Towards a Computational Model of Melody Identification in Polyphonic Music

Søren Tjagvad Madsen
Austrian Research Institute for
Artificial Intelligence, Vienna
soren.madsen@ofai.at

Gerhard Widmer
Department of Computational Perception
Johannes Kepler University, Linz
gerhard.widmer@jku.at

Abstract

This paper presents first steps towards a simple, robust computational model of automatic melody identification. Based on results from music psychology that indicate a relationship between melodic complexity and a listener's attention, we postulate a relationship between musical complexity and the probability of a musical line to be perceived as the melody. We introduce a simple measure of melodic complexity, present an algorithm for predicting the most likely melody note at any point in a piece, and show experimentally that this simple approach works surprisingly well in rather complex music.

1 Introduction

Melody is a central dimension in almost all music. Human listeners are very effective in (unconsciously) picking out those notes in a – possibly complex – multi-voice piece that constitute the melodic line. Melody is also an important aspect in music-related computer applications, for instance, in Music Information Retrieval (e.g., in music databases that offer retrieval by melodic motifs [Weyde and Datzko, 2005] or Query by Humming [Birmingham *et al.*, 2006]).

It is not easy to unequivocally define the concept of 'melody'. In a sense, the melody is the most prominent line in a polyphonic (multi-voice) piece of music. Although in Western music, the melody is often found among the higher notes, this is not always the case. Also, even in compositions that are explicitly structured into individual lines by the composer (such as, e.g., orchestral pieces consisting of monophonic instrument voices), the melody is by no means always represented by (or 'appearing in') the same line throughout a piece. In a way, which notes constitute the melody is defined by where the listeners perceive the most interesting things to be going on in the music, or what they sense to be the most coherent path through the complex interweaving of musical lines. Thus, though our experience tells us that hearing the melody is a rather simple and intuitive task for humans, it is by no means a simple task to be formalised in a machine.

In popular music, it is sometimes assumed that the melody does not change between the instruments present. Some work has been done on predicting which one of the tracks in a MIDI file contains the 'main melody' [Rizo *et al.*, 2006; Friberg and Ahlbäck, 2006]. In both cases a statistical approach was taken – learning properties of melodic and non-melodic tracks. Ideas for converting polyphonic tracks into melodically meaningful monophonic sequences have also been proposed [Uitdenbogerd and Zobel, 1998].

This paper presents first steps towards a simple, robust computational model of automatic melody *note* identification. Based on results from musicology and music psychology that indicate a relationship between melodic complexity and a listener's attention, we postulate that the notes making up the melody line may be identified by calculating and integrating some measures of perceived complexity over time. We will introduce a simple, straightforward measure of melodic complexity based on entropy, present an algorithm for predicting the most likely melody note at any point in a piece, and show experimentally that this very simple approach works surprisingly well in picking out the melody notes in quite complex polyphonic music. Still, the results are still far below what we can expect humans to achieve, and we will discuss a number of possible extensions of the approach towards a more comprehensive and effective computational model.

2 Complexity and Melody Perception

The basic motivation for our model of melody identification is the observation, which has been made many times in the literature on music cognition, that there seems to be a connection between the complexity of a musical line, and the amount of attention that will be devoted to it on the part of a listener. A voice introducing new or surprising musical material will potentially attract the listener's attention. However, if the new material is constantly repeated, we will pay less and less attention to it and become habituated or accustomed to the stimulus. Less attention is required from the listener and the voice will fall into the background [Snyder, 2000]. The notion of musical surprise is also related to the concept of 'expectation' as it has been put forth in recent music theories [Narmour, 1990; Huron, 2006]. If we assume that the melody is the musical line that commands most attention and presents most new information, it seems natural to investigate melodic complexity measures as a basis for melody detection algorithms.

Indeed, the idea of using information-theoretic complexity measures to characterise aspects of musical development is not at all new. For instance, to cite just two, in [Dubnov *et al.*, 2006], a measure of *Information Rate* computed over a piece of music was shown to correlate in significant ways with familiarity ratings and emotional force response profiles by human human subjects. In [Li and Sleep, 2005] it was shown that kernel-based machine learning methods using a compression-based similarity measure on audio features perform very well in automatic musical genre classification.

