SmartDJ: An Interactive Music Player For Music Discovery By Similarity Comparison

Maureen S. Y. Aw, Chung Sion Lim, Andy W. H. Khong
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
E-mail: {maureenaw, limcs}@outlook.com, andykhong@ntu.edu.sg

***Abstract*—We present a user-friendly method that employs acoustic features to automatically classify songs. This is achieved by extracting low-level features and reducing the feature space using principle component analysis (PCA). The songs are then plotted on a song-space graphic user interface (GUI) for manual or automatic browsing. The similarity between songs is given by the Euclidean distance in this lower-dimension song space. Using this song space, a prototype application known as the "SmartDJ" has been implemented on the MAX/MSP platform. This prototype application enables users to visualize their music library, select songs based on their similarity or automate the song selection process using a given seed song. We also describe, in this paper, several features of the application including the smooth mix transition feature which provides an enhanced experience for the users to perform song transition seamlessly.**

I. Introduction

In the digital age of music, music organization and discovery have become more challenging and time consuming than before. This is due to the large pool of music available and the introduction of new songs with time. Hence, a smart system that is capable of organizing and presenting a large collection of music in one's personal library is essential [1], [2].

The motivation behind our prototype application, which we called *SmartDJ*, is to create a system that organizes music based on acoustic similarity. In this work, we propose a new and interactive way of visualizing a personal music library by translating all songs in a music library into points on a two-dimensional song space which, in turn, serves to provide visual feedback for the user. The acoustic similarity between the songs is then determined by the proximity of these points; points which are closer correspond to songs with higher similarity. In order to achieve the acoustic similarity comparison, low-level descriptors are extracted from the raw audio signal and the large dataset is subsequently reduced via the use of the principal component analysis (PCA.) These descriptors are then plotted onto the two-dimensional song space for similarity comparison.

In addition to music organization, our proposed *SmartDJ* prototype application is capable of recommending songs to users by automatically generating a playlist of acoustically similar music based on a given seed song. Alternatively, users can manually perform song selection based on visual feedback via the song-space visualizer. In order to cater to different listening needs of users, the two-dimensional song-space model can be adjusted according to high-level concepts that indicate social contexts for music listening. In addition, deejay (DJ) transition techniques such as beat synchronization, key matching and equalization mixing allow for the smooth transition between songs so as to further enhance users' listening experience.

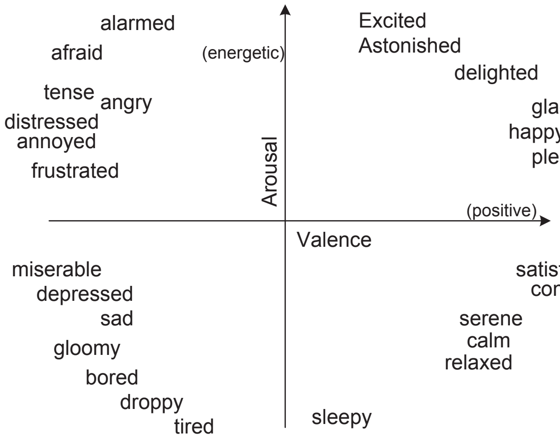


Fig. 1. The Thayer's mood model.

II. Background

Music can be classified in terms of emotions and the classification of musical mood via the use of audio features is often referred to as audio mood classification (AMC). A well-know model for AMC is the Russell's model [3] which divides mood into two uncorrelated dimension vectors: arousal and valence, as illustrated in Fig. 1. In this model arousal can be described as the energy or activation of an emotion; a low arousal corresponds to music that is perceived to be sleepy or sluggish while a high arousal corresponds to frantic or excited. Valence, on the other hand, describes how positive or negative an emotion is. A low valence corresponds to songs which are perceived to be negative, sad or melancholic while a high valence corresponds to positive feelings, happy or joyful.

A method that exploits intensity, timbre and rhythm has been proposed for the classification of music mood into four nominal classes resembling the four quadrants in the mood plane spanned by two vectors [4]. The first quadrant (excited and positive) corresponds to "happy/excited" emotion, the second quadrant (excited and negative) corresponds to "angry/anxious" emotion, the third quadrant (calm and negative) corresponds to "sad/bored" emotion while the last quadrant

TABLE I
LOW-LEVEL DESCRIPTORS

flatness	brightness
root-mean-square (RMS)	rolloff
low energy	pitch
average silence ratio (ASR)	key (chords)
event density	key (major/minor)
tempo/beats-per-minute	key (clarity)
pulse clarity	zero-crossing rate (ZCR)
centroid	MFCCs (13 coefficients)

(calm and positive) corresponds to "relax/serene" emotion. It is therefore possible to project songs onto a song space with different quadrants representing songs of different moods so as to match the user's social context. The user can then select songs from a particular region on the song space which suits his/her current social context, for e.g., working out in a gym, romantic dinner, song before bed, etc.

