Music Structure Analysis by Finding Repeated Parts

Jouni Paulus
jouni.paulus@tut.fi

Anssi Klapuri
anssi.klapuri@tut.fi

Institute of Signal Processing
Tampere University of Technology
Korkeakoulunkatu 1, Tampere, Finland

ABSTRACT

The structure of a musical piece can be described with segments having a certain time range and a label. Segments having the same label are considered as occurrences of a certain structural part. Here, a system for finding structural descriptions is presented. The problem is formulated in terms of a cost function for structural descriptions. A method for creating multiple candidate descriptions from acoustic input signal is presented, and an efficient algorithm is presented to find the optimal description with regard to the cost function from the candidate set. The analysis system is evaluated with simulations on a database of 50 popular music pieces.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Signal analysis, synthesis, and processing*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*

General Terms

Algorithms, Theory

Keywords

music structure, structure analysis, segmentation, cost function, search algorithm, structure comparison

1. INTRODUCTION

Many musical pieces, especially in the popular music genre, consist of distinguishable parts that may repeat. The structure can be, e.g., "intro, verse, chorus, verse, chorus, chorus". Structure analysis aims to determine this kind of a description for a musical piece. Depending on the method applied, the result may consist of the temporal locations the parts and arbitrary labels assigned to them (e.g., "A, B, C, B, C, C"), whereas some systems also try to label the parts meaningfully. The information about the temporal locations of these parts and their labels form the *structural description* of the piece.

Automatic analysis of the structure has been studied mainly for the application of creating a meaningful summary of a musical piece. One of the first works operating on acoustic signals was by Logan and Chu, describing an agglomerative clustering and hidden Markov model (HMM) based approaches for key phrase generation [19]. They used mel-frequency cepstral coefficients (MFCCs) from short (26 ms), overlapping frames. The clustering method grouped the frames together iteratively until a level of stability had been reached. In the HMM method they trained an ergodic HMM with only few states, hoping that each state would represent a musical part, and used the Viterbi decoded state sequence as the description of the musical structure. The HMM approach was taken further by Aucouturier and Sandler using spectral envelope as the feature [2]. It was noted in both these studies that when using such short frames, the HMM states did not model musically meaningful parts, as was hoped. Abdallah et al increased the frame length considerably and the number of states up to 80 [1]. After acquiring the state sequence, each frame was provided with some knowledge about the surrounding context by calculating a state histogram in a 15 frame window. The histograms were then used in clustering the frames by optimising a cost function with simulated annealing. Rhodes et al added a term to control the duration of stay in a certain cluster [23], while Levy et al refined the clustering method to a context-aware variant of fuzzy C-means [18].

Another popular starting point of the analysis is to calculate frame-by-frame similarities over the whole signal, constructing a self-similarity matrix. Foote proposed to use the similarity matrix for visualising music [9]. It was noted that the parts of music having similar timbral characteristics created visible areas in the similarity matrix. The borders of these areas were sought and used in segmenting the piece in [10]. In [11] Foote and Cooper used a spectral clustering method to group similar segments.

When the used feature describes the tonal (pitch) content of the signal instead of general timbre, e.g., chroma instead of MFCCs, repetitions generate off-diagonal stripes to the similarity matrix instead of rectangular areas of high similarity. Such stripes reveal similar sequential structures, e.g., melody lines or chord progressions, instead of just denoting parts having similar timbral characteristics, or sounding the same. The two main approaches (HMM-based "state" method and "sequence" method relying on stripes in the

AMCMM'06, October 27, 2006, Santa Barbara, California, USA.

similarity matrix) were compared by Peeters [22]. He noted that as the sequence approach requires a part to occur at least twice to be found, the HMM approach would be more robust analysis method. Still, the stripes have been used in structure analysis by several authors. Bartsch and Wakefield extracted chroma from beat-synchronised frames and used the most prominent off-diagonal stripe to define a thumbnail for the piece [3]. Lu et al proposed a distance metric considering the harmonic content of sounds, and used 2D morphological operations (erosion and dilation) to enhance the stripes [20]. In popular music pieces, the clearest repeated part is often the chorus section. Goto aimed at detecting it using chroma, and presented a method for handling the musical key modulation sometimes taking place in the last in the last refrain of the piece [12].

Music tends to show repetition and similarities on different levels, starting from consecutive bars to larger parts like chorus and verse. Some authors have tried to take this into account and proposed methods operating on several temporal levels. Jehan constructed several hierarchically related similarity matrices [16]. Shiu et al extracted chroma from beat-synchronised frames and then used dynamic time warping (DTW) to calculate a similarity matrix between all the measures of the piece [24]. The higher level musical structure was then modelled with a manually parametrised HMM. Dannenberg and Hu gathered the shorter repeated parts and gradually combined them to create longer, more meaningful, parts in [8]. Later, Dannenberg used the stripes in similarity matrix to find similar musical sections, and then utilised this information to aid a beat tracker [7].

Chai proposed to take the context into account by matching two windows of frame level features with DTW. Sliding the other window while keeping the other fixed provided a method to calculate the similarity on different lags and to determine the lag of maximum similarity. Gathering this information in a matrix formed stripes of prominent lags, like the stripes in a similarity matrix. The longer stripes were then interpreted information about the repeats of structural parts [6].

