A Masterpiece of All Music: A Machine Learning Approach to Automatic DJ

Yifan He
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
yifanhe@andrew.cmu.edu

Yanqiao Wang
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
yanqiaow@andrew.cmu.edu

Tiancheng Zheng
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
tzheng2@andrew.cmu.edu

Yuchen Wu
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
yuchenwu@andrew.cmu.edu

Abstract

A disk jockey (DJ) is an important and skilled job as a DJ can combine different pieces of music into a continuous and harmonious mix, which is especially important during a music festival. However, a human DJ is often expensive. Experimenting on a new dataset for electronic dance music (EDM) featured in the YouTube video by Palms Trax, we designed an automated DJ system using existing methods in the literature such as beat tracking, downbeat tracking, segmentation, key extraction, voice detection, track selection, cue point selection, and cross-fading. These methods have been proven to work on drum and bass music by Len Vande Veire and Tijl De Bie from Ghent University. Building on top of their codebase, we generalized the machine learning model to general EDM music and achieved good results during user testing.
Our code is available here on GitHub. Our data can be accessed using CMU Google accounts here.

1 Introduction and Problem Statement

In many music software we use, changing songs due to different song styles will cause an interruption in the listening experience. Many songs also have a long intro and outro, which can lessen the audience's excitement, especially in a music festival or dance club [1] [2] [3]. Therefore, it is common to have a disk jockey, also known as a DJ, produce a continuous mix of music on stage during a music festival. However, a DJ is a skilled job that requires both an understanding of music and expertise in using specialized equipment.

Given this limitation, our team aims to implement a fully automatic DJ in this project that can generate seamless music mixes harmoniously given a collection of recorded music. The automatic DJ will offer a better listening experience for the audience while also having a much lower cost than a human DJ.

2 Literature Review

Past studies are primarily focused on two types of systems regarding solving the problem of discontinuous transfer from music to music. The first system is automatic DJ which focuses on automating the DJing task [4], [5], and the other is a mash-up system that focuses on creating a new song by combining fragments of existing songs [6], [7].

In the automatic DJ systems, past research studies [8] applied several approaches to blur the gap between music, such as using cross-fading to mix songs or avoiding overlapping vocal sections among different songs using vocal detection [9]. Research on mash-up systems typically focuses on devising a measure of musical compatibility of song extracts. Music fragments are extracted based on their harmonic and rhythmic similarities.

However, most of the past research studies only focus on optimizing the work by minimizing the amount of discomfort, and very few systems consider the high-level structural properties of the mixed audio. In addition, the focus is usually on optimizing individual cross-fades, but the global song progression throughout the mix is generally not considered.

3 Datasets

We originally plan to use the 1001Tracklists dataset from Kim et al. [10]. However, the proposed dataset is hard to obtain since it is private and cannot be found online. Therefore, we found an alternative tracklist on the authors' website [11] that features the YouTube video by artist Palms Trax [12] and downloaded all the songs in this mix using the "youtube_dl" library as our final dataset.

Figure 1 shows the DTW-based mix-to-track sub-sequence alignment between the mix and songs that are played in the mix. The x-axis indicates the mix's beat frame, and the y-axis is the song's beat frame. The colored solid lines are the warping paths of the alignment with different methods, while the black dot lines are the ground truth for each individual song. If a mix and a track are considered successfully aligned, the colored solid line will become diagonal and the color bar at the bottom will also be added to indicate the success of the alignment.

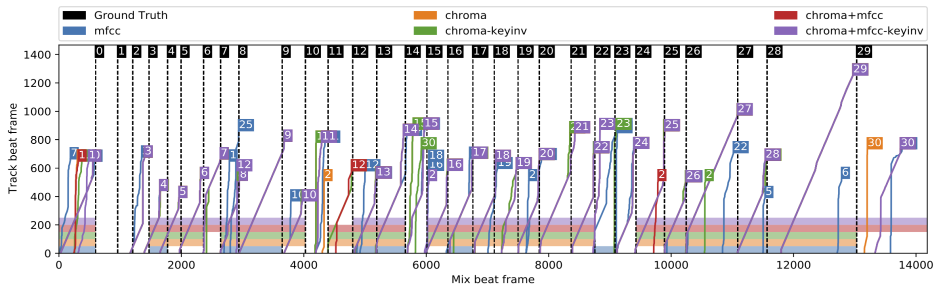


Figure 1: Mix-to-Track Alignment

4 Methods

In order to generate DJ mixes from the given library of songs, we generally have two steps. First, we analyze each song and annotate the information we need. Next, we connect songs in proper order and effect. In this section, we would explain the methods we used around these two main steps. During analysis, we considered five features: beat, downbeat, segmentation, key, and singing. Every song in the given dataset would be annotated by these five features. During our mix, we would consider Track selection and Cue point selection. Finally, we create the cross-fade between songs.

