FROM RHYTHM PATTERNS TO PERCEIVED TEMPO

Klaus Seyerlehner[1], **Gerhard Widmer**[1,2], **Dominik Schnitzer**[1,2]
[1]Department of Computational Perception
Johannes Kepler University Linz, Austria
[2]Austrian Research Institute for Artificial Intelligence, Vienna

ABSTRACT

There are many MIR applications for which we would like to be able to determine the perceived tempo of a song automatically. However, automatic tempo extraction itself is still an open problem. In general there are two tempo extraction methods, either based on the estimation of inter-onset intervals or based on self similarity computations. To predict a tempo the most significant time-lag or the most significant inter-onset-interval is used. We propose to use existing rhythm patterns and reformulate the tempo extraction problem in terms of a nearest neighbor classification problem. Our experiments, based on three different datasets, show that this novel approach performs at least comparably to state-of-the-art tempo extraction algorithms and could be useful to get a deeper insight into the relation between perceived tempo and rhythm patterns.

1 INTRODUCTION

The tempo is a basic and highly descriptive property of a song, and it is a feature that music users perceive in an intuitive and direct way. Tempo is one of the parameters a user would love to have under control. For instance, one can imagine that depending on her mood, a user would like to be able to choose faster or slower music. Therefore the *perceived tempo*, the perceived speed of music, would be a perfect and highly intuitive parameter for various new interfaces to music collections.

The automatic extraction of tempo information directly from digital audio signals has attracted a lot of research. Published tempo determination methods generally proceed in two stages: extracting low-level information related to apparent periodicities in the signal (we will use the generic term ‘‘rhythm patterns’’ for this information), and determining some assumed tempo from this information. To be useful in MIR applications, the tempo that is inferred by an algorithm should match the tempo human listeners intuitively perceive when listening to the music, not some ‘theoretical’ tempo that might be deduced from a written score of the piece. It is in this sense that we will use the term *perceived tempo* here, to denote the tempo that most human listeners would assign to a piece. (Of course, tempo may be perceived differently and may be ambiguous in certain cases, but in general, there will be substantial agreement.) In the evaluation of our tempo extraction method in this paper, we will use music collections manually annotated with tempo values; we will have to assume that the annotations correspond to the tempo that most listeners would perceive.

Despite a lot of literature on the subject, the human perception of rhythms, periodicity and pulsation is not yet very well understood. The same is true for the relation between rhythm and tempo perception. Compared to rhythm patterns, which we consider to be low-level features, the perceived tempo is a more abstract higher level feature. A common approach in tempo identification is to analyse the extracted rhythm patters (even if they not usually called so – see section 2.2 below), assuming that one of the periodicities present in the rhythm patterns corresponds to the perceived tempo. This is usually determined by using some simple peak picking algorithm to predict the most *salient* tempo. However, the relation between rhythmic patterns and tempo perception might be more complex, and simple peak picking algorithm might not be the most appropriate choice. While a lot of scientific work has focused on onset detection and rhythm pattern extraction, there has been little effort to gain further insight into this relation between rhythm and perceived tempo to improve the tempo estimation step. We propose to view this relation as a sort of machine learning problem. We model the overall *tempo extraction process* as a two stage process of *rhythm pattern extraction* and *tempo estimation*, and will investigate whether the prediction of perceived tempo from rhythm patterns can be learned by the computer.

2 BASIC NOTIONS

2.1 Perceived Tempo

Perceived tempo is not a well-defined mathematical concept, since the perception of the tempo of a song strongly depends on the listener. It is well known that a human’s preferred tempo range is around 100 to 120 bpm. But what makes perceived tempo estimation really difficult is the fact that the perceived tempo of certain pieces might be ambiguous, and human subjects may indeed perceive different tempi in the same piece, as has been confirmed in tapping experiments [10]. Not unexpectedly, many of the discrepancies in users’ tapping choices relate to different choices of the main metrical level; thus, users’ tempo

judgments often differ by a factor of two or three.

