AUTOMATED RHYTHMIC TRANSFORMATION OF MUSICAL AUDIO

Jason A. Hockman[*]
CIRMMT
Schulich School of Music, McGill University,
Montréal, Canada
jason.hockman@mail.mcgill.ca

Juan P. Bello,
Music Technology
New York University,
New York, USA
jpbello@nyu.edu

Matthew E. P. Davies and Mark D. Plumbley,[†]
Centre for Digital Music, Dept. of Electronic Engineering
Queen Mary University of London, United Kingdom
matthew.davies@elec.qmul.ac.uk

ABSTRACT

Time-scale transformations of audio signals have traditionally relied exclusively upon manipulations of tempo. We present a novel technique for automatic mixing and synchronization between two musical signals. In this transformation, the original signal assumes the tempo, meter, and rhythmic structure of the model signal, while the extracted downbeats and salient intra-measure infrastructure of the original are maintained.

1. INTRODUCTION

Rhythm is comprised of regular or irregular pulses around the tactus that when taken together, provide a sense of movement within a piece. Automated rhythmic extraction [1] seeks to computationally detect this structure within a musical recording, first by localizing measure boundaries, and then by characterizing intra-measure events within these boundaries. In this paper, we incorporate rhythmic extraction into a framework for the automated rhythmic synchronization of two musical signals.

Manual transformations of this type are time-intensive and usually performed with the use of an audio editor. First, the original waveform is sliced at rhythmically relevant points. Second, each segment is relocated to a new position based on the rhythmic pattern of a model signal. Finally, each segment is individually time-scaled to maintain continuity. The popularization of loop-based music production and mixing applications has provided the impetus for several softwares, such as *Ableton Live* [2] and *FX-Pansion GURU* [3], to offer automated tools for rhythmic contour adjustment. While these methods are designed for use with monophonic or *limited* polyphonic audio, no technique currently addresses the difficulties inherent within transformations of more complex musical audio.

Time-scaling is often utilized in an attempt towards automated synchronization. However, if the ratio between the new and original tempi (scalar ratio) is increased or reduced by too great a factor, then transient regions may become smeared or artificial, resulting in a perceptual loss of audio quality. As a secondary consequence, rhythmic structures within the original and model pieces are not guaranteed to coincide, as these structures are independently preserved and proportionally affected during time-scaling.

A partial solution is suggested by Duxbury [4], followed by Ravelli et al [5], which presents an *adaptive phase vocoder* time-scaling approach that preserves attacks following detected onsets via incorporation of a local scaling factor and phase-locking process. The result of this adaptive method is a dramatic increase in the acceptable range of global scalar ratios and thus an increase in the range of tempi for a given transformation.

This multi-scalar approach lends itself to the adjustment of swing within musical excerpt. In Gouyon et al [6] both tactus locations and subdivisions are automatically extracted, creating eighth-note sections which are then modified by an input-defined syncopation. Although this process creates a metrically-informed representation of an audio signal, the dynamism of the transform is limited by the lack of a more precise rhythmic representation.

In a technique conceptually similar to ours, Ravelli et al [7] perform an automatic rhythm modification of percussive audio, in which note events are extracted by discrete onset detection and classified as either *low*, *mid*, or *high*-pitched sounds. A pattern matching algorithm then selects a best-path sequence to match segments from the first signal to those of the second. Transient regions within the initial portion of each segment are preserved as in [5]. Although the method is well-suited for percussive audio, the extraction of discrete segment times and categorization of slices make it unsuitable for polyphonic audio, where such clear demarcation is neither guaranteed, nor likely.

Another related method is the *cross-matching* technique presented by Jehan [8], in which synchronization of musical audio signals is achieved through two stages. First, beat tracking and downbeat detection is performed to determine the tempo and measure boundaries of each signal. The two signals are then aligned by time-scaling the regions between detected downbeats. The resulting transformation provides a metrically aligned signal, however no attempt is made to adjust the timing of intra-measure components that comprise the signal.

Our modification model seeks not only to combine the downbeat alignment of [8] with the aforementioned phase-locking transient preservation of [5], but further, to incorporate intra-measure rhythmic structural changes towards a more intricate matching of

[*] research performed while a student at New York University

[†] This research has been partially funded by EPSRC grant GR/S82213/01.

two *arbitrary* polyphonic signals.

