Automatic Audio Segmentation: Segment Boundary and Structure Detection in Popular Music

Ewald Peiszer, Thomas Lidy, Andreas Rauber

Vienna University of Technology, Austria,
Department of Software Technology and Interactive Systems

Abstract. Automatic Audio Segmentation aims at extracting information on a song's structure, i.e., segment boundaries, musical form and semantic labels like *verse, chorus, bridge* etc. This information can be used to create representative song excerpts or summaries, to facilitate browsing in large music collections or to improve results of subsequent music processing applications like, e.g., *query by humming*.
This paper features algorithms that extract both segment boundaries and recurrent structures of popular songs. Special attention has been paid to the evaluation setup: We employ the largest corpus that has been used so far in this field, discuss why comparing two song segmentations is inherently delicate and propose a flexible XML format that can describe hierarchical segmentations to promote a common basis that makes future results more comparable.

1 Intro

The topic of this paper, *Automatic Audio Segmentation* (AAS), is a subfield of *Music Information Retrieval (MIR)* that aims at extracting information on the musical structure of songs in terms of segment boundaries, recurrent form (e.g., `ABCBDBA`, where each distinct letter stands for one segment type) and appropriate segment labels like *intro, verse, chorus, refrain, bridge*, etc. Automatically extracted structural information about songs can be useful in various ways, including facilitating browsing in large digital music collections, creating new features for audio playback devices (skipping to the boundaries of song segments) or as a basis for subsequent MIR tasks.

In this paper we present a two-phase algorithm for boundary and structure detection. We focused on the complete annotation of all song parts both with sequential-unaware approaches and an approach that takes temporal information into account. *(Sect. 3)*

Much attention is paid to proper evaluation. We calculate confidence intervals and use a large groundtruth corpus which contains 94 songs of various genres. Final evaluation runs are conducted on a 109 song corpus which is the largest corpus used so far in this research field. Fellow researchers are invited to use the same corpus in their experiments so that the results can easily be compared

(e.g., in the context of a MIREX task). *(Sect. 4)* The paper closes with a discussion of the results and suggestions for future work. *(Sects. 5, 6)* Groundtruth annotations, HTML reports and source code can be accessed on the web[1].

2 Related work

Foote [Foo00] was the first to use a two-dimensional self-similarity matrix (autocorrelation matrix) where a song's frames are matched against themselves. After correlating this matrix with a Gaussian tapered kernel a "novelty score" emerges whose peaks can give hints about segment boundaries.

Since then, many studies used this idea as a basis, enhancing the algorithm with other techniques. E. g., given the self-similarity matrix Chai [Cha05] uses Dynamic Time Warping (DTW) to find both segment transitions and segment repetitions. DTW computes a cost matrix from where the optimal alignment of two sequences can be derived. It is assumed that the alignment cost of a pair of similar song sections is significantly lower than average cost values.

Another frequent approach uses Hidden Markov Models (HMM) [ANS+05, ASRC06, AS01, LC00, Mad06, RCAS06, LSC06, LS06]. Feature vectors are parameterized using Gaussian Mixture Models (GMM). These parameters are used as the HMM's output values. After Viterbi decoding the most likely state sequence there are two ways to continue. Some authors use the HMM states directly as segment types, often resulting in a very fragmented song structure. Another possibility is to use a sliding window to create short-term HMM state histograms that, in turn, are clustered using a standard clustering technique to derive the final segment type assignment.

Some studies [MXKS04, Mad06, Got03] place tight constraints on a song's structure so that only a fraction of the solution space needs to be considered.

Music structure analysis combines various subtasks (segmentation, summary extraction, audio thumbnailing, semantic label assignment, etc.). Similarly, various feature sets are used in related literature. Table 1 gives a clear overview of feature sets and subtasks in individual papers.

Also, music corpora used for evaluation are different throughout related literature. For convenience, Table 2 shows a survey on them that includes this paper for comparison.

3 Audio Segmentation – System Description

3.1 Boundary Detection

Phase 1 of the algorithm tries to detect the segment boundaries of a song, i.e., the time points where segments begin and end. The output of this phase is used as the input for the next phase, structure detection.

We chose a frequently used approach [Foo00, FC03, Ong05] that uses local information change through time as the basis.

