Audio Engineering Society

Conference Paper 5

Presented at the AES International Symposium on AI and the Musician
2024 June 6–8, Boston, MA, USA

This conference paper was selected based on a submitted abstract and 750-word precis that have been peer reviewed by at least two qualified anonymous reviewers. The complete manuscript was not peer reviewed. This conference paper has been reproduced from the author's advance manuscript without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Deep Learning-based Audio Representations for the Analysis and Visualisation of Electronic Dance Music DJ Mixes

Alexander Williams[1*], Haokun Tian[1*], Stefan Lattner[2], Mathieu Barthet[1], and Charalampos Saitis[1]

[1] *Queen Mary University of London, London, United Kingdom*
[2] *Sony CSL, Paris, France*
[*] *Equal contribution*

Correspondence should be addressed to Alexander Williams (alexander.j.williams@qmul.ac.uk)

ABSTRACT

Electronic dance music (EDM), produced using computers and electronic instruments, is a collection of musical subgenres that emphasise timbre and rhythm over melody and harmony. It is usually presented through the medium of DJing, where tracks are curated and mixed sequentially to offer unique listening and dancing experiences. However, unlike key and tempo annotations, DJs still rely on audition rather than metadata to examine and select tracks with complementary audio content. In this work, we investigate the use of deep learning-based representations (Complex Autoencoder and OpenL3) for analysing and visualising audio content on a corpus of DJ mixes with approximate transition timestamps and compare them with signal processing-based representations (joint time-frequency scattering transform and mel-frequency cepstral coefficients). Representations are computed once per second and visualised with UMAP dimensionality reduction. We propose heuristics based on the identification of observed patterns in visualisations and time-sensitive Euclidean distances in the representation space to compute DJ transition lengths, transition smoothness, and inter-song, song-to-song, and full-mix audio content consistency using audio representations along with rough DJ transition timestamps. Our method enables the visualisation of variations within music tracks, facilitating the analysis of DJ mixes and individual EDM tracks. This approach supports musicians in making informed creative decisions based on such visualisations. We share our code, dataset annotations, computed audio representations, and trained CAE model. We encourage researchers and music enthusiasts alike to analyse their own music using our tools: https://github.com/alexjameswilliams/EDMAudioRepresentations.

1 Introduction

Electronic dance music (EDM) is a broad term for various music styles created using computers and electronic instruments, characterised by repetition, variation, and dance-oriented compositions [1, 2]. EDM is commonly presented through the medium of DJing, a musically creative process of sequentially mixing pre-existing au-

dio into an extended and continuous mix of sound for a unique listening and dancing experience [3]. EDM tracks, and subsequently DJ mixes, are typically built around repeating loops of melodies, vocals, drums, and sound effects (FX). These change and are layered over time to produce variation and progression in the composition. Structural changes in EDM tracks are usually indicated by an evolution of timbre and rhythm rather than melody and harmony and involve either an element entering or leaving the mix or being affected by some form of continuous process such as FX and synthesizer parameter automation [4, 5].

Timbre and rhythm play crucial roles in both concurrent and sequential grouping of musical elements. They help us perceive these elements as unified wholes and also inform the boundaries between different sections of music [6, 7]. A previous study found that selecting tracks with a similar timbre is an important factor in the ordering of tracks in a DJ mix [8], alongside key, tempo and track structure [9]. Perceptual studies indicate a link between low-level audio timbre descriptors and rhythm in EDM and how listeners experience the music cognitively, emotionally, and physically [10]. Additionally, listeners agree on the similarities between EDM tracks in terms of timbre and rhythm [4]. Furthermore, the EDM community consistently uses specific terms to describe the invariant timbral qualities of this music [11]. However, unlike key, tempo, and structure, there is no straightforward way to visually represent variations in timbre and rhythm for DJs in contemporary tools for the studio and the stage. Instead, DJs primarily examine track characteristics through listening. Recognising the limitations of current methods, this work proposes novel techniques for visualising EDM tracks, aiming to bridge the gap between how we hear and represent sound characteristics.

