Methods and Datasets for DJ-Mix Reverse Engineering

Diemo Schwarz[1] and Dominique Fourer[2] *

[1] Ircam Lab, CNRS, Sorbonne Université, Ministère de la Culture, Paris, France
[2] IBISC, Université d'Évry-Val-d'Essonne/Paris-Saclay, Évry, France
schwarz@ircam.fr

Abstract. DJ techniques are an important part of popular music culture. However, they are also not sufficiently investigated by researchers due to the lack of annotated datasets of DJ mixes. Thus, this paper aims at filling this gap by introducing novel methods to automatically deconstruct and annotate recorded mixes for which the constituent tracks are known. A rough alignment first estimates where in the mix each track starts, and which time-stretching factor was applied. Second, a sample-precise alignment is applied to determine the exact offset of each track in the mix. Third, we propose a new method to estimate the cue points and the fade curves which operates in the time-frequency domain to increase its robustness to interference with other tracks. The proposed methods are finally evaluated on our new publicly available DJ-mix dataset. This dataset contains automatically generated beat-synchronous mixes based on freely available music tracks, and the ground truth about the placement of tracks in a mix.

1 Introduction

Understanding DJ practices remains a challenging important part of popular music culture [2, 4]. The outcomes from such an understanding are numerous for musicological research in popular music, cultural studies on DJ practices and reception, music technology for computer support of DJing, automation of DJ mixing for entertainment or commercial purposes, and others. In order to automatically annotate recorded mixes, several components are required:

Identification of the contained tracks (e.g. fingerprinting) to obtain the playlist,
Alignment to determine where in the mix each track starts and stops,
Time-scaling to determine what speed changes were applied by the DJ to achieve beat-synchronicity,
Unmixing to estimate the cue regions where the cross-fades between tracks happen, the curves for volume, bass and treble, and the parameters of other effects (compression, echo, etc.),
Content and metadata analysis to derive the genre and social tags attached to the music to inform about the choices a DJ makes when creating a mix.

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

Most of these components have been addressed by recent MIR research except the alignment, time-scaling, and unmixing part for which we propose a method based on multi-scale correlation, dynamic time warping, and time-frequency gain curve estimation to increase its robustness to interferences with other tracks. To come closer to actual DJ practices, we can retrieve the alignment and volume curves from example DJ mixes, and then combine them with content and genre information to investigate the content-dependent aspects of DJ mix methods.

As a working definition, we can roughly distinguish three levels of mixing:

Level 1, *broadcast mixing*, is a simple volume cross fade without paying attention to changing content (as performed by consumer audio players such as iTunes, or in a broadcast context).

Level 2, *lounge mixing*, is beat-synchronous mixing with adaptation of the speed of the tracks and possibly additional EQ fades, while playing the tracks mostly unchanged.

Level 3, *performative mixing*, is using the DJ deck as a performance instrument by creative use of effects, loops, and mashups with other tracks.

This paper addresses the level 1 and 2 cases, while level 3 can blur the identifiability of the source tracks.

2 Related Work

There is much more existing work in the field of *studio mixing* where a stereo track is produced from individual multi-track recordings and software instruments by means of a mixing desk or DAW [4,15,16,18]. They produced ground truth databases [6] and crowd-sourced knowledge generation [7] with some overlap with DJ mixing. However, when seeing the latter as the mixing of only two source tracks, the studied parameters and influencing factors differ too much from what is needed for DJ mixing. There is quite some existing work on methods to help DJs to produce mixes [1,3,5,9,12,14,17], but much less regarding information retrieval from recorded mixes, with the exception of content-based analysis of playlist choices [13], track boundaries estimation in mixes [10,20], and the identification of the tracks within the mix by fingerprinting [24]. To this end, Sonnleitner et. al. provide an open dataset[3] of 10 dance music mixes with a total duration of 11 hours and 23 minutes made of 118 source tracks. The included playlists contain hand-annotated time points with relevant information for fingerprinting, namely the approximate instant when the next track is present in the mix. Unfortunately, this information is not accurate enough for estimating the start point of the track in the mix. As a result, it cannot be used for our aims of DJ mix analysis and let alone reverse engineering.

