Audio Engineering Society
Convention Paper

Presented at the 126th Convention
2009 May 7–10 Munich, Germany

Improving Perceived Tempo Estimation by Statistical Modeling of Higher-Level Musical Descriptors

Ching-Wei Chen, Markus Cremer, Kyogu Lee, Peter DiMaria, and Ho-Hsiang Wu

Gracenote, Inc, Emeryville, CA, 94608, USA
{cwchen, mcremer, klee, pdimaria, hhwu}@gracenote.com

ABSTRACT

Conventional tempo estimation algorithms generally work by detecting significant audio events and finding periodicities of repetitive patterns in an audio signal. However, human perception of tempo is subjective, and relies on a far richer set of information, causing many tempo estimation algorithms to suffer from octave errors, or "double/half-time" confusion. In this paper, we propose a system that uses higher-level musical descriptors such as mood to train a statistical model of perceived tempo classes, which can then used to correct the estimate from a conventional tempo estimation algorithm. Our experimental results show reliable classification of perceived tempo class, as well as a significant reduction of octave errors when applied to an array of available tempo estimation algorithms.

1. INTRODUCTION

Humans use a variety of criteria and methods to describe and categorize music, one of the most basic and intuitive of which is tempo. Most people regardless of musical knowledge can make a distinction between a slow song and a fast song. However, the word "tempo" has a broad definition: in some contexts it is used as a count of beats per minute; in other contexts tempos are descriptive words that connote not only the speed of a musical performance, but to some degree the feeling or mood to be expressed. For example, in classical music the tempo mark "Allegro" means "quick and lively", with its literal translation from the Italian being "cheerful". An everyday listener may have yet another definition of tempo, where a "slow" song is one that is quiet and suitable for listening to at night before going to sleep, while a "fast" song is one that is lively and has a dance rhythm to it, regardless of the actual number of beats per minute.

Many algorithms exist for estimating tempo automatically from audio signals. At their most basic level, these algorithms generally analyze low-level audio characteristics and look for repetitive patterns of audio events in order to estimate tempo. However, the factors that influence a human listener to classify a song as "slow" or "fast" are still not very well understood,

though they are almost certainly based on a much wider range of characteristics than the location and repetition rates of audio events or beats.

High-level music classification such as automatic mood extraction attempts to learn many of the affective-domain factors in human understanding, appreciation, and categorization of music. It is the opinion of the authors that many of the factors that influence human perception of musical mood are closely related to the factors which influence human perception of tempo. It is for this reason that we propose to investigate this relationship, by building a statistical model of perceptual tempo classes using mood descriptors. This knowledge may then be applied to improve the abilities of existing tempo estimation algorithms to correctly detect the most perceptually salient tempo for a human listener.

2. RELATED WORK

2.1. Tempo Estimation

Much attention has been paid in the past to automated tempo estimation. Most published tempo estimation algorithms (see [1], [2] for a survey of some of these) utilize some combination of and variation on onset/event detection in the time or sub-band domain, and self-similarity or autocorrelation to detect repetitive patterns and estimate the tempo, in beats-per-minute, present in an audio signal.

In 2004, a large-scale evaluation of several tempo estimation algorithms was conducted at the International Conference on Music Information Retrieval (ISMIR), comparing the estimated tempo, in beats-per-minute (BPM), of 11 submitted algorithms against an expert-annotated ground truth tempo dataset of more than 3000 items. The dataset was organized in 3 subsets, *Ballroom*, *Loops*, and *Songs*, and all audio samples, ground truth annotations, and results of evaluated algorithms are publicly available[1]. The *Songs* dataset contains 465 songs roughly drawn from the general popular and contemporary music genres, and is of most interest to the authors.