2 Related Work

Previous works have developed models for timbre (and rhythm) similarity and structural segmentation of electronic dance music (EDM) based on features such as mel-frequency cepstral coefficients (MFCCs), roughness, and spectral flatness [8, 12, 13, 14] while [9] analysed a large corpus of DJ mixes to generate statistics on key and tempo manipulations, transition lengths and transition point agreement. In terms of visualisation, [15] created a visual thumbnail to summarise and homogeneously convey information to DJs about a track's tempo, volume, "aggressiveness", genre, pitch, and bass presence, and Pioneer DJ software has introduced waveform colouring with content at different frequency bands represented on a colour spectrum [16].

3 Dataset

We have compiled a dataset of 200 recorded DJ mixes commissioned by the long-running London night club *Fabric* from its two parallel mix series titled *fabric* and *FABRICLIVE* that ran between 2001 and 2018 for 100 mixes each. The *fabric* series mainly covers styles such as house, techno, and tech house, while *FABRICLIVE* covers various styles of bass music such as drum and bass, grime, and hip-hop, and is more stylistically diverse. Each entry in the series was commercially released in physical and digital formats. The full audio was split into sequential individual tracks at the approximate track timestamps.

Statistics about the two mix series' history were generated by [17]. Collectively, the series consists of over 4000 individual tracks from over 3000 unique artists. Some mixes contain as many as 65 tracks, while others as few as 12. The majority (60%) of DJs compiling the mix are UK artists, while 16% are from the USA, and 8% are from Germany, with the rest coming from artists of 17 different nationalities around the world. 90% of the mixes are from male DJs and 10% from female DJs, reflecting the historic under-representation of women in EDM culture [18]. The mixes are dominated by the broad electronic genre, but some contain elements of hip hop, funk/soul, rock, reggae, pop, jazz, folk, world, country, and classical music and a diverse range of specific musical styles. The full distribution of styles and genres is shown in Fig 1.

We obtained metadata, tracklists and coarse corresponding track timestamps from the commercial release information, available via Discogs[1], along with the Discogs community-generated genre and style tags. The python3-discogs-client python package[2] was used to obtain the data on 6th May 2024.

Discogs is one of the largest online databases of editorial metadata used by music collectors and enthusiasts. The quality of the crowd-generated data in Discogs is considered to be high because of its strict guidelines, moderation system and a large community of involved

[1] https://www.discogs.com/

[2] https://github.com/joalla/discogs_client/

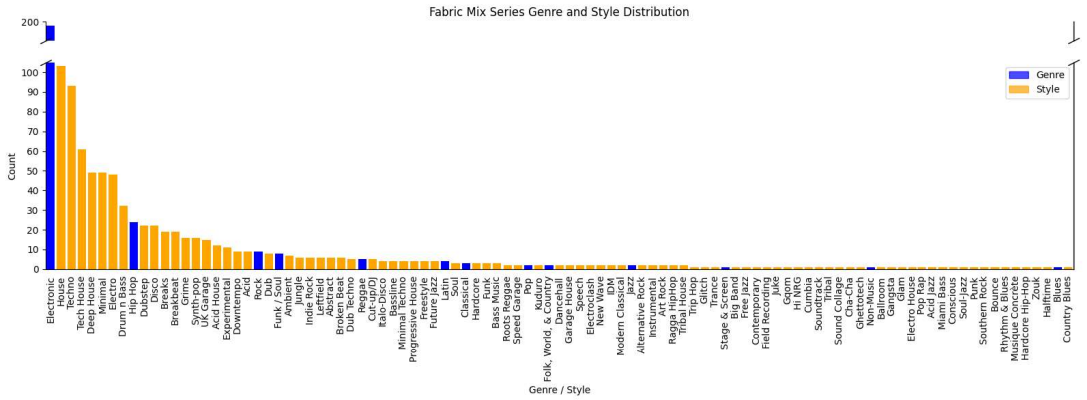


Fig. 1: The distribution of mix-level genre and style tags across the 200 mixes in the *fabric* and *FABRICLIVE* mix series obtained from Discogs in May 2024

enthusiasts [19]. Discogs have already been demonstrated as a useful resource for the analysis of trends in EDM [19, 20], but availability and accuracy of data are dependent on community interest.

