

A Comparison of Different Approaches to Melodic Similarity ^{*}

Maarten Grachten, Josep-Lluís Arcos, and Ramon López de Mántaras

IIIA-CSIC - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain.
Vox: +34-93-5809570, Fax: +34-93-5809661
Email: {maarten,arcos,mantaras}@iia.csic.es

Abstract. Computing similarities in sequences of notes is a very general problem with diverse musical applications ranging from music analysis to content-based retrieval. Choosing the appropriate level of representation is a crucial issue and depends on the type of application. In this paper, we describe some experimental work comparing four different similarity assessments. We will try to explain the results and argue that these measures can be ordered in terms of abstraction.

1 Introduction

Computing similarities in sequences of notes is a very general problem with diverse musical applications ranging from music analysis to content-based retrieval. Choosing the appropriate level of representation is a crucial issue and depends on the type of application. For example, in applications such as pattern discovery in musical sequences [1], [2], or style recognition [2], it has been established that melodic comparison requires taking into account not only the individual notes but also the structural information based on music theory and music cognition [3]. In this paper, we describe some experimental work comparing four different similarity assessments. We performed note-level, two variants of melodic contour-level [4] and what we call I/R-level similarity assessments respectively. I/R-level similarity is based on a structural description of melodic passages according to Narmours Implication/Realization (I/R) model of cognition of melodies [5]. To the best of our knowledge, this is the first attempt to develop an automatic parser to extract the I/R description of melodies and to compute similarities based on these descriptions and its comparison to similarity computations on different levels. Intuitively, the contour-level is more abstract than the note-level and I/R is more abstract than the contour-level. Our hypothesis is that the assessment of similarity increases with the level of musical abstraction. Similarity computation is done by means of a dynamic programming algorithm based on the concept of *edit distance* that was first adapted to

^{*} This research has been supported by the TIC Project 2000-1094-C02 **Tabasco: Content-based Audio Transformations**.

musical applications by Mongeau and Sankoff in [6]. The melodic passages for our experiments are 71 phrases from 21 Real Book compositions.

The paper is organized as follows: In Section 2 we introduce the Narmour’s Implication/Realization Model. In section 3 we present the parser we have developed for automating the analysis of melodies in terms of the I/R model. In section 4 we describe the four dissimilarity measures we are comparing—the note-level similarity previously presented in [6], two variants of contour-level similarity and the I/R-level similarity we propose as an alternative. In section 5 we report the experiments performed using these four similarity measures. The paper ends with a discussion of the results, and the planned future work.

2 Narmour’s Implication/Realization Model

An intuition shared by many people is that appreciating music has to do with expectation. That is, what he have already heard builds expectations on what is to come. These expectations can be fulfilled or not by what is to come. If fulfilled the listener feels satisfied. If not, the listener is surprised or even disappointed. Based on this, Narmour proposed a theory of cognition of melodies based on a set of basic grouping structures (see figure 1). These structures characterize patterns of melodic implications (or expectations) that constitute the basic units of the listener’s perception. Other resources such as duration and rhythmic patterns emphasize or inhibit the perception of these melodic implications. The use of the Implication/Realization (I/R) model provides a musical analysis of the melodic surface of the piece. The basic grouping structures are (see figure 1): the P structure (a pattern composed of a sequence of at least three notes with similar intervallic distances and the same registral direction), the ID structure (a sequence of three notes with the same intervallic difference and different registral direction), the D structure (a repetition of at least three notes), the IP structure (a sequence of three notes with similar intervallic distances and different registral direction), the VP structure (a sequence of three notes with the same registral direction, the first intervallic distance is a step and the second is a leap), the R structure (a sequence of three notes, different registral direction, the first intervallic distance is a leap and the second is a step), the IR structure (a sequence of three notes, same registral direction, the first intervallic distance is a leap and the second is a step), and the VR structure (a sequence of three notes, different registral direction, both intervals are leaps). In the example shown in figure 2, the first three notes form a P structure, the next four notes an ID together with a P (sharing the d-c interval) and the last three notes form another P. The three P structures in the example have a descending registral direction and in the first and last cases there is durational cumulation (the last note is significantly longer). The chaining of the ID and the second P is caused by the lack of durational cumulation. The second P is ended due to the metrical accent on the first beat of the measure.

