MULTI-STRATEGY SEGMENTATION OF MELODIES

Marcelo Rodríguez-López
Utrecht University
m.e.rodriguezlopez@uu.nl

Anja Volk
Utrecht University
a.volk@uu.nl

Dimitrios Bountouridis
Utrecht University
d.bountouridis@uu.nl

ABSTRACT

Melodic segmentation is a fundamental yet unsolved problem in automatic music processing. At present most melody segmentation models rely on a 'single strategy' (i.e. they model a single perceptual segmentation cue). However, cognitive studies suggest that multiple cues need to be considered. In this paper we thus propose and evaluate a 'multi-strategy' system to automatically segment symbolically encoded melodies. Our system combines the contribution of different single strategy boundary detection models. First, it assesses the perceptual relevance of a given boundary detection model for a given input melody; then it uses the boundaries predicted by relevant detection models to search for the most plausible segmentation of the melody. We use our system to automatically segment a corpus of instrumental and vocal folk melodies. We compare the predictions to human annotated segments, and to state of the art segmentation methods. Our results show that our system outperforms the state-of-the-art in the instrumental set.

1. INTRODUCTION

In Music Information Retrieval (MIR), segmentation refers to the task of dividing a musical fragment or a complete piece into smaller cognitively-relevant units (such as notes, motifs, phrases, or sections). Identifying musical segments aids (and in some cases enables) many tasks in MIR, such as searching and browsing large music collections, or visualising and summarising music. In MIR there are three main tasks associated with music segmentation: (1) the segmentation of musical audio recordings into notes, as part of transcription systems, (2) the segmentation of symbolic encodings of music into phrases, and (3) the segmentation of both musical audio recordings and symbolic encodings into sections. In this paper we focus on the second task, i.e. identifying segments resembling the musicological concept of *phrase*. Currently automatic segmentation of music into phrases deals mainly with monophony. Thus, this area is commonly referred to as *melody segmentation*.

When targeting melodies, segmentation is usually reduced to identifying *segment boundaries*, i.e. locate the points in time where one segment transitions into another. [1] Computer models of melody segmentation often focus on modelling *boundary cues*, i.e. the musical factors that have been observed or hypothesised to trigger human perception of boundaries. Two common examples of boundary cues are: (a) the perception of 'gaps' in a melody (e.g. the sensation of a 'temporal gap' due to long note durations or rests) and (b) the perception of repetitions (e.g. recognising a melodic figure as a modified instance of a previously heard figure). The first cue mentioned is thought to signal the *end* of phrases, and conversely the second one is thought to signal the *start* of phrases.

Findings in melodic segment perception studies suggest that, even in short melodic excerpts, listeners are able to identify multiple cues, and what is more, that the role and relative importance of these cues seems to be contextual [3, 6]. Yet, most computer models of melody segmentation rely on a *single strategy*, meaning that they often focus on modelling a single type of cue. For instance, [4] focuses on modelling cues related only to melodic gaps, while [1, 5] aim to modelling cues related only to melodic repetitions.

In this paper we propose and evaluate a *multi-strategy system* that combines single strategy models of melodic segmentation. In brief, our system first estimates the cues (and hence the single strategy models) that might be more 'relevant' for the segmentation of a particular input melody, combines the boundaries predicted by the models estimated relevant, and then selects which boundaries result in the 'most plausible' segmentation of the input melody.

Contribution: first, we bring together single strategy models that have not been previously tested in combination; second, our evaluation results show that our system outperforms the state-of-the-art of melody segmentation in instrumental folk songs.

The remainder of this paper is organised as follows: §2 reviews music segmentation related work using multi-strategy approaches, §3 presents a theoretical overview of the proposed system, §4 describes implementation details of the system, §5 describes and discusses our evaluation of the system, and finally, §6 provides conclusions and outlines possibilities of future work.

[1] Other subtasks associated to segmentation such as boundary pairing, as well as labelling of segments, are not considered.