Barchiesi and Reiss [2] first used the term *mix reverse engineering* (in the context of multi-track studio mixing) for their method to invert linear processing (gains and delays, including short FIR filters typical for EQ) and some dynamic

[3] http://www.cp.jku.at/datasets/fingerprinting

processing parameters (compression). Ramona and Richard [19] tackle the unmixing problem for radio broadcast mixes, i.e. retrieving the fader positions of the mixing desk for several known input signals (music tracks, jingles, reports), and one unknown source (the host and guests' microphones in the broadcast studio). They model the fader curves as a sigmoid function and assume no time-varying filters, and no speed change of the sources that is only correct in the context of radio broadcast. These two latter references both assume having sample-aligned source signals at their disposal, with no time-scaling applied, unlike our use-case where each source track only covers part of the mix, can appear only partially, and can be time-scaled for beat-matched mixing. There is rare work related to analysis [8] and inversion of non-linear processing applied to the signal such as dynamic-range compression [11] which remains challenging and full of interest for unmixing and source separation.

Hence, this work realizes our idea first presented in [21], by applying it to a large dataset of generated DJ mixes [22]. It already inspired work on a variant of our unmixing method based on convex optimization, and a hand-crafted database [26].

3 DJ Mix Reverse Engineering

The input of our method is the result of the previous stage of identification and retrieval on existing DJ mixes or specially contrived databases for the study of DJ practices. We assume a recorded DJ mix, a playlist (the list of tracks played in the correct order), and the audio files of the original tracks. Our method proceeds in five steps, from a rough alignment of the concatenated tracks with the mix by DTW (section 3.1), that is refined to close in to sample precision (section 3.2), then verified by subtracting the track out of the mix (section 3.3), to the estimation of gain curves (section 3.4) and cue regions (section 3.5).

3.1 Step 1: Rough Alignment

Rough alignment uses the Mel Frequency Cepstral Coefficients (MFCC) of the mix $X(k, c)$ (k being the mix frame index and $c \in \{1, 2, ..., 13\}$ the Mel frequency index) and the concatenated ones of the I tracks $S(l, c) = (S_1 \dots S_I)$ as input (window size 0.05 s, hop size 0.0125 s), l being the frame index of the concatenated matrix S. The motivation is that the MFCC representation is more robust in practice than discrete Fourier-based representation against possible pitch changes from time-scaling of the source tracks in the DJ mix. Since the tracks are almost unchanged in level 2 mixes, Dynamic Time Warping (DTW) [25] can latch on to large valleys of low distance, although the fade regions in the mix are dissimilar to either track, and occur separately in $S(l, c)$. To ease catching up with the shorter time of the mix, we provide a neighborhood allowing the estimated alignment path to perform larger vertical and horizontal jumps, shown in Fig. 1 (right).

