The Musical Schemagram: Time-scale Visualization of Repeated Patterns in Music

Tsung-Ping Chen and Li Su
Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: lisu@iis.sinica.edu.tw Tel/Fax: +886-2-27883799/+886-2-27824814

***Abstract*—In this paper, we propose the *schemagram*, a novel representation for visualizing the motives, themes, sections, as well as all the possible repeated patterns in symbolic musical data. And a geometry-based pattern discovery algorithm combining a clustering method is introduced for finding repeated musical patterns. Repeated patterns are essential elements for human to perceive the structure and meanings of music. Although there are many algorithms that aim to discover repeated patterns within a musical piece, the task of intra-opus pattern discovery is still an open problem due to the large variances between patterns in terms of amount, size, significance, etc., which result in the ambiguity for establishing a pattern. In this paper, we examine the geometric approach for finding repeated patterns, and explore the capability of the schemagram to interpret the structure of a musical piece through chronologically organizing the repeated patterns found within the piece.**

I. Introduction

Music is prominently repetitive, and repeated patterns are the vocabulary of music. The recognition of repeated patterns is an important step toward comprehending the music and has been stressed by music theorists and music psychologists [1], [2], [3]. Every piece of music has a foundation of patterns which builds the senses of periodicity and symmetry, and conducts various kinds of perception of musical form. The concept of repeated pattern in music is multidimensional. Most of the musical elements, such as melody, chord, and rhythm, usually repeat themselves at different levels and in different ways. Some patterns are easy to be identified, while some others are rather obscure, whether intentionally or not, leaving a mysterious realm beneath the surface of music. Because of the complexity and diversity of musical patterns, computational approaches for discovering repeated patterns in musical data have gained attention in the field of music information retrieval (MIR) [4]–[10]. Studies related to the topic have focused on different types of musical repetition, including rhythmic patterns [4], [5], musical structure [6], [7], and *pitch structure* [8]–[10]. Among these studies, mining the repetition of pitch structure is arguably the most widely investigated task since pitch is the primary element of music.

The pitch structure of a group of musical notes exhibits the internal relationship in between the notes. That is, any two members of these notes may occur simultaneously or successively, and the configuration of these notes defines their relationship. The size (e.g., the number of notes, the temporal duration) of repeated pitch structures may vary widely, ranging from a small segment consisting of a few notes, such as a musical motif, to a whole section of a work containing hundreds of notes, such as the exposition of a sonata form. When a specific pitch structure repeats, i.e., when it restates itself over a music piece, it tends to be perceived consciously as a whole. A common example is the salient and abundant recurrence of the motif in Beethoven's 5th symphony, as shown in Fig. 1, which signifies that "Fate is knocking at the door." From the perspective of the distinctive recurrence of musical patterns, music scholar V. Kofi Agawu provided exhaustive manual analysis of *topics* in music and demonstrated the role of these characteristic patterns in the structural organization of music [11]. Moreover, psychological experiments based on Agawu's analysis also showed the psychological reality of musical patterns that influences the cognitive representation of music [12]. Because that such recurrence over the course of music can be regarded as a *musical schema* that enhances the dynamic interaction between musical patterns and musical structure [13], a representation that organizes all repeated patterns of different size along time is helpful for one to realize and to analyze the form and structure of music from a micro- to a macro-scale, and vise versa.

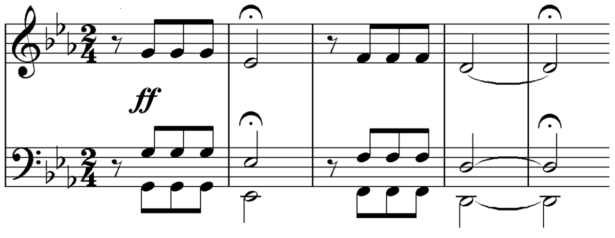


Fig. 1: The motif of Beethoven's 5th Symphony, Op. 67, Mvt. 1, MM. 1-5. The opening motif consists of 2 pitch classes, G and E^b, which form a major third. When the motif recurs right after its first appearance, the two pitch classes change into F and D, forming a minor third.