Looking at melodic groupings in this way, we can see how each pitch interval implies the next. Thus, an interval may be continued with a similar one



Fig. 1. Eight of the basic structures of the I/R model (taken from [7]).



Fig. 2. First measures of All of Me, with I/R structures.

(such as P or ID or IP or VR) or reversed with a dissimilar one. That is, a step (small interval between notes) followed by a leap (large interval between notes) in the same direction would be a reversal of the implied interval (another step was expected but, instead, a leap is heard), but not a reversal of direction. Pitch motion may also be continued by moving in the same direction (up or down) or reversed by moving in the opposite direction. The strongest kind of reversal involves both a reversal of interval and a reversal of direction. When several small intervals (steps) move consistently in the same direction, they strongly imply continuation in the same direction with similar small intervals. If a leap occurs instead of a step, it creates a continuity gap. This triggers the expectation that the gap should be filled in. To fill it, the next step intervals should move in the opposite direction from the leap. This also tends to limit pitch range and keeps melodies moving back toward a center. Basically, continuity (satisfying the expectations) is nonclosural and progressive, whereas denial of implication (not satisfying the expectation) is closural and segmentive. A long note duration after reversal of implication usually confirms phrase closure.

3 The I/R Parser

The I/R model gives rise to the analysis of melodies into a sequence of (either basic or derived) structures, the theoretical constructs of the model. Narmour supposes the simultaneous activity of top-down and bottom-up processes, both guiding the listener's expectations in the perception of music. The bottom-up process, Narmour argues, is innate, subconscious and reflexive. It leads from musical/auditive input to expectations about subsequent input in a mechanistic way, following if-then rules. In addition, the top-down process at the same time actuates learned musical schemata, and interferes with the expectations generated by the bottom-up process. This behaviour can appropriately be conceived of as the imposition of except-clauses to the if-then rules that the bottom-up process follows. The top-down contribution to expectations of the listener represents her knowledge of musical style, the idiosyncratic features that characterize

a musical style (called *extraopus style* in the I/R model), or more specifically, a particular piece of music (*intraopus style* in the I/R model).

To automate the process of deriving I/R analyses from melodies, we have implemented a parser. At this time, the parser only implements the bottom-up process of the model, thus lacking any piece or style specific knowledge. Although the cognitive plausibility of the I/R analyses could doubtlessly be improved by adding a top-down component to the parser, we nevertheless believe that the current analyses have a reasonable degree of validity (top-down interferences being occasional exceptions to the bottom-up rules) and are useful for our purposes.

The parser takes as input a melody, represented as a sequence of notes, having pitch, duration and position attributes. Additionally, meter is known. The process of parsing this melody to obtain a sequence of I/R structures, can be divided into a small number of straight-forward steps:

1. Tag pairs of intervals with labels to categorize them as I/R structures
2. Detect positions where closure occurs
3. Detect positions where inhibition of closure (non-closure) occurs
4. Categorize the interval pairs within the sequence segments in between closures
5. Assimilate chained P and D structures

In the following subsections, we will explain these steps in more detail.

3.1 Tagging Interval Pairs

This step consists of checking every subsequent pair of intervals of the melody and tagging them with two boolean labels: *intervallic similarity*, and *registral sameness*. These two features are of primary importance in the classification of a sequence of notes as a particular I/R structure.

In the I/R model, two subsequent intervals are said to have intervallic similarity if their *intervallic differentiation* (i.e. the difference between these intervals) is less or equal two a minor third. The registral sameness feature is present if subsequent intervals have the same registral direction, which can be either upward, downward, or lateral (in the case of a prime interval).

