REDUCING TEMPO OCTAVE ERRORS BY PERIODICITY VECTOR CODING AND SVM LEARNING

Aggelos Gkiokas[1,2]
[1]Institute for Language and Speech Processing / R.C. Athena
[2]National Technical University of Athens
agkiokas@ilsp.gr

Vassilis Katsouros
Institute for Language and Speech Processing / R.C. Athena
vsk@ilsp.gr

George Carayannis
National Technical University of Athens
carayan@softlab.ece.ntua.gr

ABSTRACT

In this paper we present a method for learning tempo classes in order to reduce tempo octave errors. There are two main contributions of this paper in the rhythm analysis field. Firstly, a novel technique is proposed to code the rhythm periodicity functions of a music signal. Target tempi range is divided into overlapping “tempo bands” and the periodicity function is filtered by triangular masks aligned to those tempo bands, in order to calculate the respective saliencies, followed by the application of the DCT transform on band strengths.

The second contribution is the adoption of Support Vector Machines to learn broad tempo classes from the coded periodicity vectors. Training instances are assigned a tempo class according to annotated tempo. The classes are assumed to correspond to “music speed”. At classification phase, each target excerpt is assigned a tempo class label by the SVM. Target periodicity vector is masked by the predicted tempo class range, and tempo is estimated by peak picking in the reduced periodicity vector.

The proposed method was evaluated on the benchmark ISMIR 2004 Tempo Induction Evaluation Exchange Dataset for both tempo class and tempo value estimation tasks. Results indicate that the proposed approach provides an efficient framework to tackle the tempo estimation task.

1. INTRODUCTION

Most tempo estimation systems suffer from detecting the correct metrical level, i.e. tend to result in tempi that are fractions or multiples of the groundtruth tempo. Such errors are usually found in the literature as “octave errors”. Although many methods are reported to achieve accuracy over 90% [1-3] when ignoring octave errors, i.e. accuracy for finding the exact, double, treble, half or 1/3 of groundtruth tempo (known as *accuracy2* measure), the accuracy of these methods decreases to 50~60% for finding the exact tempo (*accuracy1*). More details on rhythm analysis systems and evaluation measures can be found in [4,5].

Two certain contemporary aspects arise when considering the octave error problem. First, when allowing an algorithm to make errors that correspond to the different metrical levels, one can say that such an approach is more close to the notion of perceptual tempo. Different users would tap at different metrical levels for the same song. Even a single user might tap at different metrical levels for the same song at different psychosocial states. Thus, it can be claimed that during the evaluation process of a tempo estimation system the usage of a single groundtruth value is not always feasible. On the other hand, not all fractions and multiples can be considered as musically correct.

One solution was the P-score evaluation measure introduced in MIREX 2005 Audio Tempo Extraction Task[1] where each excerpt was annotated with two dominant tempi, and their relative strength. Algorithms should suggest two tempi and the P-score is defined as the mean relative strength of the correct estimated tempi within an 8% tolerance. In this context, deciding the correct metrical level is less crucial.

However, consider the following example. The 4th training instance on McKinley’s dataset excerpt, which exhibits a 6/8 measure and 126 bpm tempo, was annotated by 40 experts. 10 of them tapped at eight note level, while 30 tapped at dotted quarter notes. Thus tempo value 42 bpm can be considered more salient than musical tempo 126 bpm. If these people were asked to characterize this excerpt as “slow” or “fast”, probably they would judge it as slow. Although the reliability of annotations can always be questioned, we can conclude that there is a strong relation of the notion of musical “speed” to the perceptual tempo (or metrical level).

