

An Experimental Comparison of Human and Automatic Music Segmentation

Justin de Nooijer,^{*1} Frans Wiering,^{#2} Anja Volk,^{#2} Hermi J.M. Tabachneck-Schijf^{#2}

^{*}Fortis ASR, Utrecht, Netherlands

[#] Department of Information and Computing Sciences, Utrecht University, Netherlands

¹justindenooijer@gmail.com, ²{frans.wiering; volk; h.schijf}@cs.uu.nl

ABSTRACT

Music Information Retrieval (MIR) examines, among others, how to search musical web content or databases. To make such content processable by retrieval methods, complete works need to be decomposed into segments and voices. One would expect that methods that model human performance of these tasks lead to better retrieval output.

We designed two novel experiments in order to determine (1) to what extent humans agree in their performance of these tasks and (2) which existing algorithms best model human performance. Twenty novices and twenty experts participated in these.

The melody segmentation experiment presented participants with both audio and visual versions of a monophonic melody. In real time, participants placed markers at segment borders. The markers could be moved for fine-tuning.

The voice separation experiment presented participants auditorily with a polyphonic piece. They then listened to pairs of monophonic melodies and chose from these the one that best resembled the polyphonic piece. All possible pairs were ranked.

We concluded that there is high intraclass coherence for both tasks. There is no significant difference in melody segmentation performance between experts and novices, and three algorithms model human performance closely. For voice separation, none of the algorithms is close to human performance.

I. INTRODUCTION

Music Information Retrieval (MIR) examines, among others, how to search musical web content or databases. A common scenario is to submit to a MIR-system a short, monophonic sequence of musical notes. To match such sequences to polyphonic database or web content, such content must be available in segments of a similar size and with similar properties as the query. It seems reasonable to assume that those segmenting methods that are most in accordance with human performance will result in the best ranking of retrieval output in a MIR-system.

Human listeners generally possess two functions that allow them to process a continuous stream of music into understandable segments: the ability to perceive multiple, successive tones as one coherent melodic phrase (melody segmentation) and the ability to differentiate melody notes from harmony notes (voice separation). Algorithms for mimicking these human functions have been developed by various researchers. This paper describes the methods we used to measure human performance on these two functions, and to compare human and algorithmic performance. Specifically, the experiments were designed to answer the following questions:

Q1. Is there enough agreement in human melody segmentation and voice separation perception to function as a basis for measuring algorithm performance?

Q2. Which algorithm's melody segmentation and voice separation solutions most closely represent human melody segmentation and voice separation?

In order to be able to answer the above questions, we conducted two experiments in which participants were asked to carry out melody segmentation and voice separation of tunes. We developed novel methods that do not require any formal music training, such that both experts and novices could participate in the experiments. Thus, we are more likely to approach the actual target audience of a MIR-system, which does not consist of musical experts only.

There have been a limited number of earlier evaluations of melody segmentation algorithms; this one however seems to be the first larger one which is not performed by the author of an algorithm that is part of the experiment. For the voice separation task no comparison between algorithmic and human performance is known to us. This paper concentrates on the description of the actual experiments. The algorithms and the implications of the experimental results for the design of MIR-systems are only briefly summarised: this part is more elaborately described in Nooijer (2007) and Nooijer et al. (2008).

II. MELODY SEGMENTATION

In this experiment human performance in melody segmentation—a process sometimes referred to as ‘chunking’ or ‘grouping’—was studied. The segmentations that humans generated in the experiment were compared to the segmentations of the same melodies by several prominent algorithms. In this section, we discuss the experiment's setup and summarize the results.

A. Method used and comparison against other methods

Melody segmentation tasks have been carried out in previous research using very different setups, often accessible only to music experts. For instance, the experiments on human segmentation reported by Thom et al. (2002) and Ahlbäck (2004) both involved the score of the piece. This required the participants to have formal music training in order to be able to read notation. The segmentations in Thom et al. (2002) obtained from 19 trained musicians were performed solely based on the score, while Ahlbäck presented in addition a recording to the 18 participants of his experiments. Furthermore, Thom et al. (2002) used specific music terminology by asking to indicate the beginning of a phrase or sub-phrase.

In contrast to these approaches that involve music notation, a number of studies used only audio stimuli. For instance, Palmer & Krumhansl (1987) presented 16 listeners with predetermined segments that were rated as to how complete the phrase sounded. Hence, no free choice of determining a segment was given. In the experiment by Koniari et al. (2001) 41 children were asked to press a key at a segment boundary while listening to the piece three times. In a similar setup described in Spiro & Klebanov (2006), 33 participants listened three times to a piece and identified phrase starts by key-pressing. The participants' responses were recorded for all three times. Spiro and Klebanov developed a method how to conclude from the recorded key-pressing the actual segmentation meant by the participants, since listeners' real-time responses involve latency or may even contain errors, as the listeners could not adjust a response once given.

In the design of our melody segmentation task we combined the real time assignments of indicating boundaries while listening to the piece with the possibility to adjust the responses during repeated listening. Thus, participants could indicate as precisely as possible as to where a boundary was located.

