TOWARDS AUTOMATIC CONTENT-BASED SEPARATION OF DJ MIXES INTO SINGLE TRACKS

Nikolay Glazyrin
Ural Federal University
nglazyrin@gmail.com

ABSTRACT

DJ mixes and radio show recordings constitute an important and underexploited music and data source. In this paper we try to approach the problem of separation of a continuous DJ mix into single tracks or timestamping a mix. Sharing some aspects with the task of structural segmentation, this problem has a number of distinctive features that make difficulties for structural segmentation algorithms designed to work with a single track. We use the information derived from spectrum data to separate tracks from each other. We show that the metadata that usually comes with DJ mixes can be exploited to improve the separation. An iterative algorithm that can consider both content-based data and user provided metadata is proposed and evaluated on a collection of freely available timestamped DJ mix recordings of various styles.

1. INTRODUCTION

DJ mixes provide a great source of music data, which does not gain much attention from the MIR community yet. The work by Kell and Tzanetakis [6], which gives an analysis of track selection and ordering in DJ mixes is one of the few exceptions.

Besides playing in clubs many DJs nowadays produce weekly radio shows with latest and greatest and sometimes exclusive tracks. These shows are often freely available through the internet and are very popular among electronic music lovers. Tracklists for the shows are often provided by DJs themselves or by their fans.

For many people it is important to know which track is playing now. The cue sheet file format [2] suits perfectly to carry this kind of information. It was designed to describe how the tracks on CD are laid out, but later it was supported by many audio players and CD burning software. There are communities, such as `http://cuenation.com` or `http://themixingbowl.org`, which bring together the people who create cue sheets for DJ mixes and radio shows. But the wiki page [1] on the first site says nothing about any tools for automatical or semi-automatical generation of cue sheets.

The most time consuming part of this process is finding the moments when one track gives place to another. This may be a big problem for an untrained listener, because making smooth transitions between tracks is one of the skills every DJ should have. For a trained person it is not so hard, but to find a precise position of a transition one has to listen carefully through dozens of seconds of the audio. A tool that can propose most probable transition positions can facilitate this task. Such a tool can also be used by DJs who upload their mixes to special sharing services or online radio stations. These services will be able to timestamp the mix automatically instead of forcing the uploader to do this. The timestamps may be then used to provide fast access to particular tracks within the mix and to easily share previews of unreleased tracks played in radio shows. Timestamped recordings of DJ mixes can be used by recommendation systems to calculate content-based features and relate them to sequential tracks.

The task of DJ mix separation is essentially the task of audio segmentation, so the concepts and approaches can be shared between these tasks. But some conditions and requirements make them different. These differences will be discussed in section 2. In section 3 we describe the proposed method to separate tracks in DJ mix recordings. In section 4 we describe the experiments and the evaluation methodology. Finally, in section 5 we conclude and formulate open problems and directions for future work.

2. PROBLEM FORMULATION AND RELATED WORK

Music structural segmentation is a very popular and elaborated task. Paulus et al. in [9] distinguish three different classes of music segmentation methods. Repetition-based methods try to identify recurring patterns. Novelty-based methods try to find transitions between contrasting parts. Homogenity-based methods, contrary to novelty-based ones, try to determine fragments that are consistent with respect to some characteristic. Combined methods have also been proposed. Some recent ones try to combine novelty-based and homogenity-based approaches [4] or combine novelty-based approach with harmonical information in a joint probabilistic model [10].

A DJ mix can be viewed as a very long composition of individual tracks. These tracks constitute the segments in our task. It is important that no track can occur more than once within a typical mix. So repetition-based methods are

not suitable at the level of tracks.

Novelty-based approach seems to be the most suitable for track boundaries detection. Algorithms that implement this approach generally have 2 main steps: segmentation and grouping.

Segmentation is usually done using an intermediate representation in the form of self-similarity matrix (or self-distance matrix). Since the original audio is not very informative, it needs to be transformed into a sequence of feature vectors, for which this matrix is calculated. The list of features often used for this includes MFCCs, constant-Q spectrum, various low-level spectrum features, such as spectral centroid, spectral spread and others.

