

# ANALYSIS BY COMPRESSION: AUTOMATIC GENERATION OF COMPACT GEOMETRIC ENCODINGS OF MUSICAL OBJECTS

David Meredith

Aalborg University

dave@titanmusic.com

## ABSTRACT

A computational approach to music analysis is presented, based on the compression of point-set representations of musical works. The approach relates closely to the theory of Kolmogorov complexity and to psychological coding theories of perceptual organisation. A sketch of a model of musical learning based on this approach is given and it is shown how the model accounts in principle for differences between individuals in how pieces of music are understood. The approach is implemented in a greedy compression algorithm, called COSIATEC, which partitions a point-set into the covered sets of translational equivalence classes of maximal translatable patterns. The analyses generated by COSIATEC on five fugues by J. S. Bach are presented and discussed. These analyses demonstrate the potential of the approach for automatically discovering musical patterns of thematic and structural importance.

## 1. INTRODUCTION

The research presented in this paper is founded on the assumption that the goal of music analysis is to find the *best possible explanations* for musical works. This assumption immediately begs an obvious question: given two analyses of the same work, how are we supposed to decide which of the two is “better”? If we are unable to specify how we are to make this decision, then one could argue that the goal of finding the “best possible” explanations is meaningless.

Most musicologists and music analysts do not use an effective procedure or unambiguously defined algorithm for deciding which of two possible analyses they find to be superior: typically, an analyst will prefer an analysis that makes him or her *feel* that he or she has a better understanding of the piece under consideration. In other words, analysts traditionally evaluate musical analyses on subjective, even aesthetic grounds—much like the musical masterpieces that form the subjects of their

analyses.

However, I believe it is possible in principle to define reasonable, *objective* criteria for deciding which of two analyses of a given piece is the “better” one. I would like to suggest that one can reasonably define one analysis of a piece to be “better than” another one of the same piece if the first allows one to more effectively and/or efficiently carry out objectively evaluable musical tasks such as:

- memorising the piece, e.g., in order to be able to perform it without a score;
- identifying errors in a score or performance of the piece or other related pieces;
- correctly identifying the composer, place of composition, genre, form, etc. of the piece or other related pieces; or
- predicting what will come next or what came before in a piece, having been presented with only part of it.

To this extent, it therefore makes sense to suggest that one analysis of a given piece might be considered “better than” another, and that the goal of music analysis is to find the best possible explanations for pieces of music.

Given this assumption concerning the goal of music analysis, the approach presented in this paper is based on the further hypothesis that the best possible explanations for a given musical work are those that

1. are as simple as possible;
2. account for as much of the detailed structure of the work as possible; and
3. set the work in as broad a context as possible.

Clearly, these goals often conflict: accounting for the structure of a piece in more detail typically entails making one’s explanation or analysis more complex; while accounting for a piece in a broader context may entail a less parsimonious description than if the piece is considered in isolation.

In the remainder of this paper, I will explore the implications of this hypothesis and relate it to the theory of Kolmogorov complexity [1,2] and to research in psychological coding theories [3–6]. I will propose that a musical analysis can be modelled as an algorithm or computer program and that the length of this algorithmic representation can be used as an indication of the quality

of the analysis. I will also sketch a model of music perception and learning based on the idea of accounting for the structure of a newly experienced musical “object” by minimally modifying an existing explanation of a collection of previously encountered musical “objects”. Finally, I will briefly describe a greedy compression algorithm that seems to successfully model certain aspects of the cognition of musical structure.

## 2. REPRESENTING A MUSICAL ANALYSIS AS A COMPUTER PROGRAM

I would like to propose that a musical analysis can fruitfully be represented or encoded as a computer program or algorithm that generates an *in extenso* representation or description of the music to be explained as its only output. Typically, such a program will be a *compact*, *compressed* or *short* encoding or description of its output. A basic claim of this paper is that such a description (in the form of a program) becomes an *explanation* of the object being described, as soon as it is *shorter* than an *in extenso* description of that object. In other words, a compressed encoding of an *in extenso* description of an object can be considered an *explanation* of that object.

Moreover, I would like to suggest that the more parsimoniously one can describe an object on some given level of detail, the better that description explains the structure of the object on that level of detail. In other words, given two explanations (i.e., compressed descriptions) of a (musical) object, the better explanation will in general be the shorter or simpler one. This is essentially an application of “Occam’s razor”. This raises the problem of how exactly one should measure the “length” of an analysis (or a program representing an analysis) (see [7]).

The following simple example should serve to illustrate the foregoing ideas. Consider the problem of describing the set of 12 points shown in Figure 1. One could do this by explicitly giving the co-ordinates of all 12 points, thus:

$$\begin{aligned} &P(p(0,0),p(0,1),p(1,0),p(1,1), \\ & \quad p(2,0),p(2,1),p(2,2),p(2,3), \\ & \quad p(3,0),p(3,1),p(3,2),p(3,3)). \end{aligned} \quad (1)$$

In this encoding, a set of points is denoted by  $P(\cdot)$  and each point within such a set is denoted by  $p(x,y)$ . This encoding can be thought of as being a program that computes the set of points in Figure 1 simply by specifying each point individually. Representing this set of points in this way requires one to write 24 integers. Moreover, the encoding does not represent any groupings of the points into larger constituents, nor does it represent any structural relationships between the points. In other words, this description is an *in extenso* description that does not represent any of the structure in the point set and therefore cannot be said to offer any *explanation* for it.

