ALIGNED HIERARCHIES: A MULTI-SCALE STRUCTURE-BASED REPRESENTATION FOR MUSIC-BASED DATA STREAMS

Katherine M. Kinnaird
Department of Mathematics, Statistics, and Computer Science
Macalester College, Saint Paul, Minnesota, USA
kkinnair@macalester.edu

ABSTRACT

We introduce *aligned hierarchies*, a low-dimensional representation for music-based data streams, such as recordings of songs or digitized representations of scores. The aligned hierarchies encode all hierarchical decompositions of repeated elements from a high-dimensional and noisy music-based data stream into one object. These aligned hierarchies can be embedded into a classification space with a natural notion of distance. We construct the aligned hierarchies by finding, encoding, and synthesizing all repeated structure present in a music-based data stream. For a data set of digitized scores, we conducted experiments addressing the fingerprint task that achieved perfect precision-recall values. These experiments provide an initial proof of concept for the aligned hierarchies addressing MIR tasks.

1. INTRODUCTION

From Foote's field-shifting introduction of the self-similarity matrix visualization for music-based data streams in [9] to the enhanced matrix representations in [11, 17] and hierarchical segmentations in [14, 18, 21], music information retrieval (MIR) researchers have been creating and using representations for music-based data streams in pursuit of addressing a variety of MIR tasks, including structure tasks [10, 14, 17, 18], comparison tasks [2–4, 11], and the beat tracking task [1, 5, 8, 13]. These representations are often tailored to a particular task, limited to a single layer of information, or committed to a single decomposition of structure. As a result most of the representations for music-based data streams provide narrow insight into the content of the data stream they represent.

In this work, we introduce *aligned hierarchies*, a novel representation that encodes multi-scale pattern information and overlays all *hierarchical* decompositions of those patterns onto one object by *aligning* [1] these hierarchical decompositions along a common time axis. This representation uncovers repeated structures observed in matrix representations widely used in MIR (such as self-similarity matrices, self-dissimilarity matrices, and recurrence plots) and can be used to visualize all decompositions of the repeated structures present in a particular music-based data stream as well as the relationships between the repeats in that data stream. By including and aligning all repeated structures found in a music-based data stream, the aligned hierarchies exist in the middle ground of representations between the density of information in Foote's self-similarity matrix visualization [9] and the sparsity of information in representations like those found in [1, 11, 14, 20].

Beyond the visualization benefits, aligned hierarchies have several compelling properties. Unlike many representations in the literature, the aligned hierarchies can be embedded into a classification space with a natural distance function. This distance function serves as the basis for comparing two music-based data streams by measuring the total dissimilarity of the patterns present. Additionally, the aligned hierarchies can be post-processed to narrow our exploration of a music-based data stream to certain lengths of structure, or to address numerous MIR tasks, including the cover song task, the segmentation task, and the chorus detection task. Such post-processing techniques are not the focus of this paper and will be explored further in future work. In this paper, as a proof of concept for our approach to MIR comparison tasks, we use aligned hierarchies to perform experiments addressing the fingerprint task on a data set of digitized scores.

There are previous structure-based approaches to the cover song task, such as [1, 11, 20], that do not use the formal segmentation of pieces of music and instead, use enhanced matrix representations of songs as the basis of their comparisons. Like those in [9], these representations compare the entire song to itself, but fail to intuitively show detailed structural decompositions of each song. In [2–4], a variety of music comparison tasks are addressed by developing a method of comparison based on audio shingles, which encode local information. In this work, we use audio shingles as the feature vectors to form the self-dissimilarity matrices representing the scores in the data set.

In Section 2, we introduce the aligned hierarchies and the algorithm that builds them. In Section 3, we define the classification space that aligned hierarchies embed into and the associated distance function. In Section 4, we report on experiments using aligned hierarchies to address

[1] We note that 'alignment' in this case refers to placing found structure along a common axis, not to matching a score to the recording of a piece.

the fingerprint task for a data set of digitized scores, and we summarize our contributions in Section 5.

2. ALIGNED HIERARCHIES

In this section, we define the aligned hierarchies for a music-based data stream and present a motivating example. We introduce the three phases for constructing the aligned hierarchies with discussions about the purpose and motivation for each phase. For simplicity, we will use 'song' to refer to any kind of music-based data stream.