B. Participants

This and the following experiment were carried out by letting forty participants segment musical pieces; hence the experiments are considered to be statistically significant and their design satisfies the central limit theorem (De Vocht, 2002). Based on the years of formal musical education we were able to divide participants into two categories (novices and experts) using cluster analysis. The term 'expert' herein refers to a person with a musical education and/or the skills to play a piece of music, from sheet music or learned by ear, utilizing an instrument (such as piano, guitar or voice). The term 'novice' herein refers to a person with no formal musical education, nor the skills to play a musical piece. This distinction was later used to measure expert performance versus novice performance. Each category contained twenty participants. These two subgroups thus separately did not satisfy the central limit theorem, which had its implications on the statistical test we used for results analysis. Table 1 presents information on the participants' placement in the expert or novice category. Involving the additional data in the cluster analysis did not result in a different categorization of experts and novices.

Levitin (2006) argues that segmentation is actually an innate function that is further developed through one's cultural situation. Therefore, we avoided musical terminology and designed the task setup as intuitive as possible, in order to enable both experts as well as novices to successfully execute the assignments.

In addition to the innate-argument, with the inclusion of novices in the experiment, we are more likely to approach the actual composition of the target audience of a MIR-system; when commercially implemented, it is likely to attract a broad audience, ranging from music practitioners and researchers to music novices. We therefore decided to include both experts and novices in our design, and asked participants in a questionnaire presented at the end of the second experiment, how many years of formal musical education they had had.

Table 1: Participant data from questionnaire.

	Experts	Novices	Combined
Average age	27	24	25.5
Participants per gender	♀: 6 ♂: 14	♀: 9 ♂: 11	♀: 15 ♂: 25
Average music education (in years)	11.3	1.25	6.75
Frequency of listening to music	Daily	Daily	Daily
Frequency of visiting concerts	Regularly	Sometimes	Sometimes
Preferred genre	Popular music	Popular music	Popular music

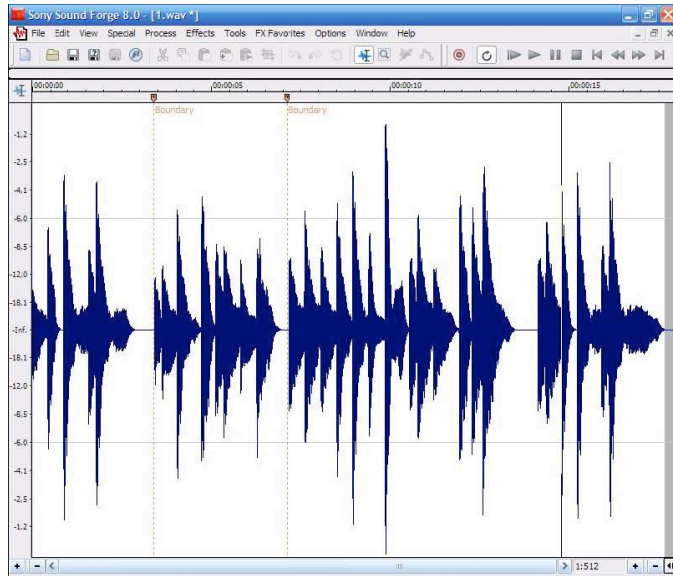
C. Materials and tools

The musical pieces for segmentation were selected from a collection of MIDI-files gathered from the Internet by a crawler. Hence, this collection contained a diverse repertoire of MIDI-files in pop-music related styles and popular classical music. We emphasised pop music, as this kind of music is most popular (hence the name) amongst the MIR system's potential group of users: the general public. The songs were selected as randomly as possible. However, for a song to be useful in our experiments, it has to satisfy three criteria: the song must contain a melody, the song's length must be at least 25 seconds, and it must be monophonic. The 25-second minimum length allows for a melody to contain several melodic segments that are to be recognised by participants in their segmentation task. We only wanted tunes with one monophonic channel, for we were only interested in finding the melody in one channel and not over multiple, changing channels throughout the entire tune. Ten tunes that satisfied these criteria were selected. We converted the selected MIDI-files to the formats needed by Sound Forge (WAVE-files). The MIDI-files were converted to WAVE-files using Steinberg Cubase 4, using the standard General MIDI piano samples.

The setup and task of this experiment are similar to Spiro's & Klebanov's (2006) experiment, in which participants pressed a key while listening to the piece. However, in our setup participants were able to fine-tune their responses. The use of Sony's Sound Forge allowed participants to place markers on the fly while the song is actually playing through the program. A cursor is displayed when the WAVE-file is playing, indicating the current location. This form of linking auditory to visual information assisted the participant to review and move placed markers. Hence, participants were allowed to move and fine-tune their markers during repeated listening. By using this iterative method, we aimed at reducing the influence of the delay between the recognition of closure and the pushing of the button to place a marker.

Figure 1 displays a screen capture of Sound Forge. The cursor (a vertical line) can be seen towards the right side of the screen. Two markers, which have been renamed to 'Boundary' for the purpose of the illustration, are displayed by dotted vertical lines towards the left side of the screen. We removed all distracting elements (a VU-meter, etc.) from Sound Forge's interface and maximized the program window, to ensure that no possible distractions were on-screen during the experiment.

Figure 1: A screenshot of Sound Forge playing a tune.



