Structural Analysis and the Musical Score

Andie Sigler

Doctor of Philosophy

School of Computer Science

McGill University

Montreal, Quebec

2017-10-15

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

ACKNOWLEDGEMENTS

I am very grateful to my academic supervisor, Doina Precup, for supporting me in pursuing this research. Doina has encouraged me in many ways over the years, since I took my first undergrad computer science class with her. Doina's brilliance is evidenced in her delight in supervising unconventional creative research, offering critical commentary and mathematical suggestions in areas at some distance from her primary intellectual turf. Doina's support of this project has been an immeasurable gift – I want to thank her for her dedication, openness, patience, and wise guidance.

I thank Prakash Panangaden for reading drafts of this thesis with an understanding and enthusiasm that mean the world to me. Prakash expressed interest the moment he learned of this project, and although I already had an advisor in Doina, Prakash was keen to tag along – the three of us often met together, which was both entertaining and edifying. I'm very grateful to Prakash as well for advancing and enriching my general mathematical education: I was honoured to have been an audience of one at many a virtuosic impromptu lecture.

Eliot Handelman receives adulatory laudations for inventing this project and sharing it with me (and with us) – and for being my best friend. Ice cream with sprinkles to Casey Basichis for his friendship and devotion, for dazzling discussions, and for helping to make it real.

Thanks to Carl Patenaude-Poulin for providing most of the French translation of the abstract, and to Marcia Basichis for reading Chapter 2 and offering good advice. Thanks to Jon Wild and Malcolm Sailor for test-driving aspects of the system described in this thesis (as part of another research project); and extra thanks to Jon for his friendly encouragement over many years. I thank Skyler Olson and David Donna for their contributions to the visualization systems. Thanks also to Jean-Benoit Chasles and Alexander MacDonald for contributing to aspects of this project not specifically discussed in this thesis. (And thanks again to Doina for helping to facilitate these collaborations!)

I thank my parents, Cathy and Dan, my brother Jacob, and the Handelman family for their loving support.

ABSTRACT

This project is about structural data analysis on musical scores. From a *structural* point of view, a score is a data object in which multi-dimensional internal relations participate in abstract musical design. The purpose of a structural analysis is to *describe* aspects of the *shape* of a score. Treating the score initially as a low-dimensional, discrete Cartesian system, we *build* higher-dimensional structure, increasing the quantity and dimensionality of available information about the internal relations within the score. Since there are many ways of finding structure, structural score analysis is framed as an open, descriptive exploration. Three particular constructions are described, illustrating the main idea (– *polyphones*, *N-sets*, and *Z-chains*). Since the premise is to explore geometric aspects of a musical score, these methods are mathematically general, and not particular to music.

Polyphones show how sets of temporal intervals form networks of *containment* and *overlap* relations; an application is shown describing the "texture" of notes on a score. *N-sets* involve the formation of maximal temporal intervals to describe regions of *content* on a score. In an application to pitch-class, all temporally maximal pitch-class sets ("PcNs") are found. The set of PcNs form a *polyphone* (since they have temporal relations of containment and overlap). *Z-chains* structure a sequence of orderable terms into a set of recursively-oriented shapes. Applied to melody, Z-chains show ways in which a low-dimensional musical representation generates structure *beyond* the concatenation of its elements, sketching a complex architecture of parallels, developments, crossings, and reversals.

ABRÉGÉ

Ce projet concerne l'analyse structurelle de partitions de musique. Du point de vue *structurel*, une partition est un ensemble de données gouvernées par des relations internes multi-dimensionnelles qui forment ensemble une abstraction de la musicalité. Le but d'une analyse structurelle est de *décrire* certains aspects de la forme géométrique d'une partition. Considérant initialement la partition de musique comme un système carésien de basse dimension, nous érigeons des structures de dimension supérieure, obtenant ainsi une connaissance plus riche des relations internes de la partition. Puisqu'il y a un grand nombre d'approches possibles à la recherche de structure, l'analyse structurelle de partitions est envisagée comme un processus d'exploration libre et descriptif. Nous présentons ici trois concepts illustrant l'idée d'analyse structurelle - les polyphones, N-ensembles, et Z-chaînes. Comme notre but est d'explorer la structure géométrique de la partition de musique, ces méthodes sont mathématiques et générales; leurs applications possibles ne sont pas restreintes au seul domaine de la musique. Les polyphones décrivent comment les ensembles d'intervalles temporels forment des réseaux de relations d'inclusion et de chevauchement; nous démontrons leur application en décrivant la "texture" formée par les notes d'une partition. Les N-ensembles consistent en la formation d'intervalles temporels maximaux cernant les différentes régions de *contenu* d'une partition. En une application aux catégories de hauteur, les ensembles temporels maximaux correspondant chacun à une ensemble harmonique sont trouvés (appelés "PcNs"). L'ensemble des PcNs forme un *polyphone*, puisqu'il contient des relations temporelles d'inclusion

et de chevauchement. Les Z-chaînes représentent une structure qu'on peut appliquer à une séquence de termes ordonnables pour obtenir un ensemble de formes orientées récursivement. Dans une application à la mélodie, les Z-chaînes nous permettent d'entrevoir de quelles façon une représentation musicale à faible dimension peut générer une structure plus riche que la simple concaténation de ses éléments, créant une architecture complexe de parallèles, développements, croisements, et renversements.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Inter-Disciplinary Context, Goals, and Commitments

This thesis presents research in computer science, with application to music analysis: we have a commitment to *study* music – that is to attempt to *observe* it, and *describe* it. The study of music is framed as an open, descriptive exploration.

We require no further motivation than a basic *interest* in observing music (– as any scientist might be interested in making observations). It is often the case that observations can be *used* in various ways, but we don't require the motivation of particular *problems* to be solved, and *description* as a main aim takes the place of *prediction*, *categorization*, *quantification*, or *generalization* (though these scientific and engineering modes are facilitated by *having* a description). We wish to develop ways of thinking about musical scores that are *general* enough to apply to an encounter with any score, but that do not necessarily *generalize* from one score to the next.

In particular, we develop (and implement) *computational* ways of thinking about scores – we have a commitment to develop a way of thinking about music that is *deeply* and *natively* computational, which is to say formal and mathematical.[1] Musical thought has developed a number of categories and methodologies for music analysis, most of which are not computational (since they were developed in an environment without computers). By "not computational," we mean that they cannot be calculated without shorthand, numerical stand-ins for what are really semantic and contextual phenomena.[2]

This study prioritizes the formation of a computational foundation for musical study, rather than the importation of prior concepts into a medium where they can't be properly expressed. We contextualize with literature about prior music-analytic concepts – not showing how we might replicate them or improve upon them, but to illuminate the formal *differences* between the practices as they address the *same* kinds of musical concerns.

Semantic and contextual phenomena *belong* in any musical study, but we adopt a computational strategy of *putting off* using them, because the computational expression of semantic and contextual phenomena is a hard unsolved AI problem. In

[1] This thesis is about *non-statistical* computing – and therefore about non-statistical theories of music. We share an interest in "letting the data speak for itself" with various nonparametric statistical models – but *as* statistical models, these use computational methods, mathematical concepts, and music-theoretic assumptions that are unrelated to ours, as well as different having goals (i.e. with respect to prediction, generalization, summarization, quantification), and producing *kinds* of results that are ontologically distinct.

[2] I.e. "heuristic" parameters – whether "theorized" or induced from data ("learned"). Even simple, intuitive musical-technical categories such as the *phrase* are of this nature.

the meantime, we can use some of the well-understood *primary* affordances of computation (e.g. organizing data and performing arithmetic) to learn more about the *structure* of musical scores. The hope is that since scores can be seen as being primarily *made of* structure, approximate semantics will be more precise and more flexible when more structural information is available.

The primary contribution of this thesis is a computer-scientific "structural" approach to data analysis invented for music, but mathematically general and probably widely applicable. It is about the development of a computational theory and practice based on the need to look at a score as a complex *shape* (which is the underlying hypothesis about how computation might be able to model some perceptually relevant aspects of music). This project is about building, from simple computational elements, ways of thinking about the structure of a score. It is a deeply *computer-scientific* project, because it is not only about *using* computation to do things, but about considering the *kinds* of knowledge that are implicated in the inputs and results of computations, and developing *particular* ways in which we can use computation to *increase* the quantity and dimensionality of information available in a data object (– instead of learning *from* a set of data, we want to learn *about* a single data object).

1.2 Organization and Content of This Thesis

Chapters:

1. Introduction
2. Structural Analysis
3. Polyphones
4. N-sets and PcNs
5. Z-chains
6. Music and McLuhan's Evolution of Media

Following this brief introduction, this thesis contains a conceptual, theoretical chapter introducing the main idea of *structural analysis*, three technical middle chapters (Chapters 3, 4, and 5), and a concluding chapter that positions the project of musical AI with respect to a historical, cultural, ethical context.

Chapters 2 and 6 are written in a non-technical style, while the central chapters contain formal mathematical definitions, enumerations and proofs of mathematical and logical properties, pseudocode, and illustrations mostly consisting of annotated scores on piano rolls and staff-notation.

All five of the main chapters contain contextualization with respect to prior literature, with reviews of music-computational sub-areas appearing in Chapters 4 and 5, and a review of a computer-science sub-area in Chapter 3. In general a "depth" approach to prior literature is preferred, consisting of a detailed, informative comparison to a *few* related ideas – rather than trying to mention large numbers of tangentially related publications, contrastive rather than contributive. We have not been able to find any *closely* related literature. Mathematical and computer-scientific concepts used are so fundamental as to be "common-property" and not tied to any particular publications (e.g. sets, sequences, graphs). Nonetheless, attention is brought again and again, in different ways, to context.

Chapter 2 provides a conceptual foundation, motivation, and context for the structural methods given in Chapters 3, 4, and 5. These structural methods (which are *examples* or *illustrations* of the main idea) are offered as specific technical contributions to computer science as well as to music analysis.

