The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

Intelligent Multimedia Annotation and Interaction Using Semantic Musical Features

Encoding Human-Centric Music Intelligence for Musically Plausible Human-Media Interactions

Ren Gang, Zhe Wen, Xuchen Yang, Mark F. Bocko, Dave Headlam
Dept. of Electrical and Computer Engineering, Edmund A. Hajim School of Engineering and Applied Sciences, Univ. of Rochester
Dept. of Music Theory, Eastman School of Music, Univ. of Rochester
Rochester, NY 14627, USA
g.ren@rochester.edu, zhe.wen@rochester.edu, xuchenyang@rochester.edu, dheadlam@esm.rochester.edu, mark.bocko@rochester.edu

***Abstract*—Semantic musical features reflect in-depth understanding of the music, instead of the uninterpreted music content, and serve as idea choices for multimedia content annotations. The proposed semantic music features are based on human music interpretations and their computational implementations. When employed for multimedia applications, these features enable us to simulate human-music interactions. This musical relevance provides significant performance improvement over conventional score or audio based multimedia annotation systems. Two types of semantic musical features, including reductive music analysis and musical expressive features, are introduced. The details of their feature extraction algorithms and semantic interpretations are also illustrated.**

Keywords- knowledge engineering; multimedia annotation; feature analysis; human-computer interaction

I. INTRODUCTION

The rich emotional and aesthetic elements in music exert important influence on our perception and understanding of multimedia content. Music-based "semantic" features, which are based on the human music understanding and interpretation, are especially suitable for multimedia content annotation applications. Because music and its attached media content are intrinsically entangled, the content labels based on the accompanying music can be directly applied as the semantic content labels: we simply compose or select the "right" accompanying music because it provides a sound description of the synchronized the media content. For example, background music in movies effectively set the mood and the pace for the visual aspects [1]. The tempo variations of the background music (a type of semantic music feature) thus serve as a semantic descriptor of this movie segment. In this paper we introduce two types of semantic musical features including reductive music analysis and musical expressive features. We then introduce their interpretations and perceptual relevance when applied as data interfaces for multimedia applications.

The term *semantic musical features* refer to interpretive musical features that reflect the understanding of musical "meaning", instead of uninterpreted music score or audio. The proposed semantic musical features are defined as large-span features to differentiate them from small-scale structures such as local harmony and voice-leading pattern. As these small-scale structures typically cover a short music segment (typically shorter than a music phrase, or approximately 8 bars), the two types of semantic music features we proposed provide multimedia annotation for music units larger or even span a whole music composition. In this paper we focused on the expectation-relaxation patterns derived from these two types of semantic music features. The proposed semantic music patterns serve as intelligent data interface for various multimedia applications such as semantic media web, electronic games, intelligent audio/video player, content-based multimedia information retrieval, automatic media production, and interactive media.

Existing frameworks for multimedia content annotation are summarized in [2-6]. The content labels of these systems are based on two types of features. 1) The content features, which include the music-score-based features [2,3] or audio signal features [4,5], are the standard musical features for multimedia annotation. These features are convenient to extract and easy to apply, but their implementations are unsatisfactory because they reflect the uninterpreted music score or audio, rather than a plausible music understanding. 2) Manually complied annotation scripts [6] can embed the music interpretation of the human producers and thus gives the end-user an impression that the computer program does understand the music. However, the manual analysis and programming burden renders this approach impractical for applications on large-scale multimedia database/network. The aim of this paper is to introduce "music meaning" based features into this application area to better simulate human-based musical interactions to achieve higher music releavance. The two types of semantic music features we introduced are based on human music understanding and are thus cognitively more relevant compared to simple content features. They can also be automatically compiled at certain level to facilitate practical implementations.

More detailed descriptions of reductive music analysis can be found in [7] for Schenkerian analysis and in [8] for Generative Theory of Tonal Music (GTTM). Existing implementations of automatic Schenkerian analysis is summarized in [9]. The computational implementations of GTTM are summarized in [10,11]. In this paper we formalize and adapt these analytic results for multimedia annotation applications. Specifically we use a multi-dimension feature sequence to encode these reductive music analysis results. This data format encodes important pattern as feature labels that can be readily applied to multimedia application. Existing frameworks of musical expressive feature extraction is illustrated in [12-14]. In this paper we

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

focused on the feature extraction process of obtaining musical expressive feature from the performance audio. This audio-based feature extraction process is more convenient than the sensor-based approaches in [15] because music performance recordings are widely available.

The concept and processing algorithms for these two types of semantic musical features is detailed in Sec. II and Sec. III. Sec. IV evaluates the cognitive relevance of these semantic music features. Sec. V provides a brief summary and illustrates future research topics.