2. RELATED WORK

Melody segmentation models often focus on modelling a single cue (e.g. [1, 4, 5]), leaving only a handful of models that have proposed ways to combine different cues. Perhaps the best known multi-strategy model is Grouper [11], which relies on three cues: temporal gaps, metrical parallelism, and segment length. Grouper employs temporal gap detection heuristics to infer a set of candidate boundaries, and uses dynamic programming to find an 'optimal' segmentation given the candidate boundaries and two regularisation constraints (metrical parallelism and segment length). Grouper constitutes the current state-of-the-art in melodic segmentation. However, Grouper relies entirely on temporal information, and as such might have difficulties segmenting melodies with low rhythmic contrast or no discernible metric.

Another multi-strategy model is ATTA [7], which merges gap, metrical, and self-similarity related cues. In ATTA the relative importance of each cue is assigned manually, requiring the tuning of over 25 parameters. Parameter tuning in ATTA is time consuming (estimated to be ~10 mins per melody in [7]). Moreover, the parameters are non-adaptive (set at initialization), and thus make the model potentially insensitive to changes in the relative importance of a given cue during the course of a melody.

The main differences between the research discussed and ours are: (a) our system integrates single strategy models that have not been previously used (and systematically tested) in combination, and (b) our system provides ways to select which single strategy models to use for a particular melody. In §5.3.2 we compare our system to the two models that have consistently performed best in comparative studies, namely Grouper [11] and LBDM [4]. [2]

3. THEORETICAL OVERVIEW OF OUR SYSTEM

In this section we describe our system, depicted in Figure 1. In module 1, our system takes a group of single strategy segmentation models (henceforth 'cue models'), selects which might be more relevant to segment the current input melody, and combines the estimated boundary locations into a single list. In module 2, the system assesses the segmentation produced by combinations of the selected boundary candidates in respect to corpus-learnt priors on segment contour and segment length. Below we describe in more detail the input/output characteristics of our system, as well as each processing module.

3.1 Input/Output

The input to our system consists of a melody and a set of boundaries predicted by cue models. The melody is encoded as a sequence of temporally ordered note events $e = e_1, \dots e_i, \dots, e_n$. In e each note event is represented by its chromatic pitch and quantized duration (onset, offset) values. The output of our system is a set of 'optimum' boundary locations b_{opt} of length m, constituting

[2] The manual tuning feature of ATTA made it impossible to include it in our evaluation.

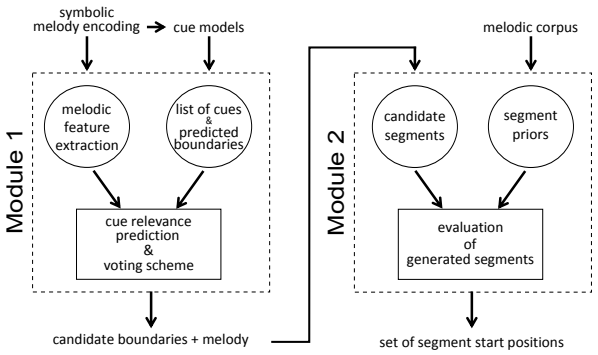

Figure 1. General diagram of our system. Within the modules ○ = input elements, and □ = processing stages.

a set of segments $S_{opt} = \{s_i\}_{1 \leq i < m}$, where each segment $s_i = [b_i, b_{i+1})$.

3.1.1 Cue Models Characteristics

Each cue model transforms e into a set of sequences, each representing a melodic attribute (e.g. pitch class, inter-onset-interval, etc.). The specific set of attribute sequences produced by each cue model used within our system is discussed in §4.2. Each cue model processes the attribute sequences linearly, moving in steps of one event, producing a *boundary strength profile*. A boundary strength profile is simply a normalized vector of length n, where each element value encodes the strength with which a cue model 'perceives' a boundary at the temporal location of the element. In these profiles segment boundaries correspond to local maxima, and thus candidate boundary locations are obtained via peak selection. The method used to select peaks is discussed in §4.2.

3.2 Module 1: Multiple-Cue Boundary Detection

Module 1 takes as input a set of features describing the melody, and a set of boundary locations predicted by cue models. Module 1 is comprised of two processing stages, namely 'cue relevance prediction' and 'voting scheme'. The first uses the input melodic features to estimate the 'relevance' of a given cue for the perception of boundaries in the input melody, and the second merges and filters the predicted boundary locations.

