A Heuristic Algorithm for DJ Cue Point Estimation

Diemo Schwarz
Ircam STMS, CNRS, Sorbonne Université,
Ministère de la Culture, Paris, France
schwarz@ircam.fr

Daniel Artur Schindler
HearDis! GmbH
Stuttgart, Germany
daniel.schindler@heardis.com

Severino Spadavecchia
HearDis! GmbH
Stuttgart, Germany
rino.spadavecchia@heardis.com

ABSTRACT

This article treats an aspect in a larger research agenda to understand DJ practices, which are an important part of popular music culture: We present a heuristic algorithm that estimates cue points where tracks should cross-fade in a DJ mix. We deduced statistics and heuristics from a list of rules provided by human experts, and from a database of example tracks with given cue regions. We then created an algorithm for cue-point estimation based on rich automatic annotations by state of the art MIR methods, such as music structure segmentation and beat tracking. The results were evaluated quantitatively on the example database and qualitatively by experts.

1. INTRODUCTION

DJ techniques are an important part of popular music culture but are not very frequently the topic of scientific research [1]. This article treats one aspect in a larger research agenda to understand DJ practices. The outcomes from such an understanding are many, for instance musicological research in popular music, cultural studies on DJ practice and reception, music technology for computer support of DJ'ing, automation of DJ mixing for entertainment or commercial purposes.

We focus in this article on the automatic estimation of *cue regions* (see figure 1), i.e. the points in the source audio tracks of a DJ mix, where two tracks should cross-fade.

The work is part of the *ABC_DJ* EU project [1] within which an automatic track annotation and DJ mixing algorithm is to be developed in the context of audio branding for in-store music delivery. It is based on input provided by one of the project partners *HearDis!* [2], an agency for audio branding. Their music experts provided heuristic rules (section 3.1) and an example database of tracks with cue-points (section 3.2). The rules were then verified with the examples (section 3.3) and those rules that could be realised computationally were implemented in a prototype automatic annotation software (section 3.4). The software was run on the example tracks and the results evaluated computationally and by human experts (section 3.5). Their feedback gave rise to adaptations in the algorithm, which improved the quality of the annotations (section 3.6). Thus, the article is structured to give an account of the informal iterative design process that allowed to keep the end-users closely in the loop. Also note that the project context is not DJ mixing for clubs or performance, but point-of-sale (PoS) automatic mixing in shops, based on semi-automatic music annotation and generated playlists. However, the automatically produced mixes should retain a certain DJ quality (beat synchronicity, cross-fades).

[1] http://abcdj.eu
[2] http://heardis.com

2. RELATED WORK

There is quite some existing work on tools to help DJs produce mixes [2–7], bit much less regarding annotation and information retrieval from audio tracks or recorded mixes. The first works opening up research on DJ-related information retrieval are on constitution of playlists [8], segmentation of mixes [9, 10], and identification of tracks within a DJ mix by fingerprinting [11]. The latter team also produce an extensive database of ground truth annotations of playlists with approximate start and stop times of tracks on a large number of Creative-Commons licensed mixes made from open-licensed dance tracks published on the *Mixotic* net label. [3] However, that database was not aimed at and is not precise enough to give information about cue points.

Schwarz and Fourer [1] introduced a larger framework of DJ music information retrieval, where the tracks constituting a DJ mix are sample-aligned to the recorded mix in order to be able to separate the source tracks and estimate the volume fade curves and cue-points. They published an openly available database of artificial but realistic DJ mixes with the necessary ground truth of their construction [12], based on the tracks collected by Sonnleitner et. al. [11].

Kim et al.'s approach of highlight detection [13] chooses segments from a collection of tracks that "stand out" as determined by a convolutional recurrent attention network, and cues them beat-synchronously.

Our work is heavily based on music structure estimation from audio, see [14, 15] for an overview of existing methods. We use the method described in [16] that fuses the representation of a piece of music as a succession of states with similar and acoustically homogeneous content (separated by peaks in a novelty function), and the approach aiming to detect the repetition of sequences in the music.