The algorithm finds meaningful repetitive structure in a song from the self-dissimilarity matrix representing that song. The algorithm aligns all possible hierarchies of that structure into one object, called the ***aligned hierarchies*** of the song. The aligned hierarchies H has three components: the onset matrix B_H with the length vector w_H and annotation vector α_H that together act as a key for B_H.

The ***onset matrix*** B_H is an $(n \times s)$-binary matrix, where s is the number of time steps in the song, and where n is the number of distinct kinds of repeated structure found in the song. We define B_H as follows

$$(B_H)_{i,j} = \begin{cases} 1 & \text{if an instance of } i^{\text{th}} \text{ repeated} \\ & \text{structure begins at time step } j, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The ***length vector*** w_H records the lengths of the repeated structure captured in the rows of B_H in terms of number of time steps, and the ***annotation vector*** α_H records the labels for the groups of repeated structure encoded in these rows. These labels restart at 1 for each distinct length of repeated structure and serve to distinguish groups of repeated structure with the same length from each other. We note that we can exchange any two rows in B_H representing repeats with the same length without losing any information stored in the aligned hierarchies and without changing either w_H or α_H.

2.1 Motivating Example

Suppose we have a song that has two kinds of non-overlapping repetitive structure, such as a verse and a chorus, denoted V and C, respectively, appearing in the order $VCVCV$. We say that the song has the *segmentation* $VCVCV$, where V and C are the repeated sections. We can segment the song in several ways: $\{V, C, V, C, V\}$, $\{(VC), (VC), V\}$, or $\{V, (CV), (CV)\}$, with (VC) representing the piece of structure composed of the V structure followed by the C structure and similarly for (CV). Noting that both (VC) and (CV) can be decomposed into smaller pieces, we would like to find an object that captures and synthesizes all possible decompositions. Figure 1 is a visualization of one such object where the V structure is 3 beats long and the C structure is 5 beats long.

The object that produces the visualization shown in Figure 1 is known as the aligned hierarchies, and it encodes the occurrences and lengths of all the repeated structure found in a song. In Figure 1, we see that repeats of (VC) and the repeats of (CV) overlap in time, but are not contained in each other. We also note that all decompositions of the repeats of (VC) and (CV) are encoded in this object.

In this example, we have four kinds of repeated structures: V, C, (VC), and (CV). Therefore B_H associated to the aligned hierarchies will have four rows, one corresponding to each kind of repeated structure, and 19 columns, one for each beat. Listing the rows in order of the lengths of the repeated structures and the initial occurrences of those repeats, we have that B_H is a sparse matrix with 1's for the V structure at $\{$(1,1), (1,9), (1,17)$\}$, with 1's for the C structure at $\{$(2,4), (2,12)$\}$, with 1's for the (VC) structure at $\{$(3,1), (3,9)$\}$, and with 1's for the (CV) structure at $\{$(4,4), (4,12)$\}$. Then w_H is the column vector $[3, 5, 8, 8]^t$ and α_H is $[1, 1, 1, 2]^t$.

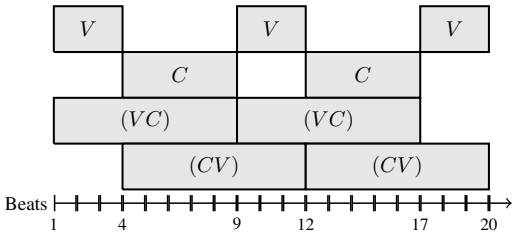


Figure 1: Visualization of aligned hierarchies for a song with segmentation $VCVCV$ incorporating all possible decompositions of the song with V structure 3 beats long and C structure 5 beats long.

2.2 Building the Aligned Hierarchies

The construction of the aligned hierarchies begins with either a self-similarity matrix or a self-dissimilarity matrix. By beginning with a matrix representation for a song, we assume that we do not have access to the original presentation of the song, such as the audio recording, score, or midi file. In a world with proprietary data, extremely high-dimensional data, and limited or restricted access to data, we believe that it is important to develop robust techniques for representing and comparing songs beginning from a data representation that cannot be reverse engineered back to the original presentation of the data. For this work, we will use a self-dissimilarity matrix to represent each song; an example of one for the score of Chopin Mazurka Op. 6, No. 1 is shown in Figure 3a.

