EMPIRICAL ANALYSIS OF TRACK SELECTION AND ORDERING IN ELECTRONIC DANCE MUSIC USING AUDIO FEATURE EXTRACTION

Author1
Retain these fake authors in submission to preserve the formatting

Author2
Retain these fake authors in submission to preserve the formatting

ABSTRACT

Disc jockeys are in some ways the ultimate experts at selecting and playing recorded music for an audience, especially in the context of dance music. In this work, we empirically investigate factors affecting track selection and ordering using mixes created for the Essential Mix. The Essential Mix is a well known weekly radio show on BBC Radio 1 that showcases various styles of electronic dance music. We use automatic content-based analysis and discuss the implications of our findings to playlist generation and ordering. Timbre appears to be an important factor when selecting tracks and ordering tracks, and track order itself matters, as shown by statistically significant differences in the transitions between the original order and a shuffled version. We also apply this analysis to ordering heuristics and suggest that the standard playlist generation model of returning tracks in order of decreasing similarity to the initial track may not be optimal, at least in the context of track ordering for electronic dance music.

1. INTRODUCTION

The invention of recording lead to the possibility of selecting recorded music to entertain a group of people. The idea of listening to records instead of listening to bands took off after the second world war, when sound systems and record players began to appear in night clubs and cafes in New York, Jamaica, London, Paris, and beyond [5].

Since then, the disk jockey (DJ) has evolved from a simple selector and orderer of music into a sophisticated performer with considerable skill and training. Although these performance aspects are compelling, the primary focus of this paper is the basic selection and ordering of music. DJs generally bring a limited amount of their music collection to any given gig, and play a reasonably large subset of it. Two important questions to consider are 'What tracks go into a playlist?', and 'What is the best ordering of these tracks?'. Track ordering is not a well understood process, even by DJs. Many DJs will say only that two tracks work or do not work together, and not be able to comment further. We investigate this selection and ordering process in terms of the automatically computed similarity between tracks, in terms of features representing timbre, key, tempo and loudness. The source data for this investigation is the British Broadcasting Corporation's 'Essential Mix' radio program [1] . Broadcast since 1993, the Essential Mix showcases exceptional DJs of various genres of electronic dance music (EDM), playing for one or two hours. It is considered one of the most reputable and influential radio programs in the world. By investigating the relationships between tracks in DJ sets we hope to better understand track selection by DJs and inform the design of algorithms and audio features for automatic playlisting.

The automatic estimation of music similarity between two tracks has been a primary focus of music information retrieval (MIR) research. Several methods for computing music similarity have been proposed based on content-analysis, metadata (such as artist similarity, web reviews), and usage information (such as ratings and download patterns in peer to peer networks). Music similarity is the basis of query-by-example which is a fundamental MIR task, and also one of the first tasks explored in MIR literature. In this paradigm the user submits a query consisting of one or more 'seed" pieces of music, sometimes also including metadata and user preferences. The system then responds by returning a playlist of music pieces ranked by their similarity to the query, and set in some order. In contrast our approach is analytic. Rather than generating playlists, we investigate existing DJ sets through audio feature extraction and examine the transitions between tracks in terms of audio features representing timbre, loudness, tempo, and key. We also compare the results of our empirical investigation to common ordering methods, and offer some suggestions for improving current playlisting heuristics.

2. RELATED WORK

Early MIR work investigating the automatic calculation of music similarity and how to evaluate different approaches formulated a general methodology that is followed by the majority of existing work to this day. In this methodology, the primary goal is assessing the relative performance of different algorithms for computing music similarity by somehow evaluating the 'quality' of the generated playlists.