1.2.1 Illustrations

The structural methods offered in the central chapters are illustrated, mostly by annotated scores. The purpose of these pictures is to give a visual intuition for the shape of the mathematical structures, helping to clarify their meaning *as* structure.

The annotated scores are not semantic *interpretations* of a score, they are visualizations of structural properties – the images are the results of simple, deterministic, and formally defined computations. Like all pictures, and like music, the illustrations invite subjective additions: in the form of discursive interpretation as well as in superstructural organization.

They also suggest analytic *variations*: while these are illustrations of mathematically general concepts, aspects of the *application* of these general concepts are necessarily *specific* and therefore defined through choices that could be made in many different ways. The space of these *kinds* of analyses (on a single score) is indefinitely big. It is usually not hard to invent an alternate view capturing a new detail.

The invitation to interpretation, superstructuring, and variation are part of the *openness* of the structural sketches shown – they invite engagement and development, rather than delivering a finished analytic picture. This is one primary purpose of these kinds of analyses and illustrations – to make structural information more available to human intuition, in order to help us study music.

The illustrations are screenshots from a computer system implementing the structure-finding algorithms in the thesis. The computer system has a domain-specific language for interacting with scores, writing programs to (flexibly and recursively) find structure in different ways, and visualizing the results as annotated scores (and in other ways). The implementation and operation details of the system are outside the scope of this thesis, but the illustrations provide secondary evidence of the feasibility and existence of a usable application of the ideas described.

1.3 Statement of Contributions

A statement about the contributions of the author to the content of this thesis:

This work is based on 10 years of collaboration with Eliot Handelman. Dr. Handelman is a source of initial ideas which (in this thesis) I elaborate, formalize, analyze, apply, contextualize, express in writing. We collaborate on the computer system which implements these ideas, and which generates the illustrations in this thesis.

The practice of *structural computing* (as described in Chapter 2) originates with Dr. Handelman, but the expression, elaboration, partial formalization, and contextualization of the idea in Chapter 2 is mine.

Polyphones (discussed in Chapter 3 and applied again in chapters 4 and 5) are my work – Dr. Handelman proposed only that polyphony might be addressed by the relations of “hold” and “fold” – and the rest of the formalization, implementation, application, analysis, and illustration of polyphones is mine.

Dr. Handelman invented and implemented PcNs, as well as key-assertions (Chapter 4). The generalization to N-sets, and all of the mathematical and musical formalization and analysis is mine, as is the comparative discussion with respect to existing music-computational literature.[3]

Likewise, Dr. Handelman invented and implemented Z-chains and Z-shapes (Chapter 5) – I formalized, analyzed, applied, and contextualized them.[4]

The concluding chapter, discussing a broader cultural context for this project, is my work (Chapter 6).

[3] [HandelmanSigler2013] describes PcNs and key-assertions.

[4] [HandelmanSigler+2012] describes the Z-chain algorithm and an application to automatic orchestration.

CHAPTER 2
Structural Analysis

2.1 The Score

This project is about data analysis on a particular kind of data object called a *score*. A score is a symbolically notated representation of a musical work. We coin a more general, data-analytic sense for the word *score*, divorcing the score data-object from any commitments to be music-related.

A score is a data object in which the multi-dimensional *relations* between its parts is of interest: its parts are internally *structured*, and cannot meaningfully be decontextualized as a statistical *population*. Further, it is a data object that bears analysis *on its own*, as a self-contained system, without reference to an *external* population of related objects.

This attitude toward score analysis comes from an attitude toward music analysis: that it is insufficient (for musical understanding) to analyze a musical score as a population – or even a sequence – of events; and that an analysis of a musical score that measures or categorizes it in relation to a set of other scores is likewise insufficient. Although aspects of the human musical experience are learned and contextually coded within the world of music and the world at large, we take the attitude that there is a core of abstract musical *design* that is available in a *single* score.

Design, or appreciable internal structure, is not limited to music – it is featured in all of the arts, including poetry, literature, theatre, film, the visual arts, architecture, and the design of clothing, shoes, cars, furniture, and everything else. Although these works of art may have semantic, referential, or pragmatic aspects, they also *necessarily* have an aspect of *design*, of abstract patterns and shapes of their internal relations in whatever featural dimensions are available. Music is a good starting point for the analysis of design, since the abstract core of design in music is readily accessible without understanding linguistic semantics or how parts of the design correspond to objects in the physical world.

Data objects that are *not* "designed" may also bear analysis as a score. Natural images, sounds, and other phenomena seem (to our senses or to our thinking) to have design, or appreciable internal structure. We seem to have a *sense* of "design," and observing the patterns, shapes, and contrasts of the internal structuration of things seems to be a fundamental part of perception and cognition. These observations occur for different sub-dimensions of visual, auditory, tactile (etc.) perceptions; they occur spatially, temporally, and on abstract axes; and dimensions can be fused at both low perceptual levels, and mid- and high cognitive levels to form multi-dimensional or multi-modal experiences or concepts.[1]

This project is about looking at data from a *structural* point of view. It is motivated by music, and illustrated with *musical* scores, but the mathematical and data-analytic discussions are not domain specific.

[1] Section 4 of this chapter is about this sense of design.

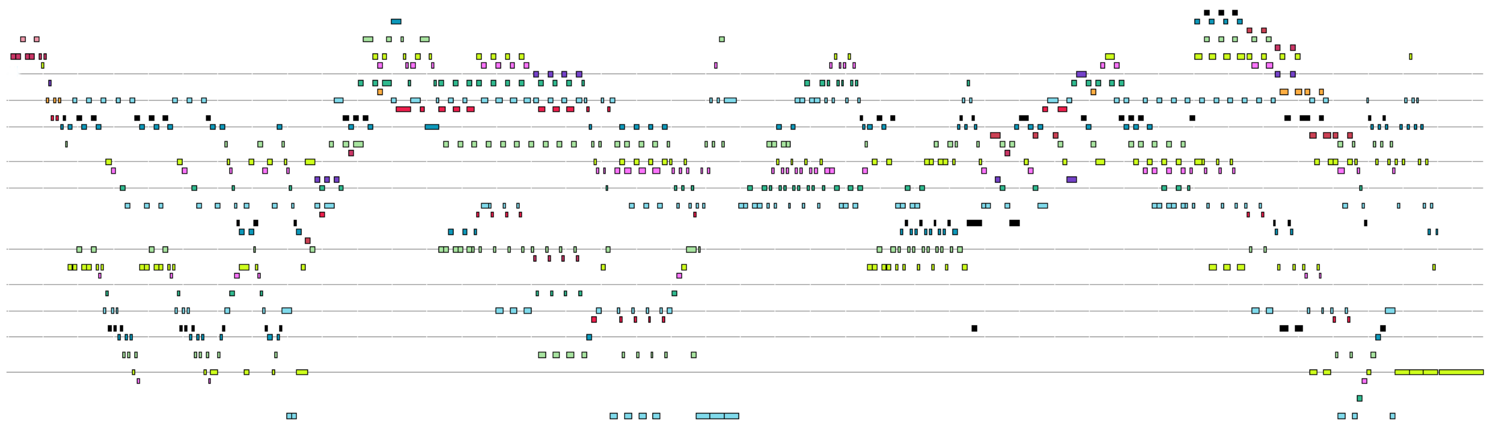

Figure 2–1: Domenico Scarlatti, Piano Sonata K. 79. Colors assigned arbitrarily to different pitch-classes.

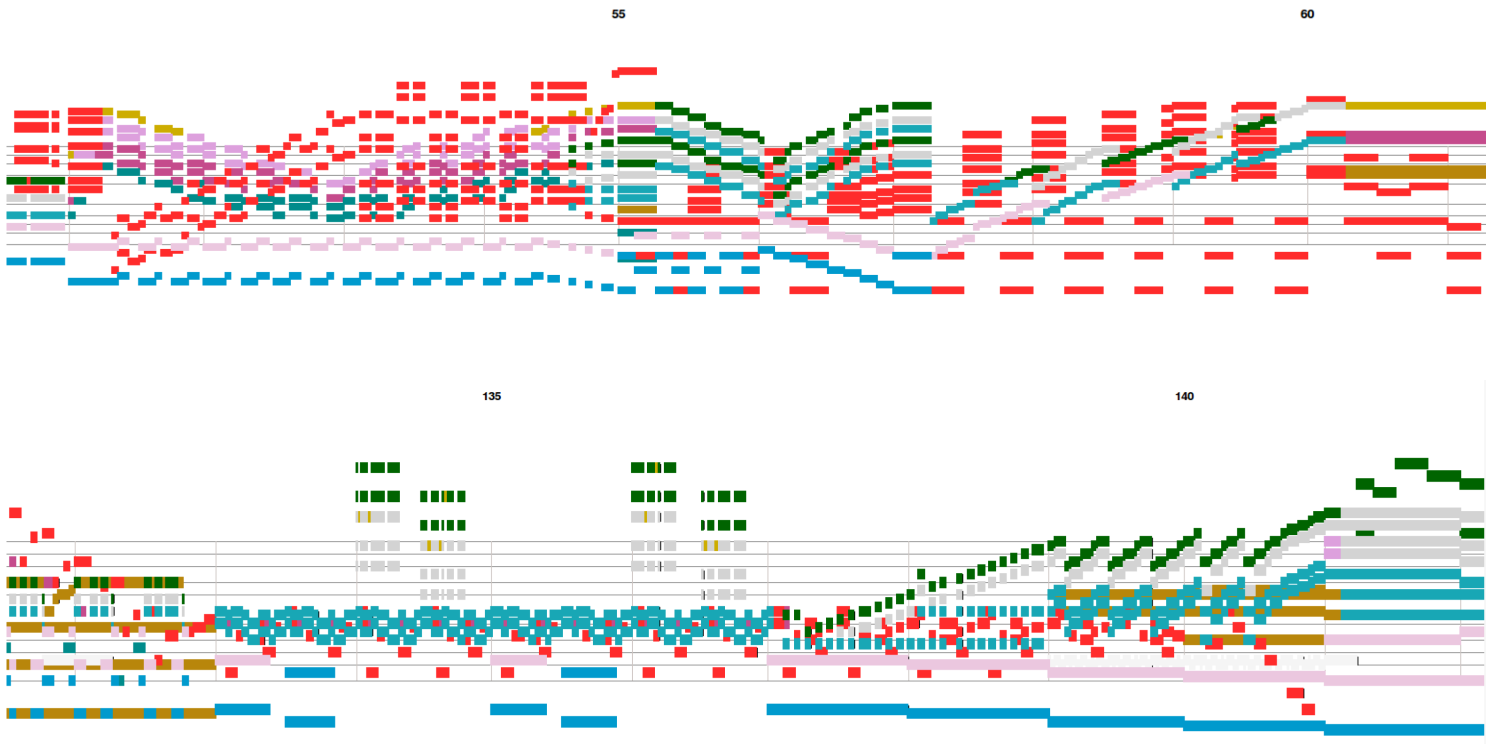

Figure 2–2: Sergei Prokofiev, excerpts from Piano Concerto No. 3. Pitch on vertical axis (with treble- and bass-clef staff lines), time on horizontal axis (with bar lines), instrumental voices assigned arbitrarily to different colors.

