ONE IN THE JUNGLE: DOWNBEAT DETECTION IN HARDCORE, JUNGLE, AND DRUM AND BASS

Jason A. Hockman[1,2], Matthew E.P. Davies[3], and Ichiro Fujinaga[1,2]
[1]Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
[2]Distributed Digital Archives and Libraries (DDMAL), McGill University, Montreal, Canada
[3]Sound and Music Computing Group, INESC TEC, Porto, Portugal
jason.hockman@mail.mcgill.ca, mdavies@inescporto.pt, ich@music.mcgill.ca

ABSTRACT

Hardcore, jungle, and drum and bass (HJDB) are fast-paced electronic dance music genres that often employ resequenced *breakbeats* or drum samples from jazz and funk percussionist solos. We present a style-specific method for downbeat detection specifically designed for HJDB. The presented method combines three forms of metrical information in the prediction of downbeats: low-level onset event information; periodicity information from beat tracking; and high-level information from a regression model trained with classic breakbeats. In an evaluation using 206 HJDB pieces, we demonstrate superior accuracy of our style specific method over four general downbeat detection algorithms. We present this result to motivate the need for style-specific knowledge and techniques for improved downbeat detection.

1. INTRODUCTION

In the early 1990s, affordable sampling technologies (e.g., Akai S900 and Commodore Amiga) and the popularity of rave culture provided the impetus for the creation of three related genres—hardcore, jungle, and drum and bass (HJDB)—unique in their fast tempi and drum sounds, which are mostly derived from samples of percussion solos in 1960s–80s funk and jazz recordings known as *breakbeats*. Since 1990, over 25,000 artists have contributed over 132,000 tracks on almost 6,000 labels. [1] HJDB became so popular in the mid-1990s that it was showcased on BBC's Radio 1 program, "One In The Jungle". Both popular press [1,16] and academic literature [10] have mostly treated HJDB from a sociology/cultural studies perspective, presenting the music within larger contextual issues, e.g., race, drugs, and cultural politics. A notable exception [3], provides tools for automated breakbeat splicing and resequencing.

[1] http://www.rolldabeats.com/stats

In this study, we present a downbeat detection model created with the intention of finding downbeats within music containing breakbeats, and provide a comparison of its performance against four pre-existing algorithms on a database of 206 HJDB excerpts. We view this as a first step in an automated analysis of the musical surface of HJDB from a computational musicology perspective, towards the eventual goal of understanding how individual artists use breakbeats (e.g., slice ordering and pitch adjustment) in modern music.

1.1 Hardcore, Jungle, and Drum and Bass

Hardcore began around 1990, and was the first of the HJDB genres to fully embrace the use of breakbeats. Tracks soon left the 120–130 beats per minute (BPM) house and techno standard and steadily became faster (upwards of 180 BPM), with longer, more intricate drum patterns. The less synth-driven, breakbeat collage art of jungle appeared around 1992. By 1994, many artists abandoned the rhythmic complexity of jungle in favor of simpler rhythms associated with drum and bass. As is the standard workflow in these genres, breakbeats are recorded into a sampler's memory, segmented, and assigned to MIDI note values. HJDB artists create the rhythmic (and sometimes harmonic and melodic) structure of their arrangements using these samples. While hundreds of breakbeats have been employed in HJDB, many artists use a handful of standards such as the "Amen" breakbeat, originally from The Winston's *Amen, Brother* [17].

1.2 Downbeat Detection

The meter of a piece of music implies a counting mechanism for hierarchical stressed and unstressed beats within a measure. A *downbeat* is the first beat within a measure (or if counting beats, the *one*). While the computational task of downbeat detection has received little attention, the related task of beat tracking has received much more attention in recent years [9,13,15]. A possible reason for this imbalance may be related to the increased complexity of the task; prior to extracting downbeats, the estimation of additional subtasks (e.g., onset detection and beat detection) is often required, which can propagate errors into downbeat estimation. Robust downbeat detection would benefit information retrieval tasks such as structural analysis [8], and would facilitate analysis of phrase structure and hypermeter; both useful in improving automated mixing and DSP effects that rely on musically relevant change-point positions. More relevant to our interests, downbeat detection provides key segmentation points that allow for a comparison of HJDB artists' drum usage.