II. Semantic Musical Features Based on Reductive Music Analysis

A. What Is Reductive Music Analysis?

The reductive musical analysis is an interpretative procedure in music theory that reveals the in-depth structure of a music score. A reductive musical analysis gives depth to a music score by decomposing the music surface into a hierarchical structure-elaboration representation. Several approaches have been developed for music theoretical reductive analysis, including Schenkerian analysis [7] and a generative theory of tonal music (GTTM) [8]. In reductive music analysis, music score is referred as music surface [7] because they act as the beginning part of an analysis and music theorists are expected to explore beyond this surface and find additional structures. A reductive music analysis provide an additional depth dimension by finding simplified music contents from the music surface of original music piece.

An example of music reductive analysis is detailed in Fig. 1. This reductive process is a typical music theoretical analysis for tonal music compositions. This short analysis example is based on Schenkerian analysis [7,9] and serves as an illustrative example. The music score at the highest analysis layer (layer 1 as music surface, in Schenkerian term, foreground) is supposed to be generated from a simpler layer (middle ground) beneath it. Comparing layer 1 and layer 2 we could observe in analysis path 1-1that the accented C in the first measure of layer 2 is extended (elaborated) to D, C, B, C in layer 1 by adding step-wise development (neighbor notes in the surface elaboration). The hierarchy created by the above reductive procedures represents the music understanding at different abstraction/resolution levels. Because each analysis layer is simpler than the previous layers, this analysis is termed "reductive".

The surface layer as in Fig. 1 is composed of sequential data from music score and represents a 'flat' topology with no specific priority of each music events. After applying the first round of reduction processing, the score events in the surface plane are sorted as core-structure and embellishment according to the rules as detailed in [7]. The score events whose roles are categorized as embellishment in the surface are reduced and the remaining part is passed to the second analysis layer. For this example here the second analysis layer is very significant in our listening experience, the simplest layer (layer 3) is not the most perceptually-silent layer but rather a goal for this reductive process. In this illustrative example we only deal with a few music phrases. For practical analysis the data score scale can expand a whole music work, as demonstrated in Henrich Shenker's original work "Free Composition [16]".

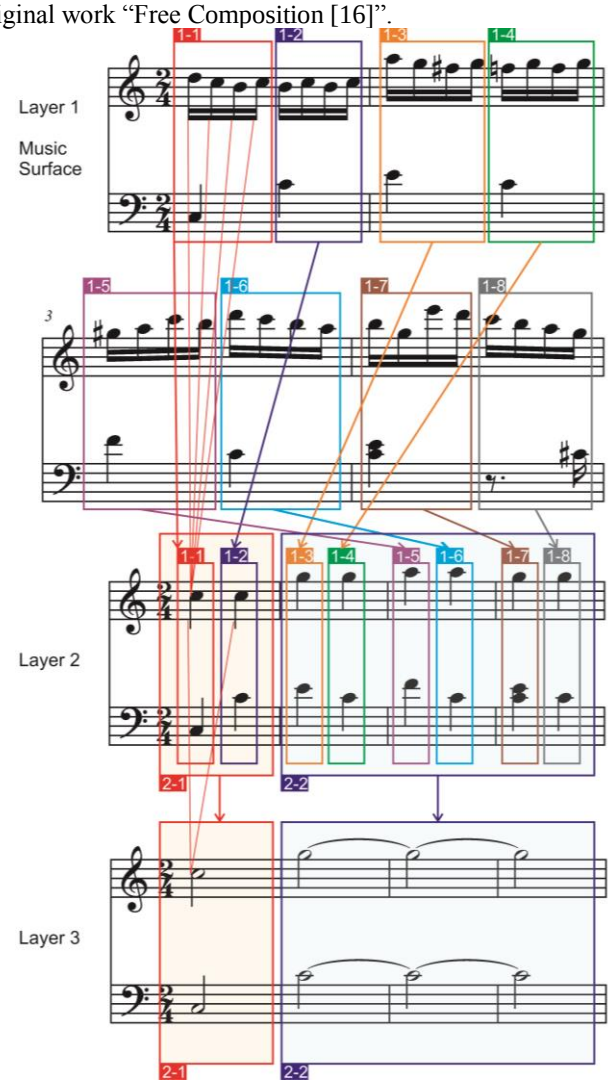

Fig. 1. An example of reductive music analysis. The original music score is illustrated as music surface in layer 1. The music surface is modeled as generated from simpler levels beneath.