D. Algorithms

The MIDI-files could be used as direct input for the algorithms we were to benchmark. We evaluated the following melody segmentation algorithms: Temporal Gestalt Units (TGU's, Tenney & Polansky, 1980), Local Boundary Detection Model (LBDM, Cambouropoulos, 1998, 2001), Grouper (GRP, Temperley, 2001), Melodic Similarity Model (MSM, Ahlbäck 2004) and Information Dynamics (ID, Pearce & Wiggins, 2006; Potter et al., 2007). Table 2 contains a brief overview of the segmentation algorithms from which the output was used in the benchmark; for a detailed description we refer to Nooijer et al. (2008). Several algorithms could not be evaluated, as the software was not available to us.

Table 2: Properties of the selected melody segmentation algorithms. Abbreviations for the algorithms are explained in the main text. Other abbreviations: API=Absolute Pitch Interval, IOI=Inter Onset Interval, OOI=Onset to Onset Interval.

Algorithm	Features	Parameters
TGU's	API, IOI; other features can be added	Weighing
LBDM	API, IOI, OOI	Threshold, weights
GRP	IOI, OOI, Meter	Threshold, ideal length, length penalty, metrical penalties
MSM	Pitch, IOI, OOI, Metric	(None)
ID	Pitch, duration, onset, key	(Unspecified)

Several algorithms can be fine-tuned by using different parameter settings. We have used the output of LBDM utilizing three thresholds (0.4, 0.5 and 0.6); in the evaluation, these are labelled as LBDM4, LBDM5 and LBDM6.

E. Design and Procedure

Participants to this experiment were asked to divide a melody into smaller units by placing markers at locations where a segment ends. The participants received an instruction sheet that contained information on their tasks, as well as a few

guidelines derived from cognitive research (for example: 'Melody chunks contain approximately ten to twelve notes'). The experimenter did not answer any questions concerning the execution of the actual task during the experiment. Experiments followed a strict scenario, to ensure consistency. The scenario is displayed in Table 3.

Table 3: Melody segmentation experiment scenario.

Time	Participant	Experimenter
<i>Before experiment</i>		
	Sit down at table.	
		Laptop, instructions and drinks on table.
<i>Experiment</i>		
0:00 – 0:10	Reads instructions	
0:10 – 0:20	Practices segmentation task	
0:20 – 1:20 (approx.)	Performs segmentation task	
1:20 – 1:30	Fills out questionnaire	
<i>After experiment</i>		
		Thanks participant for cooperation. Packs up laptop and instructions.

Before executing the actual task, participants practiced on three well-known tunes. This gave them an idea of the task at hand and how it was to be applied to a familiar tune. The gained knowledge functioned as a base for segmenting the less-known tunes in the actual experiment.

It is more likely that participants will recognize the ending of a segment instead of the beginning of a new segment, because of the experience of closure (Snyder, 2000). Thus, asking participants to mark endings of segments relied more on their intuition than asking them to recognize the beginning of a new segment as in Spiro & Klebanov (2006). Hence, participants were asked to place markers at locations where a segment ends. A marker placed at a position where a segment ends automatically initiates the start of a new segment, which in turn is closed by the following end-marker.

After finishing the experiment, participants completed a short questionnaire related to their level of musical knowledge, the results of which were later used in statistical measures as covariates.

F. Analysis method

The data gathered from this experiment was analysed by reviewing participants' boundary placements and assigning them to the onset time of a note in the original MIDI-file. The time-codes of the markers were exported to a text-file formatted region-list (see Figure 2) for comparison to the original MIDI-files' onset-times. For this purpose, the 'Start'-column of the region-list file is of importance, as it indicates the exact time in the music file where the participant has placed a marker. We quantized the participants' timings to the appropriate MIDI-notes; algorithms also place boundaries at the onset time of a note, which makes for a good comparison. Thus, we created a profile of accumulated boundary occurrence values for each note of each melody. In theory, each note can have a maximum value that is identical to the number of participants in the

experiment. The higher the cumulative value of a certain note, the more participants agreed that this note functions as the beginning of a new segment. These values are used for evaluating the algorithms' segmentations.

Figure 2: An example of a region list containing time codes for the placed markers.

```
Regions List: 1.wav

Name   Start
-----
01     00:00:00,000
02     00:00:03,451
03     00:00:07,106
04     00:00:10,495
05     00:00:14,113
```

The profile was crated as follows. Each note of the melody was marked with a number, starting with 1 and ascending with the value of one for each following note. Thus, each melody is represented by k notes, and the notes of the melody are abstracted from their properties such as start-time, pitch and duration: we only know that note 3 follows note 2, etc. We then indicate for each note, whether or not a participant or an algorithm has placed a boundary at that note. This results in a listing, such as the one in Table 4.

Table 4: An excerpt of the melody segmentation data sheet.

Melody #	Note #	Partic. 1	Partic. m	Algo 1	Algo n
1	2	0	0	0	0
1	3	0	0	0	1
1	4	1	0	1	0
...
1	24	0	0	0	0
1	25	1	1	1	0
1	k	0	0	0	1

Having the data in this format, allowed us to perform statistical analysis on the data, to determine whether or not there is an algorithm of which the segmentation is similar to that of the participants. However, the boundary that is placed – by participants as well as algorithms – on the first note of each melody is discarded for the following reason. Since we assume all melodies on the first note to begin with or within a segment – there is no contextual data before the melody starts, thus not enough information to base the ending or beginning of a segment – including these boundaries might cause anomalies in the results. Furthermore, many rows contained nothing but zeros, meaning that not one person or algorithm has placed a boundary at the particular note represented by that row. We have conducted statistical tests with and without these rows in the datasheet, and they have no influence on the results.