The most popular method of obtaining initial segmentation from a self-similarity matrix was proposed by Foote [3]. It is based on so called checkerboard novelty kernels, which are essentially an $M \times M$ matrix with checkerboard-like structure. Novelty estimations can be obtained by convolving this kernel along the main diagonal of the self-similarity matrix. Peaks of the resulting novelty function provide the initial segment borders.

Homogenity-based methods come up as a direct continuation of this novelty-based segmentation. They group similar segments together. A good review of the whole variety of methods can be found in [9]. Many of them perform clustering of segments, e.g. [7], [5]. Any information about the desired result can be helpful at this stage to build the most effective grouping procedure.

In case of DJ mix separation the grouping procedure becomes especially important. It is quite common for dance compositions to have a so called "break" in the middle, where the sound can change dramatically. Such breaks should be overcome to properly detect track boundaries. At the same time, two adjacent segments that belong to different tracks should not be joined.

A typical DJ mix lasts considerably longer than a typical musical composition. So the method must be able to work with recordings that span hours of audio. On the other hand, this loosens the requirements to border detection: an error of seconds or sometimes even tens of seconds can be acceptable. Even humans can have different opinions about one exact moment when a track has transitioned to the next one. An interesting task of detecting transition periods (where two or more tracks are playing simultaneously) comes up here, but we don't consider it in this paper. Marolt in [8] works with similar time scale and boundaries requirements, but with a limited set of possible segment types that sound quite differently.

Transitions can vary significantly for different music styles. It is more likely to find sharp cuts in drum'n'bass mixes, than in deep house mixes, which tend to have long gradual transitions. Average track length is also dependent on music style. But these are generally not the strict rules.

Radio shows often have an intro, which is played in the beginning and often becomes a part of the tracklist. Jingles, interludes or talks where the music gets faded can occur at random places within a recording. But it is not required to discriminate them, as they usually don't get included into tracklists.

The existence of tracklists also makes a great difference from structural segmentation task. It can be seen from the potential applications described in section 1, that the separation of a DJ mix is not much valuable per se. But it becomes really useful when it can be connected with metadata: artist name and track title. Because this metadata is often available, it can also be used in the algorithm. For example, the information about the number of segments in the separation gives a barrier to segmentation and/or grouping process. And if a large music base is available to the algorithm, parts of a mix can be matched to corresponding music recordings to provide even better estimation of track borders.

There may be the cases where matching is not possible though. Sometimes DJs play tracks that are not yet released officially, and therefore cannot appear in any catalogue or database. Some tracks never get released officially. Some tracks have been released years ago, and it's almost impossible to obtain rights on them or find them in any database. That is why the development of the informed automatic DJ mix separation system cannot be reduced to a number of calls to track identification software.

Therefore, further we will suppose that there is a tracklist available for a mix recording, but not the timestamps, and no identification software is available. And the task will be to determine those timestamps based on the audio data and the information from the tracklist, or to align the mix tracklist to the audio. The authors are not informed about any works on this task existing at the moment.

3. SYSTEM DESCRIPTION

We adopt the approach based on novelty-based segmentation followed by grouping of similar segments.

3.1 Features

Constant-Q log-spectrograms are calculated at first for audio recordings, which sampling frequency has been left at the default value of 44100 Hz. We used Constant Q plugin from Queen Mary vamp plugin set [1] with the following parameters: step size and block size are both equal to 16384 samples (0.37 s), 12 components per octave, spanning MIDI pitches from 36 to 84 (65 to 936 Hz) with tuning frequency of 440 Hz. A relatively large block size and zero overlap have been chosen because of the large time scale and to speed up computations. Low frequencies are captured, because electronic dance music often has very accented bass that changes from track to track. The upper frequency limit has been chosen rather arbitrarily, and we do not investigate its influence in this study.