In addition to the mood model, other forms of song and genre classification methods exist. The use of mel-frequency cepstral coefficients (MFCCs), spectral, and cepstral parameters has been proposed [5]. This method employs linear discriminant analysis for dimension reduction so as to project the data for optimal class separation. The use of growing neural gas as a form of self-organizing map and hidden Markov models (HMMs) have also been proposed in [6] and [7], respectively, for genre classification. It is useful to note that our focus is not to classify the songs into a specific category (i.e., classification.) The aim of our work, however, is to represent pattern within the musical set on a two-dimensional song space so as to establish relationships between songs.

III. SONG-SPACE DEVELOPMENT

A. Feature Extraction

To determine and quantify the acoustic similarity between songs, twenty-eight low-level descriptors listed in Table I are first extracted from the raw audio signal using the MIRtoolbox [8]. For example, defining $x(n)$ as the music signal at time n, the flatness, defined by the ratio of the geometric to arithmetic mean, i.e.,

$$\mathcal{F}(n) = \frac{\left[\prod_{n=0}^{N-1} x(n)\right]^{1/N}}{\frac{1}{N}\sum_{n=0}^{N-1} x(n)}, \qquad (1)$$

is computed. This measure indicates whether the distribution of the signal is smooth or spiky. The event density, on the other hand, estimates the average number of note onsets per second.

A corpus set of 310 songs were used for feature extraction. The stereophonic channels of each song are encoded in ".wav" lossless audio format with 16-bit resolution resulting in an audio bit rate of 1,411,200 bits/s. A window length of 50 ms with a 50% overlapping factor was used for the segmentation. As opposed to that of [5] where the first 30 s of the song is analyzed, 30 s from the middle segment of the audio were analyzed in our work. This short-segment was chosen in order to reduce the computational power, time and memory space required. The middle segment of the song was chosen since it encapsulates the gist or chorus of a song in general.

TABLE II
SPECTRAL-SHAPE FEATURES

flatness	brightness
centroid	roll-off
zero-crossing rate (ZCR)	root-mean-square (RMS)
low energy	event density
average silence ratio (ASR)	

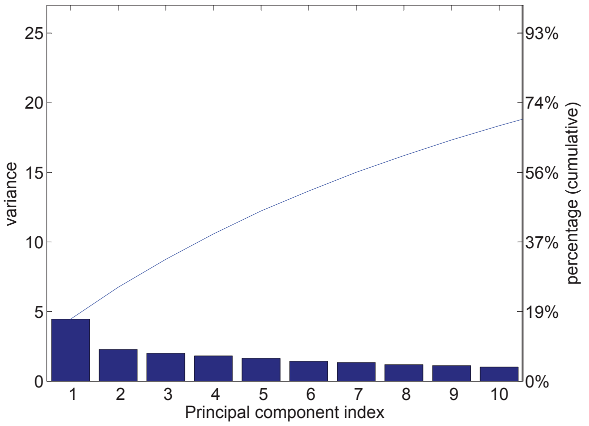


Fig. 2. Variance of the first ten principal components (bar) and their cumulative percentage (line) using low-level descriptor features listed in Table I.

To quantify the amount of dimension reduction via the use of principle component analysis (PCA) for the low-level descriptors listed in Table I, we first define σ_i^2 as the variance of the ith principal component (PC.) The variance σ_i^2, $i = 1, \ldots, 10$ obtained from each of the first ten PCs along with its cumulative variance (in percentage) defined by

$$\sigma_{\mathrm{c},i}^2 = \left[\frac{\sigma_{\mathrm{c},i-1}^2}{100} + \frac{\sigma_i^2}{\sum_{j=1}^{28}\sigma_j^2}\right] \times 100, \qquad (2)$$

where $\sigma_{\mathrm{c},0}^2 = 0$, are illustrated in Fig. 2. It can be seen that the first two PCs have a cumulative variance of only 25% of the entire data. The reason for the low percentage obtained was due to overfitting, i.e., too many features have been used to describe the dataset. To resolve this overfitting problem, the number of features employed had to be narrowed down to a more specific set of descriptors. Further tests revealed that some of the features such as tempo, key strength and MFCCs do not load the PCs significantly and are therefore removed from the feature dataset. In addition, spectral-shape parameters such as listed in Table II are well-known to be important parameters for many music classification tasks [9], [10]. Hence, these spectral-shape features were chosen to describe the proposed song space.