Additionally, we calculated several new variables, as no statistical test is available for analysis between, for example, twenty novice variables and twenty expert variables. These variables sum all novice results, all expert results and the combined results.

G. Results and discussion

First, we had to determine the degree of inter-assessor (or intraclass) agreement (Fleiss & Cohen, 1973). This measures the actual coherence between participants' scores, and thus will help us answer research question Q1. Theoretically, boundaries

could be placed at 303 different locations. Our data contains eighty different boundary cases, meaning that throughout all melodies there were boundaries placed at eighty unique points (i.e. notes) by at least one participant or algorithm. First, we look at the levels of coherence for segmenting within the expert class, and within the novice class. We used the raw data gathered directly from the participants to conduct this test, with the following results:

$$\text{agreement}_{\text{novices}} \alpha (\text{cases}=303, n=20) = 0.9675$$

$$\text{agreement}_{\text{experts}} \alpha (\text{cases}=303, n=20) = 0.9902$$

From these results, we see that the agreement α between novices and between experts can be considered very high (α of 0.9675 and 0.9902 for respectively novices and experts, where 1 is perfect agreement). When we compare all participants in a combined class, the agreement remains high:

$$\text{agreement}_{\text{nov+exp}} \alpha (\text{cases}=303, n=40) = 0.9864$$

Thus, we conclude that the segmentation results of novices and experts do not differ significantly, and that there is enough coherence between participants to function as a basis for algorithm benchmarking. This is an interesting observation, since multiple authors state that segmentation is a highly ambiguous task (Thom et al., 2002 and Ahlbäck, 2004). The material used might be one reason for the difference. Previous researches have often used classical music. We have chosen to use popular melodies, which seem to contain clear cues about segment boundaries. However, Thom et al. (2002) compare averaged F-scores between the participants in order to illustrate the ambiguity of the task but do not measure whether these scores differ significantly from each other. Testing the significance of the difference in our experiment does not support their thesis that segmentation is highly ambiguous.

However, there are a few observations that can be made when reviewing the raw data. The judgments of experts generally show a higher overall similarity amongst participants than those of the novice participants. Spiro & Klebanov (2006) attribute this phenomenon to the fact that some tunes have stronger cues (for example, more apparent rests, larger pitch intervals or recurring rhythmical patterns) than others. Novices have to rely solely on these cues, while experts can also rely on their education, which makes them think in higher hierarchical structures. Thus, tunes with stronger local cues tend to show more cohesiveness in segmentation, while the results for tunes with weaker cues tend to be more diverse amongst novice participants.

Novices have the tendency to 'over-segment' musical pieces, sometimes creating phrases of just six notes. Upon further analysis, the novices whose results display over-segmentation often place boundaries at locations where the TGU algorithm marks a clang boundary. A clang is a lower level musical element consisting of a small number of notes. Clangs together form sequences, or segments (Tenney & Polansky, 1980). This observation confirms our assumption that education forces experts to think of a piece in hierarchical structures (Levitin, 2006 and Spiro & Klebanov, 2006), while non-musicians base their decisions on local cues in the melody. A possible

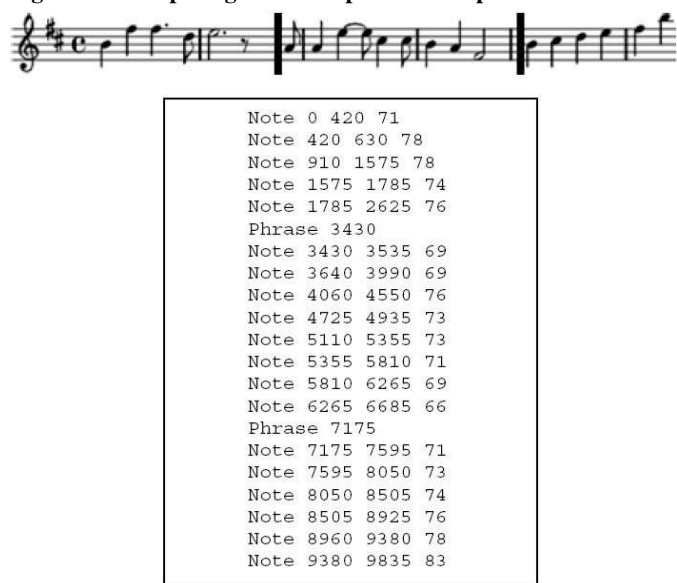
explanation for over-segmenting can thus be their lack of formal musical education.

H. Comparing algorithms against human performance.

In this section we briefly describe the results of the comparison of algorithmic output and human output (for more details see Nooijer et al., 2008).

Algorithm output was in various formats and often had to be interpreted in order to be able to match it against human segmentation. For example, Figure 3 displays a staff, and the output format of the same melody when segmented by Grouper (Temperley, 2001). Interpretation consisted of reviewing an algorithm’s output and notating it in the melody segmentation data sheet (see Table 4), similar to the way we interpreted participant data.

Figure 3: Example algorithm output for Grouper.



For comparing groups of participants to algorithm output, we use the Wilcoxon signed rank test. The Wilcoxon signed rank test is the non-parametric variant of the Student’s T-test, which is used to measure whether or not there is a significant difference between variables. Unlike the Student’s T-test, the Wilcoxon signed rank test does not require variables to be measured on an interval or ratio scale, which is of importance as the algorithm data and raw participant data are measured on a nominal scale; the data consists of zeros and ones depicting respectively ‘no boundary placed’ and ‘boundary placed’.