Our construction of the aligned hierarchies for a song is motivated by the fact that repeated structures in a song are represented as diagonals of small-valued entries in $\mathcal{D}$, the self-dissimilarity matrix representing the song [6, 16, 17]. If such a diagonal of length k exists in $\mathcal{D}$ beginning at entry (i, j), then the section of the song beginning at time step i that is k time steps long is a repeat of the k time step long section beginning at time step j, and vice versa. We call these sections a pair of repeats of size k.

We construct the aligned hierarchies from simple and meaningful repetitive structure present in a song. For example, suppose a sequence of five chords is played repeatedly in a song. We do not regard repetitions of just the

first three chords as meaningful repeats, unless there is at least one instance in the song of those three chords without the last two or at least one instance of the last two chords without the first three.

Building the aligned hierarchies has three phases:

1. Extract repeated structure of all possible lengths from $\mathcal{D}$, the self-dissimilarity matrix of the song
2. Distill extracted repeated structure into their essential structure components
3. Build aligned hierarchies for the song using the essential structure components

2.2.1 Phase 1 - Extract Repeated Structure from Self-Dissimilarity Matrix $\mathcal{D}$

There are four steps to extracting repeated structure from $\mathcal{D}$. First, we define what repeats are, in context of the data and task at hand. Second, we extract the coarsest repeated structure from $\mathcal{D}$. Third, we use this found structure to uncover further repeated structure hidden by the presentation of the song as $\mathcal{D}$. Lastly, we create groups of repeated structure from the extracted pairs of repeats. In this last step, we enforce a mimicking how humans notice and interpret patterns by removing any group of repeated structure that contains overlapping repeats.

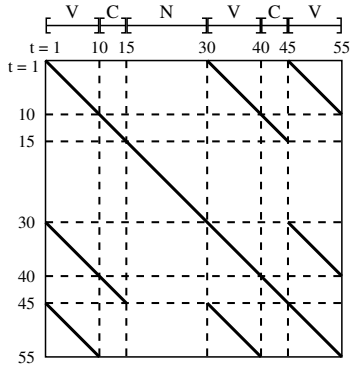
Step 1: Based on the data and the task of interest, we set a threshold T that defines how similar two sections must be in order to be considered repeats of each other and then threshold $\mathcal{D}$ accordingly. We note that many ways exist in the literature to set this threshold, such as [2, 10, 11, 16]. The resulting thresholded matrix $\mathcal{T}$ is a binary matrix of the same dimensions as $\mathcal{D}$ and is given by

$$\mathcal{T}_{i,j} = \begin{cases} 1 & \text{if } \mathcal{D}_{i,j} < T \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Step 2: We next find and extract pairs of coarse repeats in the song, by finding all non-zero diagonals in $\mathcal{T}$, recording relevant information about the pair of repeats, and finally removing the associated diagonal from $\mathcal{T}$. We loop over all possible repeat lengths, beginning with the largest possible structure (the number of columns in $\mathcal{T}$) and ending with 1 (the smallest possible structure).

To find simple and meaningful structure of exactly length k represented by diagonals of exactly length k, we must remove all diagonals of length greater than k. Suppose that we did not remove diagonals of length $(k+1)$ before searching for diagonals of length k, and let $\widehat{d}_{i,j}$ be one such diagonal of 1's in $\mathcal{T}$. Then along with the other diagonals of length k, our algorithm would find two diagonals of length k: one starting at (i, j) and another starting at $(i+1, j+1)$. Our algorithm would not be able to tell that these diagonals of length k are contained in the diagonal $\widehat{d}_{i,j}$ or that together these diagonals make the diagonal $\widehat{d}_{i,j}$. Thus our algorithm would not be finding simple and meaningful repeated structure in the song as required.

Step 3: Once we have extracted all diagonals from $\mathcal{T}$, we use the smaller extracted repeated structure to find additional repeated structures hidden in the coarse repeats. Suppose we examine a piece of text where a certain word is repeated both by itself and in a repeated phrase. In the previous step, our algorithm would find the repeated word on its own and the repeated phrase, but would not detect the repeated word as part of that repeated phrase. In this step, our algorithm realizes that our repeated word is part of the repeated phrase and that the repeated phrase breaks up into at most three pieces, those being: 1) the part of the phrase before our repeated word, 2) the repeated word itself, and 3) the part of the phrase after the repeated word.