3.2 Detecting Closure

In the I/R model, *closure* is the term for the inhibition of implication. This inhibition can be established by several conditions. The most prominent are rests, *durational cumulation* (where a note has significantly longer duration than its predecessor), metrical accents, and resolution of dissonance into consonance. These conditions can occur in any combination. Depending on the number of dimensions in which closure occurs and the degree of closure in each dimension, the overall degree of closure will differ.

The closure detection step in the parser currently detects three kinds of conditions for closure: rests, durational cumulation, and metrical accents. On each point in the sequence, the level of closure is computed for each of the dimensions.

3.3 Detecting Non-Closure

Although the conditions mentioned in the previous subsection principally imply closure, there are also circumstances where closure is inhibited. Examples of these circumstances are:

- The absence of metrical emphasis
- The envelopment of a metric accent by a P process in the context of additive (equally long) or counter-cumulative (long to short) durations
- The envelopment of a metric accent by a D process in the context of additive (equally long) durations
- The occurrence of dissonance on a metric accent

Currently, the first three of these circumstances are recognized during the step of detecting conditions for non-closure. In these cases, closure due to metrical emphasis that was detected in the previous step, is canceled.

3.4 Categorizing the Interval Pairs

When the final degree of closure is established for each point in the sequence, the next step is the categorization of the interval pairs, based on the information gathered during the first step. The information about the degree of closure is also used in this step. In between two parts of closure, the structures are overlapping, sharing two notes (i.e. one interval), due to the lack of closure. Where closure occurs, two contiguous structures either share one note, or none, depending on the degree of closure.

This procedure generally leads to structures consisting of three notes, except in cases where closure occurs on two subsequent notes, or when a note after a closural note is followed by a rest. In the first case a two note structure, or *dyad* results; in the second case a single note structure, or *monad* results.

3.5 Assimilating Chained P and D Structures

In the I/R model, there are two of the basic structures that can span more than three notes, viz. P and D structures. P Structures occur when three or more notes form an ascending or descending sequence of similar intervals; D structures occur when three or more notes continue in lateral direction, i.e. a note is repeated. A further condition is the absence of closure, for example by durational cumulation.

In the sequence of structures, only structures of three notes have been identified. Yet this is not problematic, since possible instances of longer P and D structures necessarily manifest themselves as a chain of P or a chain of D structures, overlapping each other by two notes. Subsequent P or D structures that share only one note, cannot be regarded as a single P or D structure, since the fact that the overlap is less than two notes, indicates that there is some (weak) degree of closure.

So the last step of the construction of an I/R analysis is simply to 'reduce' any chain of overlapping D or P structures into single structures.

The final I/R analysis consists of a sequence of structures. The structures of this sequence is represented by the parser as a symbols denoting structural type, with references to the notes of the melody that belong to that structure.

4 Measuring Dissimilarities

For the comparison of the musical material on different levels, we used a measure for dissimilarity that is based on the concept of *edit-distance*. In general, the edit-distance between two sequences can be defined as the minimum total cost of transforming one sequence (the source sequence) into the other (the target sequence), given a set of allowed edit operations and a cost function that defines the cost of each edit operation. The most common set of edit operations contains insertion, deletion, and replacement. Insertion is the operation of adding an element at some point in the target sequence; deletion refers to the removal of an element from the source sequence; replacement is the substitution of an element from the target sequence for an element of the source sequence. Mongeau and Sankoff [6] have described a way of applying this measure of dissimilarity to (monophonic) melodies (represented as sequences of notes). We will present an algorithm for computing the edit distance at the end of this section.

Because this dissimilarity measure is a measure for comparing sequences in general, it enables one to compare melodies not only as note sequences, but in principle any sequential representation can be compared. In addition to comparing note-sequences, we have investigated the dissimilarities between melodies by representing them as sequences of directional intervals, directions, and I/R structures, respectively. The next subsections briefly describe our decisions regarding the choice of edit-operations and weights of operations for each type of sequence. We do not claim these are the only right choices. In fact, this issue deserves more discussion and might benefit also from empirical data conveying human dissimilarity ratings of musical material.

