KNOWLEDGE-BASED MINING OF PATTERNS AND STRUCTURE OF SYMBOLIC MUSIC FILES

Frank Seifert
Department of Computer Science, University of Technology, Str. d. Nationen 62, 09107 Chemnitz, Germany

Keywords: Mining Music Data, Song Detection, Pattern Recognition, Knowledge-Based Music Modelling.

Abstract: To date, there are no systems that can identify symbolic music in a generic way. That is, it should be possible to associate the countless potential occurrences of a certain song with at least one generic description. The contribution of this paper is twofold: First, we sketch a generic model for music representation. Second, we develop a framework that correlates free symbolic piano performances with such a knowledge base. Based on detected pattern instances, the framework generates hypotheses for higher-level structures and evaluates them continuously. Thus, one or more hypotheses about the identity of such a music performance should be delivered and serve as a starting point for further processing stages. Finally, the framework is tested on a database of symbolic piano music.

1 INTRODUCTION

Imagine you visit a hotel lounge with a piano-player playing. If you know the performed song in general, you will probably recognize it, although you may have never heard the song that way before. Commonly, such musicians play by ear, that is, without any score. Thus, an interpretation of the very same song often differs substantially from one performance to another. How can we identify such music automatically? First, we want to give an overview about related works and how they may fit to our aim:

So-called audio fingerprinting systems, e.g. (Mohri, 2007), represent a large corpus of recent music identification research. These systems abstract from the technical realization of the very same song by reducing the representation to salient acoustical properties. However, music recognition is restricted to already recorded and preprocessed songs. Even slight alterations such as common deviations of live performances are not detected.

More recently, audio fingerprinting is generalized by proposals, which attempt to correlate a musical score with possible recordings, e.g. (Montecchio, 2009). This research is usually denoted as music synchronization. These approaches are strongly tied to the score representation. Mainly, this holds only for classical music, which is often interpreted newly based on only one existing score. Due to this dependence, higher musical deviations or structural modifications are not possible with these systems.

Some proposals attempt to derive the structure of a musical piece by finding self-similar segments, e.g. (Bello, 2009). However, similar music segments can show a wide range of modifications in a variety of musical parameters, such as instrumentation, tempo, dynamics, ornamentation, musical patterns, and even recording conditions. These approaches are based on a marginalization of those changes. Thus, they work only best for musical parts that do not change too much, such as in classical music. Another problem is the extraction of high-level features to describe musical content adequately. One approach that is based on high-level features uses chroma indexing for music identification (Miotto, 2008). However, using only one feature is not sufficient to describe the many facets of musical changes.

The key idea of this paper is to combine both ideas of music synchronization and structural analysis for music identification. Both concepts have to be extended in order to handle those facets of music that are usually altered when played by a bar pianist, such as structural modifications, which cover arbitrary repetitions of certain parts like refrains, removal of parts like strophes, addition of free intros, intermezzi, or endings. Parts of music can even be combined with different partial or even full musical entities. Such medleys are played very

Seifert F..
KNOWLEDGE-BASED MINING OF PATTERNS AND STRUCTURE OF SYMBOLIC MUSIC FILES .
DOI: 10.5220/0003118103580363
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 358-363
ISBN: 978-989-8425-28-7

often at piano bars. But most challenging are alterations of musical content: The same piece of music can be played at different styles, e.g. bossa, swing, or even as a waltz, which usually results in considerable rhythmic changes. Music can be reharmonized to make it sound more interesting. For the same reason, parts of music are often transposed. Finally, even melody can be changed more or less by alteration, addition or removal of tones.

2 GENERIC REPRESENTATION

Based on inherent musical knowledge a bar pianist knows how to play and how to accompany a melody in a certain style. To simulate the other way around we have to model music in a generic way that allows for a high degree of variability. Basis of our model is an elementary music description - the so-called lead sheet. A lead sheet contains the melody of a musical piece and defines harmonic context through abstract chord symbols. A skilled pianist knows, how to elaborate this information and present the music in a certain style, for instance as samba. We decompose such a representation into a sequence of patterns with musical meaning, so-called motifs. An very simple example for the well-known spiritual "When the saints go marching in" is given in Figure 1:

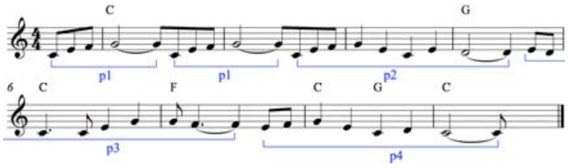

Figure 1: Patterns of "When the saints".