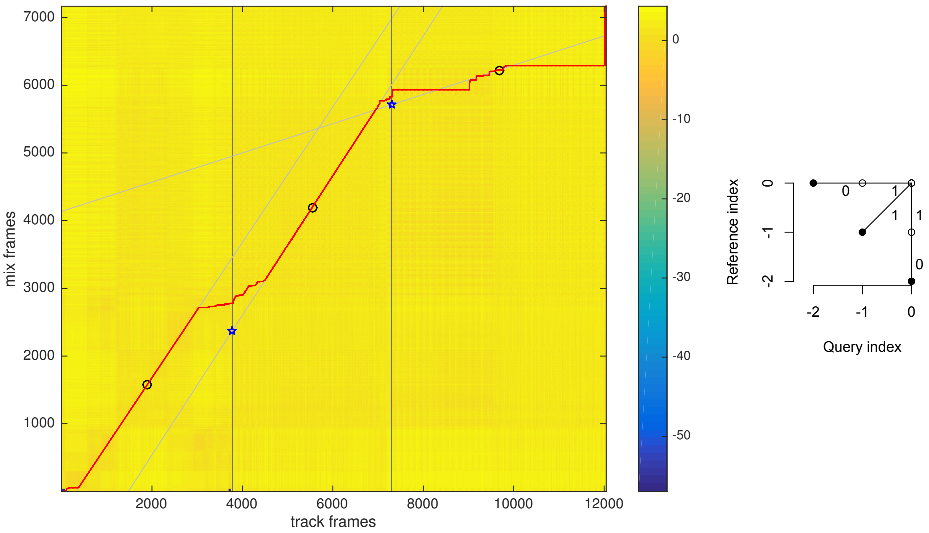

Fig. 1: Left: DTW distance matrix, alignment path (red), track boundaries (vertical lines), found slope lines anchored on track mid-point (circles), and estimated track start (blue marks) on an artificial DJ mix of 3 tracks from our dataset. Right: extended DTW neighbourhood.

The DTW alignment path not only gives us the relative positioning of the tracks in the mix, but also their possible speed change, applied by the DJ to achieve beat-synchronous mixing, see Fig. 1 (left): First, we estimate the speed factor, assuming that it is constant for each track, by calculating the mean slope of the alignment path in a window of half the track length, centred around the middle of the track. Then, the intersections of the slope lines with the track boundaries in $S(l, c)$ provide an estimate of the frame start of the tracks in the mix. The start position expresses the offset of the start of the full source track with respect to the mix, and not the point from where the track is present in the mix. Since the source tracks are mixed with non-zero volume only between the cue-in and cue-out regions, the track start point can be negative.

3.2 Step 2: Sample Alignment

Given the rough alignment and the speed estimation provided by DTW, we then search for the best sample alignment of the source tracks. To this end, we first time-scale the source track's signal according to the estimated speed factor. We then shift a window of the size of an MFCC frame, taken from the middle of the time-scaled track, around its predicted rough frame position in the mix. The best time shift is simply provided by the maximum of the cross-correlation computed between the mix and the track. Please note that this process is not directly applied during the step 1 due to the high computational cost. The sample alignment considers a maximal delay equal to the size of a window and can be computed in a reasonable time.

3.3 Step 3: Track Removal

The success of the sample alignment can be verified by subtracting the aligned and time-scaled track signal from the mix for which a resulting drop in the root-mean-square (RMS) energy is expected. This method remains valid when the ground truth is unknown or inexact. Fig. 2 illustrates the result of track removal applied on a mix in our dataset. We can observe that the resulting instantaneous RMS energy of the mix (computed on the size of a sliding window) shows a drop of about 10 dB. A short increase is also observed during the fades where the suppression gradually takes effect.

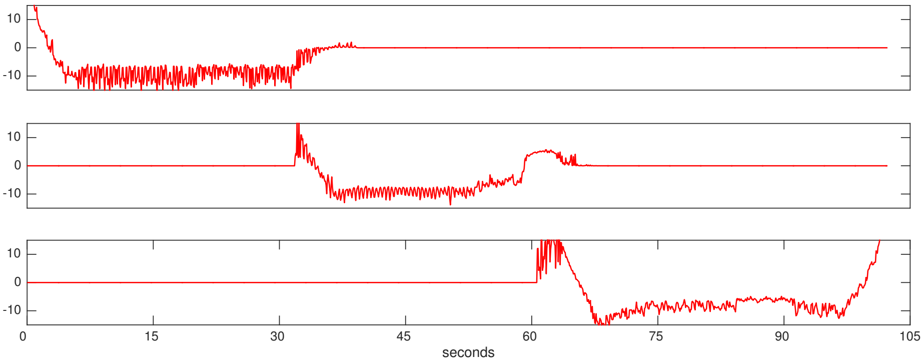

Fig. 2: Resulting RMS energy (in dB) after the subtraction of each track (3) from a mix including fades. Each source track signal is filled with zeros to obtain the same duration.