B. Basic Analysis Methodologies

Fig. 2 is an illustration of three typical reductive processes. In each reductive process, the music event in top layer is sorted as structural events and elaboration events. At the lower (deeper) analysis layer the elaboration event is reduced and only the structurally salient music notes are kept. In Fig. 2.1(a) we introduce an arpeggiation. An arpeggiation is defined as a series of music notes skip between the constituent notes of chord in the same direction [7]. Here C5, E5 and G5 at the top layer belong to C major harmony. In the lower layers they are reduced to C5. This reduction process provide the following interpretations concerning this C5-E5-G5 sequence: (1) E5 and G5 are

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

related to C5 by their decorating roles; (2) the music role of C5 is more prominent (structurally) than E5 and G5: C5 then further plays a different role in a larger scale while in reduction E5 and G5 are abstracted off so they do not play a

subjective process [16,17,18]. The same reductive music structure is open for multiple alternative interpretations so multiple reductive structures are possible.

In the examples presented in Fig. 2(a)-(c) the reductive

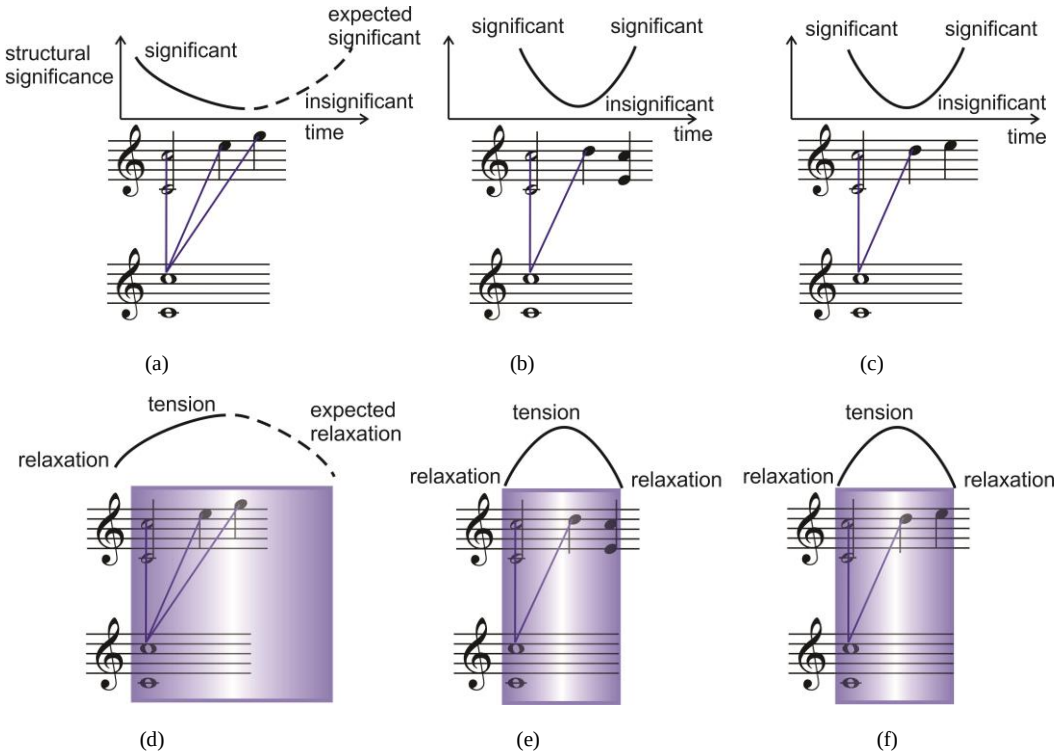

Fig. 2. Elementary reductive music analysis and their "metrical wave" interpretations. In (a)-(c) the three reductive analysis examples are attached with structural significance labels. The wave shape demonstrates that the structural significance is changing with time. In (d)-(f) we further produce a representation "mask" based on the process of building tension and relaxing tension.

In Fig. 2.1(b) we introduce a neighbor note (Nebennote in German). A neighbor note is related to the music note it elaborates by step, and is usually dissonant with the supporting harmony [7]. In this example the note D5 is related by step to the structural note C5. In (c) we illustrate a linear progression (Zug). A linear progression means step-wise motion in one direction between two harmony notes. The D5 fill in the blanks between C5 and E5. C5 and E5 are consonant notes of the supporting C major harmony and are treated as structurally significant.

C. Semantic Interpretations

The rationale behind using this reductive music analysis results as semantic music features is that this reductive structure reflects an understanding of music. The similarities in these analysis layers could be encoded in structural labels and matched to the analysis layers of another music piece so the cross-layer similarities could be identified. The GTTM further embeds a cognitive basis so the structural similarities are empirically linked to subjective listening experience.

Fig. 2(d)-(f) further provides a "metrical wave" interpretation of the music meaning of the reductive steps. This interpretation process of a reductive music structure is a time. In Fig. 2 (d)-(f) these examples are interpreted more intuitively as tension and relaxation patterns. In the arpeggiation example of Fig. 2 (a) and (d), the first music event C5 is important because (1) it is the first harmonic note of the note sequence of C major chord C5-E5-G5; (2) it is supported by C4. Since E5 and G5 are just a repetition of this harmony, they are less important in music structure.