Musical genre labelling - particularly in EDM - can be problematic in its own right [21], and any particular mix's overall labels will not necessarily reflect the genre and style of every track in that mix. However, Discogs implements a release-level, non-exclusive multi-label approach with a two-level genre hierarchy consisting of broad genre tags and more specific styles. Multi-label classification is useful for describing EDM due to the large number of (sub-)genres and styles and the nature by which musical crossover and influence occur [1, 20]. Therefore, we hope that the coarse genre and style tags are generally agreeable for describing the overall style of the mix, given their crowd-sourced nature and the popularity of the mix series.

4 Computing Audio Representations

In this study, we explore applying deep learning (DL)-based audio representations to capture comprehensive time-dependent audio features in EDM DJ mixes. We employ general-purpose audio representations capable of encoding timbral and rhythmic nuances, including the Complex Autoencoder [22] and OpenL3 [23] representations, and compare them with signal processing-based representations, including the joint time-frequency scattering transform [24] and MFCCs [25].

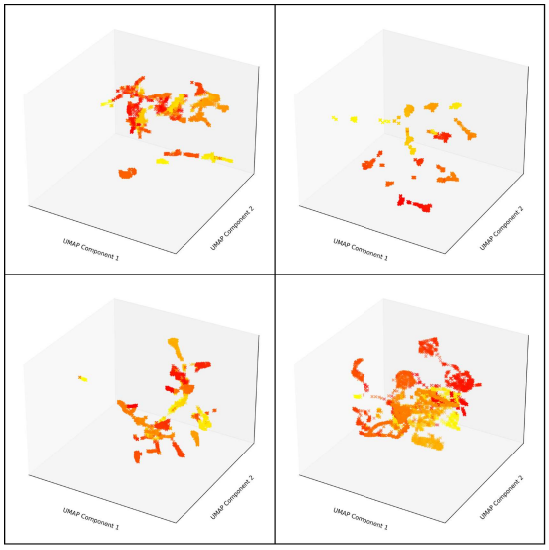


Fig. 2: 3D visualisations of the different mix-level audio representations for the Fabric 001 mix following dimensionality reduction. Top left: MFCC; top right: OpenL3; bottom left: jTFS; bottom right: CAE. Red points indicate the earliest representations, while yellow indicates the end of the mix.

Representations are extracted from audio signals with a temporal resolution of one feature per second. For the last part of audio less than one second, a feature is also computed. We introduce below specific calculations for different representations.

4.1 Mel-Frequency Cepstral Coefficients

The computation of MFCCs involves applying the discrete cosine transform (DCT) to individual time frames of the mel-log-spectrogram derived from the input audio. Empirical findings by Logan [26] demonstrated that DCT basis functions approximate principal component analysis (PCA) basis functions for music signals. This allows us to efficiently separate and analyse the various spectral features within the music.

We first compute MFCCs with a hop length of 100 ms using librosa [27], and then time-average the MFCCs within non-overlapping one-second windows to obtain per-second features. We compute the feature for the last part of the audio by time-averaging the remaining MFCC frames.

4.2 Joint Time-Frequency Scattering Transform

The joint time-frequency scattering (jTFS) transform processes audio signals with fixed filters and nonlinearities and can be implemented as convolutional neural networks [24]. With 2D filters operating on time-frequency representations derived by the wavelet transform, jTFS is invariant to time shifts and captures spectrotemporal modulations in audio signals.

Vahidi et al. [28] proposed using Euclidean distances between jTFS as the loss function for sound matching. Inspired by it, we set the hyperparameters for jTFS computation as $J = 10$, $J_{fr} = 5$, $Q_1 = 8$, $Q_2 = 2$, $Q_{fr} = 2$, $T = 1000$ ms, and $F = 16$. We split the input audio into one-second segments. We allow overlap between the last segment and the second-to-last segment to include the entire audio. We use "kymatio" [29, 30] to compute jTFS representations for these audio segments.

4.3 OpenL3

The OpenL3 embedding is produced by a self-supervised representation learning model that learns from audio-visual correspondences in large amounts of video data [23, 31]. The analysis window length of the model is fixed at one second. We compute the representations using the pretrained model provided within the Python package openl3, with a hop size of 1 second and an embedding size of 512. The last audio part is automatically padded to one second for computation.