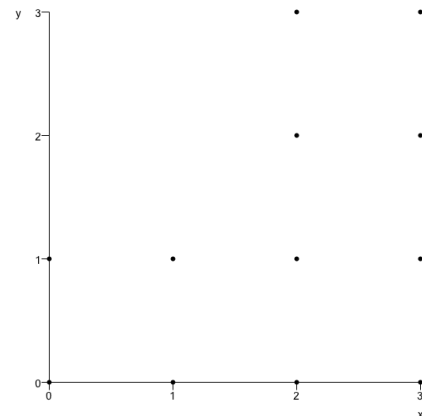
Alternatively, one could obtain a shorter encoding of the point set in Figure 1 by exploiting the fact that it consists of three copies at different spatial positions of the square configuration of points,

$$P(p(0,0),p(0,1),p(1,0),p(1,1)). \quad (2)$$

One could encode this as follows:

$$\begin{aligned} &T(P(p(0,0),p(0,1),p(1,0),p(1,1)), \\ & \quad V(v(2,0),v(2,2))) \end{aligned} \quad (3)$$

where  $T(P(\cdot),V(\cdot))$  denotes the union of the point set,  $P(\cdot)$ , and the point sets that result by translating  $P(\cdot)$  by the vectors in  $V(\cdot)$ , where  $v(x,y)$  denotes a vector. Note that expression (3) fully specifies the point set in Figure 1 using only 12 integers—that is, half the number required to explicitly list the co-ordinates of the points in the *in extenso* description given in expression (1). I contend that the description in (3) is an *explanation* of the point set in Figure 1 precisely because it represents some of the structural regularity in this point set. Moreover, it is precisely *because* it captures this structure that it manages to convey all the information in (1) (and more) while being only roughly half the length of (1).



**Figure 1.** A set of 12, two-dimensional points on a Euclidean integer lattice.

## 3. KOLMOGOROV COMPLEXITY

The *Kolmogorov complexity* of an object is a measure of the amount of *intrinsic* information in the object [1,2,8–10]. Roughly speaking, it is the length in bits of the shortest program that takes no input and computes the object as its only output. The more structural regularity there is in an object, the shorter its shortest possible description and the lower its Kolmogorov complexity. Unfortunately, it is not generally possible to determine the Kolmogorov complexity of an object, as it is usually impossible to prove that any given description of the object is the shortest possible. Nevertheless, the theory of Kolmogorov complexity supports the notion of using the *length* of a description as a measure of its complexity and it supports the idea that the shorter the description

of a given object, the more structural regularity that description captures. The theory has also been used to show formally that data compression is almost always the best strategy for both model selection and prediction [11]. For further discussion of the relationship between music analysis and Kolmogorov complexity, see [7].

#### 4. MUSIC ANALYSIS AND DATA COMPRESSION

If the best explanations are the shortest descriptions, then that would seem to imply that the goal of music analysis is to compress as much information about as much music, as much as possible. To illustrate this, let us consider a close musical analogue of the point-set example in Figure 1 discussed above.

Figure 2 shows the beginning of J. S. Bach's Prelude in C minor (BWV 847) from the first book of *Das Wohltemperierte Klavier*; and Figure 3 shows a point-set representation of this music in which the horizontal dimension represents time in tatums and the vertical dimension represents *morphic pitch*, which corresponds to the vertical position of a note-head on the staff [12–14].



**Figure 2.** The opening notes from J. S. Bach's Prelude in C minor (BWV 847) from the first book of *Das Wohltemperierte Klavier*. Patterns A, B and C correspond, respectively, to the patterns with the same labels in Figure 3. (From [12].)

The union of the patterns A, B and C could be specified by explicitly listing the 12 points in this set, thus:

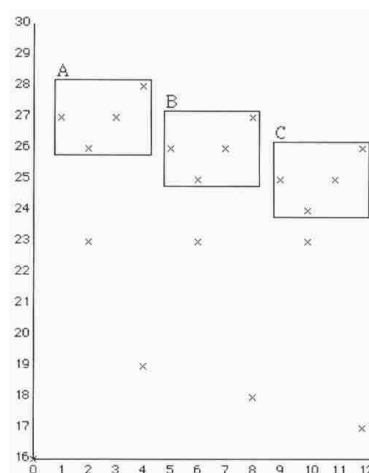
$$\begin{aligned} &P(p(1,27),p(2,26),p(3,27),p(4,28), \\ & \quad p(5,26),p(6,25),p(7,26),p(8,27), \\ & \quad p(9,25),p(10,24),p(11,25),p(12,26)) . \quad (4) \end{aligned}$$

This would require one to write down 24 integers. Alternatively, on an analogy with expression (3), one could exploit the fact that the set consists of three occurrences of the same pattern at different (modal) transpositions, and describe it more parsimoniously as follows,

$$\begin{aligned} &T(P(p(1,27),p(2,26),p(3,27),p(4,28)), \\ & \quad V(v(4,-1),v(8,-2))) . \quad (5) \end{aligned}$$

This expression not only requires one to write down only half as many numbers, but also encodes some of the analytically important structural regularity in the music—namely, that the 12 points consist of three, 4-note

patterns at different transpositions. Again, by seeking a compressed encoding of the data, we have succeeded in finding a representation that gives us important information about the structural regularities in that data.