[3] http://www.mixotic.net

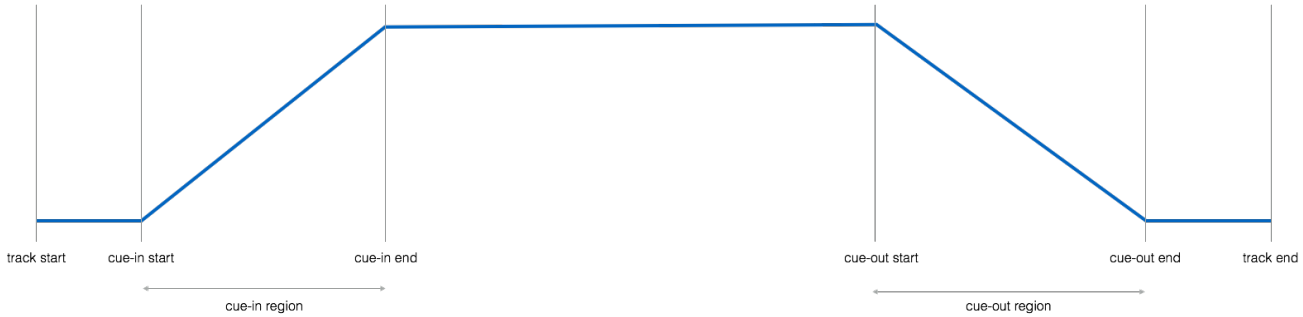

Figure 1. Cue points and cue regions with the resulting volume fade curve for one track in a DJ mix.

The novelty function is computed by convolving a self-similarity matrix with a checkerboard kernel [17].

3. CUE POINT ESTIMATION

Cue points define the regions where tracks in a DJ mix fade in or out to blend with other tracks (see figure 1). DJs will usually choose them by hand according to the context of the current DJ set, based on their experience and familiarity with the specific track. However, when computer support or automation of DJ mixing is called for, we have to devise heuristics and an algorithm that can analyse the music content of a track in order to come close to the human decision.

In our context of PoS automatic mixing, the automatically estimated cue region proposition saves time for the human annotators of new tracks to be included in new automatically generated playlists, who only have to verify the automatic annotation and correct it if necessary.

3.1 Human Expert Rules

In order to get a high-level framing of the problem of cue point estimation from the point of view of the users, project partner *HearDis!* provided the content- and context-based criteria in table 1 for the choice of cue regions for the aim of PoS automatic mixing. The concern in that case is to decide if the song can start immediately, or if the intro has to be shortened, possibly because it is a prolongued club DJ-friendly version, and if the end has to be shortened because in-store music needs to change more often than club music in order to achieve a higher level of variety.

We can already see that many of these points are dependent on audio and musical content (repetition, presence of voice) and even cultural context (what is *too noisy* or *non-musical*?). The key point of the ensuing work was to find out which of these criteria were computationally feasible with the current tools, and whether the rate of errors with regard to the unfeasible criteria not modeled in our algorithm was acceptable.

3.2 Ground Truth Database of Cue-Points

HearDis! provided a set of 30 example tracks in MP3 format, each in two versions:

1. the full length track
2. the track shortened according to human-decided cue-in and cue-out regions with fades applied to them

1. Track is too long in general (more than 6 to 7 minutes)
2. Intro is too repetitive (especially DJ-friendly versions)
3. Intro is too quiet for too long (more than 4 to 8 beats) until track is of discernible loudness (at the PoS). Exception: Artist is already singing
4. Intro is too noisy/non-musical
5. Outro is too repetitive
6. Loudness drops significantly but outro lasts longer than 4 to 8 bars. Exception: Artist is still singing
7. Outro is too noisy/non-musical
8. Generally silence at the beginning and end of a track should be shortened to a minimum

Table 1. Expert-provided criteria for choice of cue regions.

segment	mean / median start time	mean / median / number of non-zero durations
cue in	13.6 / 8.9	1.3 / 0.9 / 23
cue out	290.0 / 316.7	8.4 / 9.2 / 4
track end	369.1 / 363.0	n/a

Table 2. Statistics of start time and duration of ground truth cue regions for the 30 example tracks in seconds.

We then annotated the start and end points of the cut regions, and the durations and kinds of fades or cuts by hand in text label format as produced by AUDACITY [4]. This does provide example cases of shortening and fade times. The results of a statistical analysis of the annotations are given in tables 2–3. Statistics of cue region durations are always given with the zero-duration regions removed, since they correspond to "cut" transitions (no cross-fade).

