# MUSIC PATTERN DISCOVERY WITH VARIABLE MARKOV ORACLE: A UNIFIED APPROACH TO SYMBOLIC AND AUDIO REPRESENTATIONS

**Cheng-i Wang, Jennifer Hsu and Shlomo Dubnov**
Music Department
University of California, San Diego
{chw160, jsh008, sdubnov}@ucsd.edu

## ABSTRACT

This paper presents a framework for automatically discovering patterns in a polyphonic music piece. The proposed framework is capable of handling both symbolic and audio representations. Chroma features are post-processed with heuristics stemming from musical knowledge and fed into the pattern discovery framework. The pattern-finding algorithm is based on *Variable Markov Oracle*. The *Variable Markov Oracle* data structure is capable of locating repeated suffixes within a time series, thus making it an appropriate tool for the pattern discovery task. Evaluation of the proposed framework is performed on the JKU Patterns Development Dataset with state of the art performance.

## 1. INTRODUCTION

Automatic discovery of musical patterns (motifs, themes, sections, etc.) is a task defined as identifying salient musical ideas that repeat at least once within a piece [3, 11] with computational algorithms. In contrast to "segments" found in the music segmentation task [14], the patterns found here may overlap with each other and may not cover the entire piece. In addition, the occurrences of these patterns could be inexact in terms of harmonization, rhythmic pattern, melodic contours, etc. Lastly, hierarchical relations between motifs, themes and sections are also desired outputs of the pattern discovery task.

Two major approaches for symbolic representations are the string-based and the geometric methods. A string-based method treats a symbolic music sequence as a string of tokens and applies string pattern discovery algorithms on the sequence [2, 18]. A geometric method views musical patterns as shapes appearing on a score and enables inexact pattern matching as similar shapes imply different occurrences of one pattern [4, 16]. For a comprehensive review of pattern discovery with symbolic representations, readers are directed to [11]. For audio representations, geometric methods for symbolic representations have been extended to handle audio signals by multi $F0$-estimation with beat tracking techniques [5]. Approaches adopted from music segmentation tasks using self-similarity matrices and greedy search algorithms are proposed in [19, 20]. Most of the research involving audio representations has been focused on "deadpan audio" rendered from MIDI. In [5], the pattern discovery task is extended to live performance audio recordings with a single recording for each music piece. In the current study, instead of directly applying the proposed framework on performance recordings, multiple recordings are gathered for each musical piece to aid the pattern discovery on deadpan audio.

In this paper, the work presented in [25] focusing on pattern discovery on deadpan audio is extended to handle symbolic representations. The framework proposed in this paper can be seen as a string-based method in which input features are symbolized. The framework consists of two blocks: 1) feature extraction with post-processing routines and 2) the pattern finding algorithm. For both symbolic and audio representations, chroma features are extracted and post-processed based on musical heuristics, such as modulation, beat-aggregation, etc. The core of the pattern finding algorithm is a *Variable Markov Oracle* (*VMO*). A *VMO* is a data structure capable of symbolizing a signal by clustering the observations in a signal, and is derived from the *Factor Oracle* (*FO*) [13] and *Audio Oracle* (*AO*) [9] structures. The *FO* structure is a variant of a suffix tree data structure and is devised for retrieving patterns from a symbolic sequence [13]. An *AO* is the signal extension of a *FO*, and is capable of indexing repeated sub-clips of a signal sampled at discrete times. *AO*s have been applied to audio query [6] and audio structure discovery [8]. The *VMO* data structure was first proposed in [24] as an efficient audio query-matching algorithm. This paper shows the capability of using a *VMO* to find repeated sub-clips in a signal in an unsupervised manner.

This paper is structured as follows: section 2 introduces the *VMO* data structure and the accompanying pattern finding algorithm. Section 3 documents the experiments on symbolic and audio representations as well as the dataset, feature extraction, and task setup. Section 4 provides an evaluation of the experiment. Last, future work, observations and insights are discussed in section 5.

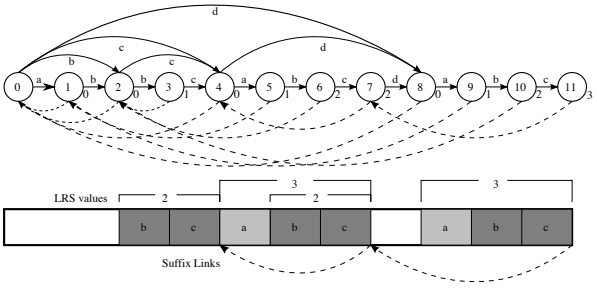


**Figure 1**. (Top) A *VMO* structure with symbolized signal $\{a, b, b, c, a, b, c, d, a, b, c\}$, upper (solid) arrows represent forward links with symbols for each frame and lower (dashed) are suffix links. Values outside of each circle are the `lrs` value for each state. (Bottom) A visualization of how patterns $\{a, b, c\}$ and $\{b, c\}$ are related to $lrs$ and $sfx$.