3.4 Step 4: Volume Curve Estimation

We introduce a novel method based on the time-frequency representation of the signal to estimate the volume curve applied to each track to obtain the mix. Given the discrete-time mix signal denoted $x(n)$ and the constituent sample-aligned and time-scaled tracks $s_i(n)$, we aim at estimating the mixing function $a_i(n)$ as:

$$x(n) = \sum_{i=1}^{I} a_i(n)s_i(n) + b(n) \quad , \forall n \in \mathbb{Z} \tag{1}$$

where $b(n)$ corresponds to an additive noise signal.

From a “correctly” aligned track s_i, its corresponding volume curve $\hat{a}_i$ is estimated using the following steps:

1. we compute the short-time Fourier transforms (STFT) of x and s_i denoted $S_i(n, m)$ and $X(n, m)$ (n and m being respectively the time and frequency indices)

2. we estimate the volume curve at each instant n by computing the median of the mix/track ratio computed along the frequencies $m' \in \mathbb{M}$, where $\mathbb{M}$ is the set of frequency indices where $|S_i(n,m')|^2 > 0$, such as:

$$\hat{a}_i(n) = \begin{cases} \text{median}\left(\frac{|X(n,m')|}{|S_i(n,m')|}\right)_{\forall m' \in \mathbb{M}} & \text{if } \exists m' \text{ s. t. } |S_i(n,m')|^2 > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

3. we optionally post-process $\hat{a}_i(n)$ to obtain a smooth curve by removing outliers using a second median filter for which a kernel size equal to 20 provides good results in practice.

The resulting volume curve can then be used to estimate the cue points (the time instants when a fading effect begins or stops) at the next step. An illustration of the resulting process is presented in Fig. 3.

3.5 Step 5: Cue Point Estimation

In order to estimate the DJ cue points, we apply a linear regression of $\hat{a}_i$ at the time instants located at the beginning and at the end of the resulting volume curve (when $\hat{a}_i(n) < \Gamma$, Γ being a threshold defined arbitrarily as $\Gamma = 0.7 \max(\hat{a})$). Assuming that a linear fading effect was applied, the cue points can easily be deduced from the two affine equations resulting from the linear regression. The four estimated cue points correspond respectively to:

1. n_1, the time instant when the fade-in curve is equal to 0
2. n_2, the time instant when the fade-in curve is equal to $\max(\hat{a}_i)$
3. n_3, the time instant when the fade-out curve is equal to $\max(\hat{a}_i)$
4. n_4, the time instant when the fade-out curve is equal to 0.

In order to illustrate the efficiency of the entire method (steps 4 and 5), we present in Fig. 3 the results obtained on a real-world DJ-mix extracted from our proposed dataset.

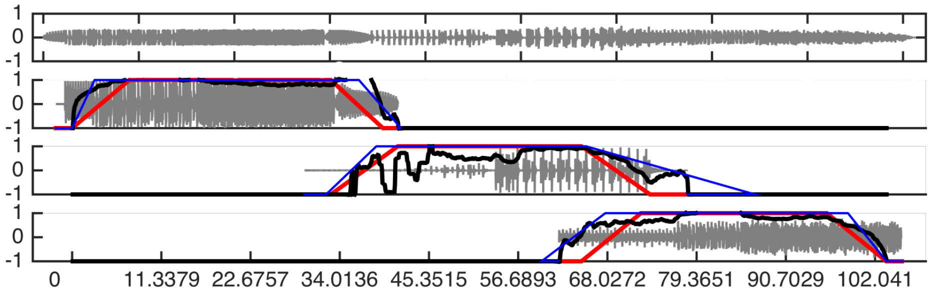

Fig. 3: Estimated volume curve (black), linear fades (blue), ground truth fades (red)

4 The *UnmixDB* Dataset

In order to evaluate the DJ mix analysis and reverse engineering methods described above, we created a dataset containing excerpts of open licensed dance tracks and their corresponding automatically generated mixes [22], available at https://zenodo.org/record/1422385. We use track excerpts of c.a. 40 seconds due to the high runtime and memory requirements, especially for the DTW that is of quadratic memory complexity.