One characteristic of this evaluation, which it shares with many other tempo evaluation methods, is that the ground truth tempo is given as a single BPM value. This value is usually derived from the "foot-tapping rate" of a human annotator, and is intended to represent the most salient tempo within the music, as perceived by a human listener. Accordingly, the output of each tempo estimation algorithm was a single BPM estimate. However, most popular music is polyphonic and polyrhythmic, with different instruments playing rhythms at periodicities which are integrally related multiples of each other. This leads to an ambiguity about which tempo or meter is the most salient, or representative of the overall "feel" or "speed" of the music. In particular, this can result in ambiguity between integrally related BPMs, also known as "double-/half-time confusion", or octave errors.

Algorithm	Exact	Half	Double	Other
A1	23%	3%	31%	43%
A2	37%	6%	24%	33%
D1	29%	8%	23%	40%
D2	19%	0%	46%	34%
D3	17%	0%	58%	25%
KL	58%	2%	30%	10%
SC	38%	6%	25%	32%
T1	21%	8%	12%	59%
T2	19%	3%	17%	62%
T3	28%	4%	20%	49%
UH	42%	3%	25%	30%

Table 1 – Results from ISMIR 2004 tempo induction evaluation on *Songs* dataset, showing occurrences of octave and other errors

Table 1 shows the results of the ISMIR 2004 tempo induction evaluation for the 11 algorithms submitted by M. Alonso (A1, A2), S. Dixon (D1, D2, D3), A. Klapuri (KL), E. Scheirer (SC), G. Tzanetakis (T1, T2, T3), and C. Uhle (UH). The accuracy rates, measured as the proportion of estimated tempos that are an exact match, within a 4% tolerance window, to the ground truth tempo are relatively low, with most algorithms scoring less than 30%, and the best algorithm scoring 58%.

However, as shown by the "Half" and "Double" columns in Table 1, a significant proportion of the errors are octave errors, where the predicted BPM is either half or twice the actual ground truth BPM. The "Other" column includes one-third and triple-time errors, though because they are not being considered at this time, are grouped together with all other errors whether an integrally related multiple or not. The

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

prevalence of octave errors encouraged the ISMIR evaluators to include a second accuracy metric which considers BPM estimates that are half, double, one-third, or three times the ground truth BPM as correct. However, the authors believe that getting the BPM estimate exactly correct is still an important goal, and the proposed method will attempt to increase the exact accuracy rate by correcting for these octave errors.

2.2. Perceptual Tempo

More recently, the concept of perceptual tempo, as distinct from the more technical notated tempo in music composition and performance, has garnered more attention [2][3]. In the Music Information Retrieval Evaluation eXchange (MIREX) evaluation of tempo extraction [4] in 2005, the ground truth annotations were updated to include the 2 most salient perceptual BPMs, along with a weighting factor indicating the relative perceptual significance of one of the BPMs.

While the knowledge of the 2 most salient BPMs and their relative weighting may be useful for certain applications, many applications can only make use of a single BPM. For example in an automatic playlist generator application, a user who wants to relax may request a list of songs labeled as "slow". If the listener is presented with an upbeat dance song, he will strongly object. It does not matter if the song actually contains some rhythmic elements at the slower tempo; the listener is only interested in the single descriptor of the perceptual tempo of the song being correct. For this reason, we will evaluate our system based solely on a single BPM estimate.

The previous example also illustrates how human perception of tempo may be influenced by more factors than simply periodicities of repeating audio events. Recent work demonstrates the relationship between perceived tempo and rhythmic pattern [5] and timbre [6]. However, these are only some of many audio features that may play a part in the human perception of tempo. Our proposed system will demonstrate the relationship between perceived tempo and high-level Mood descriptors.

2.3. Mood Classification

Musical mood ontology is very close to the affective layer of human perception and interpretation of music, and thus can be considered as being on one of the higher levels of automated analysis methods. A mood descriptor carries with it a large amount of perceptual information, some of which may relate to the perceived tempo or speed of music. For example, a mood descriptor such as "Aggressive" or "Frantic" may connote, among other things, a tendency for a human listener to categorize the song as "fast", while a mood descriptor such as "Romantic" or "Sentimental" may connote a tendency for the song to be categorized as "slow". Of course, some mood descriptors may have less of a correlation with perceived tempo. For example, mood labels such as "Optimistic" or "Serious" may not connote any tempo range in particular.