[1] http://www.ifs.tuwien.ac.at/mir/audiosegmentation/

Feature sets	[ANS+05]	[AS01]	[BW01]	[BW05]	[CS06]	[Cha05]	[CF03, FC03]	[Foo00]	[Got03]	[LSC06]	[LC00]	[LWZ04]	[MXKS04, Mad06]	[Ong05]	[PK06]	[PBR02]	(this paper)	[RCAS06]	[TC99]
MFCC		✓									✓		✓	✓	✓		✓		
CQT												✓			✓		✓	✓	
Chromagram	✓		✓	✓	✓	✓			✓	✓			✓		✓		✓		
Other		✓[a]			✓[b]			✓[c]					✓[d]			✓[e]	✓[e, f]		✓[g]
Subtasks																			
Segmentation	✓	✓				✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Recurrent structure, musical form	✓	✓			✓[h]	✓	✓			✓		✓[g]	✓		✓[g]	✓	✓	✓	
Audio summary				✓		✓	✓					✓				✓			
Key phrase / chorus detection			✓						✓		✓								
Semantic labels													✓	✓					
Vocal / instrumental section detection												✓	✓						

Table 1. Overview of feature sets used and subtasks of individual papers.

[a] Linear prediction of spectral envelope; Discrete Cepstrum
[b] Log-Frequency Cepstral Coefficients
[c] Spectrogram
[d] Octave Scale Cepstral Coefficients
[e] "dynamic features"
[f] Rhythm Patterns, Statistical Spectrum Descriptors
[g] Spectral Centroid, Roll-off and Flux; Zero Crossings, RMS
[h] concentrate on repeated segments, parts of songs may be left unexplained

Paper	Corpus; annotation	Evaluation	Notes
[ANS+05, LSC06, LS06, RCAS06]	14 songs; start, end time and label for each segment	performance measure from image segmentation (adapted); information-theoretic measure	annotations available from web[a], tracks 17–30
[AS01]	20 songs of various genres (folk to rock, pop, blues and orchestral music); no annotation	empirically: "The better the segmentation, the more coherent the different textures sound."	
[BW01, BW05]	93 songs; annotated chorus section; genres include dance, country-western, Christian hymns)	no evaluation of segmentation	
[CS06]	7 music tracks; start times of melodic repetitions	mean query rank	
[Cha05]	21 classical piano solo pieces, 26 Beatles songs; start, end time and label for each segment	roll-up procedure, edit distance	list of songs in [Cha05, Appendix A]
[CF03]	seven songs; number of verses and choruses	number of verses and choruses	list of songs in [CF03, Table 1]
[Got03]	100 songs from music database RWC [GHNO02]; start and end times of choruses	length of chorus sections	
[LC00]	50 Beatles songs	user tests (ten subjects)	
[LWZ04]	100 songs; annotated: repetitions, music structure	edit distance	
[MXKS04, Mad06]	50 songs; vocal/instrumental boundaries, chord transitions, key, song structure	number and length of segments	with the help of commercial music sheets; example annotation: [Mad06, Fig. 7]
[Ong05]	54 Beatles songs; start, end time and label for each segment	intersection of boundaries, allowing 3 s deviation	annotation according to website[b]
[PK06]	50 songs (subset of MUSIC [Hei03] and RWC databases [GHNO02], Beatles songs); start, end time and label for each segment	adapted roll-up procedure [Cha05]	songs listed on web[c]
(this paper)	94+15 songs (subset of [PK06]'s and [LS07]'s corpus, a few additional songs); start, end time and label for each segment, 2-level-hierarchy, alternative labels	weighted, normalized edit distance, independent evaluation of boundaries and form	see web[d] for a list of songs
[TC99]	10 sound files (classic, jazz, pop, radio)	number of boundaries within 0.5 s of average human boundaries	groundtruth based on multiple annotators

Table 2. Overview of corpora and evaluation methods used in the literature. Songs referred to as "Beatles songs" of different papers need not be the same.