A sliding 2D median filter is then applied to spectrogram with window size (31, 1) (which corresponds to 11.5 seconds and 1 spectral component) to smooth it.

[1] http://www.vamp-plugins.org/plugin-doc/qm-vamp-plugins.html

3.2 Segmentation

To accelerate calculations, the self-distance matrix is calculated for a spectrogram with 10 times less resolution by time axis (3.7 s per column), where each 10 sequential columns of original spectrogram are replaced with their average. We also restricted it to only include cosine distances between segments which are no more than 10 minutes apart from each other, because it is very unlikely to meet a track that lasts longer than that in a DJ mix.

Novelty score is then calculated from the self-distance matrix using the checkerboard kernels with gaussian taper proposed in [3]. We used relatively small kernels of size 16 (composed of 4 squares of size 8×8). All the peaks of the resulting novelty function form the initial set of borders.

3.3 Clustering

Here we find a use for the information from the mix tracklist. The total number of tracks provides the desired number of clusters. This is an imporant advantage over the traditional segmentation task, where the number of segments is unknown. On the other hand, there is a very strong requirement to the borders between segments. If one true border is not detected or one false border is detected in the beginning of the mix, all the subsequent tracks become misaligned with the real audio, even if all the other borders are detected perfectly.

Another piece of information from the tracklist that can be used here is the presence of intro and outro. Many radioshows and regular podcasts have such an intro, fewer ones have also an outro. These segments are relatively short (shorter than 1 minute), but are often included in tracklists. A reasonable assumption is that if the name of the first track contains the string *intro* and/or the name of the last track contains the string *outro*, then an intro and/or an outro should be expected. A good clustering algorithm could be able to detect them automatically, but we add a special handling for these cases. If an intro is expected, among the novelty function peaks during the first 60 seconds of audio the highest one is selected and declared as the intro right border. The same is done at the end of the recording if an outro is expected there.

For the remainder of the recording an iterative clustering procedure is applied. Within each segment the average of all its feature vectors is calculated and normalized by dividing all its components by the maximal one. All the pairwise distances between segments whose beginnings are not more than 600 seconds away from each other are calculated as Euclidean distances between their average feature vectors. This gives a Segment Distance Matrix similar to the one introduced in [4].

All the segment pairs $((l_i, r_i), (l_j, r_j)),\ i < j$ (where l_i and r_i are correspondingly segment's left and righ borders) for which the distance was calculated are sorted according to the following condition: $D_{ij} \cdot (r_j - l_i)$, where D_{ij} is the distance between i-th and j-th segments. Only the pair that produces the smallest value is then merged. If the segments from this pair are not contiguous, all the intermediate ones are also included. To avoid too big segments, a pair gets a penalty when $r_j - l_i > 1.25 \cdot average\_track\_length$: its condition becomes $100000 \cdot (r_j - l_i)$.

4. RESULTS

The proposed method was evaluated on a collection of 103 DJ mix recordings [2] downloaded from free online sources. The corresponding timestamped tracklists in the form of `.cue` files were downloaded from `http://cuenation.com` and used without any correctons. Timestamps have only been used to validate the correctness of track separation. All recordings were taken from different radio shows and live sessions of different disk jokeys. Most of recordings are dated 2014, but there were also recordings from 2007-2013. The dominant music style within the selected recordings is trance (uplifting, progressive, big room, psychedelic), probably due to overall popularity of DJs playing this music. But house, drum'n'bass, breakbeat, techno, hardstyle, downtempo mixes are also included.

For the reasons described in section 3.3 we pay less attention to the conventional precision and recall metrics. Instead, two values have been calculated for each mix: the average and the maximum absolute distances in seconds from true track beginnings to detected ones. This way we can evaluate the usefulness of the method in real life applications: if the average absolute distance approaches the average track length within a mix, the method becomes nearly useless for this mix. The maximum absolute distance gives an estimation of the worst case. These values are then averaged across the whole collection to give an integral measure of method performance.

