Music Similarity Evaluation Based on Onsets

Shingchern D. You and Ro-Wei Chao

Abstract This paper describes a music similarity approach based on the time differences between two adjacent onsets. To better detect onsets, temporal and spectral detection methods are employed. Each set of detected features are individually matched by using the rough longest common subsequence (RLCS) algorithm. The final score is a weighted sum of individual scores from each detection method. The simulation results show that, on the average, 85% of the audiences agree that two musical soundtracks are similar if the computed score is greater than 0.3. When compared with an existing approach, it is easier for the proposed approach to set up a threshold to recommend highly similar soundtracks.

Keywords Music similarity · Onsets · Rough longest common subsequence

1 Introduction

With the advances of technology, more and more online soundtracks are available to music lovers. For many instances, a music lover may want to listen to more soundtracks similar to his/her favorite ones. To provide this type of service, techniques for music recommendation can be applied. Currently, there are two approaches to provide the recommendation list. The first one is based on the preference of other users, whereas the second one is based on the temporal and/or spectral similarity of the soundtracks. The first approach is easy to implement. For example, if two soundtracks A and B are frequently downloaded or listened by many users together, then we may assume that A and B are similar. Therefore, if the user requests to recommend soundtracks similar to A, then soundtrack B will be recommended. Though effective, this approach, nevertheless, does not truly recommend "similar" soundtracks to the query soundtrack. Furthermore, this approach

S.D. You (✉) · R.-W. Chao
Department of Computer Science and Information Engineering,
National Taipei University of Technology, Taipei, Taiwan
e-mail: you@csie.ntut.edu.tw

N.Y. Yen and J.C. Hung (eds.), *Frontier Computing*, Lecture Notes
in Electrical Engineering 422, DOI 10.1007/978-981-10-3187-8_16

almost always recommends most popular soundtracks at the time of query, and ignores any really similar ones with only a few downloads (or browse). Finally, this kind of approach requires Internet connection, which is inconvenient for some situations.

As the second approach assesses the temporal and/or spectral similarity between two soundtracks, the similarity is truly based on the contents of the soundtracks without referring to other users' preferences. For this type of approach, we could either provide a set of musical works to train the similarity evaluation system. Or, we could alternatively use a pre-defended metric to measure the similarity between two soundtracks. In this paper, we only consider approaches without prior training for its ease to use.

According to Wikipedia [1], there are many different criteria to assess whether two pieces of music are similar, such as based on pitched similarity, non-pitched similarity, and semiotic similarity. However, in actual implementation, approaches based on timbral and/or rhythmic pattern similarity are more popular because these approaches match the perceptual intuition of human beings.

E. Pampalk [2] proposed a similarity evaluation system based on MFCC (Mel-frequency cepstrum coefficients) and other features. The overall similarity score is a weighted sum of the feature distances. A release of his program is available in [3]. In this paper, we will compare the simulation results of our approach with Pampalk's approach.

The Austrian Research Institute for Artificial Intelligence (OFAI) has released another music similarity system in its official webpage [4]. This system uses both features from timbre and rhythmic patterns to evaluate the similarity of two soundtracks [5]. The comprehensive version of the program is subject to a license fee, whereas the basic version is open to public [6].

Other than these two systems, there are still other researchers conducting research in this area. One of the well-known competition on music similarity is MIREX (Music Information Retrieval Evaluation eXchange) [7], which attracts many teams to compete each year.

So far, most available music similarity approaches measure the similarity based on spectral and/or rhythmic similarity. The rhythm mentioned here actually means the regularity of temporal repetition of strong energy. Although rhythm is an important factor for similarity measure, it, nevertheless, is insufficient in some situations. In this paper, we use the relative time differences between onsets as features to measure the similarity between two soundtracks to increase the discrimination capability of temporal similarity. The purpose of this paper aims to provide an alternative similarity method other than existing ones. With more variations of similarity evaluation methods, hopefully the user can choose, among the approaches, a particular one to better serve his/her needs in the future.