Choosing the correct metrical level usually relies on incorporating some prior knowledge, mostly in terms of calculating prior tempi distribution [2]. Other methods adopt metrical models [1,3], inference from inter-onset intervals [6] or by considering the most predominant peak in periodicity vector as the correct tempo [7]. Seyerlehner et al. [8] incorporate instance based learning techniques,

[1] http://www.music-ir.org/mirex/wiki/2005:Audio_Tempo_Extraction

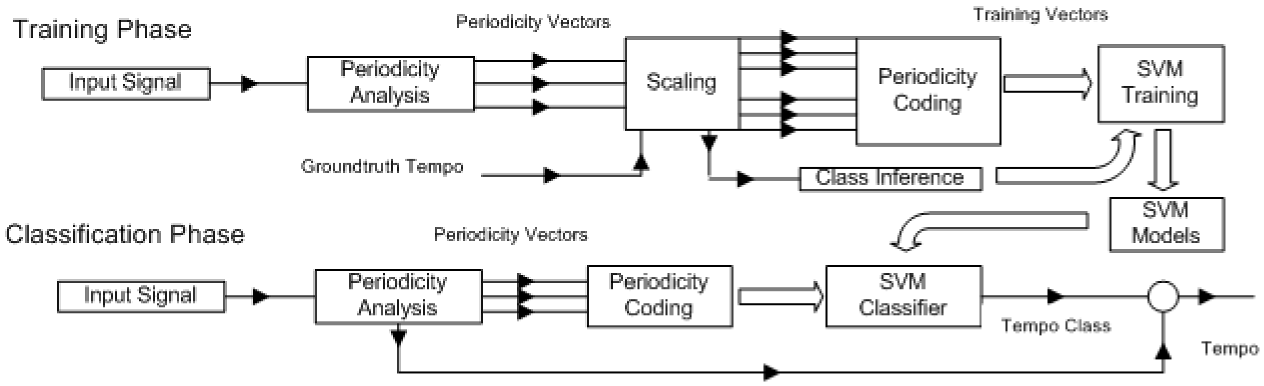

Figure 1. Overview of the proposed method.

where the periodicity vector of the target music piece is compared to periodicity functions of other annotated excerpts. The assigned tempo is equal to tempo of the excerpt with the most similar periodicity vector. In a similar manner, Peeters adopts spectral templates and a learning schema for estimating tempo [9].

Two recent approaches on characterizing the music speed are remarkable. Eronen and Klapuri [10] presented a tempo estimation system, where the predicted tempo is chosen by comparing scaled versions of the periodicity vector of the target excerpt with periodicity vectors of tempo annotated pieces. In the same paper, results were reported for a classification subsystem that classified music excerpts to three categories: slow, medium and fast. In [11] Hockman and Fujinaga proposed a system that classifies music pieces to fast/slow. Annotations were not extracted with the knowledge of any groundtruth tempo but directly from user tags on YouTube videos. Without any rhythmic analysis, but based solely on baseline frame-level features, their method achieved a classification accuracy of 96% by adopting the AdaBoost classifier. In [12] Smith proposed a system for identifying octave errors made by a baseline beat tracker.

In this paper, we present a method of learning tempo octaves, i.e. classifying a music excerpt to one of the three categories: slow, moderate and fast. The proposed method exhibits two key features. Firstly, a coded representation of periodicity vector similar to the popular MFCC features is proposed. Secondly, we adopt an SVM learner to learn tempo octaves. SVM's has been greatly used in classification tasks in the MIR domain such as [13, 14]. We applied the proposed octave learning method to a baseline tempo estimation method [3] in order to limit the target tempi space and enhance tempo extraction accuracy. Evaluation results indicate that the proposed technique enhances greatly the tempo estimation accuracy.

The rest of the paper is organized as follows. In Section 2 an overview of the proposed method is described. Section 3 is dedicated to present the periodicity function extraction procedure. SVM learning formulation is described in Section 4, while in Section 5 the tempo estimation method is presented. Evaluation results and discussion on the proposed method conclude this paper in Sections 6 and 7 respectively.

2. SYSTEM OVERVIEW

Figure 1 shows an overview of the proposed system. In training phase, periodicity analysis of the input signal is performed. A set of vectors that is supposed to contain all rhythmic information of the signal is extracted. Next, the extracted periodicity vectors are rescaled in order to produce more training instances. The periodicity vectors are then coded to a more compact representation, and along with the respective tempo class (slow, moderate, fast) which is inferred from the groundtruth tempo, are used to train the SVM model.