[a] http://www.elec.qmul.ac.uk/digitalmusic/downloads/index.html#segment
[b] http://www.icce.rug.nl/ soundscapes/DATABASES/AWP/awp-beatles_projects.shtml
[c] http://www.cs.tut.fi/sgn/arg/paulus/structure/dataset.html
[d] http://www.ifs.tuwien.ac.at/mir/audiosegmentation/

First, as a preprocessing step we downsample each music file to a 22,050 Hz mono audio stream that is split into variable sized, non overlapping frames. The frames are beat-synchronized. We regard segment boundaries as a subset of beat onsets since a new verse or chorus will not start between two beats. The durations typically range from 400 to 550 ms. Sometimes the beat tracker does not detect all beat onsets of the entire song (especially those in a soft fade-out section near the end of a song are frequently missed). In this case we extrapolated the missing onsets using the song's mean beat duration.

From each of the frames a feature vector $\mathbf{v}$ is extracted. We tried various types of features: simple spectrogram, a timbre-related feature set (Mel Frequency Cepstrum Coefficients), two rhythm-based feature set (Rhythm Patterns [RPM02], Statistical Spectrum Descriptors [LR05]) and harmony related features (Constant Q Transform [BP92]).

Then, the self-similarity matrix $\mathbf{S}$ between these feature vectors is calculated using a distance function $d_{\mathbf{S}}$. Subsequently, the novelty score N is derived and a smoothening low-pass filter H_N is applied. This produces a set of segment boundaries $\mathcal{B}_1$ that contains each peak of N unless there is a higher peak within a 6 s interval, *or* the moving average of the amplitude over a period of 5 frames is lower than a threshold T_a (usually $T_a = 0.2$).

A number of experiments and parameter tweaking has been carried out, however, without significant performance improvement (e.g., removing peaks below a threshold, boundary shifting post-processing [LSC06], Harmonic Change Detection Function HCDF [HSG06]). See [Pei07, Sect. 3.2] for a discussion.

3.2 Structure Detection

Phase 2 of the algorithm tries to detect the structure of the song, also referred to as musical form, i.e., a label is assigned to each segment where segments of the same type (verse, chorus, intro, etc.) get the same label. The labels themselves are single characters like `A`, `B`, `C`, etc., and thus not semantically meaningful. The structure of a song can be conveniently deducted from these labels, though, and it can also serve for subsequent segment type recognition.

The algorithm takes $\mathcal{B}_1$ from phase 1 as the input. The set of segments $\mathcal{S}_1$ is created by taking the intervals between the time points, including the start and end of the song.

We assume that segments of the same type are represented by similar features. Thus, we employ unsupervised clustering techniques. In particular, the following clustering experiments have been carried out:

Means-of-frames Each segment is represented by a feature vector that contains the mean values over all frame feature vectors of the segment. These are clustered using a standard k-means approach with input parameter k (number of cluster centroids). The clustering is repeated ten times and the solution with the lowest within-cluster sums of point-to-centroid distances is chosen.

Agglomerative clustering Same segment representation, hierarchical clustering approach. Complete linkage function has been used.

Voting K-means with all frame feature vectors. Segment type of each segment is assumed to be the cluster number that is assigned to most of its frames.

Dynamic Time Warping (DTW) We compute a segment-indexed similarity matrix $\mathbf{S}_{segs}$ using DTW alignment costs of each pair of segments. Then, points in the Euclidean space are created according to the distances in $\mathbf{S}_{segs}$. These points are k-means clustered.

Again, experiments have been carried out to improve performance. We used cluster validity indices (Dunn, Davies-Bouldin) [HBV01] to determine the correct number of clusters. Also, we investigated the effect of minimal user input (user chooses number of desired segment types manually). These optimizations, however, did not show a significant improvement. Please refer to [Pei07, Sect. 4.2] for a detailed description and figures.

4 Evaluation

This section describes the corpus we used, defines performance measures, introduces a new XML format for groundtruth files and presents the algorithm results, also in comparison to other studies.

4.1 Groundtruth

To be able to compare results of various research studies the algorithms should run on the same corpus. Therefore we tried to collect as much annotation data as possible that has already been used in prior studies. Eventually, we could base our work upon data used in [PK06] (50 songs, "Paulus/Klapuri corpus") and in [LS07][2] (60 songs, "qmul[3] corpus"), respectively. A subset of the latter corpus has been used in [RCAS06, CS06, LSC06, AS01, ANS+05], too (14 songs, "qmul14 corpus").

