Evaluating Melodic Segmentation*

Christian Spevak, Belinda Thom, and Karin Höthker**

Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssyteme,
Am Fasanengarten 5, 76128 Karlsruhe, Germany
{spevak,hoethker}@ira.uka.de, bthom@cs.cmu.edu
WWW home page: http://i11www.ira.uka.de/~musik

Abstract. We discuss why evaluating melodic segmentation is difficult when solutions are ambiguous and explore the nature of this ambiguity using a corpus of melodies hand-segmented by musicians. For each melody, the musicians identified different, musically coherent solutions, suggesting that ambiguity should be modeled when assessing how "well" an algorithm segments the same melody. We propose a probabilistic framework for modeling ambiguity that integrates both the segment boundaries and the lengths that the musicians preferred. The framework gives rise to several potential extensions of existing segmentation algorithms.

1 Introduction

Segmenting or parsing a melody amounts to imposing a temporal structure on a sequence of discrete pitch symbols. This structure is defined by pairs of boundaries that break the sequence up into various subsequences. It may involve different levels of hierarchy (Lerdahl and Jackendoff 1983), overlapping boundaries (Crawford et al. 1998), and unclassified areas that do not belong to any segment. From a computational point of view it makes sense to simplify the task, to focus on breaking a melody into a *segment stream* that contains a series of non-overlapping, contiguous fragments. An algorithm that automatically produces a "musically reasonable" segment stream would provide an invaluable tool for melodic modeling. For example, it could perform an important preprocessing step, dividing up a melody into "locally salient chunks" before considering higher-level melodic features.

Although we are unaware of any past research that directly supports this claim, melodic segmentation is most certainly an ambiguous affair. Ambiguity arises because it is typically not possible to determine one "correct" segmentation for a melody. Rather, the process is influenced by a rich and varied set of contexts, where local structure (gestalt principles), higher-level structure (e.g. recurring motives, harmony, melodic parallelism), style-dependent norms and the breaking

* To appear in *Proceedings of the II International Conference on Music and Artificial Intelligence (ICMAI)*, 2002.

** Author ordering was determined by stochastic simulation.

of these norms all have an impact on what is perceived as "salient." A two-fold goal drives this research. First, we want to explore qualitatively how this ambiguity manifests itself in different musical situations. Second, we strive to provide a framework for quantifying ambiguity.

We began our research by conducting a study in order to observe the solutions we would get when different musicians were asked to segment a fixed melodic corpus. This data collection and its qualitative evaluation are described in Sects. 2 and 4, respectively. In Sect. 3, we present three existing segmentation algorithms, which set the stage for considering the systematic evaluation of an algorithm's performance. In Sect. 5, we introduce three increasingly complex approaches for quantitatively assessing a solution in the context of an ambiguous set of reference segmentations. These approaches address the problem of identifying how "appropriate" a segmentation is rather than just identifying whether a segmentation is "correct." Since musically plausible segmentations tend to exhibit higher-level structure, such as a certain range of lengths, we incorporate it explicitly, obtaining more complex, yet more plausible, evaluation measures.

2 Manual Segmentation

We began exploring the task of melodic segmentation by collecting data from a number of musicians. We compiled a small corpus of ten melodic excerpts in different musical styles. Some excerpts were selected because of their ambiguous musical structure (e.g. excerpts from a Bach fugue and a Haydn string quartet), others were selected because of their noticeable lack of structure (an excerpt from Wagner's Parsival). Four folk songs from the *Essen* collection (Schaffrath 1995) were included in order to explore how ambiguous a "simple" folk tune might be. Finally, the corpus included an improvisation over a Bluegrass standard, a Pop song, and a Bebop jazz solo. For each excerpt, we provided a minimal score representing the monophonic melody (including meter) and a "dead-pan" MIDI file. Additional information, such as accompaniment, harmony, lyrics, and origin, was withheld as segmentation algorithms typically do not use such information.

Twenty-two musicians with various levels of expertise – including professionals, musicologists, music teachers, and amateurs – were asked to identify "salient melodic chunks" for each excerpt. Subjects were instructed to identify boundaries at two different levels of granularity, which we called the *phrase level* and the *sub-phrase level*. Sub-phrases were only required to be more fine-grained than phrases, providing additional structure at "the next level down." Instructions were kept deliberately vague. For instance, the term "motive" was not used because we wanted the musicians to draw upon their musical intuition rather than perform explicit melodic analyses. Subjects were instructed to identify each segment by placing a mark above its first note. In this way, each previous segment was defined as ending just before the next mark, which ensured that a musician's solution was always a segment stream.