4.1 Song Analysis

We first implemented beat tracking. We used the algorithm inspired by the work of Davies and Plumbley [13] to locate the beat positions. It assumes the tempo is steady, considering two parameters to define the positions - the beat period τ or $v = 60\tau$, and beat phase Φ. The positions of musical onsets are estimated by means of an *onset detection function*(ODF), which has a high value for time instants in the music where onset is detected, and a low value elsewhere. The detection function $\Gamma(m)$, at frame m, is calculated by measuring the sum of the Euclidean distance between the observed spectrum at frame $S_k(m)$ and a prediction of the spectrum $\hat{S}_k(m)$, for all frequency bins k.

$$\Gamma(m) = \sum_{k=1}^{K} |S_k(m) - \hat{S}_k(m)|^2. \tag{1}$$

To achieve an auto-correlation function with clear, well-defined peaks, we first process the detection function to emphasize the strongest and discard the least significant peaks. We calculate an adaptive, moving mean threshold Γ_i.

$$\bar{\Gamma}_i(m) = \text{mean}\{\Gamma_i(q)\} \quad m - \frac{Q}{2} \leq q \leq m + \frac{Q}{2} \tag{2}$$

We set all negative valued elements to zero to give a modified detection function frame:

$$\tilde{\Gamma}_{\text{HWR}}(m) = \text{HWR}(\Gamma_i(m) - \bar{\Gamma}_i(m)) \tag{3}$$

where Half-Wave Rectify operation $\text{HWR}(x) = \frac{x+|x|}{2}$. Then the beat period $\hat{\tau}$ can be calculated as:

$$\hat{\tau} = \underset{\tau}{\text{argmax}}(\frac{1}{N}\sum_{n}\frac{1}{M}\sum_{m}\Gamma_{\text{HWR}}(m)\Gamma_{\text{HWR}}(nL_{\tau} + m)) \tag{4}$$

We calculate the ODF $\Gamma(m)$ of the audio. The *melflux* ODF is used with a frame size N_F of 1024 samples and hop size N_H of 512 samples. $\mathcal{X}_{\text{mel40}}(m, k)$ being the energy of frame m in the k-th frequency bin, logarithmically spaced according to the Mel scale, and the Half-Wave Rectify operation $\text{HWR}(x) = \max(x, 0)$.

For downbeat tracking, we first extract features from the beat segments. We calculate the loudness of each beat fragment and the energy distribution of the audio along the frequency axis, binned in 12 equally spaced bins on the Mel frequency scale.

$$x_{\text{frame}}^{[N_{\hat{\tau}};N_{\hat{\tau}}]}(m) = x[N_{\hat{\tau}}(m) : N_{\hat{\tau}}(m+1)] \tag{5}$$

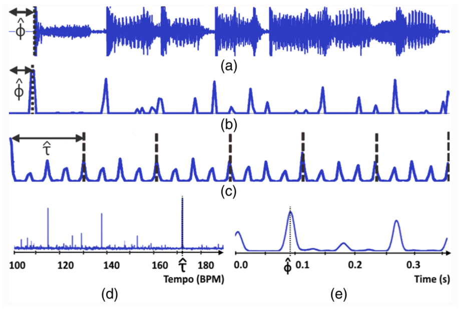
$$\mathcal{X}_{\text{loud}}(m) = \text{loudness}\left(x_{\text{frame}}^{[N_{\hat{\tau}};N_{\hat{\tau}}]}(m)\right) \tag{6}$$

$$\mathcal{X}_{\text{mel}}(m, k) = X_{\text{mel12}}\left(x_{\text{frame}}^{[N_{\hat{\tau}};N_{\hat{\tau}}]}(m)\right)(k) \tag{7}$$