4.1 An edit-distance for note sequences

In the case of note sequences, we have followed Mongeau and Sankoff's approach [6]. They propose to extend the set of basic operations (insertion, deletion, replacement) by two other operations that are more domain specific: *fragmentation* and *consolidation*. Fragmentation is the substitution of a number of (contiguous) elements from the target sequence for one element of the source sequence; conversely, consolidation is the substitution of one element from the target-sequence for a number of (contiguous) elements of the source sequence. In musical variations of a melody for example, it is not uncommon for a long note to be fragmented into several shorter ones, whose durations add up to the length of the original long note.

The weights of the operations are all linear combinations of the durations and pitches of the notes involved in the operation. The weights of insertion and

deletion of a note are equal to the duration of the note. The weight of a replacement of a note by another note is defined as the sum of the absolute difference of the pitches and the absolute difference of the durations of the notes. Additionally, there is a weight factor for the duration difference, in order to control the relative importance of pitch and duration attributes. Fragmentation and consolidation weights are calculated similarly: the weight of fragmenting a note n_1 into a sequence of notes n_2, n_3, \dots, n_N is again composed of a pitch part and a duration part. The pitch part is defined by the sum of the absolute pitch differences between n_1 and n_2 , n_1 and n_3 , etc. The duration part is defined by the absolute difference between the duration of n_1 , and the summed durations of n_2, n_3, \dots, n_N . Just like the replacement weight the fragmentation weight is a weighted sum of the pitch and duration parts. The weight of consolidation is exactly the converse of the weight of fragmentation.

4.2 An edit-distance for contour sequences

One way to conceive of the contour of a melody is as comprising the intervallic relationships between consecutive notes. In this case, the contour is represented by a sequence of signed intervals. Another idea of contour is that it just refers to the up/down pattern of the melody, discarding the sizes of intervals. In our experiment, we have computed dissimilarities for both kinds of contour sequences. The elements of the up/down sequences were either 1 (designating an ascending interval), -1 (designating a descending interval) or 0 (designating a zero interval).

We have restricted the set of edit operations for both kinds of sequences to the basic set of insertion, deletion and replacement. Fragmentation and consolidation between contour sequences are not easily justifiable, since there is no correspondence to fragmentation/consolidation as musical phenomena (as there is trivially in the case of note-sequences). Other kinds of edit operations may be conceivable for contour sequences, but these are currently not taken into account.

For the weight of replacement operations upon (both kinds of) contour sequences, there is a very intuitive and simple candidate: the absolute difference between the two elements under consideration. A more difficult question is how the insertion/deletion weights should relate to this replacement weight. We feel that on the one hand, the weight of insertion or deletion of a contour sequence element should be equal to the replacement of this element by a zero interval/direction element. On the other hand, our idea is that the weight of insertion or deletion should be independent of the size of the inserted/deleted element, implying a constant weight for insertions and deletions. We realized this by setting the weight of insertions/deletions to 1, and dividing the weight of replacement (the absolute difference between the values of the two elements) by the maximum of the absolute values of the two elements.

4.3 An edit-distance for I/R sequences

The sequences of possibly overlapping I/R structures (I/R sequences, for short) that the I/R parser generated for the musical phrases, were also compared with

each other. Just as with the contour sequences, it is not obvious which kinds of edit operations could be justified beyond insertion, deletion and replacement. It is possible that research investigating the I/R sequences of melodies that are musical variations of each other, will point out common transformations of music at the level of I/R sequences. In that case, edit operations may be introduced to allow for such common transformations. Presently however, we know of no such common transformations, so we allowed only insertion, deletion and replacement.

As for the estimation of weights for edit operations upon I/R structures, it can be noted that unlike the replacement operation, the insertion and deletion operations do not involve any comparison between I/R structures. It seems reasonable to make the weights of insertion/deletion somehow proportional to the ‘importance’ or ‘significance’ of the I/R structure to be inserted/deleted. Ideally the (unformalized) notion of significance of an I/R structure would depend on the context of the structure. However, this would not make sense in the case of editing sequences, as this would create a cyclic dependence among the weights of edit operations. Therefore we propose to take the size of an I/R structure, referring to the number of notes the structure spans, as a more practical indicator of the significance of an I/R structure. The weight of an insertion/deletion of an I/R structure can then simply be the size of the structure.