In Fig. 2 (d) the repetition of harmonic notes in a chord is interpreted as building tension. The listeners are wondering if there is changes of music chord or voice-leading directions following E5 and G5. The longer repetition leads to more intense expectations of the imaginary resolution. In Fig. 2 (b)(e) the neighbor note is dissonant with the supporting harmony and thus treated as structurally less significant. Because in tonal music dissonant music event have a strong tendency towards return back to consonance, D5 here builds up musical tension. The tension is subsequently resolved by going back to consonant note at the second C5. In Fig. 2(c)(f) the linear progression patterns follows a similar music pattern, where inharmonic note D5 is interpreted as music

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

tension. The tension is resolved by going back to consonant note E5.

D. Implementations

Currently manual interventions are still required for reliable reductive music analysis. For practical applications a complete analysis is not always necessary and minor analysis mistakes could be tolerated. In these application scenarios, automatic procedures for deriving reductive music structure analysis as summarized in [9-11] can be applied. The analysis result is then encoded as a multi-dimensional feature sequence as semantic annotations for concurring media.

The complete representation scheme for Schenkerian analysis results is detailed in [9] and here we only provide a simplified description. In this representation format, we focus on the layer-location of music events and their cross-layer links.

For each reductive layer the following basic analysis features are included:

- Score event index: A sequential index of music event. The index is based on time order of the onset of music events. For concurring music event the index order is arbitrary.
- Score pitch information: The pitch information includes the score pitch, which is the MIDI sequential number of score pitch, and the pitch class number, as the modulo 12 of pitch. Pitch class information here assumes octave equivalence. Two pitches a and b are in the same pitch Equivalence class if $b = 12 \cdot n + a$, where n is some integer [7].
- Score timing information: Each different onset location is assigned a timing sequential number, in this representation only the onset location is encoded and the duration of such music event is ignored as a simplification. For more details of the treatment of timing information in reductive music analysis see [7,11].

We also assign a structural role to each music note. The most important role of a music note is its harmonic features and voice leading features. These two types of features is based on the connection of music events within one analysis layer and serves as contextual features within a reductive layer.

Harmonic Feature:

- Harmonic support: if the music note are supported by certain harmony at the music surface (original music score). 0-Supported by concurring music event; 1-unsupported by concurring music event but belonging to a harmonic support region; 2-not supported at the music surface, and not belonging to a harmonic support region.
- Harmonic support type: the harmony category of the supported music notes.

Voice-leading feature:

- Pitch interval towards left contextual note: The pitch interval here is calculated as the ordered pitch interval, as the different between two pitches. The left contextual note is the most adjacent music event.
- Pitch class interval towards left contextual note: The pitch class interval here is calculated as the unordered pitch class interval. For two pitch classes a and b, the ordered pitch class interval is $b - a \ mod\ 12$. The unordered pitch class interval is the smaller of $b - a \ mod\ 12$ and $a - b \ mod\ 12$.
- Pitch interval toward right contextual note
- Pitch class interval toward right contextual note

The reductive structural labels then encode the connections of music notes at adjacent reductive layers.

- Reduction label: If this note is retained in the next reductive layer. 0-the note is kept; 1-this note is reduced.
- Atomic Elaboration Type: Several types of elaboration are prescribed as the basis for reduction. These elaborations are detailed in [9].

III. SEMANTIC MUSICAL FEATURES BASED ON MUSICAL EXPRESSIVE FEATURES

A. What are Musical Expressive Features?

Musical expressive features here are defined as the signal features extracted from performance audio that reflect the subtle but important variations the musicians append beyond the score. Musical expressive features are perfect carries of music meaning because they are based on human music understandings. To perform a music score a professional musician need rigorous music training which provides the contextual understanding of a musical work, and more importantly, to develop a "musical mind" that create the subtle modulations and shaping of musical lines that provide music's emotional impact. In this creative process a professional musician not only infers plausible expressive information from the score and its historical background, but also creatively append his/her personal touches [19].

B. Dimensions of Musical Expressive Features

The expressive transcription features we proposed include two feature categories. Score-level transcription features serve as a middle-layer representation that provides the feature descriptors that can be reduced to a symbolic music score. Performance-level transcription features are related to the musical expressive features that depict the essential performance information.

Conventional music transcription algorithms [20] obtain score-related audio features including fundamental frequency (F0) and onset times from audio analysis. A quantization grid is estimated from these audio features, which are further split into score-level transcription features and performance-level features. The score-level transcription features are essentially the nearest neighbor of the audio features on a quantization grid, while the performance-level features, or musical expressive features, are the deviations in the performance audio features from the estimated quantization

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

grid. The onset detection results also allow us to partition the audio into segments corresponding to individual music notes, so additional dimensions of performance-level transcription features including dynamics, timbre, articulation and vibrato can be obtained from the segmented audio as signal features or feature patterns detected from audio segments. More details of musical expressive feature is covered in [14].