Each mix is based on a playlist made of 3 track excerpts such that the middle track is embedded in a realistic context of beat-aligned linear cross fading to the other tracks. The first track's BPM is used as the seed tempo onto which the other tracks are adapted.

Each playlist of 3 tracks is mixed 12 times with combinations of 4 variants of effects and 3 variants of time scaling using the treatments of the *sox* open source command-line program.The 4 effects are:

none: no effect
bass: +6 dB bass boost using a low-shelving biquad filter below 100 Hz
compressor: heavy dynamics compression (ratio of 3:1 above -60 dB, -5 dB makeup gain)
distortion: heavy saturation with +20 dB gain

These effects were chosen to cover treatments likely to be applied to a DJ set (EQ, compression), and also to introduce non-linear treatments (distortion) to test the limits of re-engineering and unmixing methods.

The 3 timescale methods are:

none: no time scaling, ie. the tracks are only aligned on the first beat in the cue region and then drift apart
resample: linked time and pitch scaling by resampling (sox *speed* effect)
stretch: time stretching while keeping the pitch (sox *tempo* effect using WSOLA)

These 3 variants allow to test simple alignment methods not taking into account time scaling, and allow to evaluate the influence of different algorithms and implementations of time scaling.

The *UnmixDB* dataset contains the complete ground truth for the source tracks and mixes. For each mix, the start, end, and cue points of the constituent tracks are given with their BPM and speed factors. Additionally, the song excerpts are accompanied by their cue region and tempo information.

Table 1 shows the size and basic statistics of the dataset. We also publish the Python source code to generate the mixes, such that other researchers can create test data from other track collections or other variants.

Our DJ mix dataset is based on the curatorial work of Sonnleitner et. al. [24], who collected Creative-Commons licensed source tracks of 10 free dance music mixes from the *Mixotic* net label. We used their collected tracks to produce our track excerpts, but regenerated artificial mixes with perfectly accurate ground truth.

Number of tracks	37
Number of playlists	37
Number of tracks per playlist	3
Number of variants per playlist	12
Number of mixes	444
Average duration of tracks [s]	46
Average duration of mixes [s]	107
Total duration of tracks [min]	1016
Total duration of mixes [min]	2743
Median tempo of tracks [bpm]	128
Minimum tempo of tracks [bpm]	67
Maximum tempo of tracks [bpm]	140

Table 1: Basic statistics of the *UnmixDB* dataset.

5 Evaluation

We applied the DJ mix reverse engineering method on our *UnmixDB* collection of mixes and compared the results to the ground truth annotations. To evaluate the success of our method we defined the following error metrics:

frame error: absolute error in seconds between the frame start time found by the DTW rough alignment (step 1, section 3.1) and the ground truth (virtual) track start time relative to the mix

sample error: absolute error in seconds between the track start time found by the sample alignment (step 2, section 3.2) and the ground truth track start time relative to the mix

speed ratio: ratio between the speed estimated by DTW alignment (step 1, section 3.1) and the ground truth speed factor (ideal value is 1)

suppression ratio: ratio of time where more than 15 dB of signal energy could be removed by subtracting the aligned track from the mix, relative to the time where the track is fully present in the mix, i.e. between fade-in end and fade-out start (step 3, section 3.3, bigger is better)

fade error: the total difference between the estimated fade curves (steps 4 and 5, sections 3.4 and 3.5) and the ground truth fades. This can be seen as the surface between the 2 linear curves over their maximum time extent. The value has been expressed in dB s, i.e. for one second of maximal difference (one curve full on, the other curve silent), the difference would be 96 dB.

Figures 4–10 show the quartile statistics of these metrics, broken down by the 12 mix variants (all combinations of the 3 time-scaling methods and 4 mix effects). The sample alignment results given in Fig. 6 and Table 2 show that the ground truth labels can be retrieved with high accuracy: the median error is 25 milliseconds, except for the mixes with distortion applied, where it is around 100 ms. These errors can already be traced back to the rough alignment (section 3.1):

Fig. 4 shows that it is not robust to heavy non-linear distortion, presumably because the spectral shape changes too much to be matchable via MFCC distances. This error percolates to the speed estimation (Fig. 5), and sample alignment.