4.4 Complex Autoencoder

The complex autoencoder (CAE) is a framework that learns complex basis functions to transform audio into representations invariant to various transformations, such as time-shift and transposition. The input of the CAE model can be time-frequency representations such as the constant-Q transform (CQT) or raw audio signals. [22] demonstrated the effectiveness of this approach by training a model on CQT representations using 3 hours of piano dataset audio, achieving state-of-the-art performance on the repeated section discovery task.

In this work, we randomly selected 50 tracks from our EDM datasets (introduced in Section 3), totalling approximately 3 hours, and trained a new CAE model on CQT representations. We used the same hyperparameters as [22]. The model approximates a 2D Fourier transform by learning a reduced number of components applied to every 32 consecutive frames of CQT data. The resulting outputs represent the magnitude spectrograms of these 2D Fourier transforms and constitute the 256-dimensional frames used in our analysis. This transformation operates on audio signals with a hop size of 90 ms (1984 samples) and an analysis window size of 2879 ms (32 hop sizes). This process does not involve any padding.

For feature extraction, we consider the temporal centres of non-overlapping one-second windows (i.e., 0.5 s, 1.5 s, ..). The centre for the last audio segment is located at half of its audio length. For each centre, we find the nearest CAE analysis window centre in time and use the corresponding CAE output frame as our per-second feature.

5 Visualisation

We then generate a series of visualisations including the self-similarity matrix of audio representations and their low-dimensional projections. Figure 3 shows self-similarity matrices derived from audio representations, from which structural patterns can be observed, particularly at transition points. We use uniform manifold approximation and projection (UMAP) [32] dimension reduction to visualise the audio representations described in Section 4 across time. By representing per-second

features as 2D or 3D vectors, we can visualise their distribution in a 2D (Figure 4) or 3D (Figure 2) space, capturing variations in audio features. Additionally, we map the resulting 3D vectors to RGB values, enabling the use of colours to extend the visualisation of audio representations (Figure 5).

6 Computing DJ Mix Heuristics

Through visualisation, we are able to observe recurring patterns in audio representations of time series corresponding to the underlying audio content. In particular, where there is a gradual change in the audio content over time, we can observe a smooth trajectory of points in the representation space, but where there is a sudden or abrupt change in the audio, we observe more distinct clustering in the representation space, as shown in Figure 4. While these structures may emerge during any audio representation time series, we find that the emergence of such structural properties in the region of a song-to-song transition during a DJ mix can reveal characteristics of that transition. DJ mixing is an art form, and there are different ways that a DJ may choose to combine tracks. Based on our listening, we believe that the presence of a trajectory in the representation space near a transition characterises a smooth transition, for instance, the gentle blending of EQ parameters or crossfading between tracks. However, the lack of long trajectories and more distinct clusters can reveal a sharper and more abrupt transition, such as a direct volume cut to the next track.

Based on these observations, we propose a method for estimating the DJ transition region based on the audio representations alone and a rough transition time stamp. We also propose a number of ways of analysing the audio representation time series at both the track and mix levels to produce heuristics for describing a track or mix, which may characterise some of their general properties and allow for comparison with other mixes. The full list of generated heuristics at the track and mix level can be seen in Tab 1 and full implementation details are available in the associated repository.

6.1 Estimating DJ Transition Regions

We estimate the DJ transition region by comparing the Euclidean distance of points in an audio representation time series at increasing time intervals from the approximate transition time stamp. We assume that if a trajectory is present, the Euclidean distance will consistently increase until the end of the trajectory (i.e., upon arriving at another cluster or trajectory) and then become more unpredictable. The heuristic can be parameterised with a stopping condition: if the Euclidean distance does not increase for a fixed number of time steps, then we set the transition region to end at the last distance increase. Additionally, a maximum transition region length can be set.

6.2 Transition Smoothness

The estimated length of a DJ transition could also be seen as one measure of smoothness insofar as longer transitions are more likely to be a smooth blend, while very short transitions are likely to be sharp. From this, you can easily derive a ReLU-like heuristic in that there is a minimum smoothness (i.e., an abrupt transition of length 0) or an infinitely long, smooth transition. Furthermore, you could derive relative smoothness by comparing the length of several DJ transitions in or across mixes.