3 Algorithmic Segmentation

We now present three existing segmentation algorithms and consider how they have been evaluated. A key aspect of each algorithm is its output representation, for it has an impact on how ambiguity might be considered when evaluating an algorithm's performance. A segmentation algorithm's most basic input is a melodic line, encoded by a *note list* comprised of note elements. Variable n refers to the number of notes contained within the list. Note elements are temporally ordered, each defined by a discrete pitch, an onset, and an offset time. Individual notes are referred to by index i, which identifies the location a note occupies in the sequence. Different segmentation algorithms might extract different features from this list. For instance, a list of inter-onset intervals (IOIs) measures the duration between successive pitches, a list of offset-to-onset intervals (OOIs) indicates the rests between successive note pairs, and a list of absolute pitch intervals (APIs) contains the absolute difference in semitones between adjacent pitches (each interval's sign is ignored). A segmentation algorithm's most basic output is *segmentation vector* $\boldsymbol{s}$, where $\boldsymbol{s} = (s_1, ..., s_i, ..., s_n) \in \{0, 1\}^n$. Only when a segment starts at position i does s_i equal 1; otherwise it is 0. An algorithm might also output a *boundary strength* vector $\boldsymbol{w}$, where $\boldsymbol{w} = (w_1, ..., w_i, ..., w_n) \in [0, 1]^n$. w_i quantifies how "strong" an algorithm perceives the boundary at position i to be, and the vector is normalized over its entire length. Finally, one could imagine that an algorithm might output a *fitness value* that rates its solution's plausibility.

3.1 Algorithms

Grouper. This algorithm is part of the Melisma Music Analyzer developed by Temperley (2001) and Sleator.[1] Grouper was designed to extract melodic phrase structure from a monophonic piece of music. Its input is both a note list and a *beat list*, which specifies the metrical structure of the melody. Internally, only duration (IOIs and OOIs) and metric information (beat list) are considered; pitch is ignored. For a given input, output $\boldsymbol{s}$ is deterministic. Dynamic programming (Bellman 1957) is used to carry out an efficient search for an "optimal" segmentation over the set of 2^n possible solutions. Optimality is defined in terms of preference rules adapted from Lerdahl and Jackendoff (1983). Roughly speaking, these rules prefer a solution whose boundaries are both coincident with the underlying metrical structure and with events whose IOIs and OOIs are larger. Preference is also given to solutions whose individual segment lengths approximate a prespecified number of notes. These rules are controlled by high-level parameters that specify the penalty to be assigned to each segment within $\boldsymbol{s}$. The optimal solution is the one whose cumulative penalty is smallest.

Local Boundary Detection Model (LBDM). This model, developed by Cambouropoulos (2001), is motivated by the gestalt principles of similarity and

[1] Ready-to-use software is available at www.links.cs.cmu.edu/music-analysis.

proximity. As such, it quantifies the degree of discontinuity locally along each adjacent pair of notes. While LBDM is not a complete model of grouping in itself,[2] it is still compelling to consider this algorithm in isolation, especially since it might be used to segment raw MIDI data in real time. Another advantage is the model's parsimony, which contributes to making it straightforward to implement.[3] LBDM deterministically transforms a note list into a segmentation vector $\boldsymbol{s}$ and boundary strength vector $\boldsymbol{w}$. Internally, discontinuity is quantified using IOI, OOI, and API lists. At each potential boundary location, a strength is calculated that is proportional to the rate at which the neighboring intervals change and the magnitude of the current interval. For each list, a separate strength vector is calculated. $\boldsymbol{w}$ is the weighted sum of the three individual strength profiles, where high-level parameters control the contribution of each profile. Segmentation vector $\boldsymbol{s}$ is obtained by analyzing the shape of $\boldsymbol{w}$ and inserting a boundary at every "significant" local maximum. To prevent the algorithm from subdividing a melody too much, maxima are only considered at those locations where $\boldsymbol{w}$ exceeds a high-level threshold parameter (cf. Fig. 1).

Data-Oriented Parsing (DOP) Bod (2001) argued for a memory-based approach to segmentation and implemented such a system using a probabilistic grammar technique. His most sophisticated and successful grammar performs data oriented parsing (DOP), which learns probabilistic trees from a large corpus of presegmented musical data. Musically, DOP is interesting because it imposes a higher-level structural definition upon the model that is to be learned, biasing it to prefer solutions that contain an "ideal" number of segments. Internally, parsing only considers pitch (in relation to the tonic) and note length. After training, the trees can parse an input note list, producing an output $\boldsymbol{s}$. To implement DOP would involve significantly more work than LBDM, and since we did not have access to a completely ready-to-use version of this software, we do not present examples of DOP's output. We introduce this model, rather, because of its probabilistic nature, a property that provides an attractive mechanism for handling ambiguity.

3.2 Algorithm Parameter Estimation

In the examples described in Sect. 4, we present segmentations by LBDM and Grouper along with the musicians' data. Each segmentation was constructed using a high-level set of parameters that performed well on a large subset of the Essen collection. For details, see Höthker et al. (2002).

3.3 Evaluation

Computational melodic segmentation is still a relatively new field of research and many unexplored issues remain. This is especially true concerning an algorithm's

[2] For example, musical parallelism (Cambouropoulos 1998) is not considered.
[3] Our software is available at i11www.ira.uka.de/~musik/segmentation.

evaluation. Cambouropoulos (2001) and Temperley (2001) have both validated their models by demonstrating how well they behave on hand-selected musical examples.[4] To demonstrate an algorithm's "musical plausibility," an analysis using specific melodic examples is *crucial*. However, a more objective, systematic measure is also needed because it allows important computational questions to be investigated. For example, such a measure was used to explore how sensitive LBDM and Grouper are with respect to high-level parameter settings and different musical styles (Höthker et al. 2002). Even more compelling, when machine learning is used, as with DOP, a systematic measure becomes indispensable.