3.2.1 Cue Relevance Prediction

For a given set of k cue models $C = \{c_i\}_{1 \leq i \leq k}$, and a set of h features describing the melodies $F = \{f_j\}_{1 \leq j \leq h}$, we need to estimate how well a given cue model might perform under a given performance measure M as $P(M|C_i, F_j)$. In this paper we use the common F1, $precision$, and $recall$ measures to evaluate performance (see §5.2). In module 1 we focus on predicting a cue model's $precision$ (assuming high $recall$ can be achieved by the combined set of candidate boundaries).

3.2.2 Voting Scheme

Once we have estimated the relevance value of each cue model for the input melody $P(M|C_i, F_j)$, we combine the

candidate boundaries by simply adding the relevance values of candidate boundaries in close proximity (i.e. ± 1 note event apart). We assume that if boundaries from different cues are located ± 1 note event apart, one of them might be identifying a beginning and the other an end of segment, and thus for the processing in module 2 is beneficial to keep both.

The final output of this module is a single list of boundary locations b, each boundary with its own relevance value.

3.3 Module 2: Optimality-Based Segment Formation

Module 2 takes as input b, e, and length/contour [3] priors computed from a melodic corpus. The task of this module is to find the 'most plausible' set of segments S_{opt} from the space of all possible candidate segmentations. The idea is to evaluate segmentations according to two empirical constraints: one, melodic segments tend to show small deviations from a 'typical' segment length, and two, melodic segments tend to show a reduced set of prototypical melodic contour shapes. We address the task of finding the most plausible set of segments given these two constraints as an optimisation problem. Thus, for a given candidate segmentation $S_c = \{s_i\}_{1 \leq i < t}$, derived from a subset of t candidate boundaries $c \in b$, where $s_i = [c_i, c_{i+1})$, our cost function is defined as:

$$C(S_c) = \sum_{i=1}^{t-1} T(s_i) \tag{1}$$

with

$$T(s_i) = \Phi(s_i) + \alpha(\Upsilon(s_i) + \Psi(s_i)) \tag{2}$$

Where,

- $\Phi(s_i)$ is the cost associated to each candidate boundary demarcating s_i (i.e. the inverse of the relevance value of each candidate boundary).
- $\Upsilon(s_i)$ is a cost associated to the deviation of s_k from an expected phrase contour. The cost of $\Upsilon(s_i)$ is computed as $-log(\cdot)$ of the probability of the contour of the candidate phrase segment s_i.
- $\Psi(s_i)$ is a cost of the deviation from the length of s_i from an expected length. The cost of $\Psi(s_i)$ is computed as $-log(\cdot)$ of the probability of the length of the candidate phrase segment s_i.
- α is a user defined parameter that balances the boundary related costs against the segment related costs.

Details for the computation of S_{opt} and priors on segment length/contour are given in §4.4.

[3] Melodic contour can be seen as an overall temporal development of pitch height

4. SYSTEM IMPLEMENTATION

In this section we first describe the selection and tuning of the cue models used within our system, then provide some details on the implementation of modules 1 and 2.

4.1 Cue Models: Selection

We selected and implemented four cue models based on two conditions: (a) the models have shown relatively high performance in previous studies, (b) the cues modelled have been identified as being important for melody segmentation within music cognition studies. All implemented models follow the same processing chain, described in §3.1, i.e. each model derives a set of melodic attribute sequences, processes each sequence linearly, and outputs a boundary strength profile bsp. Below we list and briefly describe the cue models used within our system.

CM1 - gap detection: Melodic gap cues are assumed to correspond to points of significant local change, e.g. a pitch or duration interval that is perceived as 'overly large' in respect to its immediate vicinity. We implemented a model of melodic gap detection based on [4]. The model uses a distance metric to measure local change, [4] and generates a bsp where peaks correspond to large distances between contiguous melodic events. Large local distances are taken as boundary candidates.

CM2 - contrast detection: Melodic contrast cues are assumed to correspond to points of significant change (which require a mid-to-large temporal scale to be perceptually discernible), e.g. a change in melodic pace, or a change of mode. We implemented a contrast detection model based on [9]. The model employs a probabilistic representation of melodic attributes and uses an information-theoretic divergence measure to determine contrast. The model generates a bsp where peaks correspond to large divergences between attribute distributions representing contiguous sections of the melody. The model identifies boundaries by recursively locating points of maximal divergence.