However, a long transition is not necessarily smooth, and so we propose another heuristic for calculating its smoothness. Following the estimation of the DJ transition region, we once again use UMAP to reduce the dimensionality of the audio representation time series to 2D. We then utilise the python trajectory analysis library traja[3] to derive a measure for transition smoothness by averaging its jerk (i.e. the second derivative of the series' acceleration) in the reduced feature space for the length of the DJ transition region.

6.3 Audio Consistency

Along with smoothness, we consider audio consistency as a measure of how stylistically diverse or monotonic an audio time series is. Several heuristics are proposed for analysing an audio representation time series, with implementations that are agnostic to the length of the time series and so can be computed at the track, transition, or mix level or any desired interval.

Based on our observations of the representation space and their correspondence to the audio, we anticipate DJ mixes with greater sonic variety to include more distinct clusters. Our first heuristic is, therefore, to compute the number of clusters in a time interval. We

[3] https://traja.readthedocs.io

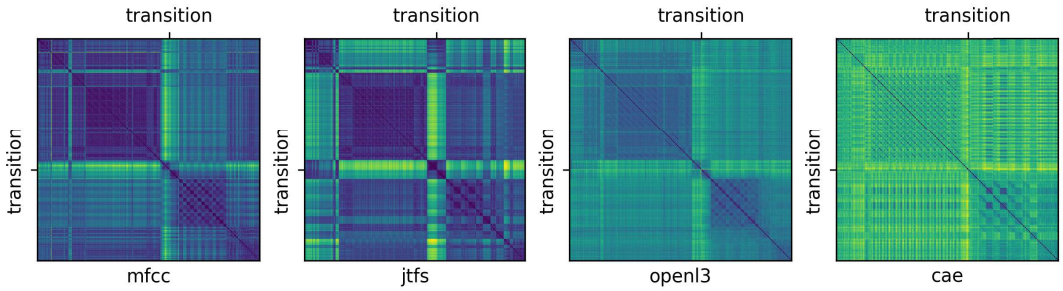


Fig. 3: Visualisation of self-similarity matrices derived from audio representations for a DJ transition. The point of transition is marked on the axes. We use the 29th and 30th tracks in the the Kode9 & Burial mix in *FABRICLIVE*.

utilise the HDBSCAN algorithm [33] and its associated library[4] for clustering. The HDBSCAN algorithm has one key tuning parameter - the number of points required for a cluster - which we set to 15 but is parameterised. We once again found that UMAP reduction was required for HBDSCAN to identify clusters more easily across all of the different audio representations.

Our second consistency heuristic computes the variance of the entire audio representation interval, as we expect DJ mixes with more sonic variety to have a greater variance in the representation space.

The final heuristic uses Maximum Mean Discrepancy (MMD) to measure audio consistency. MMD is a statistical test used to determine whether two probability distributions are the same by comparing their mean embeddings in a reproducing kernel Hilbert space. It measures the distance between the means of samples from each distribution, with a larger MMD indicating greater dissimilarity. The MMD heuristic allows us to compare the distributions of track-level and mix-level features to evaluate consistency. This heuristic can be agnostic to the length of the time interval, but a higher number of random samples should be used in larger time intervals, i.e. mix-level. In our case, we use 10 randomly sampled one-second audio representations from the time series for the song-level heuristic measurement and 100 randomly sampled one-second audio representations for the mix-level heuristic measurement to ensure reasonable coverage of the audio data at hand. However, we found that computing MMD consistency using the jTFS audio representation resulted in a long computation time - regardless of the number of random samples - due to the size of the jTFS representation. Therefore, the MMD heuristic is likely to be impractical for the jTFS representation.