(a) $\mathcal{T}$ for a toy song with sections marked

	Start Time Step	Start Time Step	Repeat Length
VC	1	31	15
V	1	46	10
V	31	46	10

(b) Pairs of repeats after Step 2 of Phase 1, the initial extraction from $\mathcal{T}$ with sections marked

	Start Time Step	Start Time Step	Repeat Length
VC	1	31	15
V	1	46	10
V	31	46	10
V	1	31	10
C	11	41	5

(c) Pairs of repeats after Step 3 of Phase 1, the second part of extraction with sections marked

Figure 2: Thresholded matrix $\mathcal{T}$ and the pairs of repeats uncovered after each step of repeat extraction for toy song with segmentation $VCNVCV$

Consider the song with segmentation $VCNVCV$ and with the thresholded distance matrix $\mathcal{T}$ shown in Figure 2a. In the initial extraction, the algorithm finds three pairs of repeats, two pairs encoding repeats of the V structure by itself and one pair encoding two repeats of (VC), as shown in Table 2b. But the algorithm has not detected that the pair of (VC) repeats contain the smaller found V structure as well as the yet to be isolated C structure. In this step, as shown in Table 2c, by using either of the pairs of V repeats, we find that the pair of (VC) repeats does contain a pair of V repeats as well as a pair of smaller repeats that is not the same as the V structure, known as the C structure.

Step 4: In the last step of this phase, we form groups of repeats from the pairs of repeats such that each kind of repeated structure has exactly one group of repeats associ-

ated to it. For the example shown in Figure 2a, we have three groups: one associated to (VC), another associated to V, and a third associated to C.

To mimic human segmentation of music, we check that each group does not contain repeats that overlap in time. For the example in Section 2.1, we are more likely to describe the structure of a popular song by saying "the verse and chorus are repeated twice together followed by the verse again," than by saying "the verse, chorus, and verse are repeated together twice such that those two repeats overlap at one verse." Thus we do not encode the repeated structure (VCV) in the aligned hierarchies shown in Figure 1, even though it occurs twice in $VCVCV$.

2.2.2 Phase 2 - Distill Essential Structure Components

Just like words that are composed of syllables, musical elements, such as motifs and chord progressions, are composed of smaller components. In this step, we distill repeats of the song into their *essential structure components*, the building blocks that form every repeat in the song.

By definition, we allow each time step to be contained in at most one of the song's essential structure components. In this phase, we pairwise compare groups of repeats, checking if the repeats in a given pair of groups overlap in time. If they do, we divide the repeats in a similar fashion as used in Step 3 in Phase 1, forming new groups of repeats that do not overlap in time. We iterate this process, dividing repeats as necessary, until each time step is contained in at most one repeated structure. The repeats remaining at the end of this phase are our essential structure components. For the example in Section 2.1 shown in Figure 1, the essential structure components are the instances of the V and C structures. Figure 3b is a visualization of the essential structure components for the score of Chopin's Mazurka Op. 6, No. 1.

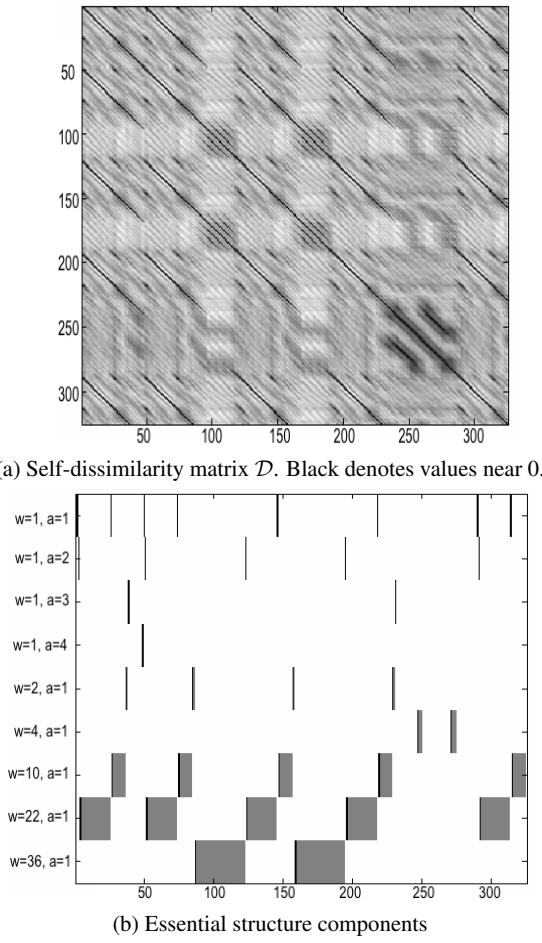
2.2.3 Phase 3 - Construct Aligned Hierarchies from Essential Structure Components