**Figure 3.** A point-set representation of the music in Figure 2. The horizontal dimension represents time in tatums; the vertical dimension represents morphic pitch (see [12–14]). Patterns A, B and C correspond, respectively, to the patterns with the same labels in Figure 2. (From [12])

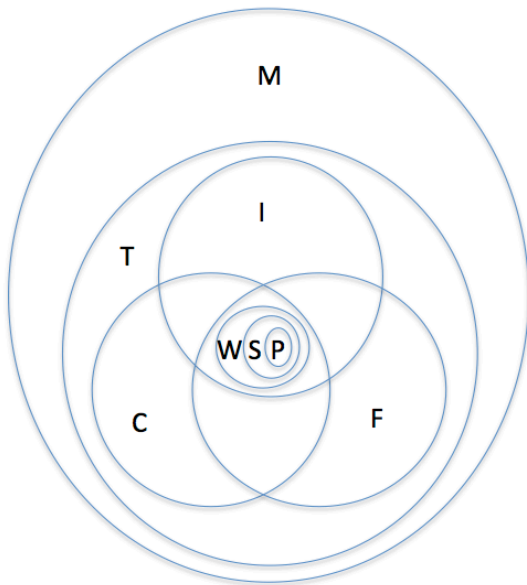
#### 5. MUSIC ANALYSIS AND PERCEPTUAL CODING

As stated at the outset, the work presented here is based on the assumption that the goal of music analysis is to find the best possible explanations for musical works. This could be recast in the language of psychology by saying that music analysis aims to find the most satisfying *perceptual organisations* that are consistent with a given *musical surface* [15].

Most theories of perceptual organisation have been founded on one of two principles: the *likelihood principle* (due to Helmholtz [16]) that proposes that the perceptual system prefers organisations that are the most *probable* in the world; and the *simplicity principle* [17], which states that the perceptual system prefers the *simplest* perceptual organisations. For many years, these two approaches were seen as being in conflict. However, Chater [3], drawing upon the theory of Kolmogorov complexity, proposed that the two principles are mathematically equivalent. In fact, Vitányi and Li [11] showed that this equivalence only strictly holds for data that is *random*. Since music is highly regular and not at all random, this result casts some doubt upon whether the likelihood principle, commonly applied in Bayesian and probabilistic approaches to musical analysis such as those proposed by Meyer [18], Huron [19] and Temperley [20], can ever successfully be used to find struc-

tural regularities such as thematic relationships and transformations.

The work presented here is therefore more akin to models of perceptual organisation based on the simplicity principle than it is to probabilistic or Bayesian models. In particular, it relates closely to those theories in the tradition of Gestalt psychology [17] that make use of *coding languages*—languages designed to represent the structures of patterns in particular domains. Theories of this type predict that sensory input is more likely to be perceived to have organisations that correspond to shorter descriptions in a particular coding language. Coding theories of this type have been proposed for serial patterns (e.g., [4]), visual patterns (e.g., [6]) and, indeed, musical patterns [5,21–23].



**Figure 4.** A Venn diagram illustrating the various possible contexts in which a musical object might be interpreted. A phrase (*P*) could be interpreted within the context of a section (*S*), which could be interpreted within the context of a work (*W*), and so on. *C* = works by the same composer; *F* = works in the same form or genre; *I* = works for the same instrumentation; *T* = tonal music; *M* = all music.

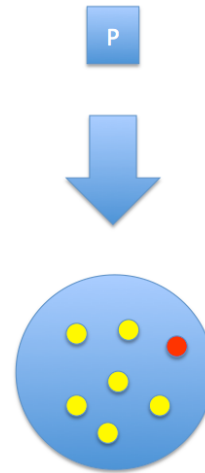
## 6. A SKETCH OF A COMPRESSION-BASED MODEL OF MUSICAL LEARNING

Let us define a *musical object* to be any quantity of music, ranging from a chord or phrase, through to a complete work or even a collection of works. A musical object is typically interpreted by a listener or an analyst in the context of some larger object that contains it (see Figure 4). In essence, the model of musical learning proposed here is as follows. The analyst or listener explicitly or implicitly tries to find the *shortest* program that computes a set of *in extenso* descriptions of a set of musical objects containing

- the object to be explained (the *explanandum*); and

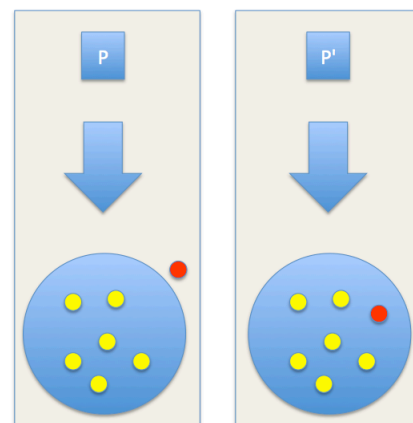
- other objects, related to the explanandum, defining a *context* within which the explanandum is to be interpreted.