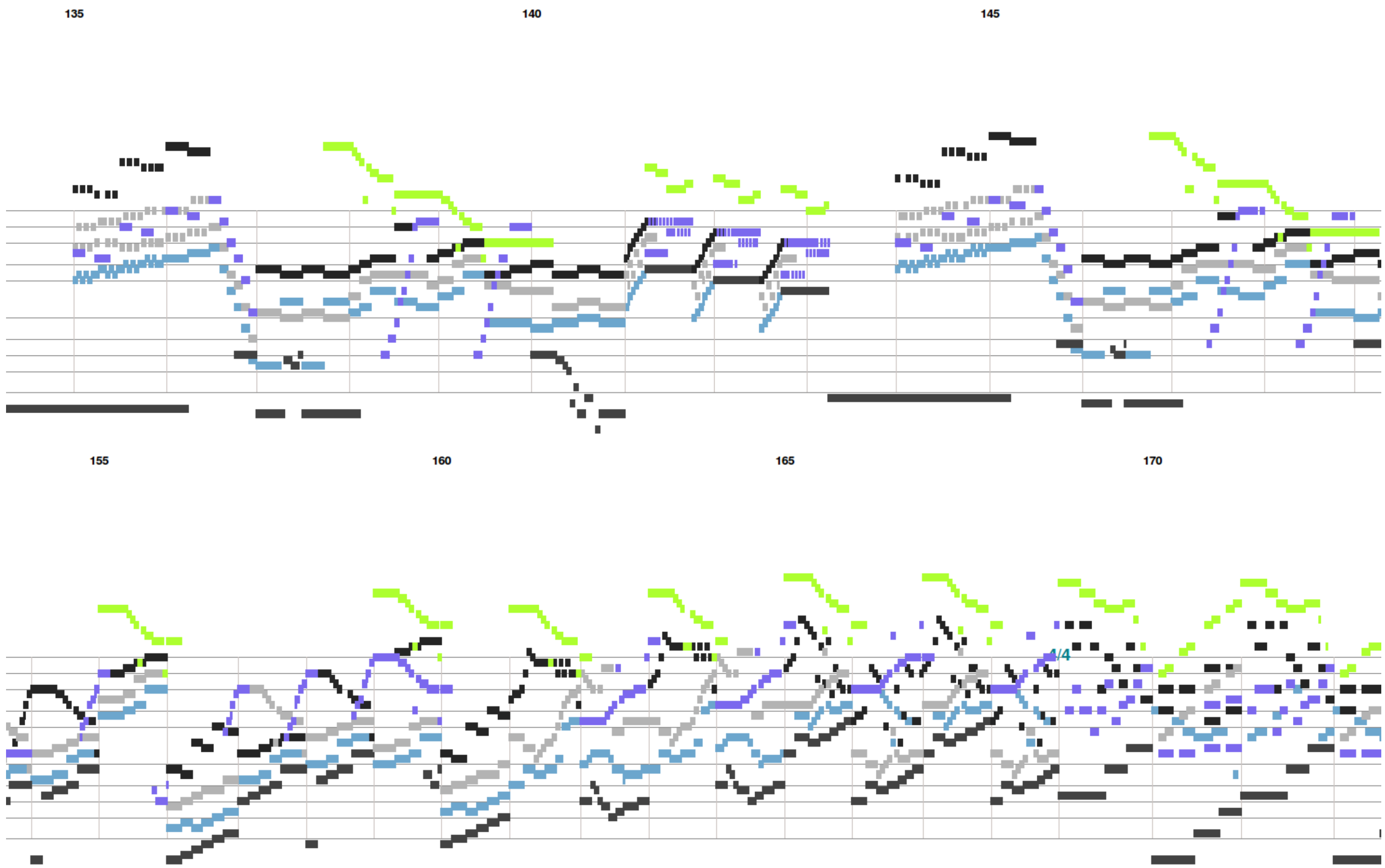

Figure 2–3: Arnold Schoenberg, excerpts from *Verklärte Nacht* (for string sextet). Colors correspond to instrumental voices.

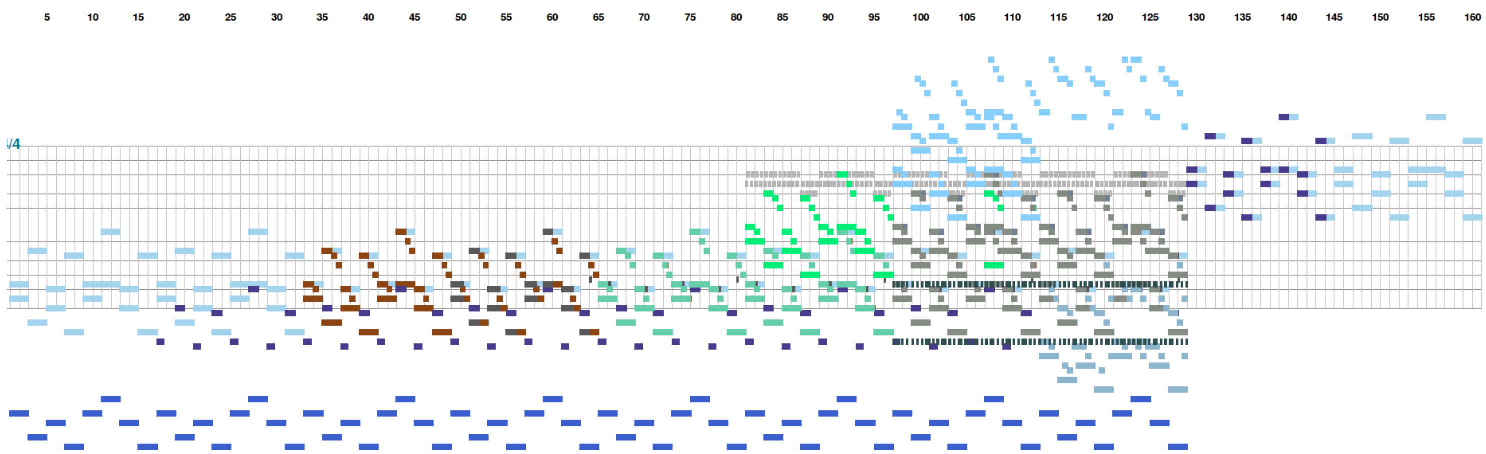

Figure 2–4: Hans Zimmer, *Time*. Colors correspond to instrumental voices.

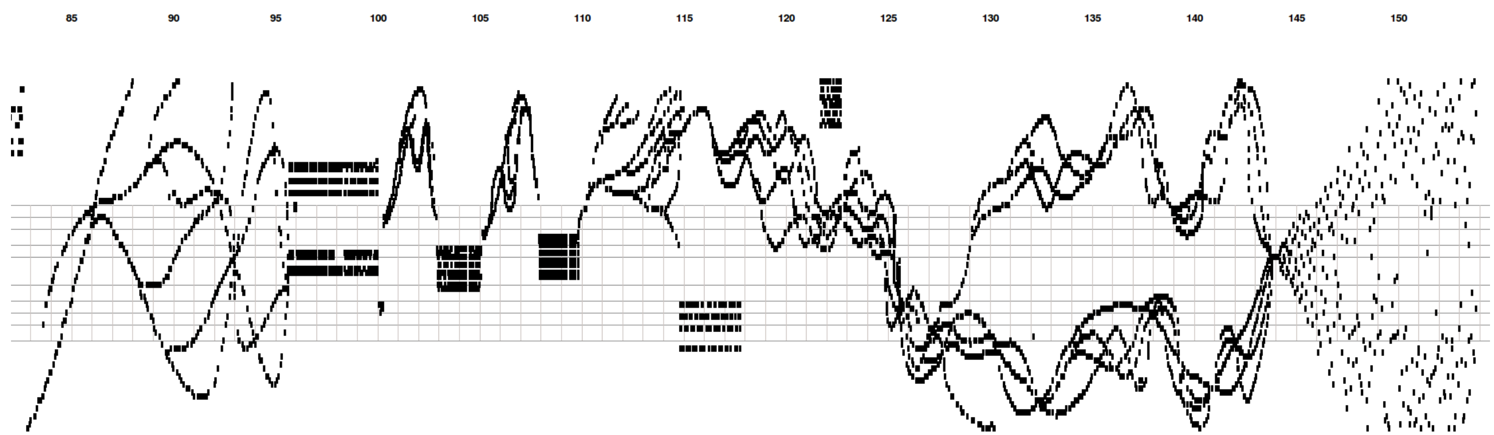

Figure 2–5: Iannis Xenakis, excerpt from *Evriyali* (for piano).