This automatic rhythmic synchronization is achieved through the following series of signal processing steps: First, for each signal we extract beat locations and downbeats, from which bar length predominant rhythmic patterns are generated. Next, rhythmic pattern events are assessed for both signals, and a comparative process is undertaken to identify structurally relevant intra-measure components of the original signal that coincide with the model signal. These points are then realigned to those of the model. Finally, regions within the structurally relevant points of the original signal are adaptively time-scaled to fit the length of those within the model, while preserving transient regions at the presence of detected onsets in each segment.

The remainder of this paper is structured as follows. In §2 we outline the rhythmic analysis. In §3 we present the rhythmic pattern matching approach and an overview of the time-scaling algorithm in §4. Results are given in §5 with discussion in §6.

2. RHYTHMIC SEGMENTATION

To modify the rhythmic properties of a musical audio signal we must first localize the start points of musical events. Towards this end, the process of discrete onset detection [9] might first be considered appropriate, as it has been used both within time-scaling [5] and existing rhythm modification techniques [6, 8]. Onset detection provides a *temporal* segmentation of the input signal. However, in our method, it is not mandatory to detect every onset. Further, an accurate representation of discrete onsets is improbable within musical audio due a random density of note events in the input signal. Instead, we wish only to locate those onsets that contribute to an underlying rhythmic pattern present in the input. Our aim then is to produce a *rhythmic* segmentation, for which we turn to existing work within beat tracking, downbeat detection and estimation of predominant rhythm patterns. We now provide a brief overview of this analysis (for a complete derivation see [10]).

The first stage in our analysis is to transform the incoming audio signal into a mid-level representation more suited to rhythmic analysis. In this case the mid-level signal should exhibit peaks at note onset locations where higher peaks represent metrically stronger events. For this purpose we use the complex spectral difference onset detection function $\Gamma(m)$ [9], calculated by measuring the short-term spectral change in both the magnitude and phase spectra of the input. The detection function (DF), with samples m, has a time resolution of 11.6ms.

The next stage is to identify beat locations γ_b. Again we make use of our existing beat tracking algorithm [10], however any beat tracking algorithm could be employed.

The beats are extracted from a two-stage process. First, the beat period (the time between beats) is given by comb filtering the autocorrelation of $\Gamma(m)$. Secondly, the beat alignment is estimated by comb filtering $\Gamma(m)$ given the beat period.

Downbeats γ_d are extracted by measuring the Kullback-Leibler divergence between the power spectra of band-limited consecutive beat-synchronous frames. The beat transitions which lead to most spectral change are identified as the downbeats.

To extract the predominant rhythmic pattern $\Gamma_{\mathbf{P}}(m)$, we reanalyze the onset detection function $\Gamma(m)$ given the beats γ_b and downbeats γ_d. We then partition the onset detection function $\Gamma(m)$ into individual bars, $\Gamma_d(m)$

$$\Gamma_d(m) = \Gamma(m) \quad \gamma_d \leq m < \gamma_{d+1}. \tag{1}$$

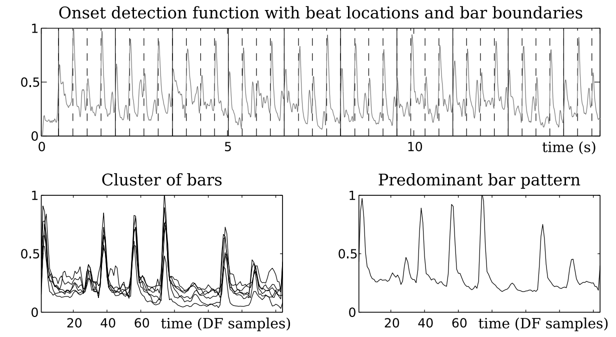

Figure 1: Overview of rhythmic analysis (top) onset detection function $\Gamma(m)$ with downbeats γ_d (solid lines) and beat locations γ_b (dashed lines); (bottom-left) cluster of bar-length patterns; (bottom-right) predominant rhythmic pattern $\Gamma_{\mathbf{P}}(m)$.