In the current paper, we intend to show that the complexity or information content of a sequence of notes may be directly related to the degree to which the note sequence is perceived as being part of the melody. Our approach is to start with a very simple (or simplistic) measure of complexity based only on note-level entropies, in order to first understand the influence of this simple parameter. In the next phases of the project, more complex complexity measures based on pattern compression and top-down heuristics derived from music theory will be added, one by one.

3 A Computational Model

The basic idea of the model consists in calculating a series of complexity values locally, over short-term musical segments, and for each individual voice in a piece.[1] Based on these series of local complexity estimates, the melody is then reconstructed note by note by a simple algorithm (section 3.4).

The information measures will be calculated from the structural core of music alone: a digital representation of the printed *music score*. Though the model works with arbitrary MIDI files which may represent actual performances, it will not consider performance aspects like expressive dynamics, articulation, timbre, etc. These may well contain a lot of useful information for decoding the melody (in fact, expressive music performance is a means used by performers to elucidate the musical structure of a piece [Gabrielsson, 1999]), but we want to exclusively focus on music-structural aspects first, in order not to mix different factors in our investigation.

3.1 The Sliding Window

The algorithm operates by in turn examining a small subset of the notes in the score. A fixed length window (with respect to duration) is slid from left to right over the score. At each step, the window is advanced to where the next 'change' happens so no window will contain exactly the same set of notes, but at the same time all possibilities of placing the window resulting in different content have been examined. The first window starts at time 0. It is then moved ahead in time until the end of the piece. At each step, the next window will begin at whatever of the following two situations occurs first:

1. offset of first ending note in current window
2. onset of next note after current window

In the first case, the change occurs since a note 'leaves' the current window; the next window will begin right after the first occurring offset of a note in the current window. If all

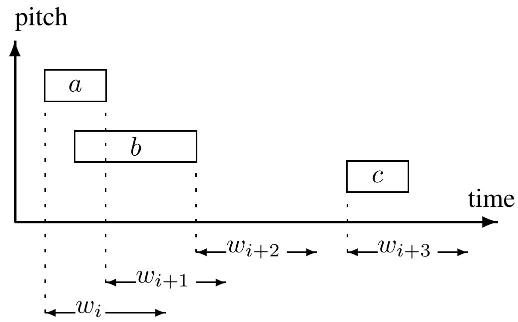

Figure 1: Sliding the window

notes in the current window end at the same time, the next window will thus begin right after, which makes it possible for the window to be empty. Only then will the second of the above cases apply (the change is the entry of the next note).

Figure 1 shows the positions of the window when sliding over three notes a, b, and c. Window w_i contains a and b, w_{i+1} contains only b, w_{i+2} is empty and w_{i+3} contains c.

From the notes belonging to the same voice (instrument) in the window, we calculate a complexity value. We do that for each voice present in the window. The most complex voice is expected to be the one that the listener will focus on in this time period. The complexity measures and the melody note prediction method are explained below.

3.2 Entropy Measures in Musical Dimensions

Shannon's entropy [Shannon, 1948] is a measure of randomness or uncertainty in a signal. If the predictability is high, the entropy is low, and vice versa.

Let X be a discrete random variable on a finite set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with probability distribution $p(x) = Pr(X = x)$. Then the entropy $H(X)$ of X is defined as:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \quad (1)$$

X could for example be the set of MIDI pitch numbers and $p(x)$ would then be the probability (estimated by the frequency) of a certain pitch. In the case that only one type of event (one pitch) is present in the window, that event is highly predictable or not surprising at all, and the entropy is 0. Entropy is maximised when the probability distribution over the present events is uniform.