This idea is illustrated in Figure 5.



**Figure 5.** The analyst’s or listener’s understanding of a musical object (in red) is modelled as a program, *P*, that computes a set of musical objects containing the one to be explained along with other related objects forming a context within which the explanandum is interpreted.

The analyst and listener differ in the degree of freedom that they have to choose the context within which they interpret an object. The analyst can explicitly choose a context of closely related objects (e.g., music in the same genre or by the same composer) that permits a more parsimonious description of the explanandum. The listener, on the other hand, is forced to interpret the explanandum in the context of his or her largely implicit understanding of all the previous music he or she has encountered.



**Figure 6.** When the listener hears a new piece (in red), the existing explanation (i.e., “program”) (*P*) for all the music previously heard is minimally modified to produce a new program (*P'*) to account for the new piece in addition to all previously encountered music.

Figure 6 illustrates the idea that when the listener hears a new piece (in red), the existing explanation (i.e., program) (P) for all the music previously heard (in yellow) is minimally modified to produce a new program (P') to account for the new piece in addition to all previously encountered music. The perceived structure of the newly encountered musical object is then represented by the specific way in which P' computes that object. On this view, music analysis, perception and learning essentially reduce to the process of compressing musical objects.

However, it is important to stress that, even though both the analyst and the listener *aim* to find the shortest possible encodings of the music they encounter, they both typically *fail* to do this in general. As Chater [3, pp. 578] points out,

“the perceptual system cannot, in general, maximize simplicity (or likelihood) over all perceptual organizations...It is, nonetheless, entirely possible that the perceptual system chooses the simplest (or most probable) organization that it is able to construct.”

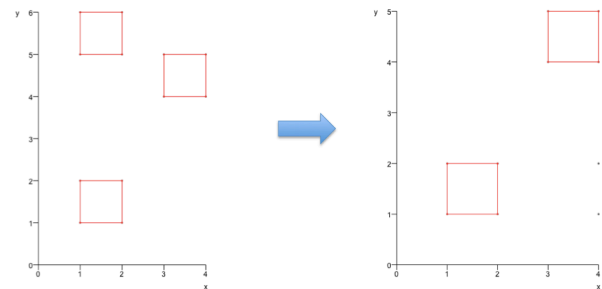
This is largely a result of the limited processing and memory resources available to the perceptual system. For example, we typically describe the structure of a piece of music in terms of motives, themes and sections, all of which are *temporally compact segments*, meaning that they are patterns that contain *all* the events that occur within a particular time span. It could well be that for some pieces, a more parsimonious description (corresponding to a better explanation) might be possible in terms of patterns containing notes and events that are dispersed widely throughout the piece. However, the listener’s limited memory and attention span constrains him or her to focus on patterns that are temporally compact (see also [24]).

## 7. USING THE MODEL TO EXPLAIN INDIVIDUAL DIFFERENCES

The model just sketched can be applied to understanding the emergence of differences between the ways that individuals understand the same piece. What I’ve just proposed is that an essentially “greedy” algorithm is used to construct an interpretation for a newly encountered piece that minimally modifies an existing “program” that generates descriptions of all the pieces in a particular “context” set. This would imply that the way that an individual understands a given piece depends not only on which pieces he or she already knows, but also on the *order* in which these pieces were encountered. This implication could fairly straightforwardly be tested empirically.

A rather crude version of the foregoing model has been implemented in an algorithm called SIATECLearn. The SIATECLearn algorithm is based on the geometric pattern discovery algorithm, SIATEC, described by

Meredith, Lemström and Wiggins [12]. The SIATEC algorithm takes as input a set of points and automatically discovers all the translationally equivalent occurrences of the maximal repeated (or *translatable*) patterns (MTPs) in the dataset. Note that SIATEC outputs a collection of such occurrence sets, called *translational equivalence classes* or TECs, such that each occurrence set (TEC) contains all the occurrences of a particular pattern (a pattern being just a set of notes). An algorithm called SIATECCompress [25] runs SIATEC on a dataset, then sorts the found TECs into decreasing order of “quality”. Given two TECs, the one that results in the better compression (in the sense of expressions (4) and (5) above) is deemed superior. If both TECs give the same degree of compression, then the one whose pattern is spatially more compact is considered superior. SIATECCompress then scans this list of occurrence sets and computes an encoding of the input dataset in the form of a set of TECs that, taken together, account for or *cover* the entire input dataset.

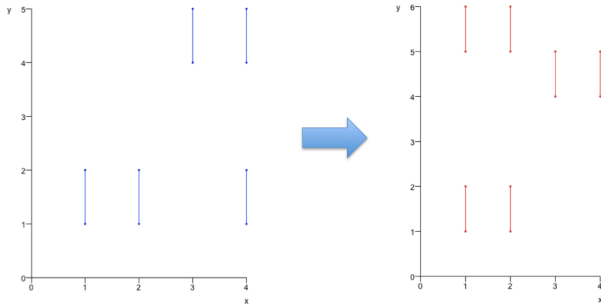


**Figure 7.** Output of SIATECLearn when presented first with the dataset on the left and then with the dataset on the right.