To account for variable bar lengths, we follow the approach of Dixon et al [11] and resample each onset detection function bar $\Gamma_d(m)$ to have normalized length, L (where L=144 detection function (DF) samples). The $\Gamma_d(m)$ are then clustered together using k-means (where k=3), and the predominant pattern $\Gamma_{\mathbf{P}}(m)$ is extracted as the temporal average of the largest cluster.

To find the rhythmic pattern events p within the bar length pattern, we peak-pick $\Gamma_{\mathbf{P}}(m)$ using the approach from [9]. The final stage is then to associate these pattern events with the extracted downbeats, γ_d, which we achieve through linear interpolation,

$$\gamma_{d,p} = \gamma_d + p\left(\frac{\gamma_{d+1} - \gamma_d}{L}\right) \quad d = 1, \ldots, D-1. \tag{2}$$

An overview of the rhythmic analysis is shown in figure 1.

3. RHYTHMIC PATTERN MATCHING

Our eventual aim is to associate the rhythmic pattern events of an original signal A to those of a model signal B, and adaptively scale the regions demarcated by these pattern points (*rhythmic slices*). As such, the aforementioned rhythmic segmentation is performed on both A and B. As the implementation is designed to operate upon arbitrary signals, there is often an unbalanced number of intra-measure pattern segments between A and B. Thus, prior to modification of the temporal locations of pattern points, it is imperative to address the number of intra-measure pattern events within the two signals. If, for example, a given measure of A contains n more rhythm pattern events than does B, and no modifications are made to absolve this disparity, the n pattern points will become the first series of events within the subsequent measure. This undesirable *wrapping* effect will then cause the transformed bars of A to have a different length than those of the target signal B, as the downbeats of the original sequence will not be preserved.

Validity of a metrically synchronized transformation is heightened by extending the rhythmic pattern to include metrical and tactus-level events, as the output should ideally contain a pulse similar to the model. Beat locations are determined as periodic, strong events within the detection function, and as such, it is highly probable that beat times are already embedded within the rhythmic pattern. Incorporation of beat times also permits us to remove

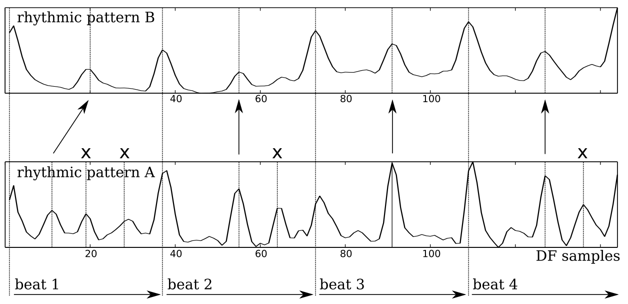

Figure 2: Rhythmic pattern matching. Beat events are always preserved. Where there are more pattern events per beat in one signal compared to the other, the weakest excess events are removed.

the metrical component from the matching stage, simplifying our model by instead focusing our analysis upon the beat level and *pattern events per beat*. Finally, to assure downbeat synchronization, audio signals are truncated such that both A and B begin with the start of a bar and terminate with the end of a bar. We modify the notation from equation (2) to refer to the individual inter-beat segments. For each beat in A there will be Q pattern events per beat, where the q^{th} pattern event in the b^{th} beat of A is referred to as $A_{b,q}$, and similarly, given R pattern events per beat in B, the r^{th} pattern event in the b^{th} beat of B is notated as $B_{b,r}$. The beat locations exist as the first pattern event per beat, e.g. $A_{b,1}$.

As mentioned, within each beat b in A and B, there is no guarantee that there will be an equal number of pattern events (i.e. $R \neq Q$). We contend with this mismatch by selectively ignoring excess pattern events in terms of their relative strength. Our measure of strength is determined by sampling the onset detection function $\Gamma(m)$, with the excess number of pattern events per beat at each pattern event $\gamma_{b,p}$. The *weakest* events p^{x} are then iteratively removed once located using

$$p^{\text{x}} = \arg\min_{p} \Gamma(\gamma_{b,p}). \tag{3}$$

In figure 2, we observe the process of removing the weakest pattern positions from each inter-beat interval, and subsequent alignment of residual salient events. Beat locations are preserved without exception, as shown by the vertical lines which are present in both patterns A and B, while excess pattern events, calculated from equation (3) are marked as terminating in 'x'.