We are going to calculate entropy of 'features' extracted from the notes in monophonic lines. We will use features related to pitch and duration of the notes. A lot of features are possible: MIDI pitch number, MIDI interval, pitch contour, pitch class, note duration, inter onset interval etc. (cf. [Conklin, 2006]). We have used the following three measures in the model presented here:

1. Pitch class (C): count the occurrences of different *pitch classes* present (the term pitch class is used to refer the 'name' of a note, i.e., the pitch irrespective of the octave, such as C, D, etc.);
2. MIDI Interval (I): count the occurrences of each melodic interval present (e.g., minor second up, major third

[1] We assume that a polyphonic piece has already been 'streamed' into individual voices. That is another non-trivial music analysis problem, but recent work of ours [Madsen and Widmer, 2006] indicates that it can be solved quite effectively, via heuristic search.

down, ...);

3. Note duration (D): count the number of note duration classes present, where note classes are derived by discretisation (a duration is given its own class if it is not within 10% of an existing class).

With each measure we extract events from a given sequence of notes, and calculate entropy from the frequencies of these events (H_C, H_I,H_D).

H_C and H_I are thought to capture opposite cases. H_C will result in high entropy when calculated on notes that form a scale, while H_I will result in low entropy. In situations where there are more different intervals than pitch classes (e.g. an 'alberti bass', or a triad chord played in arpeggio up and down), H_I will produce a higher entropy than H_C.

So far rhythm and pitch are treated separately. We have also included a measure H_{CID} weighting the above three measures: $H_{CID} = \frac{1}{4}(H_C + H_I) + \frac{1}{2}H_D$.

Entropy is also defined for a pair of random variables with joint distribution:

$$H(X, Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log_2[p(x, y)] \qquad (2)$$

We will test two joint entropy measures: Pitch class in relation to duration ($H_{C,D}$) and interval plus duration ($H_{I,D}$). These are expected to be more specific discriminators.

As melodic intervals are meant to be calculated from successive single musical events (rather than between chords), we will apply these measures only to individual lines in the music – instrumental parts. In the few cases where it does happen that a part contains simultaneous notes (e.g., a violin), only the top note is used in the entropy calculation.

3.3 An Alternative: Complexity via Compression

The entropy function is a purely statistical measure related to the frequency of events. No relationships between events is measured – e.g. the events `abcabcabc` and `abcbcacab` will result in the same entropy value. Humans, however, would probably describe the first string as three occurrences of the substring `abc` – we infer *structure*. According to Snyder, we perceive music in the most structured way possible [Snyder, 2000]. To take this into account, complexity measures based on compression could be considered. Music that can be compressed a great deal (in a lossless way) can then be considered less complex than music that cannot be compressed. Methods exist that substitute recurring patterns with a new event, and store the description of that pattern only once, e.g. run-length encoding or LZW compression [Ziv and Lempel, 1977]. This idea has been discussed in several musical application contexts (e.g., in Music Information Retrieval [Li and Sleep, 2005] or in automated music analysis and pattern discovery [Lartillot, 2006]). However, these compression algorithms are not well suited for compressing the short sequences that we are dealing with. Special algorithms for compressing short sequences will be needed.

Shmulevich and Povel [Shmulevich and Povel, 2000] have examined methods for measuring the complexity of short rhythmic patterns which are supposed to repeat infinitely. Tanguiane's measure [Tanguiane, 1993] is based on the idea that a rhythmic pattern can be described as elaborations of simpler patterns. Before turning to these more complex measures and the assumptions they make, we would like to investigate the power of a simple and well-understood measure like entropy.

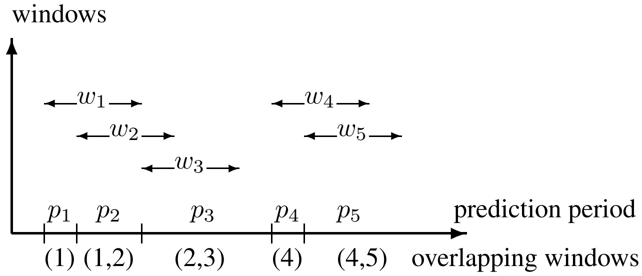

Figure 2: Prediction periods and windows overlapping them.