Generalized downbeat detection methods have been proposed in the literature. Goto [11] employs rhythmic template patterns to the output of a drum detection algorithm. In non-percussive music, downbeats are assumed to be present at temporal locations of large spectral change, and are detected through a process of peak-picking spectral frames, grouping of the resultant segments into beats, and a comparison of beats for harmonic change. Davies and Plumbley [5] present a similar approach, in which downbeats are found by selection of beat positions that maximize spectral change. Klapuri et al. [13] extract the temporal evolution of a hidden metrical sequence exhibited in the output of a comb filter bank. The joint-state estimates of the beat, sub-beat, and meter periods are chosen through a first-order Markov process. Papadopoulos and Peeters [14] propose a method for joint estimation of harmonic structure and downbeats using an HMM that models chords and their metrical position. They present an additional method in [15] that also formulates the problem within an HMM framework, in which beat templates are first estimated from the data, and beats are then associated with positions in a measure by reverse Viterbi decoding.

Unlike the aforementioned algorithms, which are generalized for arbitrary musical input, Jehan [12] presents a regression model that predicts downbeat positions based on learning style-specific characteristics from training data containing rhythmic and timbral characteristics akin to those in the testing data. Evaluation is presented in constrained circumstances, in which testing is performed on part of the same song used for training, or on a test song from the same album on which the remaining songs are used as training.

It is our belief that while generalized downbeat detection models will perform well in many circumstances, there remain niche genres that fall outside the scope of these methods [12]. HJDB, while heavily percussive and almost exclusively in 4/4, presents challenges due to its characteristic fast tempo, high note density, non-standard use of harmony and melody, and emphasis on offbeats.

1.3 Motivation

With the exception of [12,15], the above methods rely on general approaches to downbeat detection, and do not infer information about content between estimated downbeats. Our eventual aim is to use detected downbeats towards an estimation of the ordering of drum segments, and their source, i.e., the breakbeat from which the drums were sampled. To do so, our particular application requires an understanding of likely solo percussion performances. We therefore attempt to leverage knowledge of breakbeat timbres and patterns from the 1960s–80s to inform an understanding of three modern genres that utilize them. At the core of the presented model is a top-down support vector regression technique, similar to [12] trained on these building blocks of the music under analysis. Although HJDB artists often resequence segments of breakbeats, the resequenced patterns often reflect knowledge of standard breakbeat patterns. To improve the robustness of this model we incorporate additional stages including beat tracking, and low-level onset detection to focus on kick drum frequencies.

The remainder of this paper is structured as follows: Section 2 outlines our HJDB-specific downbeat detection method. Section 3 presents our evaluation methodology and dataset. Section 4 presents evaluation results and discussion, and Section 5 provides conclusions and future work.

2. METHOD

Our main interest is to determine if an algorithm trained on breakbeat patterns and timbres can find downbeats in modern forms of music that employ them. We began by re-implementing the algorithm as described in [12], with the aim of utilizing it within the full range of HJDB music. Exact parameterization of the model is not provided in [12], so we first tuned our model by optimizing results on examples described in the paper.

2.1 Support Vector Regression for Downbeats

In [12], support vector regression (SVR) is employed to infer likely downbeat positions. Audio is segmented by onset detection or a tatum grid. Each audio segment, S, is associated with a metrical position, t, within a measure with downbeats at t=0, and last sample points before the next downbeat at t=3. We used the LibSVM [2] epsilon-SVR algorithm in MATLAB with a RBF kernel.

To train the regression model, we require a feature matrix F and associated class vector C, which we derive from breakbeats. Two HJDB artists selected 29 breakbeats from several lists of breakbeats commonly used in HJDB. Audio for each breakbeat was trimmed to the portion of the signal containing only the percussion solos. Each breakbeat, β, is then segmented using an eighth-note grid, and a class vector, c_β, is created using the metrical position of each eighth-note segment in a measure.