Once the pattern events per beat have been balanced we change our notation to reflect P pattern events per beat in both A and B where we now refer to $A_{b,p}$ and $B_{b,p}$ and proceed to the time-scaling stage, to implement the rhythmic transform.

4. TIME-SCALING

We now iteratively extract each rhythmic segment as defined by the rhythmic pattern events in A, and time-scale it to match the duration of the corresponding segment in B. To perform this time-scaling, we utilize the adaptive phase-locking method of Ravelli et al [5].

Temporal masking is performed as a pre-processing step to separate the transient and non-transient regions of the input signal. A discrete onset detection of the input audio signal is performed, and transient regions are defined as the first 1/3rd of each inter-onset-interval (IOI), under the condition that the minimum transient region is always greater than 11.6ms, and always less than 50ms. Time-scaling is then only applied during the steady-state regions of the input, to prevent smearing of the transients.

Given the rhythmic segmentation described in §2 we do not require a discrete onset detection stage, as in [5]. Each rhythmic slice has a defined onset $A_{b,p}$ and a defined offset, found as either the next pattern event for the current beat $A_{b,p+1}$, or the next beat $A_{b+1,1}$. Within each slice we calculate linear scaling factor F_1 as the ratio of the durations between current pattern events in A and those in B

$$F_1 = \begin{cases} \frac{A_{b,p+1} - A_{b,p}}{B_{b,p+1} - B_{b,p}} & p < P \\ \\ \frac{A_{b+1,1} - A_{b,p}}{B_{b+1,1} - B_{b,p}} & \text{otherwise} \end{cases} \tag{4}$$

The non-linear scaling factor F_2 which accounts for the temporal masking is calculated as

$$F_2 = \frac{W_{\text{nt}}}{W_{\text{t}}}(F_1 - 1) + F_1 \tag{5}$$

where W_{nt} is the width of the non-transient region, and W_{t} is the width of the transient region. Each time-scaled rhythmic slice is then concatenated with the previous slices to form the eventual transformed signal.

5. RESULTS

Since the rhythmic pre-processing and time-scaling aspects have been evaluated previously [10, 5] we only evaluate the rhythmic transformation aspect. Our evaluation centers on the task of automatic rhythmic genre classification. We explore whether our rhythm transform technique can force the *misclassification* of musical genre, e.g. transforming a Samba into a QuickStep and determining whether the output can be classified as the input genre (Samba) or target (QuickStep).

We employ an existing technique [12] which uses a bar-length periodicity pattern to characterize the rhythmic properties of the input signal so as to maximize the similarity between excerpts of the same genre but minimize the cross-similarity between different genres. The feature calculated is the autocorrelation function of the onset detection function (described in §2). This is truncated at the bar-level periodicity (defined by the tempo and the number of beats per bar) and resampled to a fixed duration (144 DF samples [11]). Each excerpt in the test database is given a genre label and passed to the open source data mining software, WEKA [13] to perform the classification. For further details see [12].

As training data we use an existing database comprised of 523 ballroom dance excerpts of arbitrary polyphony across 6 genres: Jive, QuickStep, Tango, Samba, ChaCha and Rumba. This constitutes a subset of the full 698 test database [11] where the 3/4 time-signature Waltz and Viennese-Waltz categories (those which are beyond the scope of our techniques) have been removed. From the training set we extract a subset of 120 (20 x 6 genres) test excerpts for which we annotate beats and downbeats. For each genre in turn, we pick a random input signal and a random target signal. We perform 20 transformations from each of the 30 distinct genre pairs giving 600 total transformations. To prevent any potential inaccuracies in the rhythmic pre-processing we provide the transformation algorithm with the metrical annotations.

Evaluation Data	Accuracy (%)
CFV (training set)	87.0
$X_{B \to A} \to G_A$	51.5
$X_{B \to A} \to G_B$	13.5

Table 1: Summary of classification accuracy. CFV – 10 fold cross-validation on the 523 excerpt training data. $X_{B \to A}$ – transformed signals, G_A – target genre,G_B – input genre.

Genre	Jive	Quick	Tango	Samba	Cha	Rumba
Jive	n/a	19	12	8	13	5
Quick	7	n/a	10	3	1	2
Tango	7	14	n/a	14	10	7
Samba	13	16	5	n/a	15	8
Cha	12	9	14	17	n/a	9
Rumba	13	6	9	14	16	n/a
Total %	53	64	50	56	55	31

Table 2: Rhythmic genre classification results ($X_{B \to A} \to G_{\mathbf{A}}$) for 1-NN classifier. In each case 20 transformations are attempted. Quick is short for QuickStep.