Extracting these attributes using machine learning algorithms is not a farfetched idea. In fact, Katayose et al. considered this more than 20 years ago [7]. Current content-based musical mood detection algorithms combine a sizable set of low- and mid-level audio features (such as Mel-frequency cepstral coefficients, spectral flatness, chord class, percussiveness, and others) to classify an unknown signal into higher-level musical descriptors like mood [8]. These higher-level mood classes may be thought of as a perceptually meaningful summarization of a large number of underlying low-level audio features. As signal-based mood classification systems use a manifold of low-level attributes to determine the respective classes for an individual musical piece, they have access to information that is not intuitively associated with the tempo of a song, but can impact the judgment of human listeners when asked if a song is slow or moderately fast, as illustrated before. In this paper we will investigate how to leverage this information in order to get to a more accurate perceived tempo estimate for musical items.

3. PROPOSED SYSTEM

The proposed method is built around a statistical model of four perceived tempo classes derived from high-level musical Mood descriptors. In parallel, a baseline BPM estimate may be derived through the use of traditional tempo estimation techniques. After using the statistical model to classify an audio signal into one of the four perceived tempo classes, a set of heuristic rules is used to modify the baseline BPM. This modification is intended to correct the incidences of octave errors in the baseline tempo estimate, and hopefully achieve a more accurate estimate of perceived tempo.

The estimated tempo class may also be used on its own, in applications where a relative tempo range is more important than an exact BPM tempo estimate.

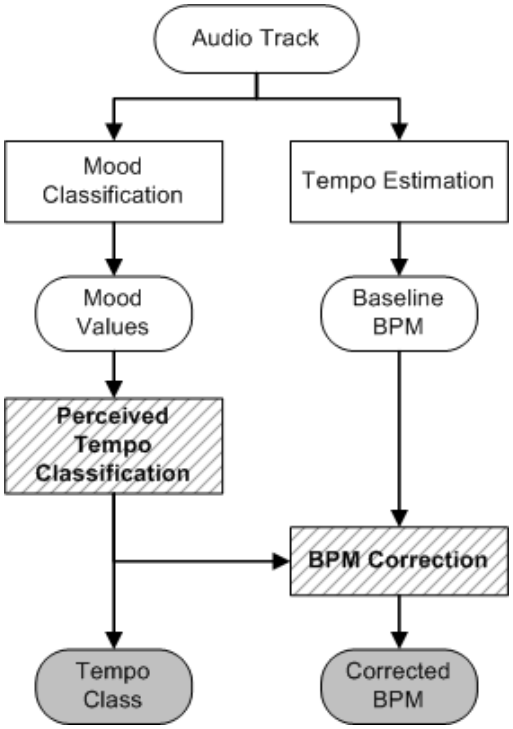

Figure 1 – Proposed System Architecture.

3.1. Mood Classification

An expert-trained Gaussian Mixture Model (GMM)-based classifier was used to assign scores in 101 different Mood categories for each of 299 songs in the dataset. The Mood classifier uses a collection of roughly 90 low- and mid-level audio features, which are not of interest here. In theory, the proposed method may work with descriptors from any reasonably designed Mood classification system, and possibly with editorially annotated Mood descriptors as well.

3.2. Tempo Estimation

Similar to the Mood Classification step, the Tempo Estimation stage is not the main focus of this paper. The proposed method of tempo classification and correction is intended to be useful for improving the accuracy of any given tempo estimation algorithm. To that end, a variety of different tempo estimation algorithms, both published and commercially available, were used in the evaluation of the system. This is described in more detail in Section 4.2.