In classification phase, the unknown input signal is processed by the periodicity analysis module. Periodicity vectors are coded as above and feed the SVM classifier. The output class is then combined with the periodicity vectors of the input signal to find the tempo value that is consistent to the metrical level of the SVM classifier.

3. REPRESENTING RHYTHMIC CONTENT

3.1 Periodicity Analysis

Periodicity analysis is performed by the adoption of the method presented in [3]. The constant Q transform is applied to the signal, and followed by the harmonic/percussive separation algorithm reported in [15]. Two feature multidimensional sequences are extracted by the harmonic/percussive parts of the signal respectively. Eight band energies from the percussive part, denoted as $x^i, i = 1..8$ and chroma vectors from the harmonic part denoted as $ch^j, j = 1..12$. Feature sequences are convolved with a bank of resonators with oscillation frequencies set to the tempo analysis range. Resonators' outputs are segmented by square windows and the maximum values of resonators' outputs are considered as the salient values of each feature sequence to each tempo value. We denote as

$p_n^{feature}[t]$ the periodicity vectors for the input signal where $feature \in \{ch^j \cup x^i, j = 1..12, i = 1..8\}$ denotes the feature type, n denotes the time index and $t = \{30..500\}$ denotes the tempo analysis range. Reader should note that this range is larger than target tempi search space. This is due to the fact that periodicity functions contain rhythmic information in frequency regions beyond the groundtruth tempo.

3.2 Scaling Training Vectors

Since there is lack of large amount of annotated tempo data, we could produce artificial data by rescaling a music signal to faster and slower tempi. However, this approach would be computational intensive. To overcome this problem we exploit the following property of the periodicity vector, i.e., tempo-scaled versions of a signal, say by a value of *α,* produce inversely scaled versions of the periodicity vector by the value of 1/*α*. Thus, for a music signal $y[i]$, with periodicity function $p_n^{feature}[t]$

$$y[\alpha i] \xrightarrow{\text{periodicity analysis}} p_n^{feature}[t/\alpha]. \qquad (1)$$

This property allows us to rescale directly the periodicity vectors, instead of the whole signal, reducing thus the complexity of the calculations.

In the same manner as in [10], all periodicity vectors extracted from each music excerpt are rescaled within a range of values for *α* around unity. If a music signal $y[i]$ is assigned a ground-truth tempo T_{ground}, then all α scaled versions of the periodicity vectors $p_n^{feature}[t]$ that result from $y[i]$ are assigned a tempo value $\alpha^{-1}T_{ground}$.

Under the assumption of almost constant tempo, periodicity functions are averaged for each feature across all segments *n* in order to capture better the overall rhythmic content of the signal as follows

$$\tilde{p}^{feature}[t] = \frac{1}{N}\sum_{n=1}^{N} p_n^{feature}[t] \qquad (2)$$

3.3 Periodicity Coding

To satisfy the necessity to capture broad classes of tempo, it seems that it would be more efficient to use a more compact representation for the periodicity vectors. We shall exploit the fact that periodicity vectors of similar tempo music pieces, will not exhibit the same peaks, but may have a similar shape, or they will exhibit peaks in nearby tempi.

Some recent works deal with the spectral modeling of rhythmic information. Holzapfel and Stylianou [16] applied the scale transform to the autocorrelation function of music signals to form a rhythmic representation and exploit aspects of rhythmic similarity. Peeters [17] combines rhythm descriptors in a rhythm classification system while in [9] the DFT of the accent function is subsampled at frequency bins that correspond to tempo harmonic series of certain meters.

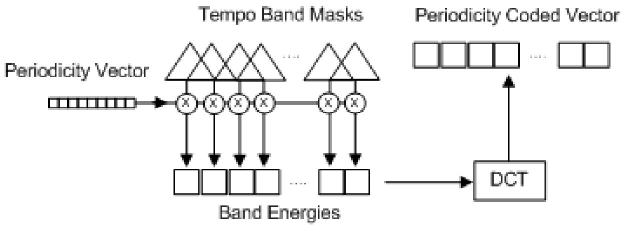

Figure 2. Periodicity vector coding process. ⊗ stands for inner product.