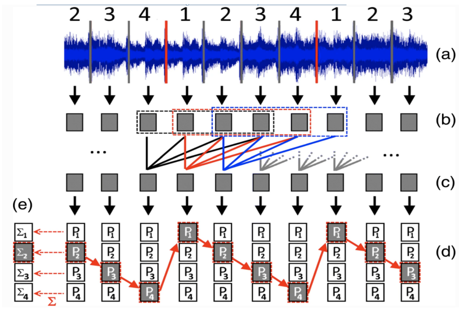
Additionally, three onset detection functions are calculated for the entire song - the flux, the high-frequency coefficient (HFC), and the complex spectral difference (CSD) ODF. Then, a logistic regression classifier, trained on manually annotated songs, determines the probability that a beat is either the first, second, third, or fourth beat in the measure it belongs to. Finally, these predictions are aggregated over the entire song for each of the four options to determine the most likely downbeat positions.

$$\mathcal{X}_{\text{odf}_i}(m, k) = \Gamma_{\text{HWR}}^{(\text{i})}(L_{\text{beat}}^{(m)} + k),\ 0 \leq k < L_{\hat{\tau}}, i \in \{\text{flux}, \text{hfc}, \text{csd}\} \tag{8}$$

The visualization for beat and downbeat tracking is shown in Figure 2 below.



(a) Beat Tracking



(b) Downbeat Tracking

Figure 2: Beat and Downbeat Tracking

For segmentation, we use the novelty-based structural segmentation method by Foote [14], which assumes that structural boundaries are accompanied by significant musical changes. We split the input audio into half-beat frames with a quarter-beat hop, nfft was set to 2048 and took 13 MFCC features. Extracting their Mel-frequency cepstrum, then using cosine distance to calculate their similarity. The adjacent two frames with the least cosine similarities can be considered as the boundaries.

$$d_{\cos}(nu, \vec{v}) = 1 - \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|_2 \|\vec{v}\|_2} \tag{9}$$

$$X_{\text{MFCC}}^{[N_{\hat{\tau}}/2;N_{\hat{\tau}}/4]}(m,k) = X_{\text{MFCC}}\left(x_{\text{frame}}^{[N_{\hat{\tau}}/2;N_{\hat{\tau}}/4]}(m)\right)(k) \tag{10}$$

$$S_{\text{MFCC}}(m,n) = d_{\cos}\left(X_{\text{MFCC}}^{[N_{\hat{\tau}}/2;N_{\hat{\tau}}/4]}(m,\cdot), X_{\text{MFCC}}^{[N_{\hat{\tau}}/2;N_{\hat{\tau}}/4]}(n,\cdot)\right) \tag{11}$$

For the key of each track, we use the algorithm by Faraldo et al. [15], which is implemented by the Essentia music analysis library [16].

When two songs both have singing voices, it is important to avoid mixing them directly. We used an SVM classifier with an RBF kernel to classify segments of one measure long as vocal or instrumental. These fragments are first analyzed in 2048-sample-long frames, with half of the frames overlapping between consecutive frames. MFCCs, spectral contrast, and the corresponding spectral valley features are calculated. Then we calculate the means, variances, and skews of these features. These are the input to the SVM classifier. The training set consists out of 55 manually annotated songs. The classifier parameters were tuned using cross-validation, giving a regularization parameter C of 1.0 and kernel coefficient γ of 0.01. Our system would prioritize mixing an instrumental song for the next if we are currently playing a vocal song.

4.2 Song Mixing

For cue point selection, in order to create a diverse listening experience, we implemented three different transition methods. From the previous structural segmentation, we know that EDM music usually consists of intro, buildup, main section (which is also called drop), bridge, and, outro. Our cue points are randomly selected among three modes: Bridge-to-Drop, Drop-to-Drop, Outro-to-Intro.

(a) For Bridge-to-Drop mode, 48-bars transition happens from the 16 bars from the last bridge until 32 bars of the next drop, which cross-fades the bridge and drop part of both songs.

(b) For Drop-to-Drop mode, 32-bars transition happens from the 16 bars from the previous drop until the 16 bars of the next drop, which cross-fades the previous drop with the bridge and drop of the next song.

(c) For Outro-to-Intro mode, 32-bars transition happens from the 16 bars from the previous drop until the first 16 bars of the next intro, which cross-fades the previous drop and outro with the intro of the next song.

The visualization for our three transition methods is shown in Figure 3 below.