Frame-based pairwise precision, recall and F-measure have also been calculated to provide more traditional estimation of segmentation quality. They are defined as follows. Each recording is separated into 1 second frames. All frame pairs where both frames belong to the same track form the sets P_E (for the system result) and P_A (for the ground truth). The *pairwise precision rate* can be calculated by $P = \frac{|P_E \cap P_A|}{|P_E|}$, *pairwise recall rate* by $R = \frac{|P_E \cap P_A|}{|P_A|}$, and *pairwise F-measure* by $F = \frac{2PR}{P+R}$. These values are then also averaged across the collection.

As a baseline we will use the same values calculated for the naive separation, where all track borders are evenly spaced within the mix and all tracks have the same duration. In case of explicit intro/outro information the naive separation will allocate them 30 seconds in the beginning or in the end of the mix.

In the first experiment [3] the system was not informed about the presence of intro and outro sections in the mixes. The results are shown in Table 1. The "Good" column shows the number of mixes where the average absolute distance is less than 90 seconds (rather arbitrary limit). From the numbers in this table it seems that the proposed method performs not much better than the naive separation, which

[2] The list of file names is available from `https://github.com/nglazyrin/MixSplitter/blob/master/mix_list.txt`

[3] Full log is available from `https://github.com/nglazyrin/MixSplitter/blob/master/logs/paper_test.log`

Separation	CAvg. abs. dist.	CAvg. max dist.	Good
Proposed	143.73 s	328.99 s	42
Baseline	152.83 s	318.35 s	30

Table 1. Results with no information about intro and outro.

Separation	CAvg. abs. dist.	CAvg. max dist.	Good
Proposed	111.82 s	286.61 s	62
Baseline	126.87 s	284.41 s	49

Table 2. Results with information about intro and outro.

is confirmed by p-value of 0.096 returned by Wilcoxon test. But looking closer at the performance on particular mixes, we can see that in some cases the proposed method has real advantage. E.g. for the mix *M.PRAVDA - Best of 2013 (Part 2) (promodj.com).mp3* it gives average absolute distance of 8.59 s (which is great) versus 60.22 s obtained by the naive separation. On the other hand, for some mixes (e.g. *Trancecoda Podcast 008 - GMix Eddie Bitar.mp3*) the average absolute distance exceeds 6 minutes, which is absolutely inacceptable.

In the second experiment [4] the system was informed about the presence of intro and outro secions and could react appropriately. From the Table 2 we can see that this information can be really helpful. In this experiment the $p < 0.01$ was returned by Wilcoxon test. The result has moved nearer to the "Good" limit of 90 seconds average difference, and the difference between the proposed and the baseline methods became bigger. And if the limit of "goodness" has decreased to 60 seconds, the difference gets more explicit: 54 good separations by the proposed method versus 24 good naive separations. For 30 seconds limit on average absolute difference only 25 versus 6 good separations are left.

This result shows that the proposed method can give good result for a reasonable amount of mixes (62 out of 103 here). But for some mixes the results are still too bad. We provide two case-studies that describe common errors of the method.

Table 3 shows the comparison of true and detected borders for one of the mixes – *4H_Community_Guest_Mix_The_2nd_Anniversary_of_Room51_Show_by_Breeze_Quadrat_PureFM.mp3* – with average absolute difference of 177.11 seconds. First 3 tracks are aligned good, but then the system detects wrong border in the middle of 4th track. In spite of more or less properly detected other borders (the detected value in row $i+1$ is near the true value in row i), they all mark beginnings of track $i+1$ instead of i-th track.

The same information is represented graphically on Figure 1. Vertical yellow lines on the constant Q spectrogram mark the true borders, vertical black lines correspond to detected borders.