We label genre of the transformed signals ($X_{B \to A}$) in two different ways: first with those of the target genres (G_A), and second (as a control), with the labels of the input signal genres (G_B). We then pass this data to WEKA [13] and use the nearest neighbor classifier (1-NN) to perform the classification. To gauge the performance of the genre classification method without rhythmic transformation, we perform a 10-fold cross-validation on the 523 excerpt training data. The overall results are summarized in Table 1 with a genre-dependent breakdown in Table 2.

The classification accuracy of the transformed signals with the target genre labels (51.5%) is considerably higher than when using the input genre labels (13.5%), i.e. the transformed signals are closer to the target genres than those before they were transformed. The 51.5% accuracy is still well below the 87.0% performance of the genre classification on the unmodified data. Examination of Table 2 reveals the hardest target genre to be Rumba, with QuickStep as the most suitable target. This latter finding is most likely due to the *swing* inherent within the QuickStep genre, which was not as prevalent in other genres tested.

6. DISCUSSION

The challenge for our method lies in the preliminary rhythmic analysis. In cases where the rhythmic analysis is successful, the resulting transformations sound surprisingly coherent and intentional[1], not least in part to the high quality of the time-scaled audio. This is especially encouraging given the implicit complexity of the task. If, however, the initially extracted beat times are inaccurate, then the subsequent analysis is likely to fail. If downbeats are calculated incorrectly, a coherent rhythmic transformation may still be produced, however the phase difference between signals will become apparent when mixed together.

Both problems could be remedied by a semi-automatic version of the transform, which would allow a user to manually correct beat times and downbeats, leaving the remaining processing to be performed automatically. Used as a post-processing tool for music production, this would not impose a significant burden on the user.

We also intend to investigate a real-time version of our rhythmic transformation algorithm. Although our current implementation processes the musical excerpts in an offline fashion, a causal version of the rhythmic pattern analysis has been implemented which can be used to predict future rhythmic segmentation points within a consistent underlying rhythmic pattern [10]. As well as potential applications within production and composition, a real-time version could provide transformation of recorded accompaniment to follow the rhythmic structure of a live musical performance.

[1] `www.music.mcgill.ca/∼hockman/projects/ARTMA`

7. REFERENCES

[1] F. Gouyon and S. Dixon, "A review of automatic rhythm description systems," *Computer Music Journal*, vol. 29, no. 1, pp. 34–54, 2005.

[2] Ableton, "Live," http://www.ableton.com, 2008.

[3] FXPansion, "GURU," http://www.fxpansion.com, 2008.

[4] C. Duxbury, *Signal models for polyphonic music*, Ph.D. thesis, Queen Mary, University of London, 2004.

[5] E. Ravelli, J. P. Bello, and M. B. Sandler, "Fast implementation for non-linear time-scaling of stereo audio signals," in *Proceedings of DAFx*, Madrid, Spain, 2005, pp. 182–185.

[6] F. Gouyon, L. Fabig, and J. Bonada, "Rhythmic expressiveness transformation of audio recordings: swing modifications," in *Proceedings of DAFx*, London, United Kingdom, 2003, pp. 94–99.

[7] E. Ravelli, J. P. Bello, and M. B. Sandler, "Automatic rhythm modification of drum loops," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 228–231, 2007.

[8] T. Jehan, *Creating music by listening*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[9] J. P. Bello, C. Duxbury, M. E. Davies, and M. B. Sandler, "On the use of phase and energy for musical onset detection in the complex domain," *IEEE Signal Processing Letters*, vol. 11, no. 6, pp. 553–556, 2004.

[10] M. E. P. Davies, *Towards automatic rhythmic accompaniment*, Ph.D. thesis, Queen Mary, University of London, 2007.

[11] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proceedings of ISMIR*, Barcelona, Spain, 2004, pp. 509–517.

[12] M. E. P. Davies and M. D. Plumbley, "Exploring the effect of rhythmic style classification on automatic tempo estimation," in *Proceedings of EUSIPCO*, 2008, To appear.

[13] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, 2nd edition, 2005.