Interestingly, it is not yet clear if ambiguities known from tapping experiments really correspond to ambiguities in tempo perception. According to Chua et al. [1, 7], the *'Foot-tapping' tempo* is not always the same as the perceived tempo. This is supported by Zhu et al. [3], who carried out a user study where the subjects had to determine the tempo with respect to some reference clips. They finally conclude that the perception biases between different individuals are not large, so that the notion of a unique perceived tempo might still be practically useful.

To deal with the ambiguity problem, some authors have proposed to extract two main tempi as well as their relative strength (e.g., [10, 5]). For a tempo descriptor that should support user interaction, this is rather problematic. Imagine a user interface having a tempo slider, such that a user can influence the tempo by selecting another tempo range. Such an interface obviously presupposes a unique tempo value.

Because of our interest in a useful tempo descriptor for MIR applications, we decided to focus on the estimation of the *major* or *strongest* perceived tempo of a piece, the tempo most people would decide for. This seems reasonable, as many pieces have unambiguous perceived tempo. For instance, DJs rely on beat databases to increase the speed of the songs during an event from slow to fast music. In section 4.1 we will make use of such publicly available tempo information from beat databases to create a representative ground truth dataset.

2.2 Rhythm Patterns

All tempo extraction algorithm have more or less the same common structure. In a first stage, information related to potential periodicities is extracted from the digital audio signal (*rhythm pattern extraction process*), and in a second final stage a tempo is estimated from the extracted pattern (*tempo estimation process*). For a comprehensive overview on tempo extraction algorithms we refer to [4], who summarize the results of the `ISMIR'04` tempo extraction contest. A detailed description of various state-of-the art tempo extraction algorithms can also be found on the `MIREX'06` homepage [1]. In this section we give a brief overview of various representations of rhythmic structures one can extract from music signals. This is of interest since most of the tempo extraction algorithms do not explicitly generate a representation of the rhythm patterns, but instead directly try to estimate the tempo in a consecutive tempo estimation block. Therefore it is often not obvious how rhythm information is represented. Our approach, in contrast, depends on an explicit representation of rhythm information, and a basic overview on different types of rhythm patterns will illustrate what types of patterns (or probably combinations of those) might be useful.

One common method is to automatically detect onsets and then perform an analysis of inter-onset intervals (IOI). For event-based approaches the detection of note onsets is of major importance. Dixon [6] and Gouyon et al. [13] give interesting overviews of features suitable for onset detection. For most of the IOI based methods an **inter-onset histogram** can be generated, which represents the essential rhythm information [9]. Another common approach is to extract periodicities based on self-similarity computations. Self-similarity based approaches in general try to detect periodicities by comparing the audio signal to delayed versions of the original signal. This can be done by using a set of comb filters that cover the range of possible tempi. The **output of a comb filter bank** can then be interpreted as a rhythm pattern. The tempo is finally estimated by predicting the tempo corresponding to the comb filter with the highest response. Another variant of the self-similarity approach is the detection of periodicities based on the **Autocorrelation Function (ACF)**. The autocorrelation measures the similarity of a signal to a delayed version of the original signal. The autocorrelation function represents this self-similarity relation for different time lags. For music signals the peaks in the ACF reflect the occurrence of regular musical events. Thus the ACF itself is some sort of rhythm pattern. The ACF can be computed based on the time domain representation or in the frequency domain, on a frame basis. Foote et al. [2] cut the audio signal into frames, perform an FFT for each frame and finally derive the so-called **Beat Spectrum** from the self-similarity matrix of the FFT frame representation. Another interesting self-similarity approach is **detrended fluctuation analysis** (DFA). The DFA can measure two-point correlations and is especially suitable for non-stationary signals like music. In [8] DFA is used to generate a rhythm pattern useful for genre classification. Pampalk et al. [11, 12] measure the fluctuations of the loudness in twenty different frequency bands to capture detailed rhythm information. The **Fluctuation Patterns** are a common descriptor for content-bases audio similarity computations and will be used in our evaluations in section 4.