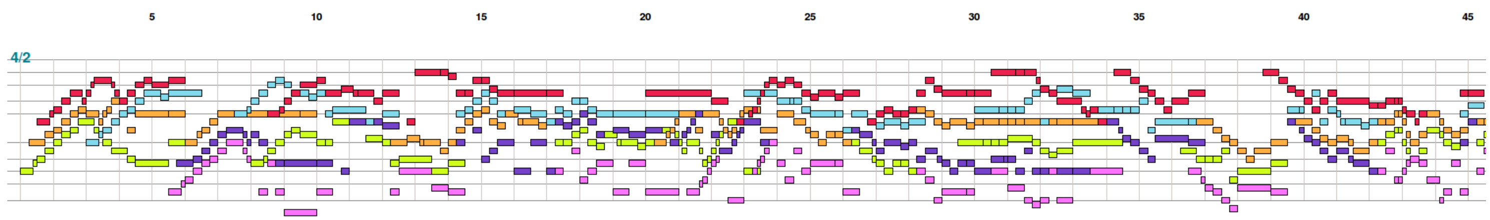

Figure 2–6: T.L. de Victoria, *Ardens Est Cor Meum* (excerpt). Colors correspond to voices.

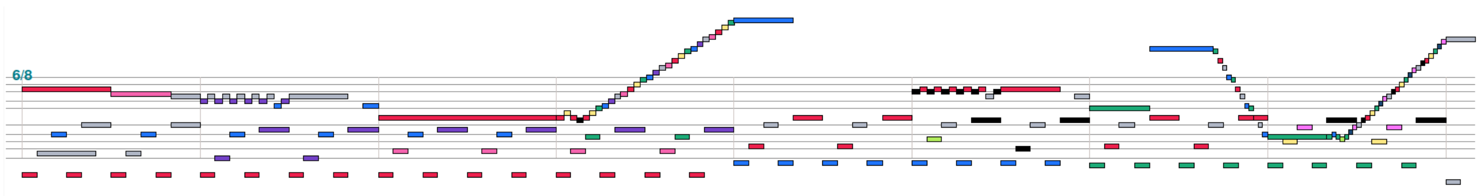

Figure 2–7: Frédéric Chopin, Prelude No. 24 (excerpt). Colors correspond to pitch-classes.

2.1.1 The musical score

A musical score is a notated or encoded set of instructions for, or description of, a piece of music.[2] In many cases it is the *direct* output of music creation. In other cases it is a (lossy) transcription from another medium. Many different notation languages exist, including staff notation and various symbolic computer encodings.

The primary content of a musical score is a set of points (i.e. notes) each consisting of a temporal interval and a set of features (e.g. pitch, loudness, instrumental timbre). Some features of these notes might encode their relation to a mutual abstract context (e.g. their places on a virtual temporal grid, or an index grouping them into different "voices"). There could be any number of features available, but typically there are few, sometimes limited to just pitch, or pitch and voice index. The content of a score feature is typically simple: a quantity, a code, an abstract pattern term, or a relative position, though more complex representations are not ruled out.

A musical score is *temporally oriented*, in the sense that its Cartesian axes are normally positioned with *time* running from left to right, because all of its data points have well-defined and meaningful temporal position and extension.

---

[2] Piano-roll visualizations of musical scores are shown in Figures 2–1–2–7.

Most scores don't include much detail about *sound*. While music is based on affordances of the auditory system, a score represents music *as design* that is abstractable from sound. What may be surprising is how low-dimensional and low-resolution this abstraction is when compared with auditory perception, while still being effective *as* music.[3]

A musical score is typically discretely valued, with low resolution. There are usually 12 pitches available per octave, and only about 8 audible octaves. Likewise, the rhythmic dimension usually contains only a small number of different values, with a basis in multiplicative relationships, so that they tend to be highly distinguishable. In contrast to the continuum of acoustic affordance, a musical score is written in broad, bold strokes, well *above* perceptual thresholds for differentiation.

A musical score is *noiseless* because it is not a model or representation of something else.[4] Every feature is deliberately placed in order to participate in the music. While some notes may contribute more than others to the structure of a given *interpretation* – while any one *projection* through the score might only contain a subset of notes – for *any* note, there is some point of view from which it contributes to musical structure.

[3] This is not meant to be a controversial ontological statement. A score is effective *as* music in whatever sense music *can* be encoded in a score, and evidently this capacity is nontrivial.

[4] A score might have *mistakes* in it, but this is not the same thing as *noise*, as in data based on approximations or containing statistical effects. We work from the premise that the data at hand is what is to be analyzed, and that it is not an approximation.

In data terms, a musical score is very small. The first movement of Beethoven's fifth symphony has about 10500 notes in it; *Happy Birthday* has significant structural complexity over 21 notes. The complexity of music is not in the *size* of the data, but in the intricacy of the interactions within it.

If we took a large number of scores at once, the size of the data would be much bigger. But, for now, this is not the kind of analysis we're considering. Each score is a unique object, and is assumed to have a meaningful status as a *single* data object. We ask: within the world of *one* piece of music, what are its parts and patterns, and how do they interact? Questions about the *comparison* of different scores, or about the interaction of people with scores, can only be addressed *after* we have come to grips with the kind of analysis demanded by the individual score.

2.2 Structural Analysis

This project is about the *structural* analysis of a score. The *structure* of something is characterized by the relations between its parts. What the *parts* of something are, and how they *relate* to one another – not as a population, but as a system – is the topic of a structural analysis. There are many ways of dividing a score into parts, but not all ways of dividing up a score are "structural." For instance, if we take every 5th note of a score and call that one "part," it's unlikely that this is a coherent slice of the score. The reason for this is that we have taken a knife arbitrarily to the score without observing *anything at all* about it beforehand, rather than allowing the data to split at its natural joints.

2.2.1 Carving Data at Its Joints: Structure, Heuristics, and Structural Relativity

The concept of "carving nature at its joints," as a criterion for a successful theory, is given by Socrates in Plato's *Phaedrus*, and has more lately been taken up as a slogan by the structuralist metaphysician Theodore Sider in his *Writing the Book of the World* [Sider2011]. Sider argues that although different conceptual systems may be able to form equally *true* theories, some concepts are objectively *better* than others because they reflect the natural structure of reality. For instance, while one conceptual frame has a concept called a *car* which can be driven in and out of a garage, another way of framing the world is that there are two different kinds of things called an *in-car* (a car in a garage) and an *out-car* (a car not in a garage). When an in-car touches the threshold of the garage, it disappears and a corresponding out-car appears outside of the garage, and vice versa.[5] Although this way of looking at things is logically workable, it doesn't seem "natural," because it doesn't carve reality at its joints. This is a toy example, and Sider's real discussion is about the ontology and logic of *everything*.

The metaphysical problem of discerning the natural structure of reality is a tough call. It's considerably easier to talk about natural structure in a data-analytic context. Besides this significant restriction in scope, our view of natural structure differs from Sider's in that Sider thinks there is *one* most natural way in which

[5] The in-car / out-car example was given in a lecture by Sider ("Is Metaphysics about the Real World" https://www.youtube.com/watch?v=RKYZ8U-P5jA), and is attributed to Eli Hirsch.

reality is structured, while our thesis is that there are *many* different ways of carving a data object at its joints (we call this a *structural* analysis). Furthermore, there are ways of carving a data object that are *purely* structural, and other ways that are heuristically assisted with a light touch, such that they are *relatively* natural in comparison to more brutal, nonstructural methods. *Heuristic* analysis, broadly defined as the injection of *external* data into the analysis of a data object, is strictly necessary.[6] Heuristics, the *semantic* level of data interpretation, are what allows data to represent things. The scientific study of data, however, demands that we take precautions against *the fallacy of misplaced concreteness*, in which theoretical constructs or abstractions are taken to be completely descriptive, taking precedence over actuality and open observation.[7] A musical example would be to look at scores from the point of view of seeking theoretical entities like keys, chords, etc. While these concepts are *themselves* worthy of empirical investigation with respect to scores, a data science that can *only* take this kind of a priori point of view on a score is ungeneral and underpowerful.

[6] A common understanding of the term "heuristic" is any intentionally non-rigorous computational shortcut. The definition we offer is related, but with a refinement designed to bring attention to the *information content* of the heuristic method.

[7] Whitehead's "fallacy of misplaced concreteness" means "mistaking the abstract for the concrete" [Whitehead1925, p.52], or "... neglecting the degree of abstraction involved when an actual entity is considered merely so far as it exemplifies certain categories of thought. There are aspects of actualities which are simply ignored so long as we restrict thought to these categories."[Whitehead1929, p.8].

It may seem that *any* data process is a heuristic about how a data object might be treated, in the sense that it is an external information-context applied to the data object. We wish to make a more subtle distinction. Consider an analysis that only uses relations such as $<$, $>$, and $=$; these are ones that we take for granted as mathematical primitives, and that work by comparing internal parts of the score. We would propose that such a function is *structural*. In contrast, a *heuristic* involves the injection of a *semantic context* against which the data is interpreted. As a simple example, consider a function that includes a filter somewhere for values < 5. This filter is a cue that in a *context* that is not inherent in the data object itself, 5 is a cutoff for "big" versus "small" numbers.

Discernment about how heuristics are applied poses both methodological and ontological distinctions. While a structural analysis can avoid becoming statistical until a statistical analysis is explicitly desired, all but the most austerely mathematical of structural analyses use heuristics. For this reason, a mathematical interest in structural data analysis leads to thinking about the degrees and kinds of heuristic interference that inhere in a given structural analysis.