B. Dimension Reduction

It is useful to note that although a large dataset can be reduced in dimension via PCA or support vector machines (SVMs,) SVMs may not be well-suited for visualization

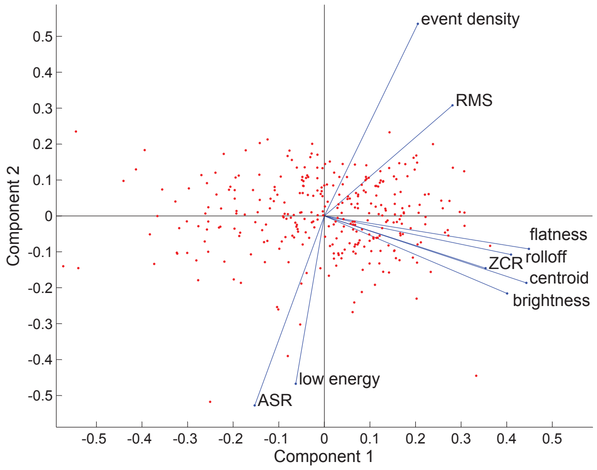


Fig. 3. The song-space model. Each dot denotes a song in the database.

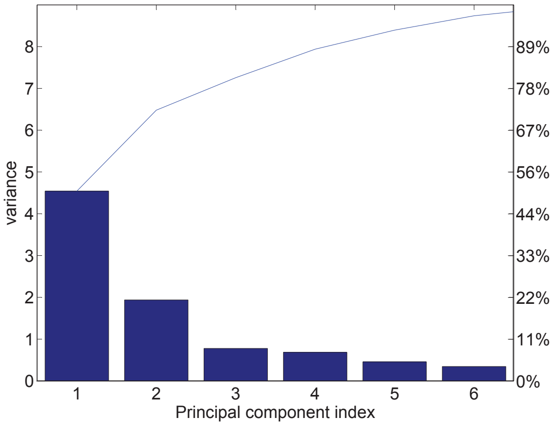


Fig. 4. Variance of the first six principal components (bar) and their cumulative percentage (line) using spectral-shape features listed in Table II.

since the classifiers tend to result in a number of tight clusters [11]. Therefore, using PCA, the dataset is projected onto a two-dimensional song space as a form of visual feedback to the user. The similarity between the songs is given by the Euclidian distance in a lower-dimension song space with similar songs being situated near each other. With PCA, the highest variance is retained in the first PC while the second PC accounts for the second largest variance in the data, orthogonal to the first. Hence, as shown in Fig. 3, the data is plotted against the first two PCs to illustrate the maximum amount of variance in the song database.

The percentage variance obtained from the first six spectral-shape features (out of the nine in Table II) is shown in Fig. 4. The cumulative percentage variance obtained from the first three PCs account for 51.05%, 72.87% and 81.64% of the data respectively. This is a significant improvement from the 25% cumulative variance with the twenty-eight low-level features in Table I. A sharp bend at the second PC in Fig. 4 indicates that the variability contributed by the third and subsequent PCs are not as significant. Hence, a two-dimensional plot using the first two PCs is employed in our model.

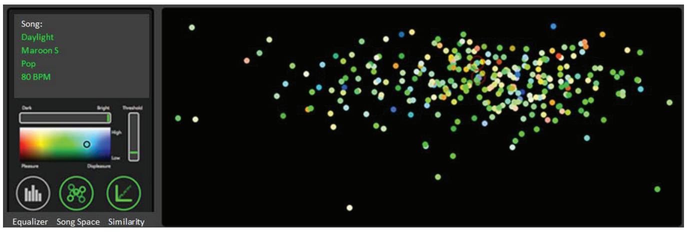

Song:
Daylight
Maroon 5
Pop
80 BPM

Fig. 5. Song-space visualizer in *SmartDJ* GUI.