One difficulty in assessing an algorithm's performance is that the classification task – to segment or not to segment at each note – is highly skewed. For instance, in the 52-note melody in Fig. 1, between 3 and 12 boundaries seem plausible. Thus, an algorithm that predicted no boundaries would appear to have a very low error rate. In the literature, a melodic segmentation algorithm's performance has been systematically measured using *F score* (Bod 2001, Höthker et al. 2002):

$$\mathrm{F}(\boldsymbol{s}, \boldsymbol{s}^*) = \frac{1}{1 + \frac{\mathrm{FN+FP}}{2\mathrm{TP}}} \in [0, 1], \tag{1}$$

where $\boldsymbol{s}$ and $\boldsymbol{s}^*$ are two segmentations of the same note list. Whereas TP, the number of true positives, records how many boundaries are identical in both segmentations, FP and FN, the number of false positives and false negatives, record how many boundaries are inserted in only one of the two solutions. F score is an appropriate evaluation measure because it excludes true negatives, and keeps them from misleadingly dominating the assessment. On the other hand, it makes the questionable assumption that errors at different boundary locations are independent.

4 Results

In this section, we present results from the data collection described in Sect. 2 and the segmentation algorithms referred to in Sect. 3.2. When we began analyzing the data, three subjects' segmentations were eliminated because they were incomplete. Among the remaining segmentations, a few seemed to be inconsistent or "strange." However, we decided not to interfere with this remaining data by eliminating potential outliers. For example, when one experienced musician obtained a substantially different segmentation for a folk song – either because a strong upbeat was disregarded or went unnoticed – we did not remove it because the task is so subjective and ambiguous. Additionally, robust outlier detection generally requires more than twenty data points, a practical issue, for the data collection and entry is quite time-consuming.

Despite these problems, the data clearly corroborates the hypothesis that ambiguity is an inherent property of the segmentation task and should not be ignored. Even for the "simple" folk song displayed in Fig. 1, nineteen musicians

[4] Temperley has also validated his model using songs from the Essen collection.

produced nine different segmentations on the sub-phrase level. In a more complex excerpt (Fig. 3), the number of segmentations rose to eighteen (only two of the nineteen musicians gave the same answer). In none of the ten melodic excerpts was unanimous agreement obtained. One cause for this ambiguity concerns granularity – one musician's notion of a phrase might more closely coincide with another's notion of a sub-phrase – yet in terms of identifying "locally salient chunks," both are musically reasonable. In other cases, musicians might agree on a phrase's length, but disagree on location, and in this situation, ambiguous boundaries are often adjacent to one another. The examples that follow clarify these points.

4.1 Example 1: Folk Song

The first example (Fig. 1) is a 19th century folk song taken from the Essen collection (E0356). The phrase structure contained in the Essen notation divides the song into 4+4+2+2+2+3 bars. The first two Essen phrases comprise a plain repetition, and are confirmed by practically all of the musicians at the phrase level. On the sub-phrase level, there is structural ambiguity, and a choice between "parallel alternatives" presents itself, where one can either emphasize the downbeat (more common) or the upbeat eighth-note G (less common). A notable feature of this ambiguity is its distribution on adjacent locations. There is also considerable disagreement on the phrase structure of the last five bars. The spectrum ranges from no phrase boundary, to boundaries at bar 13 or bar 15, to boundaries at both locations. Different notions of granularity are prevailing, resulting in a variable number of phrases per song (cf. Fig. 2). Again, either choice – whether to insert a boundary at bar 13 or at bar 15 – is reasonable, the former alternative focusing on the implicit harmonic structure and the latter emphasizing the melodic build-up in bars 9–14.

These findings suggest that the explicit segmentations in the Essen collection should not be regarded as the only, or even the most musically plausible, possibilities. It should rather be assumed that, in principle, different solutions can be equally well-founded. Our assumption is supported by a quotation from Dahlig (2002) that describes the Essen segmentation process:

> When we encode a tune without knowing its text, we do it just intuitively, using musical experience and sense of the structure. The idea is to make the phrase not too short (then it is a motive rather than a phrase) and not too long (then it becomes a musical period with cadence etc.). But, of course, there are no rules, the division is and will be subjective.