As the corpora overlap and because we could not obtain all songs the corpus which we used for the experiments finally contained 94 distinct songs. This is one of the largest corpora used so far in this field. At the end we enlarged the corpus by fifteen additional songs which eventually led to the largest corpus against which an AAS algorithm has ever been evaluated. We included the corpus information (the corpora a song belongs to) in the groundtruth files to calculate also corpus-specific performance measures. The resource website includes the complete list of songs we used.

[2] http://www.elec.qmul.ac.uk/digitalmusic/downloads/index.html#segment
[3] Centre for Digital Music, *Q*ueen *M*ary, *U*niversity of *L*ondon

4.2 Performance Measures

Boundaries Following the approach used, e.g., in [Cha05] we calculate precision P, recall R and F-measure F. Let the sets $\mathcal{B}_{algo}$ and $\mathcal{B}_{gt}$ denote begin and end times of automatic generated segments and groundtruth segments respectively, then P and R are calculated as follows:

$$P = \frac{|\mathcal{B}_{algo} \cap_w \mathcal{B}_{gt}|}{|\mathcal{B}_{algo}|} \tag{1}$$

$$R = \frac{|\mathcal{B}_{algo} \cap_w \mathcal{B}_{gt}|}{|\mathcal{B}_{gt}|} \tag{2}$$

F is the harmonic mean of P and R. A parameter w determines how far two boundaries can be apart but still count as one boundary (e.g., 3 s).

Structure Following Chai's notion we use the *formal distance* metric f which basically is the *edit distance ed* between strings representing the two structures, independent of the actual naming of the distinct segments as long as segments with the same label get the same character. That is,

$$f(\texttt{ABABCCCABB}, \texttt{ABCBBBBACC}) = 3 \tag{3}$$

because

$$ed(\texttt{ABABCCCABB}, \texttt{A}\boxed{\texttt{CBCCCC}}\texttt{A}\boxed{\texttt{BB}}) = 3 \tag{4}$$

(in the second argument `B` and `C` have been swapped). In addition, the characters are weighted with the length of the segment they represent. To relate f to the song duration dur_s we use the formal distance ratio

$$r_f = 1 - f/dur_s \tag{5}$$

4.3 Audio Segmentation File Format

We introduce a new XML based file format, **SegmXML**, describing audio segmentations. Both groundtruth annotations and automatically generated ones are encoded in this format. These files contain information about song metadata (title, artist, etc.) and segments (begin and end times, labels, alternative labels, hierarchy of segments).

Flexibility It is hardly possible to decide upon *one* song segmentation everyone would agree with. This means that even if two experts segment the same song, quite a different structure will probably emerge. As a matter of fact this was true for the songs that are contained in both the *Paulus/Klapuri corpus* and the *Queen Mary corpus*.

From this perspective we decided to add flexibility to our file format. This includes **hierarchical segments** (A segment can be (non-recursively) divided

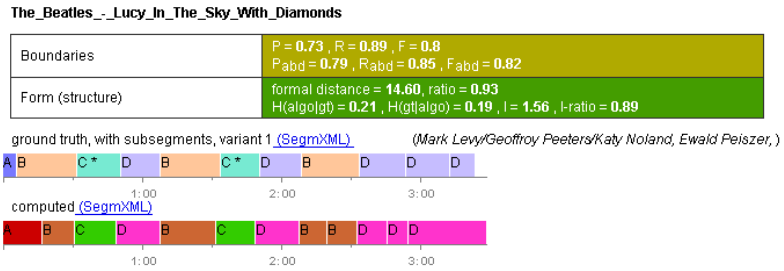

Fig. 1. Part of the body of an HTML report. The upper part contains performance numbers, the lower part is a representation of both the groundtruth (upper panel) and computed (lower panel) segmentation. Same color and letter within one panel correspond to same segment type.

into subsegments. This leads to a two-level hierarchy.) and **alternative labels** (Each segment has 1 to k labels. So, e.g., one segment can be seen as a *chorus* or as a chorus variant *chorusB*; the same is true for *bridge* versus *solo*.) We think that this format can be used for various genres of popular music, it probably does not fit for the more complex structured classical music, though.

4.4 Evaluation Procedure

This system includes an OS independent evaluation procedure, also available at the resource webpage.