3.4 Predicting Melody Notes

Given the set of windows described above and an entropy value for each voice present in each window we will predict the notes expected to belong to the melody. We consider in turn the notes in the interval between the starting points of two consecutive windows – the *prediction period*. The prediction period p_i is thus the interval between the beginning of window w_i and the beginning of w_{i+1} (see Figure 2).

We want to mark, in each prediction period, the notes that pertain to the most complex voice. Complexity measures from each window a note appears in will influence the prediction. In a prediction period p_i, we will consider the set $o(p_i)$ of all windows that overlap this period. For p_i the set will contain all windows that have not ended at the start time of p_{i+1}. Figure 2 gives an example of prediction periods and overlapping windows – e.g. $o(p_2)$ is the set $\{w_1, w_2\}$.

For each voice in each window, a complexity value was calculated by the time the windows were calculated. The average complexity value for each voice present in the windows in $o(p_i)$ is calculated. Based on these values, we can now rank the voices according to their average complexity over $o(p_i)$, and a 'winning' voice is found. Every note in w_i (the window where p_i begins) gets its melody attribute set to true if it is part of the winning voice, and to false otherwise.

In each prediction period, exactly the windows overlapping (or partly overlapping) this period are contributing to the final score of each voice. The policy is to mark all notes in the entire window w_i starting at p_i. When w_{i+1} occurs before w_i has ended, the contribution from this new window can change the prediction of w_i from that point in time and onwards. A new prediction period p_{i+1} begins – including all windows now present at this time, and the notes' melody attribute is reset accordingly. So if a note is only contained in a single prediction period, its status will only be set once. A long note contained in a number of prediction periods will finally be judged by the period it last occurs in.

Thus the predicted notes may belong to different voices from period to period – the role as the most prominent voice may change quickly. We are not just predicting an entire voice to be the most interesting, but every note is considered.

In case there are several voices that share the highest complexity value for a given prediction period, the voice with the highest pitch will be predicted (e.g., of two voices playing in octaves, the higher one is chosen).

4 Experiments

4.1 The Musical Test Corpus

To perform experiments we need music which is composed for different parts, and encoded in such a way that each part is separately available. Furthermore, since our complexity measure assumes monophonic music, each voice in the piece should be close to monophonic. This places some restrictions on the experiments we are able to do. A typical piano sonata, lacking the explicit voice annotation (or being one non-monophonic part), will not be an appropriate choice.

Since the model presented here does not take into account any performance aspects but only the musical surface (score), we will use MIDI files generated from the MuseData format (http://www.musedata.org). The durations of the notes in these files are nicely quantised. The model will run on all types of MIDI files, but performed music tends to bias the rhythm complexity. Two pieces of music were chosen to be annotated and used in the experiment:

1. Haydn, F.J.: String quartet No 58 op. 54, No. 2, in C major, 1st movement
2. Mozart, W.A.: Symphony No 40 in G minor (KV 550), 1st movement

This is not an awful lot of data, but since the music will have to be annotated manually, this will have to do for our initial experiments.

4.2 Annotating Melody Notes

Melody annotation is non-trivial and rather tedious task. A trained musicologist was asked to serve as expert annotator. She was given a piano roll representation of the music with ‘clickable’ notes and simple playback options for listening to the MIDI sound. The annotator, who was not aware of the purpose of this task, was instructed to mark notes she would consider as ‘melody’.

The test pieces turned out to have quite distinguishable melodic lines. The annotator solved the task by marking the notes that she was humming during listening to the pieces. Some immediate differences between the ‘true’ concept of melody and some assumptions behind our complexity-based melody prediction model became immediately clear after talking to the annotator:

Our model assumes that there is always one most complex voice present, but a melody was not found to be present at all times. Furthermore, melodic lines were sometimes marked as overlapping, which the model does not allow. The annotator chose not to mark thematic lines that clearly were a (melodic) response in another voice to the melody rather than a continuation of the melody line. Our model might mark both the melody and the response. Another source of error is situations where the melodic line consists of long sustained tones while the accompanying notes are doing all the action. The model will erroneously predict an accompanying voice. (In these situations it is also difficult to tell what a listener will actually listen to).