When the participant data are accumulated in new variables (summing all novice results, all expert results and the combined results), they are measured on a ratio scale; one can for example state that a boundary that was indicated by 34 participants is twice as strong as one that 17 participants marked. However, since the algorithm data is still measured on a nominal scale, it is necessary to rescale the variables containing the accumulated participant data to fall within the appropriate range between zero and one. Therefore, the total score for each boundary location is divided by the total number of participants accumulated in that variable: cases of the variable containing accumulated novice data are divided by twenty and the same goes for the similar expert-variable. Cases of the accumulated participant data (containing expert and novice data combined)

are divided by forty. As stated above, the Wilcoxon signed rank test measures whether or not there is a statistically significant difference between the two variables. By conducting this test, we can determine whether or not, for example, the segmentation results of novices are significantly different from the LBDM’s results. These results are displayed in Table 5. This table also contains data on how the algorithms’ outputs differ from each other.

Table 5: P-values indicating differences between algorithms and participants. Significant scores ($p < 0.025$) are shown in bold print. Abbreviations: NOV=novices, EXP=expert, ALL=all participants.

		Algorithms						
		TGU	GRP	MSM	LBDM4	LBDM5	LBDM6	ID
Algorithms	TGU		.160	.239	.071	.732	.450	.007
	GRP	.160		.007	.683	.108	.003	.157
	MSM	.239	.007		.003	.117	.655	.000
	LBDM4	.071	.683	.003		.003	.000	.276
	LBDM5	.732	.108	.117	.003		.014	.013
	LBDM6	.450	.003	.655	.000	.014		.000
	ID	.007	.157	.000	.276	.276	.000	
	Participants	NOV	.017	.085	.000	.420	.001	.000
EXP	.092	.985	.002	.643	.102	.003	.072	
ALL	.014	.128	.000	.568	.003	.000	.220	

Since there is a high intraclass agreement between experts and novices, their chunking results can function as a basis for algorithm benchmarking. Based on this benchmark, we can state that the MSM and LBDM6 algorithms differ the most from human segmentation (experts – respectively $p = 0.003$ and $p = 0.002$ – as well as novices – both with $p = 0.000$), followed by the Temporal Gestalt units algorithm (respectively $p = 0.092$ and $p = 0.017$ for experts and novices) and LBDM5 ($p = 0.001$ for novices).

We therefore can conclude that the results of LBDM4, Information Dynamics and Grouper neither differ significantly from the participants’ results, nor from each other. Hence, based on the results of this experiment, none of the three models can be selected as the best one. Thus, LBDM4, Information Dynamics and Grouper are plausible candidates for implementation in a MIR system.

III. VOICE SEPARATION

In this experiment human performance in voice separation—a process sometimes referred to as ‘melody finding’—was studied and compared to algorithmic performance of the same task. No previous experiments are known to us that compare the output of voice separation algorithms to human performance. In this section, we present the setup and results of this experiment.

A. Participants

The same group of people – consisting of twenty experts and twenty novices – participate in this experiment as in the melody segmentation experiment. The same division of experts and novices is applied to the data from this experiment.

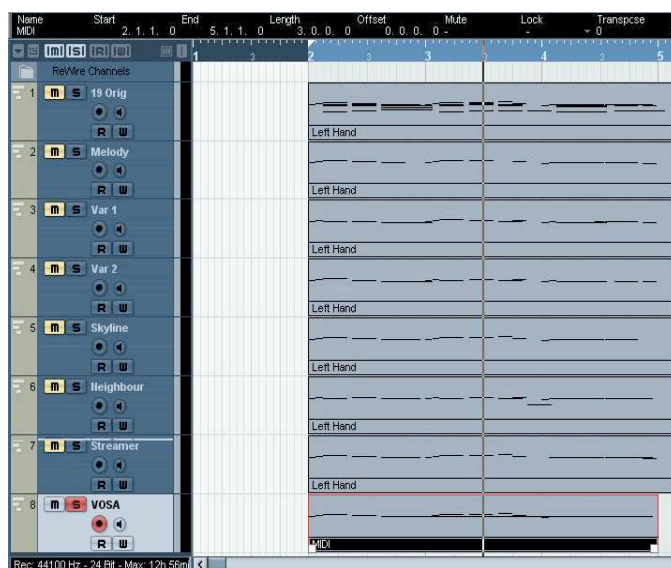
B. Materials and tools

The MIDI files used in this experiment were gathered from the Internet with a crawler, as in the melody segmentation task. For this experiment, we used eight polyphonic pieces of approximately ten seconds, which is long enough to contain a melodic unit along with some preceding and following contextual information. Melody and harmony notes are of the same timbre, since algorithms do not take timbre into account for separation; therefore, participants could have an advantage when hearing different timbres.

A set of melodic variants was created for each original piece. Each set was composed as follows. Five of the variants were derived from the output of the voice separation algorithms (discussed below). When an algorithm outputs multiple voices containing monophonic lines – sometimes up to more than ten voices – we selected the one that most accurately represents the melody for inclusion in the set of variants. If multiple monophonic lines contained similar amounts of the actual melody notes, we selected only one of these. Furthermore, the set contained an interpretation of the melody, as it was perceived by the first author of this paper. In addition, each set contained two variants consisting of randomly selected notes within the polyphonic piece. While the notes were selected at random, we introduced no new notes to the composition; every note is present at the same location in the original polyphonic piece.