To this end, we propose the *musical schemagram*, a time-scale representation that employs a geometric-based pattern discovery algorithm to illustrate the evolution and occurrence of the repeated patterns along time. The pattern discovery algorithm which originates from [8] is elaborated first and is revised to combine with a clustering method to extract the representative patterns in symbolic musical data. Then, the musical schemagram of a musical piece, as a case study, is

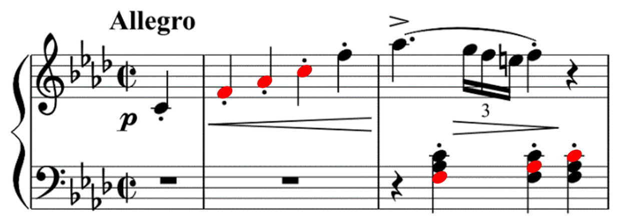


Fig. 2: The opening theme of Beethoven's Piano Sonata No. 1, Op. 2-1, Mvt. 1, MM. 1-2.

represented to demonstrate the comprehensive visualization of musical structure through the musical patterns within the piece.

II. Discovery of Repeated Patterns

Pattern discovery algorithms can be categorized into two approaches: string-based [14]–[16] and geometry-based [8]–[10]. In the field of MIR, the earlier works on musical pattern discovery mainly aimed at finding melodic patterns, hence the music to be processed is represented as a string of notes or as a set of such strings. For monophonic or polyphonic music which has precise voice leading, it is quite simple and direct to encode music as string(s). However, not every music can be represented in the same way with little effort since the voices of music sometime could be indistinct. In addition, there are many patterns other than melodies.

As distinct from the string-based methods, geometric approach represents the music to be analyzed as a multidimensional dataset, i.e. representing all notes of a piece as a set of points in a Euclidean space. Then, the set of points is used to calculate all the displacement vectors[1], i.e. onset differences and pitch differences, constituted by the notes. To put it briefly, geometric approach calculates the displacement vectors and finds repeated patterns by inspecting these vectors. Suppose that there are a pattern A and a repetition of the pattern A′ in a space. Then all notes in A should experience a same displacement when translating to their counterparts in A′, as shown in Fig. 3. In other words, when a pattern repeats, we can always find a group of identical displacement vectors whose initial points form the pattern, and whose terminal points form the repetition of the pattern. Generally speaking, pattern discovery algorithms which adopt geometric approach like variants of SIA described in [8] can find classes of perceptually significant musical patterns that are very difficult to compute with string-based approach.

Although geometric approach allow us to process polyphonic music[2] as simply and efficiently as monophonic music, it suffers from two issues. First, as the result of calculating all displacements within a musical piece in a way of brute-force search strategy, tens of thousands of patterns would be found, and most of them are of little musical interest. Though the variants of SIA attempt to extract only the musically interesting patterns, it turns out that there are still a huge pile of output patterns for a large musical work such as a piano sonata. Second, SIA finds all occurrences of a given pattern in an iterative way and thus has the worst-case time complexity $O(kn^3)$ for a k-dimensional dataset of size n. To tackle the issues mentioned above, the following algorithm adopts the idea of *compactness* to expel probably redundant patterns and utilizes a clustering method to directly get sets of repeated patterns with all repetitions of a pattern being in the same set.

[1]A displacement vector is a geometric object that has magnitude and direction. A displacement vector is frequently represented graphically as an arrow, connecting an initial point with a terminal point. That is, a displacement vector points from an initial point to a terminal point.

[2]The term *polyphonic music* here refers to the music of any kind of musical texture expect the monophonic.

Onset	Pitch
-1.0	60
0.0	65
1.0	68
2.0	72
3.0	77
4.0	80
5.0	53
5.0	56
5.0	60
5.5	79
5.6667	77
5.8333	76
6.0	53
6.0	56
6.0	60
6.0	77
7.0	53
7.0	56
7.0	60

TABLE I: The geometric representation of the musical clip in Fig. 2. Each row in the table indicates a note event.