The second pattern is a repetition of the first one. General rules for identifying patterns involve expert knowledge in the musical domain and are beyond the scope of this paper. For computational approaches see (Pearce, 2008).

Adjacent notes are difference-coded in time and space, because pattern can be transposed or stretched in time.

2.1 Structural Coding

The serialization of patterns within a structure is done by delta-coding again. This time we use the difference between the first notes of adjacent patterns in space and time. Applied on "When the saints" we get the following structure s_1:

s_1: {p_1 <0, 1.0> p_1 <0, 1.0> p_2 <4, 2.125> p_3 <0, 2.0> p_4}

The first value after each pattern represents the spatial difference in semitones, the second value denotes the time difference. This note-length-notation codes a quarter note as 0.25, an eighth as 0.125 and 2.125 means two wholes plus an eighth.

Structure s_1 describes only the melodic content of "When the saints". Within our model, an entire piece of music is represented by a special template. Such a template is usually associated with one or more melodic structures, a harmony representation and additional parameters.

Applied on "When the saints" we get a generic template t_1 for that song, which consists of the melodic structure s_1 and a non-detailed chords representation c_1. Model components are best illustrated by a tree-like representation. Edges are marked by delta-shifts in time and space.

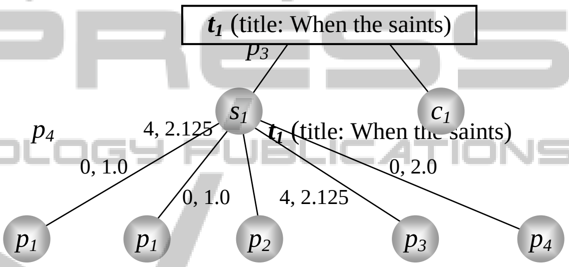

Figure 2: Generic template of "When the saints".

The melodic content of Figure 1 is usually associated with "When the saints". Actually, this is only the refrain. The strophe is shown in Figure 3:

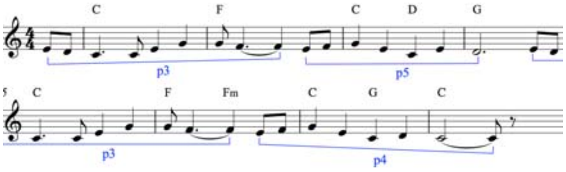

Figure 3: Strophe and patterns "When the saints".

This strophe could be modeled as structure s_2:

s_2: {p_3 <0, 2.0> p_5 <0, 2.0> p_3 <0, 2.0> p_6}

As you can see, components p_3 and p_4 are reused again. Now, we can create another structure s_3, which consists of s_2 and s_1. Then, we create a new template t_2, which integrates s_3 and a new harmony structure c_2. The resulting template t_2 *is* depicted in simplified form in Figure 4. Finally, we have got two generic representations of the song, with and without strophe. This should reflect real world scenarios, where both representations can be valid. By marking structural edges with a meaning tag (not visualized), we can express common musical meaning of structures, such as "refrain".

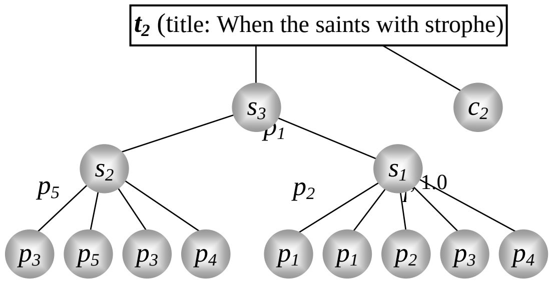

Figure 4: Template of “When the saints“ with strophe.