The track removal time results in Fig. 8 show sensitivity to the bass and distortion effect (because both of these introduce a strong additional signal component in the mix that is left as a residual when subtracting a track), and also perform less well for time-scaled mixes.

The fade curve volume error in Fig. 10 shows a median of around 5 dB s, which corresponds to a very good average dB distance of 0.3 dB, considering that the fades typically last for 16 seconds.

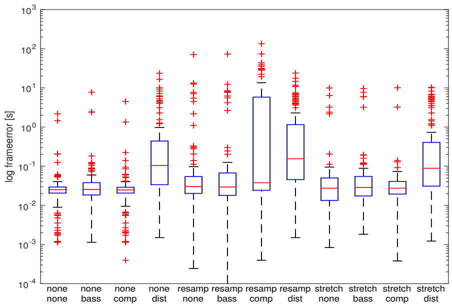

Fig. 4: Box plot of absolute error in track start time found by DTW per variant.

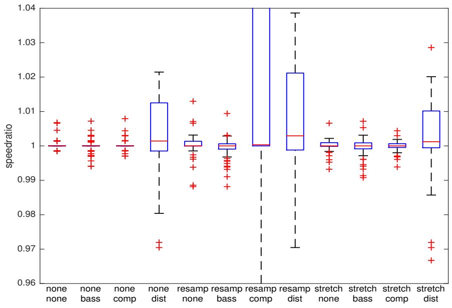

Fig. 5: Box plot of ratio between estimated and ground truth speed per variant.

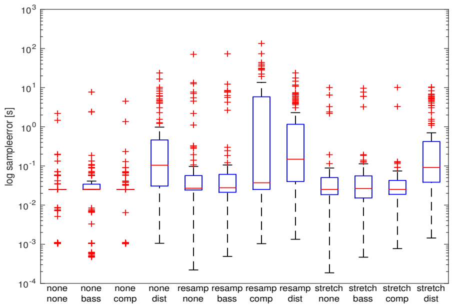

Fig. 6: Box plot of absolute error in track start time found by sample alignment per variant.

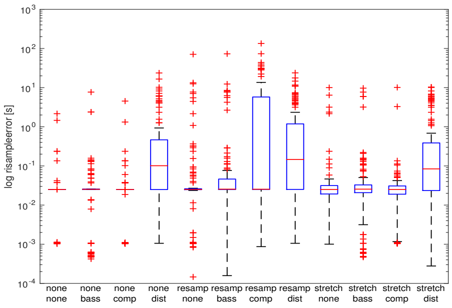

Fig. 7: Box plot of absolute error in track start time found by sample alignment when re-injecting ground truth speed.

While developing our method, we noticed the high sensitivity of the sample alignment and subsequent track removal (steps 2 and 3, sections 3.2 and 3.3) on the accuracy of the speed estimation. This is due to the resampling of the source track to match the track in the mix prior to track removal. Even an estimation error of a tenth of a percent results in desynchronisation after some time.

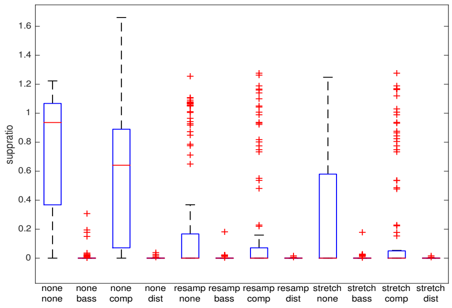

Fig. 8: Box plot of ratio of removal time (bigger is better) per variant.

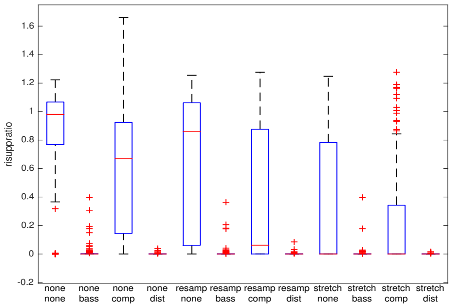

Fig. 9: Box plot of ratio of removal time when re-injecting ground truth speed (bigger is better) per variant.