3 REFORMULATING THE TEMPO ESTIMATION PROBLEM

Within the tempo extraction process the estimation of the tempo based on the extracted rhythm pattern is a crucial step. A perfect rhythm pattern would capture all periodic elements in a piece, and their relative strengths. Estimating the tempo from a rhythm pattern means picking the *"correct"* periodicity. Which periodicity is most significant depends on the human perception and is up to now not yet fully understood. Various strategies have been implemented to pick the correct tempo, but most tempo extraction algorithms just pick the highest peak from the beat pattern.

The basic idea of our approach is based on the assumption that songs having a similar rhythmic structure are likely to have a similar (perceived) tempo as well. Thus if one has got a set of rhythm patterns manually annotated

[1] http://www.music-ir.org/mirex2006

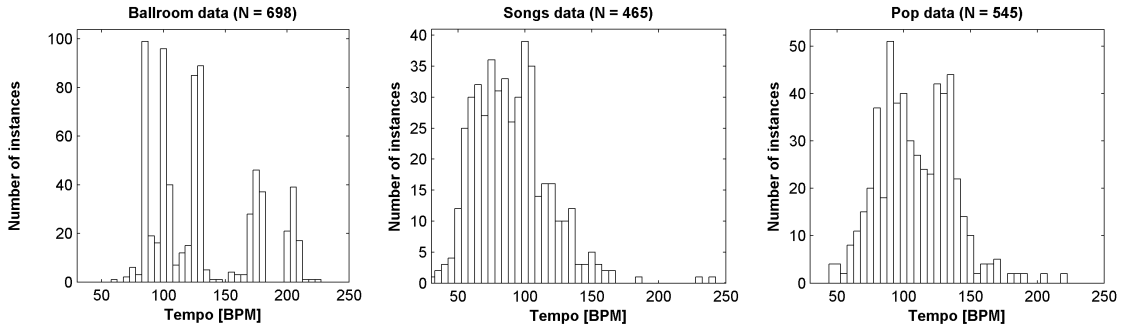

Figure 1. Histograms of ground truth tempo values in 5 bpm steps for the *ballroom*, *songs* and *pop* dataset.

with the correct perceived tempo, the mapping of an unseen rhythm pattern to a perceived tempo can be realized just by looking for similar rhythm patterns in the annotated training set. Using the N most similar rhythm patterns from the annotated training set, we can predict the perceived tempo by predicting the most frequent tempo of these N rhythm patterns (*nearest neighbor classification*). Thus, we learn the mapping from rhythm patterns to perceived tempo from a ground truth dataset using an instance-based machine learning approach.

While this approach is straightforward, we still have to define a distance metric that measures how similar two rhythm patterns are. For our experiments we decided to use the correlation coefficient. For two rhythm patterns $\vec{x}$ and $\vec{y}$ the correlation is given by equation (1).

$$r = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \hat{x})^2 \sum_{i=0}^{n}(y_i - \hat{y})^2}} \quad (1)$$

Two different types of rhythm patterns are used in our experiments, the *Autocorrelation Function (ACF)* and the *Fluctuation Patterns (FPs)*.

- **Autocorrelation Function (ACF)**
  In our experiment, the Autocorrelation Function is computed from a log-magnitude 40-channel Mel-frequency spectrogram for an 8 kHz down-sampled mono version of the original audio signal with a 32 ms window and 4 ms hop between frames, exactly as described in [14]. For each frequency channel the first-order difference is half-wave rectified and finally summed across the frequency bands. After high-pass filtering the resulting onset signal to remove the d.c. offset the signal is autocorrelated out to a maximum lag of 4 seconds. Finally we smoothen the ACF using an average filter with a window size of 20, because the ACF can be extremely *"spiky"* and even pretty similar rhythm patterns could be judged as rather dissimilar by our distance measure. The resulting autocorrelation function is then used as a representation of the periodicities in the original audio signal.

- **Fluctuation Patterns (FP)**
  The Fluctuation Pattern of a songs describes the amplitude modulation of the loudness for 20 frequency bands, spaced according to the Bark scale. The FPs were originally designed for rhythm-based audio similarity computations and were never expected to be useful for automatic tempo extraction. Our implementation is based directly on [11] and [12]. The only modification we apply is to reduce the rhythmic information captured for 20 different frequency bands to just one by simply summing across the bands. This reduces the dimensionality from 1200 to only 60 dimensions, which is expected to be beneficial for the nearest-neighbor classifier to be used in our experiments — nearest-neighbor methods are notorious for their sensitivity to high-dimensional feature spaces.