The variations of these musical expressive features can be readily interpreted as large-scale expectation-relaxation patterns. For example, the music tension caused by pitch deviation increase is subsequently resolved by pitch decrease. The tempo-tension of compressed timeline could be resolved by subsequent tempo-relaxation.

C. Feature Extraction Using Score-Audio Alignment Methods

When a matching music score of the performance audio is available, the performance-level features are obtained by comparing the score with the audio. First we perform the music event alignment algorithm as in [21]. This alignment algorithm maps the score music event to the time-frequency locations of performance audio using dynamic time warping, which optimally aligns the variation pattern of pitch/timing features obtained from score and audio. The score pitch is converted to a fundamental frequency using a temperament system which is derived from a reference frequency point $\bar{f}_R$ with symbolic pitch value p_R as:

$$\bar{f}_m = \text{tpa}(\bar{f}_R, p_R; p_m) \quad (1)$$

where p_m is the symbolic pitch value of frequency point $\bar{f}_m$, tpa() indicates a temperament function. For equal temperament scale $\bar{f}_m$ could be calculated as:

$$\bar{f}_m = \text{tpa}_e(\bar{f}_R, p_R; p_m) = 2^{\frac{p_m - p_R}{12}} \cdot \bar{f}_R \quad (2)$$

The logarithmic value of $\bar{f}_m$ is:

$$12 \cdot \log_2 \bar{f}_m = 12 \cdot \log_2 \bar{f}_R + p_m - p_R \quad (3)$$

Here p_m and p_R is specified in MIDI value. Since human frequency discernment is most acute at mid-frequency region, the reference point $[p_R : \bar{f}_R]$ could be selected at this frequency region. In our implementation an initial reference point is selected as [69:440Hz]. Then we shift the frequency reference point in 160 small steps within 1/6 of a semitone interval and find the best reference frequency point $\bar{f}_R + \Delta\bar{f}_R^*$ where a F0 alignment cost is minimized. The F0 alignment cost here is a weighted sum of frequency mis-alignments d_l between the audio F0 and the score pitch according to the temperament grid as:

$$C(\Delta\bar{f}_R) = \sum_{l=1}^{L} \eta(f_l) \left| d_l(\Delta\bar{f}_R) \right| \quad (4)$$

Here $\left| d_l(\Delta\bar{f}_R) \right|$ denotes the frequency distance of l th alignment event when the reference point shift is $\Delta\bar{f}_R$. The small variation $\Delta\bar{f}_R$ is incorporated into the reference pitch to shift the temperament grid so the d_l values are changing with $\Delta\bar{f}_R$. The weights $\eta(f_l)$ are based on the frequency discrimination model as introduced in [5] and f_l is the F0 of of lth alignment event. Larger $\eta(f_l)$s are assigned for higher frequencies (especially for frequencies higher than 2kHz) since human can discern frequency better at these frequencies. Using the optimal reference frequency point $\bar{f}_R + \Delta\bar{f}_R^*$ where the alignment result is minimized, we can calculate the pitch deviation of each music event by comparing the audio pitch and the score pitch. The pitch deviation of music event l in the units of cents (A cent represents 1/100 of a semitone) is calculated as:

$$\Delta p_m = 1200 \cdot \log_2 \frac{d_l(\Delta\bar{f}_R^*)}{\bar{f}_l} \quad (5)$$

For our proposed expressive transcription applications, the alignment algorithm as in [21] provides score-aided music event segmentation functionalities. For monophonic music the segmentation results provide the onset and offset of each music events. For polyphonic music the segmentation results further group sonic partials into instrument tracks. For monophonic music or an instrument track of polyphonic music the segmentation result is represented as $\{[e_s, t_s] | s \in 1, \dots S\}$, where e_s denotes a music event prescribed by the music score and t_s denotes its onset time location. The expressive timing features are obtained by comparing the score timing and the performance timing. The time deviation [13] of music event e_s is calculated as the normalized difference between audio onset timing $t(e_s)$ and the interpolated score timing $\hat{t}(e_s)$:

$$F_T(e_s) = \frac{t(e_{s+1}) - t(e_s)}{\hat{t}(e_{s+1}) - \hat{t}(e_s)} \quad (6)$$

Here onset time deviation is normalized by the interpolated score note duration [9] and the deviation value of previous notes is deduced. $t(e_{s+1})$ denotes the next onset location. $F_T(e_s)$ can be viewed as an indicator of the extension ($F_T(e_s) > 1$) or compression ($F_T(e_s) < 1$) of the audio segment of current notes. From different interpolation settings of score timing this method produces an expressive timing hierarchy. If the score timing interpolation is based on a long audio segment, macro-scale timing is obtained. We can then shorten the interpolation range to music phrase or individual meter for a micro-scale analysis. The score-audio alignment results also segment the audio so other performance-level feature dimensions including dynamics, timbre, articulation and vibrato could be obtained using the same feature extraction algorithms as in [14].