III. Algorithm for Pattern Discovery[3]

A. Geometric Representation and Displacement Vectors

In the algorithm, a music piece with totally N notes is represented as a list of note events $\mathbf{G} := [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_i, \cdots, \mathbf{g}_N]^T$, $\mathbf{G} \in \mathbb{R}_+^{N\times 2}$, where $\mathbf{g}_i := [o_i; p_i]$, $\mathbf{g}_i \in \mathbb{R}^2$, represents the ith note. A note event has two values: o_i is the onset time (in crotchet beats), and p_i is the pitch number (in MIDI number). These note events are sorted in ascending order, where onset time is the primary sort key and pitch number is the secondary sort key. For instance, the music clip shown in Fig. 2 can be represented as a list shown in TABLE I, where every note is represented by a 2-D vector. As a result, every note $\mathbf{g}_i$ in a musical piece is represented as a point in the 2-D Euclidean space with onset and pitch as its position. This is called the *geometric representation* of a music piece. With such a representation, we compute $\mathbf{d}_{ij}$, the 4-D *displacement vector* pointing from the jth note to the ith note, defined as follows:

$$\mathbf{d}_{ij} = (o_{ji}, p_{ji}, j, i), \tag{1}$$

where $o_{ji} = o_i - o_j$ and $p_{ji} = p_i - p_j$. This means o_{ji} and p_{ji} are the differences of onset and pitch respectively. Note that

[3]The source code can be found at https://github.com/Tsung-Ping/Pattern-Discovery.

the initial point j and the terminal point i of the displacement are saved together with o_{ji} and p_{ji} for better accessibility to these indices. Because that $\mathbf{d}_{ii}$ is a translation to self, and that $\mathbf{d}_{ij}$ and $\mathbf{d}_{ji}$ are of the same magnitude, we only calculate the $\mathbf{d}_{ij}$ for $1 \leq j < i \leq N$.

B. Find groups of identical displacement vectors

After computing the all displacement vectors between any two notes, the next step is to find groups of displacement vectors having equal o_{ji} and p_{ji}. Since the pitch structure of a pattern may vary when repeating, tolerance values are defined so that two displacement vectors with the variance smaller than the tolerance values are considered *identical*. In particular, we set two types of tolerance, *horizontal tolerance* θ_h and *vertical tolerance* θ_v. More specifically, given two *note groups* $\mathbf{A} := \{\mathbf{g}_1, \mathbf{g}_2, \cdots \mathbf{g}_K\}$ and $\mathbf{A}' := \{\mathbf{g}'_1, \mathbf{g}'_2, \cdots, \mathbf{g}'_K\}$, $\mathbf{A}, \mathbf{A}' \subset \mathbf{G}$, and $\mathbf{A} \neq \mathbf{A}'$, then $\mathbf{A}$ and $\mathbf{A}'$ are two repeated patterns if for every $\mathbf{g}_j \in \mathbf{A}$, there exists one $\mathbf{g}'_i \in \mathbf{A}'$ and two constants $(\Delta o, \Delta p)$ such that

$$|o_{ji} - \Delta o| \leq \theta_h \quad \text{(horizontal tolerance)} \tag{2}$$

$$|p_{ji} - \Delta p| \leq \theta_v \quad \text{(vertical tolerance)} \tag{3}$$

The above conditions are implemented as follows. First, the displacement vectors are sorted according to the onset difference o_{ji}, and those vectors with their onset time differing within the horizontal tolerance are grouped together. Second, each group of vectors from previous step are sorted again based on the pitch difference p_{ji}, and similarly, the vectors whose pitch differences differ within the vertical tolerance are grouped together.

After the two steps, the displacement vectors are divided into small groups with all vectors within a group being identical. And the initial points and the terminal points of each group form a pattern and its repetition respectively. However, a vector in a group might not be regarded as a member of that group even if the vector is identical to the other vectors. The reason is that the initial point of the vector may be far from those of the other vectors, as the note p_3 in Fig. 3. Therefore, we further adopt the *adjacent tolerance* θ_a to prevent such a situation. Technically, the vectors in each group are sorted by their initial points; if the onset difference between any two initial points is larger than θ_a, that is,

$$o_{k+1} - o_k > \theta_a \quad \text{(adjacent tolerance)}, \tag{4}$$

the group will be split into two new groups. In this paper, we set $\theta_h = 0.1$, $\theta_v = 1$, and $\theta_a = 2$, respectively.

C. Remove patterns of low compactness

Having the groups being established, the initial points and the terminal points of the vectors in each group will be draw out respectively to construct a pattern and a repetition of this pattern. That is, a pair of patterns can be obtained out of each group by reading the initial and terminal points off from the vectors in the group. Nonetheless, a lot of patterns found in this way just coincidentally share identical displacement vectors and should not be considered musically meaningful.