In the next section we report on our ground truth data and the results obtained from our experiments.

4 EXPERIMENTS

4.1 Ground truth data

Our results are based on three different datasets. The first two datasets have been used in [4] for tempo extraction evaluations and should help make the results of our experiments more comparable to previous work. The *ballroom* dataset is a well known dance music collection from *BallroomDancers.com* and consists of 698 audio excerpts — each of about 30 seconds of playing time. The *songs* dataset, also used in [4], is publicly available [2] . This dataset contains 465 audio clips (20 seconds each) from nine genres (Rock, Classic, Electronica, Latin, Samba, Jazz, AfroBeat, Flamenco, Balkan and Greek).

The third dataset, *pop*, was generated by ourselves, by crawling publicly available beat databases from the web and retrieving the tempo information for thousands of popular songs. The authors' private collections were then searched for songs for which the tempo information was now available, using a approximative string matching algorithm. The resulting dataset was checked manually for plausibility. The final set consists of 545 full length songs belonging to various different popular genres. The distributions of the annotated tempi of the test databases are shown in Figure 1.

[2] http://www.iua.upf.es/mtg/ismir2004/contest/tempoContest/

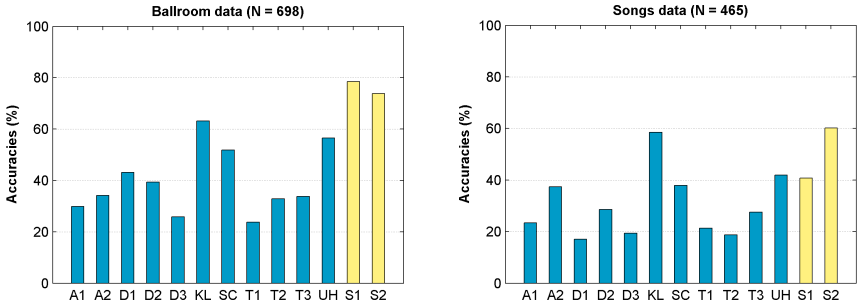

Figure 2. Comparative experimental results on *ballroom* and *songs* datasets.

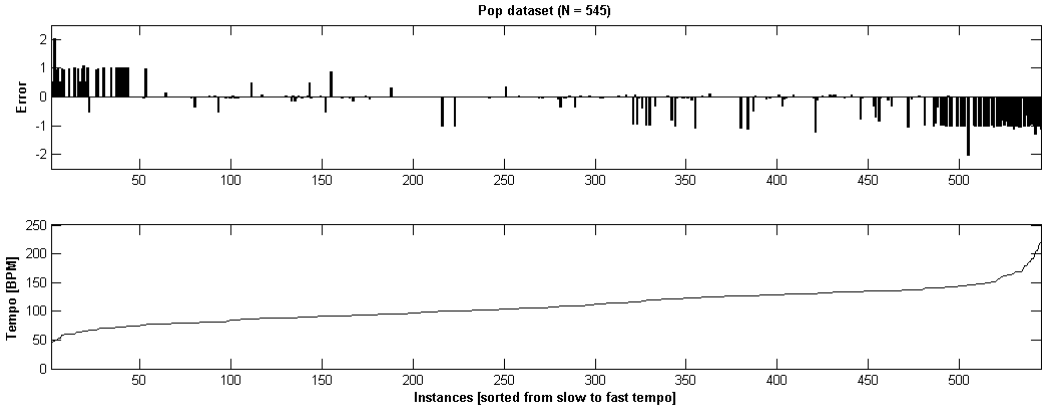

Figure 3. Visualisation of errors of S2 on *pop* dataset, sorted in ascending order according to ground truth tempo. Top panel: prediction errors. Bottom panel: ground truth tempo.