3.3. Perceived Tempo Class Dataset

A ground truth dataset of perceived tempo classes was generated by collecting human annotations for 299 randomly selected songs selected from within the general contemporary and popular music genres. Five annotators were instructed to listen to these 299 unlabeled audio files, and to divide the dataset into four groups representing broad but distinct perceptual tempo classes:

1 - *Very Slow*
2 - *Somewhat Slow*
3 - *Somewhat Fast*
4 - *Very Fast*

The first and fourth classes ("Very Slow" and "Very Fast") represent songs whose tempos are clearly and unambiguously slow or fast, while the second and third classes ("Somewhat Slow" and "Somewhat Fast") represent songs with a somewhat ambiguous tempo, with a slight tendency towards being moderately slow or moderately fast.

Tempo class	Number of tracks	Variance
1	33	0.158
2	111	0.289
3	125	0.249
4	30	0.210

Table 2 – Distribution of tempo class annotations

The perceived tempo class scores were averaged across the 5 annotators, and rounded to the nearest integer to give the aggregate perceived tempo class. Table 2 shows the distribution of the aggregate tempo class labels for this dataset. Of note is the variance of the perceived tempo classes, which illustrates the degree of disagreement among the 5 annotators. As might be expected, the variances for tempo classes 1 and 4 are lower than those for tempo classes 2 and 3, since extreme tempos tend to be less ambiguous. However, the fact that the average variance of all 4 tempo classes is close to a quarter of a full tempo class is a reminder that even labeling songs into coarse tempo classes is a subjective task.

3.4. Perceived Tempo Classification

Using the 101-dimensional Mood descriptor extracted from Section 3.1 as a feature vector, a support vector machine (SVM)-based classifier was trained using the aggregate ground truth labels for the 299 song dataset. Because of the high dimensionality and relative sparseness of the Mood feature vector, linear discriminant analysis (LDA) was used as a preprocessing stage to improve the separability of the data by the classifier.

The LIBSVM [9] software library was used to perform classification of the 4 perceived tempo classes. The weighting feature of the SVM software implementation was used to compensate for the noted imbalance in the class sizes of the training data.

3.5. BPM Correction

Using the predicted tempo class, we applied the following heuristic rules for adjusting a baseline tempo estimate to correct for possible octave errors.

If tempo class is 1, and baseline BPM > 90:
Divide BPM by 2

If tempo class is 2, and baseline BPM > 115:
Divide BPM by 2

If tempo class is 3, and baseline BPM < 80:
Multiply BPM by 2

If tempo class is 4, and baseline BPM < 110:
Multiply BPM by 2

All other cases:
Leave BPM unchanged

For example, if a track is predicted by the tempo classifier to be "Very Slow" (Tempo Class 1), but the baseline tempo is estimated to be 160 BPM, we know there is a good chance that the estimate is too high by a factor of 2. Therefore, we divide the tempo estimate by 2 to arrive at a corrected estimate of 80 BPM. The difference in the threshold values for each tempo class reflects the relative ambiguity of each of the tempo classes.

4. EXPERIMENTS AND RESULTS

4.1. Perceived Tempo Classification Results

10-fold cross validation was performed on the Tempo Classifier using the annotated 299 song dataset. The distribution of the predicted tempo classes is shown in Table 3. Because of the unbalanced class sizes in the training set noted in 3.1, overall accuracy was measured by averaging the accuracy rate in each class over the size of that class, and then summing up the averaged class accuracy rates over all 4 classes. SVM parameters were accordingly selected to optimize this class-averaged accuracy rate, rather than the total accuracy rate usually derived by averaging all correct instances over the size of the entire dataset.