In other words, an algorithm's segmentations should be assessed in the light of these remarks. For instance, in the second half of Example 1, Grouper and Essen agree, yet they differ in the first half. Alternatively, when compared with the musicians' solutions, Grouper and LBDM both miss important boundaries at bars 5 and 9. The segmentation boundaries suggested by LBDM are located at large IOIs and pitch intervals, which do not provide reliable cues for musically plausible segmentations in this example. The musicians' segmentations are

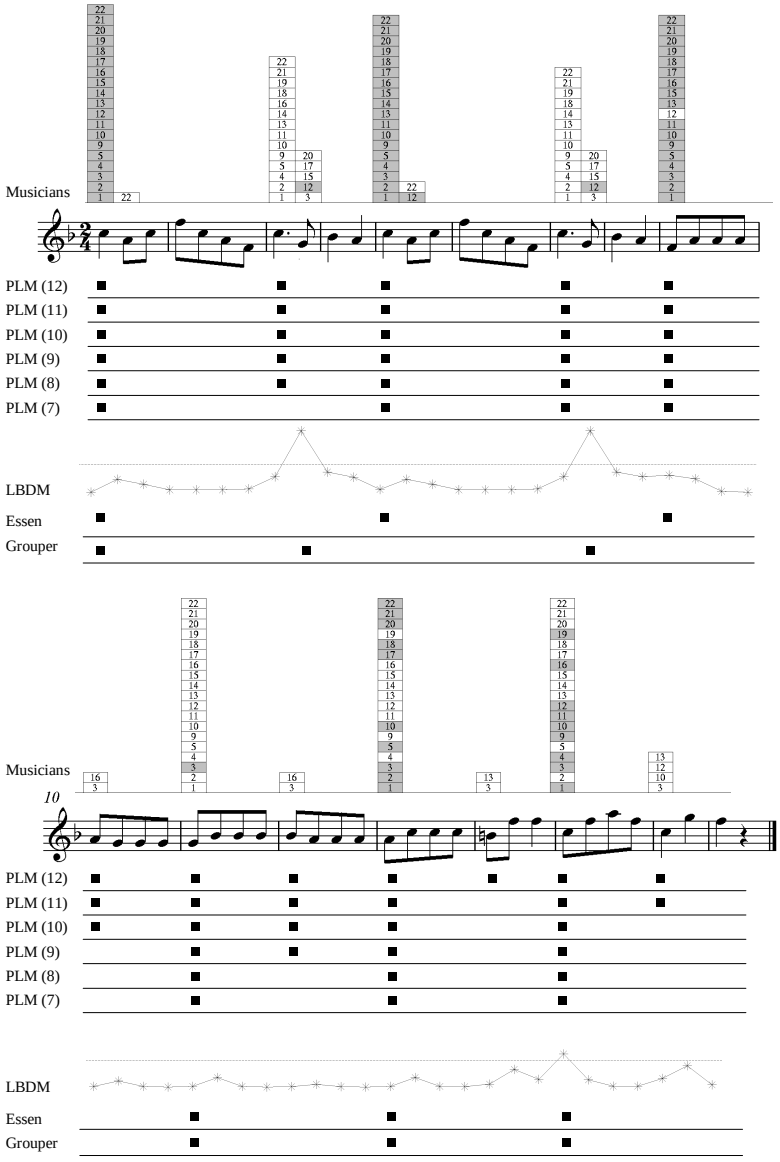

Fig. 1. Example 1: Folk song E0356, annotated with segmentation results from the musicians (upper histogram) and the algorithms (lower table). Each histogram count number identifies a particular musician. Grey squares identify the first note of both a phrase and sub-phrase; white squares correspond to sub-phrases only. The black squares refer to segmentations generated by the Position Length Model (Sect. 5.5). The last three rows show the boundary strength calculated by LBDM, the segmentation provided in the Essen collection, and Grouper's solution. The dashed line demonstrates how LBDM uses its threshold to derive $\boldsymbol{s}$ from $\boldsymbol{w}$.

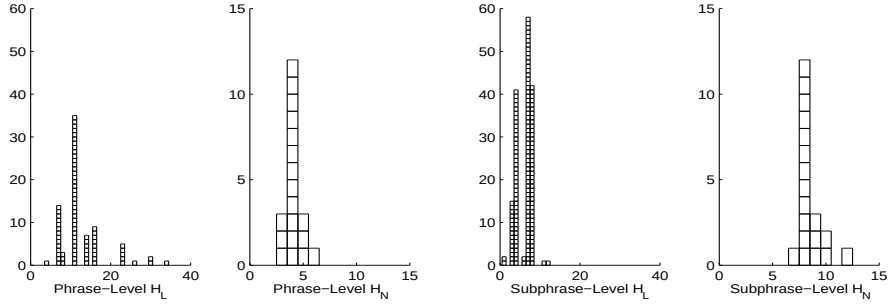

Fig. 2. Example 1: Distributions of the number of notes per segment (H_L) and the number of segments per song (H_N). For definitions, see Table 2.

clearly influenced by motivic parallelism, something neither algorithm considers. Metrical parallelism is also an influence, which only Grouper has access to.