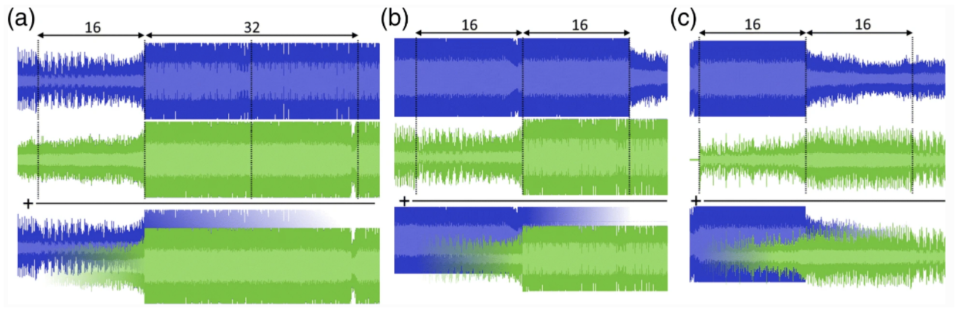


Figure 3: Three Modes of Cue Point Selection

For track selection, tone and rhythmic similarities are in our considerations. Also, songs that are one semitone lower or higher from the current song would be considered to mix for the next song. Time-stretching and up or down-sampling would be used during the mix. The ODF portions corresponding to the overlapped music segments are compared by means of the dynamic time warping (DTW) algorithm. The DTW method stretches the ODFs during comparison and hence eliminates the need to recalculate the ODFs after time-stretching the original audio. The song and cue point combination which leads to the lowest DTW score is selected as optimal next song.

Given the cue point positions and the next song, we need to generate cross-fades in between to mix. Time-stretching is performed by applying harmonic-percussive separation and independently stretching the harmonic respectively using the frequency-domain phase vocoder and the time-domain OLA time scale modification algorithm proposed by Driedger and Müller [17]. After that, volume fading and equalization filters are applied to both the last and next songs. Finally, the audio of the songs is added together in a beat-matched way, effectively mixing the two songs together.

5 Evaluation Metrics

At first, we plan to use the Jensen–Shannon divergence (JSD) to calculate the similarity between the adjacent music, the smaller the JSD, the better the mix, so it seems to be suitable.

$$JSD(p;q) = h\left((p+q)/2\right) - (h(p) + h(q))/2 \tag{12}$$

where $h(\cdot)$ is the Shannon information, and p and q are two probability distributions. A smaller JSD indicates a larger similarity between p and q.

However, we later found that this evaluation method was not suitable, since the time it needs to mix different songs is not the same, we don't know how much time each piece actually takes, so there is no way to calculate the JSD.

Since it is hard to test against ground truth data for the generated music mix, we decided to conduct a user test which is conducted in a form of single-blind. We also developed a baseline automated DJ mix which only applied cue point selection and cross-fades on it, without any analysis information we annotated at first, so it basically is a random mixing list.

We provided the listeners with three mixes in random order, and let them score 1-5 for the first three questions and evaluate whether they think the mix is generated by the automated DJ or the human DJ at the end of the test. The questionnaire was conducted among 40 people to compare the mix by the automated DJ, baseline mix, and human DJ mix. Their answers are recorded and summarized in the table in the "Results and Analysis" section.

6 Results and Analysis

The scores of each question corresponding to different types of the mix are calculated as mean and recorded in the first three lines of the table. The percentage on the last line indicates how many percent of people think the mix they are hearing is created by an automated DJ or a human DJ. Since the participants don't know how many songs are produced by human DJ or automated DJ, the sum of percentages are not equal to 100%.

Question 1	How do you think about the rhythm?
Question 2	How do you think about the tone?
Question 3	Do you feel any discord in the mix?
Question 4	Is the mix you are listening produced by machine or human?

Table 1: Survey Questions

Question \ Mix Types	Automated DJ Mix	Baseline Mix	Human DJ Mix
Question 1	3.93	1.34	4.79
Question 2	4.02	1.52	4.85
Question 3	3.56	1.08	4.71
Question 4 (% think it is machine)	67.5%	100%	7.5%

Table 2: User-Testing Results

As we can see, the automated DJ received good feedback from our audience from the first three questions. Only a few people did not recognize the mix made by a human, and 32.5% thought the mix made by an automated DJ was composed by a human. People who recognized the difference between the automated DJ and the human DJ suggested that some transitions in the automated mix sounded noisy or discordant compared to the human mix.