## 2. VARIABLE MARKOV ORACLE

A *VMO* symbolizes a time series $O$, sampled at time $t$, into a symbolic sequence $Q = q_1, q_2, \ldots, q_t, \ldots, q_T$, with $T$ states and with frame $O[t]$ labeled by a symbol $q_t$. The symbols are formed by tracking suffix links along the states in an oracle structure. An oracle structure (either *FO*, *AO* or *VMO*) carries three kinds of links: forward link, suffix link and reverse suffix link. A suffix link is a backward pointer that links state $t$ to $k$ with $t > k$, without a label, and is denoted by $\texttt{sfx}[t] = k$.

$$\texttt{sfx}[t] = k \iff \text{the longest repeated suffix of } \{q_1, q_2, \ldots, q_t\} \text{ is recognized in } k.$$

Suffix links are used to find repeated suffixes in $Q$. In order to track the longest repeated suffix at each time index $t$, the length of the longest repeated suffix at each state $t$ (denoted as $\texttt{lrs}[t]$) is computed by the algorithm described in [13]. A reverse suffix link, $\texttt{rsfx}[k] = t$, is the suffix link in the reverse direction. `sfx`, `lrs` and `rsfx` allow for the proposed pattern discovery algorithm described in section 2.2.

Forward links are links with labels and are used to retrieve any of the factors from $Q$. Since forward links are not used in the proposed algorithm, readers are referred to [13] for details.

The last piece for the construction of a *VMO* is a threshold value, $\theta$. $\theta$ is used to determine if the incoming $O[t]$ is similar to one of the frames following the suffix link beginning at $t-1$. Two frames, $O[i]$ and $O[j]$, are assigned the same symbol if $|O[i] - O[j]| \leq \theta$. In extreme cases, a *VMO* may assign different symbols to every frame in $O$ ($\theta$ excessively low), or a *VMO* may assign the same symbol to every frame in $O$ ($\theta$ excessively high). In these two cases, the *VMO* structure is incapable of capturing any patterns (repeated suffixes) in the signal. The optimal $\theta$ can be found by calculating the *Information Rate* (*IR*), a music information dynamics measure, and this process is described in section 2.1. An example of an oracle structure with extreme $\theta$ values is shown in Fig. 2.

The on-line construction algorithms of *VMO* are introduced in [24] and not repeated here. Fig. 1 shows an example of a constructed *VMO* and how `lrs` and `sfx` are related to pattern discovery. The symbols formed by gathering states connected by suffix links share the following properties : 1) the pairwise distance between states connected by suffix links is less than $\theta$, 2) the symbolized signal formed by the oracle can be interpreted as a sample from a variable-order Markov model because the states connected by suffix links share common suffixes with variable length, 3) each state is labeled by a single symbol because each state has a single suffix link, 4) the alphabet size of the assigned symbols is unknown before the construction and is determined by $\theta$.

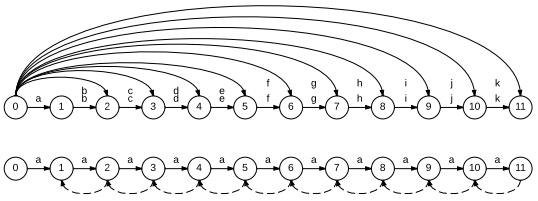

**Figure 2**. Two oracle structures with extreme values of $\theta$. The characters near each forward link represent the assigned labels. (Top) The oracle structure with $\theta = 0$ or extremely low $\theta$ value. (Bottom) The oracle structure with a very high $\theta$ value. In both cases the oracles are not able to capture any structure in the time series.

### 2.1 Model Selection via Information Rate

The same input signal may be associated with multiple *VMO*s with different suffix structures and different symbolized sequences if different $\theta$ values are used to construct the *VMO*s. To select the one symbolized sequence with the most informative patterns, *IR* is used as the criterion in model selection between different structures generated by different $\theta$ values. *IR* is an information theoretic measure capable of measuring the information content of a time series [7] in terms of the predictability of its source process on the present observation given past ones. In the context of pattern discovery with a *VMO*, a *VMO* with higher *IR* value captures more of the repeating sub-clips (ex. patterns, motives, themes, gestures, etc) than the ones with lower $IR$ values.