The most common approach of generating a playlist is to consider the N closest neighbors in terms of automat-

[1] `http://www.bbc.co.uk/programmes/b006wkfp`

ically calculated similarity to a particular query. Several automatic playlists are generated from seed queries representing the desired diversity of the music considered. This set of automatically generated playlists is then evaluated, typically using one or both of two approaches: objective evaluation using proxy ground truth for relevance, or subjective evaluation through user studies. The basic idea is to evaluate a playlist by considering it good if it contains a high number of 'relevant items" to the query [1]. The relevance ground truth can be provided by users in subjective evaluation but this is a time consuming and labor intensive process that does not scale well.

Objective evaluation has the advantage that it can scale to any number of queries and playlists, as long as the tracks have some associated meta-data that can be used as a proxy for relevance. Common examples of such proxy sources include artist, genre, and song [1, 10]. In addition to music similarity calculated based on audio content analysis, other sources of information such as web reviews, download patterns, ratings and explicit editorial artist similarity [7] can also be used for estimating music similarity [6]. Playlists themselves have also been used to calculate artist and track similarity based on co-occurrence. Sources of playlist data include the Art of the Mix, a website that contains a large number of hobbyist playlists [4], and listings of radio stations [12]. DJ sets remain an untapped resource, however.

The most common relevance-based evaluation measures (such as precision, recall and F-measure [2]) are borrowed from text information retrieval and only consider the items contained in the set of returned results, without taking into account their order. The paradigm of a single seed query song creating a list of N items ranked by similarity has remained a common approach to automatic playlisting and music recommendation. Some notable exceptions in terms of ordering include: heuristics about trajectory for the ordering of returned items [10], using song sets instead of single seeds [11], ordering based on the traveling salesman problem [17], and considering both a start track and an end track for the playlist [8]. The assumption of similarity has also been challenged by the finding that in many cases users prefer diverse playlists [18] as measured by automatic feature analysis. This is the closest work in terms of approach to the work described in this paper.

Another theme of more recent work has been providing more user control to the process of automatic playlist generation. If the tracks considered are associated with a rich set of attribute/value pairs then techniques from constraint satisfaction programming and inductive learning can be used to generate playlists that to some extent optimally satisfy the user preferences [16]. The ability to control what attributes are used for estimating music similarity has also been investigated [20]. One of the simplest forms of user control is skipping behavior, which has been used to iteratively improve playlist generation [15]. A more elaborate method for steerable playlist generation is based on tag clouds and and a music similarity space derived from radio station playlists [13]. Physiological data such as heart rate has also been investigated for playlist generation [14]. The novelty aspect of playlist generation by tracking user listening information has also been explored [9]. A different approach altogether is to create playlists visually, based on some graphical representation of the music collection [19]. In all of this literature, different approaches to playlist generation are also evaluated with a combination of objective measures and user studies, comparing different configurations to a random or a simple algorithmic baseline. For example, a recent study compared two recommender systems (based on artist similarity and acoustic content) with the Apple iTunes genius recommender which is believed to be based on collaborative filtering [3].

3. MOTIVATION AND PROBLEM FORMULATION

The motivation behind our work is to investigate the process of playlist/mix creation by analyzing existing mixes created by experts, i.e DJs. Existing work has mostly focused on more general playlists created by average listeners. Rather than relying on user surveys, we focus on empirical analysis based on audio feature extraction. This allows us to investigate what audio attributes DJs use when selecting and order their tracks. We further compare these attributes to collections of random EDM tracks, and to artist albums. We specifically investigate whether track order matters. We also examine important assumptions that are frequently made by automatic playlist generation systems. Specifically, we investigate if ordering based on similarity ranking is a good choice, and if so, in what manner.

In existing literature these assumptions are typically manifested in the design of an automatic playlist algorithm, and the results are evaluated through objective or subjective approaches. Issues such as the sameness problem in playlists formed from collections of music that do not have stylistic diversity, or the playlist drift problem in large diverse collections are also discussed but are not empirically supported [8]. In contrast, our approach is complimentary and attempts to test these assumptions directly on existing mixes. Our methodology can be also viewed as an empirical musicological approach to understanding how DJs select and order music.