4.2 Learning and evaluation methods

To estimate the tempo for a given song from one of our datasets, we use a simple k-nearest-neighbor (k-NN) classifier, which searches for the k most similar songs in the database and predicts the tempo that appears most frequently within these k songs. In our experiments, we used $k = 5$. Evaluating the method on a whole music collection is done via a strategy known as leave-one-out cross validation: the tempo estimate of one single audio excerpt is computed by using all other examples from the dataset (except the audio excerpt we estimate the tempo of) as training examples. Since there are no duplicates in the datasets, there is no general advantage of this approach compared to the tempo estimation algorithms evaluated in [4].

To make the results comparable to [4], we define the tempo estimation to be correct if the predicted tempo estimate is within 4% (the precision window) of the ground-truth tempo. For visualizing the errors, (see Fig. 3) we introduce a slightly modified error measure compared to [4], which is defined in equation (2), where t denotes the ground truth tempo and $\hat{t}$ the estimated tempo of a piece. Compared to the error measure in [4], which is based on the logarithm, incorrect tempo estimates exceeding half or double the ground truth tempo will be judged in a more linear way.

$$e = \begin{cases} \frac{\hat{t}}{t} - 1 & \hat{t} > t \\ -(\frac{t}{\hat{t}} - 1) & \hat{t} <= t \end{cases} \quad (2)$$

Correct or almost correct estimates will yield an error value close to zero, whereas tempo estimates double or half the ground truth value are considered to be equally erroneous. Positive values indicate that the tempo extraction algorithm predicts too fast a tempo, while negative values indicate that the predicted tempo is too slow. In particular, predicting twice the correct tempo gives an error of 1.0, predicting half the tempo gives -1.0.

4.3 Results

The results obtained from the datasets *ballroom* and *songs* can easily be compared to the results obtained in [4]. The detailed evaluation results for eleven algorithms from six different participants in the ISMIR'04 tempo induction contest, namely *Miguel Alonso* (A1, A2), *Simon Dixon* (D1, D2, D3), *Anssi Klapuri* (KL), *Eric Scheirer* (SC), *George Tzanetakis*(T1, T2, T3) and *Christian Uhle* (UH) can be found on the web [3]. We will denote our own algorithms as S1 (the NN approach using *Fluctuation Patterns*) and S2 (NN prediction based on the *Autocorrelation Function*), respectively. Figure 2 illustrates the results for

[3] http://www.iua.upf.es/mtg/ismir2004/contest/tempoContest

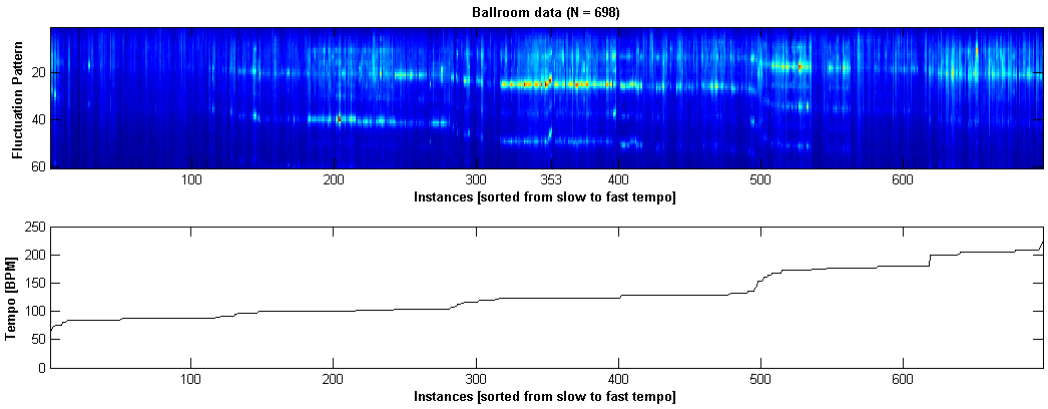

Figure 4. Visualization of the relation of annotated tempo and the *Fluctuation Patterns* for the *ballroom* dataset.