Furthermore, the examples revealed other content-based decisions, such as, in one track, removing one repetition of the exposition of a synth line by cutting the intro in half, or removing redundancy in long end parts of songs.

[4] http://audacity.sourceforge.net

segment	min / max start time	min / max end time	min / max non-zero duration
cue in	0.0 / 57.4	0.0 / 57.4	0.2 / 4.3
cue out	163.9 / 418.0	166.6 / 428.8	2.0 / 14.6
track end	182.0 / 741.1	182.0 / 741.1	n/a

Table 3. Statistics of minimum/maximum start and duration of ground truth cue regions for the 30 example tracks in seconds.

3.3 Comparison of Ground Truth Cue Points with Music Structure Analysis

The audio-based music structure analysis algorithm [16] divides a piece of music into its significant parts, and organises them into classes (e.g. corresponding to intro, outro, chorus, verse). This is done on multiple-levels, where, from the lowest to the highest level, the structural segments are fused into larger classes.

Our hypothesis was that the intro and outro segments would stand out and could be a good basis for cue regions. We verified this by calculating two versions of automatic structural analysis, using the tool IRCAMSUMMARY, on the full length example tracks. The first version, *state* mode [16], is based on a fusion of homogeneous state and sequence repetition segmentations. The second version, *NSMF* mode [18], uses non-negative matrix factorisation (NMF) of similarity matrices as a mid-level representation to classify the structure. This mode is called *NSMF* for *Non-negative Similarity Matrix Factorization*.

We then compare the structural segments with the human suggestions of cue regions. The plots in figures 2 and 3 show the ground truth cue regions of the example tracks overlaid with multi-level structure analysis regions. Each level's segment boundaries are shown on one horizontal line as coloured dots, from lower levels in violet to higher levels in cyan, which proceed by fusing lower-level segments. Right-pointing triangles show the cue-in fade region, or cut, when only one triangle is visible (length of zero). Left-pointing triangles show the cue-out region and end point of the full length example. Tracks are sorted by length, for easier observation of maximum final track length.

These plots reveal that, first, the lowest (most detailed) level *state* mode summary in figure 2 is more pertinent, since it has a segment structure better coinciding with the annotations (it is also beat-synchronous, unlike the *NSMF* mode summary in figure 3), and, second, that almost half of the songs were not shortened at the beginning:

- 14 cue-in start points are cuts at song start
- 16 cue-in start points are within the first structure segment
- only 3 cue-in segments are longer than 1s [5]
- the cut-off point for long tracks is mostly between 5:30 and 6:00, with 3 tracks going until 7 minutes

3.4 Cue Point Estimation Heuristic Algorithm

The first proposed algorithm detailed below is solely based on the song structure estimation by the IRCAMSUMMARY module, which determines significant parts of the track, including the intro and outro sections which are used to place the cue regions. For this first iteration, we wanted to see how far we would come only with song structure information, without taking the audio content into account.

The above observations from the example tracks suggest the following heuristic algorithm for the cue region estimation (always of 10 s length):

```
1: function CUE-IN
2:   return cue region such that its end coincides with
         the end of the first long enough (10 s) struc-
         tural region at the lowest level
3: end function
```

```
1: function CUE-OUT
2:   if the song is not too long (<= 6 min) then
3:     return start of the last structural region that is
           long enough (10 s) as cue-out start
4:   else
5:     ▷ for long songs, we apply the explicit rule to
         shorten songs that are too long (see 3.1)
6:     if there is a structural segment longer than 10 s in
         the time span between 5:30 and 7 minutes [6]
7:     then
8:       return the cue-out region placed at its start
9:     else
10:      return cue-out start at 5:30
11:    end if
12:  end if
13: end function
```

3.5 First Evaluation of Cue-Point Detection

We created cue region estimations for the tracks in the cue-point example database. To facilitate evaluation, we also exported the example tracks faded at the estimated cue-points. These examples were evaluated by music annotators at *HearDis!* in order to validate the heuristics. (But keep in mind that the cue-point estimation is always only a starting point for a human annotator who solely is in the position to correctly judge the content and context of the tracks.)