The errors of this kind can be overcome with a better

[4] Full log is available from `https://github.com/nglazyrin/MixSplitter/blob/master/logs/paper_test_explicit_intro_outro.log`

No.	Detected	True	Difference
1	0.00 s	0.00 s	0.00 s
2	308.38 s	312.08 s	3.70 s
3	628.57 s	613.00 s	-15.56 s
4	872.56 s	1029.11 s	156.55 s
5	1025.19 s	1363.48 s	338.29 s
6	1360.54 s	1757.29 s	396.75 s
7	1757.15 s	1961.62 s	204.47 s
8	1970.53 s	2292.27 s	321.74 s
9	2321.36 s	2552.34 s	230.98 s
10	2748.30 s	2979.58 s	231.28 s
11	3247.66 s	3198.78 s	-48.88 s

Table 3. Detailed result for the mix by 4H Community.

Separation	Precision	Recall	F-measure
Proposed (1)	0.8145	0.7761	0.7941
Baseline (1)	0.7024	0.6397	0.6688
Proposed (2)	0.8077	0.7892	0.7977
Baseline (2)	0.7069	0.6637	0.6839

Table 4. Framewise precision, recall and F-measure.

sorting function for segment pairs or with a different segment grouping strategy. As can be seen from Table 4 (the number in parentheses in the first column corresponds to the experiment number), the proposed method really locates borders much better than the baseline. But since some borders are misplaced, the final pairwise precision and recall rates are not so close to 1 as they could be.

Another source of errors are mixes that contain tracks of various durations, e.g. a pile of 1 minute long tracks followed by 4 minute long tracks, or several interludes throughout the recording. An example of such mix is *01-friction_-_bbc_radio1_(chase_and_status_special)-sat-10-13-2013-talion.mp3*, which contains 35 tracks per 2 hours, and 6 of them are grouped between 55 and 65 minutes. The separation is shown on the Figure 2. The described method tends to join short segments and to return more or less evenly spaced track borders because of the sorting condition and the penalty for long tracks. So it does not fit to these highly-variable mixes, which are characteristic for music genres such as drum'n'bass. But the separations obtained without using the penalty were worse than the ones obtained by the baseline method.

Table 5 groups the resutls by music genres, which were manually annotated for each mix. The mixes labeled as having *various* genre contain tracks from two or more very different genres, such as house and drum'n'bass. The Cnt column gives the total count of mixes of a given genre within our test collection.

Because the test set is very unbalanced by music genre (which is dictated by the available cue sheet files), it's hard to make conclusions for music genres other than house and trance (which can be themselves separated into various subgenres). The proposed system outperforms the baseline method on these genres, but both methods are failing on

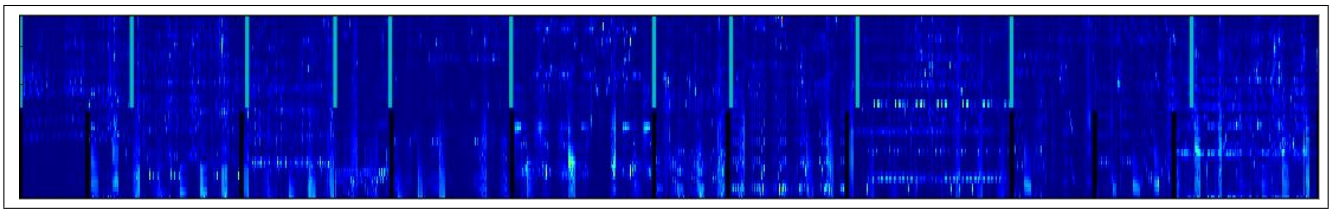

Figure 1. The separation for the mix by 4H Community.

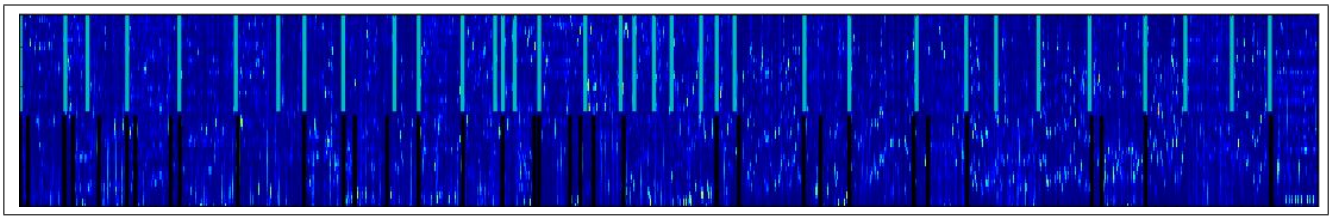

Figure 2. The separation for the mix by Chase & Status.