The *VMO* structure uses the same approach as the *AO* structure [8] to calculate $IR$. Let $x_1^N = \{x_1, x_2, \ldots, x_N\}$ denote time series $x$ with $N$ observations, $H(x)$ the entropy of $x$, the definition of *IR* is

$$IR(x_1^{n-1}, x_n) = H(x_n) - H(x_n|x_1^{n-1}). \quad (1)$$

*IR* is the mutual information between the present and past observations and is maximized when there is a balance between variations and repetitions in the symbolized signal. The value of *IR* can be approximated by replacing the entropy terms in (1) with complexity measures associated with a compression algorithm. These complexity measures

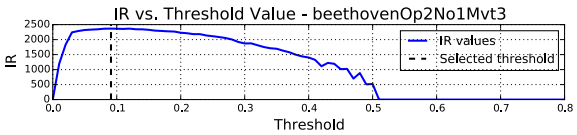


**Figure 3**. *IR* values are shown on the vertical axis while $\theta$ are on the horizontal axis. The solid blue curve shows the relationship between *IR* and $\theta$, and the dashed black line indicates the chosen $\theta$ by locating the maximum *IR* value. Empirically, *IR* curves exhibit quasi-concave function shapes, thus a global maximum can be located.

**Algorithm 1** Pattern Discovery using *VMO*

**Require:** *VMO*, V, of length $T$ and minimum pattern length $L$.
**Ensure:** $\texttt{sfx}, \texttt{rsfx}, \texttt{lrs} \in V$
1: Initialize $Pttr$ and $PttrLen$ as empty lists.
2: Initialize $prevSfx = -1, K = 0$
3: **for** $i = T : L$ **do**
4: $pttrFound = False$
5: **if** $i - \texttt{lrs}[i] + 1 > \texttt{sfx}[i] \wedge \texttt{sfx}[i] \neq 0 \wedge \texttt{lrs}[i] \geq L$ **then**
6: **if** $\exists k \in \{1, \ldots, K\}, \texttt{sfx}[i] \in Pttr[k]$ **then**
7: Append $i$ to $Pttr[k]$
8: $PttrLen[k] \leftarrow \min(lrs[i], PttrLen[k])$
9: $pttrFound = True$
10: **end if**
11: **if** $prevSfx - \texttt{sfx}[i] \neq 1 \wedge pttrFound == False$ **then**
12: Append $\{\texttt{sfx}[i], i, \texttt{rsfx}[i]\}$ to $Pttr$
13: Append $\min\{\texttt{lrs}[\{\texttt{sfx}[i], i, \texttt{rsfx}[i]\}]\}$ to $PttrLen$
14: $K \leftarrow K + 1$
15: **end if**
16: $prevSfx \leftarrow \texttt{sfx}[i]$
17: **else**
18: $prevSfx \leftarrow -1$
19: **end if**
20: **end for**
21: **return** $Pttr, PttrLen, K$

are the number of bits used to compress $x_n$ independently and compress $x_n$ using the past observations $x_1^{n-1}$. The formulation of combining the lossless compression algorithm, *Compror* [12], with *AO* and *IR* is provided in [8]. A visualization of the sum of *IR* values versus different $\theta$s on one of the music pieces tested in this paper is depicted in Fig. 3.

### 2.2 Pattern Discovery

Algorithm 1 shows the *VMO*-based algorithm for the automatic pattern discovery task. The idea behind Algorithm 1 is to track patterns by following `sfx` and `lrs`. `sfx` provides the locations of patterns, and `lrs` indicates the length of these patterns. In line 5 of Algorithm 1, checks are made so that redundant patterns are avoided, and the lengths of patterns are larger than a user-defined minimum $L$. From line 6 to 10, the algorithm recognizes occurrences of established patterns, and from line 11 to 15 it detects new patterns and stores them into $Pttr$ and $PttrLen$.

Algorithm 1 returns $Pttr, PttrLen$ and $K$. $Pttr$ is a list of lists with each $Pttr[k], k \in \{1, 2, \ldots, K\}$, a list containing the ending indices of different occurrences of the $k$th pattern found. $K$ is the total number of patterns found. $PttrLen$ has $K$ values representing the length of the $k$th pattern in $Pttr$.