Maddage et al proposed a method for analysing a musical piece combining different sources of information. They used beat-synchronised pitch class profile as the feature and detected chords with pre-trained HMMs. Using assumptions of the lengths of the repeated parts, fixed length segments were matched to get a measure of similarity. Finally heuristic rules, claimed to apply on English-language pop songs, were used to deduce the high-level structure of the piece. [21]

Like in some of the earlier publications, we are interested in the structure defined by parts that are repeated. In other words, the proposed analysis method is not interested, nor able, to find musical parts that occur only once during the piece. These remain as unexplained segments in between the repeating parts and in principle they can be labelled, but no substructure is imposed on them. This limits the information that can be extracted from music, but as mentioned, e.g., in [9, 7], musical pieces usually have a structure relying to some extent on repetitions of different parts.

This paper focuses on estimating the musical structure from the result of low-level signal analysis front-end. Formulation of the problem is given in Section 2.1. Then we propose a parametric cost function for evaluating the fitness of an description. Depending on the parameter values it allows emphasising different properties: the complexity of the description, amount of the piece left unexplained, and consistency of the found parts. The parameters provide a way of circumventing the ambiguity of the structural description by allowing controlling the desired properties of the result. In Section 2.5, we propose a computationally efficient algorithm for searching the structural description that minimises the cost function by solving the computationally expensive combinatorial problem.

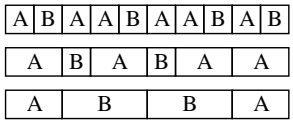

Figure 1: A simple example of three different structural descriptions for the same piece. All three are valid from the point of view of detecting repetitions.

The structural descriptions found by the system are evaluated using a database of 50 musical pieces with manually annotated structural information. The evaluation procedure and results are presented in Section 3.

As stated, e.g., in [5], the level of structural description may be different depending on the algorithm, person in question, etc. An example of the level differences on the descriptions is illustrated in Figure 1. The description on the top is the most detailed and gives the largest amount of structural information. However, that level of detail is not always needed, so the description on middle or on bottom might be adequate. Even though they do not contain so fine details, the coarse structure of the piece is more readily visible. The proposed analysis algorithm discards all hierarchical information. All operations changing the hierarchical level, i.e., combining parts always occurring in conjunction or splitting longer segments, is left for an optional post-processing stage.

2. PROPOSED METHOD

A block diagram of the proposed system can be seen in Figure 2. The front-end signal analysis consists of methods that have been utilised earlier and found to be working relatively well (musical meter analysis [17], chroma extraction [12] with multi-octave modifications, beat synchronised feature extraction [3], measure-level similarity matrix [24], signal segment border generation from timbral novelty [10], and segment matching with DTW [5]).

The front-end analysis provides features and initial set of candidate section boundaries (see Section 2.3) for the latter stages. The proposed method then creates longer segments from the blocks, and finds the optimal description of the piece with non-overlapping repeated parts in respect to the cost function defined in Section 2.1.

2.1 Defining a "Good" Structural Description

The musical piece to be described is subdivided into *blocks* $b_1, b_2, \ldots, b_N$, for example, at the times of bar lines. Each block is described uniquely by its start and end times, $b_j = (\tau_j, t_j)$, where τ_j is the start time and t_j is the end time. A *segment* s_k is a part of the piece consisting of one or more consecutive blocks, $s_k = b_i, b_{i+1}, \ldots, b_j$, i.e., an occurrence of a structural part. A *segment group* g_i is a set of non-overlapping segments, and represents the multiple occurrences of one structural part. These concepts are illustrated in Figure 3.

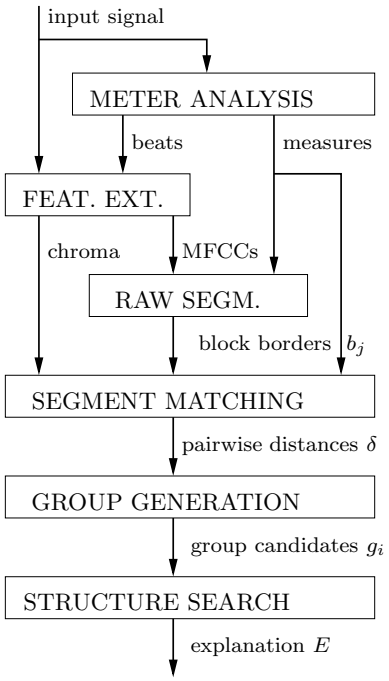

Figure 2: A block diagram of the whole system.

A segment group is characterised by properties termed *within-group dissimilarity* d_i, and *coverage* c_i. The first one describes how much the segments in a group differ from each other, and is detailed in Section 2.4. The coverage of a group c_i describes how much of the whole piece it covers,

$$c_i = \frac{\sum_{s_k \in g_i} \sum_{b_j \in s_k} t_j - \tau_j}{\sum_{j=1}^{N} t_j - \tau_j}. \quad (1)$$

All segments groups form a set of *segment group candidates* $G = \{g_1, g_2, \ldots, g_M\}$. An *explanation* E of a piece is a subset of the group candidates, $E \subset G$, such that the segments of the included groups do not overlap in time. If some part of the piece is not covered by the explanation E, it remains unexplained.