SIATECLearn runs SIATECCompress, but also stores the patterns it finds on each run and will preferably re-use these patterns rather than newly found ones on subsequent runs of the algorithm. Thus, when SIATECLearn is run on the 12-point pattern on the left in Figure 7, it “interprets” the dataset as being constructed from three occurrences of the square pattern shown. This square pattern is therefore stored in its “long-term” memory. When the algorithm is subsequently run on the 10-point dataset on the right, it prefers to use the stored square pattern than any of the patterns that it finds in this dataset, so it interprets the data as containing two occurrences of the square pattern along with two extra points.

Conversely, when SIATECLearn is first presented with the 10-point dataset, it interprets the dataset as being composed from 5 occurrences of the two-point vertical line configuration shown on the left in Figure 8. This pattern is then stored in long-term memory, so that when the algorithm is subsequently presented with the 12-point dataset, it interprets this set as consisting of 6 occurrences of this vertical line rather than 3 occurrences

es of the square pattern. This very simple example illustrates how the way in which objects are interpreted may depend on the order in which they are presented.



**Figure 8.** Output of SIATECLearn when presented first with the dataset on the left and then with the dataset on the right.

## 8. COSIATEC: MUSIC ANALYSIS BY POINT-SET COMPRESSION

COSIATEC [25–27] is a greedy compression algorithm based on SIATEC. The algorithm takes a dataset,  $S$ , as input and computes an exhaustive, exclusive partition of  $S$  such that  $S$  is equal to the union of the covered sets of a set of TECs. The basic idea behind the algorithm is sketched in the pseudo-code in Figure 9.

```

COSIATEC( $S$ )
  while  $S$  is not empty
    Find the best TEC,  $T$ , using SIATEC
    Add  $T$  to the encoding,  $E$ 
    Remove the points covered by  $T$  from  $S$ 
  return the encoding  $E$ 

```

**Figure 9.** The COSIATEC algorithm.

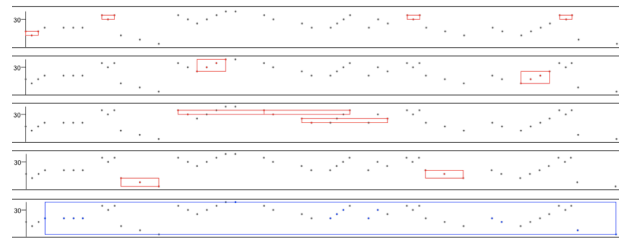
As shown in Figure 9, the COSIATEC algorithm first finds the “best” TEC in the output of SIATEC for the input dataset,  $S$ . The best TEC is the one that has the best compression ratio, which is the ratio of the number of points in the union of the occurrences of the TEC to the sum of the number points in one occurrence of the TEC’s pattern and the number of occurrences (minus one, because one of the occurrences is explicitly encoded in the pattern). That is, the compression ratio of a TEC,  $T$ , denoted  $CR(T)$ , is given by

$$CR(T) = \frac{|U_{s \in T} s|}{|p| + |v| - 1}$$

where the sets  $s$  are occurrences in  $T$ ,  $p$  is one occurrence in  $T$  and  $v$  is the set of vectors that translate  $p$  onto the other occurrences in  $T$ . If two TECs have the same compression ratio, then COSIATEC chooses the TEC in which the first occurrence of the pattern is the more *compact*: the *compactness* of a pattern is the ratio of the number of points in the pattern to the number of dataset

points in the bounding box of the pattern. These heuristics for evaluating the quality of a TEC are discussed in more detail by Meredith *et al.* [12] and Collins *et al.* [24].

As shown in Figure 9, once the best TEC,  $T$ , has been found for the input dataset,  $S$ , this TEC is added to the encoding ( $E$ ) and the set of points *covered* by  $T$  is removed from  $S$ . The set of points *covered* by a TEC is simply the union of the occurrences in the TEC. Once the covered set of  $T$  has been removed from  $S$ , the process is repeated, with SIATEC being run on the new (somewhat depleted)  $S$ . The procedure is repeated until  $S$  is empty, at which point  $E$  contains a list of TECs that cover  $S$ . Moreover, because the TECs that give the best compression ratio are selected on each iteration,  $E$  is typically a *compact* or *compressed* encoding of  $S$ .



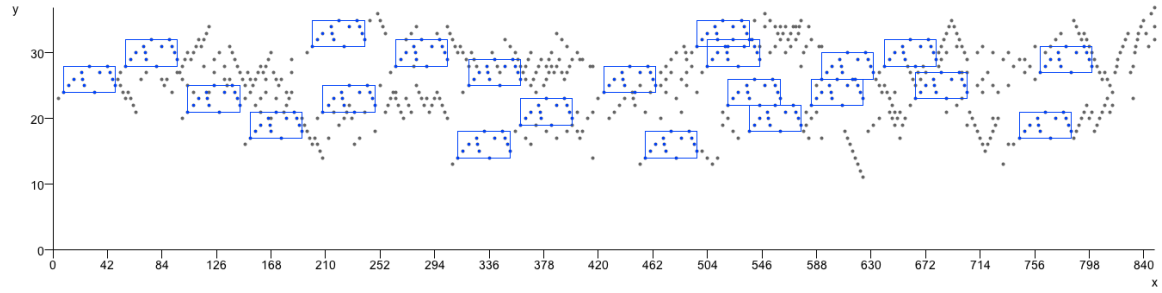
**Figure 10.** The set of TECs computed by COSIATEC for a short Dutch folk song, “*Daar zou er en maagdje vroeg opstaan*” (from the Nederlandse Liederbank, <http://www.liederenbank.nl>). Courtesy of Peter van Kranenburg.)