## 3. EXPERIMENTS

The dataset chosen for the music pattern discovery is the JKU Pattern Development Dataset (JKU-PDD) [3]. This dataset consists of five polyphonic classical music pieces or movements in both symbolic and audio representations. The ground truth of repeated patterns (motifs, themes, sections) for each piece is annotated by musicologists. The details of the experimental setup are provided in the following sections.

### 3.1 Feature Extraction

For the automatic musical pattern discovery task, the chromagram is the input feature to Algorithm 1 for both the symbolic and audio representations. The chromagram is a feature that characterizes harmonic content and is a commonly used in musical structure discovery [1].

#### *3.1.1 Symbolic Representation*

For the experiments described in this paper, the symbolic representation chosen is MIDI, but other symbolic representations may be used instead. The chromagram derived from the symbolic representation is referred to as the "MIDI chromagram".

The MIDI chromagram is similar to the MIDI histogram described in [23] and represents the presence of pitch classes during each time frame. To create a MIDI chromagram with quantization $b$ in terms of MIDI whole note beats, frame size $M$, and hop size $h$, the MIDI file is first parsed into a matrix where each column is a MIDI beat quantized by $b$ and each row is a MIDI note number $(0 - 127)$. For each analysis frame, the velocities are summed over $M$ MIDI beats, and then folded and summed along the MIDI notes to create a single octave of velocities. In other words, all velocities that correspond to MIDI notes that share the same modulo 12 are summed. The analysis frame then hops $h$ MIDI beats forward in time, repeats the folding and summing, and continues on until the end of the MIDI matrix is reached. The bottom plot in Fig. 4 is an example of the MIDI chromagram extracted from the Beethoven minuet in the JKU-PDD.

#### *3.1.2 Audio Recording*

The routines for extracting the chromagram from an audio recording used in this paper is as follows. For a mono audio recording sampled at 44.1 kHz, the recording is first downsampled to 11025 Hz. Next, a spectrogram is calculated using a Hann window of length 8192 with 128 samples overlap. Then the constant-Q transform of the spectrogram is calculated with frequency analysis ranging between $f_{min} = 27.5$ Hz to $f_{max} = 5512.5$ Hz and 12 bins per octave. Finally, the chromagram is obtained by folding the constant-Q transformed spectrogram into a single octave to represent how energy is distributed among the 12 pitch classes.

To achieve the pattern discovery on a music metrical level, the chroma frames are aggregated with a median filter according to the beat locations found by a beat tracker

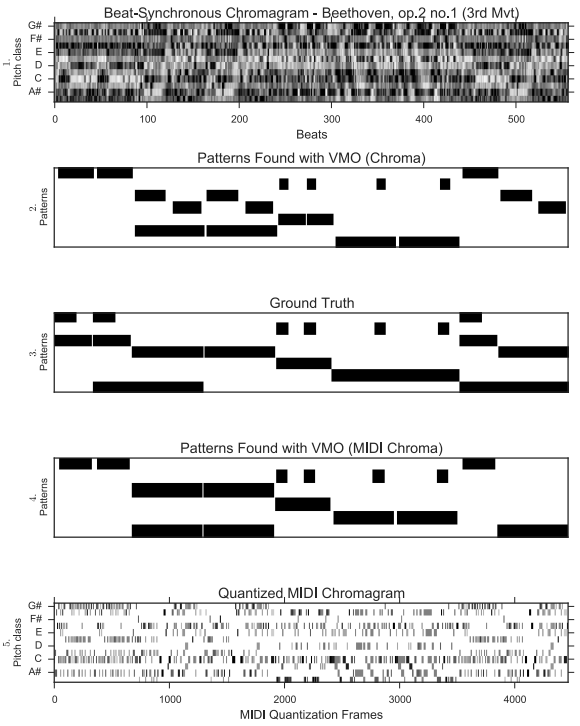


**Figure 4**. Features, found patterns, and ground truth for the Beethoven minuet in the JKU-PDD. 1. Beat-synchronous chromagram from the deadpan audio recording. 2. Patterns found by Algorithm 1 using the chromagram shown above. 3. Ground truth from JKU-PDD. 4. Patterns found by Algorithm 1 using the MIDI chromagram. 5. Quantized MIDI chromagram. For 2., 3. and 4., each row is a pattern place holder with dark regions representing the occurrences on the timeline. The order of found patterns is manually sorted to best align with the ground truth for visualization purpose. Notice the hierarchical relations of patterns embedded in the ground truth and found from the algorithms.