CM3 - repetition detection: Melodic repetition cues are assumed to correspond to salient (exact or approximate) repetitions of melodic material. We implemented a model to locate salient repeated fragments of a melody based on [5]. The model uses an exact-match string pattern search algorithm to extract repeated melodic fragments, and includes a method to score the salience of repetitions based on the length, frequency, and temporal overlap of the extracted fragments. The model generates a bsp where peaks correspond to the starting points of salient repetitions.

CM4 - closure detection: Tonal closure cues are assumed to correspond to points where an ongoing cognitive process of melodic expectation is disrupted. One way in which expectation of continuation might be disrupted is when a melodic event following a given context is unexpected. We implemented an unexpected-event detection model based on [8]. [5] The model employs unsupervised probabilistic

[4] The model employs both pitch and temporal information, but in our tests only temporal information is used

[5] Our implementation is however less sophisticated than that of [8], as it requires the user to provide an upper limit for context length (specifed

learning and prediction to measure the degree of unexpectedness of each note event in the input melody, given a finite preceding context. The model generates a bsp where peaks correspond to significant increases in (information-theoretic) surprise. Candidate boundaries are placed before surprising note events.

4.2 Cue Models: Tuning

We tuned the cue models used within our system to achieve maximal precision. This involved a selection of melody representation (choice of melodic attribute sequences to be processed), [6] tuning of parameters exclusive to the cue model, and choice and tuning of a peak selection mechanism.

The choice of attribute sequence selection and parameter tuning per cue model is listed in Table 1. The abbreviations of melodic attributes correspond to: `cp`: chromatic pitch, `ioi`: inter onset interval, `ooi`: onset to offset interval, `cpiv`: chromatic pitch interval, `pcls`: pitch class. To select peaks as boundary candidates, we experimented with several peak selection algorithms, settling for the algorithm proposed in [8]. [7] This peak selection algorithm has only one parameter k. The optimal values of k for each cue model are given in the rightmost column of Table 1. We also provide details on the choice of parameters exclusive to each cue model, for an elaboration on their interpretation we refer the reader to the original publications.

Cue model	attribute sequence set	parameters
CM1	{`cpiv`, `ioi`, `ooi`}	$k = 2$
CM2	{`pcls`, `ioi`}	-
CM3	{`cp`, `ioi`}	$F = 3$
		$L = 3$
		$O = 1$
		$k = 3$
CM4	{`cp`, `pcls`, `cpiv`}	PPM-C, with exclusion
		STM: order 5
		LTM: order 2
		LTM: 400 EFSC melodies
		$k = 2.5$

Table 1. Attributes and parameter settings of cue models.

4.3 Module 1: Predictors and Feature Selection

To evaluate cue relevance prediction, we first select a subset of 200 boundary annotated melodies from the melodic corpora used in this paper (see §5.1), and then run the cue models to obtain precision performance values for each melody. To allow an estimation of precision we partition its range into a discrete set of categories. [8]

To determine cue relevance prediction, we experimented with several off-the-shelf classifiers available as part of *Weka* [9] . We selected features using the common *BestFirst* with a 10-fold cross validation. The selected features were those used in all folds.

The melodic features used to predict precision by the classifiers where taken from the *Fantastic* [10] and *jSymbolic* [11] feature extractor libraries, which add up to over 200. After selection, 17 features are kept: 'melody length', 'pitch standard deviation, skewness, kurtosis, and entropy', 'pitch interval standard deviation, skewness, kurtosis, and entropy', 'duration standard deviation, skewness, kurtosis, and entropy', 'tonal clarity', 'm-type mean entropy', 'm-type Simpson's D', 'm-type productivity' (please refer to the Fantastic library documentation for definitions).

The classifiers we experimented with are Sequential Minimal Optimization (*SMO*, with the radial basis function kernel), K-Nearest Neighbours (*K**) and Bayesian Network (*BNet*). To evaluate each classifier we use 10-fold cross validation. The classifier with the best performance-to-efficiency ratio is *SMO* for models `CM2-CM4`, with an average accuracy of 72.21%, and the simple *K** for `CM1` with an average accuracy of 66.37%.