One way to manage the application of heuristics in a structural analysis is to apply them as *late* as possible. If a structural analysis is applied *first*, then the semantic power of a heuristic *increases*, while its ability to cut the data against the grain *decreases*. Once a structural analysis has made an initial "natural" structuration, then making a heuristic cut with respect to these *structures* means that the heuristic is operating on higher-informational, more internally contextualized data, and that any cuts are at a more structural (less atomistic) level.

Another way to integrate heuristics into a structural methodology is to alternate between structural analyses and heuristic reductions of the resulting structure-space. This forms a cycle in which information is *grown* structurally, cut heuristically, and then the remaining structural subspace is grown again in another structural pass. This allows us to form higher-informational inner-contextual objects, *and* focus on those that are of particular semantic (external-contextual) interest.

The idea that a structural analysis defines a *natural*, internally motivated cut through a data object does not imply that such an analysis is *objectively* constructed, such that it is not somebody's *idea* about how to structure a score. In fact, there are any number of ways to structure a score, and the structure that is found in a score depends on the structural *point of view* taken.[8] This is the principle of *structural relativity*. Structural relativity implies that a structural analysis is never complete. The meaning of an individual point in the data is also *relative*, since it can instantiate different places in different structures at the same time.

"Structural relativity" is a phrase coined by the mathematical structuralist Michael Resnik [Resnik1997]. *Mathematical structuralism* is a philosophy of mathematics in which mathematical objects are *structures* (systems of relations) rather than real or ideal objects. From this point of view, e.g. integers are not things

[8] "Any number of ways" means a massively combinatorial number of ways with respect to a finite score size and resolution. Not only is it not feasible to enumerate these because there are a lot of them, but we don't even have a *method* for enumerating them, nor for evaluating them relative to their informativeness on a score or semantic value relative to music – so the set of functions is currently an open possibility space exemplified by a small number of invented methods.

(or concepts) that happen to stand in certain relations to one another by virtue of their relative magnitudes or their ordering; they are *only* the positions of these relationships. The implications that Resnik gets from structural relativity are slightly different than the ones we get, because he is working in a different conceptual domain (and one less "concrete" than data analysis).

One difference is that for Resnik, there is *no* "fact of the matter" as to whether the real-number two is equivalent to the natural-number two, because they are places in *different* structures. Since, for Resnik, the number two doesn't exist apart from these structures, there is no coherent way of comparing places in different structures (unless one structure is a part of the other).[9] On the other hand, a score-analysis begins with points that exist *apart from* structures, so we can assert that the *same* point *is* a part of *different* structures – but notice that (as for Resnik) the structural *meaning* of the point with relation to different structures is *different*; it is only that the places in the different structures happen to have the same *reference*.

Resnik argues for an ontology of mathematics in which "mathematical objects are featureless, abstract positions in structures." [Resnik1997, p.4]. This project suggests an inverse, *grounded* version of structuralism, where musical objects *have* features (i.e. pitch, duration, timbral variation, physical location, etc.), and it is precisely these features that allow them to *be structured by* structure-finding operations. As Resnik says, a number is *nothing* but a position in a structure. A *note* by itself

[9] Mathematical structuralism embraces a multiplicity of structures without talking about which is best, or most natural (– differently from Sider).

is something – it has features, and can be heard and have sensory qualities – but it doesn't have meaning from the point of view of musical structure.

Apart from this contextual difference, however, Resnik could be addressing the structural relativity of score analysis: *"In thinking about formulating a theory of structures we must take into account a phenomenon I will call* structural relativity: *the structures we can discern and describe are a function of the background devices we have available for depicting structures. This relativity arises whether we think of patterns and structures as a kind of mould, format, or stencil for producing instances, or as whatever remains invariant when we apply a certain kind of transformation, or as an equivalence class or type associated with some equivalence relation. The structures we recognize will be relative to our devices for specifying forms, or transformations or equivalence relations. Furthermore, by enriching or curtailing these devices we will obtain different notions of structure, count different things as having the same structure, and recognize different relationships between structures."* [Resnik1997, p.250].

2.2.2 Characteristics of Structural Processing

We have characterized a structural analysis as one that *carves the data at its joints*, and contrasted this with the tendency of heuristics to *inject* semantic "joints" into the data, by relating objects to external standards.[10] We've also contrasted structural analysis with statistical analysis, because statistics treats *populations* of

[10] The difference between "internal" and "external" to the data is not really a binary distinction, since there are intermediates like cross-dimensional cuts (when one dimension is used to cut another), and questions about whether something like

objects, whereas structure treats *relations* between *individual* objects. Structure is about *relating* datapoints, and ways of doing this non-heuristically include the kinds of relations that we are willing to take as fundamental, including $<$, $=$, and set-operations such as membership, union, intersection, set-difference, etc.

In this section we give more details about what kinds of techniques are typical of structural analysis.

Typically, a structural analysis works *bottom-up*, linking simpler structures together into larger structures. While the resulting structures are hierarchical (a superstructure made of substructures), a substructure may contribute to several different superstructures, so that the set of superstructures is not constrained to partition the score, or to interrelate hierarchically.

The structural analysis that links substructures into superstructures typically admits a combinatorial set of relations, such that *any* data is describable in terms of these relations (this is in contrast to a *search* or a *parse*, which seek a subset of the total possibility space). This kind of structural analysis therefore covers the *entire* score with (super)structures, and a score *cannot fail to be structured* by a structural analysis (though it may be trivially structured).

A structural analysis is different from a search: a search locates a *specified* schema, whereas a structural analysis locates *all* structures of a given class on a

---

"every fifth event" after all *does* have internal context, which may depend on the nature of the data.

score.[11] A structural analysis is also not a powercut (a term coined by analogy to powerset – n-grams are an example): a powercut locates all *cuts* of a given class, whereas a *subset* of these may be the (unanalyzed) set of structures of a given class.

A structural analysis is different from a parse: it is not constrained to give a single, hierarchically decomposable description of the entire score. Unlike a parse, a structural analysis cannot fail, because it does not describe a language or have a grammar.

Since a structural analysis can work on any data, this kind of analysis does not distinguish between "good" and "bad" data (e.g. good-music vs. bad-music/fake-music/non-music; or type-1-music vs. not-type-1-music), though its results may be instrumental in *heuristically* making qualitative distinctions.

A structural analysis does not *measure*, *classify* or *summarize* the score, although the *structures* it produces will afford measurements or summarizations. A structural analysis does not *simplify* the score, it *increases* the amount of available information by pointing out relations.

A structural analysis does not *optimize* any quantity. A structural analysis does not make *decisions*. A structural analysis does not *target* a ground-truth. A structural analysis cannot be *wrong*, or have an accuracy rating, because it is not trying to guess anything. A structural analysis is typically deterministic.

[11] A "schema" (which means "shape") is one of a combinatorial set (or *class*) of structure-descriptions.

A structural analysis *generates information* – that can be used to heuristically optimize, decide, or target, to quantify or classify, to search – or to generate still more information.

2.3 Comparison with Non-Structural Processing

This section offers two case studies contrasting structural analysis with other, related methodologies: music information dynamics, and structural information theory. The purpose of these case studies is to aid intuition on what does and doesn't count as *structural*, in the sense that we're trying to define it. No particular criticism is made here about the *merits* of music information dynamics or structural information theory, regardless of how we find them "non-structural" – we take them as worthy foils.

2.3.1 Music Information Dynamics

Information theory is basically statistical and quantitative (e.g. the measures of entropy and of mutual information), and not structural.[12] It deals with topics such as signal and noise, whereas a score has no noise, and a piece of music doesn't seem to have a particular "message" to be communicated. An information channel is basically two-dimensional, offering a sequence of symbols, whereas a structural

[12] Entropy is a measure of unpredictability, or a measure of the average (or "expected") amount of "information" or surprisingness per data-unit; mutual information is a measurement of the interdependence of two variables.

analysis makes projections such that a sequence of events is represented in a higher dimensionality.[13]

Since the beginning of information theory in the mid-twentieth-century, there have been proposed applications for music. It has been noticed, for instance, that music tends to be highly *redundant*: if music were considered as *information*, much of its content would be repeated or inefficiently transmitted – music is highly compressable. It's clear, however, that without this redundancy, music isn't *effective*. Therefore it follows that some level of redundancy is *necessary* for music. Quantifying this redundancy has sometimes been suggested as a music analytic measure [Hiller+1966, Youngblood1958].

Another information theoretic concept, *surprisal*, has been proposed as a measure for surprisingness of musical events. The surprisal of an event is its negative log probability (with respect to some probability distribution). This corresponds to how "unlikely" something is, and therefore how "surprising."

An interest in "surprise" in music theory can be traced to L.B. Meyer's *Emotion and Meaning in Music* [Meyer1956]. Meyer's thesis is that emotion and meaning are caused by setting up and fulfilling or violating *expectations*. These can occur as pattern processes that are eventually discontinued, or as reference and deviation with respect to cultural tropes.

[13] I.e. each projection is lower-dimensional, but brings into existence *new* informational dimensions, increasing the total dimensionality of available information.