[10] conforming to the music metrical grid. For finer rhythmic resolution, each beat identified is spliced into two sub-beats before chroma frame aggregation. Last, the sub-beat-synchronous chromagram is whitened with a $log$ function. Whitening boosts the harmonic tones implied by the motifs so that the difference between the same motif with and without harmonization is reduced. See the top plot in Fig. 4 for an example of the the beat-synchronous chromagram extracted from the Beethoven minuet in the JKU-PDD.

### 3.2 Repeated Themes Discovery

For both symbolic and audio representations, after the chroma feature sequence $O$ is extracted from the music piece as described in section 3.1.1 and 3.1.2, $\theta \in (0.0, 2.0]$ is used to construct multiple *VMO*s with $O$. The $L_2-$norm is used to calculate the distance between incoming observations and the ones stored in a *VMO*. The single *VMO* with the highest *IR* is fed into Algorithm 1 with $L$ to find patterns and their occurrences. Instead of setting $L = 5$ for all pieces as in [25], $L$ is set according to `lrs` as $L = \frac{\gamma}{T}\sum_{t=1}^{T} \texttt{lrs}[t]$, where $L$ is adaptive to the average length of repeated suffixes found in the piece. $\gamma$ is a scaling parameter which is set to $0.5$ empirically.

To consider transposition (moving patterns up or down by a constant pitch interval), the distance function used for *VMO* structures is a cost function with transposition invariance. For a transposition invariant cost function, a cyclic permutation with offset $k$ on an $n$-dimensional vector $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ is defined as

$$cp_k(\mathbf{x}) := \{x_i \rightarrow x_{(i+k \bmod n)}, \forall i \in (0, 1, \ldots, n-1)\},$$

and the transposition invariant dissimilarity $d$ between two vectors $x$ and $y$ is defined as, $d = \min_k\{\|x - cp_k(y)\|_2\}$. $n = 12$ for the chroma vector, and the cost function is used during the *VMO* construction.

In addition to the basic chromagram, a stacked chromagram using time-delay embedding with $M$ steps of history as in [22] is also used. Experiments reveal that choices for $b$, $M$, and $h$ for both the MIDI chromagram and the stacked MIDI chromagram can greatly alter the accuracy of patterns discovered. The values used in the experiments were quantization sizes $b = [\frac{1}{8}, \frac{1}{16}, \frac{1}{32}]$, frame size $M = [1, 8, 16, 32]$, and hop lengths $h = [1, 2, 4]$ where $M$ and $h$ are described in terms of MIDI beats of size $b$. It was found that the stacked MIDI chromagram with $b = \frac{1}{32}$, $M = 16$, and $h = 2$ resulted in the best pattern discovery. For the audio representation, there is no significant difference in terms of the patterns found or the evaluation metrics between regular and stacked chromagrams.

Fig. 4 shows the chromagram found from audio and MIDI for the Beethoven minuet in the JKU-PDD along with the patterns found by the *VMO* structure and the ground truth patterns. The patterns found by the audio and symbolic representations share similarities and visually resemble the ground truth patterns. In section 4, quantitative measures for evaluating the patterns found by the *VMO* are explained and reported.

### 3.3 Performance Recordings to Aid Pattern Discovery

Five performance recordings for each of the pieces included in the JKU-PDD are collected in order to further explore the discovery of repeated themes. The motivation behind this experiment is to explore the notion that music performances contain information about how performers interpret the musical structure embedded in the score [21] and to examine whether or not the patterns found on deadpan audio could be improved with the addition of such information.

For each of the performance recordings, the chromagram is extracted and aggregated along the beats as described in section 3.1.2. Dynamic Time Warping [17] is used to align the beat-synchronous chromagram from the performance audio with the beat-synchronous chromagram of the deadpan audio. Since motif annotations on these performance recordings do not exist yet, the alignment between the deadpan audio and performance recordings are necessary so that the patterns found from the performance

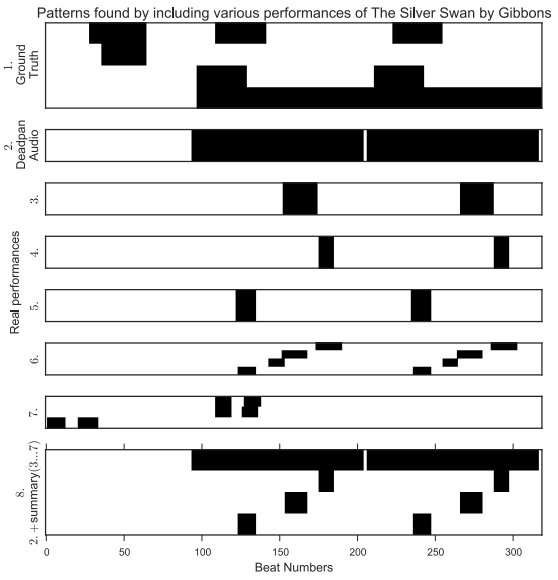


**Figure 5**. 1. Ground truth from JKU-PDD. 2. Patterns found from deadpan audio with the *VMO*. $3-7$. Patterns found from the five performances. 8. Patterns from deadpan and performance audio.