Given an explanation E of the structure of the piece, its fitness or cost can be calculated. For this purpose, we propose a cost function consisting of three terms as follows:

$$C = \text{dissimilarity} + \alpha \text{ unexplained} + \beta \text{ complexity}, \quad (2)$$

where α is a weighting factor for the cost due to unexplained blocks in the piece, and β is a weighting factor for the complexity (number of structural parts g_i) of the explanation. The first term penalises groups that have segments differing from each other, i.e., a group should consist only of the occurrences of a certain part. The second term defines the cost for the unexplained parts of the piece: the larger the explained proportion is, the smaller the cost will be. The last term defines the cost of the complexity of the explanation. The smaller the number of groups in an explanation $|E|$, the better. The effect of the weight parameters is illustrated in Figure 4, which contains the annotated structure of a piece and analysis results with three different parameter values.

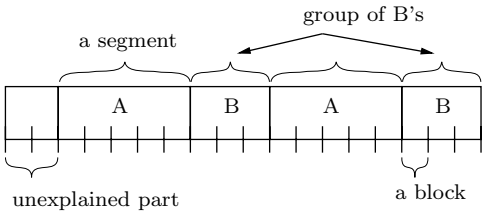

Figure 3: A piece is divided into blocks. Consecutive blocks form segments. Segments with same label create groups. Some part of the piece may be left unexplained.

The cost (2) is evaluated as

$$\begin{aligned} C(E) &= \sum_{g_i \in E} d_i c_i + \alpha(1 - \sum_{g_i \in E} c_i) + \beta \log(1 + |E|) \\ &= \alpha + \sum_{g_i \in E} c_i(d_i - \alpha) + \beta \log(1 + |E|). \end{aligned} \quad (3)$$

Structure analysis can now be viewed as an optimisation task to minimise the cost (3), and an algorithm for this purpose is presented in Section 2.5. It operates on segment groups and does not depend on how the groups are formed, there are several ways how the piece can be divided into blocks b_j, how the blocks are combined into segments s_k, and how the dissimilarities d_i of segment groups are calculated. Before describing the optimisation method, we present the way we used to perform these tasks.

2.2 Front-End Signal Analysis

As noted, e.g., in [24, 5, 4], the result of musical structure analysis may improve if the low-level analysis is done in synchrony with the metrical structure of the piece. The proposed system initially estimates the time-varying period and phase of the metrical levels: beat and musical measure. The estimation is done with the method presented in [17], utilising a bank of comb filter resonators and a probabilistic model to determine the periods and phases. The periods are here halved to minimise the problems due to possible *pi*-phase errors in the estimated pulses, which cause the pulses to be shifted by half a period from the correct locations.

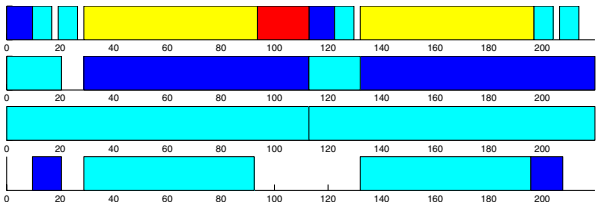

Figure 4: An illustration of the effect of the cost function parameter values. The top panel is the ground truth annotation of the structure of the piece. The lower panels contain example analysis results with varying values of the parameters α and β. The second panel shows the result with "reasonable" parameter values. Increasing the complexity weight β leads to the result in the third panel, and by decreasing the weight of unexplained parts, the result of the bottom panel was obtained.

The acoustic signal is then divided into frames according to the beat pulse, i.e., a frame is taken between two beat occurrences. A 36-dimensional chroma vector is extracted from each frame. Normally, the chroma vector is 12-dimensional, corresponding to the 12 pitch classes on the Western musical scale and describing the amount of energy present in all the frequency bands corresponding to occurrences of the pitch classes, ignoring the absolute height.

Here, we extract the chroma feature from three frequency ranges ($63.5 - 254$ Hz, $254 - 1020$ Hz, and $1.02 - 4.07$ kHz) using the chroma calculation method from [12] in each range separately. The results are concatenated into a feature vector. The use of three separate frequency regions gives more information about the tonal contents of the signal. This is roughly similar to [20], where the chroma was extracted from a range of three octaves and the pitch class equivalences were not summed together, resulting in a 36-dimensional feature vector. Principal component analysis is applied to reduce the feature vector dimensionality and only the 15 largest components are retained.

In addition to the chroma, 13 MFCCs are calculated with the same frames as the chroma values. The zeroth coefficient which is often discarded, is also retained. For each measure, the mean and variance of the MFCCs from the beat frames within it are calculated. The mean and variance values are concatenated to yield a 26-dimensional feature vector, and the statistics vectors are normalised to zero mean and unity variance over the whole piece.