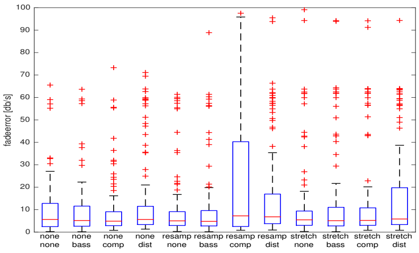

Fig. 10: Box plot of volume difference of fades per variant per mix variant.

To judge the influence of this accuracy, we produced a second set of the *sample error* and *suppression ratio* metrics based on a run of steps 2 and 3 with the ground truth speed re-injected into the processing. The rationale is that the speed estimation method could be improved in future work, if the resulting reductions of error metrics are worthwhile. Also note that the tempo estimation is inherently inaccurate due to it being based on DTW's discretization into MFCC

	mean	**std**	**min**	**median**	**max**
none none	0.0604	0.2469	0.0010	0.0251	2.1876
none bass	0.1431	0.7929	0.0005	0.0254	7.7191
none compressor	0.0806	0.4424	0.0010	0.0251	4.4995
none distortion	1.3376	3.3627	0.0011	0.1042	23.7610
resample none	1.1671	7.0025	0.0002	0.0270	71.0080
resample bass	1.3337	7.2079	0.0005	0.0277	73.1192
resample compressor	6.8024	17.0154	0.0010	0.0372	134.2811
resample distortion	1.8371	3.8551	0.0013	0.1483	23.8355
stretch none	0.2502	1.1926	0.0002	0.0251	10.0048
stretch bass	0.3300	1.4249	0.0005	0.0264	9.6626
stretch compressor	0.1520	1.0025	0.0008	0.0251	10.1076
stretch distortion	1.0629	2.2129	0.0014	0.0911	10.3353
all	1.2131	6.2028	0.0002	0.0282	134.2811

Table 2: Statistics of absolute error in track start time found by sample alignment.

frames. In mixes with full tracks, the slope can be estimated more accurately than with our track excerpts simply because more frames are available.

Figures 7 and 9 show the quartile statistics of the sample error and suppression ratio with re-injected ground truth speed. We can see how most variants are improved in error spread for the former, and 4 variants are greatly improved for the latter, confirming the sensitivity of the track removal step 3 on the speed estimation.

6 Conclusions and Future Work

The presented work is a first step towards providing the missing link in a chain of methods that allow the retrieval of rich data from existing DJ mixes and their source tracks. An important result is the validation using track removal in section 3.3 to compute a new metric for the accuracy of sample alignment. This metric can also be computed even without ground truth. A massive amount of training data extracted from the vast number of collections of existing mixes could thus be made amenable to research in DJ practices, cultural studies, and automatic mixing methods. With some refinements, our method could become robust and precise enough to allow the inversion of fading, EQ and other processing [2, 19]. First, the obtained tempo slope could be refined by searching for sample alignment at several points in one source track. This would also extend the applicability of our method to mixes with non-constant tempo curves. Second, a sub-sample search for the best alignment should achieve the neutralisation of phase shifts incurred in the mix production chain. We could also check whether a DTW with relaxed endpoint condition [23] for the beginning and end of a mix could be advantageous. Furthermore, the close link between alignment, time-scaling, and unmixing hints at the possibility of a joint and possibly iterative estimation algorithm, maximising the match in the three search spaces simultaneously.

References

1. Felipe Aspillaga, J. Cobb, and C-H Chuan. Mixme: A recommendation system for DJs. In *ISMIR*, October 2011.
2. Daniele Barchiesi and Joshua Reiss. Reverse engineering of a mix. *Journal of the Audio Engineering Society*, 58(7/8):563–576, 2010.
3. Rachel M. Bittner, Minwei Gu, Gandalf Hernandez, Eric J. Humphrey, Tristan Jehan, Hunter McCurry, and Nicola Montecchio. Automatic playlist sequencing and transitions. In *ISMIR*, October 2017.
4. Brett Brecht De Man, R; King, and J. D. Reiss. An analysis and evaluation of audio features for multitrack music mixtures. In *ISMIR*, 2014.
5. Dave Cliff. Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks. Technical report, Hewlett-Packard Laboratories, 2000. HPL 104.
6. Brecht De Man, Mariano Mora-Mcginity, György Fazekas, and Joshua D Reiss. The open multitrack testbed. In *Audio Engineering Society Convention 137*, 2014.