4. METHODOLOGY

4.1 Data

We obtained 114 Essential Mix DJ sets from the archival website `themixingbowl.org` (**DJS**). These sets cover three years (2009-2011) of the radio show. In addition, a collection of 189 artist albums (**ALBS**) (from the author's own collection, covering a wide range of music and genres) was used for comparison purposes. Finally, 100 random EDM (**REDM**) track sets were created by randomly picking tracks from the collection of electronic dance music (covering 1,261 tracks) of one of the authors who frequently performs as a DJ. (It is possible, but unlikely, that these is overlap between the collections. If there is overlap, our method of using transitions should make its impact minimal.) In order to investigate track ordering we created a random shuffle of each Essential Mix DJ set (**RDJS**).

The DJ sets consist of continuous audio, without timing information about the individual tracks. Each set was split into two-minute exceprts followed by two-minute gaps, giving 30 excerpts of audio per two hour set. This accounts for any mixing that the DJ might be doing, and factors in track length. We considered more elaborate manual approaches such as extracting each track from each set manually or selecting the most representative part of each track for processing. As this is a very time consuming process this would have severely limited the amount of audio data we would have been able to process.

The main issue with the two-minute exceprt approach is that it may miss vital sections, in terms of audio features and include transitions between sections of the same track. By using the exact same approach for albums and for the randomized EDM sets we believe that whatever effects this arbitrary segmentation has will be approximately the same for all data sets and the relative comparisons we make are valid. It must be noted that the REDM set was not mixed, unlike the DJS and RDJS set. However, the audio was concatenated and the same 2-minute exceprt / 2-minute gaps methodology was applied to it. Thus, some exceprts will contain parts of two tracks, which we hope will approximate the impact of mixing.

4.2 Audio Feature Extraction

Our goal is to examine different factors affecting selection and ordering, based on automatic audio feature extraction. More specifically, we examine the effects of features representing timbre, key, loudness, and tempo. The audio features used for the evaluation were computed using two sources: the Echo Nest AnalyzeAPI [2] and Marsyas [3]. The Echo Nest Analyze API returns timbre data as a 12-dimensional vector, where each element matches a spectral characteristic: the first dimension is loudness, the second indicates a strong attack, and so on. Timbre data is given for each 'segment', which roughly corresponds to musical events detected by onsets (on average about half a second of audio). The mean of each dimension was taken to supply a single vector for each 2-minute excerpt of audio. The Analyze API returns loudness as decibels, and tempo as beats per minute. Key is returned as a tuple of pitch class and major or minor mode.

Marsyas returns a 63-dimensional vector for representing timbre. The features used are based on the Spectral Centroid, Roll-Off, Flux and Mel-Frequency Cepstral Coefficients (MFCC). To capture feature dynamics, a running mean and standard deviation over the past M frames of 23 milliseconds is computed. The features are computed at the same rate as the original feature vector but depend on the past M frames (e.g. M=40, corresponding approximately to a so-called "texture window" of 1 second). This results in a feature vector of 32 dimensions at the same rate as the original 16-dimensional one. The sequence of feature vectors is collapsed into a single feature vector representing the entire audio clip by taking again the mean and standard deviation across the track (the sequence of dynamics features) resulting in the final 64-dimensional feature vector per 2-minute excerpt. For tempo, a method based on autocorrelation of an onset detection function and the creation of a beat histogram is utilized. There is no direct key estimation implemented in Marsyas.

[2] http://echonest.com
[3] http://marsyas.info

4.3 Metrics

From this audio featured data, we characterize the transitions between successive feature vectors corresponding to 2-minute excerpts in order to examine track selection during the course of a set. Tempo and loudness were simply subtracted, and key was represented by the change in key signature. In order to characterize the transition of the timbre vectors, we considered both the L1 (Manhattan) distance and L2 (Euclidean) distance between successive vectors after each dimension was max/min normalized across the data set under consideration. The analysis conclusions were similar for the two distance metrics (L1 and L2). Due to space limitations, we only report numbers based on the L1 metric. Although more elaborate distance metrics between the timbre vectors can be devised, we prefer to a use a simple metric and normalize each feature dimension as this provides a consistent, easily repeatable method.