Style	Cnt	Separation	Abs. dist.	Max dist.
trance	59	Proposed	91.43 s	244.26 s
		Baseline	114.85 s	255.38 s
house	29	Proposed	106.55 s	280.78 s
		Baseline	117.88 s	270.36 s
techno	4	Proposed	122.11 s	284.51 s
		Baseline	104.61 s	219.00 s
downtempo	3	Proposed	304.59 s	702.77 s
		Baseline	308.58 s	609.34 s
hardstyle	2	Proposed	81.91 s	232.66 s
		Baseline	93.22 s	221.95 s
drum'n'bass	2	Proposed	330.67 s	798.21 s
		Baseline	343.03 s	865.92 s
various	2	Proposed	211.28 s	429.65 s
		Baseline	203.85 s	407.48 s
breakbeat	2	Proposed	191.88 s	399.71 s
		Baseline	124.22 s	346.01 s

Table 5. Results by music genre.

downtempo and drum'n'bass music.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method for informed content-based separation of DJ mixes into single tracks that outperforms a naive baseline evenly separating method. We showed that this method provides good results for a reasonable amount of mixes. The resulting separations are good enough to use them for further applications. We also showed how a simple information about the presence of intro and outro sections in the mix can improve the separation quality.

This paper establishes a basis for further work on DJ mixes separation. Another clustering methods need to be developed to prevent false border detection errors and border miss errors. It makes sense also to include higher frequencies into the initial spectrum, as they may carry some meaningful details. On the other hand, the novelty detection method does not seem to have a major impact, because the initial border candidate set if sufficiently large to select values nearby the true borders.

More feature types need to be exploited. It also makes sense to consider the tempo information to avoid false border detections, because the tempo does not change often during transitions, but changes within a track when a break starts or ends. A deeper modification or a new method is needed to handle mixes that contain tracks with highly-varying durations. A separate method to detect interludes and talks can be helpful here.

Finally, a significant improvement may be expected from the usage of a track identification system, as it may help to align at least some of the tracks properly. But this poses a separate technical and legal task.

6. REFERENCES

[1] Online: `http://wiki.themixingbowl.org/Cue_sheet`, acessed on May 5, 2014.

[2] Online: `http://wiki.hydrogenaudio.org/index.php?title=Cue_sheet`, accessed on May 5, 2014.

[3] J. Foote: "Automatic audio segmentation using a measure of audio novelty" *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 452–455, 2000.

[4] F. Kaiser, and G. Peeters: "A simple fusion method of state and sequence segmentation for music structure discovery" *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 257-262, 2013.

[5] F. Kaiser, and T. Sikora: "Music Structure Discovery in Popular Music using Non-negative Matrix Factorization" *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 429–434, 2010.

[6] T. Kell and G. Tzanetakis: "Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction" *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 505-510, 2013.

[7] M. Levy, M. Sandler, and M. Casey: "Extraction of High-Level Musical Structure From Audio Data and Its Application to Thumbnail Generation" *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2006*, Vol. 5, 2006.

[8] M. Marolt: "Probabilistic Segmentation and Labeling of Ethnomusicological Field Recordings" *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pp. 75–80, 2009.

[9] J. Paulus, M. Müller, and A. Klapuri: "Audio-based Music Structure Analysis" *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 625–636, 2010.

[10] J. Pauwels, F. Kaiser, and G. Peeters: "Combining harmony-based and novelty-based approaches for structural segmentation" *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 601-606, 2013.