4.4 Module 2: Computation of Priors and Choice of α

To compute the optimal sequence of segments S_{opt} we minimise the cost function in Eq. 1 using a formulation of the Viterbi algorithm based on [10]. The minimisation of Eq. 1 is subject to constraints on segment contour and segment length, and to a choice for parameter α. We tuned α manually (a value of 0.6 worked best in our experiments). To model constraints in segment contour and segment length we use probability priors. Below we provide details on their computation.

A prior $P(contour(s_k))$ is computed employing a Gaussian Mixture Model (GMM). Phrase contours are computed using the polynomial contour feature of the Fantastic library. A contour model with four nodes was selected. The GMM (one Gaussian per node) is fitted to contour distributions obtained from a subset of 1000 phrases selected randomly from the boundary annotated corpora used in this paper (see §5.1).

A prior of segment length $P(l_k)$ is computed employing a Gaussian fitted to a distribution of lengths obtained from the same 1000 phrase subset used to derive contours.

5. EVALUATION

In this section we describe our test database and evaluation metrics, and subsequently describe experiments and results obtained by our system. A prototype of our system was implemented using a combination of Matlab, R, and Python. Source files and test data are available upon request.

as the Markov order in Table 1).

[6] While some cue models, e.g. [4, 11] have already a preferred choice of melodic attribute representation, the other cue models used within our system allow for many choices, and where thus selected through experimental exploration.

[7] This algorithm proved to work better than the alternatives for all models but CM3, for which its own peak selection heuristic worked best.

[8] In our experiments we used a set dividing a model's precision into two categories (1:*poor*, 2:*good*). The exact mapping $precision : [0, 1] \rightarrow \{1, 2\}$ was selected manually for each cue model, to ensure a sufficient number of melodies representing each performance category is available for training.

[9] `http://www.cs.waikato.ac.nz/ml/weka/`

[10] `http://www.doc.gold.ac.uk/~mas03dm/`

[11] `http://jmir.sourceforge.net/jSymbolic.html`

5.1 Melodic Corpora

To evaluate our system we employed a set of 100 instrumental folk songs randomly sampled from the Liederenbank collection [12] (LC) and 100 vocal folk songs randomly sampled from the German subset of the Essen Folk Song Collection [13] (EFSC). We chose to use the EFSC due to its benchmark status in the field of melodic segmentation. Additionally, we chose to use the LC to compare the performance of segmentation models in vocal and non-vocal melodies. [14]

The EFSC consists of ∼6000 songs, mostly of German origin. The EFSC data was compiled and encoded from notated sources. The songs are available in EsAC and **kern formats. The origin of phrase boundary markings in the EFSC has not been explicitly documented (yet it is commonly assumed markings coincide with breath marks or phrase boundaries in the lyrics of the songs).

The instrumental (mainly fiddle) subset of the LC consists of ∼2500 songs. The songs were compiled and encoded from notated sources. The songs are available in MIDI and **kern formats. Segment boundary markings for this subset comprise two levels: 'hard' and 'soft'. Hard (section) boundary markings correspond with structural marks found in the notated sources. Soft (phrase) boundary markings correspond to the musical intuition of two experts annotators. [15]

5.2 Evaluation Measures

To evaluate segmentation results, we encode both predicted and human-annotated phrase boundary markings as binary vectors. Using these vectors we compute the number of true positives tp (hits), false positives fp (insertions), and false negatives fn (misses). [16] We then quantify the similarity between predictions and human annotations using the well known $F1 = \frac{2 \cdot p \cdot r}{p+r}$, where precision $p = \frac{tp}{tp+fp}$ and recall $r = \frac{tp}{tp+fn}$. While the $F1$ has its downsides (it assumes independence between boundaries), [17] it has been used extensively in the field and thus allows us to establish a comparison to previous research.

5.3 Experiments & Results

In our experiments we compare our system to the melody segmentation models that have consistently scored best in comparative studies: GROUPER [11] and LBDM [4]. The first is a multi-strategy model, and the second a single strategy (gap detection) model. [18] We also compare our system to its performance when only one module is active. Additionally we compare to two naïve baselines: *always*, which predicts a segment boundary at every melodic event position, and *never* which does not make predictions.