2.3 Raw Segmentation

After extracting the features from the signal, a rough segmentation is employed to create a candidate set of the locations where a structural part may start or end. These segment borders can be defined in several ways. The simplest alternative would be not to define them at all, but to allow a segment to start and stop anywhere. As noted in [8], the main problem with this is that the computational complexity becomes very high at the latter stages. If there are N different locations where the part borders may lie, there exists $O(N^4)$ pairs of non-overlapping segments between them.

A data-adaptive method for determining the lengths and locations of segments utilising their recurrence information was described in [5]. Here, we chose to use a simple method relying on musical texture changes [10]. It has been noted that often different musical parts in a piece have clearly different timbral properties, and the method utilises this observation. A self-similarity matrix $\mathbf{S}$ is calculated from the measure-wise MFCC statistics. The similarity between feature vectors $\mathbf{m}_i$ and $\mathbf{m}_j$ from measures i and j, respectively, is defined to be

$$\sigma(\mathbf{m}_i, \mathbf{m}_j) = 0.5 + 0.5\frac{< \mathbf{m}_i, \mathbf{m}_j >}{\parallel \mathbf{m}_i \parallel \parallel \mathbf{m}_j \parallel}, \quad (4)$$

which stems from cosine distance measure. The elements in the matrix are $[\mathbf{S}]_{i,j} = \sigma(\mathbf{m}_i, \mathbf{m}_j)$.

A novelty vector is calculated from the similarity matrix by correlating a Gaussian taper checkerboard kernel matrix along the main diagonal of the similarity matrix (see [10] for details). Effectively it operates as a 2D border detection method, resulting into peaks in the novelty vector wherever the texture changes considerably. The border locations are assigned to the locations of the largest peaks. The exact number is not critical, as the locations do not define the final structural segmentation of the piece. However, it should be large enough to make sure that the real segment borders are contained within them, and still small enough not to make the subsequent matching computationally too expensive.

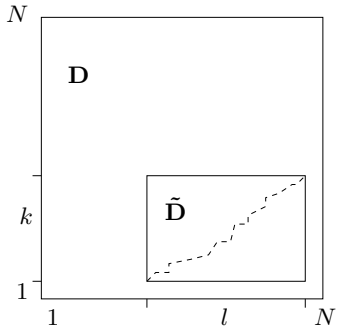

Figure 5: Pairwise distance calculation.

2.4 Segment Matching

To enable evaluating the cost function (3), the within-group dissimilarity d_i must be defined. Initially, all possible non-overlapping segment pairs are generated, so that the segments in a pair may start and end at any of the border locations, as long as they do not overlap each other. After that, the distance between the segments in a pair is calculated by utilising a two-level dynamic time warping, motivated by the multi-scale processing in [15] and the measure-level similarity matrix in [24]. The distance calculations are done using the reduced-dimensionality chroma vectors.

First, a measure-level distance matrix $\mathbf{D}$ is calculated with a method similar to the one described in [24]. The element $[\mathbf{D}]_{i,j}$ denotes the distance between the measures i and j in the piece. For each measure, the chroma frames contained within its range are used. The frames of two measures are matched with DTW. It allows small adjustments to the temporal alignment of the frames, enabling the matching even if the metrical estimation has had problems and the number of frames differs between the matched measures. At the feature vector level, the used distance measure is cosine, i.e., distance between vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is $1 - \sigma(\mathbf{v}_i, \mathbf{v}_j)$.

The pairwise distance $\delta(k, l)$ between the segments s_k and s_l is calculated by finding the cheapest path through the sub-matrix $\tilde{\mathbf{D}}$ of $\mathbf{D}$ defined by the borders of the matched segments, as illustrated in Figure 5. The path can be found using dynamic programming. The pairwise distance $\delta(k, l)$ is defined to be the total cost of the optimal path, normalised with the length of the diagonal of the sub-matrix.

Since all occurrences of a part should be approximately of the same length, we prune all pairs with length ratio outside the range $r = [\frac{5}{6}, \frac{6}{5}]$, resulting to pruning of a large quantity of the pairs. The within-group dissimilarity d_i of the group g_i is then defined to be the average of all the pairwise distances in the group

$$d_i = \frac{\sum_{k,s_k \in g_i} \sum_{l,s_l \in g_i, l \neq k} \delta(k, l)}{\sum_{k,s_k \in g_i} \sum_{l,s_l \in g_i, l \neq k} 1}. \quad (5)$$

The group candidate set G can now be constructed iteratively with the following steps:

1. Initialise the group set G with all the possible segment pairs after pruning pairs with length ratio outside r.

2. Create new groups by extending the groups that were added in the previous iteration with all non-overlapping segments. I.e., for each group, as many new groups are generated as there are non-overlapping segments. If the ratio of the largest and smallest pairwise distance between the segments in a group is outside r, delete the group.

3. Calculate new within-group dissimilarities d_i for the added groups using (5).

4. Repeat from Step 2, until no more items can be added or some maximum group size is reached.

Now the optimal explanation can be found by creating all such subsets of G that are valid explanations E, and evaluating the cost function for them. However, this is computationally very expensive.