In this final phase, we build the aligned hierarchies from the essential structure components. We employ a process that is akin to taking right and left unions of the essential structure components to find all possible non-overlapping repeats in the song. We encode these repeats in the onset matrix and form the length and annotation vectors that together are the key for the onset matrix. Figure 4 is a visualization of the aligned hierarchies for a score of Chopin's Mazurka Op. 6, No. 1.

3. COMPARING ALIGNED HIERARCHIES

To compare aligned hierarchies, we embed them into a classification space with a distance function measuring the total dissimilarity between pairs of songs. In Section 3.1, we explain how aligned hierarchies embed into this classification space, and we present the distance function used for comparing aligned hierarchies of songs in Section 3.2. [2]

[2] The proofs for the material in this section can be found in the author's doctoral thesis [12].



(a) Self-dissimilarity matrix $\mathcal{D}$. Black denotes values near 0.

(b) Essential structure components

Figure 3: Visualizations for a score of Chopin's Mazurka Op. 6, No. 1 with repeat markers observed.

3.1 Classification Space for Aligned Hierarchies

To define $(\mathcal{S}^*)^n$, the space that we embed aligned hierarchies into, while simultaneously demonstrating how this embedding occurs, we begin by representing aligned hierarchies as a sequence of matrices.

Definition 3.1. Given a particular song with s time steps and its aligned hierarchies H, we define a sequence of s binary matrices $\left\{B^k\right\}_{k=1}^{s}$ where the k^{th} binary matrix B^k is the rows of B_H such that $w_H = k$, which are the rows corresponding to repeats of exactly k time steps. If there are no repeats of exactly k time steps, then B^k is a row of s zeros. For brevity, we will use $\left\{B^k\right\}$ for $\left\{B^k\right\}_{k=1}^{s}$.

We note that each binary matrix in $\left\{B^k\right\}$ does not have a pair of vectors acting as a key for it, as we have in H. Our definition of $\left\{B^k\right\}$ naturally encodes the information from w_H in $\left\{B^k\right\}$. Similarly, we construct α_H so that the labels for the groups of repeats restart at 1 for each distinct repeat length l. Thus, for each l, the label corresponding to a row in $B^l \in \left\{B^k\right\}$ is simply that row's index in B^l.

We recall that we can exchange any two rows of B_H with $w_H = l$ without changing the annotation labels. So we say that two matrices encoding repeats of exactly

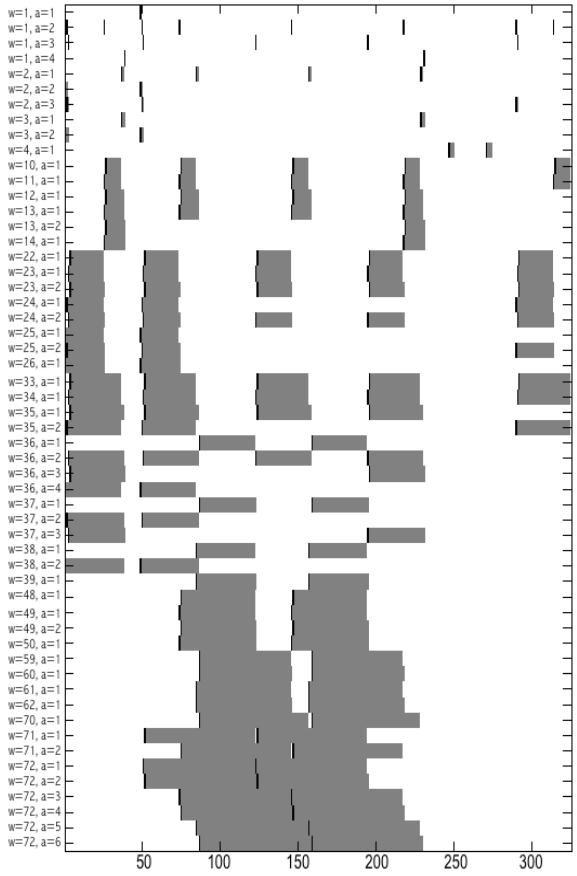


Figure 4: Visualization of aligned hierarchies for a score of Chopin's Mazurka Op. 6, No. 1 with repeat markers observed.