2 Proposed Approach

According to Wikipedia [8], "*onset refers to the beginning of a musical note or other sound, in which the amplitude rises from zero to an initial peak.*" Currently there are many different approaches to detect onsets, including temporal and spectral approaches [9]. In this paper, both approaches are employed in onset detection.

As shown in Fig. 1, the proposed approach combines four different onset detection methods to find the similarity score. The first method is based on the variation of energy in time domain, denoted as temporal detection in Fig. 1. Onsets detected based on temporal approach provides acceptable accuracy if the audio signal contains strong energy variation. On the other hand, temporal detection is not accurate enough if relatively smooth (or slow) musical waveforms are encountered. For this type of signal, it is better to use spectral-based methods than temporal ones. To this end, we introduce three spectral-based methods in the proposed model, denoted in Fig. 1 as HFC (high-frequency contents) detection, spectral difference detection, and up-count detection. Once the onsets from a particular method are detected, the time difference between two adjacent onsets becomes one feature. The collected features are to be matched with the features in the database by using the

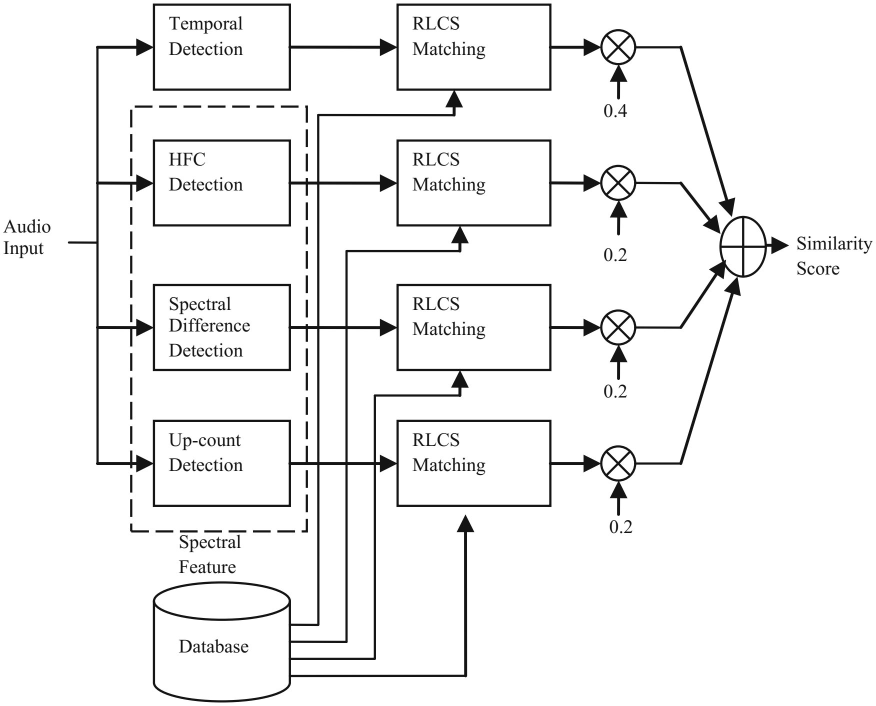

Fig. 1 Block diagram of the proposed approach

RLCS (rough longest common subsequence) algorithm [10]. The final score is a weighted sum of individual scores. The following briefly describes these methods.