The feature matrix f_β is comprised of 58 features extracted from each segment in β consisting of: mean segment Mel-frequency spectral coefficients; loudness of the onset (dB) of each segment; maximum loudness (dB) of the onset envelope; and chroma. Segments are then associated with metrical positions in c_β as in [12]. f_β is normalized to have zero-mean and unit variance across each row (all segments). Features are shingled (time-lagged and weighted linearly) [2] to emphasize more recent segments. We then aggregate feature matrices and

[2] http://www.csie.ntu.edu/~cjlin/libsvm/

class vectors across all breakbeats, creating an aggregate feature matrix F and aggregate class vector C. A feature and parameter optimization stage found best results using 40 Mel-frequency spectral coefficients and as in [12], 8 to 16 past segments (equivalent to 1 to 2 bars). Principal Component Analysis (PCA) feature reduction is applied to F to extract the top ten features across all breakbeats. A model is then trained using F and C.

To test the regression model using test audio, A, we require a feature matrix F_A. We first segment the audio using an eighth-note grid created by interpolating the temporal location of beats (we assume beats are found at the quarter-note level), γ, as found by Beatroot [7]. F_A is created similarly to f_β. The PCA model prepared in the training set is applied for feature reduction. We then use the trained model created above with feature matrix F_A to predict class values, C_A, which contain the estimated metrical position of each segment. In [12], the derivative of C_A is used as a detection function from which downbeats are chosen.

While we were able to recreate the examples in [12] using the reimplemented method, training on breakbeats and testing on HJDB music showed that C_A often differed significantly from the idealized output (i.e., pure sawtooth waveform), which resulted in the derivative of C_A being an unreliable detection function on its own.

2.2 Limitations of the Model

We now discuss three conditions that might cause these irregularities in C_A. First, breakbeat patterns are not universal; i.e., one breakbeat may employ a kick drum on beat one and snare drum on beat two, yet another may contain a kick drum on beats one and two, and a snare on the offbeat of two. As a result, C_A may not monotonically increase between downbeats. Second, HJDB artists often re-order slices, which will also cause undesirable output between downbeats. However, breakbeats almost invariably begin with kick drums, and drum-types most associated with downbeats are kick drums. This is also the case for breakbeat usage within HJDB, where artists mostly apply downbeat-preserving transformations, in which segments are reordered and manipulated in such a way to preserve the perception of downbeats. Third, C_A may diverge due to a mismatch in training and testing data. The training data contains percussion-only sections of audio, while the testing data is comprised of excerpts of full HJDB pieces, which may include a variety of transformations (e.g., pitch modifications) to the original breakbeats. To overcome these potential problems, we propose subsequent stages to improve the accuracy of the model: post-processing of C_A (Section 2.3); extraction of additional metrical information—namely, a low-frequency detection function (Section 2.4) and weighting at beat-times (Section 2.5); and information fusion with a final estimation of downbeats by dynamic programming (Section 2.6). An overview of the complete algorithm is presented in Figure 1.

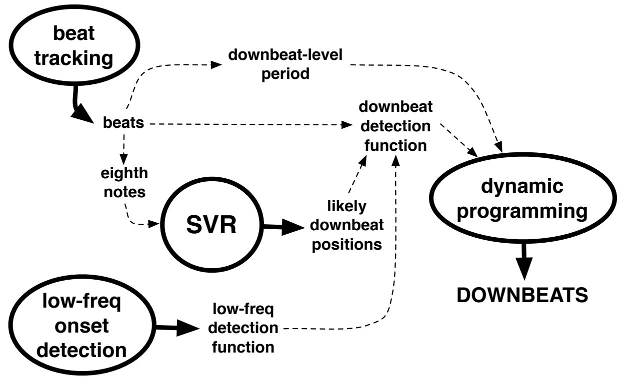

Figure 1: Overview of proposed method. Circles denote stages in the method; solid lines point to variables created in these stages; and dotted lines point to variables created in subsequent steps.