In this paper, we introduce a filterbank-like analysis on the periodicity vectors, which is illustrated in Figure 2. The range of target tempi is divided into *K* equally tempo intervals with a 50% overlap between successive intervals. From each tempo interval, we utilize a symmetric triangular weighting mask.

The strength $s^{feature}[k]$ of the periodicity vector $\tilde{p}^{feature}[\cdot]$ for each of the K tempo intervals is calculated as the inner product with the respective mask:

$$s^{feature}[k] = \sum_{t=T_{\min}}^{T_{\max}} \tilde{p}^{feature}[t] \cdot mask^k[t] \qquad (3)$$

where $mask^k[\cdot]$ denotes the mask of *k* tempo band.

Henceforth two problems arise from this modeling. Firstly, there is a strong correlation between features, caused mainly by the overlap of adjacent tempo bands. Secondly, different feature type sequences for the same piece will result to different periodicity vectors. For example energy evolution of lower spectral bands exhibit higher values in lower tempi whereas higher spectral bands exhibit faster changes, and thus higher tempi. Therefore periodicity vectors calculated from different features cannot be compared directly and cannot be treated in the same manner. To suppress the effects of band correlation we apply the Discrete Cosine Transform to each tempo-band strength vector $s^{feature}[\cdot]$, in order to obtain the uncorrelated coefficients $m^{feature}[\cdot]$:

$$m^{feature}[l] = \mathrm{DCT}(s^{feature}[\cdot]) \qquad (4)$$

To cope with the different feature behaviour, the periodicity representation is finally formed by appending all coefficients $m^{feature}[\cdot]$ for each segment *n* to a single 20*K*-dimensional vector $\mathbf{m}$:

$$\mathbf{m} = [\mathbf{m}^{x^1} \,|\, \mathbf{m}^{x^2} \,|...|\, \mathbf{m}^{x^8} \,|\, \mathbf{m}^{ch^1} \,|\, \mathbf{m}^{ch^1} \,|..|\, \mathbf{m}^{ch^{12}}] \quad (5)$$

4. LEARNING TEMPO CLASSES

Let $\{(\mathbf{m}_l, t_l), l \in L\}$ denote the vectors extracted from the music signals using the method described in Section 3, where t_l are the annotated tempi. Depending on the value of t_l we assign excerpts to one of the following classes:

$$c_l = c(t_l) = \begin{cases} 1, & T_{slow} \geq t_l \\ 2, & T_{slow} < t_l < T_{fast} \\ 3, & t_l \geq T_{fast} \end{cases} \quad (6)$$

The thresholds T_{slow} and T_{fast} can either be user specified or inferred by data and they divide the target tempi range into the music speed classes of slow, moderate and fast.

We formulate two SVM problems for inferring the tempo classes; a **classification SVM** where we learn each class from the training data $\{(\mathbf{m}_l, c_l), l \in L\}$ and a **regression SVM** where we estimate a target tempo function from the training data $\{(\mathbf{m}_l, t_l), l \in L\}$ Then excerpts are classified one of the three classes by applying Eq. 6 on the estimated tempo value.

The conceptual difference between the two formulations is that while in classification we learn a function from feature space $\mathbb{R}^{20K}$ to {slow, moderate, fast}, i.e. discretization takes place directly on the training data (Eq. 6), in the case of regression discretization is applied to the regression estimate of the target tempo $\hat{t}_i$.

For the classification SVM the multiclass problem is split up to binary classification problems by applying the "one-vs-one" strategy. There is evidence [18] that the one-vs-one strategy is more suitable than the more common "one-vs-all" strategy, especially when there are imbalances between train classes, which is the case of the evaluation datasets (see Sec. 6).

In the case of the regression SVM the continuous tempo estimate $\hat{t}_i$ cannot be directly interpreted as an accurate tempo value, since in the signal representation $\mathbf{m}_i$ much of the rhythmic information such as the peaks in the periodicity vectors are suppressed by the periodicity coding process. However, the value $\hat{t}_i$ would give a rough estimate of the tempo that will be used in Eq. 6 to infer the tempo class of the excerpt.