[4] https://hdbscan.readthedocs.io

	Track-Level	*Mix-Level*
Mix ID	Yes	Yes
Track ID	Yes	
Track / Mix Duration	Yes	Yes
Number of Tracks		Yes
Number of Clusters *MFCC | OpenL3 | JTFST | CAE*	Yes	Yes
Variance *MFCC | OpenL3 | JTFST | CAE*	Yes	Yes
MMD *MFCC | OpenL3 | CAE*	Yes	Yes
Transition Length *MFCC | OpenL3 | JTFST | CAE*	Yes	
Transition Start *MFCC | OpenL3 | JTFST | CAE*	Yes	
Average Transition Length *MFCC | OpenL3 | JTFST | CAE*		Yes
Average Smoothness *MFCC | OpenL3 | JTFST | CAE*		Yes

Table 1: Generated heuristics at the track and mix time interval level

7 Conclusion and Potential Applications

The heuristics and visualisations presented in this work offer a novel approach to analyzing EDM music. We suggest that they can help identify variations in audio features within tracks and across DJ transitions. Furthermore, these tools have the potential to infer the structure of a DJ mix based on its audio characteristics.

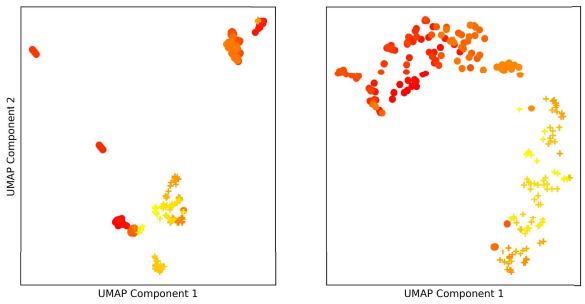


Fig. 4: 2D space visualisation of the openl3 representation over time for DJ transitions. The circle markers indicate the first track, and the plus markers indicate the second track. The red colour fades to yellow over time. The left and right figures visualise respectively the 20th and 21th tracks, and the 34th and 35th tracks, in the Kode9 & Burial mix in *FABRICLIVE*. We observe a sharper transition in the left figure and a smoother transition in the right figure.



Fig. 5: Colour visualisation of audio representations over time for a DJ transition. We use a hard transition between the first two tracks in the Kode9 & Burial mix in *FABRICLIVE*. The vertical line indicates the moment of transition from the first track to the second track, as labelled in the tracklist. Each dot represents a second of audio.

In turn, this could lead to an increased understanding of time-varying characteristics in an EDM DJ mix context and subsequently augment creative decision-making. Furthermore, the visualisation technique could be used as a basis for video mapping with complex colours, shapes, and textures, constituting a timbre-aware VJ (visual DJ) similar to [34]. The heuristics could also serve as a basis for playlist sequencing and structural segmentation.

The Fabric dataset could be also useful for other research purposes. For example, as a benchmark for mix-level structural segmentation given the approximate track timestamps.

7.1 Future Work

Future work should prioritise conducting user studies to evaluate the utility of visualisations for creative applications. It would also be beneficial to evaluate the DJ transition estimation heuristic on predicting a dataset where we have the ground truth transition data such as UnmixDB [35] so that its parameters can be fine tuned and behaviour can be refined.

We would also like to carry out a thorough evaluation of the heuristics for analysing DJ mixes. For example, we could visualise the computed heuristic features of DJ mixes and EDM tracks in 3D space, with axes corresponding to transition length, transition smoothness, and timbral consistency. Clustering could be applied to the mix-level features, and the results can be compared with categories defined by mix characteristics using metrics such as normalised mutual information (NMI) [36] to test how DL-based representations compare to conventional audio features at capturing semantic information such as genre and style.

Finally, the dataset described in this work, while useful in some ways, has the aforementioned limitations of

the UK and Western musical bias and gender bias in the mix selection [17]. Some styles of EDM music are not covered, and as the mix series ceased in 2018, contemporary EDM production will not be fully represented. Therefore, we would like to compile additional mix data for analysis that addresses these limitations while also acquiring data to explore statistical variation in other DJ mix differentiators, such as different mixes by an individual DJ, solo DJ mixes versus back-to-back DJ mixes, and studio versus live mixes. As Discogs is biased towards formally released music, this may require acquiring data from alternative resources such as 1001 Tracklists[5] as in other works [9].

8 Acknowledgements

Alexander Williams and Haokun Tian are research students at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1], Queen Mary University of London, and Sony CSL. We wish to thank Fabric and the artists involved in the *fabric* and *FABRICLIVE* DJ mix series for their contribution to our dataset and Christopher Mitcheltree for helpful discussions on the jTFS.