Nevertheless, the numerical comparison between the estimated cue points and the hand-annotated ground-truth cue points from the test database in figure 4 shows that over 50% of estimated cue points are within 10 seconds from the manual choice and for two thirds of the examples the absolute time differences are smaller than 20 seconds. The two outliers with cue end estimation differences over 60 s are due to a more flexible interpretation of the shortening rule by the human annotators (some tracks were left at much longer than the cutoff rule of 6 minutes).

[5] The frequent presence of cuts instead of fade regions are due to the fact that the existing PoS playout system at *HearDis!* does not yet do cross-fade mixing, so that the experts are trained to find cue points where tracks can cut from one to the next.

[6] These times are suggested by the ground truth database.

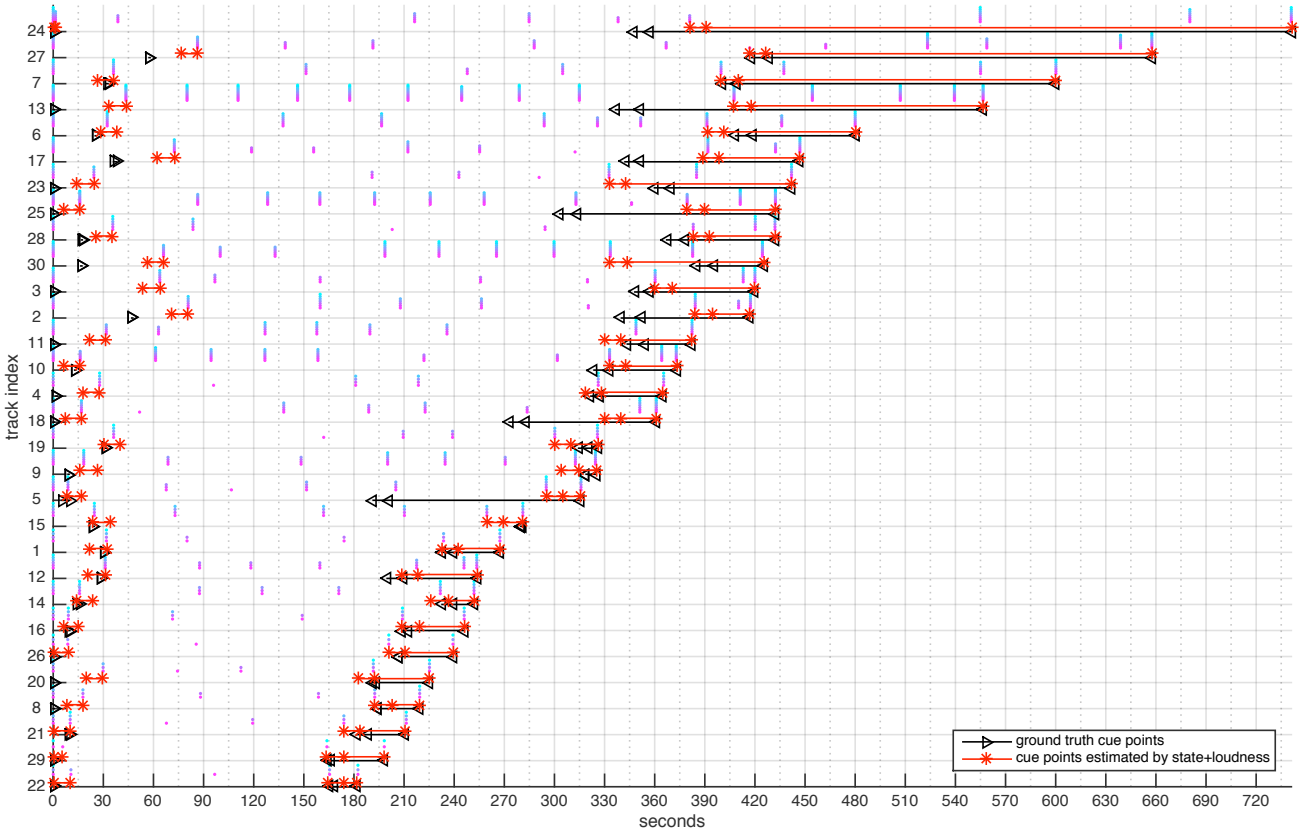

Figure 2. Ground truth cue regions of the example tracks (black) overlaid with IRCAMSUMMARY multi-level structure analysis regions in *state* mode (violet to cyan dots), and cue-points estimated by final algorithm (red).