The set of variants thus contained at most eight monophonic lines. However, in some cases, the output of, for example, the skyline algorithm can be the same as the author’s interpretation of the melody. In such cases, it makes no sense to include both variants. Thus, some sets contained fewer variants.

Figure 4: Screenshot of Cubase playing a variant



The participants listened to MIDI versions of the melodies, which were played using Steinberg’s Cubase 4’s MIDI playing functionalities. An added advantage of using Cubase is that it

offered the ability to load multiple tracks into different channels, playable one by one. This allowed us to load all variants of one set into the program. The variants can then be played back by ‘soloing’ the appropriate channel. Figure 4 displays a screen capture of Cubase playing the VoSA variant of a melody.

C. Algorithms

The algorithms that we compared in this experiment are Skyline (see Clausen, n.d., Uitdenbogerd & Zobel, 1998), Nearest Neighbour (NN, see Clausen, n.d.), Streamer (see Temperley, 2001), Voice Separation Analyzer (VoSA, see Chew & Wu, 2004) and Stream Separation Algorithm (SSA, see Madsen & Widmer, 2006). Table 6 lists important properties of these algorithms; for a detailed description we refer to Nooijer et al. (2008).

Table 6: Properties of the selected voice separation algorithms. Abbreviations are explained in the main text.

Name	Features	Parameters
Skyline	Pitch, onset, duration	(None)
NN	Pitch, OOI	(None)
Streamer	Pitch, onset, duration	Max. voices, max. collisions, penalties for violating preference rules
VoSA	Pitch, onset, duration	(None)
SSA	Pitch, onset	Penalties for starting notes, ending notes, inserting rests and leap size

D. Design and Procedure

Before we discuss the actual design of the experiment we briefly describe a rejected version of the voice separation experiment. In this version, the piece was presented to the participants in the Cubase environment as piano roll notation. They were asked to listen to the piece and to erase those notes from the piano roll notation that in their opinion did not belong to the melody. They could listen to the piece as often as they wanted, and deletions could be reversed. The advantage of this setup was that each participant would have only one solution for each voice separation task, whereas in the final experiment they would produce a considerable number of judgments on the qualities of different alternatives. However, even for experienced musicians the erasing task was a very challenging one, so the likelihood that the experimental results would reflect the participant’s melody perception was quite low, especially for the novices. Therefore, this version of the experiment was rejected.

The final version of the experiment was as follows. Participants were presented with a polyphonic piece. The polyphonic piece (also referred to as ‘the original’) was played, and then followed by a number of monophonic versions, of which the participant had to decide which version best represented the melody as heard in the original piece. This was done as follows. The experimenter would play the original followed by a pair of monophonic versions—for example, variant 1 and variant 2—and the participant then had to decide

which of the two variants he or she considered to be more similar to the melody heard in the polyphonic piece. The participant could again listen to the original as often as (s)he wanted before the next pair of variants was presented—for example, variant 1 and variant 3. This process is continued until each variant had been judged against the other variants. Thus variants were compared pair-wise against the original.

Table 7: Voice separation experiment scenario.

Time	Participant	Experimenter
<i>Before experiment</i>		
Sit down at table.		
		Laptop, instructions and drinks on table.
<i>Experiment</i>		
0:00 – 0:10	Reads instructions	Starts Cubase, loads sound files
0:10 – 0:20	Practices separation task	Plays sound files
0:20 – 1:20		Plays sound files, notates scores
1:20 – 1:35	(break)	
1:35 – 2:35 (approx.)	Performs separation task	Plays sound files, notates scores
<i>After experiment</i>		
	Signs payment form	Pays and thanks participant for cooperation. Packs up laptop, score sheets and instructions.

E. Analysis method

The results were analysed by combining participants' answers and assigning summed values to each variant, for each melody: by combining the variant rankings of the participants, we created a combined ranking per melody.

F. Results and discussion

First, we used Cronbach's alpha to measure the coefficient of reliability (or consistency). However, the complexity and multifaceted nature of the data gathered through this experiment – it contains multiple pairwise choices per melody per participant – prohibits it from being directly used for statistical analysis. Therefore, we reduced the data as follows: we only used the highest ranked variant (depicted in the datasheet with a corresponding number, to make statistical analysis possible) for each participant for each algorithm, hereby discarding the lower ranked algorithms. This may not give us insight into the entire dataset, but gave a clear indication on the consistency of top position rankings. A snippet of the reduced dataset can be seen in Table 8.

Table 8: Reduced voice separation dataset in SPSS example. For each melody and each participant, the number of the best-rated variant is shown.

melody	n1	n2	n3	n...	n19	n20	e1	e2	e3	e...	e19	e20
1	7	6	7	...	7	1	7	1	7	...	1	8
2	3	3	3	...	3	6	3	3	3	...	3	3
3	3	4	3	...	4	3	2	3	2	...	2	2
4	5	5	4	...	4	5	5	5	4	...	7	5
5	3	2	3	...	3	3	5	7	7	...	7	7
6	3	3	3	...	3	3	3	1	3	...	3	3
7	3	3	3	...	3	3	3	3	3	...	3	3
8	2	2	2	...	2	2	2	1	1	...	2	1

When computing Cronbach's alpha for inter-rater coherence within the novice group, within the expert group and within the entire group of participants, the results are as follows:

$$\alpha_{\text{novices}} = 0.8966$$

$$\alpha_{\text{experts}} = 0.9389$$

$$\alpha_{\text{allparticipants}} = 0.9230$$

Since these values are all high (taken into account that an α value of 0.70 is considered acceptable), we conclude that the inter-rater coherence between all groups is high. The coherence amongst experts is higher than coherence amongst novices, but both α values are high.