4.2 Example 2: Fugue

The second example (Fig. 3), an excerpt from the beginning of Fugue XV of Bach's Well-Tempered Clavier I, is melodically more complex. In contrast to the previous example, we have no indication of a theoretically "correct" segmentation, and Bach did not notate explicit phrase marks.[5]

A quick glance at the histograms in Fig. 3 conveys an impression of the disagreement between the subjects, particularly on the sub-phrase (i.e. the motivic) level, where boundary marks can often be found right next to each other. A closer look, however, reveals that adjacent phrase marks never stem from the same musicians, and some segmentations run parallel to one another for several bars (e.g. along the melodic sequence in bars 6–7). While Grouper explicitly models such a behavior (preferring boundaries at parallel metric positions), in this example it nevertheless produces incoherent results in bars 6–7. Because of the complex interplay between the weighted rules and the dynamic programming optimization over the entire melody, it is difficult to explain why the algorithm chose exactly these boundaries. The behavior of LBDM is easier to predict because it operates on a strictly local context. The huge peak in the boundary strength in bar 9 shows that a single long note can create an unreasonably large amplitude that makes threshold selection more difficult.[6]

[5] Temperley (2001) points out: "Even structural information (phrasing slurs, bar lines, etc.) added by the composer is really only one 'expert opinion,' which one assumes generally corresponds to the way the piece is heard, but may not always do so." (p. 365).

[6] Cambouropoulos (2001) recommends using another high-level parameter to limit the boundary strength for large intervals.

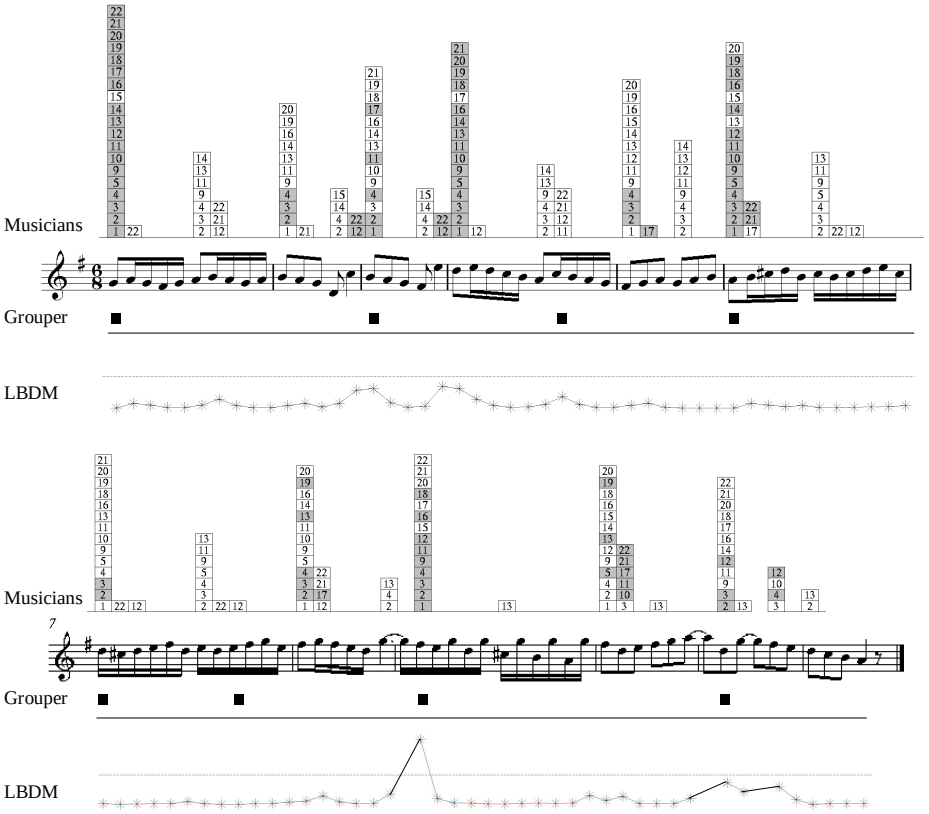

Fig. 3. Example 2: Excerpt from Fugue XV of Bach's Well-Tempered Clavier I. For details, see Fig. 1.

5 Modeling Ambiguity

Although the musician data obtained for a note list is clearly valuable, by itself it provides no mechanism for automatically determining how some new segmentation for the same note list should be evaluated. For a systematic evaluation of this sort, one would like a fitness evaluator:

$$\text{fit}(\boldsymbol{s}, \mathcal{S}) \rightarrow [0, 1]$$

that maps the new $\boldsymbol{s}$ into a larger value when it is "more plausible" given the musically motivated segmentations in *reference set* $\mathcal{S}$.

Several questions arise when attempting to evaluate a new and possibly *novel* segmentation given an ambiguous reference set. First, when should certain segmentations be perceived as more or less "similar" to one another? Second, when should an explicit parametric model be used and what parametric form is appropriate? Alternatively, should the mapping be constructed in a completely non-

parametric, data-driven way, for instance comparing segment $\boldsymbol{s}$ to each $\boldsymbol{s}^* \in S$ and then averaging these values? When allowing for ambiguity, a note list can map into many segmentations, and the musical quality of these solutions might vary considerably. A probabilistic, parameterized model becomes attractive provided it assigns larger probabilities to those segmentations that are musically more plausible.