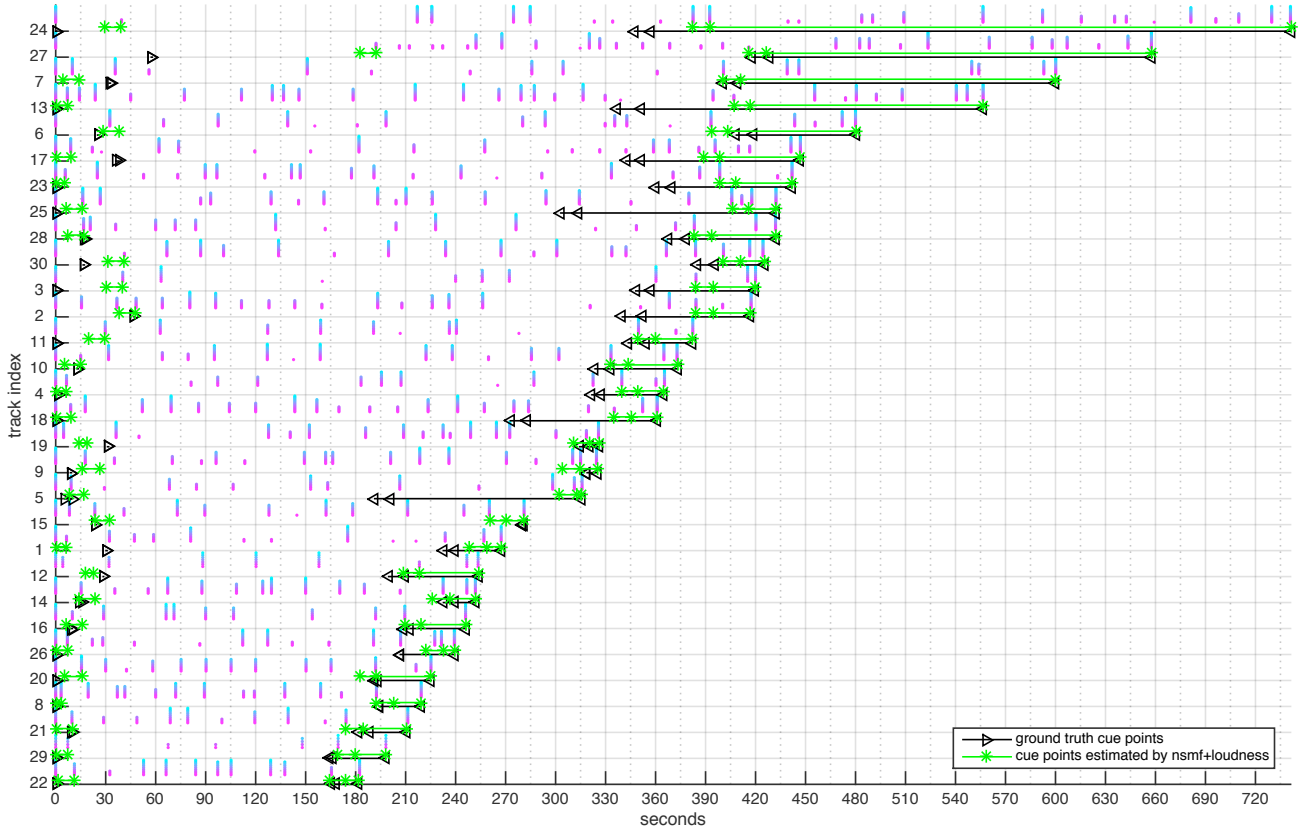

Figure 3. Ground truth cue regions of the example tracks (black) overlaid with IRCAMSUMMARY multi-level structure analysis regions in *NSMF* mode (violet to cyan dots), and cue-points estimated by final algorithm (green).