2.3 Regression Output Post-processing

As we are unable to rely solely on the derivative of C_A for an exact location of downbeats, we propose its use in providing a coarse estimation of downbeats. We create likely downbeat position function, E, as the first-order coefficient of the linear regression at each eighth-note position, by applying linear regression of a sliding buffer of eight segments (equivalent to the length of a measure) across C_A. If the eight points of C_A under analysis resemble a positive linear slope, as they do at downbeats, the value of E will be positive. As the buffer shifts, such that it no longer begins on a downbeat (but now includes a downbeat at buffer position 8), the value of E will decrease as it will no longer maintain a positive linear slope. Once the buffer has reached the end of C_A, E is normalized to values between 0 and 1.

2.4 Low-Frequency Onset Detection

The coarseness of E led us to incorporate low-level onset event information related to salience and timing. We introduce a low-frequency onset detection function, L, as follows: As in [6], we segment the input audio into 40 ERB-spaced sub-bands and calculate complex spectral difference across each (with a temporal resolution of 11.6 msec per onset detection function sample). We apply our knowledge of standard usage of basic rock drum kit drum-types (i.e., kick drum, snare drum, and hi-hats) within breakbeats and HJDB music. Since drum types found at downbeats are likely to be kick drums, we focus on lower frequencies and sum the output of the lowest ρ bands to produce L. While the precise number of bands is not critical, we found ρ=5 to provide adequate results.

2.5 Beat-Time Weighting

In Section 2.1, beat time locations, γ, are used to create the eighth-note grid used in the segmentation of the test audio for the SVR model. We also use γ to generate a beat-time weighting, U, for emphasis in L. At γ (here quantized to the resolution of L), U=ω, and otherwise U=1. The precise value of ω is not crucial, however we found ω=1.3 to perform well. To contend with alignment issues of beat times and peaks in L, we additionally weight U=ω at ± 2 detection function samples of γ.

2.6 Information Fusion and Decision

In this stage, we combine low-frequency onset detection function, L, with beat-time weighting, U, and likely downbeat position function, E, to create a final detection function, Θ, used in the determination of downbeat times.

Our motivation in combining these three forms of information is as follows: L provides low-level information pertaining to event location and salience, while E provides informed knowledge of likely downbeat positions based on similarity of the test segment patterns to patterns of drums in the breakbeat training set. The integration of beat-time weighting provides alternate possible downbeat positions that E has either missed or erroneously measured.

As none of these information sources alone is capable of accurate downbeat detection, our hope is that fusing them in a meaningful way will create a hybrid detection function that imparts the key attributes of each, resulting in a more robust detection function from which we will select downbeats. We first interpolate E to match the temporal resolution of L. We then combine L, E, and U:

$$\Theta = (L(1+E)) * U, \tag{1}$$

where $*$ refers to element-wise multiplication.

An example of the usefulness of both E and U in emphasizing peaks of L at likely downbeat positions (and suppressing peaks not likely associated with downbeats) is presented in Figure 2. The top graph shows L (solid line) without scaling by E (dot-dashed line), and annotated downbeat positions (vertical dashed line). The middle graph shows L after scaling by E (solid line). The bottom graph depicts L after scaling by E and U (solid line).

For the final selection of downbeat positions from Θ, we require a peak-finding method capable of finding strong peaks that exist at regular intervals. Dynamic programming (DP) has been shown useful for such purposes in beat detection [9]. We similarly adopt DP to find downbeats within Θ, with a likely downbeat period τ. Given a high probability of 4/4 time signature and steady tempo in HJDB, it is sufficient to estimate τ as 4 times the median of all inter-beat intervals derived from γ.

3. EVALUATION

The aim of our evaluation is to determine the efficacy of our method and four general models on a dataset solely consisting of HJDB. In this section, we present our dataset, algorithms under evaluation, and methodology.