The weight of a replacement of two I/R structures would preferably take into account the two structures under consideration. Ideally, one would assign high weights to replacements that involve two very different I/R structures and low weights to replacements of an I/R structure by a similar one. The rating of dissimilarities between different I/R structures (which to our knowledge has as yet remained unaddressed) is a difficult issue, that can be approached from several angles. One approach is to base the dissimilarity-judgments only on *abstract* attributes of the I/R structures. That is, the dissimilarity ratings are fixed a priori for each pair of I/R structures, just by taking into account intrinsic characteristics of the structures. An example of such characteristics could be the continuation/change of registral direction within the structure. A more theoretical characteristic is the extent to which the latter part of the structure satisfies the expectation that was generated by the former part. Non-intrinsic attributes of I/R structures are for example the number of notes that the structure spans, or the exact intervals of the structure.

Aiming at a straight-forward and simple definition of replacement weights for I/R structures, we decided to take into account just three attributes. The first term in the weight expression is the difference in size (i.e. number of notes) of the I/R structures. Secondly, a cost is added if the direction of the structures is different (where the direction of an I/R structure is defined as the direction of the interval between the first and the last note of the structure). Lastly, a cost is added if the two I/R structures are not of the same kind. A special case occurs when one of the I/R structures is the *retrospective* counterpart of the other (a retrospective structure generally has the same up/down contour as its prospective counterpart, but different interval sizes; for instance, a retrospective P structure typically consists of two large intervals in the same direction, see [5]

for details). In this case, a reduced cost is added, representing the idea that a pair of retrospective/prospective counterparts of the same kind of I/R structure is more similar than a pair of structures of different kinds.

4.4 Computing the dissimilarities

The minimum cost of transforming a source sequence into a target sequence, can be calculated relatively fast, using the following recurrence equation for the dissimilarity d_{ij} between two sequences a_1, a_2, \dots, a_i and b_1, b_2, \dots, b_j :

$$d_{ij} = \min \begin{cases} d_{i-1,j} + w(a_i, \emptyset) & \text{deletion} \\ d_{i,j-1} + w(\emptyset, b_j) & \text{insertion} \\ d_{i-1,j-k} + w(a_i, b_{j-k+1}, \dots, b_j), \quad 2 \leq k \leq j & \text{fragmentation} \\ d_{i-k,j-1} + w(a_{i-k+1}, \dots, a_i, b_j), \quad 2 \leq k \leq i & \text{consolidation} \\ d_{i-1,j-1} + w(a_i, b_j) & \text{replacement} \end{cases}$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$, where m is the length of the source sequence and n is the length of the target sequence. Additionally, the initial conditions for the recurrence equation are:

$$\begin{aligned} d_{i0} &= d_{i-1,j} + w(a_i, \emptyset) && \text{deletion} \\ d_{0j} &= d_{i,j-1} + w(\emptyset, b_j) && \text{insertion} \\ d_{00} &= 0 \end{aligned}$$

For two sequences a and b , consisting of m and n elements respectively, we take d_{mn} as the dissimilarity between a and b .

The weight function w , defines the cost of operations (which we discussed in the previous subsections), such that e.g. $w(a_4, \emptyset)$ returns the cost of deleting element a_4 from the source sequence, and $w(a_3, b_5, b_6, b_7)$ returns the cost of fragmenting element a_3 from the source sequence into the subsequence $\langle b_5, b_6, b_7 \rangle$ of the target sequence. For computing the dissimilarities between the contour and I/R sequences respectively, the clauses corresponding to the cost of fragmentation and consolidation are simply left out of the recurrence equation.

5 Experimentation

The comparison of the different dissimilarity measures was performed using 71 different musical phrases from 21 different jazz standards. The musical phrases have a mean duration of eight bars. Among them are jazz ballads like ‘How High the Moon’ with around 20 notes, many of them with long duration, and Bebop themes like ‘Donna Lee’ with around 55 notes of short duration.