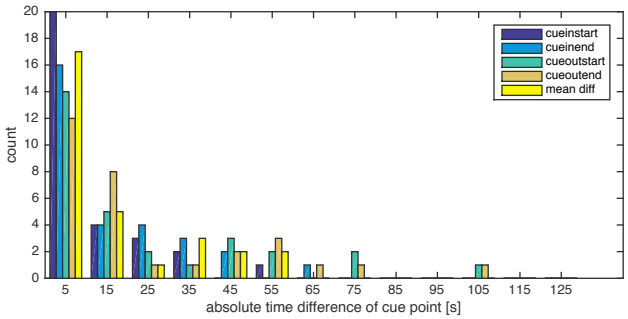

Figure 4. Histogram of the time difference between hand-annotated ground-truth cue points and cue points estimated based on *state* mode.

A subjective but systematic evaluation of the automatically faded example tracks was carried out by the human music experts at *HearDis!*. They provided precise feedback in the form of screenshots with the 3 versions of each track aligned (original, human-cut, automatically cut) and remarks for the problematic cases (see figure 5).

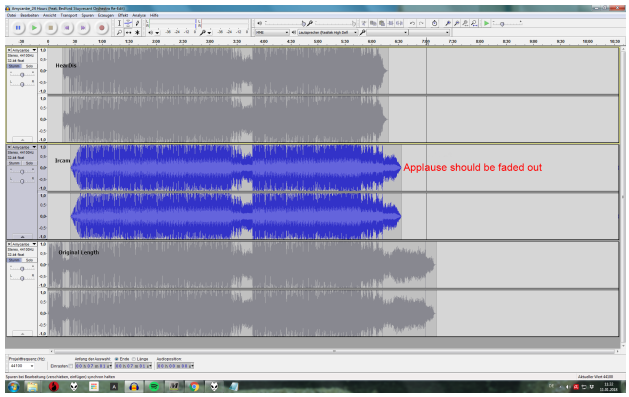

Figure 5. Example of subjective feedback: The human expert hand-aligned the original track (bottom), the manually cut and faded track (top), and the automatically cut and faded track (middle) to evaluate and comment on important differences.

The feedback was positive about the algorithmic choice of cue points, with the remark that, for PoS applications, it is always OK to cut more than a human annotator at beginning and end. There were only 5 problematic cases, listed in table 4. The remarks show the limits of our simple algorithm, where the human decision mobilises deep content- and context-dependent knowledge up to the cultural level (e.g. that applause is a special noise that marks the end of a performance).

3.6 Second Iteration

Based on the above feedback, we chose to implement the only computationally feasible content based criterion for cue-in/cue-out estimation of loudness. From an analysis of the example tracks, we could determine that a threshold of -9 dB relative to the max loudness of the track catches all the cases where an intro or outro had been considered as too quiet, without introducing false positives. [7]

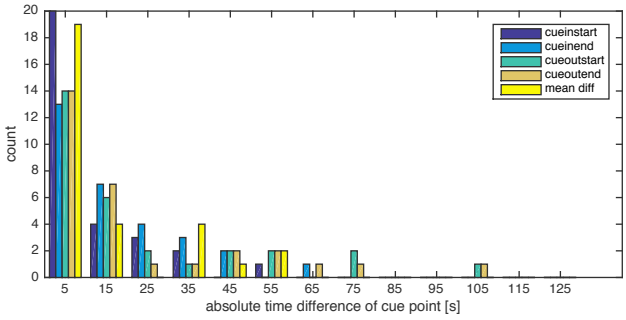

Figure 6. Histogram of the time difference between hand-annotated ground-truth cue points and estimated cue points with loudness criterion.

We then shift the cue region until its minimum loudness is larger than -9 dB relative to the max loudness of the track. Loudness of a segment is calculated as the max peak RMS energy in 2s windows.

The second evaluation results, shown in figure 6, slightly improve the difference between estimated cue-regions and manual choice, and has been approved by the human experts.The found cue points are also shown in figures 2 and 3.

Musically, the decision to place the cue-in region at the end of a song-structure region work very well, since at the end of the fade in of the new track, just when it has reached full volume, and the previous song has just vanished, there is a clear change in content (as predicted by the song structure), that catches the ear and clearly signals the start of the track.

4. CONCLUSIONS

We presented a heuristic algorithm to estimate cue points for generating DJ-like mixes based on automatic annotations by state of the art MIR methods of music structure segmentation, coupled with domain knowledge of human experts, and backed by a database of example tracks. The iterative design process created a close feedback loop between researchers, developers, and expert users, quickly reaching a satisfactory solution for their specific needs of in-store music playout and audio branding.