7. Brecht De Man and Joshua D Reiss. Crowd-sourced learning of music production practices through large-scale perceptual evaluation of mixes. *Innovation in Music II*, page 144, 2016.
8. Dominique Fourer and Geoffroy Peeters. Objective characterization of audio signal quality: Application to music collection description. In *Proc. IEEE ICASSP*, pages 711–715, March 2017.
9. Tsuyoshi Fujio and Hisao Shiizuka. A system of mixing songs for automatic DJ performance using genetic programming. In *6th Asian Design International Conference*, October 2003.
10. Nikolay Glazyrin. Towards automatic content-based separation of DJ mixes into single tracks. In *ISMIR*, pages 149–154, October 2014.
11. Stanislaw Gorlow and Joshua D Reiss. Model-based inversion of dynamic range compression. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1434–1444, 2013.
12. Hiromi Ishizaki, Keiichiro Hoashi, and Yasuhiro Takishima. Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort. In *ISMIR*, pages 135–140, 2009.
13. Thor Kell and George Tzanetakis. Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction. In *ISMIR*, pages 505–510, 2013.
14. Adrian Kim, Soram Park, Jangyeon Park, Jung-Woo Ha, Taegyun Kwon, and Juhan Nam. Automatic DJ mix generation using highlight detection. In *Proc. ISMIR, late-breaking demo paper*, October 2017.
15. Jacob A Maddams, Saoirse Finn, and Joshua D Reiss. An autonomous method for multi-track dynamic range compression. In *DAFx*, 2012.
16. Stuart Mansbridge, Saorise Finn, and Joshua D Reiss. An autonomous system for multitrack stereo pan positioning. In *Audio Engineering Society Convention*, 2012.
17. Pablo Molina, Martín Haro, and Sergi Jordá. Beatjockey: A new tool for enhancing DJ skills. In *NIME*, pages 288–291. Citeseer, 2011.
18. Enrique Perez-Gonzalez and Joshua Reiss. Automatic gain and fader control for live mixing. In *Proc. IEEE WASPAA*, pages 1–4, October 2009.
19. Mathieu Ramona and Gaël Richard. A simple and efficient fader estimator for broadcast radio unmixing. In *Proc. Digital Audio Effects (DAFx)*, pages 265–268, September 2011.
20. Tim Scarfe, W Koolen, and Yuri Kalnishkan. Segmentation of electronic dance music. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, 22(3):4, 2014.
21. Diemo Schwarz and Dominique Fourer. Towards Extraction of Ground Truth Data from DJ Mixes. In *Late-break Session of ISMIR*, Suzhou, China, October 2017.
22. Diemo Schwarz and Dominique Fourer. Unmixdb: A dataset for DJ-mix information retrieval. In *Late-break Session of ISMIR*, Paris, France, September 2018.
23. Diego Furtado Silva, Gustavo Enrique de Almeida Prado Alves Batista, Eamonn Keogh, et al. On the effect of endpoints on dynamic time warping. In *SIGKDD Workshop on Mining and Learning from Time Series, II.* Association for Computing Machinery-ACM, 2016.
24. Reinhard Sonnleitner, Andreas Arzt, and Gerhard Widmer. Landmark-based audio fingerprinting for DJ mix monitoring. In *ISMIR*, New York, NY, 2016.
25. Robert J Turetsky and Daniel PW Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. In *ISMIR*, October 2003.
26. Lorin Werthen-Brabants. Ground truth extraction & transition analysis of DJ mixes. Master's thesis, Ghent University, Belgium, 2018.