2.1 Temporal Onset Detection

The processing flow of the temporal onset detection is given in Fig. 2. The incoming audio samples have a sample rate of 11,250 s/s. When the audio samples pass through a four-band filterbank, four sets of subband samples are obtained. The frequency bands of the filterbank are 0–630 Hz, 630–1720 Hz, 1720–4400 Hz, and higher than 4400 Hz. Let the obtained subband samples be denoted as $x_p(n)$, $1 \leq p \leq 4$. The subband samples are divided into frames of 512 samples. The energy of each frame is computed as follows:

$$E_p(n) = \frac{1}{512} \sum_{m=-256}^{255} \left| x_p(256n + m) \right|^2 w(m) \tag{1}$$

where $w(m)$ is the Hamming window. This step is denoted as computing band energy in Fig. 2. As the Hamming window is used, overlapping of 50% samples between successive frames are carried out. The obtained energy E_p is then undergone a first-order difference after taking the logarithm value [11]

$$A_p(n) = \log\left(E_p(n)\right) - \log\left(E_p(n-1)\right). \tag{2}$$

If $A_p(n_0)$ is a local maximum value, it is an onset candidate. To reduce the number of candidates, we remove any local maximum whose value is less 0.01 of the average amplitude, denoted as min peak threshold in Fig. 2. If $A_p(n_0)$ is a local maximum within 100 ms centered around n_0, then n_0 is a candidate position for an onset. In the decision-making step, if $A_p(n_0)$ is a candidate for all four bands, then $A_p(n_0)$ is determined as an onset [12]. The features used in the similarity comparison are based on the time difference between two adjacent onsets.

2.2 Onset Detection Based on Spectral Domain

This subsection describes the computational steps of the spectral-based onset detection blocks. As shown in Fig. 3, the pre-processing step for all spectral-based methods is to divide the incoming audio samples into frames, with each frame containing 512 samples. Samples in a frame are multiplied by a Hamming window with 50% overlapping. The windowed samples are transformed to spectral domain by FFT (fast Fourier transformation).

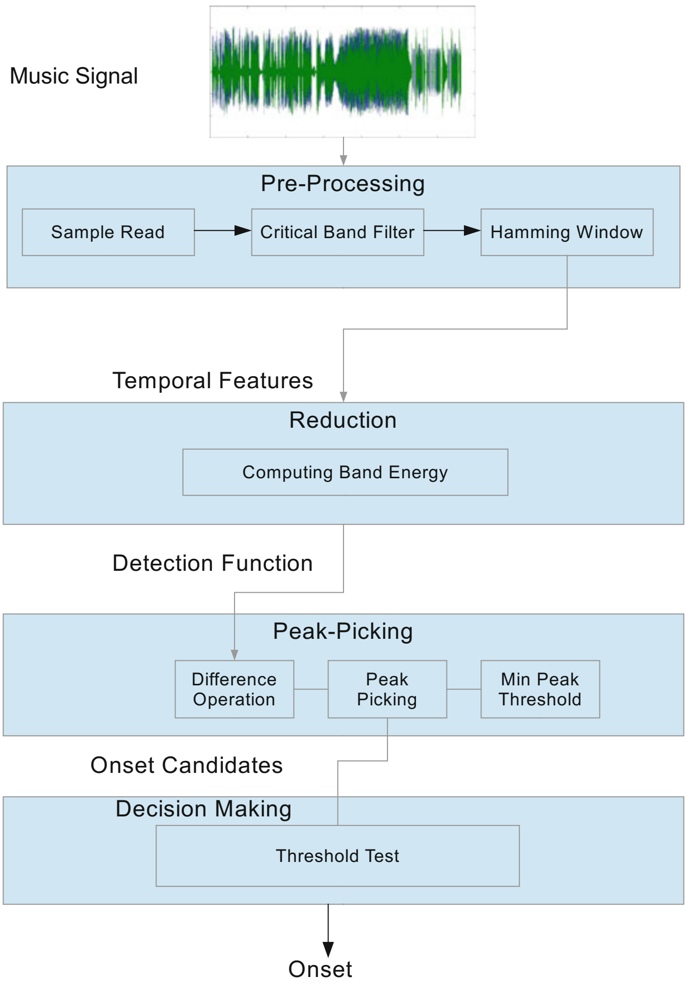

Fig. 2 Block diagram of the temporal onset detection

The HFC (high-frequency component) detection method [13] assumes that the variation of high-frequency energy is strongly correlated with onsets. Specifically, assume that (after FFT) the obtained spectral coefficient for frame n is denoted as $X_n(k)$. Then, the frequency-weighted energy is computed as