3 RECOGNITION

Modeling music by a hierarchy of structures, which represents a sequence of generic music patterns at the lowest level, is the foundation of the music identification framework. If we want to identify an arbitrary song, we have to correlate it with a corresponding generic music template. Primarily, this process consists of two steps: First, we have to identify all existing pattern instances. Second, we have to structure these pattern instances and hierarchize the determined structure instances recursively. During this step repetitions of structure instances should be automatically detected and handled adequately. Ideally, after finishing step two an instance should be correlated with a top-level structure and therefore, can be associated with a generic music template. Let us examine these steps in more detail:

3.1 Recognition of Pattern Instances

For each tone of a music file we match each pattern of our database and try to find all occurrences of that pattern within a search-range. This range is determined by the estimated beat of the corresponding region plus a defined tolerance range. Although there are some heuristics for most likely melody placements, it is necessary to consider each occurrence of a pattern at this step. If our brain knows a certain pattern, it does not have to be very salient to recognize it (Dowling, 1986). Of course, this step will usually result in a vast amount of false hits.

During the pattern recognition process, which is purely bottom-up, we cannot decide, whether a detected pattern instance is really perceived. Unfortunately, the denser a sound mixture is, the more patterns will be detected and most of them will be irrelevant. However, we can use some heuristics and rate a pattern instance by perceptual salience and similarity with the pattern template. Some essential criteria are presented as follows:

Patterns can be stretched or expanded in time. To compensate for the varying length we normalize a pattern instance with its length. Then, we evaluate both absolute positions of tones within a pattern instance and relations of adjacent tones.

Let p be a detected pattern instance (a sequence of tones) and P the corresponding pattern template, p_i and P_i are the i-th tone of p and P, n is the number of pattern tones. ${}^{time}p_i$ and ${}^{time}P_i$ are timestamps of p_i and P_i. The rating of positions r_{pos} is defined as

$$r_{pos} = \max\left(1 - \frac{1}{n-2}\sum_{i=2}^{n-1}\left|\frac{{}^{time}p_i - {}^{time}p_1}{{}^{time}p_n - {}^{time}p_1} - \frac{{}^{time}P_i - {}^{time}P_1}{{}^{time}P_n - {}^{time}P_1}\right|, 0\right) \qquad (1)$$

Each rating is normalized within range [0, 1]. A value 1 means the best possible rating. Analogously, we get the rating of relations r_{rel} by evaluating the relation of time differences between adjacent notes:

$$r_{rel} = \max\left(1 - \frac{1}{n-2}\sum_{i=2}^{n-1}\left|\frac{{}^{time}p_i - {}^{time}p_{i-1}}{{}^{time}p_{i-1} - {}^{time}p_{i-2}} - \frac{{}^{time}P_i - {}^{time}P_{i-1}}{{}^{time}P_{i-1} - {}^{time}P_{i-2}}\right|, 0\right) \qquad (2)$$

Both ratings can only be evaluated while n > 2.

How salient is a pattern instance compared with simultaneous events? On the one hand, we check, whether the events of a pattern instance are more salient than simultaneous events and on the other hand, we assess the spatial position of a pattern instance. Very often, a melodic pattern forms the top voice. Generally, it should be isolated from surrounding events to become perceived dominantly. That is, top or lowest voice is separated better than a middle voice, but harmonic simultaneous events contribute more evidence for such a hypothesis.

Let t denote a sequence of tones in same time range of p, t_j is the j-th tone of t; m is number of tones in t. For t and p $t)p=|$ has to be valid. ${}^{vel}p_i$ and ${}^{vel}t_i$ are the velocity values of p_i and t_i. Rating of salience r_{sal} is defined as

$$r_{sal} = \max\left(\min\left(\log_2\left(\frac{1}{n}\sum_{i=1}^{n}{}^{vel}p_i\right) - \log_2\left(\frac{1}{n}\sum_{i=1}^{n}{}^{vel}t_i\right), 1\right), 0\right)$$

Let *top* be the number of tones in p, which form the top voice and *low* the number of components of the lowest voice. Rating of place r_{place} is then defined as

$$r_{place} = \frac{2top + low}{3n} \qquad (3)$$

With this definition, the upper voice is better rated than the lower voice and the highest rating is only possible with monophonic pattern instances.

