible range of tempi by also searching for shifted versions of the feature vectors. Specifically, when we search for the nearest neighbor to a song with feature vector $\mathbf{x_m}$, we find the song whose feature vector $\mathbf{x_n}$ is the solution to

$$\arg\min_{\mathbf{n}} \min_{j\in\{-1,0,1\}} \|\mathbf{x_m}^j - \mathbf{x_n}\| \qquad (1)$$

where $\mathbf{x_m}^j$ is $\mathbf{x_m}$ shifted j steps, i.e.,

$$\mathbf{x_m}^j = \begin{cases} [(\mathbf{x_m})_2\ (\mathbf{x_m})_3\ \cdots\ (\mathbf{x_m})_{60}\ 0]^{\mathrm{T}} & \text{for } j=-1, \\ \mathbf{x_m} & \text{for } j=0, \\ [0\ (\mathbf{x_m})_1\ (\mathbf{x_m})_2\ \cdots\ (\mathbf{x_m})_{59}]^{\mathrm{T}} & \text{for } j=1. \end{cases} \qquad (2)$$

To obtain something similar with the linear autocorrelation sequence, we would need to resample it to different tempi. However, since the displacement of a peak at lag k is proportional to k, the number of resampled autocorrelation functions must be high to ensure sufficiently high resolution also for large k.

A Matlab implementation of the proposed system is available as part of the Intelligent Sound Processing toolbox[1].

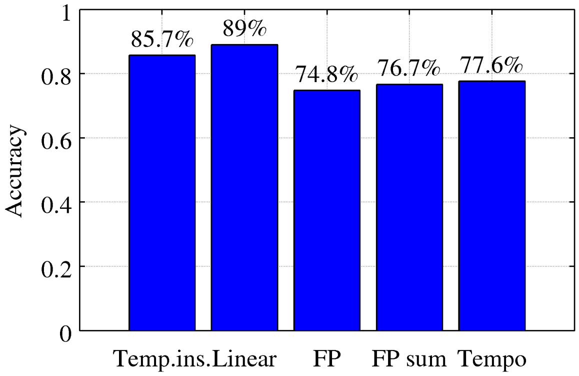

Figure 5: Rhythmic style and tempo classification results when allowing the distance measures to match on tempo. From left to right, the distance measures are our proposed tempo insensitive distance measure, the linear version from [10], the Fluctuation Patterns from [20], the modified version of the Fluctuation Patterns from [10], and finally the absolute difference between the songs' ground truth tempi.

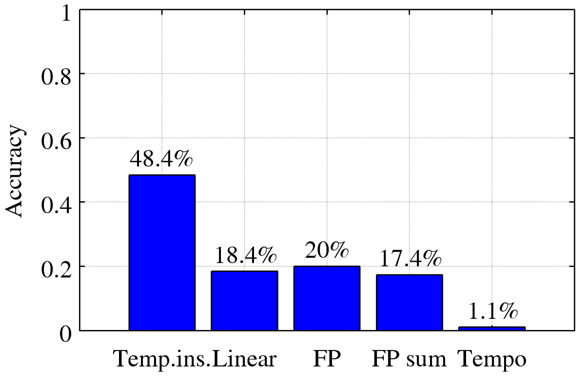

Figure 6: Rhythmic style and tempo classification results when ignoring potential nearest neighbors with the same style and similar in tempo to the query.

3. EXPERIMENTS

Using the ISMIR 2004 ballroom dataset, we have compared the linear autocorrelation as proposed by [10], our proposed logarithmic version, the fluctuation patterns from [20], and the modification to the fluctuation patterns also proposed in [10]. As a reference, we have also used the absolute difference between the true tempi of songs. We have compared the rhythmic distance measures using two different setups. First, we have used the ballroom dataset as intended by finding the nearest neighbor in the training set to each song in the test set and see how often the rhythmic styles match. These results are shown in Figure 5. Note that since we use the same par-

[1] `http://isound.es.aau.dk/`

titioning into test and training as in the ISMIR 2004 contest, results are comparable to [2], but although the numbers are similar, the results are not directly comparable to e.g. [8–10] who all use 10-fold cross validation.

To see how much these results depend on tempo, we have repeated the experiment with the difference that when searching for the nearest neighbor to a query song, we reject candidates that have the same rhythmic style and a tempo that is within 4% of the query (we use 4% similarly to [10]). The results when incorporating this constraint are shown in Figure 6. Test songs with Viennese Waltz had to be ignored when computing the accuracy, since their tempi are all within 5% of each other.