D. Feature Extraction Using Audio Feature Quantization Methods

When a music score is not available, a quantization process [24] is implemented to transform the audio features including F0 and onset timing to score-level transcription features including score-pitch and score-timing, or further format the features as a symbolic music score. The residue signal of this quantization process serves as the basis for performance-level transcription features including pitch deviation and performance timing after calibration. In this section the process of audio feature quantization and calibration is detailed, with an emphasis on pitch and timing features. The process of F0 estimation and onset detection is detailed in [20] and will not be covered here.

Suppose an F0 sequence we obtained is represented as $f_1 \dots f_M$, the goal of the pitch quantization process is to

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

obtain the score pitches as the quantized value $p_1 \dots p_M$ and the performance pitch deviations as the residual values (after caliberation) $d_1 \dots d_M$. This task is equivalent of finding a quantization code book $\left[\check{f}_m, \hat{f}_m; \bar{f}_m, p_m\right], m = 1, \dots, M$. Here $\check{f}_m$ and $\hat{f}_m$ s serves as the decision boundaries of the quantization grid. $\bar{f}_m$ is the quantized value that is selected if $\check{f}_m \leq f_m \leq \hat{f}_m$ and p_m is its symbolic value. The quantized value $\bar{f}_m$s form a temperament grid same as (3). The detected F0 values are quantized using the initial temperament grid. Suppose the frequency quantized values $\bar{f}_1 \dots \bar{f}_M$ are selected as the nearest neighbors in the quantization grid of the F0 sequence $f_1 \dots f_M$. The residual values of this quantization process are denoted as $d_1 \dots d_M$. Then we shift the frequency reference point to find the best reference frequency point $\bar{f}_R + \Delta\bar{f}_R^*$ where a weighted sum of the residual values $\sum_{m=1}^{M} \eta(f_m)\left|d_m\left(\Delta\bar{f}_R\right)\right|$ as calculated in (4) is minimized. After this calibration processes the residual frequencies according to this optimal temperament grid with values $d_1\left(\Delta\bar{f}_R^*\right) \dots d_M\left(\Delta\bar{f}_R^*\right)$ are pitch deviation values. The pitch deviation is then calculated using (4). The pitch calibration process we introduced here does not have a significant effect on symbolic score detection since in most application scenarios the musical pitch is well calibrated and errors in the process will never surpass the detection grid of adjacent semitones. For expressive feature extraction this calibration functionality is crucial because the calibration value is within the same range of pitch deviation values.

An onset sequence $t_1 \dots t_N$ is detected using the algorithms in [24] and its *inter-onset intervals* (IOI) are the time distances of adjacent onsets and are calculated as $v_n = t_{n+1} - t_n$. The rhythmic analysis result is obtained from the IOI sequence using the algorithm described in [13]. The rhythmic analysis result as obtained can be represented as a hierarchical time sequence. In Fig. 3 a musical timing hierarchy and its alignment with audio onsets and IOIs is illustrated. The timing hierarchy is stretched or compressed according to the beat tracking result. Since we only have a limited number of onsets, only part of the music events in this rhythmic hierarchy is observed (annotated as open circles) and the other part (annotated as filled circles) has to be interpolated from adjacent observations. From Fig. 4 it is clear that the observed events are concentrated at one rhythmic layer. This layer is the most salient rhythmic layer in our analysis. For most occasions this layer overlaps with the 'tactus' layer or foot tapping rate, which is most significant in a music cognition perspective [16]. The timing value of this layer is denoted as $r_1 \dots r_P$, This beat pattern forms a timing descriptor in the macro-analysis level as:

$$F_T(p) = \frac{P \cdot r_p}{\sum_{p=1}^{P} r_p} \qquad (7)$$

In practical implementation $F_T(p)$ could be smoothed by convolving a smoothing kernel [13]. The expressive timing values of the music events at other rhythmic layer are then sampled from this smoothed variation curve. The micro-analysis level timing is obtained by comparing the audio onset locations and an interpolated mechanical timing.

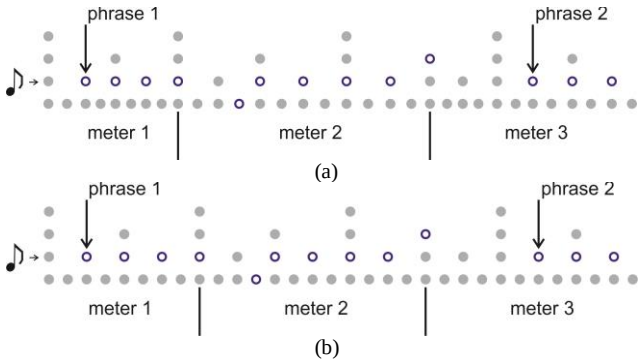

Fig. 3. A hierarchical rhythmic analysis result. The rhythmic grid from a expressive performance (a) is compared with its mechanical version (b) The comparison is at a music phrase level. (a) shows a dramatic expectation-relaxation pattern as compression and expansion of performance timeline.