$$E_{HFC2}(n) = \sum_{k=1}^{256} k^2 \cdot |X_n(k)| \tag{3}$$

where k is the spectral index. The energy difference is then calculated as

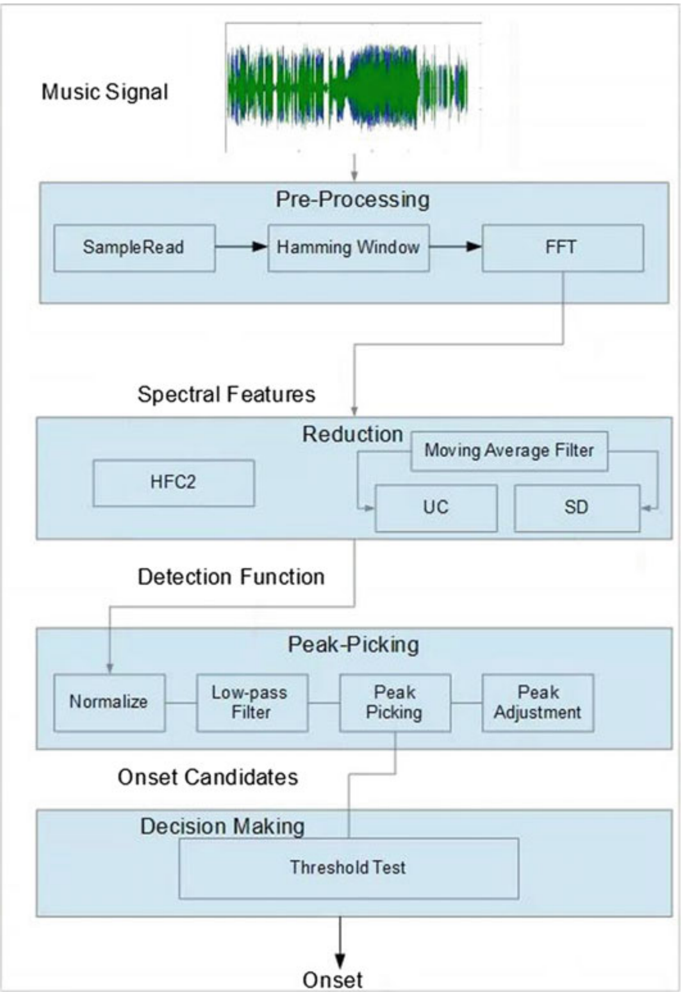

Fig. 3 Block diagram of the spectral onset detection

$$A_{HFC2}(n) = E_{HFC2}(n) - E_{HFC2}(n-1) \tag{4}$$

We will use $A_{HFC2}(n)$ in the decision-making step to determine onset locations.

The spectral difference detection method considers the spectral difference for each spectral index k [14]. To this end, this method computes $A_{SF}(n)$ by

$$A_{SF}(n) = \sum_{k=1}^{256} H\left(|X_n(k)| - \hat{X}_n(k)\right)^2 \tag{5}$$

where

$$\hat{X}_n(k) = \frac{1}{10}\sum_{m=1}^{10} |X_{n-m}(k)| \tag{6}$$

is a simple moving average of the past spectral coefficients to reduce the influence of noise, and

$$H(x) = (x + |x|)/2 \tag{7}$$

returns 0 for any non-positive argument x. Again, $A_{SF}(n)$ is to be used in the decision-making steps.