Another objective of our analysis is to investigate the importance of ordering in DJ sets assuming a fixed set of tracks (in this case, 2-minute excerpts) to be played. In order to characterize different orderings of a set of tracks, we use ideas from combinatorics and permutations. More specifically, we want to compare excerpt orders created using different heuristics to the excerpt order of the original DJ set. Therefore, we require some measure of similarity between different permutations.

We utilize the concept of inversions, which are a way of measuring the differences between an ordered list and a permutation of that ordered list. For example, given some ordered list (e.g. [A, B, C, D, E]) and a permutation of it (e.g. [B, C, A, E, D]), an inversion is a pair of positions where the entries are in the opposite order. The permutation in our example has three inversions: (0, 2) for the pair (B, A); (1, 2) for the pair (C, A), and (3, 4) for the pair (E, D). The number of inversions between two different orderd lists of the same items gives a measure of how similar they are in terms of ordering. We consider the original DJ set order as the original sequence and the sequences created by four different ordering heuristics as permutations.

Assuming a fixed set of excerpts corresponding to a particular DJ set, we consider the following heuristics: *Rank* - the excerpts are ranked by increasing distance to the initial excerpt of the original order; *NN* - each successive excerpt is the nearest neighbor of the previous one, without allowing repetitions; *Median* - the distances of all the clips to the initial exceprt are computed and the one that is closest to the median distance is selected as the next excerpt, without allowing repetitions; *Furthest Neighbor* - each successive clip is the furthest neighbor of the previous one, without al-

Table 1. Timbre transition statistics (EN: Echonest, MRS: Marsyas)

	Mean ± Std	Q1	Median	Q3
DJS-EN	1.09 ± 0.44	0.76	1.02	1.34
ALBS-EN	0.86 ± 0.4	0.58	0.80	1.05
REDM-EN	1.44 ± 0.57	1.03	1.39	1.80
RDJS-EN	1.20 ± 0.44	0.87	1.13	1.45
DJS-MRS	5.75 ± 2.02	4.35	5.41	6.82
ALBS-MRS	4.43 ± 1.91	3.11	4.11	5.31
REDM-MRS	6.85 ± 2.51	5.18	6.64	8.40
RDJS-MRS	6.24 ± 2.09	4.76	5.9	7.37

Table 2. Tempo transition statistics (EN: Echonest, MRS: Marsyas), Values are in BPM

	Mean ± Std	Q1	Median	Q3
DJS-EN	8.93 ± 23	0.03	0.12	1.51
ALBS-EN	17.82 ± 20.69	1.87	10	27.92
REDM-EN	6.53 ± 15.25	0.02	0.88	5.05
RDJS-EN	11.49 ± 24.07	0.06	1.04	7.88
DJS-MRS	3.86 ± 13.17	0	0	1
ALB-MRS	20.19 ± 22.62	1	11	35.5
REDM-MRS	7.41 ± 16.18	0	0.5	6
RDJS-MRS	6.6 ± 15.01	0	1	4

lowing repetitions. The *Rank* heuristic corresponds to the common scenario of ranked list retrieval.

Table 3. Statistics of transitions for loudness

Loudness	Mean ± Std	Q1	Median	Q3
DJS	0.76 ± 0.71	0.28	0.59	1.06
ALBS	3.39 ± 4.13	0.92	2.16	4.32
REDM	3.18 ± 3.07	1.02	2.26	4.37
RDJS	0.88 ± 0.85	0.31	0.67	1.19

5. DATA ANALYSIS

5.1 Transition Analysis

We characterize the distribution of transition values (L1 distances) for each configuration (DJ sets, artist albums, random EDM and random shuffle DJ sets) by computing statistics (mean, standard deviation, median and quartiles) for each different factor (timbre, tempo, loudness, key). In order to characterize statistical significance we use the Welch *t-test* which is appropriate for samples that may have unequal variance. All differences reported here are strongly statistically significant, with $p < 0.0001$.