C. *The Song-Space Model*

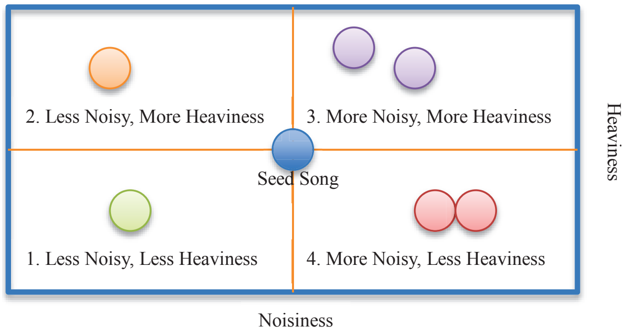
With the song-space model obtained in Fig. 3, it was generally observed that as songs progress along the horizontal axis (first PC,) it gets “nosier.” In terms of genre, this implies that songs transverse from Jazz, Acoustics and light-hearted Country songs to Rock, Techno, Pop and House music. The vertical axis (second PC) increases in energy level as it progresses from bottom to top; Jazz music is generally located at the bottom, while Pop music is generally located at the upper half of the song space. Therefore the second PC relates to the massiveness or the heaviness of a song. This analysis is supported by the individual feature loadings illustrated by lines in Fig. 3. It can be seen that loadings for features such as flatness, rolloff, zero-crossing rate (ZCR), centroid and brightness lie closer to the horizontal axis as they quantify the amount of high-frequency energy and the amount of oscillation of the signal. This is in line with the fact that the horizontal axis quantifies how noisy, or saturated the music is with regard to the high-frequency content. Contrary to the above, loadings of event density and root-mean-square (RMS) lie in the direction close to the positive y-axis while loadings of low energy and average silence ratio (ASR) point in the negative y-axis direction. This agrees with the fact that the vertical axis quantifies the amount of energy or how massive or heavy the music is.

IV. The *SmartDJ* Application and its Features

The song-space model shown in Fig. 3 is next ported to the MAX/MSP platform for the development of the graphic user interface (GUI.) A new song-space visualiser has been created and, as shown in Fig. 5, it offers a new way of interaction between the user and the music player. This is achieved by providing a visual feedback with regard to song similarity and adding the relationship between key proximity and beats-per-minute (BPM) into the music library. This helps users to visualize the relationship between songs around a particular seed song. The visualizer incorporates five-dimension data on a 2D space. The two dominant dimensions define the noisiness and heaviness as described in Sections III-B and III-C whereas the BPM, pitch and key proximity attributes of a song are represented using the hue-saturation-luminance (HSL) color model on the song space.

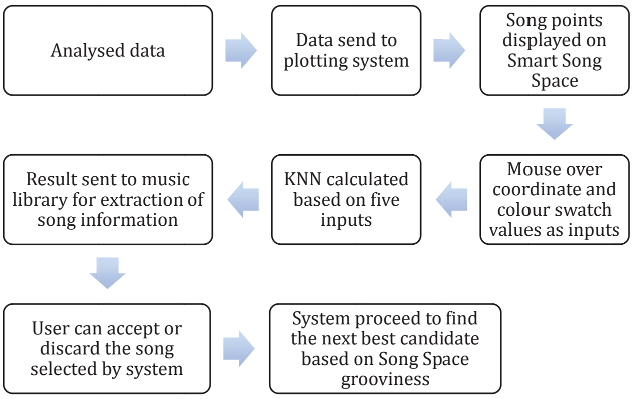
A. *Song Recommendation Feature*

In addition to the above, our proposed *SmartDJ* application allows the user to opt for automatic song selection based on



2. Less Noisy, More Heaviness
3. More Noisy, More Heaviness
Seed Song
1. Less Noisy, Less Heaviness
4. More Noisy, Less Heaviness
Heaviness
Noisiness

Fig. 6. Song-space categorization.