The up-count detection method is a modified version of the spectral difference method. As the former one is sensitive to noise, a possible modification is to count only the number of spectral lines with increasing energy, and ignores the actual (positive) value. Therefore, we use $A_{UC}(n)$ as the basis to determine the location of an onset:

$$A_{UC}(n) = \sum_{k=1}^{256} G\left(|X_n(k)| - \left|\hat{X}_n(k)\right|\right) \tag{8}$$

where

$$G(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Once we obtain $A_X(n)$ (x is either $HFC2$, SF, or UC), we use a moving average filter to reduce the fluctuation to obtain $\bar{A}_X(n)$. A onset candidate point is a location n_0 with $\bar{A}_X(n_0)$ is a local maximum in the vicinity of 100 ms. An onset is determined as a local maximum with its value exceeding a pre-defined threshold. Finally, the time difference between two adjacent onsets is a feature to be compared by the matching algorithm.

2.3 RLCS Algorithm

In addition to (time difference) features, we also need a matching algorithm to evaluate how similar two sequences of features are. For this purpose, we adopt a string-matching algorithm. Some well-known matching algorithms include dynamic warping, edit distance, and longest common subsequence. In this paper, we use the extension version of longest common subsequence algorithm, called rough longest common subsequence (RLCS) algorithm. Previously, we have used the RLCS algorithm for copy detection of music [15] with satisfactory results, and therefore we again use this algorithm for the proposed approach. For the sake of completeness and clear explanation, we outline the RLCS algorithm below.

Assume that there are two sequences (of strings) given as $A_i = \, < a_1, \, \ldots, a_i >, 1 \le i \le M$ and $B_j = \, < b_1, \, \ldots, b_j >, \; 1 \le j \le N$ with A_0 及 B_0 as empty sequences. The longest common subsequence can be computed as

$$L_{CS}(A_i, B_j) = \begin{cases} 0, & i \cdot j = 0 \\ L_{CS}(A_{i-1}, B_{j-1}) + \delta, & i \cdot j > 0, a_i \approx b_j \\ \max(L_{CS}(A_i, B_{j-1}), L_{CS}(A_i - 1, B_j)), & i \cdot j > 0, a_i ! \approx b_j \end{cases} \tag{10}$$

where "≈" means $|a_i - b_j| \le T_d$, "!≈" means $|a_i - b_j| > T_d$, and $\delta = 1 - \frac{|a_i - b_j|}{T_d}$. In the experiment, we use $T_d = 3$. We then compute width across reference (WAR) W_R and width across query (WAQ) W_Q functions as follows:

$$W_R(A_i, B_j) = \begin{cases} 0, & i \cdot j = 0 \\ W_R(A_{i-1}, B_{j-1}) + 1, & i \cdot j > 0, a_i \approx b_j \\ W_R(A_{i-1}, B_j) + 1, & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) \ge L_{CS}(A_i, B_{j-1}), W_R(A_i, B_{j-1}) > 0 \\ 0, & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) \ge L_{CS}(A_i, B_{j-1}), W_R(A_i, B_{j-1}) = 0 \\ W_R(A_i, B_{j+1}), & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) < L_{CS}(A_i, B_{j-1}) \end{cases} \tag{11}$$

and

$$W_Q(A_i, \, B_j) = \begin{cases} 0, & i \cdot j = 0 \\ W_Q(A_{i-1}, B_{j-1}) + 1, & i \cdot j > 0, a_i \approx b_j \\ W_Q(A_{i-1}, B_j), & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) \ge L_{CS}(A_i, B_{j-1}) \\ W_Q(A_i, B_{j-1}) + 1, & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) < L_{CS}(A_i, B_{j-1}), W_Q(A_i, B_{j-1}) > 0 \\ 0, & i \cdot j > 0, a_i ! \approx b_j, L_{CS}(A_{i-1}, B_j) < L_{CS}(A_i, B_{j-1}), W_Q(A_i, B_{j-1}) = 0 \end{cases} \tag{12}$$