recordings can be compared to the ground truth or added to the found patterns from the deadpan audio. The drawback of the alignment is that timing variations containing the performer's structural interpretation are lost. Although timing variations are lost in this experiment, velocity variations applied across time and different voices are retained. The aligned performance audio chromagram is then whitened, normalized and fed into the *VMO* pattern finding algorithm. For patterns found across multiple performances of one piece, the intersection of patterns for any two performances of one piece that are longer than $L$ are kept and added to the found patterns from the deadpan audio. Fig. 5 is an example of how incorporating performance recordings can change the discovered patterns from deadpan audio.

## 4. EVALUATION

The evaluation follows the metrics proposed in the Music Information Retrieval Evaluation eXchange (MIREX) [3]. Three metrics are considered for inexact pattern discovery. For each metric, standard $F_1$ score, defined as $F_1 = \frac{2PR}{(P+R)}$, precision $P$ and recall $R$ are calculated. The first metric is the establishment score ($est$) which measures how each ground truth pattern is identified and covered by the algorithm. The establishment score takes inexactness into account and does not consider occurrences. The second metric is the occurrence score ($o(c)$) with a threshold $c$. The occurrence score measures how well the algorithm performs in finding occurrences of each pattern. The threshold $c$ determines whether or not an occurrence should be counted. The higher the value for $c$, the lower the tolerance. $c = \{0.5, 0.75\}$ are used in standard MIREX evaluation. The last metric is the three-layer score that considers both the establishment and occurrence score. The results of the proposed framework are listed in Table 1 along with a comparison to previous work.

From the evaluations for both symbolic and audio representations, the establishment scores are generally lower than the occurrence scores, meaning that the proposed algorithm is better at finding occurrences of established patterns than finding all possible patterns. With the symbolic representation, the standard $F_{\text{est}}$, $F_{o(.75)}$, and $F_3$ scores are better than previously published results. The establishment, occurrence, and three-layer precision scores are also as good as or better than previous algorithms [5, 15]. The recall scores reveal that this is a part of the algorithm that could be improved as previous algorithms all scored higher on recall than the proposed algorithm. Similar to the symbolic results, the proposed audio algorithm achieves high $F1$ and precision scores for the establishment, occurrence, and three-layer scores. The recall of the audio algorithm is higher than previously reported results [5, 19, 20]. The recall rates of the proposed framework are inferior when compared to the precision scores and previous work in symbolic representation. This may occur because chroma features were used and the folding of the constant-Q spectrogram discards information contained in different voices.

The inclusion of performance recordings is the effort made in this work to improve both the coverage and accuracy of the pattern discovery framework for audio representations. Due to space limitations, the detailed metrics for each piece in the JKU-PDD is not shown here. The effects of including performance recordings are described here. The establishment recall rate and occurrence precision rate with threshold $0.5$ are improved when performance recordings are included, but in general the pattern discovery task is not improved because the decrease in establishment precision rate is larger than the improvement on recall rates. This result indicates that more patterns and their occurrences could be discovered if different versions of the same piece are used in the pattern discovery task, but more false positive patterns will be found.

The proposed pattern finding algorithm completed in less time than previously reported algorithms on both symbolic and audio representations. Although the *VMO* data structure is used for both the proposed symbolic and audio algorithms, there is a discrepancy in the time that it takes to find the patterns for all five songs. The audio algorithm takes much less time because the analysis frames are larger than the frames used in the symbolic representation (32th note versus 8th note relatively). Thus, there are less frames to analyze with the audio representation and building a *VMO* takes less time.