	Haydn	Mozart
Total number of notes	5832	13428
Number of melody notes	2555	2295
Melody note percentage	43.8 %	17.1 %
Number of voices	4	10
Duration	11:38 min	6:55 min

Table 1: The test data

In any case, the annotations made by the musicologists were taken to be authoritative; they were adopted unchanged and used as ground truth for the evaluation. Table 1 shows some information about the test data.

4.3 Evaluation Method

We can now measure how well the predicted notes correspond to the annotated melody in the score. We express this in terms of recall (R) and precision (P) values [van Rijsbergen, 1979]. Recall is the number of correctly predicted notes (true positives, TP) divided by the total number of notes in the melody. Precision is TP divided by the total number of notes predicted (TP + FP (false positives)). The F-measure combines recall and precision into one value:

$$F(R, P) = 1 - \frac{2RP}{R + P} \tag{3}$$

A high rate of correctly predicted notes will result in high values of recall, precision and F-measure (close to 1.0).

4.4 Results

We performed prediction experiments with four different window sizes (1-4 seconds) and with the six different entropy measures described above. Table 2 shows recall, precision and F-measure values from all experiments with the two evaluation pieces. In addition, the columns marked P show the value of the F-measure achieved by the simple strategy of always picking the *highest* note (see below). The highest values in each row are printed in bold (ignoring the P columns).

The string quartet turned out to be the less complex of the two pieces. This is not much of a surprise. It is easier to discriminate between 4 than 10 voices, and also the compositional structures of the pieces are quite different. Overall the joint pitch class and duration measure $H_{C,D}$ was found to have the greatest predictive power, generally followed by the weighted combination H_{CID}. Pitch class seems to be the single most important measure in the string quartet. In total, the joint measures perform better than the measures based on a single feature.

We can conclude that there is indeed a correlation between melody and complexity in both pieces. The precision value of 0.60 in the best symphony experiment with a resulting F-measure of 0.51 (window size 3 seconds) tells us that 60 % of the predicted notes in the symphony are truly melody notes.

In the string quartet, starting in bar 105 (see Figure 3) the second violin is alternating between a single note and notes from a descending scale, making the voice very attractive (lots of different notes and intervals) while the ‘real melody’

Win		Haydn							Mozart						
		P	H_C	H_I	H_D	H_{CID}	$H_{C,D}$	$H_{I,D}$	P	H_C	H_I	H_D	H_{CID}	$H_{C,D}$	$H_{I,D}$
1 s	R	*0.91*	0.83	0.79	0.67	0.84	**0.87**	0.80	*0.27*	0.36	0.31	0.32	0.41	**0.47**	0.32
	P	*0.96*	0.78	0.73	0.80	**0.81**	**0.81**	0.74	*0.61*	0.41	0.31	0.52	**0.54**	**0.54**	0.32
	F	*0.94*	0.81	0.76	0.73	0.83	**0.84**	0.77	*0.37*	0.38	0.31	0.40	0.47	**0.51**	0.32
2 s	R	*0.91*	0.83	0.80	0.64	0.80	**0.87**	0.81	*0.23*	0.35	0.33	0.27	0.37	**0.45**	0.37
	P	*0.96*	0.81	0.75	0.81	0.82	**0.87**	0.76	*0.58*	0.41	0.36	0.52	0.57	**0.58**	0.41
	F	*0.93*	0.82	0.77	0.71	0.81	**0.87**	0.78	*0.33*	0.38	0.35	0.36	0.45	**0.51**	0.39
3 s	R	*0.91*	0.83	0.79	0.63	0.78	**0.84**	0.80	*0.20*	0.33	0.28	0.24	0.33	**0.45**	0.36
	P	*0.97*	0.83	0.75	0.81	0.83	**0.87**	0.78	*0.53*	0.40	0.31	0.49	0.54	**0.60**	0.42
	F	*0.94*	0.83	0.77	0.71	0.81	**0.85**	0.79	*0.29*	0.36	0.29	0.33	0.41	**0.51**	0.38
4 s	R	*0.93*	**0.84**	0.76	0.59	0.74	0.80	0.79	*0.18*	0.34	0.24	0.19	0.31	**0.44**	0.33
	P	*0.96*	0.84	0.74	0.78	0.84	**0.85**	0.78	*0.50*	0.40	0.28	0.41	0.55	**0.61**	0.41
	F	*0.95*	**0.84**	0.75	0.67	0.78	0.83	0.79	*0.26*	0.37	0.26	0.26	0.40	**0.51**	0.36