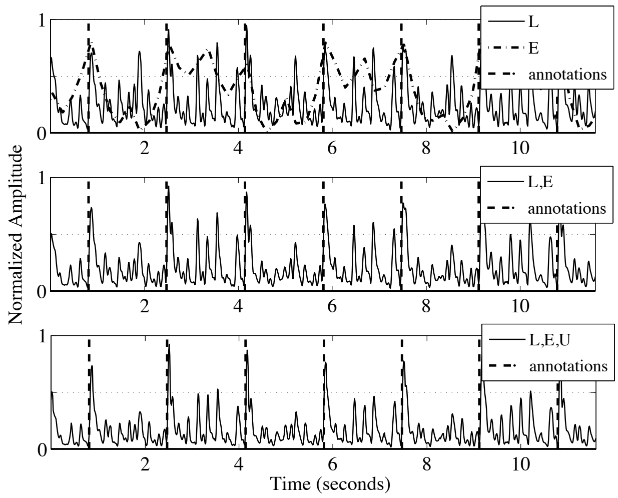

Figure 2: Effect of stages in information fusion: (*top*) L with no scaling, E, and annotations; (*middle*) L scaled by E, and annotations; (*bottom*) L scaled by E and U, and annotations.

3.1 Hardcore, Jungle and Drum and Bass Dataset

Our dataset is comprised of 236 excerpts [3] of between 30 seconds and 2 minutes in duration. Each excerpt was selected from a full-length HJDB piece digitized from its original vinyl format to a 16-bit/44.1kHz WAV file. The pieces span the five years (1990–4) of hardcore's subtle transformation through jungle and into drum and bass.

Well-known, popular HJDB pieces were chosen for inclusion in the dataset. An effort was taken to ensure a wide distribution of artists, styles, and breakbeats used; three professional HJDB DJs were consulted for their opinions. Downbeat annotations were made by a professional drum and bass musician using Sonic Visualiser. [4] 30 excerpts were removed from the test dataset to create a separate parameter tuning dataset used to optimize the parameters in the algorithm presented in Section 2. The remaining 206 excerpts were then used in our evaluation.

3.2 Evaluation Methodology

For evaluation metrics, we chose to modify the continuity-based beat tracking evaluation metrics used in the MIREX 2011 beat-tracking evaluation [4]. The principal difference is that we assess downbeats as the subject of evaluation, rather than beats. Additional modifications include adjustment of the tolerance window threshold, alteration of the possible interpretations of the downbeat to reflect whole beat offsets, and exclusion of the longest continually correct segment metric in [4]. We create a tolerance window of 1/16th note around each annotated downbeat in our dataset (i.e., 6.25% of the inter-annotation-interval). For an estimated downbeat to be correct, it must fulfill three conditions: First, it must be located within the 6.25% tolerance window around the nearest annotation. Second, the previous estimated downbeat must be located

[3] For the track list, see: http://ddmal.music.mcgill.ca/breakscience/dbeat/
[4] http://www.sonicvisualiser.org/

within the 6.25% tolerance window around the previous annotation. Finally, the inter-downbeat-interval must be within 6.25% of the inter-annotation-interval. We then count the total number of correct downbeats and provide a mean accuracy for a given excerpt. Among the various beat offsets allowed by our evaluation measure, our main interest is in the **1** statistic, which indicates how well the estimated downbeats align with annotations. **1** is the mean accuracy across all excerpts. We provide additional statistics, **2**, **3**, and **4**, to quantify errors in downbeat estimations, offset by whole beats. A potential problem for general models is HJDB's fast tempo. We therefore include an additional metric, **1/2x**, which provides an error statistic for estimated downbeats found at the half-tempo rate. **1/2x** is calculated by using the evaluation method above, with the annotations sub-sampled by a factor of two.

3.3 Algorithms Included in Evaluation

Our evaluation focuses on a comparison of the performance of the HJDB-specialized model with four generalized models. We expect this evaluation to be challenging for generalized models due to the lack of harmonic change, fast tempo, and high note density in HJDB music. We compare the following five models: commercial software #1 (CS1); commercial software #2 (CS2); Klapuri et al. (KL) [13]; Davies and Plumbley (MD) [5]; and our HJDB specialized method (HJ). The MD and KL methods are briefly described in Section 1.2. CS1 and CS2 are commercial products from two separate companies. [5] As we do not have access to the methods in CS1 or CS2, we treat them as black boxes.