IV. Cognitive Relevance of Semantic Musical Features

A. Cognitive Relevance

Here we present a simple musical example to illustrate the concept of a simple formula that can provide the cognitive relevance of our proposed semantic musical features. In this example we first program a visualization interface based on music expressive features. Specifically this visualization interface produces different colors for different music timbres. The user watching this visualization while listening to the music would notice something after a while. He/she finds that this visualization is not mechanically following the music score but intuitively "feels" the relevance (*perceptual relevance*). The user feels the visualization is better than a mechanical score inference-based interface because the connections between performance timbre and music score is not direct (Actually the music timbre is an interpretation produced by the musician/performer. This interpretation is related to the musical score in a complex manner [19] but an audience could easily discern its relevance).

In this simple visualization example, the cognitive relevance is generated from (1) the plausible interpretation produced by the musician (2) the audience's music understanding. As a successful communication process, we deliver the right content (control data generated from semantic musical structure) to the right audience (who appreciate these semantic music structures) and produced a musically relevant perceptual experience. Alternatively, we could use the feature sequence generated by manual reductive music analysis to control the visualization interface. A certain level of musical relevance is then expected because the reductive music analysis is "semantic data" generated from music theorists (or automatic analysis that simulate their procedures) [1,17].

In this visualization example the musicians who generate the music understand the music at a "semantic" level. By

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

implementing content annotation labels related to these semantic music features, the user interacting with these content labels will feel that a human musician is in control of the visualization interface. This human-computer interaction based on these annotation labels becomes "interesting" because we are indirectly communicating with human intelligence. The semantic music patterns we proposed serve as a "medium" to distribute human-based music intelligence since the human-computer interactions based on these features become essentially human (the user) to human (the musicians who produced these patterns) interaction.

A system based on semantic music features possesses dramatic advantages compared to a system that employs score-based inference. For example, a score-based or music transcription based system [20] would just turn the screen to certain color for a certain pattern of concurring music notes. If the mechanical link between the visualization and the music score is too simplistic, the user may easily uncover the "formula". In such a mechanical process and the user could quickly become bored discovering the visualization "formula".

B. Perceptual Tolerance

The goal of multimedia content annotation is to provide a musically-plausible user experience. The perceptual process of our audience provides further interpretation of the information. This interpretation process provides an additional degree of perceptual tolerance for media content. Here are some examples of simple computer algorithms can give humans the impression of a high degree of cognitive relevance, or even a high degree of "machine intelligence".

- Joseph Weizenbaum's 1966 conversation program, ELIZA, has consistently fooled people into thinking it is human. ELIZA actually employs a simple natural language processing algorithm that substitutes a part of the sentence with the user input to produce its parts in conversation. The program "understood" nothing about the conversation. [25:*pp.38*]
- The big news in 1997 was the defeat of the world chess champion, Garry Kasparov, by IBM's "Deep Blue" chess-playing computer. "Deep Blue" performs heuristic searches to evaluate a chess position. Several websites mentioned that, after his loss, Kasparov said that he sometimes saw deep intelligence and creativity in the machine's moves. [25:*pp. 481-482*]
- The 1979 electronic game Pac-Man by Midway Games West, Inc. only has simple control routines for the game characters. Nevertheless human game players reported strategies of game characters: "The four of them are programmed to set a trap, with "Blinky" leading the player into an ambush where the other three lie in wait." [26]

The perceptual tolerance demonstrated by these examples arises from the interpretation of the audience in the human-computer exchanges. For these three examples the computer algorithms possess no intelligence but rather, when he human obtains the processing results from these algorithms they infer the presence of an "intelligent agent". This interpretation provides humans with a generous dose of tolerance in the perception of our proposed semantic music features. The Eliza example is particularly important because the program is merely a mirror that "reflects" the intelligence of the user. Such a perceptual process is ubiquitous in music and other types of representational arts, where we emphasize our perceptions and emotions over instrumental observations and objective facts.

V. SUMMARY

In this paper we present musical features based on semantic music analysis for multimedia annotation applications. The resulting features convey human musical understanding and provide strong interpretations of music data. As intelligent human-music interfaces, these semantic musical features find important multimedia applications.

Several possible extensions of the proposed framework include: (1) encoding extended semantic feature dimensions such human emotion, human brain response, performer gesture (and its recognition), and visual features extracted from video; (2) forming a standardized format for music semantic annotations.

