LOCATING RHYTHMIC PATTERNS IN MUSIC RECORDINGS USING HIDDEN MARKOV MODELS

Iasonas Antonopoulos, Aggelos Pikrakis and Sergios Theodoridis

University of Athens
Department of Informatics & Telecommunications
Panepistimioupolis, TYPA Buildings, Ilisia, 15784, Athens, Greece

ABSTRACT

This work addresses the problem of locating *rhythmic patterns* in music recordings. During the feature extraction stage, a short-term processing technique is applied, in order to detect *significant* changes in the spectral and energy evolution of the music signal. The detected changes are in turn treated as onsets of events and a sequence of inter-onset intervals is extracted. The resulting sequence is long-term segmented and is fed as input to a Hidden Markov Model (HMM) which models a predefined *rhythmic pattern*. An *enhanced Viterbi* algorithm is proposed, that extracts a best-state sequence, which determines the pattern location boundaries. Our method was tested on a set of music recordings of music meter $\frac{2}{4}$, $\frac{3}{4}$, $\frac{7}{8}$ and $\frac{9}{8}$ and steady tempo. The proposed method exhibits excellent precision (100%) over pattern locations and a recall ranging from $\sim 34\%$ up to $\sim 74\%$ depending on the music genre.

Index Terms— *rhythmic patterns*, Hidden Markov Models, *Enhanced Viterbi* algorithm

1. INTRODUCTION

Automatic location of *rhythmic patterns*, is a difficult yet highly desirable task for applications in the context of music information retrieval. Locating *rhythmic patterns* is associated with finding the positions of beats and meter in audio files. Efforts, so far, have been focused on the beat tracking task for western music meters, such as $\frac{4}{4}$, some of which are highlighted below.

Goto et al. presented two systems, the first detecting drum sounds [1] and the second chord changes [2], in order to determine the temporal positions of quarter and half notes, while using a multi-agent expert system for the prediction of the beat. Scheirer [3] detected the changes in the amplitude envelopes of six subbands and tracked the beat and beat phase from the maximum output of 150 comb filters, representing possible tempo values. Sethares et al. [4], using low level features, presented two approaches for beat tracking in musical performances, one based on Bayesian decision framework while the other implemented a gradient strategy. Klapuri et al. [5] demonstrated beat *and* meter tracking using the combination of the subband decomposition of [2, 3] for onset extraction and three *HMM*s for the estimation of tempo and meter in three metrical levels. Finally, Dixon [6] extracted the locations of *rhythmic patterns*, for western type dances, by providing the location of the first beat, in an attempt to characterize music via their *rhythmic patterns*.

In this work, we use *HMM*s to locate *rhythmic patterns* in music recordings by employing an *enhanced Viterbi* algorithm. The proposed method operates on the assumption that the music meter and a rough estimate of the tempo are known. In our work the tempo estimator, proposed in [7], is used to provide this information for each recording. It is assumed that tempo remains approximately constant throughout the recordings. To our knowledge, this is the first time that the problem of beat and meter tracking is addressed in the context of complex meters *without* any prior knowledge of any pattern location. Our focus is on complex meters, such as $\frac{7}{8}$, $\frac{9}{8}$, which appear in eastern folk music. Furthermore, we studied patterns for music meter $\frac{2}{4}$, $\frac{3}{4}$ which are also frequently encountered in traditional dances. In our approach, a *rhythmic pattern* is modeled by means of a Hidden Markov Model, where each event of the pattern corresponds to a *HMM* state. In order to locate occurrences of such a pattern in a recording, the *HMM* is fed with overlapping segments of the feature sequence that has been extracted from the audio data and at a next step the extracted patterns (if any) are connected creating a chain of *rhythmic patterns*.

The paper is structured as follows: Section 2 focuses on feature extraction, Section 3 analyzes the *HMM* modeling of a *rhythmic pattern* and also presents the *enhanced Viterbi* algorithm. Section 4 evaluates the proposed system, and finally future research priorities are drawn in Section 5.

2. FEATURE EXTRACTION

A short-term processing technique is first applied in order to detect *significant changes* in the audio, marking candidate beat locations. Each short term frame ($\simeq$ 93 *msecs* long, with $\simeq$81.3 *msecs* overlap between successive frames) is multiplied by a Hamming window and is given as input to a Mel-

scale filter bank [7]. The center frequencies of the filters coincide with the frequencies of whole tones on a chromatic scale, starting from 110Hz and moving up to $\simeq$ 12.KHz, resulting into 42 filters, which cover approximately seven octaves.