5. ESTIMATING TEMPO

To extract the final tempo estimate from the periodicity function and the tempo class assigned by the SVM, we calculate an overall periodicity function by the superposition of the individual periodicity functions. In particular, the periodicity vectors are summed across the two feature types and the resulting vectors are multiplied to give the decision periodicity vector:

$$p[t] = \left(\sum_{i=1}^{8} p^{x_i}[t] \right) \left(\sum_{j=1}^{12} p^{ch_j}[t] \right) \quad (7)$$

Accordingly to the estimated class, $p[t]$ is reduced to the corresponding tempi range prescribed by Eq. 6. Final tempo estimate is decided as the most predominant peak in the reduced periodicity vector.

6. EVALUATION

The proposed method was evaluated on the ISMIR 2004 Tempo Induction Evaluation Exchange Dataset: *ballroom* and *songs* datasets [4]. Periodicity vectors were rescaled in the range [0.8, 1.2] with a 0.02 step. We divided the target tempi to classes by setting $T_{slow} = 80$ bpm and $T_{fast} = 130$ bpm in Eq. (6). Tempo bands number was set to K=20, tempo analysis region was set to [30..500] and target tempi space to [30..300]. Feature vector values where normalized to [-1, 1]. We adopted the LIBSVM implementation of SVM [19]. We used an RBF kernel for the SVM and parameter γ of the kernel was set to 1/20K. For regression, we adopted the ε-support vector regression method. Experiments were run for various values of the parameter C (Eqs. 1 and 9 in [19]). Variations of ε (Eq. 9 in [19]) did not affect significantly the overall performance, and was set to 0.1.

To measure the generalization ability of the proposed method we adopted a three fold cross validation approach. Each evaluation set was split randomly to three equal subsets. Each subset was used as a test set and the remaining two as the training set. Evaluation measures were averaged on every train-test sets combination.

6.1 Assignment to Tempo Classes

The first series of experiments involves the classification accuracy to tempo classes. Figure 3 (top) illustrates the accuracy for various values of the parameter C on ballroom/songs datasets respectively, for both methods (classification / regression). The accuracy is almost constant for a wide range of C values, say for $10<C<500$. It must be noted that SVM classification approach outperforms the regression formulation. It is clear that although the tempo discretization process from the assignment of the music excerpt to one of the three classes introduces ambiguities for ground-truth tempi that are closer to either T_{slow} or T_{fast}, the classification approach is more efficient than the continuous regression approach. This can be explained by the fact that learning a continuous function on a high dimensionality space is much more demanding than separating instances into classes. In addition, the small number of training instances is probably not sufficient to learn such a function. However, there is no evidence that for larger scale experiments classification strategy will be more effective than the regression formulation.

To get a better insight to classification errors Table 1 presents the confusion matrix between classes for the classification approach (C=100), for both datasets. In the *ballroom* dataset classification fails for slow excerpts,

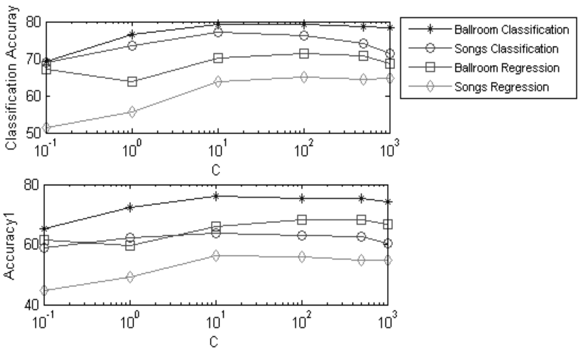

Figure 3. Top: Tempo class classification accuracy for both datasets/classifiers. Bottom: *Accuracy1* for both datasets/classifiers.

	Ballroom			Songs		
	Slow	Mod	Fast	Slow	Mod	Fast
Slow	**22**	33	44	**79**	16	5
Mod	2	**86**	12	17	**82**	1
Fast	1	32	**67**	46	28	**26**

Table 1. Confusion matrix in tempo category classification percentages for both datasets. Rows correspond to ground-truth and columns to estimates**.**

since most of them are classified as fast. This is due to the fact that there are very few excerpts with slow tempi. Thus, during training phase SVM fails to find reliable boundaries for this class. The same effect takes place in the case of fast excerpts in the *songs* dataset.