		Predicted			
		Class 1	**Class 2**	**Class 3**	**Class 4**
Ground truth	**Class 1**	**20**	12	0	1
	Class 2	20	**59**	28	4
	Class 3	7	26	**68**	24
	Class 4	0	1	14	**15**

Table 3 – Distribution of predicted tempo classes

The overall class-averaged accuracy of the tempo classifier was 55%. This relatively low accuracy rate reminds us again of the subjective nature of classifying tempos even into coarse tempo classes, as we saw from the variance of user annotations in Table 2. However, it is notable in Table 3 that most incorrect predictions are within 1 tempo class of the ground truth, with almost no instances of 2 class prediction errors. Thus a "very slow" song is rarely mistaken as "fast" (either class 3 or 4), while a "very fast" song is rarely mistaken as "slow" (either class 1 or 2).

As stated earlier, there are certain applications where an exact BPM or set of BPMs is not required, but where a relative tempo range is sufficient, such as automatic song selection and music categorization. In these applications, the perceived tempo class alone may provide performance on par with state-of-the-art BPM-centric tempo estimation methods, the best of which can

only achieve around 60% accuracy in estimating the most salient BPM for popular music content.

4.2. BPM Correction Results

4.2.1. ISMIR 2004 Dataset

On the ISMIR 2004 *Songs* dataset of 465 songs, we detected the perceived tempo class of each song, and then applied our heuristic rules for BPM correction to all the submitted algorithms' BPM estimates, including our own tempo estimation algorithm, labeled GN, as shown in Table 4.

Using the exact match (within 4% tolerance) accuracy metric, our tempo class-based BPM correction was able to increase the overall accuracy rate of exact matches for all 12 algorithms (See Figure 2). The BPM correction stage improved the accuracy of the best algorithm (KL from A. Klapuri) from 58% to 69% - an 18% improvement. Our own algorithm (GN) improved from a baseline accuracy of 56% to 62% after correction (an 11% improvement), ranking it 2nd amongst the algorithms submitted to ISMIR 2004 both before and after correction. The average improvement in accuracy across all algorithms was 45%.

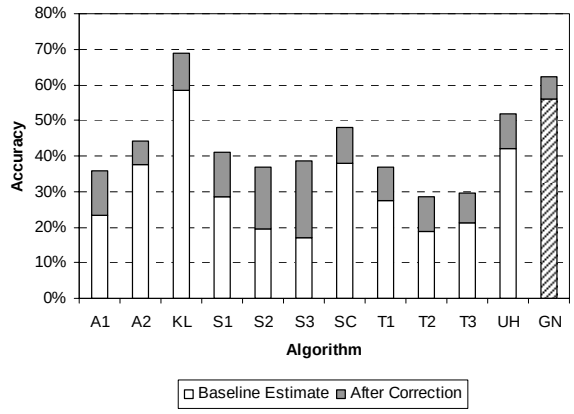

Figure 2 – Accuracies of tempo estimation algorithms for ISMIR 2004 *Songs* dataset before and after BPM correction. Gracenote algorithm (GN) is highlighted by diagonal pattern.

The "Half" and "Double" columns in Table 4 show that in most cases the incidence of half-time errors are decreased, while in all cases the incidence of double-time errors are decreased, in many cases quite significantly. Note that "Other" errors, such as one-third, triple, and non-integer multiple accuracy errors are not changed significantly by our BPM correction rules.

Algorithm	Exact	Half	Double	Other
GN	56%	5%	18%	21%
GN corrected	**62%**	**5%**	**12%**	**21%**
A1	23%	3%	31%	43%
A1 corrected	**36%**	**3%**	**19%**	**42%**
A2	37%	6%	24%	33%
A2 corrected	**44%**	**5%**	**18%**	**32%**
D1	29%	8%	23%	40%
D1 corrected	**41%**	**4%**	**17%**	**38%**
D2	19%	0%	46%	34%
D2 corrected	**37%**	**3%**	**28%**	**32%**
D3	17%	0%	58%	25%
D3 corrected	**38%**	**3%**	**36%**	**22%**
KL	58%	2%	30%	10%
KL corrected	**69%**	**7%**	**14%**	**11%**
SC	38%	6%	25%	32%
SC corrected	**48%**	**5%**	**15%**	**32%**
T1	21%	8%	12%	59%
T1 corrected	**30%**	**3%**	**11%**	**56%**
T2	19%	3%	17%	62%
T2 corrected	**29%**	**2%**	**9%**	**60%**
T3	28%	4%	20%	49%
T3 corrected	**37%**	**2%**	**14%**	**48%**
UH	42%	3%	25%	30%
UH corrected	**52%**	**5%**	**14%**	**30%**