Let $melStd(n)$, be the smoothed and normalized standard deviation of the filter bank outputs and $dEner(n)$, $n = 1, \ldots, N$, the first derivative of signal energy for each frame, where N is the number of short-term frames. A peak picking algorithm selects those maxima with frame index m for which $stdMel(m) > stdMel(k), \forall k \in [k_1, k_2]$ and m being the center of the $[k_1, k_2]$ interval. Let also i be the number of frame for which $dEner(i) > dEner(k), \forall k \in [k_1, k_2]$ with i being, also, the center of $[k_1, k_2]$. Our goal is to select those frames whose frames indices i, m coincide within a threshold value. For our experiments this value was chosen to be equal to $0.1secs$. These frames are selected to indicate *onsets* and we choose the respective value of m to indicate the onsets. The value of $k_2 - k_1$ depends on the recording tempo. For the tempo values of the examined corpus, ranging from 89-385 beats per minute, *bpm* (as shown in Table 2), $k_2 - k_1$ was set equal to the time equivalent of $\frac{tempo}{2}$ for recordings with $tempo \leq 150$*bmp* and $tempo$ all others with tempo $tempo \geq 150$*bpm*, respectively.

The physical meaning of these onsets is that they signal the beginning of an event, i.e., a *significant change* in terms of spectrum (*melStd*) and energy (*dEner*). Each event will, therefore, have an onset and an associated time duration. Let, m_k, m_{k+1} be two consecutive selected onsets. Then $m_k < m_{k+1}$ and $m_{k+1} - m_k$ is the so-called inter-onset-interval (*IOI*). The feature sequence R that is given as input to the *HMM*, is formed by zeroing the *melStd* of all frames, except those that correspond to onsets, i.e.,

$$R = \{O_{z_0}, a(m_1), O_{z_1}, a(m_2), \ldots, O_{z_{M-1}}, a(m_M), O_{z_M}\},$$

where O_{z_j} stands for z_j successive zeros. As a result, $a(m_j)$ is the amplitude of the j-th onset and O_{z_j} it's associated duration.

At a next step, sequence R is long-term segmented by means of a moving window technique. The length and step of the moving window depend upon the expected length, P_L, of the *rhythmic pattern*. P_L is computed from the music meter and *tempo* of the recording. For example, for a *rhythmic pattern* of music meter $\frac{7}{8}$ and *tempo* 250*bpm* the pattern length $P_L \simeq 1.7sec$. In our experiments the length and step of the moving window are set equal to $\frac{5*P_L}{2}$ and $\frac{P_L}{2}$, respectively. Each long-term segment is processed separately by the *HMM* that models the pattern.

3. MODELING RHYTHMIC PATTERNS BY MEANS OF HMMS

Rhythmic structures can be considered to build upon fundamental *rhythmic patterns*. For example, recordings of music meter $\frac{7}{8}$ with *tempo* ranging from 200 - 290*bpm*, as is the case with a number of traditional music genres, are perceived as a sequence of *rhythmic patterns* of [$dotted\ quarter\ note$ - $quarter\ note$ - $quarter\ note$]. This is also consistent with the performance of the accompaniment instruments and the singing voice in such recordings. To construct the corresponding *HMM*, each component of the above *rhythmic pattern* will be represented by a *HMM* state, as shown in Figure 1. Each state models by means of a *Gaussian pdf* with mean value, μ_i, the time duration of the respective event (within an allowable *tempo* fluctuation). That is, for the above example we have three states each tuned to the respective note duration.

Table 1. *HMM* rhythmic pattern.

$\frac{2}{4}$	$dotted\ eighth$ - $dotted\ eighth$ -$eighth$
$\frac{3}{4}$	$quarter$ - $eighth$ - $eighth$ - $eighth$ - $eighth$
$\frac{7}{8}$	$dotted\ quarter$ - $quarter$ - $quarter$
$\frac{9}{8}$	$quarter$ - $quarter$ - $quarter$ - $dotted\ quarter$

Other patterns explored, are shown in Table 1 and constitute the most popular patterns in traditional Greek dance music, which has been studied for our experiments. Let us now proceed with the *Markov* modeling details.