The statistics for timbral transitions in Table 1 can be used to support various assumptions that are commonly made in automatic playlist generation systems. For example, the relations between DJS and REDM show that there is more to DJ selection than just randomly selecting tracks of EDM. The average timbral transition between exceprts for the DJ sets is smaller than the timbral transition between exceprts of random EDM tracks. This implies that that DJs try to pick tracks that are similar in timbre, and order them in ways that further minimize the timbral differences. The timbral transitions between excerpts of albums are the smallest. This is reasonable, as albums tend to be sonically coherent, featuring the same instruments and orchestration throughout. Figure 1 shows timbre transition numbers for single examples of each category. Furthermore, by comparing DJS and RDJS it can be seen that track ordering is important: the original ordering results in smaller transitions on average than a random shuffle of the same clips. These findings are supported by both the Echo Nest (EN) and Marsyas (MRS) feature extraction, increasing our confidence in their validity.

Another factor to examine is tempo. As can be observed from Table 2, DJ sets have the least amount of tempo change. Tempo is something that can be (and is almost always) controlled by the DJ. Therefore it is not surprising that small tempo transitions are observed. As expected, due to the large variety of genres and the unreliability of tempo detection for some genres, the ALBS dataset shows the highest tempo transitions. The effect of ordering is less pronounced than in the case of timbre, as can be observed by comparing DJS and RDJS, but is still there. Somewhat surprisingly, the random selection of EDM tracks also exhibits small tempo transitions, probably due to the consistent use of a small range of tempi in this style of music (House music ranges from 110 BPM to 130 BPM, with a large peak around 120 BPM, for example). However, note the increase in the median tempo transition from the DJS dataset to RDJS and REDM: the tempo transitions for randomized DJ sets are similar to those of randomized EDM tracks. When a DJ takes control of the tempo, the median transition drops significantly.

Loudness can easily be (and is almost always) controlled by the DJ. We expect that DJ sets will be relatively homogeneous. This is clearly shown by examining the data in Table 3, and contrasting DJS with REDM. Ordering also has a small effect, as can be seen by examining the relation between DJS and RDJS. Thus, DJs appear to vary volume slightly over the course of a set. We also examined key using the Echo Nest's analysis, but did not find any stastical significance in the differences observed. We can thus suggest that DJs do not use key as a primary concern when selecting and ordering their tracks - unlike classical musicians, for whom key and harmony are paramount.

5.2 Analysis of Ordering Heuristics

In addition to transition analysis, we also examined different heuristics for ordering playlists and compared them to the 'golden' order of the original DJ set. Table 4 shows the results of this comparison using the number of inversions as an estimate of how 'close' two orderings are. Based on the above transition analysis, timbre is the dominant factor

Table 4. Average number of inversions per heuristic

Method	Echo Nest	..	Marsyas	..
..	HEUR	RND	HEUR	RND
Ranked	186.27	201.10	183.71	192.28
NN	187.64	202.5	186.77	189.43
Median	171.85	203.86	175.89	172.59
FN	192.71	202.00	191.33	189.35
Equal-Step	172.75	199.50	170.16	174.71

affecting playlist selection and ordering. Thus, it is the parameter used in Table 4. For each heuristis we report the average number of inversions, across all sets in DJS, comparing the original order of the tracks with the heuristic order (HEUR) and as a baseline the number of inversions comparing the original order of the trakcs with the order of a random shuffle (RND). From Table 4 it can be seen that all ordering heuristics come closer to the original ordering than to a random shuffle, indicating that they are reasonable choices.