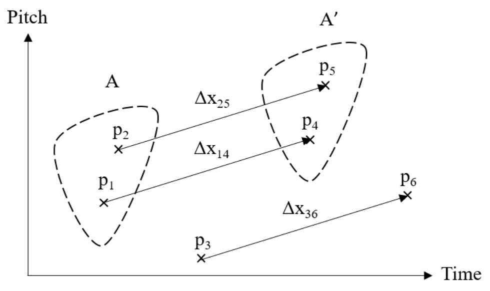


Fig. 3: The translation of a pattern. The two notes p_1 and p_2 in pattern A will experience a same displacement Δx when translating to their counterparts p_4 and p_5 in pattern A′. That is, Δx_{14} is equal to Δx_{25}. Note that p_3 and p_6 are outsiders for A and A′, i.e., they do not belong to A and A′, although Δx_{36} is equal to Δx_{14} and Δx_{25}.

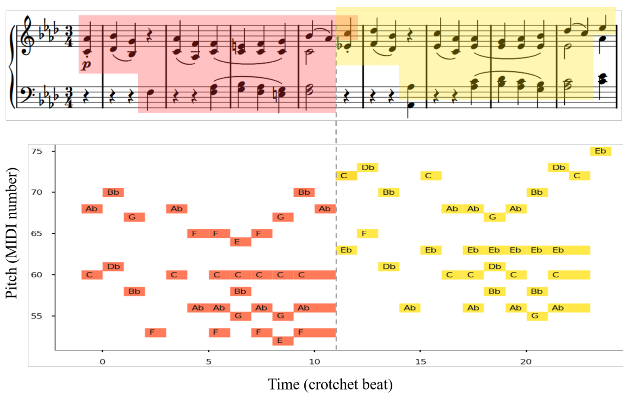


Fig. 4: One of the output patterns for the Beethoven's Piano Sonata No. 1, Op. 2-1, Mvt. 3, MM. 1-8. The excerpt shows the first and the second appearances of the pattern, marked in red and in yellow respectively.

For instance, we may find a pair of patterns from the example shown in Fig. 2: the three notes marked in red at measure 2 are exactly a repetition of the three notes marked in red at measure 1; this repetition is semantically non-meaningful and should be expelled by the algorithm.

To filter out this kind of patterns, we add a constraint on the *compactness* of each pattern and eliminate those patterns of low compactness. The compactness of a patten, as defined in [8], is the ratio of the number of notes in the pattern to the total number of notes that occur within the region spanned by the pattern. For a pattern $\mathbf{P}$ containing N notes with $\mathbf{g}_f$ and $\mathbf{g}_l$ respectively as its first note and last note, the compactness of the pattern C_P is computed as:

$$C_P = \frac{N}{l - f + 1} \tag{5}$$

The idea of inspecting the compactness is that a coincidently repeated pattern often consists of notes which distribute sparsely over space. The threshold for compactness is set to

Data	P_{est}	R_{est}	$F1_{est}$	P_{occ}	R_{occ}	$F1_{occ}$	P_{TL}	R_{TL}	$F1_{TL}$
Bach	0.76003	0.74567	0.75278	0.36535	0.36535	0.36535	0.37077	0.31691	0.34173
Beethoven	0.67776	0.81890	0.74168	0.84026	0.80010	0.81969	0.64948	0.79493	0.71488
Chopin	0.45012	0.52498	0.48467	0.98925	0.98925	0.98925	0.51173	0.55465	0.53233
Gibbons	0.80000	0.48676	0.60525	0.02462	0.02462	0.02462	0.11289	0.07713	0.09165
Mozart	0.61332	0.63742	0.62514	0.93882	0.86827	0.90217	0.59863	0.63540	0.61647
Average	0.66025	0.64275	0.64190	0.63166	0.60952	0.62022	0.44870	0.47580	0.45941

TABLE II: Evaluation on JKU-PDD.

be the mean of total compactness plus 2 standard deviations. Moreover, since that there are usually *isolated membership* (i.e., fragmentary notes) attached to extracted patterns, we further employ a *trawling* method to trim off the patterns [9]. Explicitly speaking, the algorithm goes through each pattern from both ends, returning a subset of the pattern that has a compactness greater than a threshold t and that contain at least n points.