2.5 Structure Search Algorithm

An efficient search algorithm can be implemented by constructing and evaluating the explanations incrementally after some data reorganisation. The group candidates in G are sorted into ascending order based on the within-group dissimilarities d_i. Initialise global variables
$C_{best} \leftarrow \alpha$, and
$E_{best} \leftarrow \{\emptyset\}$, initialise iteration base with
$E \leftarrow \{\emptyset\}$,
$l \leftarrow 1$, and
$C_{old} \leftarrow \alpha$, and call the function `Cutting-Search` with the initialised parameters.

```
1: function CUTTING-SEARCH(E, l, C_old)
2:   for i ← l, M do
3:     if g_i ∩ g_j = {∅}, ∀g_j ∈ E then
4:       E_new ← E ∪ g_i
5:       C_new ← C_old + c_i(d_i − α) + β log (|E|+2)/(|E|+1)
6:       if C_new < C_best then
7:         C_best ← C_new
8:         E_best ← E_new
9:       end if
10:      C_limit ← C_new + (1 − Σ_{j,g_j∈E_new} c_j)(d_i − α) +
     β log (|E|+3)/(|E|+2)
11:      if C_limit < C_best then
12:        CUTTINGSEARCH(E_new, i + 1, C_new)
13:      end if
14:    end if
15:  end for
16:  return C_best, E_best
17: end function
```

The loop upper limit M is the number of group candidates in the set G. Here g_i refers to the member i of the list of group candidates G constructed above, and $|E|$ denotes the number of groups in the explanation E.

The algorithm would evaluate all possible combinations of the candidate groups by extending the existing explanation with the recursive call on line 12, but the size of the search space is reduced. The reduction is done by estimating the lower bound, C_{limit}, of the cost that can be obtained by extending this explanation (line 10). If the obtainable cost is worse than the minimum cost found so far, there is no use of extending the search in this direction. Hence, the algorithm finds the globally optimal combination of groups g_i in respect to the cost function (3).[1] This explanation is returned in E_{best}.

Table 1: Statistics about the musical structures of the pieces in the used database.

data set	parts	occurrences	length
MUSIC	4.2	3.4	19.2s
RWC-Pop	4.3	2.4	21.4s
The Beatles	3.7	2.8	13.2s
whole db	**4.1**	**3.1**	**18.9s**

3. EVALUATION

The performance of the system was evaluated in simulations analysing the structure of a set popular music pieces and the structural explanations given by the algorithm were compared with a manually annotated reference structures.

3.1 Material

To enable evaluating the analysis result of the algorithm, a database of structural annotations for 50 musical pieces was collected. The used pieces were from three different sources: 34 popular music pieces from the MUSIC database, described in [14], 10 pieces from the RWC Popular Music database [13], and 6 pieces performed by The Beatles.[2]

The structure annotations were done manually, without any automatic tools. From each piece, the clear structural elements were annotated, i.e., some temporal locations of the piece may be left unannotated, as they did not belong into any clear structural part. An annotation consists of a list of the occurrences of structural parts, each with a start and a stop time and a name. The main focus was on parts that occurred more than once during the piece. Still, if there was some clear parts that could be named (e.g., intro, bridge, solo), they were annotated too. Also, if some part could be interpreted to be an occurrence of some other label, these alternative labels were annotated. E.g., if the solo part was very similar to the main verse part, but still different enough, it was annotated to be solo, but with the alternative label of verse. This was done because the labelling is not always unambiguous even for a human listener, let alone to a computer. By defining the alternative labels, more structural interpretations were allowed. The pieces in the database contain in total 642 annotated part occurrences, and 31 of these occurrences have an alternative label definition. As the system is capable of recognising only the parts that occur at least twice, the parts occurring only once were removed from the reference before evaluation.

Some statistics about the structural annotations of the individual parts and the whole database are presented in Table 1. The column *parts* tells the average number of unique labels in each piece on average, the column *occurrences* how many time each label occurs in a piece on average, and the column *length* the average length of a part occurrence in a piece. The statistics have been calculated from all annotated parts, without discarding the ones occurring only once.

[1]For a 5 min piece, with reasonable values for α and β, the optimisation algorithm finds the solution in a matter of seconds, when run on a 1.7 GHz Pentium 4 PC.

[2]The used pieces are listed at `http://www.cs.tut.fi/sgn/arg/paulus/structure/dataset.html`.

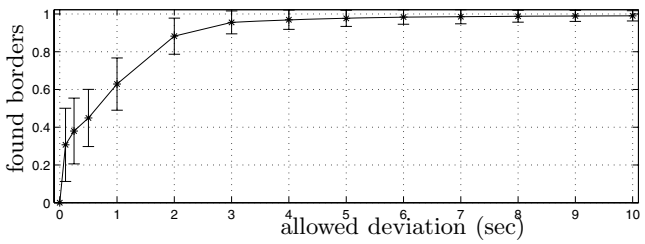

Figure 6: Ratio of found segment borders with different accuracy requirements. The error bars specify the standard deviation over all test pieces.