Due to the flexibility of the **SegmXML** file format one groundtruth file actually contains several *groundtruth variants*. The evaluation procedure is executed for each pair of computed segmentation and groundtruth variant that can be extracted from the corresponding groundtruth file. The maximum of these performance numbers is chosen as the finally reported result. Note, however, that there is a semantic that controls the variants a SegmXML file can be "expanded to". E. g., in case that the alternative label for one specific segment is chosen, all other segments' alternative names (if available) are used, too. [Pei07, Sect. 2.3.2] states this semantic in pseudo code.

The performance numbers are output into one XML file, including mean values and confidence intervals, as well as both segmentation and evaluation algorithm parameters, remarks, debug output and warnings if appropriate. These XML files are finally rendered into HTML files that include graphical representations of all song segmentations (Fig. 1). All evaluation results are statistically well grounded by calculating and publishing **confidence intervals** from which statistical (in)significance can be derived. We use Student's T distribution and a significance level of $\alpha = 0.05$.

4.5 Results

The results given here have been produced using the following parameters: 40 MFCC coefficients, frame size 2^{14} frames, beat synchronized, Euclidean $d_{\mathbf{S}}$.

Boundary Detection Full corpus results are: $P = 0.58 \pm 0.036$, $R = 0.77 \pm 0.033$, $F = 0.66 \pm 0.034$ (11 frames sized H_N).

As mentioned earlier not all papers use the same corpus. For better comparability, we include results based on qmul14 corpus that has been used in [LWZ04] and [LSC06] in the results presented in Fig. 2.

If you look at *mean* P and F, disregarding the confidence interval, you can notice that results on qmul14 corpus are (much) higher (Fig. 2, last two columns). This shows one fact very impressively: The evaluation numbers depend to a larger degree on the corpus than on the algorithm or parameters. This again emphasizes the importance of carefully selecting songs for a common corpus if an audio segmentation benchmark evaluation is going to take place. You can also see how important it is to compute and publish proper confidence intervals: the mean values alone could be misleading.

Structure Detection Full corpus result based on automatically extracted (imperfect) boundaries is $r_f = 0.707 \pm 0.025$, using means-of-frames approach with $k = 5$ and an 8 frames sized H_N. Fig. 3 shows a comparison of structure detection results using various parameter sets and clustering approaches.

Unfortunately our performance numbers can not be compared to already published results: Both Chai [Cha05] and Lu *et al.* [LWZ04] publish mean edit distances in their evaluation section, they do, however, not normalize them against song duration / string length. Clearly, if structure strings are somewhat like `ABCBD` then edit distance will be lower than if a song's structure is represented by a string like `AABBBBCBBCBBCCCAAA`. Thus, we can not use their numbers for comparison.

We also calculated an alternative performance measure based on information theory which is proposed in [ANS+05] and [ASRC06]. The mean performance of the proposed algorithm is similar to that in [ANS+05] if based on the same corpus, qmul14. Full corpus results are insignificantly lower.

4.6 Additional Test

We applied one of the best performing parameter sets to a larger corpus than the one used so far. The fifteen additional songs comprise ten songs from the RWC pop collection [GHNO02] that also are part of the corpus used by Paulus *et al.* [PK06], and five songs that are personal favorites of the respective annotators. In a Machine Learning sense this set can be seen as a test set, i.e., a set whose contents have been omitted completely in the parameter selection phase. Table 3 contains the results for the original "full" corpus, for the set of additional songs and for the union set.