REFERENCES

[1] A. D. Patel, Music, *Language, and the Brain*, Oxford University Press, 2010, pp. 299-352.
[2] R. Rove, *Interactive Music Systems: Machine Listening and Composing*, The MIT Press: Cambridge, MA, 1994, pp. 12-35.
[3] S. Tsuruta, M. Fujimoto, M. Mizuno, Y. Takashima, "Personal computer-music system-song transcription and its application", *IEEE Transactions on Consumer Electronics, Vol. 34 , Issue 3*, 1998, pp. 819 - 823.
[4] J. Bae, H. Joo, and L. Song, "A spectrum-based searching technique for the most favorable section of digital music", *IEEE Transactions on Consumer Electronics, Vol. 55 , Issue 4*, 2009, pp. 2122 - 2126.
[5] X.Zhu, Y. Y. Shi,H. G. Kim, K. W. Eom, "An integrated music recommendation system", *IEEE Transactions on Consumer Electronics, Vol.52 , Issue: 3*, 2006, pp 917 - 925.
[6] H. Rosas, R. M. Kil, S. Han, "Automatic media data rating based on class probability output networks", *IEEE Transactions on Consumer Electronics, Vol. 56 , Issue 4,* 2010, pp 2296 - 2302.
[7] A. Cadwallader, D. Gagné, *Analysis of Tonal Music: A Schenkerian Approach, 3rd ed.*, Oxford University Press: New York, NY, 2010.
[8] F. Lerdahl, R. Jackendoffs, *A Generative Theory of Tonal Music*, The MIT Press: Cambridge, MA, 1983.
[9] A. Marsden, "Schenkerian analysis by computer: a proof of concept," *Journal of New Music Research*, v.39, No. 3, pp 269-289, 2010.
[10] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing A Generative Theory of Tonal Music", *Journal of New Music Research*, 35, 2006, pp. 249–277.
[11] M. Hamanaka, K. Hirata, and S. Tojo, "FATTA: Full automatic time-span tree analyzer", *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, August 2007, pp.153–156.
[12] A. Kirke and E. R. Miranda, "A survey of computer systems for expressive music performance," *ACM Comput. Surv.*, Vol. 42, No. 1, pp. 1-41, 2009.
[13] G. Widmer; W. Goebl, "Computational Models of Expressive Music Performance: The State Of The Art", *Journal of New Music Research*, Vol. 33, pp. 13, 2004.
[14] G. Ren, J. Lundberg, G. Bocko, D. Headlam, M. F. Bocko: "What Makes Music Musical? A Framework for Extracting Performance

Doi: 10.7321/jscse.v3.n3.36 e-ISSN: 2251-7545

Expression and Emotion in Musical Sound," *Proceedings of the IEEE Digital Signal Processing Workshop*, pp. 301–306, 2011.
[15] W. Goebl, S. Dixon, G. DePoli, A. Friberg, R. Bresin, G. Widmer, 'Sense' in Expressive Music Performance: Data Acquisition, Computational Studies, and Models, In P. Polotti & D. Rocchesso (Eds.), Sound to Sense – Sense to Sound: A State of the Art in Sound and Music Computing (pp. 195–242). Berlin: Logos, 2008.
[16] H. Shenker, *Free Composition*, Pendragon Pree, 2001.
[17] L. B. Meyer, *Emotion and Meaning in Music*, University Of Chicago Press, 1961, pp.1-42.
[18] M.E. Bonds, *A History of Music in Western Culture, 3rd ed*., Prentice Hall, 2009.
[19] S. Gordon, *Mastering the Art of Performance: A Primer for Musicians*, New York: Oxford University Press, 2010, pp. 25-32.
[20] A. Klapuri, and M. Davy, *Signal Processing Methods for Music Transcription*, Springer, New York, NY, 2006.
[21] M. Müller, *Information Retrieval for Music and Motion*, Springer, New York, NY, 2007.
[22] F. Auger, F. Hlawatsch, *Time-Frequency Analysis*, Wiley-ISTE, Hoboken, NJ, 2008.
[23] H. Pang, D. Yoon: "Automatic Detection of Vibrato in Monophonic Music", *Pattern Recognition*, Vol. 38, pp.1135 – 1138, 2005.
[24] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler: "A Tutorial on Onset Detection in Music Signals," *IEEE Trans. on Speech and Audio Processing*, vol. 13, pp. 1035-1047, 2005.
[25] N. J. Nilsson, *The Quest for Artificial Intelligence: A History of Ideas and Achievements*, New York: Cambridge University Press, 2010
[26] I. Millington, and J. Funge, *Artificial Intelligence for Games, 2nd Edition*, Burlington, MA: Morgan Kaufmann, 2009, pp. 19-21.