Table 2: Recall, precision, and F-measure for melody note predictions.

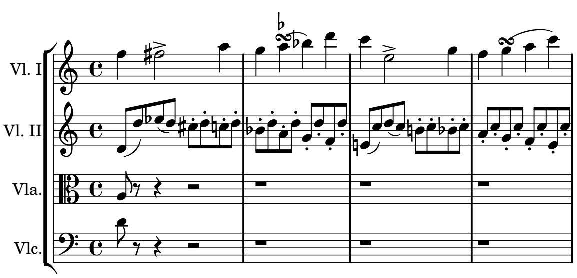

Figure 3: The string quartet, measures 105-108

in the first violin is playing fewer different notes, but has a more varied rhythm. We took a closer look at this passage. Setting the window size to 2 seconds, the measures H_D, H_{CID}, and $H_{C,D}$ recognise the upper voice as melody whereas H_C, H_I, and $H_{I,D}$ suggest the lower. The measures based on intervals are naturally led astray in this case, and the measure based solely on pitch class is also.

The reader may also ask if it would not be more effective to just always predict the *highest* note as the melody note – after all, that is where the melody is most often, at least in relatively simple music. We checked against this baseline. The results are shown in the columns labeled P in Table 2. Indeed, it turns out that in Hadyn string quartet, the simple strategy of always picking the highest note outperforms all of our heuristics – at least in terms of global precision and recall, though not necessarily in terms of the local *musical coherence* of the resulting melodic line.

In the more complex Mozart piece, however, we witness that the highest note prediction strategy was outperformed by most of our methods. In the symphony movement the melody is not on top as often. Also, the melody role is taken by different instruments at different times. Figure 4 shows 6 bars (bar 17-22) from the symphony. In the first three, the flute (top voice) was annotated to be the melody, but the melody then moves to the first violin in bar 19. In bar 21 the first oboe enters on a high-pitched long note – above the melody. Such high-pitched non-melodic notes (doubling chord tones) occur frequently in the movement. The $H_{C,D}$ measure correctly catches the entry of the theme in bar 19, but the bassoon (Fg.) is falsely predicted in bar 17-19 – the wind instruments play very similar lines, but because the bassoon continues a bit further than the others it becomes the most complex.

Figure 4: The symphony, measures 17-22

5 Discussion

In our opinion, the current results, though based on a rather limited test corpus, indicate that it makes sense to consider musical complexity as an important factor in computational models of melody perception. The results of the experiments show that a simple entropy based incremental algorithm can identify parts of melodies in quite complex music. We can also turn the argument around and interpret this as empirical evidence that melody in classical tonal music is indeed partly definable by complexity – a finding that musicology may find interesting.

Clearly, recall and precision values of around 50% (as in the Mozart piece) are not sufficient, neither for justifying our approach as a full model of melody perception, nor for practical applications. The next steps towards improvement are quite clearly mapped out. The notion of musical complexity must be extended to encompass *structural aspects* (i.e., recognisable patterns and their recurrence). We will study various compression-based complexity measures, in particular with respect to their suitability in the context of very short sequences (windows).