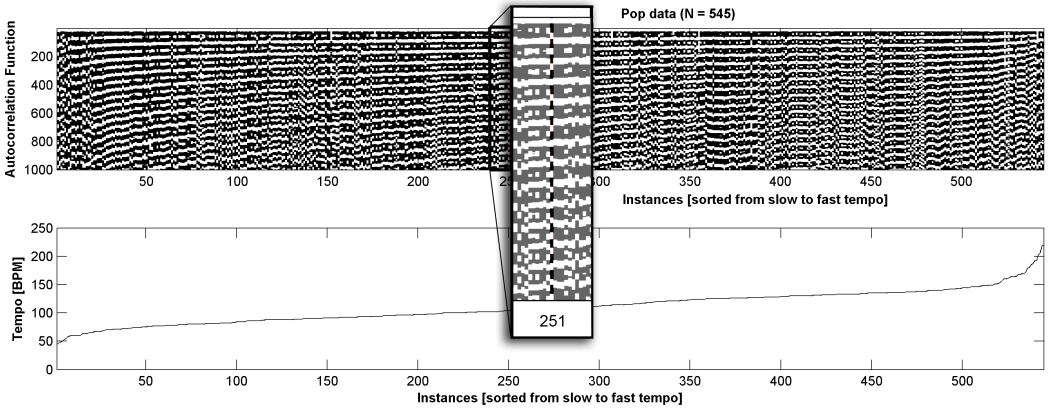

Figure 5. Visualization of the relation of annotated tempo and the *Autocorrelation Function* for the *pop* dataset (ACF data binarized to enhance visibility of patterns).

the *ballroom* dataset and for the *songs* dataset. For the *ballroom* dataset the k-NN approach clearly outperforms the other tempo extraction algorithms. S1 achieves an accuracy of **78.51%** and S2 an accuracy of **73.78%**. For the *songs* dataset we obtain accuracies of **40.86%** for S1 and **60.43%** for S2, which amounts to rank 4 (S1) and rank 1 (S2) in this comparison, respectively. Thus, we may conclude that our learning approach performs roughly at the same level as the best current tempo identification algorithms, at least on these two data sets.

On our own data set *pop* we obtain a classification accuracy of **68.8%** for S1 and **74.5%** for S2. Since no comparision is possible for our own dataset, Figure 3 just illustrates the estimation errors of S2. As can be seen, the errors are almost exclusively due a commitment to the 'wrong' metrical level — the errors are either 1.0 or -1.0. Also, the errors occur mainly in those parts of the music collection where there are extreme tempi and (thus) few similar pieces (i.e., pieces with similar rhythm patterns).

Finally, joining all three datasets into one large set of more diverse styles and running our algorithms on this set gives accuracies of **64.06%** for S1, and **68.91%** for S2, which shows that the good performance on the individual sets is not just due to the narrow stylistic range of the sets.

Our tempo estimation NN-approach is based on the assumption that songs having similar rhythm patterns tend to have the same perceived tempo. To check if this is a reasonable assumption, we can sort all the rhythm patterns according to the annotated ground truth tempo and visualize the result. Figure 4 shows the *Fluctuation Patterns* for all the 698 instances of the *ballroom* dataset. Each column represents the FP of the corresponding audio excerpt. The tempo curve in figure 4 indicates the increasing tempo for each sample from the dataset. One can visually observe that audio clips of similar annotated tempo tend to have similar rhythm patterns. Figure 5 visualizes the *Autocorrelation Function* for the *pop* dataset. The patterns of the ACF seem to change smoothly as the annotated tempo increases. In figure 5, instance 251 is marked. This instance is rather dissimilar compared to its neighbors. A further investigation of the corresponding song *Five - When The Lights Go Out* led to the conclusion that the ground truth annotation of this song (*104 bpm*) is incorrect and should be changed to about *140 bpm*. This case illustrates that one can even visually explore mistakes in the ground truth annotation based on this representation. [4]

[4] The outlier is not well visible in the printed version of Figure 5 – not even in the enlarged excerpt –, but it clearly sticks out in the coloured

5 CONCLUSIONS

The reformulation of the tempo estimation step in terms of a nearest neighbor classification problem permits to learn the relation of rhythm patterns and perceived tempo. Thus, we can make the implicit tempo information from rhythm patterns explicit, such that it can be used in form of a tempo descriptor for MIR applications. We have demonstrated this both for the Autocorrelation Function and for Fluctuation Patterns. Experimental results based on three different datasets indicate that this approach performs at least equally well as other state-of-the-art tempo identification methods.