An important criterium for assessing the bottom-up quality of a pattern instance is the number of events *num*

num_{events} that interfere with the perception of the ideal template of an instance. That is for two adjacent tones, how many perceivable events are positioned around an imaginary line between both tones. Due to space constraints the algorithm is not detailed here. We suppose this number as given and define the corresponding rating r_{events} as follows:

$$r_{events} = \max\left(1.0 - \frac{num_{events}}{n}, 0\right) \qquad (4)$$

These single ratings are individually weighted and combined as the pattern instance rating r_p:

$$r_p = \frac{w_{pos}r_{pos} + w_{rel}r_{rel} + w_{sal}r_{sal} + w_{place}r_{place}}{w_{pos} + w_{rel} + w_{sal} + w_{place}} \qquad (5)$$

Concluding this step, we have got a set of potential pattern instances, which are rated according to the likeliness of being perceived. Nevertheless, patterns that show a worse rating than competing ones, could be perceived due to context. This decision can be made at the next stage at the earliest.

3.2 Predicting Higher-level Structures

For a homogeneous definability patterns respective pattern instances are considered structures and structure instances of level 0 by now.

For each detected structure instance, we know, in which higher-level constructs it is contained. Thus, we try to find evidence for these higher-level structures by attempting to detect instances of all remaining components. Within this step, we get more or less complete sequences of structure instances that belong to a hypothesized structure. Again, we have to rate these sequences of components regarding completeness, perceptual salience and similarity with the predicted structure.

Structural integrity is characterized by the relative onset and the length of components within a structure. Again, we compare sequences of detected component instances with the predicted structure.

Let *s* be a predicted structure instance and *S* the corresponding structure. Then, s_i and S_i denote the *i*-th component of *s* respective *S*, |*s*| is the number of detected components in *s* and |*S*| the number of component structures in *S*. ${}^{time}s_i$ and ${}^{time}S_i$ are the timestamps of s_i and S_i; *b* denotes the *b*-th component s_b, which is the foundation for hypothesis formation. Onset rating r_{onset} compares the relative start times of components between instance and template. Roughly, it corresponds to r_{pos}. This time, however, component instances may not be complete. For missing components, bounds are extrapolated on basis of s_b. Otherwise, they are ignored and not evaluated:

$$r_{onset} = \max\left(1 - \frac{1}{|s|}\sum_{i=1}^{|S|}\left|\frac{{}^{time}s_i - {}^{time}s_b}{{}^{time}s_{|S|} - {}^{time}s_1} - \frac{{}^{time}S_i - {}^{time}S_b}{{}^{time}S_{|S|} - {}^{time}S_1}\right|, 0\right) \qquad (6)$$

The length rating r_{len} compares the relative lengths of components. Components may have a correct onset but last too short or too long. This measure differentiates the onset rating further. To streamline the formula, we define

$$\begin{gathered} {}^{time}\Delta s = {}^{time}s_{|S|} - {}^{time}s_1 \text{ and} \\ {}^{time}\Delta S = {}^{time}S_{|S|} - {}^{time}S_1 \\ r_{len} = \max\left(1 - \frac{1}{|s|}\sum_{i=1}^{|S|}\left|\frac{{}^{time}\Delta s_i}{{}^{time}\Delta s} - \frac{{}^{time}\Delta S_i}{{}^{time}\Delta S}\right|, 0\right) \end{gathered} \qquad (7)$$

.

The transposition rating r_{trans} evaluates the spatial coherence of detected components. Due to being gestalts, a component can be transposed. Usually, transpositions of a whole tone or a semi tone happen. With many popular songs, the final strophe or refrain is transposed one tone up to create more tension. Besides, many endings repeat final patterns several times while transposing them up and down. We define the deviation regarding the relative transposition of components as

$${}^{tone}\Delta s_i = \left|\left({}^{tone}s_b - {}^{tone}s_i\right) - \left({}^{tone}S_b - {}^{tone}S_i\right)\right|$$

The transposition rating r_{trans} is then defined as

$$r_{trans} = \frac{1}{|s|}\sum_{i=1}^{|S|}\begin{cases} 1.0 & {}^{tone}\Delta s_i = 0 \\ 0.6 & {}^{tone}\Delta s_i \equiv 0 \bmod 12 \\ 0.2 & {}^{tone}\Delta s_i < 3 \\ 0.0 & else \end{cases} \qquad (8)$$

We have to combine these distance measures to evaluate the overall validity of a predicted structure. This rating can not be better than the average rating of its components. Therefore, we have to weight the total structural rating r_s with the average rating of each individual component rating r_i.

$$r_s = \frac{1}{3}\left(r_{onset} + r_{len} + r_{trans}\right)\frac{1}{|S|}\sum_{i=1}^{|S|} r_i \qquad (9)$$

If r_s exceeds a given limit, we can instantiate a structure hypothesis. If a component does not exist we instantiate it nevertheless and mark it as "hypothesized". In further processing stages such segments can be examined deeper and multidimensional similarity can be computed.