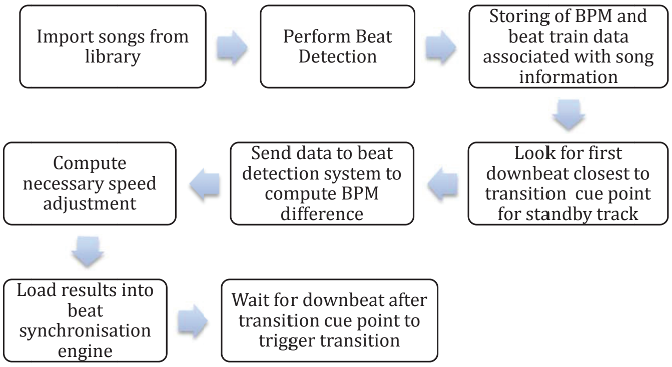
Analysed data
Data send to plotting system
Song points displayed on Smart Song Space
Mouse over coordinate and colour swatch values as inputs
KNN calculated based on five inputs
Result sent to music library for extraction of song information
User can accept or discard the song selected by system
System proceed to find the next best candidate based on Song Space grooviness

Fig. 7. Schematic of song recommendation process.

a seed song (a song recommendation feature.) To achieve this feature, the k-th nearest neighbor algorithm is used to select songs that are closest to the seed song. As shown in Fig. 6, the selected candidates will be categorized into four groups with different types of music characteristic. This categorization helps the system to decide which direction to move across the song space based on the user-defined "grooviness" parameter, which defines how a set of songs in the playlist build up or down throughout the playtime. As an illustrative example, a DJ may need to build up the atmosphere by queuing songs such that the BPM increases with time. *SmartDJ* also includes a history playlist which acts as a filter to exclude songs that have been played recently. The final outcome will be loaded into buffer as subsequent track. Figure 7 shows the schematic flow of the song recommendation process.

B. Song-transition feature

Similar to existing DJ software, our *SmartDJ* prototype application also includes a song transition feature that employs the detection and synchronization of tempo, spectral mixing and tonality matching. These three criteria serve as primary filtering components for the automatic song selection feature such that once the subsequent song is selected by the application, mixing occurs when the cue-out point is reached. It is useful to note that spectrum mixing is applied to ensure the frequency content of both songs will not overwhelm each other (within a frequency band) and real-time frequency analysis is performed while a song is playing.



Import songs from library
Perform Beat Detection
Storing of BPM and beat train data associated with song information
Look for first downbeat closest to transition cue point for standby track
Send data to beat detection system to compute BPM difference
Compute necessary speed adjustment
Load results into beat synchronisation engine
Wait for downbeat after transition cue point to trigger transition

Fig. 8. Beat detection and synchronization.

In terms of tempo (or BPM) estimation in an audio recording, several algorithms have been proposed [12], [13]. For beat synchronization between two songs, the speed of song playback is normally altered by DJs. However, it is well-known that changing the speed of the playback will alter the pitch of the song. Therefore, to avoid changing the speed of the playback significantly, we have additionally applied a 3% rule where the tempo difference between two consecutive songs should be within 3% of each other during transition. Therefore, two songs in F Minor that have a BPM of 130 and 131 can be harmonically mixed together since the tempo difference is less than 3%. Figure 8 illustrates the process of beat detection and synchronization.

Spectral mixing is one of the basic skills that DJs apply. This technique involves the reduction of signal energies at a particular range of frequencies while transiting to the next song. In the music arena, DJs apply such EQ blend techniques on a parametric equalizer so as to avoid "overcrowding" of signals at a particular frequency range [14]. To automate the process, one possible approach is via the use of spectrum centroid to determine the center mass of the frequency content of both songs within each frequency band. These bands can be segregated to within 20-400 Hz for bass signal, 400-5.2 kHz for vocal information and signals within the human-sensitive hearing range, and 5.2-20 kHz for high-pitch instruments such as cymbals. Therefore, a real-time spectrum analyzer can be employed to perform spectral analysis and mixing is performed by attenuating the frequency content when there is significant frequency band overlap between the two songs.

The purpose of tonality matching is to ensure that transition can be done smoothly [15]. The concept of harmonic mixing is to ensure two songs will be harmonically compatible with each other based on certain conditions such as having the same key (tonic,) relative Major/Minor key, sub-dominant key (perfect 4th) and dominant key (perfect 5th.) A matrix of weights representing the proximity between tonalities is applied to select subsequent songs with the best harmonic matching possible. This matrix can be derived based on the Camelot wheel chart as shown in Fig. 9. Each sector of the wheel are described by the inner and outer sections which correspond to the major and relative minor keys, respectively. Using this wheel, any adjacent sections should produce a