G. Comparing algorithms against human performance

Now that we have established that the inter-rater coherence is high, we calculate algorithm rankings based on the participants' results. The final rankings of the variants per melody are shown in Tables 9 and 10. These tables also include the non-algorithmic melody variants – named 'Author' (for the first author's interpretation of the melody), 'Var #1' and 'Var #2' (the two random variants) – that were included in each set as fillers.

Table 9: Expert voice separation rankings (* indicates algorithms with different variants, ranked at same location).

Rank	Melody 1	Rank	Melody 2	Rank	Melody 3	Rank	Melody 4
1	VoSA		Author	1	SSA		1 Skyline
2	Var #2	1	VoSA	2	Author		2 Neighbor
3	Var #1		Skyline	3	Streamer		3 Var #1
4	Streamer	2	SSA	4	VoSA		4 Author
5	SSA	3	Streamer	5	Skyline		5 VoSA
6	Skyline	4	Neighbor	6	Var #1		6 SSA
	Neighbor	5	Var #1	7	Neighbor		7 Streamer
7	Author	6	Var #2	8	Var #2		

Rank	Melody 5	Rank	Melody 6	Rank	Melody 7	Rank	Melody 8
1	Var #1		Author		Author		*1 VoSA
2	Author	1	SSA		Skyline		SSA
3	Skyline		Skyline	1	VoSA		2 Streamer
4	Streamer	2	VoSA		Streamer		3 Skyline
5	SSA	3	Neighbor		SSA		4 Var #1
6	VoSA	4	Var #2	2	Var #2		5 Neighbor
7	Neighbor	5	Streamer	3	Neighbor		6 Author
8	Var #2	6	Var #1	4	Var #1		7 Var #2

Table 10: Novice voice separation rankings (* indicates algorithms with different variants, ranked at same location).

Rank	Melody 1	Rank	Melody 2	Rank	Melody 3	Rank	Melody 4
1	Var #1	1	Author	1	Author	1	Skyline
2	Var #2		VoSA	2	Neighbor	2	Neighbor
3	VoSA		Skyline	3	SSA	3	Author
4	Streamer	2	Streamer	4	Streamer	4	SSA
5	SSA	3	SSA	5	Skyline	5	Var #1
6	Skyline	4	Neighbor	6	VoSA	6	VoSA
	Neighbor	5	Var #1	7	Var #1	7	Streamer
7	Author	6	Var #2	8	Var #2		

Rank	Melody 5	Rank	Melody 6	Rank	Melody 7	Rank	Melody 8
1	Author	1	Author	1	Author	1	SSA
2	SSA		Skyline		Skyline	2	Author
3	Var #1		SSA		VoSA	*3	Streamer
4	Skyline	2	VoSA	2	Streamer	4	Skyline
5	Streamer	3	Neighbor	2	Var #2	5	Neighbor
6	VoSA	4	Var #2	3	Neighbor	6	Var #2
7	Var #2	5	Streamer	4	Var #1	7	Var #1
8	Neighbor	6	Var #1				

Table 11. Number of times a variant is ranked first place by experts (left) and novices (right).

Experts		Novices	
Variant	# 1st	Variant	# 1st
VoSA	4	Author	5
SSA	4	SSA	3
Skyline	4	Skyline	3
Author	3	VoSA	2
Var #1	1	Var #1	1
Streamer	1	Streamer	1
Var #2	0	Var #2	0
NN	0	NN	0

Table 11 integrates the results from Tables 9 and 10 by indicating how often an algorithm’s variant is ranked first. From these results we can conclude that both novices and experts prefer the melodies generated by the SSA and Skyline algorithms. Experts additionally prefer the VoSA variants equally well, when solely considering algorithms. However, novices prefer the melody hand-segmented by the author to computer-segmented melodies. Hence, we cannot identify one optimal algorithm for the voice separation task.

IV. CONCLUSIONS AND FURTHER RESEARCH

A. Conclusions

Summarized, we can state the following in answer to our research questions Q1 and Q2.

- There is a high degree of intraclass among novices and experts; thus there is enough consistency in the results to function as a basis for algorithm benchmarking. Interclass agreement is also very high.

- For the melody segmentation task, Grouper, Information Dynamics and LBDM4 are plausible candidates for implementation in a MIR-system.
- For the voice separation task, there is no single algorithm that for the majority of the tunes matches human perception.

B. Method improvements

For the melody segmentation experiment, we chose Sound Forge as user interface. The main advantage of using Sound Forge was that it allowed participants to make up for any latency occurring between hearing a boundary and pressing the appropriate key to place a boundary. However, one could argue that Sound Forge’s method of visualization might trigger visual cues about the musical piece. Participants could then, instead of relying solely on the supplied auditory information, use these visual cues to segment the musical piece, which in turn could lead to biased results. This effect might be more apparent in some musical pieces than others, depending on the actual visual representation of the waveform. To eliminate the effect of the visualization, it would therefore be useful to develop an interface, which utilizes a custom visualization technique for displaying the waveform. This cannot be done in Sound Forge, as all manipulations on the visual representation will automatically result in a change of the auditory form. An improved visualization should have the same advantage as Sound Forge, which is that it allows the participants to apply latency correction by using the visual form. In addition, the waveform would be abstract enough to not offer any visual cues for segmentation.