A second research direction is dictated by the musicological insight that melody, and generally the perception of what constitutes a coherent musical line, has something to do with *continuity* in certain parametric dimensions (e.g., in terms of intervallic motion). Such Gestalt-like principles are usually difficult to formalise, due to their inherently top-down nature, but local measures of continuity and melodic coherence over short windows should be quite straightforward.

Generally, the more different aspects are integrated in a model of music perception, the more conflicts may arise between them (for instance the obvious conflict between continuity and unpredictability). Good music, in fact, thrives on these kinds of conflicts – divergent cues make music interesting to the listeners, maintaining their attention at a high level. That is precisely why performers are so important in music. Their task is, in part, to disambiguate different possible 'readings' of a piece through their playing. Thus, eventually we will also have to take into account *performance information*, such as dynamics (relative loudness), articulation, etc. But we plan to proceed in stages, evaluating contributions of individual components step by step, in order to arrive at a model that not only works, but tells us something interesting about a human musical ability.

Acknowledgments

This research was supported by the Viennese Science and Technology Fund (WWTF, project CI010) and by the Austrian Federal Ministries of Education, Science and Culture and of Transport, Innovation and Technology.

References

[Birmingham *et al.*, 2006] William Birmingham, Roger Dannenberg, and Bryan Pardo. Query By Humming with the VocalSearch System. *Communications of the ACM*, 49(8):49–52, August 2006.

[Conklin, 2006] Darrell Conklin. Melodic analysis with segment classes. *Machine Learning*, Special Issue on Machine Learning in Music(in press), 2006.

[Dubnov *et al.*, 2006] S. Dubnov, S.McAdams, and R. Reynolds. Structural and affective aspects of music from statistical audio signal analysis. *Journal of the American Society for Information Science and Technology*, 57(11):1526–1536, September 2006.

[Friberg and Ahlbäck, 2006] Anders Friberg and Sven Ahlbäck. A method for recognising the melody in a symbolic polyphonic score (Abstract). In *Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC)*, Bologna, Italy, 2006.

[Gabrielsson, 1999] Alf Gabrielsson. The performance of music. In Diana Deutsch, editor, *The Psychology of Music (2nd Ed.)*, pages 501–602, San Diego, CA, 1999. Academic Press.

[Huron, 2006] David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, Cambridge, Massachusetts, 2006.

[Lartillot, 2006] Olivier Lartillot. A musical pattern discovery system founded on a modeling of listening strategies. *Computer Music Journal*, 28(3):53–67, 2006.

[Li and Sleep, 2005] Ming Li and Ronan Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, London, U.K., 2005.

[Madsen and Widmer, 2006] Søren Tjagvad Madsen and Gerhard Widmer. Separating voices in MIDI. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, Canada, 2006.

[Narmour, 1990] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures*. University of Chicago Press, Chicago, IL, 1990.

[Rizo *et al.*, 2006] David Rizo, Pedro J. Ponce de León, Antonio Pertusa, Carlos Pérez-Sachno, and José M. Iñesta. Melodic track identification in MIDI files. In *Proceedings of the 19th International FLAIRS Conference*, Melbourne Beach, Florida, 2006.

[Shannon, 1948] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[Shmulevich and Povel, 2000] Ilya Shmulevich and Dirk-Jan Povel. Measures of temporal pattern complexity. *Journal of New Music Research*, 29(1):61–69, 2000.

[Snyder, 2000] Bob Snyder. *Music and Memory: An Introduction*. MIT Press, 2000.

[Tanguiane, 1993] A.S. Tanguiane. *Artificial Perception and Music Recognition*. Springer, Berlin, 1993.

[Uitdenbogerd and Zobel, 1998] Alexandra L. Uitdenbogerd and Justin Zobel. Manipulation of music for melody matching. In *ACM Multimedia*, pages 235–240, 1998.

[van Rijsbergen, 1979] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, London, 1979.

[Weyde and Datzko, 2005] T. Weyde and C. Datzko. Efficient melody retrieval with motif contour classes. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, London, U.K., 2005.

[Ziv and Lempel, 1977] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.