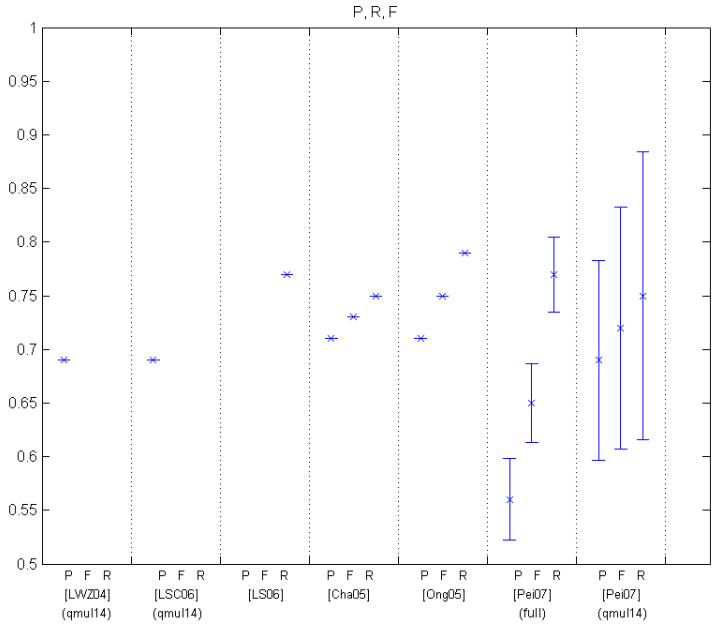

Fig. 2. Boundary detection evaluation numbers collected from various papers. Error bars indicate confidence interval (where available). Results based on qmul14 corpus are marked *(qmul14)*, all other columns can not be compared directly since they are not based on a common corpus. Precision of all three qmul14 results are on an equal level (see first, second and last column).

Corpus	**Boundary detection**	**Structure extraction**
"full" (94 songs)	$F = 0.66 \pm 0.034$	$r_f = 0.698 \pm 0.024$
"test set" (15 songs)	$F = 0.7 \pm 0.083$	$r_f = 0.668 \pm 0.088$
union (109 songs)	$F = 0.67 \pm 0.031$	$r_f = 0.694 \pm 0.024$

Table 3. Evaluation results of the independent test set, the full corpus and the union of these two corpus sets. Parameters: 40 MFCC coefficients, frame size 2^{14} frames, beat synchronized, Euclidean $d_{\mathbf{S}}$, 11 frames sized H_N, means-of-frames approach with $k = 5$. Note that the test set does not perform statistically significantly worse than the "full" corpus.

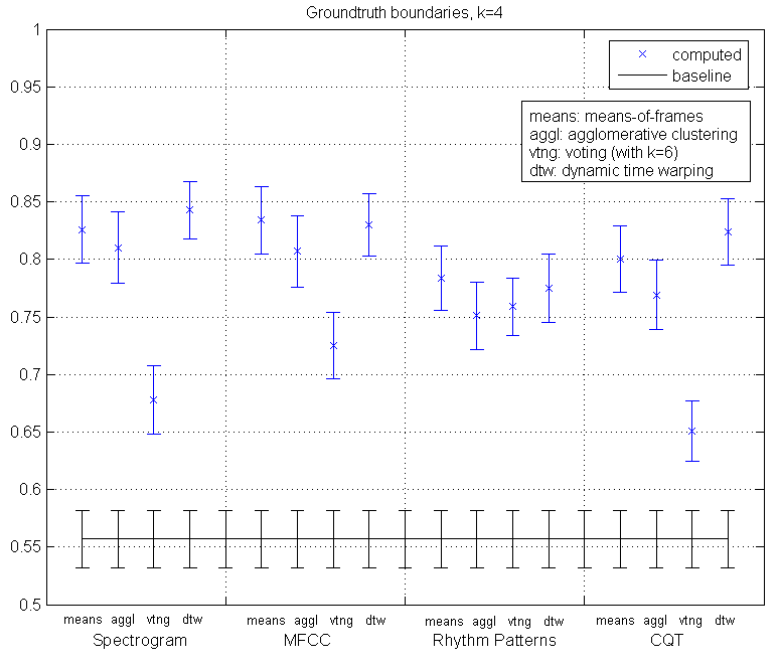

Fig. 3. Structure detection results when using various feature sets and clustering methods. The algorithm runs are based on groundtruth boundaries. k was set to 4, except for the voting approach where $k = 6$ since this approach usually produces segmentations with less than k segment types. It is observable that means-of-frames, agglomerative clustering and DTW approach perform similar, voting approach is significantly worse, though (except for Rhythm Patterns RP).

From the figures it is observable that there is no statistically significant difference between the results of the traditional corpus that has been used for parameter selection and those of the unseen test set. Thus, it can be concluded that no overfitting took place and that the algorithm in combination with these parameter values is general enough to be applied also to unseen songs.