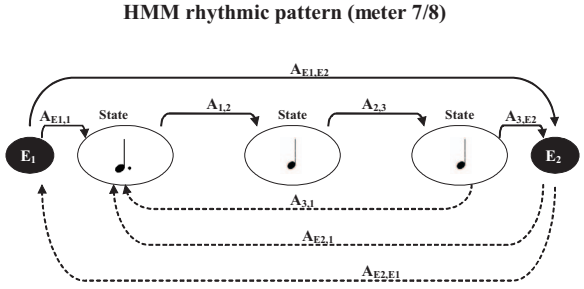

Fig. 1. 3-state HMM modeling of a $\frac{7}{8}$ rhythmic pattern.

3.1. End States and Enhanced Viterbi Algorithm

In Figure 1 except from the three rhythmic *pattern* states, two more states are added, namely E_1, and E_2 (displayed in black). These states will be referred to as *end* states and are allowed to emit all the detected *IOI*s with a uniform probability. The physical meaning of these states is that the *HMM* can bounce between them whenever a sequence of *IOI*s does not conform with the pattern being modeled. On the other hand, the states that model the *rhythmic pattern* are assumed to emit *IOI*s following a Gaussian probability.

In *HMM* terminology [8], let $\lambda = \{\pi, \mathrm{A}, \mathrm{B}\}$, be the parameters of the *HMM* that models a rhythmic pattern. π_i is the initial state probability, $A_{S \times S}$ the state transition matrix, B_i the Gaussian probability distribution (*pdf*) of each *pattern* state, and *S* the number of states (including *pattern* and *end* states). Each Gaussian *pdf*, is associated with a *pattern*

component (i.e., $dotted\ quarter$), with mean time duration μ_i and standard deviation σ_i,where time is measured in frames. In our experiments σ_i was set to $\sim 0.06sec$ (3 frames) in order to generate high probabilities when the correct onset durations are detected. The initial probabilities were set to $\pi_{=}[\frac{1}{2}\ \ \frac{1}{2}\ \ 0 \ldots 0]$, (*S*-1 zeros), forcing all paths to start from the first *end* state or the first *pattern* state. Furthermore, all self transition probabilities are set to zero, i.e., $A_{i,i} = 0$. The only allowable *right* to *left* transitions are those from the second *end* state and the last *pattern* state to the first *end* state and the first *pattern* state, marked with dashed arrows in Figure 1. This allows for tracking repetition in terms of *rhythmic patterns*.

To find a single best state sequence $Q = \{\mathrm{q_1}, \mathrm{q_2}, \ldots, \mathrm{q_T}\}$, where *T* is the time instance of the last observation, the *Viterbi* algorithm is used. First, let us define the forward variable [8]:

$$a_t(j) = P(q_1\ q_2\ \ldots\ q_t,\ state\, j\, ends\, at\, t \mid \lambda),\ \mathrm{j} = 1, \ldots, \mathrm{S} \tag{1}$$

where $a_t(j)$ stands for the probability of the model finding itself in the j-th state after the first t observations have been emitted. It can be shown that ([8, 9]):

$$a_t(j) = \max_{1 \le t \le T, 1 \le i \le S, i \ne j} [\delta_t(i,j)] \tag{2}$$

$$\delta_t(i,j) = a_{t-1}(i) A_{i,j} B_j(t) \tag{3}$$

where t is the time index.

From Equation (3), it can be seen that the standard *Viterbi* algorithm employs a *Type B* cost function for the generation of the *trellis* diagram [10]. A *Type B* cost function takes into consideration *both* the transition costs between nodes $[i,j]$ ($A_{i,j}\ B_j(t)$), as well as the accumulated node costs ($a_{t-1}(i)$). In our approach, a *Type T* cost function was used instead, that only accounts for the transition cost between nodes. It is worth mentioning that a *Type T* cost retains the *Markovian* nature of the *trellis* diagram [10]. In *Markov* model terminology Equation (3) reduces to:

$$\delta_t(i,j) = A_{i,j} B_j(t) \tag{4}$$

By eliminating the forward probability, this cost function takes into account only the "local" activity of the most recent transition. If the *HMM* enters several times the *end* states before entering the *pattern* states, this will not affect local high probability transitions between *pattern* states which indicate that the pattern has been found.