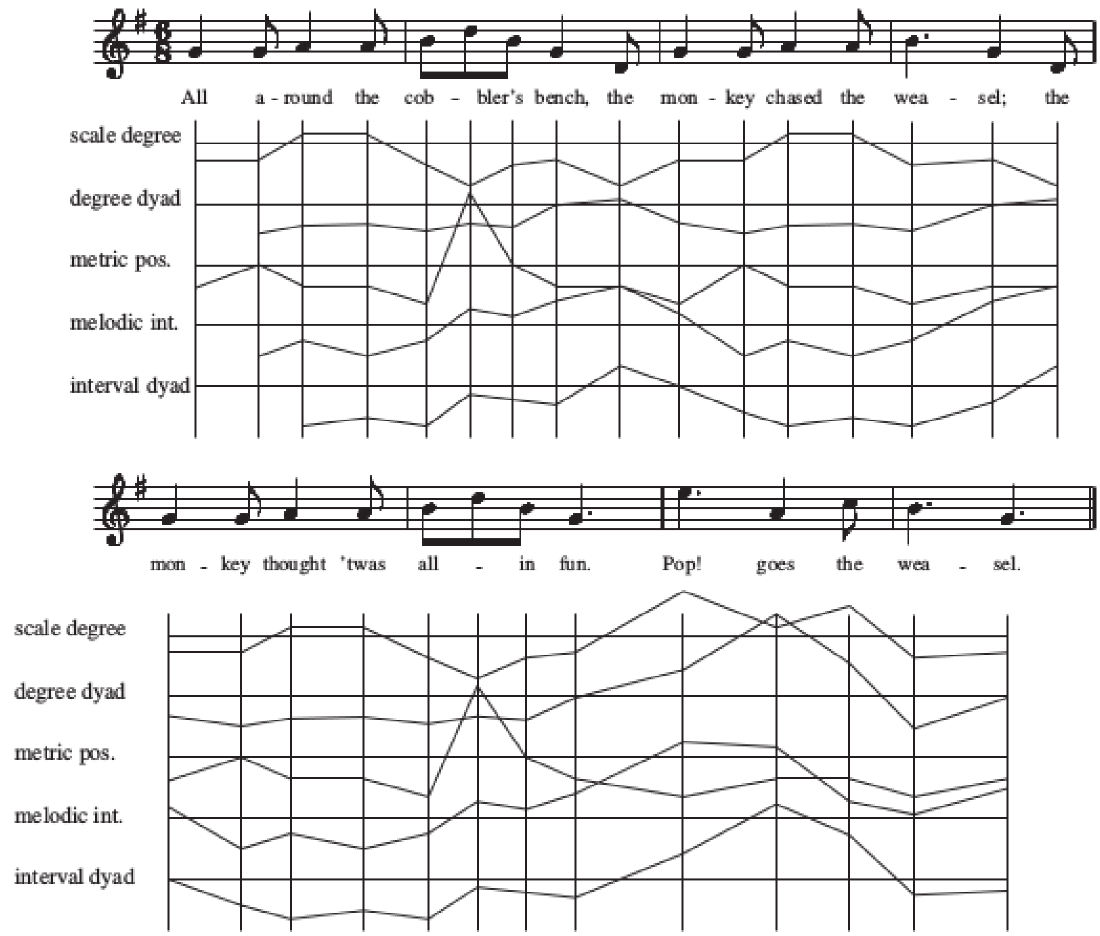

Figure 2–8: (From David Huron, *Sweet Anticipation: Music and the Psychology of Expectation*, published by The MIT Press. [Huron2006, p.115]). Plot of "information" from various viewpoints in "Pop Goes the Weasel." Information is measured by the "probability" (i.e. corpus frequency) of each scale degree, scale degree pair, interval, interval pair, and metric position, such that less frequent events convey more information.

Theorists after Meyer have attempted to systematize the concept of expectation, but until recently these have been low-dimensional and local in scope – for example, Narmour's account of "implication" in three-note figures, applicable to longer melodies through concatenation and super-structuring [Narmour1990], as well as studies counting frequencies of notes, intervals, metric positions, etc., in a corpus and attributing something like "unexpectedness" to less frequent items without context, as in Figure 2–8, from [Huron2006].

Marcus Pearce's IDyOM ("information dynamics of music") system offers a "predictive" model of melody [Pearce2005]. It calculates uncertainty and surprisal for each event in a melody using a Markov model. The Markov model is built by taking frequencies of occurrences in a corpus as a stand-in for probabilities, and inducing a distribution over possible next terms for any melody prefix.

IDyOM uses multiple "viewpoints," so that subdimensional projections through a melody can be modeled: these include basic note features (e.g. pitch, duration, accidental, voice, etc.), features requiring some local or global context with respect to the note in question (e.g. interval from first note; place in the measure), features requiring pre-analysis, like position in phrase and key-tonic, and Cartesian combinations of these features.

IDyOM uses a *variable-order* Markov model in which multiple lengths of prefix are combined. It also uses two different Markov models at once, one with a "long term memory" of previously analyzed melodies, and another with a "short term memory" pertaining just to the current score. The *uncertainty* of each prefix can be computed with respect to the model (i.e. measuring the unpredictability of the *next* event), as well as the *surprisal* of each event (i.e. its unexpectedness given its prefix).

The Markov model built by IDyOM is induced through memoization and summarization of previously encountered data (i.e. "learned"). Subsequences that have already been encountered will be counted as familiar and therefore less surprising. Abstraction is made by considering subdimensions, so that a sequence can be recognized when it recurs if it is not completely identical, but only identical from a subdimensional viewpoint (e.g. two sequences of notes with identical durations but different notes, or a sequence of the same pitch-intervals but different pitches).

The *memory* model of music in IDyOM contrasts with structural analysis because structural analysis is based on the assumption that there are basic structural properties in music that are available without reference to other music. An example of a simple structural property is that some feature (e.g. pitch, loudness) is *increasing, decreasing*, or *staying the same* over some timespan. Another structural property could be that for a while only three different notes are used, but at some point a fourth note is introduced. A structural analysis approaches the design of a score from properties like these without reference to other scores, whereas IDyOM aims to compute how like or unlike other "remembered" situations a given musical position is, based on multidimensional sequence-matching.

Although IDyOM is multidimensional in constructing Markov models with variable prefix length and multiple viewpoints, the analysis *re-linearizes* all of these into a sequence of predictive measurements for moment-by-moment uncertainty and surprisal. The result looks like a jagged line, unexpectedness over time: Figure 2–9

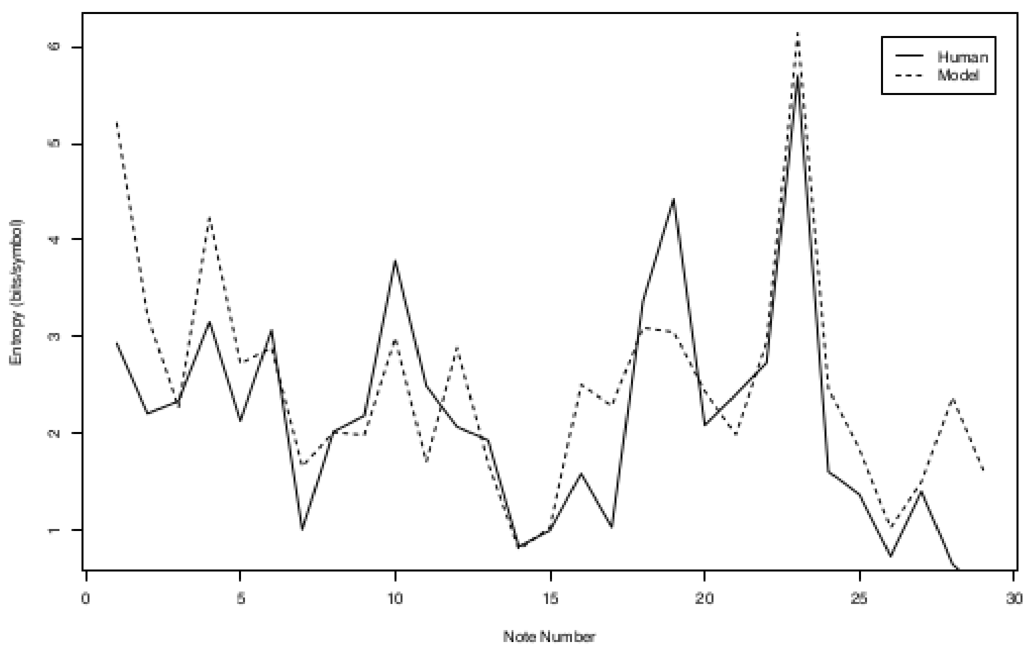

Figure 2–9: From [Pearce2005, p.170]. Comparison of modeled entropy over time with humans betting on the next note, for the melody of Chorale 151 from the Riemenschneider edition of Bach's chorales.

shows such a chart, compared against humans playing a betting game to guess the next note. The re-linearization is performed in order to generate a *predictive* model. A *descriptive* model, by contrast, would not perform a dimensional reduction, but would maintain a *map* of structures on the score.

In the IDyOM analysis, the operational model of "time" is one where each event is part of a sequence, each linear prefix can be used to predict the next event, and each next event is more or less surprising from the point of view of its prefix. In contrast, the kind of structural analysis proposed in this essay uses a model of the temporal axis that allows an event to be part of a structure (an appreciable shape or pattern) that is not yet complete, in which only *some* of the preceding (and succeeding) events may participate. Local shapes furthermore can be positioned as taking place in larger scale structures that may be distributed over the score.

2.3.2 Structural Information Theory

Structural Information Theory is a theory of (primarily visual) perceptual organization. Structural information theory develops a systematic account of perceptual *regularities* (– iteration, alternation, and symmetry) that are available in an image [Leeuwenberg+2013]. The regularities in an image can be *organized* so as to determine a "simplest" way of seeing the image – which turns out to be how it most often *is* seen.

In order to predict the simplest view of an image, the following analysis takes place. A human analyst *interprets* the image (e.g. as an object or scene consisting of several parts), and encodes geometric features of the image-as-interpreted as an alphabetic string. The image is interpreted several times, so that the interpretations can be compared. Then each encoding is parsed into a *tree* of regularities (iterations, alternations, symmetries), such that a complexity metric on the resulting parse is minimized (– the step of finding the simplest parse of a symbol-string is computable). The interpretation with the least complex minimal-parse is predicted to be the preferred interpretation (a prediction borne out by experimental results).[14]

Figure 2–10 shows an example of what is meant by "how" an image is seen: there are several ways of interpreting the image (i.e. segmenting it into parts), with some of them seeming *simpler* or *more obvious* than others – structural information theory is a formalization that seeks to measure the perceptual simplicity of possible interpretations, predicting which will be most obvious to viewers.