5 Discussion

Automatic segment boundary detection results are at a reasonable high level if you consider that

- the algorithm operates exclusively on local information, i.e., it does not take the rest of the song into account.
- it does not make use of restricting domain knowledge which means that there is little restriction about the songs that can be processed.
- the corpus contains songs of various genres (pop, folk, rap, dance, etc.; no classical music, though).
- it is illusory to reach the "ideal" value of $F = r_f = 1$ because of inherent ambiguity. For *Michael Jackson: Black or White*, e.g., r_f values if evaluating two human annotations against a third one are 0.84 and 0.68. Thus, the mean value of 0.76 could be considered to be the upper limit for this song.

It can be noticed that computed segmentations tend to have too many boundaries which leads to a rather low precision value. Reasons for that may be:

- Frequently there is a novelty score peak at the change of instrument, which is not necessarily a segment boundary, leading to false positives.
- Boundaries in slow and soft songs are often shifted some time from the correct positions since the edges in the similarity matrix are not that distinct (e.g., in *Sinhead O Connor: Nothing Compares To You*).
- On the other hand, non-melodic audio parts like in rap songs exhibit fast changing feature vector distances leading to a jagged novelty score and too many boundaries.
- Also, songs with dense, distorted guitar sound seem to perform worse than melodic ones.

We were surprised that the large number of experiments and heuristics we tried did not lead to a significant mean performance improvement. The individual heuristics typically improved results of a subset of songs and impaired those of the rest, leading to almost the same mean performance. One reason could be that the groundtruth annotations we used are not consistent enough. Another possibility is that there is noise in terms of segmentation ambiguity which can not be eliminated.

We learned from the evaluation reports that it was not always the same songs which performed badly. There are, of course, songs that generally are easier to segment, e.g., *KC and the Sunshine Band: That's the Way I Like It* because of its distinctive segments' timbre differences and highly repetitive structure, but

the songs on the lower end change according to the feature set used and other parameters.

We frequently noted that the algorithm extracts a finer structure than the one used as groundtruth. As Goto assumed in [Got03] many chorus sections contain two subsegments. This decreases performance numbers but informally it is obvious that the extracted segmentations can still be useful. Subjectively, computed musical form results are more useful and accurate than the boundaries.

Again, we would like to note that it is not easy to compare the published performance numbers to results in other papers. We saw that the choice of the underlying corpus has a larger effect on the final evaluation numbers than the change of algorithm parameters and the use of heuristics.

Similarly, it is not obvious how evaluation should take place. Consideration must especially be given to the ambiguity of song segmentations. We decided to model it explicitly in the groundtruth annotation data.

6 Outro

6.1 Summary

In this work we presented an algorithm for Automatic Audio Segmentation (AAS). It consists of a segment boundary detection and a structure extraction phase.

Performance measures for both algorithm tasks have been defined. We proposed a novel XML file format that can describe two-level hierarchical song segmentations (*SegmXML*). Evaluation using a large and quite diverse groundtruth corpus showed that the algorithm is robust and comparable to already published results.

Groundtruth annotations, HTML reports and Perl/Matlab source code are available at http://www.ifs.tuwien.ac.at/mir/audiosegmentation/.

6.2 Future Work

Finally, we give the following suggestions for further research:

Chords Chord transcription can be used to obtain the chord sequence of a song. It can be investigated whether this chord representation used as feature vectors improves results over using audio signal feature vectors directly.

Select parameter values song-by-song Since the songs that perform poorly are different for various parameter configurations it seems advisable to develop a procedure or criterion to be able to select an appropriate parameter setting from a pool for each song individually.

User input The potential ameliorative effect of minimal user input can further be looked into. Results may become even better if the system works iteratively, reacting on user input. Possibilities of simple user input include: indication of beginning and end of *one* section; rejection of individual incorrect segment boundaries; etc.

Evaluation (Consistent groundtruth) It is desirable to have a well-founded groundtruth, e.g., by consistently employing always the same musicological approach to all songs. In addition, the performance numbers could be related to the mean evaluation results among groundtruth annotations from different subjects.

MIREX Audio segmentation algorithms from various research teams could be compared within a future MIREX evaluation task. There was no such task in previous MIREX events.

References

[ANS+05] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes. Theory and evaluation of a Bayesian music structure extractor. In *Proc. ISMIR 2005*, pages 420–425, London, UK, 2005.