To find the best state sequence, $Q = \{q_1, q_2, \ldots q_T\}$ for each long-term segment the arguments that maximize Eq. (2) are first stored in a two dimensional array ψ, as $\psi(j,t)$

$$\psi(j,t) = argmax\ [\delta_t(i,j)],\ 1 \le i \le S,\ i \ne j \tag{5}$$

At a next step, a backtracking procedure is applied on every node that corresponds to the last state of the *rhythmic pattern*, irrespective of time instance. This is expected to yield a number of paths. In order to select the best one (with the highest probability), the path probabilities have to be computed. To this end, if $Q = \{q_1, q_2, \ldots q_T\}$ is an extracted path, the associated probability is calculated from the equation:

$$p_{model} = \prod_{\forall q \in Q} a_t(q),\ and\ q\ not\ an\ end\ state. \tag{6}$$

As shown in the above Equation (6), the *end* states do not participate in the calculation of the pattern recognition probability since they do not belong in the *rhythmic patterns* modeled by the *HMM*s.

Due to the nature of polyphonic music, it is obvious that the onsets returned during the feature extraction process will outnumber the onsets corresponding to the *correct* beat locations. To address the above problem, an *enhancement* of the *Viterbi* algorithm was employed, which is a variation of that introduced in [11]. Let us consider the onset sequence R for an audio region, i.e.:

$$\begin{aligned} R \ = \ & \{\ldots, a(m-3), O_{z_{m-3}}, a(m-2), O_{z_{m-2}}, \\ & a(m-1), O_{z_m - 1}, a(m), O_{z_m}, a(m+1), O_{z_{m+1}}, \ldots\} \end{aligned}$$

Let $a(m)$ and $a(m-3)$ be two *correct* onsets with two *false* ones, $[a(m-2), a(m-1)]$ in between. Their corresponding durations of $[a(m-2), a(m-1)]$ are $[O_{z_{m-2}}, O_{z_{m-1}}]$. Although $a(m-3)$ is a *correct* onset, it's corresponding duration $O_{z_{m-3}}$ is erroneous, due to the presence of the events $a(m-2), a(m-1)$. Taking into account the zero components, the correct duration can be derived as $\sum_{i=1}^{3} O_{z_{m-i}}$. In this way, we offer to the *HMM* the possibility to eliminate *false* onsets and keep the *correct* ones, while searching for the optimal path and if a lower cost (higher probability) is achieved by eliminating events, the *Viterbi* is given the means to do it. In other words, the cost now becomes "context" dependent. This context dependency of the *Viterbi* algorithm leads to the modification of Equation (4) as:

$$\hat{\delta}_t(i,n,j) \ = \ A_{i,j} \hat{B}_j, \tag{7}$$

where: $\hat{B}_j \ = \ B_j(\sum_{d=t-n+1}^{t} O_{z_d})$, where n is the index of the zero component being added and D the maximum number of observations allowed to be summed. The maximum number of observation symbols over which a state is allowed to sum, depends upon the tolerance of each state mean duration variation $\Delta\mu_i$. This is expressed as: $\forall i \in$ *pattern* states, $\sum_{d=1}^{D} O_{z_d} \ \le \ \Delta\mu_i$. In this work, a constant state duration variation $\Delta\mu_i \simeq 20\%\mu_i$ was allowed, based on extensive experimentation. Equation (2) is now transformed to:

$$\hat{a}_t(j) = \max_{1 \le t \le T, 1 \le n \le D, 1 \le i \le S, i \ne j} \left[\hat{\delta}_t(i,n,j)\right] \tag{8}$$

Unlike the *pattern* states, the *end* states are not allowed to sum consecutive onsets. This is justified by the fact that *end* states

are not actually a part of the examined *rhythmic pattern*, but rather serve as "collectors" for erroneous and "off-beat" onsets. After the whole feature sequence has passed through the *HMM* the resulting pattern's locations are examined. Among the correct locations returned by the algorithm, false pattern locations may appear. An iterative procedure connecting patterns with consecutive locations or locations that differ a maximum distance of 10 pattern lengths, P_L, within a time threshold of $0.1sec$ takes place. In this manner, chains of *rhythmic patterns* are formed. This is employed under the assumption that the same *rhythmic pattern* is encountered throughout a single recording, as it is the case with Greek Traditional dances. The chain holding the maximum number of patterns is returned as the answer to the *rhythmic pattern* search.