The actual evaluations were done with a 3-fold cross-validation scheme. In each fold, the values for the parameters α and β were trained with a grid search for the 2/3 of the material, and the determined values were used in evaluations for the remaining 1/3 of the material. The division of the available pieces to the three subsets was done randomly from each data source, i.e., it was taken care that the pieces from each source were divided approximately evenly to each subset. The presented results are calculated over all the folds.

3.2 Segmentation Accuracy

Since the performance of the border candidate generation affects the performance of the rest of the proposed system, it was tested separately. The border candidates were generated by using a convolution kernel of the length 12 s in novelty vector calculation, and choosing at most 50 largest peaks.[3] Each annotated segment border was matched with the closest found border if no other annotated border had been coupled with it yet and their time difference was smaller than the allowed deviation. The matching was performed with several different allowed deviation values and the ratio of found segment borders was calculated. The result is illustrated in Figure 6, where the line is the mean ratio of found segment borders and the error bars illustrate the standard deviation over the whole test sets.

3.3 Comparison of Structures

Given the explanation E_{best} for the structure of the piece by the algorithm and the reference structure annotation E_{ref}, the task is to compare the similarity of the two. As noted in [5] and illustrated by Figure 1, the structure of musical pieces is often hierarchical, at least in Western popular music. The annotated reference structure and the explanation given by the algorithm may be on a different level of the hierarchy, making a direct comparison impossible. Also, the labelling of the structural parts is not determined by the algorithm: it only assigns unique labels to the parts. Still, it would be desired that if the reference explanation and the one given by the algorithm describe the same structure in some level of the hierarchy, the similarity is recognised. An extreme case is illustrated in the top panel of Figure 7, the explanations describe different fine structure, but still they both have the same structure on the highest level of hierarchy, and hence they should be accepted as a match.

[3]The used maximum number of peaks was determined empirically to be a reasonable tradeoff between the segmentation accuracy and them computational complexity of the further steps in the algorithm.

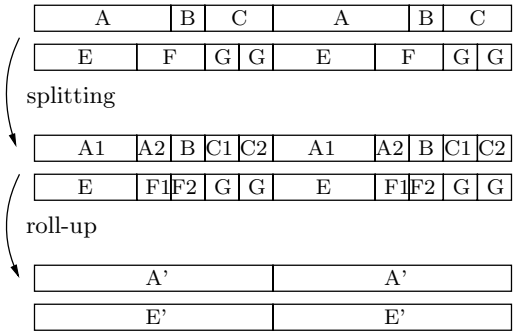

Figure 7: An example of how the fine structure of two descriptions differ, but the proposed evaluation method will judge the two similar.

To our knowledge, the level ambiguity has been addressed only by Chai [5]. She proposed to use a roll-up process in the evaluations to find a common level for comparison. In roll-up, if some part is split into finer structural description in either reference annotation or analysis result, the repetition of the finer structure are replaced with occurrences of the longer part. However, the proposed roll-up process does not yield a desired result in some cases, e.g., the one illustrated in top panel of Figure 7. Here, we extend the method presented therein.

The first problem is the temporal alignment of the reference and proposed structures: because the segmentation method described in Section 2.3 may not always find the same borders for parts as those assigned in reference annotations, or the borders lie at different levels of the hierarchy, it is required to create a common time base for both structural representations. This can be done by concatenating all the start and end times of all segments in both the reference and the analysis structures, sorting the times into ascending order, and removing possible duplicates. The main advantage of using the common time base is that the descriptions may be compared as sequences of characters.

The structural descriptions must now be represented using the generated time base. This is done by splitting the original segments, separately in both reference and analysis result, from the common temporal borders with the following steps:

1. Take the occurrences of a structural part that has not yet been handled.
2. Divide the occurrences according to the common time base. The resulting shorter sections are referred to as subparts.
3. Go through the subparts and assign them with new labels: (a) If a subpart occurs at the same location within more than one of the unsplit part occurrences with the same duration, all these occurrences with a same label. (b) If some subpart occurs without any repetition, assign it with its own label.

This operation is illustrated in the top part of Figure 7. The two structural descriptions with different segment borders in the top of the figure are split with common borders to result the descriptions in the middle of the figure.

The next step is to rise on the structural hierarchy with a procedure called the roll-up. It is illustrated in the lower

Table 2: Evaluation results for each data subset and for the whole database. The results are calculated over the cross-validation folds.

segm.	data	R	P	F
perfect	MUSIC	72.3%	74.9%	72.9%
	RWC-Pop	80.9%	88.1%	84.2%
	The Beatles	95.8%	99.8%	97.7%
	whole db	**76.8%**	**80.6%**	**78.1%**
novelty	MUSIC	62.2%	68.5%	63.6%
	RWC-Pop	73.3%	85.3%	78.2%
	The Beatles	59.0%	81.5%	67.6%
	whole db	**64.0%**	**73.4%**	**67.0%**

part of Figure 7. In the roll-up, if similar sequence of segments occurs in both descriptions, ignoring the labels, the sequence is replaced in both descriptions with a part on a higher hierarchical level. In the illustrated example, the sequence "A1, A2, B, C1, C2" in the split reference occurs at the same location as the sequence "E, F1, F2, G, G" in the split analysis result, and hence the sequences may be rolled up in hierarchy to result to the lowest two descriptions.