Data collection and entry are expensive, so this important issue must be considered. The more *complex* a model, i.e. the more parameters it contains, the more likely it is to *overfit*, which is to say that it predicts the behavior of $\mathcal{S}$ very closely yet fails to generalize well on unseen examples. Overfitting typically occurs when a model is fit using too little data (Bishop 1995), and often non-parametric methods are very sensitive to this because their complexity is not restricted by a parametric form. We are concerned about overfitting because the data in our reference set seems to be *incomplete*, meaning that it does not adequately represent all reasonable segmentations. A strong argument for this incompleteness comes from the histograms in Figs. 1-3. In each case, if a single segmentation had been withheld, bins could become empty. Consequently, unless we are careful when building a probabilistic model from such data, we might easily derive a model that could map a musically reasonable segmentation onto a fitness value of zero. We handle this issue by *smoothing*, adding an additional, small value ϵ to every bin and then renormalizing.

5.1 An Evaluation Example

In Table 1, we present a simple, musically motivated example for a 12-note segmentation task. In the 2nd column of rows (i)-(iii), a reference set of three segmentations is depicted, two of which are identical. This set would arise when two of three musicians disagreed about whether or not a repeated melodic line began on an upbeat or an adjacent downbeat. The top stave in Fig. 1 presents one example of such *positional parallelism* for adjacent locations. In rows (a)-(d) of Table 1, four novel segmentations are given. These were chosen because they were musically less reasonable given the reference set $\mathcal{S}$. Example (d) provides a baseline worst case, demonstrating how a model assesses a ridiculous solution. Musically, the positional parallelism that characterizes the reference solutions justifies the inferiority of examples (a)-(c). Assuming that segments of length one are exceptional, examples (b) and (c) are also musically less plausible than (a). The remaining columns display the values assigned to each example under a particular fitness evaluation model. These models will now be described and considered in light of these examples. Since the fitness values for the different models grow at different rates between zero and one, they will only be directly compared per-column. Only the differences between these corresponding rankings are relevant across columns.

Table 1. An ambiguous evaluation example

		Segmentations	fit($\boldsymbol{s}$|PM)	fit($\boldsymbol{s}$|AFM)	fit($\boldsymbol{s}$|PLM)
$\mathcal{S}$	(i)	100010001000	0.24	0.77	0.13
	(ii)	100010001000	0.24	0.77	0.13
	(iii)	100100010000	0.16	0.55	0.031
	(a)	100100001000	0.19	0.66	0.037
	(b)	100110000000	0.19	0.66	0.0099
	(c)	100110011000	0.17	0.75	0.00071
	(d)	111111111111	0.02	0.4	0.000075

5.2 Notation

For convenience, two transformations of segmentation vector $\boldsymbol{s}$ are introduced. Each element in *segmentation index vector* $\boldsymbol{s}' = (s'_1, \ldots, s'_k)$ identifies the first position of a segment, that is $s'_j = i$ if and only if $s_i = 1$. Variable k refers to the total number of segments in solution $\boldsymbol{s}$. The *segmentation length vector* $\boldsymbol{l} = (l_1, \ldots, l_k)$ is calculated as the number of notes in each segment. For $j \in \{1, \ldots, k-1\}$, we set $l_j = s'_{j+1} - s'_j$; for $j = k$, we set $l_k = n + 1 - s'_k$. Three histograms, which summarize various average features in the reference set, are defined in Table 2. The sums range over $\boldsymbol{s} \in \mathcal{S}$; k and $\boldsymbol{l}$ depend on the current segment $\boldsymbol{s}$ being summed; $\mathbb{1}$ is the indicator function. $H_S(i)$ refers to bin i in histogram H_S and indicates how often any musician began a segment at location i. $H_S(1)$ is not included because, according to our task definition, the first segment always starts at position one, which does not provide any relevant information. Thus, to construct H_S for Example 1, use the counts in bins 2 through 52 in Fig. 1. The notation of H_L and H_N are similar. $H_L(i)$ indicates how often any musician chose a segment of length i, and $H_N(i)$ denotes how many musicians used i segments in their segmentation.

In our probabilistic models, histograms are always smoothed. For instance, the position histogram in Table 1 yields

$$\begin{aligned} H_S \ = (\ & 0.318, 0.005, 0.005, 0.109, 0.214, 0.005, \\ & 0.005, 0.109, 0.214, 0.005, 0.005, 0.005\ ). \end{aligned}$$

when smoothed with $\epsilon = 0.05$.

Table 2. Histogram definitions

Histogram	Definition	Range	Normalization
H_S	$H_S(i) = \frac{1}{C_S} \sum_{\boldsymbol{s}} s_i$	$i \in \{2, \ldots, n\}$	$C_S = \sum_{\boldsymbol{s}} (k-1)$
H_L	$H_L(i) = \frac{1}{C_L} \sum_{\boldsymbol{s}} \sum_{j=1}^{k} \mathbb{1}(l_j = i)$	$i \in \{1, \ldots, n\}$	$C_L = \sum_{\boldsymbol{s}} k$
H_N	$H_N(i) = \frac{1}{C_N} \sum_{\boldsymbol{s}} \mathbb{1}(k = i)$	$i \in \{1, \ldots, n\}$	$C_N = \lvert\mathcal{S}\rvert$