The similarity is given as

$$S_{RLCS}(A, B) = \max_{i,j} score(i, j) \tag{13}$$

where

$$score(i, j) = \begin{cases} \frac{L_{CS}(A_i, B_j)}{N} \cdot \left(\frac{\alpha \cdot L_{CS}(A_i, B_j)}{W_R(A_i, B_j)} + \frac{(1-\alpha) \cdot L_{CS}(A_i, B_j)}{W_Q(A_i, B_j)} \right), L_{CS}(A_i, B_j) \ge \lambda \cdot N \\ 0, \text{otherwise} \end{cases} \tag{14}$$

In the experiment, λ is ($1/N$) and α is 0.5. We know from (14) that the value of $S_{RLCS}(i, j)$ is between 0 and 1 and 1, means perfectly matched.

3 Experiments and Results

To perform the experiments, we collect 38 soundtracks of classic music and 94 soundtracks of pop music from various albums. The duration of each soundtrack is 30 s. The original sample rate for each soundtrack is 44,100 s/s. However, the sample rate is reduced to 11,025 s/s before conducting the experiments.

When the user input a particular soundtrack, the proposed system computes the features for the input soundtrack. The computed features are then compared with the features in the database through the weighted sum S_{WRLCS} of four S_{RLCS} scores.

To understand the correlation between the computed S_{WRLCS} and the perceptual impression of a human listener, we conduct a listening test. In the test, five (5) soundtracks are selected as the input to the system. The computed S_{WRLCS} are divided into four categories: $0 < S_{WRLCS} \leq 0.1$, $0.1 < S_{WRLCS} \leq 0.2$, $0.2 < S_{WRLCS} \leq 0.3$, and $S_{WRLCS} > 0.3$. The soundtrack corresponding to the greatest and smallest scores in each category is selected. Thus, totally eight soundtracks are picked for each testing input. Ten (10) audiences are asked to give opinions regarding whether the testing soundtrack is similar to one of the eight picked soundtracks (individually compared). The experimental results are given in Table 1. It can be observed that if the S_{WRLCS} score is greater than 0.3, on the average, 85% of the audiences feel that both soundtracks are perceptually similar. Therefore, we can use this value as a threshold to recommend soundtracks to a user.

To further investigate the performance of the proposed approach, we compare ours with the approach proposed by Pampalk [3]. We use the same testing soundtracks for listening tests as the input to both systems. The scores for both systems are given in Table 2. For the proposed system, the range of the score is between 0 and 1, and 1 means highest similarity. On the other hand, the scores of

Table 1 Subjective similarity versus S_{WRLCS}

S_{WRLCS} score	Similar percentage (%)	Dissimilar percentage (%)
$S_{WRLCS} > 0.3$	85	15
$0.2 < S_{WRLCS} \leq 0.3$	40	60
$0.1 < S_{WRLCS} \leq 0.2$	0	100
$0 < S_{WRLCS} \leq 0.1$	0	100

Table 2 Similarity scores of both approaches

Query index	Score (proposed)		Score (Pampalk)	
	First	Fifth	First	Fifth
1	0.3552	0.3019	9.2669	9.3085
2	0.3691	0.3111	9.3068	9.3922
3	0.5376	0.2644	9.0575	9.4085
4	0.3691	0.3084	9.3586	9.44
5	0.3102	0.2701	9.2172	9.4061

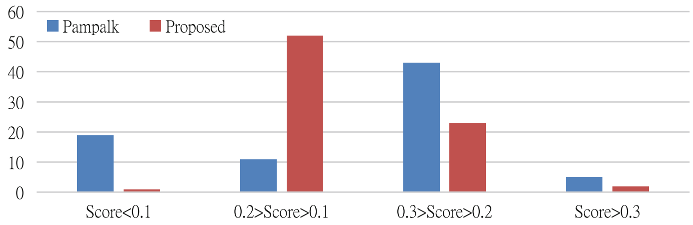

Fig. 4 Cross comparison between the proposed system and the Pampalk system

the Pampalk approach ranges between 9 and 13, with 9 as the highest similarity. It can be seen that the proposed system has larger (normalized to the full range of 1) score differences between the first (best) match and the fifth match, whereas scores obtained by the Pampalk approach have relatively smaller differences (normalized to the full range of 4). Conceptually, a larger difference (wider distribution) means that it is easier to set a threshold to recommend truly similar soundtracks. In this regard, the proposed system is a better choice.