4. RESULTS AND DISCUSSION

4.1 Parameter-Tuning Set Results

We first compare results of four possible configurations of our model using the 30-excerpt parameter-tuning set, to determine the best system to use in the full evaluation (Section 4.2). Table 1 presents results for these configurations using the **1**, **2**, **3**, and **4** statistics described above. While two of the configurations do not contain beat-time weighting, U, all configurations contain the dynamic programming stage with likely downbeat-level periodicity τ, derived from beats. Informal evaluation of Beatroot's performance on our dataset resulted in an F-measure of 83.0%. The base system (labeled LDF) containing low-frequency detection function, L, performs well, which demonstrates the effectiveness of focusing on kick drum frequencies. Adding either emphasis U (LDF, U) at estimated beat times or estimated likely downbeat detection function E (LDF, E) has a similar positive effect. Adding both U and E has a further positive effect, indicating independence between these features. In addition, errors in statistics **2**, **3**, and **4** in either LDF, U or LDF, E are reduced by addition of the other features—e.g., the 6% error found in LDF,E in the **4** statistic is reduced to 3.3%. Similarly, the 2.8% error found in the LDF, U on the **2** statistic is reduced to 0.6%. Addition of either or both emphasis U or E results in an improvement in accuracy over LDF alone, and a reduction in error rates **2**, **3**, and **4**.

[5] of which one was a beta version

	1	**2**	**3**	**4**
LDF	72.8	3.7	3.4	6.4
LDF, E	79.3	0.8	9.6	6.0
LDF, U	79.9	2.8	**2.8**	4.8
LDF, U, E	**83.4**	**0.6**	3.1	**3.3**

Table 1: Accuracy measure **1** and error metrics **2**, **3**, **4** (in percentages) for four configurations of the presented system using the parameter-tuning dataset. Bold scores denote highest accuracy in **1**, and lowest error in **2**, **3**, **4**.

4.2 HJDB Evaluation Results

Evaluation performance for the five compared methods is displayed in Table 2. Our specialized algorithm HJ (using the LDF, U, E configuration) performs best in the **1** statistic. In addition, HJ achieves the smallest **2** and **1/2x** error statistics (with a low **4** error rate), which when coupled with high **1** performance, is seen rather favorably.

	1	**2**	**3**	**4**	**1/2x**
CS1	38.5	2.8	**4.0**	4.2	2.8
CS2	7.4	11.7	9.5	6.7	1.1
KL	51.3	2.8	9.6	**0.2**	3.0
MD	29.3	4.7	5.5	3.0	1.2
HJ	**74.7**	**2.3**	5.8	2.0	**0.0**

Table 2: Accuracy measure **1** and error metrics **2**, **3**, **4**, **1/2x** (in percentages) for the five models under evaluation using HJDB test dataset. Bold scores denote highest accuracy in **1**, and lowest error in **2**, **3**, **4**, **1/2x**.

When a model finds a downbeat on beats two or four in HJDB music, it is likely to indicate a preference for high-energy note events such as snares (often played on beats two and four). All models have some degree of error reported in the **3** metric, possibly due to similarities in breakbeat drum patterns starting on beats one and three, which results in a confusion of phrase boundaries at these positions. Surprisingly, none of the models displayed an affinity for the **1/2x** metric that our intuition led us to believe generalized models would find more favorable.

4.3 Discussion

While our specialized method outperformed the generalized models, results should be examined with the understanding that only our approach had access to the parameter-tuning set used to adjust parameters of the SVR algorithm. While this may make the comparison somewhat imbalanced, our model is the only algorithm necessitating such parametric tuning, as the other models are general approaches. We have incorporated specific attributes of HJDB music in a model used for its analysis: information about timbre, pitch, and loudness of segments; knowledge of likely patterns; and emphasis on kick drum events and potential downbeat candidates at beat locations. Intuition tells us that the model in its present configuration may not perform as well in a generalized evaluation or niche genres excluding breakbeats, as downbeats in these datasets may not be conveyed similarly.