With the 71 jazz phrases we performed all the possible pair-wise comparisons (2485) using the four different measures. The resulting dissimilarity values were normalized per measure. Figure 3 shows the distribution of dissimilarity values for each measure. In addition, the measures were compared pairwise in a number of ways. The results of these comparisons are shown in table 1. The

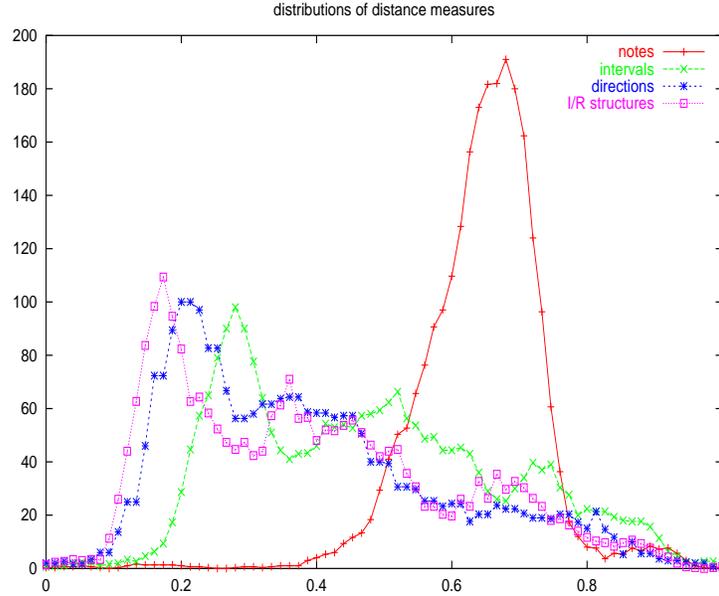


Fig. 3. Distribution of dissimilarities for four dissimilarity measures. The x axis represents the normalized values for the dissimilarities between pairs of phrases. The y axis represents the number of pairs that have the dissimilarity level shown in the x axis.

first row of data shows correlation values. The second shows the fraction of data pairs for which the second dissimilarity measure turned out lower than the first. In the third row, the mean difference between the first and second measure is displayed. The fourth row shows the standard deviation of this difference.

	X1: notes X2: intervals	X1: notes X2: directions	X1: notes X2: I/R	X1: intervals X2: directions	X1: intervals X2: I/R	X1: directions X2: I/R
$r(X1, X2)$.10	.08	.12	.95	.91	.91
$P(X2 \leq X1)$.76	.86	.84	1.00	.85	.53
$E(X2 - X1)$	-.17	-.26	-.26	-.09	-.09	.00
$\sigma(X2 - X1)$.21	.21	.21	.06	.08	.09

Table 1. Four dissimilarity measures compared pairwise. The measures are shown in the uppermost row. In the left column, $X1$ and $X2$ respectively refer to the two measures under comparison.

The first thing to notice from figure 3 is the difference in dissimilarity assessments at the note-level on the one hand, and the interval, direction and I/R-levels on the other hand. Whereas the dissimilarity distributions of the last three measures are spread across the spectrum with peaks around .2, the note level measure has its values highly concentrated around approximately .65. This suggests that the note-level measure has a low discriminative power. One reason for this could be that our implementation of the note-level measure is not invariant to transpositions and mode.

The interval, direction and I/R-level measures are more resembling. On the average, the interval measure tends to rate dissimilarities between phrases slightly higher than the direction and I/R-level measures. For the direction measure, this is not surprising, because similar intervals imply similar direction, but not vice versa. The fact that the I/R-level measure rates dissimilarities lower on average, can be accounted for by several reasons. Firstly, the I/R structures P and D, can possibly span a large number of notes. Although the number of notes is taken into account by the edit operation weights for I/R structures, it determines only part of the weight. In some cases this feature can facilitate transformation of the source sequence into the target sequence. A second reason could be that I/R structures do not incorporate intervals between notes where closure occurs, whereas the interval-level measure considers intervals between every pair of consecutive notes. When this interval is large, this adds to the transformation cost of the interval measure.