One of the most interesting findings is that the traditional ordering based on ranked similarity is not the best heuristic. Both the *Median* and *Equal-step* heuristics appear to be closer to the original set order. This implies that consistent transitions are more important than tracks near the start of the mix being similar to the initial track. Figure 2 shows the timbral transitions for a specific DJ set as well as two orderings. As can be seen in the middle subfigure the *Rank* heuristic results in playlist drift near the end while the *Median* heuristic provides more balanced transitions.

6. DISCUSSION AND FUTURE WORK

We have proposed and demonstrated the use of automatic audio feature extraction to examine the selection and order of tracks in DJ mixes. Our approach is distinct and complimentary to the traditional approach of generating playlists automatically and evaluating them through proxy ground truth and user studies. It is an analytic approach that uses the playlists as data to be analyzed. We also specifically focus on DJ selection of EDM tracks rather than music in general.

Our transition analysis has shown timbre to be an important attribute used by DJs when selecting and ordering tracks. DJ mixes are more timbrally similar than random EDM tracks, though not as timbrally similar as artist albums. Tempo and loudness tend to be controlled by the DJ, and this is also reflected in our findings. Our results support the intuitive idea that DJs tend to play tracks that broadly 'sound the same', and fits with the typical statement by DJs that two tracks 'work' together although there is probably many more subtle factors involved in DJ track selection and ordering. The results also support the emphasis on timbral similarity that is common in automatic playlist generation systems. Our findings are consistent between the two different audio feature front-ends and confirm design choices that have been made in music recommendation and automatic playlist generation systems.

The order of the selected tracks matters, as shown by statistically significant differences in the transitions between the original order and a shuffled version. These ordering differences were found in all factors considered except key. The investigation of ordering heuristics implies that the standard playlist generation model of returning tracks in order of decreasing similarity to the initial track may not be optimal (at least in the context of DJ ordering for EDM). Returning results ranked by similarity may not be optimal, and transitions of roughly equal size are probably a better choice for automatic playlist generation algorithms.

Future work includes the analysis of more data, such as the total 900 Essential Mixes rather than 114 mixes considered here. Commercial mixes are also a possibility, as are DJ mixes from the wider internet. We would also like to follow a similar approach to the analysis of playlists across a variety of genres, as well as playlists created by everyday listeners and music recommendation systems. In terms of informing automatic playlist generation algorithms, the most promising direction is to investigate the effectiveness of ordering heuristics that emphasize smoothness of transitions rather than absolute ranking.

Track selection and ordering is a tricky process that is not totally understood even by DJs themselves: It is hoped that this paper has shed some light on the role that timbre, key, volume and tempo play in this process. We hope that our work informs future work in automatic playlist generation and music recommendation, and that the proposed methodology inspires more empirical musicological analysis of how DJs select and order tracks.

7. REFERENCES

[1] J.J. Aucouturier and F. Pachet. Music similarity measures: Whats the use. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

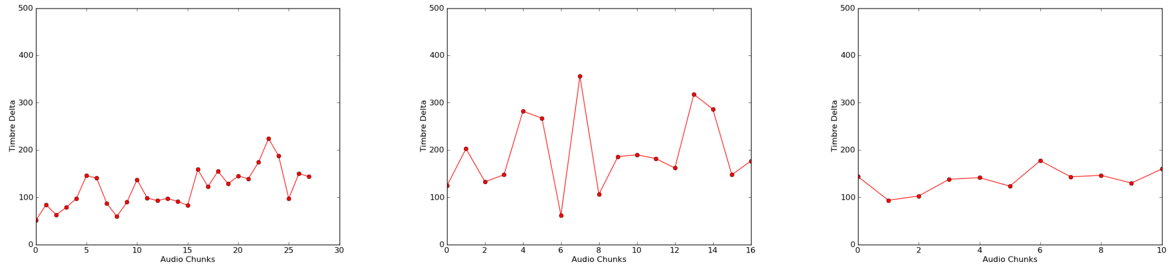
[2] J.J. Aucouturier and F. Pachet. Tools and architecture for the evaluation of similarity measures: case study of timbre similarity. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2004.