References

[1] McLeod, K., "Genres, Subgenres, Sub-Subgenres and More: Musical and Social Differentiation Within Electronic/Dance Music Communities," *Journal of Popular Music Studies*, 13(1), pp. 59–75, 2001, ISSN 1524-2226, 1533-1598, doi:10.1111/j.1533-1598.2001.tb00013.x.

[2] Wiltsher, N., "The Aesthetics of Electronic Dance Music, Part I: History, Genre, Scenes, Identity, Blackness," *Philosophy Compass*, 11(8), pp. 415–425, 2016, ISSN 1747-9991, doi:10.1111/phc3.12333.

[3] Munro, K., Ruthven, I., and Innocenti, P., "Can you feel it? The information behaviour of creative DJs," *Journal of Documentation*, 79(4), 2022, ISSN 0022-0418, doi:10.1108/JD-05-2022-0106.

[4] Honingh, A., Panteli, M., Brockmeier, T., López Mejía, D. I., and Sadakata, M., "Perception of Timbre and Rhythm Similarity in Electronic Dance Music," *Journal of New Music Research*, 44(4), pp. 373–390, 2015, ISSN 0929-8215, 1744-5027, doi:10.1080/09298215.2015.1107102.

[5] Smith, J. W., "The Functions of Continuous Processes in Contemporary Electronic Dance Music," *Music Theory Online*, 27(2), 2021.

[6] McAdams, S., "Timbre as a structuring force in music," *Timbre: Acoustics, perception, and cognition*, pp. 211–243, 2019.

[7] Deliège, I., "A perceptual approach to contemporary musical forms," *Contemporary Music Review*, 4(1), pp. 213–230, 1989.

[8] Kell, T. and Tzanetakis, G., "Empirical Analysis of Track Selection And Ordering In Electronic Dance Music Using Audio Feature Extraction," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[9] Kim, T., Choi, M., Sacks, E., Yang, Y.-H., and Nam, J., "A Computational Analysis of Real-World DJ Mixes Using Mix-to-Track Subsequence Alignment," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, Montreal, Canada, 2020.

[10] Wesolowski, B. C. and Hofmann, A., "There's More to Groove than Bass in Electronic Dance Music: Why Some People Won't Dance to Techno," *PLoS ONE*, 11(10), p. e0163938, 2016, ISSN 1932-6203, doi:10.1371/journal.pone.0163938.

[11] Perevedentseva, M., "Timbre and Affect in Electronic Dance Music Discourse," in *Proceedings of the 2nd International Conference on Timbre*, Thessalonkiki, Greece, 2020.

[12] Panteli, M., Rocha, B., Bogaards, N., and Honingh, A., "A model for rhythm and timbre similarity in electronic dance music," *Musicae Scientiae*, 21(3), pp. 338–361, 2017, ISSN 1029-8649, doi:10.1177/1029864916655596, publisher: SAGE Publications Ltd.

[13] Rocha, B., Honingh, A., and Bogaards, N., "Segmentation and Timbre Similarity in Electronic Dance Music," in *Proceedings of the 10th Sound and Music Computing Conference*, 2013.

[5] https://www.1001tracklists.com/

[14] Tideman, V., *Organization of Electronic Dance Music by Dimensionality Reduction*, Master's thesis, Umeå University, Sweden, 2022.

[15] Chen, Y.-X. and Klüber, R., "ThumbnailDJ: Visual Thumbnails of Music Content," in *11th International Society for Music Information Retrieval Conference*, 2010.

[16] Morse, P., "Your Questions: How To "Read" The Colours In DJ Waveforms?" in *Digital DJ Tips*, 2018.

[17] Sherburne, P., "A Deep Dive Into What Made Fabric Mixes So Essential," *Pitchfork*, 2018.

[18] Farrugia, R. and Olszanowski, M., "Introduction to women and electronic dance music culture," *Dancecult: Journal of Electronic Dance Music Culture*, 9(1), pp. 1–8, 2017.

[19] Bogdanov, D. and Serra, X., "Quantifying Music Trends and Facts Using Editorial Metadata from the Discogs Database," in *8th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[20] Heuguet, G., "Electronic music history reloaded: Ishkur's "Guide to electronic music 3.0"," *Sound Studies*, 6(2), pp. 275–279, 2020, ISSN 2055-1940, 2055-1959, doi:10.1080/20551940.2020.1794332.