When cross-comparing the number of soundtracks in the four categories mentioned previously, the results becomes apparent. As shown in Fig. 4, the proposed system has many more soundtracks with scores less than 0.2 and fewer soundtracks with scores of greater than 0.3. As a score (in the proposed system) less than 0.2 means that both soundtracks are not similar at all, the proposed system can better discriminate dissimilar soundtracks than the Pampalk approach.

4 Conclusions

This paper describes an approach for music similarity evaluation based on the detected onsets. When combining scores computed from individual onset features with the RLCS algorithm, the proposed approach is able to provide a final, weighted score for two soundtracks. The listening tests confirm that if two soundtracks have a similarity score of 0.3 or higher, these two soundtracks are perceptually similar according to the opinions of the listeners. When compared with an existing system, the proposed approach has a better score distribution to ease the determination of a threshold to recommend highly similar soundtrack titles among the titles in the database. Overall, the proposed approach is a possible choice for users to choose other than existing similarity evaluation methods.

References

1. https://en.wikipedia.org/wiki/Musical_similarity, retrieved Dec. 10, 2015
2. Pampalk, E.: Computational Models of Music Similarity and Their Application in Music Information Retrieval. Doctoral Thesis, Vienna University of Technology, Austria (2006)
3. http://www.pampalk.at/ma/, retrieved Dec. 10, 2015
4. The Austrian Research Institute for Artificial Intelligence, (OFAI), Music Similarity and Recommendation, available at http://www.ofai.at/research/impml/technology/musly.html
5. Pohle, T., et al.: On Rhythm and General Music Similarity. In: 10th International Society for Music Information Retrieval Conference, pp. 525–530 (2009)
6. Audio Music Similarity, available at http://www.musly.org/, retrieved Dec. 10, 2015
7. http://www.music-ir.org/mirex/wiki/MIREX_HOME, retrieved Dec. 10, 2015
8. https://en.wikipedia.org/wiki/Onset_(audio), retrieved Feb. 25, 2016
9. Bello, J. P., et al: A Tutorial on Onset Detection in Music Signals. IEEE Trans. Speech and Audio Processing. 13, 1035–1047 (2005)
10. Lin, H.-J., Wu, H.-H., Wang, C.-W.: Music Matching Based on Rough Longest Common Subsequence. J. Info. Sci. Eng. 27, 95–110 (2011)
11. Klapuri, A.: Sound Onset Detection by Applying Psychoacoustic Knowledge. In: 1999 IEEE International Conference on. Acoustics, Speech, and Signal Processing, pp. 3089–3092. IEEE Press, New York (1999)
12. Ricard, J.: An Implementation of Multi-band Onset Detection. In: 1st Annu. Music Inf. Retrieval Evaluation eXchange (MIREX), pp. 1–4. (2005)
13. Masri, P. and Bateman, A.: Improved Modeling of Attack Transients in Music Analysis-resynthesis. In: 1996 International Computer Music Conference, pp. 100–103. (1996)
14. Duxbury, C., Sandler, M., and Davies, M. A Hybrid Approach to Musical Note Onset Detection. In: 2002 Digital Audio Effects Conf. (DAFX,'02), pp. 33–38. (2002)
15. You, S. D. and Pu, Y.-H.: Using Paired Distances of Signal Peaks in Stereo Channels as Fingerprints for Copy Identification. ACM Trans. Multimedia Comput. Commun. Appl. 12, 1–22 (Aug. 2015)