Fig. 6 is a summary of the three-layer $F_1$ scores for each of the 5 pieces in the JKU-PDD for the proposed audio and symbolic frameworks along with the current state of the art results. The small quantization value for the MIDI representation leads to a higher score in the case of the Beethoven and Chopin pieces. The proposed audio

| Algorithm | $F_{\text{est}}$ | $P_{\text{est}}$ | $R_{\text{est}}$ | $F_{o(.5)}$ | $P_{o(.5)}$ | $R_{o(.5)}$ | $F_{o(.75)}$ | $P_{o(.75)}$ | $R_{o(.75)}$ | $F_3$ | $P_3$ | $R_3$ | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VMO symbolic | **60.79** | **74.57** | 56.94 | 71.92 | **79.54** | 68.78 | **75.98** | **75.98** | 75.99 | **56.68** | **68.98** | 53.56 | **4333** |
| [5] | 33.7 | 21.5 | **78.0** | **76.5** | 78.3 | **74.7** | – | – | – | – | – | – | – |
| [15] | 50.20 | 43.60 | 63.80 | 63.20 | 57.00 | 71.60 | 68.40 | 65.40 | 76.40 | 44.20 | 40.40 | **54.40** | 7297 |
| VMO deadpan | **56.15** | **66.8** | 57.83 | **67.78** | 72.93 | **64.3** | **70.58** | **72.81** | **68.66** | **50.6** | **61.36** | 52.25 | **96** |
| deadpan + real | 52.76 | 53.2 | 58.25 | 67.35 | **74.42** | 63.31 | 70.51 | 72.73 | 68.58 | 48.25 | 50.2 | **52.84** | – |
| [20] | 49.8 | 54.96 | 51.73 | 38.73 | 34.98 | 45.17 | 31.79 | 37.58 | 27.61 | 32.01 | 35.12 | 35.28 | 454 |
| [5] | 23.94 | 14.9 | **60.9** | 56.87 | 62.9 | 51.9 | – | – | – | – | – | – | – |
| [19] | 41.43 | 40.83 | 46.43 | 23.18 | 26.6 | 20.94 | 24.87 | 32.08 | 21.24 | 28.23 | 30.43 | 31.92 | 196 |

**Table 1**. Results from various algorithms on the JKU-PDD for both symbolic (upper three) and audio (bottom four) representations. Scores are averaged across pieces. Missing values were not reported in their original publications.

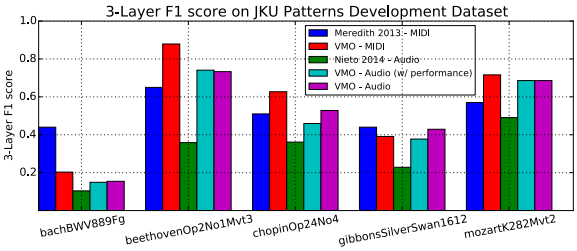


**Figure 6**. Three-layer $F_1$ score ($F_3$ in Table 1) for the proposed audio and symbolic method on the 5 pieces in the JKU-PDD plotted along with state of the art results.

and symbolic framework have the highest $F_1$ value on the Beethoven minuet and the lowest $F_1$ value with the Bach Fugue. When looking at the proposed method along with current the state of the art results, it is evident that the Bach Fugue and the Gibbons piece are songs where patterns are embedded in different voices, and that the Beethoven piece has more consistent repeated phrases. The algorithm for symbolic data described in [15] performs better with Bach and Gibbons in comparison to *VMO* and [20], most likely because of its capability to discover patterns embedded in different yet simultaneous voices.

In summary, our method has improved upon the $F_1$ and $P$ scores as well as time to find patterns. The patterns found using audio and symbolic representations are similar and the evaluation scores reflect this similarity. Improving recall and allowing for inexact occurrences should be a focus for future studies. Source codes and details about the experiments are accessible via Github [1] .

## 5. DISCUSSION

In this work, a framework for automatic pattern discovery from a polyphonic music piece based on a *VMO* is proposed and shown to achieve state of the art performance on the JKU-PDD dataset. With both the regular and stacked MIDI chromagram, a smaller quantization value $b$ results in better pattern discovery because finer details are captured with smaller quantization. From the results, it seems that a larger frame size $M$ for smaller quantization $b$ resulted in better pattern finding. For hop size $h$, it is observed that $h = 2$ results in a hop of a 16th note which is the shortest note in the JKU-PDD ground truth annotations. Results from both the audio and MIDI representations show that the recall of discovered themes could be improved. Although it is possible for a *VMO* to identify inexact patterns from the input feature sequence with symbolization from $\theta$, different occurrences of the same pattern are sometimes not recognized because chroma features discard information from various voices in the music piece. Our framework could be improved if the feature used allows for separation of voices from polyphonic MIDI and audio. Incorporating techniques for identifying multiple voices in polyphonic audio would improve the proposed framework.