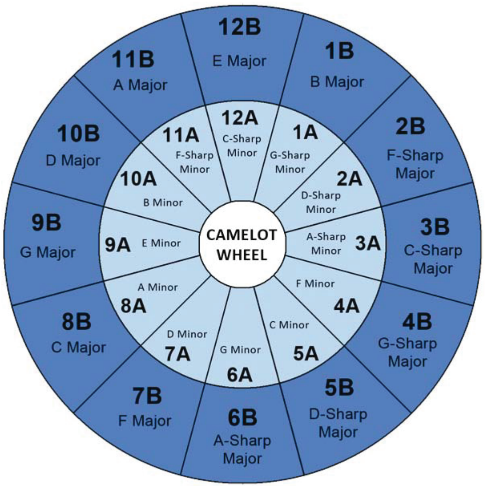


Fig. 9. The Camelot wheel chart. [after [15]].

smooth-sounding mix. Therefore, as an illustrative example, if a song that was in E-minor, another song that is either in E-minor, B-minor, A-minor, G-major, D-major or A-major should be chosen to ensure smooth transition.

V. Conclusion

In this work, we proposed a novel way in which music can be displayed for the user based on similarity of the acoustic features. By translating all songs in the music library onto a two-dimensional feature space, the user can better understand the relationship between the songs, with the distance between each song reflecting its acoustic similarity. We achieve the above by employing low-level acoustic features extracted from raw audio signals and performing dimension reduction using PCA on the feature space. The proposed approach avoids the need to depend on contextual data (such as metadata) and other collaborative filtering methods. With the song-space visualizer, the user can make song choices or allow the system to automate the song selection process given a seed song. The above has been implemented on a MAX/MSP platform and additional DJ features such as song-transition and beat synchronization have also implemented in our *SmartDJ* application to enhance user's listening experience.

VI. Acknowledgment

The authors would like to thank PerMagnus Lindborg from the School of Art, Media and Design, Nanyang Technological University for his insights into music mixing.

References

[1] S.-H. Chen, S.-H. Chen, and R. C. Guido, "Music genre classification algorithm based on dynamic frame analysis and support vector machine," in *Proc. IEEE Int'l Symp. Multimedia*, 2010, pp. 357–361.

[2] D. Byrd and E. Isaacson, "Music representation in a digital music library," in *Proc. IEEE Joint Conf. Digital Libraries*, 2003, pp. 234–236.

[3] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 1980.

[4] D. Liu, L. Lu, and H. Zhang, "Automatic mood detection from acoustic music data," in *Proc. Int'l Symposium Music Information Retrieval*, 2003, pp. 81–87.

[5] P. Davalos, "Automatic music genre classification," Master's thesis, Texas A&M University, 2009. [Online]. Available: http://caritags.com/pedro/

[6] S. Clark, D. Park, and A. Guerard, "Music genre classification using machine learning techniques," Sep. 2012. [Online]. Available: http://web.cs.swarthmore.edu/ meeden/cs81/s12/index.php

[7] T. Langlois and G. Marques, "A music classification method based on timbral features," in *Proc. 10th Int'l Soc. Music Information Retrival Conf.*, 2009, pp. 81–86.

[8] O. Lartillot and P. Toiviainen, "A MATLAB toolbox for musical feature extraction from audio," in *Proc. 10th Int'l Conf. Digital Audio Effects*, 2007.

[9] B. Schuller, J. Dorfner, and G. Rigoll, "Determination of nonprototypical valence and arousal in popular music: Features and performances," *EURASIP Journal Audio, Speech, and Music Processing*, vol. 2010, 2010.

[10] T. Eerola, O. Lartillot, and P. Toiviainen, "Prediction of multidimensional emotional ratings in music from audio using multivariate regression models," in *Proc. 10th Int'l Society Music Information Retrieval Conf.*, 2009, pp. 621–626.

[11] P. Lamere and D. Eck, "Using 3D visualizations to explore and discover music," in *Proc. 8th Int'l Conf. Music Information Retrieval*, 2007, pp. 173–174.

[12] J. Laroche, "Estimating temp, swing and beat locations in audio recordings," in *Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 135–138.

[13] S. Gao and C.-H. Lee, "An adaptive learning approach to music tempo and beat analysis," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Process.*, vol. 4, 2004, pp. 237–240.

[14] C. Cartledge. (2012, Mar.) EQ mixing: critical techniques and theory. [Online]. Available: http://www.djtechtools.com/2012/03/11/eq-critical-dj-techniques-theory/

[15] S. Langford. (2010, Nov.) Live DJ'ing. [Online]. Available: http://www.soundonsound.com/sos/nov10/articles/live-tech-1110.htm