Table 2 shows the performance results of all models over the instrumental and vocal melodic sets. We refer to our model as COMPLETE, and to the configurations when either module 1 or 2 are active as MOD1ON and MOD2ON, respectively.

We tested the statistical significance of the paired F1 differences between the three configurations of our system, the two state-of-the-art models, and the baselines. For the statistical testing we used a non-parametric Friedman test ($\alpha = 0.05$). Furthermore, to determine which pairs of measurements significantly differ, we conducted a post-hoc Tukey HSD test. All pair-wise differences among configurations were found to be statistically significant, except those between MOD1ON and MOD2ON in the vocal set and between LBDM and MOD2ON in the instrumental set. In Table 2 the highest performances are highlighted in bold.

Database	Instrumental			Vocal		
Model	$\overline{R}$	$\overline{P}$	$\overline{F1}$	$\overline{R}$	$\overline{P}$	$\overline{F1}$
COMPLETE	0.56	0.62	**0.54**	0.49	0.67	0.56
GROUPER	0.81	0.31	0.44	0.60	0.62	**0.61**
LBDM	0.57	0.49	0.45	0.56	0.55	0.52
MOD2ON	0.51	0.49	0.44	0.48	0.45	0.47
MOD1ON	0.52	0.47	0.42	0.63	0.42	0.46
always	0.06	1.00	0.09	0.08	1.00	0.12
never	0.00	0.00	0.00	0.00	0.00	0.00

Table 2. Performance of models and baselines sorted in order of mean recall $\overline{R}$, precision $\overline{P}$, and $\overline{F1}$ for instrumental and vocal melodies. The results presented in this table were obtained comparing predictions to the 'soft' boundary markings of the LC.

5.3.1 Summary of Main Results

In general, F1 performances obtained by the segmentation models in the vocal set are consistently higher than in the instrumental set. This might be simply an indication that in the instrumental set melodies constitute a more challenging evaluation scenario. However, the F1 differences might also be an indication that relevant perceptual boundary cues are not covered by the evaluated models.

In the instrumental set, COMPLETE outperforms both LBDM and GROUPER by a relatively large margin ($\geq 10\%$). In the vocal set, on the other hand, GROUPER obtains the best performance. Below we discuss the three configurations of our system (COMPLETE, MOD1ON, MOD2ON).

5.3.2 Discussion

In both melodic sets MOD1ON shows considerably higher recall than precision. These recall/precision differences agree with intuition, since the output of MOD1ON consists of the combination of all boundaries predicted by

[12] http://www.liederenbank.nl/

[13] http://www.esac-data.org

[14] Vocal music has dominated previous evaluations of melodic segmentation (especially large-scale evaluations), which might give an incomplete picture of the overall performance and generalisation capacity of segmentation models

[15] Instructions to annotate boundaries were related to performance practice (e.g. "where would you change movement of bow").

[16] The first and last boundaries are treated as trivial cases which correspond, respectively, to the beginning and ending notes of a melodic phrase. These trivial cases are excluded from the evaluation. Also, we allow a tolerance of ± 1 note event for the computation of tp.

[17] By assuming independence between boundaries aspects such as segment length and position are discarded from the evaluation

[18] For our tests we ran GROUPER and LBDM with their default settings.

the cue models, and can hence be expected to contain a relatively large number of false positives. On the other hand, MOD2ON shows smaller differences between precision and recall values, and shows higher F1 performances than MOD1ON in both melodic sets (although the difference between performances is significant only for the instrumental set). This last result highlights the robustness of the optimisation procedure driving MOD2ON. [19]

The large F1 differences between MOD1ON and MOD2ON in respect to COMPLETE suggest that segmentation at the phrase level is a perceptual process which, despite happening in 'real time' (i.e. as music unfolds itself, represented more closely by module 1), might still require repeated exposure and retrospective listening (represented more closely by module 2).