5. CONCLUSIONS AND FUTURE WORK

We have presented a style-specific model for finding downbeats in music that we applied to hardcore, jungle and drum and bass. At the core of our approach is a learning technique trained on classic breakbeats that form the rhythmic and timbral basis of these musical styles. We expanded this model to incorporate information related to likely onsets in low-frequency bands and beat tracking. Through fusion of these complementary information sources we create a downbeat detection function from which we infer downbeats using dynamic programming.

Evaluation of our style-specific model with generalized downbeat detection methods demonstrates a wide gap in performance. This not only highlights the efficacy of our approach in the confines of HJDB, but also provides further evidence towards the style-specific nature of downbeat detection. We consider the latter conclusion more critical, and expect our method to be less effective in music without breakbeats, and in music in which downbeats are conveyed by chord changes.

In building our model we have attempted to keep as many components as general as possible, leaving the training of the SVR as the sole part explicitly style-adapted to HJDB. In this way, we believe our approach could be readily adapted to other music styles through style-specific training of the SVR. This strategy will form a key component of our future work; both by training multiple models on different styles and investigating methods for automatic selection between these models. We believe the most profitable future advances in downbeat detection will be style-specific, rather than generalized models. Within the domain of HJDB music, we intend to harness the knowledge of downbeats to explore the relationships between the musical corpus and specific breakbeats amid a large-scale study of the genres.

6. ACKNOWLEDGEMENTS

This work is partially funded by the Social Sciences and Humanities Research Council of Canada, the ERDF through the Programme COMPETE and by the Portuguese Government through FCT—Foundation for Science and Technology, project ref. PTDC/EAT-MMU/112255/2009. The authors would like to thank Conor O'Dwyer (*Code*), Jason Chatzilias (*0=0*), and Daniel Lajoie (*ESB*) for their contributions and fruitful discussion.

7. REFERENCES

[1] B. Belle-Fortune, All crews, Vision, London, 2004.

[2] A. Z. Broder, S. C. Glassman, M. Manasse, and G. Zweig, "Syntactic clustering of the web." *J. of Comp. Networks*, Vol. 29, No. 8, pp. 1157–66, 1997.

[3] N. Collins, *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems.* PhD. diss., Cambridge University, 2006.

[4] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms." Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06, 2009.

[5] M. E. P. Davies and M. D. Plumbley, "A spectral difference approach to downbeat extraction in musical audio." In *Proc. of EUSIPCO,* 2006.

[6] M. E. P. Davies, M. D. Plumbley, and D. Eck, "Towards a musical beat emphasis function." In *Proc. of WASPAA,* pp. 61–4, 2009.

[7] S. Dixon, "Evaluation of the audio beat tracking system BeatRoot." *JNMR*, Vol. 36, No. 1, pp. 39–50, 2007.

[8] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterization of music via rhythmic patterns." In *Proc. of 5th ISMIR Conf.,* pp. 509–16, 2004.

[9] D. P. W. Ellis, "Beat tracking by dynamic programming." *JNMR*, Vol. 36, No. 1, pp. 51–60, 2007.

[10] E. Ferrigno, *Technologies of emotion: Creating and performing drum 'n' bass.* PhD. diss., Wesleyan University, 2008.

[11] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds." *JNMR*, Vol. 30, No. 2, pp. 159–71, 2001.

[12] T. Jehan, *Creating music by listening.* PhD. diss., Massachusetts Institute of Technology, 2005.

[13] A. P. Klapuri, A. J. Eronen and J. T. Astola, "Analysis of the meter of acoustic musical signals." *IEEE TASLP,* Vol. 14, No. 1, pp. 342–55, 2006.

[14] H. Papadopoulos and G. Peeters, "Joint estimation of chords and downbeats from an audio signal." *IEEE TASLP,* Vol. 19, No. 1, pp. 138–52, 2010.

[15] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation." *IEEE TASLP,* Vol. 19, No. 6, pp. 1754–69, 2011.

[16] S. Reynolds, Energy flash: A journey through rave music and dance culture (2nd Ed.). Picador, London, 2008.

[17] "Seven seconds of fire." *The Economist,* pp. 145–6, 17th December 2011.