D. Cluster the patterns

To find all occurrences of each pattern, we employ an unsupervised clustering method, the Density-based Spatial Clustering of Applications with Noise (DBSCAN) [17]. The algorithm first constructs the pitch structure of each pattern, and then calculates the Levenshtein distances [18] between pitch structures as the distance metrics for later clustering. The pitch structure of a pattern is represented as all the displacement vectors within the pattern. The DBSCAN method is a density-based clustering algorithm which does not require to predetermine the number of clusters and at the same time can leave data in sparse regions to be classified as noise. This benefits the task of clustering patterns since most of the time we have no idea of how to choose the number of repeated patterns. As a result, the patterns that are assigned to a cluster are considered a repeated pattern, and the patterns that are not assigned to a cluster are discarded.

IV. Evaluation

A. Pattern Discovery

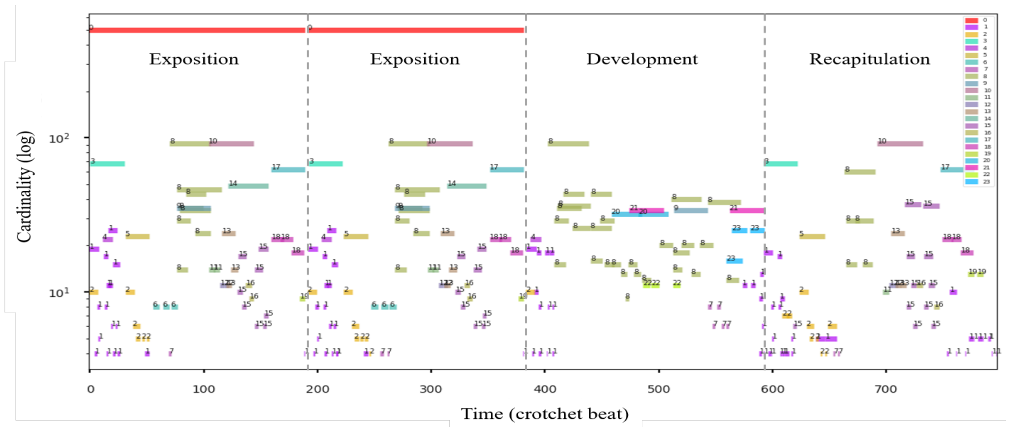
We evaluate the proposed algorithm on the JKU Patterns Development Database (JKU-PDD) [19], which is the database of classical music for the pattern discovery task of the MIREX evaluation campaign. This task is evaluated against the following metrics: establishment precision (P_{est}), establishment recall (R_{est}), and establishment F1 score ($F1_{est}$); occurrence precision (P_{occ}), occurrence recall (R_{occ}), and occurrence F1 score ($F1_{occ}$); three-layer precision (P_{TL}), three-layer recall (R_{TL}), and three-layer F1 score ($F1_{TL}$). In short, the establishment metrics focus on the ability to find out at least one occurrence of a repeated pattern, the occurrence metrics measure the capacity of retrieving all the occurrences of a repeated pattern, and the three-layer metrics test the overall similarity between the output patterns and the ground truth [20]. Comprehensive definition of these metrics can be found in [21].

Experiment results listed in TABLE II indicate that the performance of the pattern discovery algorithm varies a lot over different musical styles. For instance, for Bach's fugue, the algorithm reached the highest establishment F1 score 0.75278 but a relatively low occurrence F1 score 0.36535, while for Chopin's mazurka, the algorithm got the lowest establishment F1 score 0.48467 but the highest occurrence F1 score 0.98925. In other words, the proposed algorithm has a good ability to establish at least one occurrence of the patterns in the fugue, but has some difficulties in finding all their occurrences. And it is just opposite to the case of the mazurka. This is partly because that the *subject* and the *countersubjects*[4] of the fugue appear successively and then get intertwined subsequently so that it is easy to discover them at the beginning of the piece but hard to retrieve one of them from the others over the course of the music. On the other hand, the algorithm found many patterns not identified in the ground truth as the mazurka is the largest piece in the dataset with more than 2,000 notes, and such a result decreased the establishment F1 score. In terms of overall performance, the algorithm got the highest three-layer F1 score 0.71488 for Beethoven's piano sonata, and both its establishment and occurrence F1 scores are above the average. Nevertheless, the output patterns of the piano sonata disclosed some limitations of the algorithm. Take the opening section of the piece as an example, the algorithm established the red pattern in Fig. 4, with the note C5 at measure 4, beat 3 attached to the end of the pattern. This note surely translates a third upwards to the yellow one along with the red pattern; however, the red pattern semantically ends with a cadence at measure 4, beat 2. In such a case, the semantically redundant note is hard to expelled by the algorithm. This algorithm has been tested in the MIREX 2017 competition on Discovery of Repeated Patterns and Themes and its performance is comparable with other submissions to the competition. Detailed results can be found in http://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections_Results.