Figure 10 shows the output of COSIATEC for a short Dutch folk song. The complete piece can be encoded as the union of the covered sets of 5 TECs, as shown. The first TEC, at the top of Figure 10, consists of the occurrences of the lower-neighbour-note figure. This TEC has the best compression ratio of any TEC for any maximal translatable pattern in the dataset. After these three-note sets have been removed from the piece, the next best TEC is the second one from the top in Figure 10, namely the two occurrences of the four-note rising scale segment. The fifth TEC, at the bottom of the figure, consists only of the 14 blue points inside the indicated bounding box. These are points that are left over after removing the sets of repeated patterns that give the best compression ratio. This final set of “residual” points, which cannot be compressed by the algorithm, is essentially seen by the algorithm as being random “noise” that it cannot “explain”.

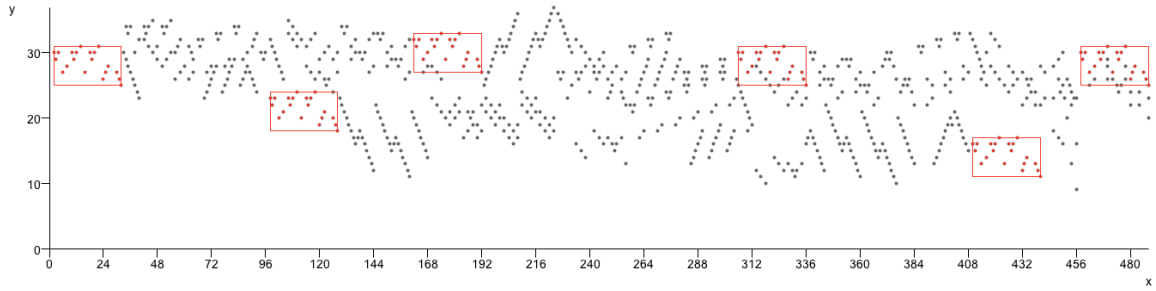
## 9. ANALYSING FUGUES WITH COSIATEC

I am currently exploring the extent to which COSIATEC can be used to automatically generate meaningful analyses of the first book of J. S. Bach’s *Das Wohltemperierte Klavier* (BWV 846–869).

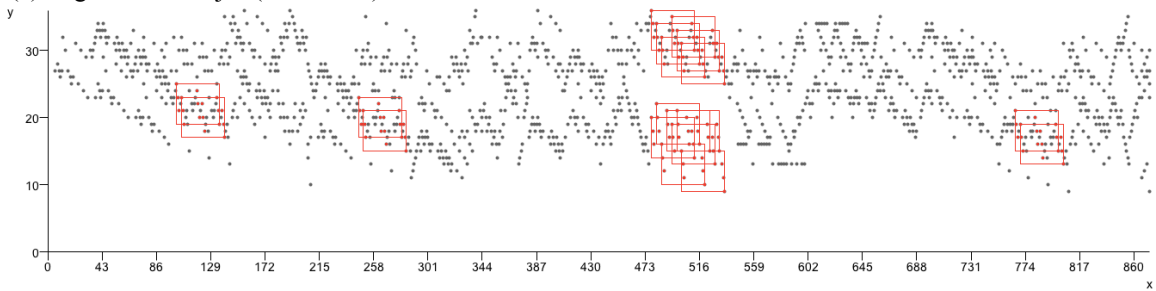
(a) Fugue in C major (BWV 846)



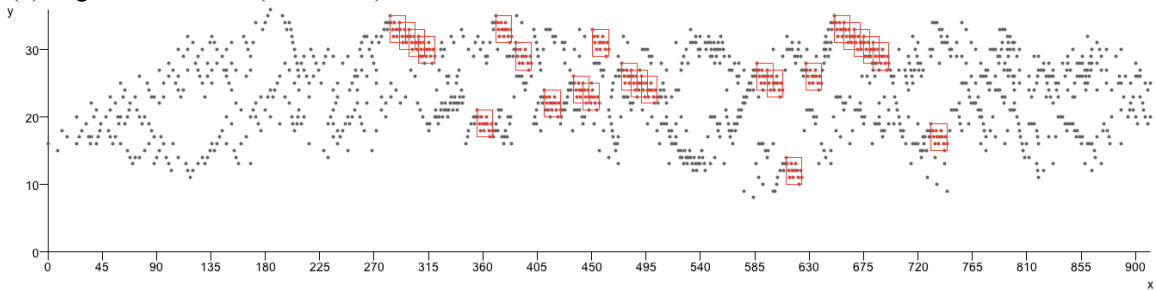
(b) Fugue in C minor (BWV 847)