length l are the same if one is a row permutation of the other. Therefore the space that we embed the aligned hierarchies into must inherit this notion of matrix equality.

Definition 3.2. Let $\mathcal{S}$ be the space of $(m \times t)$-binary matrices with $m, t \in \mathbb{Z}_{\geq 1}$. Consider the symmetric group S_m, the group of all permutations for the rows of the matrices. The matrix denoted $M_{\sigma(r)} \in \mathcal{S}$ is the matrix with the rows of $M \in \mathcal{S}$ in the order prescribed by $\sigma(r) \in S_m$.

Proposition 3.1. *Let $\sim$ be the relation on $\mathcal{S}$ such that for $M, Q \in \mathcal{S}$, we say that $M \sim Q$ if $M = Q_{\sigma(r)}$, for some $\sigma(r) \in S_m$. Then $\sim$ is an equivalence relationship on $\mathcal{S}$.*

Definition 3.3. Let $\mathcal{S}^*$ be the quotient space $\mathcal{S}/\sim$. Then the product space $(\mathcal{S}^*)^n$ is composed of n-copies of $\mathcal{S}^*$.

We embed H into $(\mathcal{S}^*)^n$ by setting $t = s$, the number of time steps in the song, and choosing $m = \kappa_{\max}$, where $\kappa_{\max} = \max_{l \in \{1,\ldots,s\}} \left\{ r | r \text{ is the number of rows in } B^l \right\}$. Then we place each $B^l \in \left\{B^k\right\}$ into the l^{th} space of $(\mathcal{S}^*)^n$. So the l^{th} quotient space in $(\mathcal{S}^*)^n$ corresponds to the classification space of patterns of length l present in songs.

Notation: The elements of product space $(\mathcal{S}^*)^n$ are sequences of elements in $\mathcal{S}^*$, pedantically denoted as $(q^g)_{g=1}^n$ with $q^g \in \mathcal{S}^*$, for each $g \in \{1, \ldots, n\}$. For brevity, we will use (q^g) for $(q^g)_{g=1}^n$.

3.2 Metric for Comparing Aligned Hierarchies

To define a metric on the space $(\mathcal{S}^*)^n$ that will measure the total dissimilarity between two songs represented by their aligned hierarchies, we begin by defining a function that measures the dissimilarity between patterns of a *fixed* size present in those two aligned hierarchies.

Definition 3.4. Let $||\cdot||_1$ be the entry-wise 1-norm. Given any $s_1, s_2 \in \mathcal{S}^*$, let $f : \mathcal{S}^* \times \mathcal{S}^* \to \mathbb{R}$ be the function given by

$$f(s_1, s_2) = \min_{\substack{\delta \in s_1 \\ \beta \in s_2}} ||\delta - \beta||_1 \tag{3}$$

Proposition 3.2. *The function $f : \mathcal{S}^* \times \mathcal{S}^* \to \mathbb{R}$ is a distance function.*

To define the metric that measures the *total* dissimilarity between two songs, we use the above function f to compute the dissimilarity between the repeated patterns at each size and total the measured dissimilarities. This gives us the *total* dissimilarity between the repeated patterns of *all* sizes present in two aligned hierarchies.