The relationship between the distance measure and the I/R measure is not obvious. From figure 3, it appears as if the I/R dissimilarities are generally lower than the direction dissimilarities. This is not true however, as table 1 indicates that the two measures assess the phrase pairs as equally dissimilar on average. The standard deviation of the differences between the I/R and direction dissimilarities is .09 (which implies that 75% of the differences were smaller than .18). Plotting the differences between direction and I/R measures against the I/R dissimilarity showed that for pairs of phrases that were I/R-similar, the I/R dissimilarities were generally smaller than direction dissimilarities, whereas for pairs of pairs that were I/R-dissimilar, the direction dissimilarities were generally smaller than I/R dissimilarities.

The results seem to partly confirm our hypothesis, as stated in section 1, that dissimilarity assessments for a given pair of phrases are highest on the note-level, and lowest on the I/R level, with contour-level in between. In figure 4, a pair of phrases is shown for which the difference in similarities between note-level and I/R-level is relatively easy to see. However, the musical abstraction ordering between I/R-level and direction-level is not very convincing. We intend to focus on the cases where similarity is higher on the lower musical abstraction levels than on the higher levels. Pointing out the conditions for such cases could help in the evaluation of the different similarity measures. This is however beyond the scope of this paper.

A last point of interest are the three ‘bumps’ in the dissimilarity distributions of the interval, direction and I/R measures. These accumulations of data-points around certain values might be taken as an indication that the data consists of several clusters of similar phrases. The intra-cluster comparisons would be in the first bump, and the inter-cluster comparisons in any subsequent bumps, depending on the distance between clusters. In fact, most of the phrase pairs rated as highly dissimilar (by any of the three measures), involved phrases that were known to be Bebop style (like ‘Donna Lee’, ‘Confirmation’ and ‘Dexterity’). For now however, the idea of style consistent clusters of similar phrases is just a conjecture that requires more investigation.

Second phrase of Autumn Leaves

First phrase of How High The Moon

Fig. 4. Two phrases that exhibit decreasing dissimilarity on increasing levels of musical abstraction. note-level: .64; Interval-level: .26; Direction-level: .14; I/R-level: .10

6 Conclusions and future work

The discriminative power of the note-level similarity measure is less than that of the two contour-level measures (i.e. interval-level and direction-level) and the I/R-level measure. It is conceivable that the note-level similarity measure is too fine-grained for musical phrases and would be more appropriate to assess similarities between smaller musical units (e.g. motifs). On average, the interval-level measure rates dissimilarities lower than the note-level measure, but not as low as the direction- and I/R-level measures. The performance of direction- and I/R-level measures is very close in this respect. In the future, we wish to address the following issues:

- to investigate if the I/R measure can be approached further by the contour measures by taking into account durations in addition to intervals or directions
- to investigate further the possibility of identifying clusters of similar phrases through the use of similarity measures
- to identify different jazz styles and see how the similarity measures perform within these styles.

References

1. Cope, D.: *Computers and Musical Style*. Oxford University Press (1991)
2. Hörnel, D., Menzel, W.: Learning musical structure and style with neural networks. *Computer Music Journal* **22** (4) (1999) 44–62
3. Rolland, P.: Discovering patterns in musical sequences. *Journal of New Music Research* **28** (4) (1999) 334–350
4. Dowling, W.J.: Scale and contour: Two components of a theory of memory for melodies. *Psychological Review* **85** (1978) 341–354
5. Narmour, E.: *The analysis and cognition of basic melodic structures: the implication realization model*. University of Chicago Press (1990)
6. Mongeau, M., Sankoff, D.: Comparison of musical sequences. *Computers and the Humanities* **24** (1990) 161–175
7. Narmour, E.: The melodic structures of music and speech: Applications and dimensions of the implication-realization model. In Sundberg, J., et al., eds.: *Music, language, speech and the brain*. Volume 59., Macmillan (1991) 48–56