4. EXPERIMENTS

Our music dataset consists of 69 audio tracks containing 4447 *rhythmic patterns*, as shown in Table 2. Table 2 suggests that the algorithm has a 100% precision in the location of the patterns returned, meaning that the locations of patterns always coincide with the perceived ones. In other words, the algorithm guarantees that *all* identified pattern locations are correct. The algorithm's recall is also satisfactory in music meters $\frac{2}{4}$, $\frac{7}{8}$, $\frac{9}{8}$. By recall we mean the percentage of the correctly identified patterns in respect to their total number. The recall in $\frac{3}{4}$ is lower due to the fact that in this specific dance style the leading instrument and/or singing voice does not always follow the accompaniment instruments resulting to only a few *true* onsets. In general, the algorithm's performance is affected when large *tempo* variations occur, i.e. *true IOI*s fall outside the range specified by $\Delta\mu_i$. In addition, the recall of the proposed method decreases when correctly located patterns form isolated chains that are distant with each other.

It is worth mentioning that, if a *HMM* that models a specific pattern is fed with the feature sequence R of a music recording that does not follow the specific pattern, then the *HMM* will not form any chains of patterns *at all*. Therefore there exists zero confusion among the patterns modeled. As a result, the existence of a *rhythmic pattern* chain can be used as the means to associate audio recording and *rhythmic patterns* occurring in specific dances.

Audio examples demonstrating the algorithm's performance can be found in:
www.di.uoa.gr/∼jantonop/rhythmexamples.htm

5. CONCLUSIONS

In this work, we have developed an algorithm that can track *rhythmic patterns* within an audio file under the assumption that these exist. In our future research priorities, local beat estimation will be attempted to account for varying tempo recordings. In addition, an effort for a more efficient *rhythmic pattern* connection algorithm will be looked into.

Table 2. Pattern location results per *rhythmic pattern*.

music meter	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{7}{8}$	$\frac{9}{8}$
# patterns	1352	1496	1029	570
# located	1069	479	776	403
# not located	283	1017	253	167
bpm	89 -95	93-105	240-270	270-385
precision (%)	100	100	100	100
recall (%)	~ 76	~ 34	~ 72	~ 70.7

6. REFERENCES

[1] Masataka Goto and Yoichi Muraoka, ," in *Music Understanding At The Beat Level — Real-time Beat Tracking For Audio Signals*. IJCAI-95 Workshop on Computational Auditory Scene Analysis, 1995, pp. 68–75.

[2] Masataka Goto and Yoichi Muraoka, "Real - time beat tracking for drumless audio signals: Chord change detection for musical decisions," *Speech Communication*, vol. 27, pp. 291–294, April 1999.

[3] E. Scheirer, "Tempo and beat analysis of acoustic music signal," *Journal of Acoustic Society*, vol. 103(1), pp. 588–601, October 1993.

[4] William A. Sethares, Robin D. Morris, and James C. Sethares, "Beat tracking of musical performances using low-level audio features," *IEEE Transactions on Speech, Audio and Processing*, vol. 13(1), pp. 275–285, 2005.

[5] Anssi Klapuri, Antti J. Eronen, and Jaakko T. Astola, "Automatic estimation of the meter of acoustic musical signals.," *IEEE Transactions on Speech, Audio and Language Processing*, vol. 14(1), 2006.

[6] Simon Dixon, Fabien Gouyon, and Gerhard Widmer, ," in *Towards characterization of music via rhythmic patterns*. ISMIR Proceedings, 2004, pp. 1281–1284.

[7] Aggelos Pikrakis, Iasonas Antonopoulos, and Sergios Theodoridis, ," in *Music Meter and Tempo Tracking from Raw Polyphonic Audio*. ISMIR Proceedings, 2004.

[8] L.R. Rabiner, "A tutorial on hidden markov models," *Proceedings of the IEEE*, vol. 77(2), pp. 257–286, 1989.

[9] S. Theodoridis and K. Koutroumbas, "Pattern recognition," *Academic Press, 3d Edition*, 2006.

[10] J.G. Proakis J.R. Deller and J.H.L. Hansen, "Discrete-time processing of speech signals," *Macmillan, 2nd Edition*, 1999.

[11] Aggelos Pikrakis, Sergios Theodoridis, and Dimitris Kamarotos, ," in *Classification of Musical Patterns Using Variable Duration Hidden Markov Models*. Proc EUSIPCO, 2004, pp. 1281–1284.