After the roll-up, the structural descriptions should be on the same level of hierarchy and the actual evaluation can be done. The mapping between the label sets of the reference and analysis result is done so that each label from other set can be associated with one or no label in the other set. Each mapping is evaluated according to two measures: recall rate R and precision rate P. Recall rate describes how large temporal portion of the parts annotated in the reference are also explained by the analysis result. Precision rate describes how much of the analysis result is correct. From these two metrics, the harmonic F-measure is calculated to give one value describing the performance by $F = 2RP/(R + P)$.

3.4 Results

The overall evaluation results are presented in Table 2. The column *segm.* describes the method used in the segment border candidate determination, the value *novelty* denotes that the segmentation was generated with the audio novelty method described in Section 2.3, and the value *perfect* that the segment borders candidates were taken from the reference annotation. This was done to allow evaluating the effect of the border candidate generation method. The evaluation measures are presented for each data subset and for the whole database. The results for each song in the database are illustrated in Figure 8.

It can be observed that the performance on the MUSIC data set is considerably lower than with other subsets. When inspecting the reason for this, it was noted that the analysis had failed completely on some pieces. The reason for which similar failures were not encountered in the other two data sets is probably due to the musical diversity of the pieces in MUSIC. This leads to the conclusion that it may not be possible to find such values for the parameters α and β that they would perform equally well with all pieces. It can also be seen that the used segment border candidate generation method needs attention in future work.

Some typical results are illustrated in Figures 9-11. The figures contain the annotated structure in top panel and the analysis result in the lower panel. The analysis results were obtained using the system generated border location candidates and fixed parameter values found in the cross-validation runs.

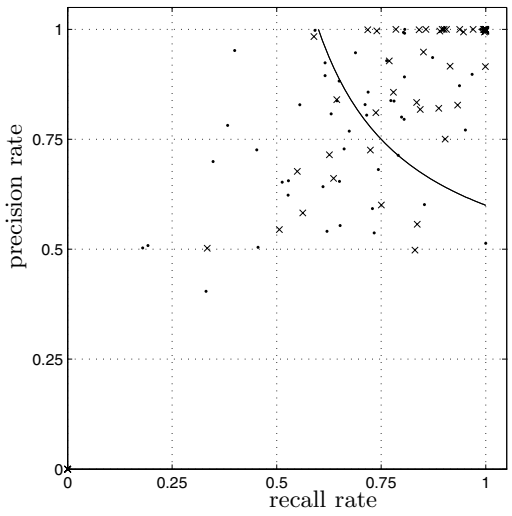

Figure 8: System performance on different pieces of the test database. The dots (·) are the performance using the novelty based border candidate generation, the crosses (×) are the performance with simulated perfect segmentation, and the solid line illustrates F-measure value 0.75 for a reference point.

4. CONCLUSION

The problem of structural analysis of musical pieces can be formulated in terms of a parametric cost function which allows varying the costs assigned to different aspects of the resulting description. The computational load incurred by the search for the description that minimises the cost is negligible compared to that of the signal-processing front-end. This allows to create an analysis application in which the user may control the cost function parameters interactively. An example of a such application is illustrated in Figure 12.

5. ACKNOWLEDGEMENTS

The musical structures for the evaluation material database was annotated by Teemu Karjalainen.

6. REFERENCES

[1] S. Abdallah, K. Nolad, M. Sandler, M. Casey, and C. Rhodes. Theory and evaluation of a Bayesian music structure extractor. In *Proc. of 6th International Conference on Music Information Retrieval*, London, UK, Sept. 2005.

[2] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden Markov models. In *Proc. of 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001.

[3] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. of 2003 IEEE Workshop on Applications of Signal Processing to Audio and*

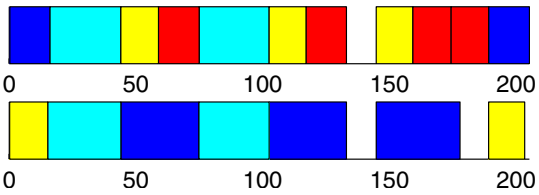

Figure 9: Structure annotation (top panel) and analysis result (lower panel) of Abba's song "S.O.S.". The result is relatively good with only one totally missed part.

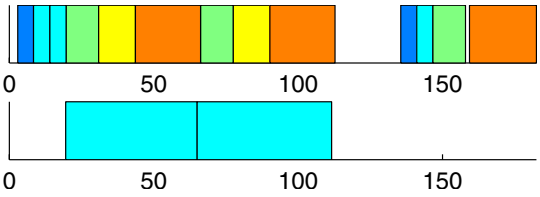

Figure 10: Annotation and analysis result of the piece RM-P035 ("Midarana kami no moushigo" by Hiromi Yoshii) from RWC Popular Music Database. The result is not that good, but still it has found a long part occurring twice.