Table 4 – Results of tempo estimation algorithms on ISMIR 2004 *Songs* dataset, showing accuracy rates before and after BPM correction

4.2.2. MIREX 2006 Dataset

The procedure was repeated on a dataset of 20 audio clips released as a training dataset for the MIREX 2006 tempo extraction task[2]. Two commercial tempo estimation software packages, MixMeister BPM Analyzer [10] and BeaTunes [11] were evaluated against our internal algorithm (Gracenote), as well as

[2] http://www.music-ir.org/mirex/2006/index.php/Audio_Tempo_Extraction

publicly available algorithms from D. Ellis [12], and G. Tzanetakis' MARSYAS music analysis library [13].

The ground truth of this dataset was annotated as a pair of BPM values along with a weighting factor. As discussed in Section 2.2, our goal is to improve the rate of correctly and exactly estimating only the single most perceptually salient BPM for a given song. Therefore, the ground truth BPM indicated by the weighting factor as being the most perceptually salient was used as the single reference BPM for evaluation. The algorithms from D. Ellis and MARSYAS also produce tempo estimates in the form of 2 BPMs along with a weighting factor. Similarly, only the more salient of these 2 estimates, as indicated by the weighting factors, was used in the evaluation.

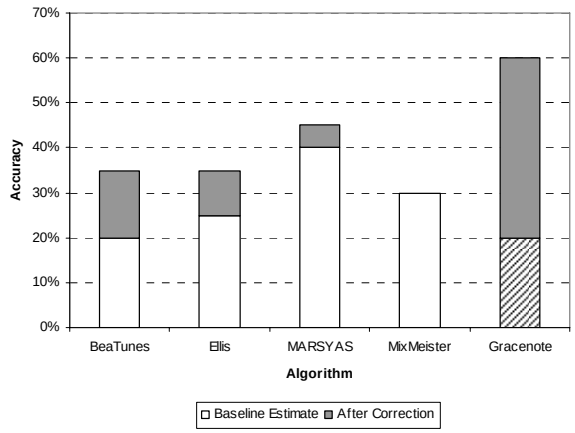

Figure 3 – Accuracies of tempo estimation algorithms for MIREX 2006 training dataset before and after BPM correction. Gracenote algorithm is highlighted by diagonal pattern.

The results of this evaluation are shown in Figure 3 and Table 5. All but 1 algorithm showed improvements in exact match accuracy after the BPM correction stage, while showing consistent decreases in the double and half-time errors. There was no change in the MixMeister results. The Gracenote algorithm showed the greatest increase, going from a baseline accuracy of 20% to 60% after BPM correction – an improvement of 200%. The average improvement across all algorithms was 65.5%.

It is worth noting that the proportion of "Other" errors (not exact, half, or double of the ground truth BPM) in this set is relatively large compared to the double and half errors, thus limiting the effect of our BPM correction stage. This could be the attributed to the relatively small dataset coupled with a relatively large proportion of tracks with triple meter rhythmic patterns in the dataset.