5.3 The Position Model

A very simple but naive approach for combining the reference set into a parametric model assumes that histogram H_S alone is sufficient in defining how likely a particular segmentation is. We refer to this approach as the *Position Model* (PM) because no higher-level structural information, such as segment length, is considered. A segment's probability or likelihood determines its fitness:

$$\text{fit}(\boldsymbol{s}|\text{PM}) = Pr(\boldsymbol{s}|H_S)^{\frac{1}{k}} \propto \left(\prod_{i=2}^{k} H_S(s'_i)\right)^{\frac{1}{k}}. \tag{2}$$

This model assumes that $\boldsymbol{s}$ was generated by a Multinomial distribution that prefers bin i with probability $H_S(i)$. The proportional symbol $\propto$ arises because the probability distribution has not been normalized. Provided only solutions with a fixed number of segments k are being compared, normalization can be ignored (each scales by the same divisor). When solutions are composed of different ks, however, the power of $\frac{1}{k}$, which computes the *geometric mean*, is needed to remove the dependence of a solution's likelihood on k.[7]

The third column in Table 1 illustrates the behavior of PM and motivates why considering only the positions of $\mathcal{S}$ provides a musically inadequate assessment. First, note that example (d) would have had a fitness of zero if no smoothing had been used ($\epsilon = 0.05$). Also note that the rankings of examples (a)-(c) are problematic, for they all rank higher than segment (iii). Examples (a) and (b) are undifferentiated even though the latter contains an implausible segment of length one. While this behavior sheds light on how PM generalizes – an ability that is especially important given our belief that $\mathcal{S}$ is incomplete – the way it generalizes is musically inadequate. One problem with Equation 2 is that it only considers where segments begin ($s_i = 1$), which is reminiscent of only considering the number of true positives. As a consequence, both the length of the individual segments and the number of segments within a solution are ignored. Regardless of these deficiencies we introduced this model because it is all-too-easy to mistakenly assume that H_S adequately quantifies a melody's ambiguity. For example, at first we assumed that, in terms of LBDM's performance, ambiguity could be explicitly modeled by considering H_S and $\boldsymbol{w}$ together. The problems we encountered are similar to those described above.

5.4 Average F Score Model

The *Average F score Model* (AFM) improves on PM by considering additionally the false negatives and positives in its calculation:

$$\text{fit}(\boldsymbol{s}|\text{AFM}) = \text{AFM}(\boldsymbol{s}) = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{s}^* \in \mathcal{S}} \text{F}(\boldsymbol{s}, \boldsymbol{s}^*). \tag{3}$$

[7] The geometric mean rescales the probability mass contained in a k-sided volume into the length of one side of the corresponding k-sided cube, providing a probabilistic measure that is decoupled from length. Without this term, solutions with larger k are usually significantly smaller than solutions with smaller k.

Because AFM is obtained directly by averaging over all pairs of F scores between $\boldsymbol{s}$ and each reference segmentation, it is non-parametric.

At first glance, one might expect this model to grasp positional parallelism because the locations of each reference solution's zeros and ones are considered together. Upon deeper consideration, however, it can be shown that this parallelism is not adequately handled. For example, even though example (iii) results from (i) by sliding its boundaries one position to the left, one can prove that their musically unrelated combination, i.e. example (c), has a higher average F score than reference segmentation (iii) (Höthker et al. 2002). The primary problem with AFM is that an error occurring on one boundary is treated independently from all other boundary errors, i.e. a segment's length is never explicitly considered. Another disadvantage is that AFM does not model a segment's likelihood and thus cannot be used to generate new "typical" solutions stochastically. This inability is a real handicap because simulation provides valuable insight into a model's behavior.

5.5 Position Length Model

In the *Position Length Model* (PLM), we combine the musicians' preferences for certain local position boundaries, a solution's segment lengths, and the total number of segments. The benefit of this approach is that it incorporates higher-order information, which allows us to model ambiguity simultaneously at different levels of abstraction. Likelihood is estimated as:

$$\begin{aligned} \text{fit}(\boldsymbol{s}|\text{PLM}) &= Pr(\boldsymbol{s}, \boldsymbol{l})^{\frac{1}{k}} && (4)\\ &\approx Pr(\boldsymbol{s}, \boldsymbol{l}|H_S, H_L, H_N)^{\frac{1}{k}} && (5)\\ &\propto H_N(k) \left(H_L(l_k) \prod_{j=2}^{k} H_S(s'_j) H_L(l_{j-1}) \right)^{\frac{1}{k}} && (6) \end{aligned}$$

The first equality indicates that the model is based on the joint distribution of both a solution's position and its length. Again, the geometric mean decouples the dependence between fitness and the number of segments in $\boldsymbol{s}$. The first approximation assumes that this joint can be described adequately by the empirical distributions estimated for the reference set. The second approximation makes two important assumptions: that the likelihood of each pair (s_j, l_{j-1}) is independent of all others (hence the product over j); and that the likelihood of each pair is independent in length and position (hence each pair of $H_S(s'_j)$ and $H_L(l_{j-1})$ terms).