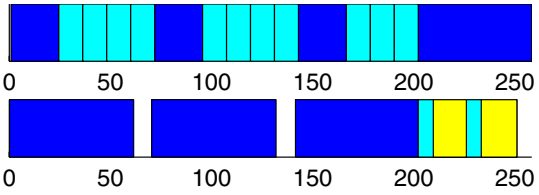

Figure 11: Annotation and analysis result of the piece "You got me" by The Roots. The analysis has grouped the continuation of chorus and three occurrences of the verse as one part and revealed a finer structure within the outro/chorus.

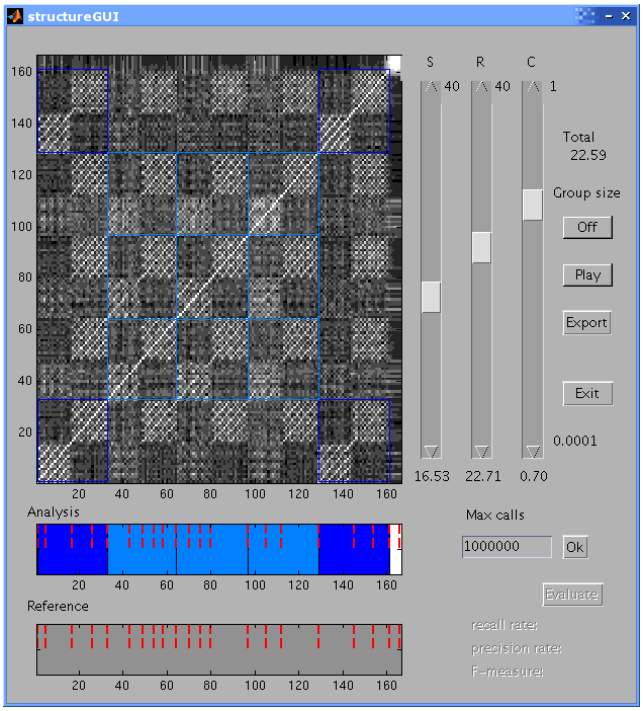

Figure 12: An example of an interactive graphical user interface allowing the user to control the cost function parameter values while getting an immediate feedback. The UI displays, among others, the measure-wise similarity matrix, border candidate locations, and the analysis result.

Acoustics, pages 15–18, New Platz, New York, USA, Oct. 2003.

[4] M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, Feb. 2005.

[5] W. Chai. *Automated Analysis of Musical Structure*. PhD thesis, Massachusetts Institute of Technology, Sept. 2005.

[6] W. Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *IEEE Signal Processing Magazine*, 23(2):124–132, Mar. 2006.

[7] R. B. Dannenberg. Toward automated holistic beat tracking, music analysis, and understanding. In *Proc. of 6th International Conference on Music Information Retrieval*, pages 366–373, London, UK, Sept. 2005.

[8] R. B. Dannenberg and N. Hu. Pattern discovery techniques for music audio. In *Proc. of 3rd International Conference on Music Information Retrieval*, pages 63–70, Paris, France, Oct. 2002.

[9] J. Foote. Visualizing music and audio using self-similarity. In *Proc. of ACM Multimedia*, pages 77–80, Orlando, Florida, USA, 1999.

[10] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 452–455, New York, USA, Aug. 2000.

[11] J. T. Foote and M. L. Cooper. Media segmentation using self-similarity decomposition. In *Proc. of The SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, San Jose, California, USA, Jan. 2003.

[12] M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 437–440, Hong Kong, 2003.

[13] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proc. of 3rd International Conference on Music Information Retrieval*, pages 287–288, Paris, France, Oct. 2002.

[14] T. Heittola. Automatic classification of music signals. Master's thesis, Tampere University of Technology, Dec. 2003.

[15] T. Jehan. *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, Sept. 2005.

[16] T. Jehan. Hierarchical multi-class self similarities. In *Proc. of 2005 IEEE Workshop on Applications of*

Signal Processing to Audio and Acoustics, pages 311–314, New Platz, New York, USA, Oct. 2005.

[17] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Speech and Audio Processing*, 14(1):342–355, Jan. 2006.

[18] M. Levy, M. Sandler, and M. Casey. Extraction of high-level musical structure from audio data and its application to thumbnail generation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 13–16, Toulouse, France, May 2006.

[19] B. Logan and S. Chu. Music summarization using key phrases. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, Istanbul, Turkey, June 2000.

[20] L. Lu, M. Wang, and H.-J. Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proc. of Workshop on Multimedia Information Retrieval*, pages 275–282, New York, USA, Oct. 2004.

[21] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proc. of ACM Multimedia*, pages 112–119, New York, New York, USA, Oct. 2004.

[22] G. Peeters. Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach. In *Lecture Notes in Computer Science*, volume 2771, pages 143–166. Springer-Verlag, 2004.

[23] C. Rhodes, M. Casey, S. Abdallah, and M. Sandler. A Markov-chain Monte-Carlo approach to musical audio segmentation. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 797–800, Toulouse, France, May 2006.

[24] Y. Shiu, H. Jeong, and C.-C. J. Kuo. Musical structure analysis using similarity matrix and dynamic programming. In *Proc. of SPIE Vol. 6015 - Multimedia Systems and Applications VIII*, 2005.