Manual examination COMPLETE reveals that, when segmenting the vocal melody set, the prediction stage of module 1 tends to overestimate the importance of cue models (i.e. it often misclassifies models as relevant when they are not). However, when altering the settings of COMPLETE so that the prediction stage of model 1 is more conservative (i.e. so that it predicts fewer boundaries), there is no significant improvement in performance. Closer analysis of these results points to a trade-off in performance, i.e. while a conservative setting increases precision (predictions have fewer 'false positives'), it also decreases recall (predictions have fewer 'correct positives'). This suggests that the prediction stage of module 1 might require estimation of cue relevance at a local level, i.e. on subsections of the melody rather than on the whole melody.

6. CONCLUSION

In this paper we introduce a multi-strategy system for the segmentation of symbolically encoded melodies. Our system combines the contribution of single strategy models of melody segmentation. The system works in two stages. First, it estimates how relevant the boundaries computed by each selected single strategy model are to the melody being analysed, and then combines boundary predictions using heuristics. Second, it assesses the segmentation produced by combinations of the selected boundary candidates in respect to corpus-learnt priors on segment contour and segment length.

We tested our system on 100 vocal and 100 instrumental folk song melodies. The performance of our system showed a considerable (10% $F1$) improvement upon the state-of-the-art in melody segmentation for instrumental folk music, and showed to perform second best in the case of vocal folk songs.

In future work we will test if the relevance of cue models can be accurately estimated for sections of the melody (and not the whole melody as it is done in this paper). This 'local' account of relevance might play a major role in improving the system's precision. Also, we will incorporate a more advanced model of prior segment knowledge of segment structure in our system. We hypothesise that a model of the characteristics of [2] could constitute a good alternative to model not only segment length and contour, but also to incorporate knowledge of 'template' phrase structure forms. Lastly, we will continue testing our model's generalisation capacity by evaluating on larger sample sizes and genres other than folk (for the latter the authors are currently in the process of annotating a corpus of Jazz melodies).

Acknowledgments: We thank Frans Wiering, Remco Veltkamp, and the anonymous reviewers for the useful comments on earlier drafts of this document. Marcelo Rodríguez-López and Anja Volk (NWO-VIDI grant 276-35-001) and Dimitrios Bountouridis (NWO-CATCH project 640.005.004) are supported by the Netherlands Organization for Scientific Research.

[19] If we consider that (with MOD1ON bypassed) the number of candidate boundaries taken as input to MOD2ON often exceeds 'correct' (human annotated) boundaries by a factor 2 or 3, then the number of possible segmentations of the melody shows an exponential increase, leading to local minima issues, and so it would be reasonable to expect a performance equal or worse than that of MOD1ON.

7. REFERENCES

[1] S. Ahlbäck. Melodic similarity as a determinant of melody structure. *Musicae Scientiae*, 11(1):235–280, 2007.

[2] R. Bod. Probabilistic grammars for music. In *Belgian-Dutch Conference on Artificial Intelligence (BNAIC)*, 2001.

[3] M. Bruderer, M. McKinney, and A. Kohlrausch. The perception of structural boundaries in melody lines of western popular music. *Musicae Scientiae*, 13(2):273–313, 2009.

[4] E. Cambouropoulos. The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC01)*, pages 232–235, 2001.

[5] E. Cambouropoulos. Musical parallelism and melodic segmentation. *Music Perception*, 23(3):249–268, 2006.

[6] E. Clarke and C. Krumhansl. Perceiving musical time. *Music Perception*, pages 213–251, 1990.

[7] M. Hamanaka, K. Hirata, and S. Tojo. ATTA: Automatic time-span tree analyzer based on extended GTTM. In *ISMIR Proceedings*, pages 358–365, 2005.

[8] M. Pearce, D. Müllensiefen, and G. Wiggins. The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. *Perception*, 39(10):1365, 2010.

[9] M. Rodríguez-López and A. Volk. Melodic segmentation using the jensen-shannon divergence. In *International Conference on Machine Learning and Applications (ICMLA12)*, volume 2, pages 351–356, 2012.

[10] G. Sargent, F. Bimbot, E. Vincent, et al. A regularity-constrained Viterbi algorithm and its application to the structural segmentation of songs. In *ISMIR Proceedings*, 2011.

[11] D. Temperley. *The cognition of basic musical structures*. MIT Press, 2004.