The behavior of PLM ($\epsilon = 0.05$) is shown in the fifth column of Table 1. As in the previous models, the musically most plausible segmentation (a) has the highest likelihood among those not contained in the reference set. This segmentation's fitness is also higher than the likelihood of reference segmentation (iii), which demonstrates that positional parallelism is not fully captured by PLM. In contrast to the previous models, however, segmentations with unlikely lengths

(i.e. (b) and (c)) have significantly smaller fitness values than the segmentations with more plausible lengths.

In Fig. 1, PLM solutions with different numbers of segments ($k = 7$ through 12) are shown. These examples demonstrate how PLM can accommodate interpretations of different granularities. These solutions were calculated by determining which segmentation was most likely for a given k. Their boundaries line up with those chosen by the musicians. Interestingly, as k ranges from 7 to 12, the segments are arranged hierarchically, i.e. the solution for a larger k contains the solution for a smaller k within it. This behavior is fairly typical. Due to lack of space, another important aspect of this model is not displayed in Fig. 1. For a fixed k, PLM can also be sampled (rather than just reporting the most likely solution). When this is done, multiple solutions are obtained and therein the model's preference for position-parallel solutions is further evidenced.

6 Conclusions and Future Directions

The data we collected from the musicians suggests that ambiguity is an inherent part of the segmentation task and should not be ignored when evaluating an algorithm's performance. We argue that the Position Length Model provides a more compelling fitness evaluation for handling musical ambiguity than some simpler models that do not consider the positions, lengths, and number of segments together. PLM maps an algorithm's solution into a fitness value that defines how probabilistically similar it is to the musicians' solutions. Such insight can provide guidance about what aspects of a segmentation algorithm's behavior are most problematic.

Ambiguity also motivates a natural extension to the LBDM and Grouper algorithms: these algorithms should report fitness values for their segmentations so that these could be compared to those reported by the musician model. Although LBDM does output boundary strength vector $\boldsymbol{w}$, the issues that arose in the Position Model demonstrate why this information alone is insufficient for constructing an ideal fitness value. Grouper should also report how fit its solutions are, and since, during optimization, this algorithm explicitly considers segment length and position, perhaps the penalties it assigns to individual segments could be used to construct a fitness. In addition, these values might be used to construct a $\boldsymbol{w}$-like vector, which would provide insight into Grouper's behavior in complex situations (cf. the Bach fugue). More generally, it is worth recasting these algorithms within a probabilistic framework because then one can generate "typical" solutions stochastically, providing a valuable tool for exploring the model's underlying behavior systematically.

This realization brings us back to DOP, whose inherent probabilistic basis should result in a straightforward sampling procedure and likelihood calculation. Currently, however, DOP has only been trained on segmentations from the Essen collection, which presents a single (as opposed to multiple, ambiguous) solution for each folk song. Thus, DOP's performance should be explored in more explicitly ambiguous settings. DOP also suffers from a major practical problem:

it has great need for hand-segmented data. Perhaps the most promising future direction for algorithmic segmentation is to combine Grouper's and LBDM's musically relevant features into a probabilistic, machine learning based method in order to reduce the amount of data needed to configure the model.

Acknowledgments

We are grateful to the musicians who took the time to segment our corpus, and to Rens Bod, Emilios Cambouropoulos, Ewa Dahlig, Ralf Schoknecht, and three anonymous reviewers for their valuable input. This research was supported by the Klaus Tschira Foundation and the German-American Fulbright Commission. For a more complete acknowledgment and a listing of the musician data, see http://i11www.ira.uka.de/~musik/segmentation.

References

R. Bellman (1957). *Dynamic Programming*. Princeton University Press.

C. M. Bishop (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

R. Bod (2001). Memory-based models of melodic analysis: challenging the gestalt principles. *Journal of New Music Research*, 30(3):In Press.

E. Cambouropoulos (1998). Musical Parallelism and Melodic Segmentation. In *Proceedings of the XII Colloquium on Musical Informatics*, Gorizia, Italy, pages 111-114.

E. Cambouropoulos (2001). The *Local Boundary Detection Model (LBDM)* and its application in the study of expressive timing. In: *Proceedings of the International Computer Music Conference (ICMC)*, Havana, Cuba, pages 290-293.

T. Crawford, C. S. Iliopoulos, and R. Raman (1998). String-matching techniques for musical similarity and melodic recognition. In: W. B. Hewlett and E. Selfridge-Field, editors, *Melodic Similarity: Concepts, Procedures, and Applications*, pages 73–100. MIT Press.

E. Dahlig (2002). Personal communication.

K. Höthker, B. Thom, and C. Spevak (2002). Melodic segmentation: evaluating the performance of algorithms and musicians. Technical Report 2002-3, Faculty of Computer Science, University of Karlsruhe.

F. Lerdahl and R. Jackendoff (1983). *A Generative Theory of Tonal Music*. MIT Press.

H. Schaffrath (1995). *The Essen Folksong Collection*. D. Huron, editor. Center for Computer Assisted Research in the Humanities, Stanford, California.

D. Temperley (2001). *The Cognition of Basic Musical Structures*. MIT Press.