A visualization of the datasets supports our basic assumption that songs with similar rhythm patterns tend to have the same perceived tempo. The proposed visualization of rhythm patterns ordered according to the associated perceived tempo reveals interesting structures and might be a useful representation to get some more insight in the relation of rhythm patterns and perceived tempo. Also, possible annotation failures in the ground truth data can be visually explored using the proposed representation.

As with each instance based learning approach, the effectiveness of the tempo extraction strongly depends on the training instances, the similarity measure, and the ability of the rhythm patterns to capture periodicity information. To be useful in terms of a general tempo extraction algorithm a large annotated training set is needed that covers various tempo ranges and various genres and musical styles. This may be a limiting factor in practical MIR applications. We do, however, hope that the kind of approach proposed here, if investigated further, could help in getting deeper insights in the relation of rhythm and tempo perception.

6 ACKNOWLEDGMENTS

This research was supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant L112-N04.

7 REFERENCES

[1] Chua, B.Y. Ku, G. "Improved Perceptual Tempo Detection of Music", *Proceedings of the 11th International Multimedia Modelling Conference (MMM'05)*, Melbourne, Australien, 2005.

[2] Foote, J. Uchihashi, S. "The Beat Spectrum: A new Approach to Rhythm Analysis", *Proceedings of the IEEE International Conference on Multimedia Expo (ICME'01)*, Tokyo, Japan, 2001.

[3] Zhu, J. Lu, L. "Perceptual Visualization of A Music Collection", *Proceedings of the IEEE International Conference on Multimedia Expo (ICME'05)*, Amsterdam, The Netherlands, 2005.

computer display.

[4] Gouyon, F. Klapuri, A. Dixon, S. Alonso, M. Tzanetakis, G. Uhle, C. Cano, P. "An experimental comparison of audio tempo induction algorithms", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1832-1844, 2006.

[5] McKinney, M.F. Moelants, D. "Deviations from the resonance theory of tempo induction", *Proceedings of the Conference on Interdisciplinary Musicology (CIM'04)*, Graz, Austria, 2004.

[6] Dixon, S. "Onset Detection Revisited", *Proceedings of the International Conference on Digital Audio Effects (DAFx'06)*, Montreal, Canada, 2006.

[7] Chua, B.Y. Lu, G. "Determination of Perceptual Tempo of Music", *Lecture Notes in Computer Science*, vol. 3310, pp. 61-70, 2005.

[8] Jennings, H.D. Ivanov, P.C. Martins, A.M. Silva, P.C. Viswanathan, G.M. "Variance fluctuations in nonstationary time series: a comparative study of music genres", *Physica A: Statistical and Theoretical Physics*, vol. 336, Issue 3-4, pp. 585-594, 2004.

[9] McKinney, M.F. Moelants, D. "Extracting the Perceptual Tempo From Music", *Proceedings of the International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.

[10] McKinney, M.F. Moelants, D. "Tempo Perception and Musical Content: What makes a piece fast, slow or temporally ambiguous?", *Proceedings of the International Conference on Music Perception and Cognition (ICMPC'04)*, Evanston, USA, 2004.

[11] Pampalk, E. "Islands of Music: Analysis, Organization, and Visualization of Music Archives", MSc Thesis, Technical University of Vienna, 2001.

[12] Pampalk, E. Rauber, A. Merkl, D. "Content-based organization and visualization of music archives", *Proceedings of the 10th ACM International Conference on Multimedia*, Juan les Pins, France, pp. 570-579, 2002.

[13] Gouyon F., Widmer G., Serra X., Flexer A. "Acoustic Cues to Beat Induction: A Machine Learning Perspective", *Music Perception*, vol. 24, Issue 2, pp. 177-188, 2006.

[14] Ellis, D.P.W. "Beat Tracking with Dynamic Programming", *Music Information Retrieval Evaluation eXchange (MIREX'06)*, 2006.