The voice separation experiment has an important drawback in that it is closely coupled to the algorithms that are being tested, as the variants that are played to the participants were created by the algorithms. This means that, if one wishes to evaluate another algorithm, its variants must be added to the experiment. This effectively means that the whole human experiment must be redone. Devising a modified version of the experiment that does not suffer from this drawback is an important future goal.

C. Benchmarking

One of the most interesting questions that still needs to be answered in the context of this research and MIR performance in general, is whether a MIR-system in which melody segmentation and voice separation are done by cognition-based algorithms perform better than a MIR-system in which chunks are generated by brute-force methods.

To be able to answer this question, further research has to be conducted, which involves actual tests with MIR-systems in which these methods are implemented. For the voice separation task, Grouper, Information Dynamics and LBDM4 are good candidate algorithms. Further research has to be done on voice separation, since our experiment was unable to identify one or more algorithms as likely candidates for successful incorporation in a MIR-system. There are two possible explanations for this outcome. On the one hand, our evaluation method may have shortcomings, but on the other it is not unlikely that the present voice separation algorithms are not refined enough to model human performance.

ACKNOWLEDGMENTS

We would like to thank the authors of the algorithms who have kindly answered our questions and requests for cooperation, in alphabetical order: Sven Ahlbäck, Emiliios Cambourooulos, Elaine Chew, Søren Madsen, Marcus Pearce and David Temperley. We also thank all the participants in the experiments for their cooperation. Bas de Haas gave some valuable comments on an earlier draft of this paper.

REFERENCES

- Ahlbäck, S. (2004). *Melody beyond notes: A study of melody cognition*. Göteborg: Göteborg University.
- Cambourooulos, E. (1998). Musical parallelism and melodic segmentation. In: *Proc. of the XII Colloquium of Musical Informatics*, Gorizia, Italy, 111-114.
- Cambourooulos, E. (2001). The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing. In: *Proc. of the International Computer Music Conference*, Havana, Cuba.
- Chew, E., Wu, X. (2004). Separating voices in polyphonic music: a contig mapping approach. In: *Proc. of Computer Music Modeling and Retrieval 2004*, Esbjerg, Denmark.
- Clausen, M. (n.d.) *Melody extraction*. Retrieved 4-7-2006 from: www-mmdb.iai.uni-bonn.de/forschungsprojekte/midilib/english/
- Fleiss, J. L., Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. In: *Educational and Psychological Measurement*, 33, 613-619.
- Koniari, D., Predazzer, S., & Melen, M. (2001). Categorization and schematization processes in music perception by children from 10 to 11 years. *Music Perception*, 18, 297-324.
- Levitin, D.J. (2006). *This is your brain on music: The science of a human obsession*. New York: Dutton.
- Madsen, S. T., Widmer, G. (2006). Separating voices in MIDI. In: *Proc. of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 8-12.
- Nooijer, J. de. (2007). *Cognition-based segmentation for music information retrieval systems*. Master's thesis, Utrecht University.
- Nooijer, J. de, Wiering, F., Volk, A., Tabachneck-Schijf, H.J.M. (2008). Cognition-based segmentation for music information retrieval systems. In: C. Tsougras, R. Parncutt (Eds.). *Proceedings of the fourth Conference on Interdisciplinary Musicology (CIM08)*. Thessaloniki, Greece, 2-6 July 2008.
- Palmer, C., Krumhansl, C. (1987). Independent Temporal and Pitch Structures in Determination of Musical Phrases. *Journal of Experimental Psychology: Human perception and Performance*, 13 (1), 116-126.
- Pearce, M.T., Wiggins, G.A. (2006). The information dynamics of melodic boundary detection. In: *Proceedings of the Ninth International Conference on Music Perception and Cognition*, Bologna, 860-865.
- Potter, K., Wiggins, G.A., Pearce, M.T. (2007). Towards greater objectivity in music theory: Information-dynamic analysis of minimalist music. *Musicae Scientiae* 11(2), 295-322.
- Spiro, N., Klebanov, B. (2006). A new method for assessing consistency or real-time identification of phrase-parts and its initial application, *Proceedings of the Ninth International Conference on Music Perception and Cognition*, Bologna.
- Temperley, D. (2001). *The cognition of basic musical structures*. Cambridge, MA: MIT Press.
- Tenney, J., Polansky, L. (1980). Temporal Gestalt Perception in Music. In: *Journal of Music Theory* (24), 205-241.
- Thom, B., Spevak, C., Höthker, K. (2002). Melodic segmentation: Evaluating the performance of algorithms and musical experts. In: *Proc. of the International Computer Music Conference*. Göteborg, Sweden.
- Uitdenbogerd, A. L., Zobel, J. (1998). Manipulation of music for melody matching. In: *Proc. of the ACM Multimedia Conference '98*, Bristol, UK., 235-240.
- Vocht, A. de. (2002). *Basishandboek SPSS 11*. Utrecht: Bijleveld.