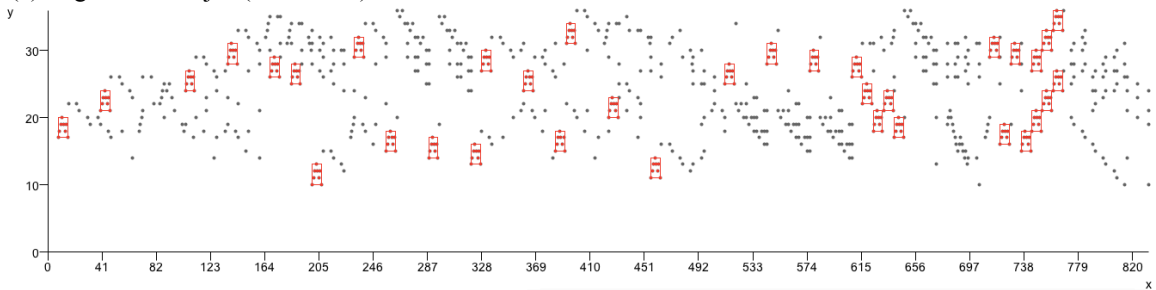
(c) Fugue in C# major (BWV 848)



(d) Fugue in C# minor (BWV 849)



(e) Fugue in D major (BWV 850)



**Figure 11.** The TEC computed on the first iteration of COSIATEC for each of five fugues from the first book of J. S. Bach's *Das Wohltemperierte Klavier*.



Figure 11 shows the first TEC discovered by COSIATEC for each of five fugues from this collection.

Figure 11 (a) shows the first TEC computed for the Fugue in C major (BWV 846). As can be seen, this TEC contains occurrences of the main subject of the fugue. However, the first note of the subject is not included in the TEC’s pattern. The fact that this pattern occurs as the first TEC found by COSIATEC indicates that the shown pattern is a *maximal translatable pattern* for some particular vector (i.e., it contains all the points that can be translated by some vector onto other points in the dataset) and that no other pattern provides as good a compression ratio. If we interpret the degree of compression achieved by a TEC as an indication of how “explanatory” that TEC is of the dataset’s structure, then we can suggest that no other pattern (according to the COSIATEC algorithm) is as “explanatory” of this piece as the one shown.

Note that the bass entry of the subject at around 546 tatum lacks the first note of the first entry. If the first note were included in this pattern, this would mean that the bass entry at 546 would not be included in the TEC, which would reduce the overall compression achieved.

Figure 11 (b) shows the first TEC found by COSIATEC for the Fugue in C minor (BWV 847). Note that, again, the algorithm automatically identifies the subject of the fugue as being the structurally “most explanatory” pattern (i.e., the one that allows the data to be most compressed). It is interesting to consider that this monophonic pattern that occurs entirely within one voice was discovered by the algorithm in a dataset that provides no information whatsoever about the voices to which the notes in the piece belong. This suggests that the approach embodied in COSIATEC might be useful in inferring voice-leading and contrapuntal structure from data in which this information is not encoded (e.g., single-channel MIDI data). Note that the tonal answers in BWV 847 are not included among the occurrences of this TEC: this is because COSIATEC only discovers occurrences of maximal patterns that are translationally *exactly* equivalent to the pattern. For a recent approach incorporating *inexact* matching in a geometric pattern discovery algorithm, see [28].



**Figure 12.** Bars 7–9 of the Fugue in C# major (BWV 848), showing (in red boxes) the pattern of the first TEC computed by COSIATEC. The pattern consists of all 9 of the indicated notes.

Figure 11 (c) shows the first TEC discovered by COSIATEC in the Fugue in C# major (BWV 848). This time, the first TEC does not correspond to the subject of

the fugue. Figure 12 shows this pattern in context. As can be seen, it consists of three occurrences of a falling semiquaver arpeggio figure that together act as a kind of “fingerprint” for a sequential episode that recurs several times throughout the piece. Note that, again, the pattern occurs wholly within one voice, even though voice information is not provided in the input data.

Figure 11 (d) shows the first TEC discovered by COSIATEC for the Fugue in C# minor (BWV 849). Again, the pattern does not correspond to the principal subject of this fugue. Instead, as shown in Figure 13, it corresponds to the segment of the second subject of this triple fugue that forms a counterpoint with the principal subject.



**Figure 13.** Bars 37–40 of the Fugue in C# minor (BWV 849), showing (in red ellipses), the pattern of the first TEC computed by COSIATEC. The pattern consists of all 14 of the indicated notes.

Finally, Figure 11 (e) shows the first TEC found by COSIATEC for the Fugue in D major (BWV 850). This TEC consists of all occurrences of the characteristic, demisemiquaver flourish that begins every entry of the main subject in this fugue.

The foregoing discussion seems to suggest that the patterns discovered on the early iterations of COSIATEC often correspond (at least in contrapuntal music) to patterns of thematic and structural importance.

## 10. CONCLUSIONS

In this paper I have proposed a new computational approach to music analysis, based on the compression of point-set representations of musical objects. I have indicated how this approach relates to both the theory of Kolmogorov complexity and psychological coding theories of perceptual organisation. I have also sketched a model of musical learning in which the process of gaining an understanding of a new piece is modelled as the minimal modification of an existing compact encoding of a collection of known pieces so that that the modified encoding also includes a description of the new piece.