Corollary 1. *Let $(q^g), (r^g) \in (\mathcal{S}^*)^n$. The function $d_H : (\mathcal{S}^*)^n \times (\mathcal{S}^*)^n \to \mathbb{R}$ is a distance function, where d_H is given by*

$$d_H\left((q^g), (r^g)\right) = \sum_{g=1}^{n} f(q^g, r^g). \tag{4}$$

4. PROOF OF CONCEPT RESULTS

In this section, we consider the fingerprint task for a data set of digitized musical scores. These experiments serve as a proof of concept for our method of comparing songs via their aligned hierarchies. With the exception of the feature extraction, the code implementing the creation and comparison of aligned hierarchies is written in MATLAB. [3]

4.1 Data Set and Features

Our data set is based on 52 Mazurka scores by Chopin. For each score, we download two human-coded, digitized versions, called ***kern files*, posted on the KernScore online database [19]. [4] The first version has the repeated sections repeated as many times as marked in the score and the second has the repeated sections presented only once per time written. For scores that have no sections that are repeated in their entirety, we download the single **kern file twice, marking one copy as having the repetitions repeated and the second copy as having the repetitions not repeated. Each version of a score is referred to as a song and there are 104 songs in our data set.

In this data set, the notion of time is in terms of beats with one time step per beat. For each beat, we extract the chroma feature vector, encoding the amount of each of the 12 Western pitch classes present in that beat [15]. To do

[3] The URL to the code used for the experiments can be found at https://github.com/kmkinnaird/ThesisCode/releases/tag/vT.final2

[4] The **kern files can be accessed at: http://kern.humdrum.org/search?s=t&keyword=Chopin

this, we used the `music21` Python library [7]. [5] We form audio shingles, that encode local contextual information, by concatenating γ consecutive chroma feature vectors, for a fixed integer γ.

We create a symmetric self-dissimilarity matrix $\mathcal{D}$ using a cosine dissimilarity measure between all pairs of audio shingles. Let a_i, a_j be the audio shingles for time steps i and j, respectively. Then we define

$$\mathcal{D}_{i,j} = \left(1 - \frac{< a_i, a_j >}{||a_i||_2||a_j||_2}\right) \quad (5)$$

By setting γ, we set the smallest size of repeated structure that can be detected. For this work, we set $\gamma = 6$ or $\gamma = 12$. Assuming that the average tempo of a Mazurka is approximately 120 beats per minute, if $\gamma = 6$, our shingles encode about 3 seconds of information similar to the audio shingles in [2,3]. Similarly, if $\gamma = 12$, our shingles encode four bars of three beats each or about 6 seconds.

4.2 Evaluation Procedure

For all of our experiments, given a particular threshold, we construct the aligned hierarchies for each score. We compute the pairwise distances between the songs' representations as described in Section 3.2. Using these pairwise distances, we create a network for the data set with the songs in the data set as the nodes. In the fingerprint task, we only match songs that are exact copies of each other, and so we define an edge between two nodes if the distance between the two aligned hierarchies associated to the songs is 0.

We evaluate the results of our experiments by computing the precision-recall values for the resulting network compared against a network representing the ground truth, which is formed by placing edges between the two identical copies of the score present in the data set. This ground truth was informed by a data-key based on human-coded, meta-information about the scores.

For each experiment, we set γ, the width of the audio shingles, and T, the threshold value for defining when two audio shingles are repeats of each other. The choice of γ and T affects the amount of structure classified as repeats, which impacts whether or not a song has aligned hierarchies to represent it. If a song does not have aligned hierarchies, due to the choice of γ and T, we remove the node representing that song from consideration in both our experiment network and in our ground truth network, as there would be nothing for our method to use for comparison.

4.3 Results

We conducted 10 experiments with the threshold $T \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ and with $\gamma \in \{6, 12\}$. Each experiment yielded a perfect precision-recall value. For the experiment with $T = 0.01$ and $\gamma = 12$, we had 5 songs without aligned hierarchies including 2 pairs of songs based on scores without repeated sections. For the experiment with $T = 0.02$ and $\gamma = 12$, we had 1 song without aligned hierarchies, but this song was based on a score with repeated sections and thus, under the fingerprint task, would not be matched to another song in our data set.

We note that our method did discover an error in the data key for the Mazurka scores. According to the human-coded, meta-information, Mazurka Op. 17, No. 1 was classified as having sections marked in the score as being repeated. However, the score of Mazurka Op. 17, No. 1 in fact does not have any sections marked to be repeated. Our algorithm correctly detected this error, and we corrected our version of the data key for these Mazurka scores. The corrected data key is our ground truth, which is what our precision-recall values are based on. To our knowledge, there is no published work using this data set, therefore we cannot provide numerical comparisons between our method and other ones.