4. DISCUSSION

By constructing a measure of rhythmic distance that is designed to be insensitive to different tempi, we sacrifice a few percentage points of performance in the baseline test in Figure 5, where the linear autocorrelation function has the highest performance. However, as seen in Figure 6, if the songs in the training set with the same rhythmic style as the query do not include songs that also share the same tempo, our proposed distance measure significantly outperforms the other distance measures. Due to the good generalization behaviour, we expect our proposed measure to supplement for instance a timbre-based music search engine quite well.

Several aspects of the proposed distance measure are somewhat arbitrary, leaving room for improvement. For example, using other onset functions, e.g. the one used in [21], or using more sophisticated classification algorithms, such as support vector machines, might increase performance.

An interesting aspect of our proposed representation of rhythmic patterns is that by simply shifting the feature vector, it allows searching for slower or faster music with a similar rhythmic structure. This could e.g. be useful if listening to music when exercising, where the push of a button could find similar, faster music that better matches ones pulse.

REFERENCES

[1] M. I. Mandel and D. P. W. Ellis, “A web-based game for collecting music metadata,” in *Proc. Int. Symp. on Music Information Retrieval*, 2007.

[2] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, “Ismir 2004 audio description contest,” Music Technology Group of the Universitat Pompeu Fabra, Tech. Rep., 2006.

[3] T. Lidy and A. Rauber, “Genre-oriented organization of music collections using the somejb system: An analysis of rhythm patterns and other features,” in *Proc. of the DELOS Workshop on Multimedia Contents in Digital Libraries*, 2003.

[4] J. Foote, M. Cooper, and U. Nam, “Audio retrieval by rhythmic similarity,” in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 265–266.

[5] J. Paulus and A. Klapuri, “Measuring the similarity of rhythmic patterns,” in *Proc. Int. Symp. on Music Information Retrieval*, 2002, pp. 150–156.

[6] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[7] S. Dixon, F. Gouyon, and G. Widmer, “Towards characterisation of music via rhythmic patterns,” in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 509–516.

[8] G. Peeters, “Rhythm classification using spectral rhythm patterns,” in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 644–647.

[9] N. Scaringella, “Timbre and rhythmic trap-tandem features for music information retrieval,” in *Proc. Int. Symp. on Music Information Retrieval*, 2008, pp. 626–631.

[10] K. Seyerlehner, G. Widmer, and D. Schnitzer, “From rhythm patterns to perceived tempo,” in *Proc. Int. Symp. on Music Information Retrieval*, 2008, pp. 519–524.

[11] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, “A review of automatic rhythm description systems,” *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.

[12] S. Dixon, E. Pampalk, and G. Widmer, “Classification of dance music by periodicity patterns,” in *Proc. Int. Symp. on Music Information Retrieval*, 2003.

[13] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, “Evaluating rhythmic descriptors for musical genre classification,” in *Proc. Int. AES Conference*, 2004, p. 196204.

[14] F. Gouyon, “A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing,” Ph.D. dissertation, University Pompeu Fabra, 2005.

[15] M. E. P. Davies and M. D. Plumbley, “Exploring the effect of rhythmic style classification on automatic tempo estimation,” in *Proc. European Signal Processing Conf.*, 2008.

[16] J. H. Jensen, M. G. Christensen, D. P. W. Ellis, and S. H. Jensen, “A tempo-insensitive distance measure for cover song identification based on chroma features,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2008, pp. 2209–2212.

[17] J. H. Jensen, M. G. Christensen, and S. H. Jensen, “A chroma-based tempo-insensitive distance measure for cover song identification using the 2D autocorrelation function,” in *Music Information Retrieval Evaluation eXchange*, 2008.

[18] D. P. W. Ellis, “Beat tracking with dynamic programming,” in *Music Information Retrieval Evaluation eXchange*, 2006.

[19] S. Saito, H. Kameoka, T. Nishimoto, and S. Sagayama, “Specmurt analysis of multi-pitch music signals with adaptive estimation of common harmonic structure,” in *Proc. Int. Symp. on Music Information Retrieval*, 2005, pp. 84–91.

[20] E. Pampalk, “A Matlab toolbox to compute music similarity from audio,” in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[21] M. Alonso, G. Richard, and B. David, “Accurate tempo estimation based on harmonic + noise decomposition,” *EURASIP J. Applied Signal Processing*, vol. 2007, 2007, doi:10.1155/2007/82795.