Finally, I have presented an algorithm, COSIATEC, that implements this new, compression-based, geometric approach to music analysis, along with examples of analyses generated by this algorithm for some of the fugues from the first book of J. S. Bach’s *Das Wohltemperierte Klavier*. These analyses illustrate the potential of this approach for discovering patterns of thematic and structural importance in musical works.



## 11. REFERENCES

- [1] A. N. Kolmogorov: “Three approaches to the quantitative definition of information,” *Problems of Information Transmission*, Vol. 1, No. 1, pp. 1–7, 1965.
- [2] M. Li and P. Vitányi: *An Introduction to Kolmogorov Complexity and Its Applications*, Springer: Berlin. Third ed., 2008.
- [3] N. Chater: “Reconciling simplicity and likelihood principles in perceptual organization,” *Psychological Review*, Vol. 103, No. 3, pp. 566–581, 1996.
- [4] H. A. Simon: “Complexity and the representation of patterned sequences of symbols,” *Psychological Review*, Vol. 79, No. 5, pp. 369–382, 1972.
- [5] D. Deutsch and J. Feroe: “The internal representation of pitch sequences in tonal music,” *Psychological Review*, Vol. 88, No. 6, pp. 503–522, 1981.
- [6] E. L. J. Leeuwenberg: “A perceptual coding language for visual and auditory patterns,” *American Journal of Psychology*, Vol. 84, No. 3, pp. 307–349, 1971.
- [7] D. Meredith: “Music analysis and Kolmogorov complexity,” in *Proceedings of the 19<sup>th</sup> Colloquio d’Informatica Musicale (XIX CIM)*, Trieste, 21–24 Nov., 2012.
- [8] G. J. Chaitin: “On the length of programs for computing finite binary sequences,” *Journal of the Association for Computing Machinery*, Vol. 13, No. 4, pp. 547–569, 1966.
- [9] R. J. Solomonoff: “A formal theory of inductive inference (Part I),” *Information and Control*, Vol. 7, No. 1, pp. 1–22, 1964.
- [10] R. J. Solomonoff: “A formal theory of inductive inference (Part II),” *Information and Control*, Vol. 7, No. 2, pp. 224–254, 1964.
- [11] P. M. B. Vitányi and M. Li: “Minimum description length induction, Bayesianism, and Kolmogorov complexity,” *IEEE Transactions on Information Theory*, Vol. 46, No. 2, pp. 446–464, 2000.
- [12] D. Meredith, K. Lemström and G. A. Wiggins: “Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music,” *Journal of New Music Research*, Vol. 31, No. 4, pp. 321–345, 2002.
- [13] D. Meredith: “The *ps13* pitch spelling algorithm,” *Journal of New Music Research*, Vol. 35, No. 2, pp. 121–159, 2006.
- [14] D. Meredith: *Computing Pitch Names in Tonal Music: A Comparative Analysis of Pitch Spelling Algorithms*, PhD thesis, University of Oxford, 2007.
- [15] F. Lerdahl and R. Jackendoff: *A Generative Theory of Tonal Music*, MIT Press: Cambridge, MA., 1983.
- [16] H. L. F. von Helmholtz: *Treatise on Physiological Optics*, New York: Dover, 1910/1962. Trans. and ed. by J.P. Southall. Originally published in 1910.
- [17] K. Koffka: *Principles of Gestalt Psychology*, New York: Harcourt Brace, 1935.
- [18] L. B. Meyer: *Emotion and Meaning in Music*, Chicago University Press: Chicago, 1956.
- [19] D. Huron: *Sweet Anticipation: Music and the Psychology of Expectation*, MIT Press: Cambridge, MA., 2006.
- [20] D. Temperley: *Music and Probability*, MIT Press: Cambridge, MA., 2007.
- [21] H. A. Simon and R. K. Sumner: “Pattern in music,” in *Formal Representation of Human Judgment* (B. Kleinmuntz, ed.), Wiley: New York, 1968.
- [22] D.-J. Povel and P. Essens: “Perception of temporal patterns,” *Music Perception*, Vol. 2, No. 4, pp. 411–440, 1985.
- [23] D. Meredith: “A geometric language for representing structure in polyphonic music,” in *Proceedings of the 13<sup>th</sup> International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, 8–12 October, 2012.
- [24] T. Collins, R. Laney, A. Willis and P. H. Garthwaite: “Modeling pattern importance in Chopin’s *Mazurkas*,” *Music Perception*, Vol. 28, No. 4, pp. 387–414, 2011.
- [25] D. Meredith: “COSIATEC and SIATECCompress: Pattern discovery by geometric compression,” in *Music Information Retrieval Evaluation Exchange (Competition on “Discovery of Repeated Themes & Sections”)* (MIREX), 2013.
- [26] D. Meredith, K. Lemström and G. A. Wiggins: “Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music,” in *Proceedings of the Cambridge Music Colloquium*, University of Cambridge, 2003.
- [27] D. Meredith: “Point-set algorithms for pattern discovery and pattern matching in music,” *Proceedings of the Dagstuhl Seminar on Content-Based Retrieval* (No. 06171), Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.
- [28] T. Collins, A. Arzt, S. Flossmann and G. Widmer: “SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations”, in *Proceedings of the 14<sup>th</sup> International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.