Algorithm	Exact	Half	Double	Other
Gracenote	20%	45%	10%	25%
Gracenote Corrected	**60%**	**15%**	**0%**	**25%**
BeaTunes	20%	5%	10%	65%
BeaTunes Corrected	**35%**	**0%**	**0%**	**65%**
MixMeister	30%	15%	10%	45%
MixMeister Corrected	**30%**	**15%**	**10%**	**45%**
Ellis	25%	5%	20%	50%
Ellis Corrected	**35%**	**5%**	**10%**	**50%**
Marsyas	40%	5%	10%	45%
Marsyas Corrected	**45%**	**5%**	**5%**	**45%**

Table 5 – Results of tempo estimation algorithms on MIREX 2006 training set, showing accuracy rates before and after BPM correction

5. CONCLUSIONS

The experimental results support the hypothesis that a significant and perceptually relevant relationship exists between high-level mood descriptors and perceived tempo. A statistical model of tempo classes using mood descriptors as features proved to be effective for discriminating between rough perceptual tempo classes, without any low-level analysis of temporal events and repetition rates in the audio signal.

The detected tempo class was also found to be extremely effective at improving the accuracy of existing tempo estimation algorithms, resulting in an improvement in all but one of 16 different algorithms, with an average improvement in accuracy of 45% and 65.5% for each of the two datasets utilized.

6. FUTURE WORK

While our method was successful in correcting for many double- and half-time errors, the heuristic rules used were fairly basic, and did not deal with one-third or

triple-time errors which occur frequently in music with a triple meter tempo. Future work may be done to create a more sophisticated set of BPM correction rules to decrease the instances of one-third and triple-time errors. A separate analysis stage may be required to estimate the metrical structure of the music in order to distinguish between duple and triple meters.

An outstanding issue with many of the evaluations performed in this paper is that of the tempo ground truth. As we have seen, the concept of perceived tempo is extremely subjective, and hard to boil down to a single BPM label. It would be interesting to attempt a large-scale collection of perceived tempo annotations, both at the BPM "foot-tapping rate" level as well as the coarse perceived tempo class level, from a large group of listeners to study the variability in user's perception of tempo.

A natural extension of this work would be to investigate the corresponding relationship between perceived tempo and other high-level musical descriptors, such as genre.

7. REFERENCES

[1] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An Experimental Comparison of Audio Tempo Induction Algorithms", IEEE Transactions on Audio, Speech, and Language Processing 14(5), pp. 1832-1844, 2006

[2] B.Y. Chua, G. Ku, "Determination of Perceptual Tempo of Music", Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 3310, 61–70, 2005

[3] D. Moelants, M. McKinney, "Tempo Perception and Musical Content: What Makes a Piece Fast, Slow, or Temporally Ambiguous?", Proceedings of the 8th International Conference on Music Perception & Cognition, 2004

[4] F. Gouyon, S. Dixon, "Influence of Input Features in Perceptual Tempo Induction", 2nd Annual Music Information Retrieval eXchange (MIREX), 2005

[5] K. Seyerlehner, G. Widmer, D. Schnitzer, "From Rhythm Patterns to Perceived Tempo", Proceedings of the International Conference on Music Information Retrieval, 2007

[6] L. Xiao, A. Tian, W. Li, J. Zhou, "Using a Statistic Model to Capture the Association between Timbre and Perceived Tempo", Proceedings of the International Conference on Music Information Retrieval, 2008

[7] H. Katayose, M. Imai, "Sentiment extraction in music", Proceeding of the 9th International Conference on Pattern Recognition, 1988

[8] L. Lu, D. Liu and H.-J. Zhang, "Automatic Mood Detection from Acoustic Music Data", Proceedings of the International Symposium of Music Information Retrieval, 2003

[9] C.-C. Chang and C.-J. Lin, "LIBSVM : A Library for Support Vector Machines", software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

[10] MixMeister BPM Analyzer, available from http://www.mixmeister.com

[11] BeaTunes, available from http://www.beatunes.com

[12] D. Ellis, "Beat Tracking by Dynamic Programming", Journal of New Music Research, Special Issue on Beat and Tempo Extraction, vol. 36 no. 1, pp. 51-60, 2007

[13] G. Tzanetakis, Music Analysis Retrieval and Synthesis for Audio Signals (MARSYAS) software library, available from http://marsyas.sness.net