The next step we will take is to determine fade-in and fade-out times relative to the beat positions of the track, as estimated by the beat marker annotation of the IRCAM-BEAT module. The cue regions would last for 4 measures, and it is straightforward to adapt the algorithm to search for the beat position closest to a structural boundary to anchor a cue region.

In future work, we could examine which of the criteria in section 3.1 are possibly detectable by content descriptors and classifiers available in MIR research, e.g. voice detection, or develop specific descriptors and classifiers for the "music-ness" of audio. However, this would need many

[7] Note that our algorithm will only ever encounter professionally produced music that is optimised for being loud and punchy to stand out in radio or streaming listening conditions, so we're fairly confident that that threshold will be generalisable.

position	evaluator remark	computable	observations
end	end applause should be faded out	no	music continues during applause
end	just noises?	no	free guitar + voice
end	too noisy and weird	no	fade in of 2sec of Morse code
start	intro too long?	yes	intro very low volume -24dB
end	outro too long	yes	outro -15 dB (last struct segment silence)

Table 4. Feedback on individual tracks in the first version of cue-point detection output by human expert, and assessment of computability.

more annotated tracks to train the method. Before this effort is made, feedback should be gathered about the number of problematic cases in real-world usage of the existing system.

Acknowledgments

The *ABC_DJ* project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 688122.

5. REFERENCES

[1] D. Schwarz and D. Fourer, "Towards Extraction of Ground Truth Data from DJ Mixes," in *International Symposium on Music Information Retrieval (ISMIR), late breaking demos*, Suzhou, China, Oct. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01671768

[2] D. Cliff, "Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks," *HP Laboratories Technical Report HPL*, vol. 104, 2000.

[3] T. Fujio and H. Shiizuka, "A system of mixing songs for automatic DJ performance using genetic programming," in *6th Asian Design International Conference*, 2003.

[4] H. Ishizaki, K. Hoashi, and Y. Takishima, "Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort." in *ISMIR*, 2009, pp. 135–140.

[5] F. X. Aspillaga, J. Cobb, and C.-H. Chuan, "Mixme: A recommendation system for DJs," in *Late-break Session of the 12th International Society for Music Information Retrieval Conference*, 2011.

[6] P. Molina, M. Haro, and S. Jordá, "Beatjockey: A new tool for enhancing DJ skills." in *NIME*, 2011, pp. 288–291.

[7] M. E. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: Automatic creation of multi-song music mashups," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1726–1737, 2014.

[8] T. Kell and G. Tzanetakis, "Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction." in *ISMIR*, 2013, pp. 505–510.

[9] N. Glazyrin, "Towards automatic content-based separation of DJ mixes into single tracks," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, Oct. 2014, pp. 149–154.

[10] T. Scarfe, W. Koolen, and Y. Kalnishkan, "Segmentation of electronic dance music," *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 22, no. 3, p. 4, 2014.

[11] R. Sonnleitner, A. Arzt, and G. Widmer, "Landmark-based audio fingerprinting for DJ mix monitoring," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, New York, NY, 2016.

[12] D. Schwarz and D. Fourer, "A dataset for DJ-mix reverse engineering," in *submitted to International Symposium on Music Information Retrieval (ISMIR), late breaking demos*, Paris, France, Sep. 2018.

[13] A. Kim, S. Park, J. Park, J.-W. Ha, T. Kwon, and J. Nam, "Automatic DJ mix generation using highlight detection," in *Proc. ISMIR, late-breaking demo paper*, 2017.

[14] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *ISMIR*, 2010, pp. 625–636.

[15] G. Peeters, "Deriving musical structures from signal analysis for music audio summary generation: "sequence" and "state" approach," in *Computer Music Modeling and Retrieval*, U. K. Wiil, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 143–166.

[16] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery," in *ISMIR (International Society for Music Information Retrieval)*, Curitiba, Brazil, 2013, pp. –. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01106873

[17] J. Foote, "Visualizing music and audio using self-similarity," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. ACM, 1999, pp. 77–80.

[18] F. Kaiser, "Music structure segmentation," Ph.D. dissertation, TU Berlin, 2012.