Figure 2–11 shows how the regularities (iteration, symmetry, and alternation) are used to structure symbol strings; Figure 2–12 shows one symbol string with two different structural parses.[15]

---

[14] Structural information theory bears a relation to *Algorithmic Information Theory*, in which minimum-description-length encodings are also used to quantify complexity; but structural information theory is based on the *perceptual* relevance of *particular* regularities based on their empirical predictive power with respect to perceptual tests.

[15] We omit explanations about how to interpret an image as a symbol string, because this is specialized and intricate and doesn't bear on the current discussion. Details can be found in [Leeuwenberg+2013].

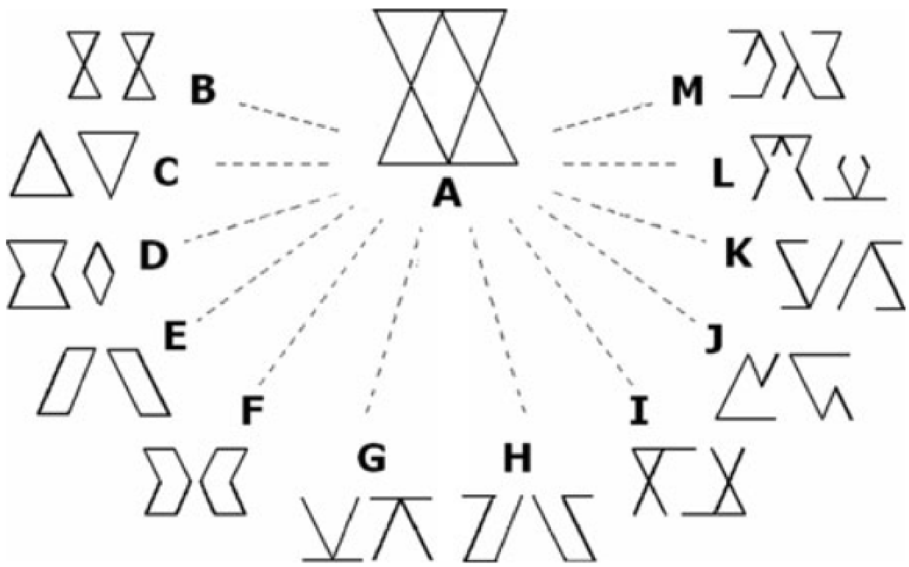

Figure 2–10: Figure 1.11 from [Leeuwenberg+2013] (© Cambridge University Press). Image A is interpreted in several different ways, of which B and C are the most plausible and are about equally plausible. This is a very simple example; 3D scenes with occlusions can also be represented.

	strings		codes
iteration	AAAAAA	←	6*(A)
	ABABAB		3*(AB)
symmetry	ABCCBA	←	S[(A)(B)(C)]
	ABCBA	←	S[(A)(B),(C)]
	ABCAB	←	S[(AB),(C)]
	ABCDAB	←	S[(AB),(CD)]
alternation	ABACAD	←	<(A)>/<(B)(C)(D)>
	ADBDCD	←	<(A)(B)(C)>/<(D)>
	ABCABD	←	<(AB)>/<(C)(D)>
	ACDBCD	←	<(A)(B)>/<(CD)>

Figure 2–11: Figure copied from [Leeuwenberg+2013, p.110]. The caption reads "A few symbol strings with codes to illustrate how the three kids of regularity (iteration, symmetry, and alternation) are formally represented. The arrows point left to indicate that a code represents just one pattern whereas a pattern can be represented by various codes."

String:	a b a c d a c d a b a b a c d a c d a b
Code 1:	ab 2*(acd) S[(a)(b),(a)] 2*(cda) b
Organization:	ab(acd)(acd)(a)(b)(a)(b)(a)(cda)(cda)b
Code 2:	2*(<(a)> / <S[((b))((cd))]>)
Organization:	(((a)(b))((a)(cd))((a)(cd))((a)(b)))(((a)(b))((a)(cd))((a)(cd))((a)(b)))

Figure 2–12: Figure copied from [Leeuwenberg+2013, p.99]. Two encodings of one symbol string. We haven't formally defined the complexity metric, but to give a feeling for it: *Code 1 has complexity measure of 14, "because it contains twelve symbols and two chunks that contain neither one symbol nor one S-chunk, namely, the chunks (acd) and (cda)," and Code 2 has a complexity measure of 8, "because it contains four symbols and four chunks that contain neither one symbol nor one S-chunk, namely, the chunks (cd), ((b)), ((cd)), and the repeat of the I[teration]-form."* [Leeuwenberg+2013, p. 99]

Structure, superstructure, and contradiction

Although structural information theory is a perceptual theory developed by psychologists, and is not[16] a computational methodology for image processing, it offers a conceptual contrast to the information-theoretic, memory-based model of data processing shown in IDyOM (and currently prevailing everywhere). It affords comparison and contrast with structural analysis as we are attempting to define it, as well as offering opportunity for a discussion of *perception*, a necessary question when studying music.

The *structural* approach to information theory contrasts with the "learning" or memory based approach in likelihood or Bayesian based methods, because it posits that certain things are (or may be) obvious, salient, or relevant to the construction of a perception, regardless of how often the particulars of the stimulus have been encountered. In this way, structural information theory is similar to the structural analysis that is the subject of this essay.

In structural information theory, the symbol-string representing the image is parsed into a *tree* of regularities. The tree is a *hierarchy* of structures, with substructures together forming superstructures. This is in contrast to the subdimensional string-matching model of IDyOM, in which no superstructures are formed.

Substructure and superstructure are a part of the structural analysis we propose – *structures* are often hierarchical (for example a set or a chain *of* pitch chains or pitch-class sets). But a structural *analysis* (unlike a structural-information-theory

[16] yet?

analysis) is usually *not* hierarchical, since it contains many different structures that do not have this relation.

A tree of regularities in structural information theory is constrained to be *complete* and *consistent*. It is *complete* in the sense that it must describe the *entire* image (or symbol string), and *consistent* in that it does not contain "contradictory" interpretations of any part of the image. This contrasts with the structural analysis we propose, because we focus on finding *partial* descriptions that describe *some* structural relations, without trying to find subsets of these structural relations that cover all points with no contradictions.

As a simple example, take the term pattern ABABCABC. The term pattern has two iterations: ABAB and ABCABC, but these cannot *both* be used in a tree of regularities, because they overlap and cannot be nested hierarchically. Structural information will be lost when a decision has to be made to construct a tree that will include only one of these regularities. In contrast to the decision process in structural information theory, our analytic challenge is not to choose an interpretation, but to organize the existence of multiple "contradictory" structures.

Our attitude toward "contradiction" in structure and perception is in contrast to structural information theory, perhaps because of our different fields of study. Structural information theory deals primarily with visual perception, where there is a sense of *veridicality* – seeing things as they are versus illusions and misperceptions. An image is typically seen *as* an object or scene. Objects and scenes are physical things that have certain properties, and are *represented* by images that must be interpreted to infer properties of the physical things. That's the game of a certain kind of visual perception – but we don't take it for granted that music perception plays a similar game. Because there are no veridical objects in the musical domain (and very often nothing is being *represented*), musical structure can be multiply determined and can be heard as such.

In music (as in some other artforms, including some visual ones), where there is no requirement of veridicality, there is value in obtaining more than one interpretation. Creativity can involve *re*interpretation – the ability to perceive in a new way or in several different ways, perhaps bringing to light *less* salient, "hidden" structures. Therefore, while the structures we describe could be used to devise predictive tests about how people most often interpret things, our position is that music listening is a fundamentally more open act of perception, so we try to devise ways of computing with *many* possibilities for interpretations.

When multiple interpretations are available in a stimulus, there is sometimes a switching effect, where perceptual interpretation goes one way and then another, or primarily different ways for different people. But even in cases where one interpretation is much stronger than the other, structural cross-currents can be relevant to (and sometimes central to) design and its interpretation.

A musical example is the opening measure of the *Presto* from the sonata in G minor for solo violin by J.S. Bach, in which the notated meter suggests groups of two notes, while the while the pitch-patterning suggests groups of three notes, and no performance indication is given to resolve the ambiguity (Figure 2–13). According to [Lester1999], no matter which way a violinist performs it, the *other* interpretation

Figure 2–13: *Presto* from the sonata in G minor for solo violin by J.S. Bach. The 3/8 meter suggests groups of *two* notes (i.e. the measure is to be divided into 3 parts), while the while the pitch-shapes suggest groups of *three* notes. Structural ambiguity is part of the design of this score.

remains audible and is sometimes selected by other musicians (in an informal experiment), "confirming that the metric ambiguity here is so deeply embedded that some residue of it projects no matter how hard the violinist aims for a single version." Whichever way the passage is "heard," (and it's not certain that a *decision* is *always* a perceptual imperative), the multiple metric affordances contribute to the effect of the passage, and are fundamental to the design of the score.

Structure and (Perceptual) Simplicity

Structural information theory, like the structural analysis proposed in this essay, is based on the availability of universal, simple structures.

Structural information theory (like the IDyOM model) is a predictive model for *perception* of the data it analyses. Structural analysis is descriptive *of scores*, and does not take any direct position on perception. Structures are more musically useful, however, if they are *available* to perception.

Structural information theory offers a *specific* set of structural regularities for vision (iteration, alternation, symmetry), that are empirically validated as being important for visual perception. In contrast, we offer some *examples* of structures for music analysis, but these are not meant to be a closed or fixed set, nor are they empirically validated with regards to perception. It is to be hoped, however, that these *are* (sometimes) available to perception.

One basis for this hope is that structural analyses often have a quality of straightforwardness, simplicity, or obviousness. Structural information theory also values *simplicity* as a criterion for salience but while they offer a predictive *metric* for simplicity, we mention simplicity only as informal quality of being straightforward and obvious. Part of this characteristic is accomplished by the external-context-independence of structural analysis, which generally forces structural relations to be made up of relatively local feature-comparisons – which are usually evident to perception, and not open to much interpretation.