[21] Caparrini, A., Arroyo, J., Pérez-Molina, L., and Sánchez-Hernández, J., "Automatic subgenre classification in an electronic dance music taxonomy," *Journal of New Music Research*, 49(3), pp. 269–284, 2020, ISSN 0929-8215, doi:10.1080/09298215.2020.1761399.

[22] Lattner, S., Dorfler, M., and Arzt, A., "Learning Complex Basis Functions for Invariant Representations of Audio," in *20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[23] Cramer, A. L., Wu, H.-H., Salamon, J., and Bello, J. P., "Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856, IEEE, Brighton, UK, 2019, ISBN 978-1-4799-8131-1, doi:10.1109/ICASSP.2019.8682475.

[24] Andén, J., Lostanlen, V., and Mallat, S., "Joint time–frequency scattering," *IEEE Transactions on Signal Processing*, 67(14), pp. 3704–3718, 2019.

[25] Davis, S. and Mermelstein, P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pp. 357–366, 1980, ISSN 0096-3518, doi:10.1109/TASSP.1980.1163420.

[26] Logan, B. et al., "Mel frequency cepstral coefficients for music modeling." in *ISMIR*, volume 270, p. 11, Plymouth, MA, 2000.

[27] McFee, B. et al., "librosa/librosa: 0.10.2," 2024, doi:10.5281/zenodo.4923181.

[28] Vahidi, C., Han, H., Wang, C., Lagrange, M., Fazekas, G., and Lostanlen, V., "Mesostructures: Beyond Spectrogram Loss in Differentiable Time–Frequency Analysis," *J. Audio Eng. Soc*, 71(9), pp. 577–585, 2023.

[29] Andreux, M., Angles, T., Exarchakis, G., Leonarduzzi, R., Rochette, G., Thiry, L., Zarka, J., Mallat, S., Andén, J., Belilovsky, E., Bruna, J., Lostanlen, V., Chaudhary, M., Hirn, M. J., Oyallon, E., Zhang, S., Cella, C., and Eickenberg, M., "Kymatio: Scattering Transforms in Python," *Journal of Machine Learning Research*, 21(60), pp. 1–6, 2020, ISSN 1533-7928.

[30] Muradeli, J., Vahidi, C., Wang, C., Han, H., Lostanlen, V., Lagrange, M., and Fazekas, G., "Differentiable Time-Frequency Scattering On GPU," in *Digital Audio Effects Conference (DAFx)*, 2022.

[31] Arandjelovic, R. and Zisserman, A., "Look, Listen and Learn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 609–617, IEEE, Venice, 2017, ISBN 978-1-5386-1032-9, doi:10.1109/ICCV.2017.73.

[32] McInnes, L., Healy, J., Saul, N., and Großberger, L., "UMAP: Uniform Manifold Approximation and Projection," *Journal of Open Source Software*, 3(29), p. 861, 2018, ISSN 2475-9066, doi:10.21105/joss.00861.

[33] Campello, R. J. G. B., Moulavi, D., and Sander, J., "Density-Based Clustering Based on Hierarchical Density Estimates," in J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pp. 160–172, Springer, Berlin, Heidelberg, 2013, ISBN 978-3-642-37456-2, doi:10.1007/978-3-642-37456-2_14.

[34] Arai, K., Hirao, Y., Narumi, T., Nakamura, T., Takamichi, S., and Yoshida, S., "TimToShape: Supporting Practice of Musical Instruments by Visualizing Timbre with 2D Shapes based on Cross-modal Correspondences," in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pp. 850–865, 2023.

[35] Schwarz, D. and Fourer, D., "Methods and Datasets for DJ-Mix Reverse Engineering," in R. Kronland-Martinet, S. Ystad, and M. Aramaki, editors, *Perception, Representations, Image, Sound, Music*, Lecture Notes in Computer Science, pp. 31–47, Springer International Publishing, Cham, 2021, ISBN 978-3-030-70210-6, doi:10.1007/978-3-030-70210-6_2.

[36] Ana, L. F. and Jain, A. K., "Robust data clustering," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pp. II–II, IEEE, 2003.