3.3 Handling Repetitions

An important aspect of music is repetition. The entire piece of music can be repeated as a whole. However, more challenging are arbitrary repetitions of partial structures such as refrains, or even parts at a smaller scale. Our model defines a parameter for the general repeatability of a structure. Structures with a meaning such as "strophe" or "refrain" are automatically assigned to a probability value of 1. For structures without a special meaning, this value can be in the range of 0 to 1. Structures at the lowest level, i.e. patterns should usually have a repetition probability of 0.

If this probability value is greater than 0 for a detected instance, we have to collect possible instances of the same structure, which may follow within a defined tolerance range. If we find one or more repetitions of a component, we create a virtual instance that integrates those components. Virtual repetition instances are evaluated like a non-repeating component. Thus, if musical structures are played more than actually intended, this situation will be automatically resolved by now.

For each of the resulting structure instances, we repeat this process and predict of the next level, try to find evidence for their components, rate them, and finally instantiate or reject them. That is, we repeat steps 3.2 and 3.3 until the highest level of the template database is reached. Ideally, this process should finally result in one or more sequential structure instances, which cover the entire music.

4 RESULTS & CONCLUSIONS

At the moment, our template database contains 150 songs, most of them jazz standards, the remaining are folk songs, christmas carols, pop and rock songs. For these songs, we collected 352 midi-files of piano performances. The midi-files have been either picked up from the Internet or were specifically recorded by some persons, who work either as music lecturers or play at piano bars occasionally. Besides song selection, no restrictions were made. The music contributors should play given songs the "usual way", which means that normal listeners should be able to recognize the song. However, free improvisations of a theme were possible and performed frequently, indeed. Let us demonstrate the performance of our framework with some examples:

Figure 5 shows the results of recognizing an instance of "Yesterday" by The Beatles. Depicted are top-level structure instances and their components recursively. The corresponding melody notes are highlighted by green dots. The specific rating of structure instances is visualized by color intensity.

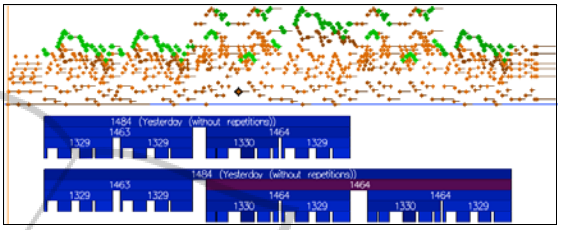
1484 (Yesterday (without repetitions))
1463
1464
1329
1329
1330
1329
1484 (Yesterday (without repetitions))
1463
1464
1329
1329
1464
1330
1329
1464
1330
1329

Figure 5: Recognition of "Yesterday".

In this particular case, all patterns could be detected and meaningfully structured. The uncorrelated regions at the begin and at the end correspond to a free introduction and a free ending. Two special features should be highlighted: First, our template of "Yesterday" is modeled in form AABA, which is detected in the upper instance 1484. However, within the midi-file the last BA section is repeated, which results in the form AABABA. The framework automatically detects a repetition of substructure 1464 and derives an additional instance 1484 of "Yesterday" with a repeated refrain. Second, within the refrain, the pianist sometimes transposes melodic parts one octave up or down. Without the model-inherent decomposition of a melody into structures of elementary patterns this refrain had not been detected at all.

Figure 6 shows an instance of the lullaby "Guten Abend, gut' Nacht" by Johannes Brahms, which after free introduction is played twice:

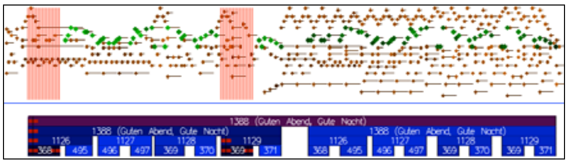
1388 (Guten Abend, Gute Nacht)
1388 (Guten Abend, Gute Nacht)
1126
1127
1128
1129
368
495
496
497
369
370
369
371
1388 (Guten Abend, Gute Nacht)
1126
1127
1128
1129
368
495
496
497
369
370
369
371

Figure 6: Recognition of a Brahms' lullaby.