Figure 4 illustrates the tempo class error with respect to ground-truth tempo, along with dataset tempo distribution for both datasets. As expected, there are more classification errors near the tempo boundaries T_{slow} and T_{fast} with respect to the total test instances with similar tempo. Finding the optimal values for T_{slow} and T_{fast} is dataset depended and is out of the scope of this paper. T_{slow}, T_{fast} were chosen arbitrarily based on authors intuition and not on tempi distribution across data. For example, choosing $T_{slow} = 110$ and $T_{fast} = 150$ for *ballroom* dataset would give more separable classes (see Fig. 4). However, the errors ought to the quantization of tempi values demonstrate an inherent limitation of the proposed method.

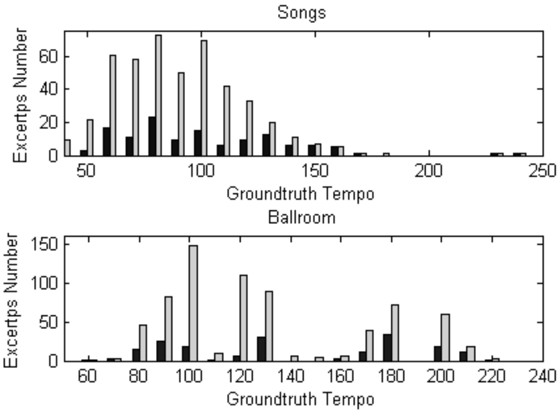

Figure 4. Distribution of classification errors (dark bars) with respect to ground-truth tempo compared to the overall dataset tempo distribution (light bars).

	Ballroom	**Songs**
Our Method	**75.93**	**63.87**
Baseline	59.89	58.49
SE1 [6]	78.51	40.86
SE2 [6]	73.78	60.43
Peeters [9]	75.2	-
Peeters [1]	65.2	49.5
Klapuri [2,4]	63.18	58.49
Uhle [4]	56.45	41.94
Scheirer [4]	51.86	37.85

Table 2. *Accuracy1* of the proposed method compared to best performing methods reported on ballroom/songs datasets.

Figure 3 (bottom) illustrates the *accuracy1* measure of tempo estimation for both classifiers (classification, regression) for various values of *C*. As expected by the results of previous section, the classification approach performs significantly better than regression. Comparing figures in Fig. 3, we can see that tempo class and tempo values estimations are very similar for the *ballroom* dataset: *accuracy1* is about 4 percent below tempo class accuracy. However, this is not the case for *songs* dataset, where *accuracy1* is significantly lower (>10%) than classification accuracy. To verify this, we estimated tempo for both datasets by providing the correct tempo class. Accuracies reported are 88% and 76% for *ballroom*/*songs* datasets respectively. Thus, for the *songs* dataset, even with prior knowledge of the tempo class, periodicity analysis and peak-picking are not always adequate.

Table 2 shows the performance of the proposed method compared to the baseline method adopted and the best performing algorithms reported in the literature for both datasets. It is evident that the proposed method outperforms all other methods. It should be mentioned that although Seyerlehner's SE1 [8] performs better in *ballroom* dataset, results are not directly comparable because they adopt a leave-one-out cross validation. Moreover SE1 reports very low accuracy for *songs* dataset. The significant performance increase of our method is somewhat expected, since it incorporates prior knowledge of the datasets. Although the cross-fold validation strategy splits data to independent subsets, there is still some prior information propagated to test sets caused by the uniformity of the datasets, i.e. most artists/styles are always present in both train/test sets. However the proposed method offers a promising approach to handle large datasets.

7. DISCUSSION AND FUTURE WORK

We presented a method for learning tempo classes with Support Vector Machines. Tempo class classification accuracies of 75% were achieved for both datasets, while most errors were made for excerpts close to class boundaries. The limitation of the target tempi decision space accordingly to the tempo class found for a given excerpt, reduced octave errors made by a baseline tempo estimation system significantly. Estimation accuracies where increased by a margin of 16% and 5% for *ballroom*/*songs* datasets respectively.