[3] L. Barrington, R. Oda, and G. Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2009.

[4] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

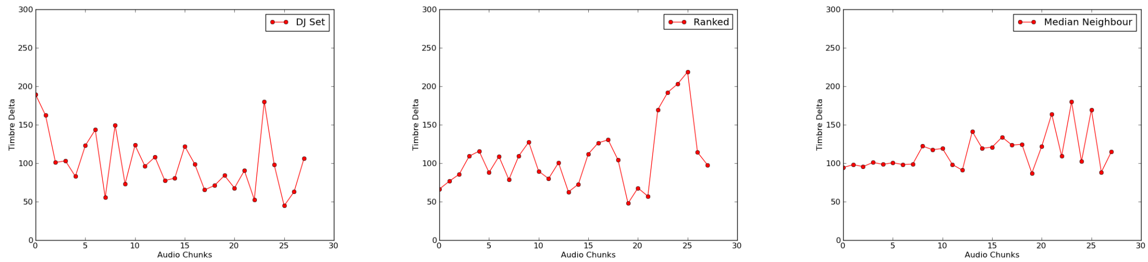
[5] B. Brewster and F. Broughton. *Last Night a DJ Saved My Life: The History of the Disc Jockey*. Grove Press, 2000.

[6] D.P.W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.



(a) DJ set (b) Random Singles (c) Artist Album

Figure 1. Change in timbre over time, for examples of each dataset



(a) Original DJ set (b) Rank heuristic ordering (c) Median heuristic ordering

Figure 2. Specific example of transitions for different orderings

[7] B. Fields, C. Rhodes, M. Casey, and K. Jacobson. Social playlists and bottleneck measurements: Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2008.

[8] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proc. Int. Conf. of Music Information Retrieval (ISMIR)*, 2008.

[9] Yajie Hu and Mitsunori Ogihara. Nextone player: A music recommendation system based on user behavior. In *Proc. Int. Conf. of the Soc. for Music Information Retrieval (ISMIR)*, 2011.

[10] B. Logan. Content-based playlist generation: Exploratory experiments. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2002.

[11] B. Logan. Music recommendation from song sets. In *Proc Int. Conf. on Music Information Retrieval (ISMRI)*, 2004.

[12] F. Maillet, D. Eck, G. Desjardins, and P. Lamere. Steerable playlist generation by learning song similarity from radio station playlsits. In *Proc. Int. Conf. of the Soc. for Music Information Retrieval (ISMIR)*, 2009.

[13] François Maillet, Douglas Eck, Guillaume Desjardins, Paul Lamere, et al. Steerable playlist generation by learning song similarity from radio station playlists. In *Proc. Int. Conf. on Music Information Retrieval*, 2009.

[14] N. Oliver and L. Kreger-Stickles. Papa: Physiology and purpose-aware automatic playlist generation. In *Proc. 7th Int. Conf. Music Inf. Retrieval*, pages 250–253, 2006.

[15] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[16] S. Pauws and S. van de Wijdeven. User evaluation of a new interactive playlist generation concept. In *Proc. Int.l Conf. on Music Information Retrieval (ISMIR)*, 2005.

[17] T. Pohle, E. Pampalk, and G. Widmer. Generating similarity-based playlists using traveling salesman algo- rithms. In *Proc. Int. Conf. on Digital Audio Effects (DAFX)*, 2005.

[18] M. Slaney and W. White. Measuring playlist diversity for recommendation systems. In *Proc. ACM workshop on audio and music computing multimedia*, 2006.

[19] R. Van Gulik and F. Vignoli. Visual playlist generation on the artist map. In *Proc. Int. Conf. on Music Information Retrieval(ISMIR)*, 2005.

[20] F. Vignoli and S. Pauws. A music retrieval system based on user-driven similarity and its evaluation. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.