[AS01] J.J. Aucouturier and M. Sandler. Segmentation of musical signals using Hidden Markov Models. In *Proc. 110th AES Convention*, Amsterdam, The Netherlands, 2001. Preprint Number: 5379.

[ASRC06] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey. Using duration models to reduce fragmentation in audio segmentation. *Machine Learning*, 65(2):485–515, 2006.

[BP92] J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[BW01] M.A. Bartsch and G.H. Wakefield. To catch a chorus: using chroma-based representations for audiothumbnailing. In *Proc. WASPAA*, pages 15–18, New Paltz, New York, USA, 2001.

[BW05] M.A. Bartsch and G.H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, 2005.

[CF03] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In *Proc. WASPAA*, pages 127–130, New Paltz, New York, USA, 2003.

[Cha05] W. Chai. *Automated analysis of musical structure*. PhD thesis, Massachusetts Institute of Technology, MA, USA, September 2005.

[CS06] M. Casey and M. Slaney. The importance of sequences in musical similarity. In *Proc. ICASSP*, volume 5, Toulouse, France, 2006.

[FC03] J. Foote and M. Cooper. Media segmentation using self-similarity decomposition. In *Proc. SPIE Storage and Retrieval for Media Databases*, volume 5021, pages 167–175, Santa Clara, California, USA, 2003.

[Foo00] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. ICME*, volume 1, New York City, New York, USA, 2000.

[GHNO02] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: popular, classical, and jazz music databases. In *Proc. ISMIR*, pages 287–288, Paris, France, 2002.

[Got03] M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. ICASSP*, volume 5, Hong Kong, China, 2003.

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[Hei03] T. Heittola. Automatic classification of music signals. Master's thesis, Tampere University of Technology, 2003.

[HSG06] C. Harte, M. Sandler, and M. Gasser. Detecting harmonic change in musical audio. In *Proc. ACMMM*, pages 21–26, Santa Barbara, California, USA, 2006. ACM Press New York, New York, USA.

[LC00] B. Logan and S. Chu. Music summarization using key phrases. In *Proc. ICASSP*, volume 2, Istanbul, Turkey, 2000.

[LR05] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proc. ISMIR*, pages 34–41, London, UK, 2005.

[LS06] M. Levy and M. Sandler. New methods in structural segmentation of musical audio. In *Proc. EUSIPCO*, Florence, Italy, 2006.

[LS07] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech and Language Processing*, 16(1):318–326, 2007.

[LSC06] M. Levy, M. Sandler, and M. Casey. Extraction of high-level musical structure from audio data and its application to thumbnail generation. In *Proc. ICASSP*, Toulouse, France, 2006.

[LWZ04] L. Lu, M. Wang, and H.J. Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proc. MIR*, pages 275–282, New York, New York, USA, 2004. ACM Press New York, New York, USA.

[Mad06] N.C. Maddage. Automatic structure detection for popular music. *IEEE Transactions on Multimedia*, 13(1):65–77, 2006.

[MXKS04] N.C. Maddage, C. Xu, M.S. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In *Proc. ACMMM*, pages 112–119, New York, New York, USA, 2004. ACM Press New York, New York, USA.

[Ong05] B.S. Ong. *Towards automatic music structural analysis: identifying characteristic within-song excerpts in popular music*. PhD thesis, Universitat Pompeu Fabra, 2005.

[PBR02] G. Peeters, A. La Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *Proc. ISMIR*, Paris, France, 2002.

[Pei07] E. Peiszer. Automatic audio segmentation: Segment boundary and structure detection in popular music. Master's thesis, Vienna University of Technology, Vienna, Austria, 2007.

[PK06] J. Paulus and A. Klapuri. Music structure analysis by finding repeated parts. In *Proc. ACMMM*, pages 59–68, Santa Barbara, California, USA, 2006. ACM Press New York, New York, USA.

[RCAS06] C. Rhodes, M. Casey, S. Abdallah, and M. Sandler. A Markov-chain Monte-Carlo approach to musical audio segmentation. In *Proc. ICASSP*, Toulouse, France, 2006.

[RPM02] A. Rauber, E. Pampalk, and D. Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proc. ISMIR*, pages 71–80, Paris, France, October 13-17 2002.

[TC99] G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *Proc. WASPAA*, pages 103–106, 1999.