It must be noted that classification errors are propagated to the final tempo decision, especially for excerpts that have tempo close to the tempo class decision boundaries. A softer classification decision may be more sensible, as for example providing a confidence measure instead of a hard decision. Moreover, a different treatment of the periodicity function such as analyzing metrical levels considering knowledge of music speed may be proved more efficient. These two main aspects of the proposed method would be investigated in future research.

8. REFERENCES

[1] Peeters G., "Template-based estimation of time-varying tempo," in *EURASIP Journal on Applied Signal Processing,* Volume 2007 Issue 1, 2007 .

[2] Klapuri A., Eronen A. and Astola J., "Analysis of the Meter of Music Acoustic Signals," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14(1), 2006.

[3] Gkiokas A., Katsouros V., Carayannis G. and Stafylakis T., "Music Tempo Estimation and Beat Tracking by Applying Source Separation and Metrical Relations," in *Proc. of the 37th IEEE ICASSP*, Kyoto, Japan, March 25-30, 2012.

[4] Gouyon F., Klapuri A., Dixon S., Alonso M., Tzanetakis G., Uhle C., and Cano P., "An Experimental Comparison of Audio Tempo Induction Algorithms," in *IEEE Transactions on Audio, Speech, and Language Processing,* Vol. 14(5) , September 2006.

[5] Gouyon F. and Dixon S., "A Review of Automatic Rhythm Description Systems," *Computer Music Journal, 29:1, pp 34-54,* Spring 2005.

[6] Dixon S., "Automatic Extraction of Tempo and Beat from Expressive Performances," *J. New Music Research*, 30(1):39–58, 2001.

[7] Alonso M., Richard G. and David B., "Accurate Tempo Estimation Based on Harmonic + Noise Decomposition," *EURASIP Journal on Applied Signal Processing,* Volume 2007, Issue 1, January 2007

[8] Seyerlehner K., Widmer G., and Schnitzer D., "From Rhythm Patterns to Perceived Tempo," in *Proc. of ISMIR,* Vienna, Austria, 2007.

[9] Peeters G., "Template-Based Estimation of Tempo: Using Unsupervised or Supervised Learning to Create Better Spectral Templates," in *Proc. of DAFx-10*, Graz, Austria, 2010.

[10] Eronen A. and Klapuri A., "Music Tempo Estimation with k-NN Regression," in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 18, No. 1, January 2010.

[11] Hockman, J.A. and I. Fujinaga. "Fast vs slow: Learning tempo octaves from user data," in *Proc. of ISMIR*, Utrecht, Netherlands, 2010.

[12] Smith L.M., "Beat-Critic: Beat-Tracking Octave Error Identification by Metrical Profile Analysis," in *Proc. of ISMIR*, Utrecht, Netherlands, 2010.

[13] Mandel M.I., Poliner G.E. and Ellis D.P.W., "Support vector machine active learning for music retrieval," *Multimedia systems*, 12(1):1-11, August 2006.

[14] Wack N., Laurier C., Meyers O., Marxer R., Bogdanov D., Serrà J., Gómez E. & Herrera P., "Music Classification Using High-Level Models," in *Proc. of ISMIR,* Kobe, Japan, 2009.

[15] FitzGerald D., "Harmonic/Percussive Separation Using Median Filtering," in *Proc. of DAFx-10*, Graz, Austria, 2010.

[16] Holzapfel A. and Stylianou Y., "Scale transform in rhythmic similarity of music," in *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19(1), 2011.

[17] Peeters G., "Spectral and Temporal Periodicity Representations of Rhythm for the Automatic Classification of Music Audio Signal," in *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19 (5), 2011

[18] Hsu C.W. and Lin C.J., "A comparison of methods for multiclass support vector machines," in *IEEE Transactions on Neural Networks,* Vol. 13(2), March 2002.

[19] Chang C.-C. and Lin C.-J., "LIBSVM : a Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011.