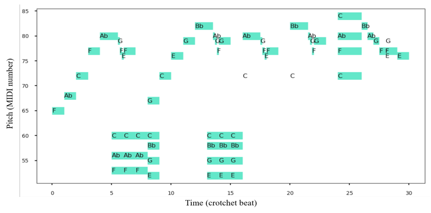
B. Schemagram: The Illustration of Musical Structure

Most of the demonstrations of musical structure display the different sections of a musical piece, usually with some symbols or names attached to them. This kind of representation can easily show the overall plan of a musical piece, yet fail to explain by itself that why these sections are built and how the boundary of a section is determined. Due to the fact that the

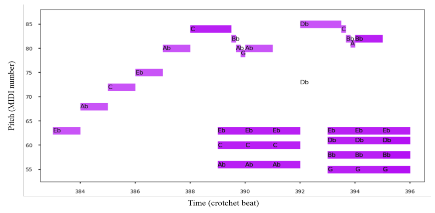
[4]In a fugue, a subject is the material, usually a recognizable melody, upon which part or all of a composition is based. And a countersubject is the material proposed after the subject occurs that figures prominently but is subordinate in importance to the subject.



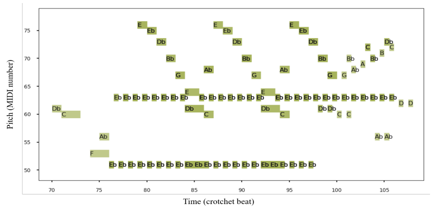
(a) The schemagram of Beethoven's Piano Sonata No. 1, Op. 2-1, Mvt. 1.



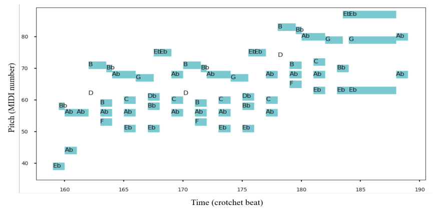
(b) Cluster #3 in Fig. 5a.



(c) Cluster #1 in Fig. 5a.



(d) Cluster #8 in Fig. 5a.



(e) Cluster #17 in Fig. 5a.

Fig. 5: (a) The schemagram shows the occurrences of each repeated pattern along the time axis. Each line in the schemagram indicates one occurrence of a pattern, and lines of the same color stand for patterns in the same cluster. The length of the lines exhibits the time span of a pattern, and the hight of the lines signifies the number of notes in a pattern. (b) The 1st theme of the piano sonata. (c) The transposed 1st theme. (d) The 2nd theme of the piano sonata. (e) The coda of the piano sonata.

cognition of musical structure relies greatly on the repeated patterns, we take advantage of the pattern discovery algorithm to find repeated patterns of a music and plot these repeated patterns all together in one diagram to better illustrate the structure of the music. This representation of musical patterns as well as musical structure can be regarded a kind of *schema* which helps us organize and make sense of information, and thus we call the diagram *schemagram*.

To construct the schemagram, the patterns extracted by our algorithm are clustered first, and a number is assigned to each cluster according to the chronological order of their first appearances. The larger the number is, the latter the cluster appears. Then, we illustrate the period of time spanned by each pattern along the time axis and indicate the cardinality (i.e., the number of notes) of each pattern. In consequence, the schemagram is a time-cardinality representation, where the time axis is horizontal and the cardinality axis is vertical. And musical patterns in the schemagram are shown as horizontal bars of different colors and different lengths.