7 Future Work

For our future work, first, we can choose some new methods to use since some of our implemented methods are a little bit outdated. For example, our beat tracking method is a traditional method, and nowadays some methods such as CRNN, BiLSTM, or TCN may give us better results. Second, for evaluation, some efforts will be needed to generate ground truth that could be used in the quantitative comparison with our experiment results. Our evaluation is more of a qualitative one. For example, we might be able to use our initial JSD metric in some experiment settings. Therefore, in future studies, we would like to find a method to better evaluate the similarity. For the music part, both our dataset and methods are based on the assumption that that the music is in 4/4 measure, so we did not consider more complicated beat patterns or the double/half tempo issues. For future work, we might take these ideas into consideration.

References

[1] K. R. Scherer, “Which emotions can be induced by music? what are the underlying mechanisms? and how can we measure them?” *Journal of new music research*, vol. 33, no. 3, pp. 239–251, 2004.

[2] N. Sunderland, L. Istvandity, A. Lakhani, C. Lenette, B. Procopis, and P. Caballero, “They [do more than] interrupt us from sadness: Exploring the impact of participatory music making on social determinants of health and wellbeing for refugees in australia,” *Health, Culture and Society*, vol. 8, no. 1, pp. 1–19, 2015.

[3] R. Reber, *Critical Feeling*. Cambridge University Press, 2016.

[4] A. Kim, S. Park, J. Park, J.-W. Ha, T. Kwon, and J. Nam, “Automatic dj mix generation using highlight detection,” Oct. 2017.

[5] P. Molina, G. Haro, and S. Jordà, “Beatjockey: A new tool for enhancing dj skills,” in *New Interfaces for Musical Expression (NIME)*, Oslo, Norway, May 2011, pp. 288–291. [Online]. Available: `files/publications/G20-Molina.pdf`.

[6] M. E. P. 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. DOI: `10.1109/TASLP.2014.2347135`.

[7] A. Eigenfeldt, “Generating electronica: A virtual producer and virtual dj,” ser. C&C ’13, Sydney, Australia: Association for Computing Machinery, 2013, p. 396, ISBN: 9781450321501. DOI: `10.1145/2466627.2481234`. [Online]. Available: `https://doi.org/10.1145/2466627.2481234`.

[8] N. Martelaro and W. Ju, “Dj bot: Needfinding machines for improved music recommendations,” in *2017 AAAI Spring Symposium Series*, 2017.

[9] D. Bouckenhove and J.-P. p. (Martens, *Automatisch mixen van muzieknummers op basis van tempo, zang, energie en akkoordinformatie*, dut, 2007. [Online]. Available: `http://lib.ugent.be/catalog/rug01:001312437`.

[10] T. Kim, M. Choi, E. Sacks, Y.-H. Yang, and J. Nam, *A computational analysis of real-world dj mixes using mix-to-track subsequence alignment*, 2020. arXiv: `2008.10267 [eess.AS]`.

[11] T. Kim. “Summary | a computational analysis of real-world dj mixes using mix-to-track subsequence alignment,” GitHub. (Oct. 2020), [Online]. Available: `https://mir-aidj.github.io/djmix-analysis/`.

[12] P. Trax. “Palms trax | boiler room: Streaming from isolation,” YouTube. (Apr. 2020), [Online]. Available: `https://www.youtube.com/watch?v=cPo-qzbGLqE`.

[13] M. E. P. Davies and M. D. Plumbley, “Context-dependent beat tracking of musical audio,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007. DOI: `10.1109/TASL.2006.885257`.

[14] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 1, 2000, 452–455 vol.1. DOI: `10.1109/ICME.2000.869637`.

[15] Á. Faraldo, E. Gómez, S. Jordà, and P. Herrera, “Key estimation in electronic dance music,” *Lecture Notes in Computer Science Advances in Information Retrieval*, pp. 335–347, 2016. DOI: `10.1007/978-3-319-30671-1_25`.

[16] D. Bogdanov, N. Wack, E. Gómez, *et al.*, “Essentia: An open-source library for sound and music analysis,” in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM ’13, Barcelona, Spain: Association for Computing Machinery, 2013, pp. 855–858, ISBN: 9781450324045. DOI: `10.1145/2502081.2502229`. [Online]. Available: `https://doi.org/10.1145/2502081.2502229`.

[17] J. Driedger, M. Müller, and S. Ewert, “Improving time-scale modification of music signals using harmonic-percussive separation,” *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 105–109, 2013.