Again, our framework automatically detects the repetition and creates an additional integrating instance. However, more interesting are the hatched regions, which correspond to pattern instances 368 and 369 respectively. For these patterns, no direct evidence could be found. Thus, they are hypothesized based on context. Our framework provides these regions and additional information, such as predicted bounds or beat positions to further processing stages. In the second pass of the lullaby, the melody is located between upper and lower tones and is not easy to recognize for human listeners. The framework detects this structure even without the need of creating hypotheses for any parts.

How can we evaluate the overall performance of our hypothesis generator? Due to the high degree of possible musical alterations, an adequate evaluation is no easy task. How can we rate improvisations? How should we rate missing repetitions of an already detected structure? How can medleys of refrains or partially detected structures be evaluated? We decided to consider only those regions of a music file, for which sequential structures should be found ideally. Then, we count the number of really detected structures, which have to be normalized by a weight regarding its duration within an instance. Additionally, each predicted or detected structure has to be examined regarding correct bounds. Finally, we get a percentage of correct coverage.

This coverage-measure shows some advantages: Parts of music, which cannot be detected inherently, such as free intros, intermezzi, improvisations, or endings do not affect the rating. If a repetitive or composed structure cannot be correlated entirely but one or more of its higher-level components are detected, that structure will not be rejected. That is, each example from Figures 6 and 7 would get coverage of 1.0, which means that all detectable sections have been recognized correctly. Of course, each covered region gets an additional similarity rating as illustrated in Section 3.2.

To automatize the evaluation of our midi-file collection, we manually identified all contained templates or high-level structures and their ranges. This information has been added as metadata to the database. For all 352 test files we got an average coverage of 0.55. This result is characterized by a high deviation. A lot of music had coverage of 1.0 but many files showed no coverage at all:

Table 1: Coverage of test files.

Kind of coverage	**Number of songs**	**Percent age**	**Average coverage**
Full coverage	119	34 %	1.0
Partial coverage	143	41 %	0.52
No coverage	90	25 %	0.0
Total	**352**	**100 %**	**0.55**

Due to the high degree of melody-alteration especially jazz piano music showed a bad performance. At the moment, our predictive algorithms are based on exact pattern matching techniques. Therefore, the groundwork for successful creation of hypotheses was insufficient in some cases. For most of these unrecognized files, we could manually adjust some accuracy parameters and increase the recall at the expense of precision.

To get more results, the overall need of computation time and memory grows exponentially in most cases. It would be pointless to broaden the search at this development stage. The quality of the structure prediction will most likely improve if hypothesized components could be evaluated by future similarity processing stages. At this time we parameterized our system by optimizing the tradeoff between computation time and recall while getting precision as high as possible. Indeed, for top-level prediction, precision equals one for the entire database.

Further work should complete the framework conceptually first. That is, integrating similarity measures for chords, chord progressions and melodic alterations into the framework. Next, to improve pattern recognition, recent matching techniques should be extended to “fuzzy” matching. Using a more contour-oriented representation of melody would compensate slight variations such as altered or missing tones, e.g. playing a motif in minor instead of major. Finally, by implementing symbolic extraction algorithms, the framework should be able to guide even an audio identification process.

REFERENCES

Bello, J. P., 2009. Grouping Recorded Music By Structural Similarity. *In Proceedings of ISMIR*, Kobe, Japan.

Dowling, W. J., Harwood, D. L., 1986. *Music Cognition*, Academic Press.

Miotto, R., Orio, N., 2008. A Music Identification System Based on Chroma Indexing and Statistical Modeling. In *Proceedings of ISMIR*, Philadelphia, USA.

Mohri, M., Moreno, P., Weinstein, E., 2007. Robust Music Identification, Detection, and Analysis. In *Proceedings of ISMIR*, Vienna, Austria.

Montecchio, N., Orio, N., 2009. A Discrete Filter Bank Approach to Audio to Score Matching for Polyphonic Music. In *Proceedings of ISMIR*, Kobe, Japan.

Pearce, M. T., Müllensiefen, D., Wiggins, G. A., 2008. A Comparison of Statistical and Rule-Based Models of Melodic Segmentation. In *Proceedings of ISMIR*, Philadelphia, USA.