The schemagram of a musical piece chronologically depicts the occurrence of repeated patterns of different semantic levels, including the motif, theme, and even an entire repeated section. As the repetitions of various patterns collaboratively shape the sense of structure, the schemagram can interpret the abstract construction of the musical structure in a more concrete and comprehensive way. A glance at the schemagram of the 1st movement from Beethoven's Piano Sonata No. 1 manifests the idea. As shown in Fig. 5a, the sections of sonata form,

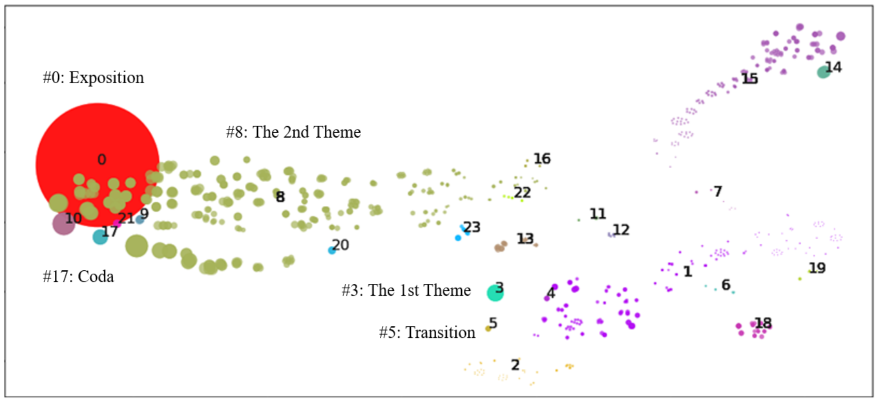


Fig. 6: The spatial relationships of the patterns in Beethoven's Piano Sonata No. 1, Op.2-1, Mvt. 1. Each dot indicates a pattern, and the size of the dots is proportional to the cardinality of the patterns.

i.e., the exposition (and its repetition), the development, and the recapitulation, are clearly delineated by the patterns of the piece. The exposition presents the 1st theme (#3 at time zero, Fig. 5b), then modulates to the 2nd theme which is in a contrasting style (#8 at time 79, Fig. 5d), and concludes with a coda (#17 at time 162, Fig. 5e). After the exposition repeats, the development briefly restates the 1st theme in a transposed way (#1 at time 383, Fig. 5c) and immediately jumps to the 2nd theme (#8 at time 411) to explore its harmonic and textural possibilities almost throughout the left of the development, as we can see that there are many patterns assigned #8 in this section. At the end of the development, the music returns to the tonic key, and some fragments of the 1st theme (patterns assigned #1 around the time period 570-590) appear to signal the crucial moment of a sonata form, that is, the return of the 1st theme. As an altered repeat of the exposition, the recapitulation comes out with the 1st theme (#3 at time 592) and has the pattern distribution similar to that of the exposition.

Finally, we employ a dimension reduction method, the t-Distributed Stochastic Neighbor Embedding (t-SNE) [22], to further demonstrate the relationship among these patterns according to their the similarity. Fig. 6 illustrates the relative spatial relationship among these patterns on a two-dimensional plane, in terms of the Levenshtein distance mentioned in Section III-D. The figure shows that, first, the transition part of the music piece is close to the 1st theme, since some materials in the 1st theme are reused in the transition; second, the 2nd theme is far from the 1st theme, probably because the two themes are contrary to each other in the convention of composition. Moreover, the coda is close to both the 2nd theme and the exposition for it functions as the continuation of the 2nd theme and as the connection with the repetition of the exposition.

It is worth mentioning that, for this piano sonata, the 2nd theme is much closer to the exposition than the 1st theme is, although both of them are definitely essential parts of the exposition. Such a phenomenon can be explained in several ways: first, the cardinality of the 2nd theme is larger than that of the 1st theme; second, the descending intervals (i.e., the descending intervals in # 8) are characteristic of both the the 2nd theme and the whole exposition; third, the 2nd theme gets more emphasis in the development section.

From the above observations, we summarize that the schemagram can not only portray the outline of musical structure, but can further elaborate the subtleties among structural sections in terms of the repeated patterns that make a section distinct from the others on the one hand while integrate these different sections together on the other hand. Furthermore, the organization of repeated patterns in a large scale can be further used to study the functions of repeated patterns as a way of establishing musical form or the *agency* of repeated patterns to mediate with the structure.