For all 10 experiments, we have correctly identified all scores, with aligned hierarchies, that do not have sections marked in the score to be repeated. Based on the construction of the score data set, a perfect recall rate was expected. More interestingly, the perfect precision rate means that we do not falsely match scores using the aligned hierarchies.

5. CONCLUSION

In this paper, we have introduced the aligned hierarchies, an innovative, multi-scale structure-based representation for music-based data streams. The aligned hierarchies provide a novel visualization for repeated structure in music-based data streams. Differing from the literature of enhanced matrix representations, instead of showing a comparison of a data stream to itself, the visualization of the aligned hierarchies synthesizes all possible hierarchical decompositions of that data stream onto one time axis, allowing for a straightforward understanding of the temporal relationships between the repeated structures found in a particular data stream.

The aligned hierarchies also provide a mathematically rigorous method for comparing music-based data streams using this low-dimensional representation. We performed experiments addressing the fingerprint task for data based on digitized scores. These experiments had perfect precision-recall rates and provided a proof of concept for the aligned hierarchies.

In future work, we will develop post-processing techniques for the aligned hierarchies. These techniques will allow us to address additional MIR tasks, such as the cover song task and the chorus detection task. We also will continue to develop the theory and metrics associated with aligned hierarchies and their derivatives.

Acknowledgements

This work is a portion of the author's doctoral thesis [12], which was partially funded by the GK-12 Program at Dartmouth College (NSF award # 0947790). Part of this work was performed while the author was visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation. The author thanks Scott Pauls, Michael Casey, and Dan Ellis for their feedback on early versions of this work.

[5] See http://web.mit.edu/music21/ for information about `music21`.

6. REFERENCES

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

[2] M. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):1015 – 1028, 2008.

[3] M. Casey and M. Slaney. Song intersection by approximate nearest neighbor search. *Proceedings of the International Society for Music Information Retrieval*, pages 144–149, 2006.

[4] M. Casey and M. Slaney. Fast recognition of remixed audio. *2007 IEEE International Conference on Audio, Speech and Signal Processing*, pages IV – 1425 – IV–1428, 2007.

[5] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668– 696, 2008.

[6] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 127 –130, 2003.

[7] M. S. Cuthbert and C. Ariza. `music21`: A toolkit for computer-aided musicology and symbolic music data. *11th International Society for Music Information Retrieval Conference*, 2010.

[8] D.P.W. Ellis and G.E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages IV – 1429–1432, 2007.

[9] J. Foote. Visualizing music and audio using self-similarity. *Proc. ACM Multimedia 99*, pages 77–80, 1999.

[10] M. Goto. A chorus-section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1783–1794, 2006.

[11] P. Grosche, J. Serrà, M. Müller, and J.Ll. Arcos. Structure-based audio fingerprinting for music retrieval. *13th International Society for Music Information Retrieval Conference*, 2012.

[12] K. M. Kinnaird. *Aligned Hierarchies for Sequential Data*. PhD thesis, Dartmouth College, 2014.

[13] B. McFee and D. P. W. Ellis. Better beat tracking through robust onset aggregation. In *International conference on acoustics, speech and signal processing*, ICASSP, 2014.

[14] B. McFee and D. P. W. Ellis. Learning to segment songs with ordinal linear discriminant analysis. In *International conference on acoustics, speech and signal processing*, ICASSP, 2014.

[15] M. Müller and S. Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. *12th International Society for Music Information Retrieval Conference*, 2011.

[16] M. Müller, P. Grosche, and N. Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. *12th International Society for Music Information Retrieval Conference*, pages 615–620, 2011.

[17] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. *11th International Society for Music Information Retrieval Conference*, pages 625–636, 2010.

[18] C. Rhodes and M. Casey. Algorithms for determining and labelling approximate hierarchical self-similarity. *8th International Society for Music Information Retrieval Conference*, 2007.

[19] C.S. Sapp. Online database of scores in the humdrum file format. *Proceedings of the International Society for Music Information Retrieval*, pages 664–665, 2005.

[20] D.F. Silva, H. Papadopoulos, G.E.A.P.A. Batista, and D.P.W. Ellis. A video compression-based approach to measure music structure similarity. *Proceedings of the International Society for Music Information Retrieval*, pages 95–100, 2013.

[21] K. Yoshii and M. Goto. Unsupervised music understanding based on nonparametric bayesian models. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5353–5356, 2012.