In addition to the proposed framework for both symbolic and audio representations, using multiple performance recordings in the repeated themes discovery task for deadpan audio is another novelty presented in this paper. The work done in this paper differs from [5] in that the performance audio recordings are used as supplements to deadpan audio and not analyzed as separate musical entities. The original intention behind using deadpan audio for repeated themes discovery is to allow for the use of audio signal processing techniques, but deadpan audio contains the same amount of information as its symbolic counterpart with less accessibility because of its representation. This is evident by the similarity between the MIREX metrics for the MIDI and deadpan audio since similar techniques are applied. Performance recordings, on the other hand, contain expressive performance variations on phrasing and segmentation. In this paper, it is shown that adding performance recordings to the proposed framework achieved improvements on some of the standard metrics. The next step for advancing the repeated themes discovery task is to annotate the performance recordings so that these recordings can be used as a dataset directly without referencing back to the deadpan audio version. By observing the results from the pattern finding with performance recordings, the patterns found for each performance show informative cues as to how each rendition of the same piece differs from the others visually (Fig. 5). These visualizations are interesting discoveries on their own, even without a comparison to ground truth annotations, and could be further investigated for use in expressive performance analysis and music structural segmentation.

[1] https://github.com/wangsix/VMO_repeated_themes_discovery

## 6. REFERENCES


[1] Juan Pablo Bello. Measuring structural similarity in music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(7):2013–2025, 2011.

[2] Emilios Cambouropoulos, Maxime Crochemore, Costas S Iliopoulos, Manal Mohamed, and Marie-France Sagot. All maximal-pairs in step–leap representation of melodic sequence. *Information Sciences*, 177(9):1954–1962, 2007.

[3] Tom Collins. Discovery of repeated themes and sections. *Retrieved 4th May, http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_&_Sections*, 2013.

[4] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.

[5] Tom Collins, Sebastian Böck, Florian Krebs, and Gerhard Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[6] Arshia Cont, Shlomo Dubnov, Gérard Assayag, et al. Guidage: A fast audio query guided assemblage. In *International Computer Music Conference*, 2007.

[7] Shlomo Dubnov. Spectral anticipations. *Computer Music Journal*, 30(2):63–83, 2006.

[8] Shlomo Dubnov, Gérard Assayag, and Arshia Cont. Audio oracle analysis of musical information rate. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 567–571. IEEE, 2011.

[9] Shlomo Dubnov, Gerard Assayag, Arshia Cont, et al. Audio oracle: A new algorithm for fast learning of audio structures. In *International Computer Music Conference*, 2007.

[10] Daniel P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[11] Berit Janssen, W. Bas de Haas, Anja Volk, and Peter Kranenburg. Discovering repeated patterns in music: potentials, challenges, open questions. In *10th International Symposium on Computer Music Multidisciplinary Research*. Laboratoire de Mécanique et d'Acoustique, 2013.

[12] Arnaud Lefebvre and Thierry Lecroq. Compror: online lossless data compression with a factor oracle. *Information Processing Letters*, 83(1):1–6, 2002.

[13] Arnaud Lefebvre, Thierry Lecroq, and Joël Alexandre. An improved algorithm for finding longest repeats with a modified factor oracle. *Journal of Automata, Languages and Combinatorics*, 8(4):647–657, 2003.

[14] Brian McFee and Daniel P. W. Ellis. Analyzing song structure with spectral clustering. In *The 15th International Society for Music Information Retrieval Conference*, pages 405–410, 2014.

[15] David Meredith. COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In *International Society for Music Information Retrieval Conference*, 2013.

[16] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[17] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[18] Oriol Nieto and Morwaread Farbood. Perceptual evaluation of automatically extracted musical motives. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pages 723–727, 2012.

[19] Oriol Nieto and Morwaread Farbood. MIREX 2013: Discovering musical patterns using audio structural segmentation techniques. *Music Information Retrieval Evaluation eXchange, Curitiba, Brazil*, 2013.

[20] Oriol Nieto and Morwaread Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *The 15th International Society for Music Information Retrieval Conference*, 2014.

[21] John Rink, Neta Spiro, and Nicolas Gold. Motive, gesture, and the analysis of performance. *New Perspectives on Music and Gesture*, pages 267–292, 2011.

[22] Joan Serrà, Meinard Mueller, Peter Grosche, and Josep Ll Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, 16(5):1229–1240, 2014.

[23] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[24] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[25] Cheng-i Wang and Shlomo Dubnov. Pattern discovery from audio recordings by variable markov oracle: A music information dynamics approach. In *Acoustics, Speech, and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.