V. Conclusions

We propose the geometry-based pattern discovery algorithm combining the clustering method DBSCAN to efficiently find the patterns and their recurrences within a musical piece. The evaluation on the JKU-PDD indicates the comparability of the algorithm with the previous works in MIREX while also shows the limitations for automated pattern discovery. To improve the performance of the algorithm, a further study on the descriptive knowledge of musically meaningful patterns is required. Besides, the schemagram, a novel visualization of repeated patterns, is introduced to demonstrate the roles of repeated patterns in forming musical structure. An elaboration on the schemagram of a piano sonata exhibits the interpretability of such representation. The pattern discovery algorithm along with the schemagram is useful for automated music analysis by which the investigations in both the intra-opus and inter-opus scenarios may dig out more subtleties of music.

References

[1] Heinrich Schenker, *Harmony*, University of Chicago Press, London, 1954. Edited by Oswald Jonas and translated by Elisabeth Mann-Borgese from the 1906 German edition.
[2] Leonard B. Meyer, *Emotion and Meaning in Music*, Chicago University Press, 1956.

[3] Fred Lerdahl and Ray Jackendoff, *A Generative Theory of Tonal Music*, MIT Press, Cambridge, MA, 1983.
[4] Emiru Tsunoo, Nobutaka Ono, and Shigeki Sagayama, "Rhythm map: Extraction of unit rhythmic patterns and analysis of rhythmic structure from music acoustic signals," in *ICASSP*, pp. 185-188, 2009.
[5] Florian Krebs, Sebastian Böck, and Gerhard Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *ISMIR*, pp. 227-232, 2013.
[6] Geoffroy Peeters, "Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach," in *ISMIR, 2007*, pp. 35-40.
[7] Nanzhu Jiang and Meinard Müller, "Automated methods for analyzing music recordings in sonata form," in *ISMIR*, pp. 595-600, 2007.
[8] David Meredith, Kjell Lemström, and Geraint A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," in *Journal of New Music Research*, Vol. 31, No. 4, p. 321-345, 2002.
[9] Tom Collins, Jeremy Thurlow, Robin Laney, Alistair Willis, and Paul H. Garthwaite, "A comparative evaluation of algorithms for discovering translational patterns in Baroque keyboard works," in *ISMIR*, pp. 9-13, 2010.
[10] 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*, pp. 549-554, 2013.
[11] Victor Kofi Agawu, *Playing with signs,* Princeton, NT: Princeton University Press, 1991.
[12] Carol L. Krumhansl, "Topic in Music: An Empirical Study of Memorability, Openness, and Emotion in Mozart's String Quintet in C Major and Beethoven's String Quartet in A Minor," in *Music Perception*, Vol. 16, No. 1, Language and Music Processingpp. pp. 119-134, 1998.
[13] Carol L. Krumhansl and Mary A. Castellano, "Dynamic processes in music perception," in *Memory and Cognition*, pp. 325-334, 1983.
[14] Jia-Lien Hsu, Chin-Chin Liu, and Arbee L. P. Chen, "Efficient repeating pattern finding in music databases," in *CIKM*, pp. 281-288, 1998.
[15] Pierre-Yves Rolland, "Discovering patterns in musical sequences," in *Journal of New Music Research*, pp. 334-350, 1999.
[16] Darrell Conklin and Christina Anagnostopoulou, " Representation and discovery of multiple viewpoint patterns," in *ICMC*, pp. 479-485, 2001.
[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231, 1996.
[18] Heikki Hyyrö, "Explaining and extending the bit-parallel approximate string matching algorithm of Myers," Technical report, University of Tampere, Finland, 2001.
[19] Tom Collins, *JKU Patterns Development Database*, 2013. Available at http://tomcollinsresearch.net/research/data/mirex/JKUPDD-Aug2013.zip
[20] David Meredith, *Computational music analysis*, Springer, 2016.
[21] Tom Collins and David Meredith, *2017: Discovery of Repeated Themes & Sections*, 2017. http://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections
[22] Laurens van der Maaten and Geoffrey Hinton, "Visualizing high-dimensional data Using t-SNE," in *Journal of Machine Learning Research*, pp.2579-2605, 2008.