For instance, the following are structures which, when pointed out, are easy to perceive: "this event happens during the timespan of this other event," "this series of events has a monotonic increase in this feature," "this temporal span contains a small set of kinds of events, and the temporal span is delimited by different kinds of events." These are "easy" to apprehend when pointed out, and tend to elicit little inter-subjective or inter-contextual disagreement.

More complex structures are recursions of structures like these. They may be less salient or easy to mentally or perceptually construct, and may be easier to "see" than to "hear." They may operate musically on a larger-scale expressive level, which is qualitatively (i.e. semantically) distinct from a momentary perceptual level, but yet which contains the same kinds of ("obvious") structural relations, and thus is part of the same fractal structural medium.

Because of the straightforwardness of the structures, structure-finding algorithms tend to be computationally straightforward as well, without need for advanced algorithmic techniques. A score annotated with structures sometimes seems a "too" obvious result. It seems that nothing has really been discovered; that nothing *new* was revealed about the score. It's easy enough to draw boxes around pitch-class sets, or to trace lines over ascending melodic passages, and the computational setting offers nothing more than a convenience in doing this. This is a good sign – that the relational structures gathered are not obscure, and therefore have a chance of being perceptually available.

We are not trying to discern *mysterious* properties of scores, we are trying to capture *obvious* ones. Ones that as musical thinkers and perceivers, we find natural to comprehend.

The complexity is not in finding the simple structures, but in organizing a large multidimensional network of them. This is something that the perceiving mind does only partially and impressionistically. What a structural analysis does is not to predict what will be perceived or effective, but to make a map of (what may be) perceptual affordances of the score.

2.4 The Structural Sense

Structural score analysis doesn't directly address perception – but since the phenomenon of music is based on perceptual affordances, ideas about perception are necessary to motivate any analysis. This section provides such a motivation by describing a "sense" of design that is accessible through external sensory modalities like vision, audition, and touch, as well as being available to abstract thought.[17]

2.4.1 Pattern and Shape: Whole and Parts

The sense of design has to do with the subjective apprehension of things like *pattern* and *shape* – a perceptual *organization* level in which structural relations may participate in the cognitive construction of a perceived scene. *Pattern* and *shape* are more or less intuitive words for the subjective apperception of structure.[18]

Pattern and shape are *qualitative* in the sense that everything is that's subjectively apperceived, but we suppose that the underlying structural basis is firm – we may expect individual differences in the *saliencies* of patterns and shapes (i.e. their ready availability to perception), and in their semantic interpretation, but not in their construction as such.

Shape and pattern are two aspects of the sense of design – these are not different senses, but duals. The duality has to do with whether a figure is taken as a *whole* thing (made up of parts): a shape; or as a configuration of individual (but interrelating) things: a pattern. The difference is subjective, conceptual, qualitative, contextual, optional – but perhaps not structural.

[17] The descriptions of perception here are neither empirically tested nor argued very closely – no *claims* are made in this section: it is an informal conceptual background about how structure might relate to human perception, cognition, and art.

[18] "Apperception" is consciousness or reflective knowledge of perceptions.

A sense of *shape* describes a relation of parts, in which the parts together form something continuous at some level – a square, for example, has a configuration of angles and sides, but it can be sensed as a shape: □ , as can any of the typographical glyphs. A shape (as sensed) may be more or less regular and more or less complex. For a stimulus that is complex, the sense of shape may offer partial pictures at many levels – an overall sketch, as well as specific details of shape in different regions. A simple shape can be appreciated as a unity but a more general sense of shape doesn't demand closure, a view of the entirety, or a clear separation of one shape from another.

The sense of *pattern* is active when parts do *not* seem to form something continuous, but are instead taken as a collection of independent (but interrelating) things. A low-dimensional example is a term pattern: AABABBABC. Like a shape, a pattern may be of variable regularity and complexity. It may have several levels of organization or detail, and it may or may not be of fixed size or determinate identity.

Pattern and shape are inter-recursive: they can operate simultaneously at different levels of detail. Shapes can be made of smaller shapes and/or patterns, and patterns of smaller patterns and/or shapes. For example, the square is made of a pattern of lines and angles, which are themselves shapes. Likewise, a term pattern can be taken as a sequence of abstract terms, or as a series of glyphs – a pattern of shapes.

Pattern and shape can also operate simultaneously on the *same* level of structural resolution: the same structure can be seen as a shape or a pattern or both. A shape can have a pattern, or a pattern can have a shape. For example, the following

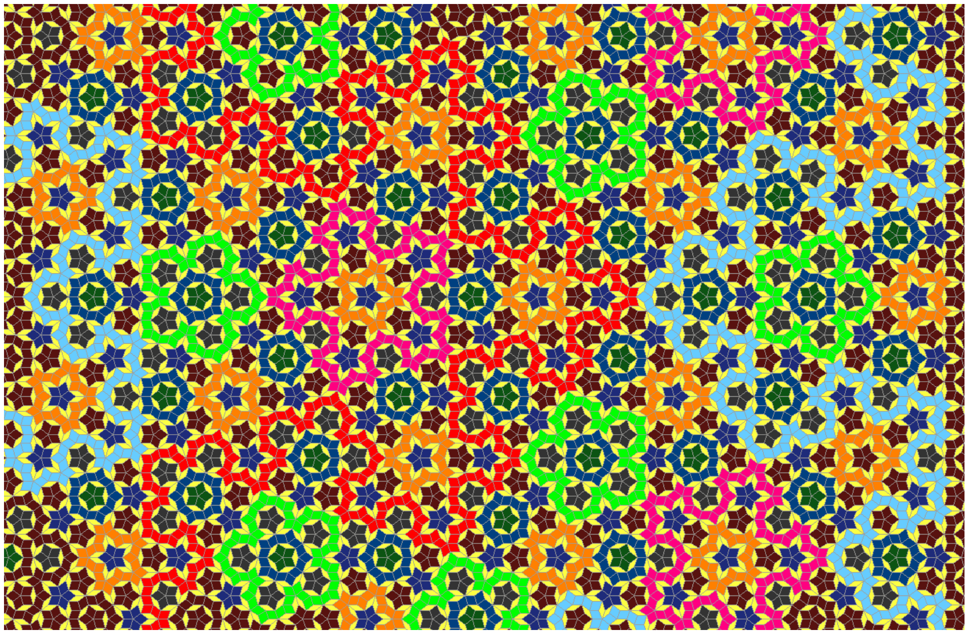

Figure 2–14: In this Penrose tiling, a pattern of shapes forms patterned super-shapes. Patterns of tiny colored shapes (quadrilateral tiles) give rise to larger shapes (stars, circles, flowers, coastlines), which are themselves patterned. "Zooming out" affords a sense of higher-level shape or pattern (e.g. the framing on the right and left by pale blue coastlines along with the individuated green flowers in between them), while the details of the middle region with its similar colors recede into a sense of texture. (Image © Xah Lee. http://xahlee.info/math/algorithmic_math_art_2.html)

term patterns "have shape" if considered synoptically: ABBBCBBBA ABBAAAA AAAABBA.

A shape is "really" a pattern of its parts, and a pattern is simultaneously a shape: as dual perspectives, they are two ways of looking at the same thing.[19]

[19] The interoperability between pattern and shape is why Structural Information Theory is able to make predictions about the perception of geometric shapes by analyzing abstract term patterns.

Figure 2–14 shows a Penrose tiling, which has multiple levels of organization available as pattern and shape. It is made of are quadrilateral tiles of nine different colors and two different shapes. These are patterned in two-dimensional space. Some parts of the pattern can be grouped into larger shapes: stars, circles, flowers, and coastlines, which themselves afford pattern and shape on a higher level.

2.4.2 Seeing and Hearing Pattern and Shape

Shape and pattern are accessible in visual, tactile, and auditory perceptions. Physical objects sometimes offer their shapes and patterns to both visual and tactile perception. The physical stimuli that afford auditory shape and pattern are not available to vision or touch. Auditory shape and pattern necessarily have a temporal duration, while visual and tactile stimuli sometimes do. Auditory shapes and patterns, however, are the same kinds of things as visual ones: the same *sense* of pattern and shape is active through different perceptual modalities.

Pattern and shape are both common informal terms in talking about music. Figures 2–1 to 2–7 show piano-roll visualizations of musical scores, in which pattern and shape are apparent.

Looking at a score is not the same as hearing it. A score can be visualized and seen, or sonified and heard – but are the *same* patterns and shapes available in both experiences? Following a visual score while listening demonstrates that the resulting experience is (at least) coherent from the point of view of observing patterns and shapes. But this confounds the issue of what is available in music *without* visual reinforcement.

A visualization of a score is not a (quasi-linguistic) translation, or a description of the score, it's an inter-modal transduction, a projection of data onto new perceptual axes (heuristically, but as directly as possible).[20] That musical pitch goes "up" and "down" is not a metaphor: *up* and *down* are not "real" only in reference to spatial axes: they refer to quantitative orientations *generally.* If *up* and *down* mean the "same" thing on different perceptual axes, the same shapes that are *made* of ups and downs are also coherent when perceptual axes are swapped – though this doesn't mean they will be qualitatively similar or similarly salient.

A projection from one perceptual mode to another changes the low-level sensory qualities, and thus the feeling, meaning, and expression. The phenomenon of music is *not* available visually. The visible patterns and shapes may be "the same ones" available to hearing – but they have different effects, and a different balance of saliencies. But since patterns and shapes *are* available inter-modally, the observations made by looking at a drawing of a score *can* be used to think about musical pattern and shape.

2.4.3 Time and Motion

Shape and pattern are observable in both temporal and non-temporal senses. A mapping between spatial visual perception and temporal hearing preserves pattern and shape – but not qualities or saliencies.

[20] This is distinct from music *writing* (e.g. staff notation), which involves a deeper kind of literacy.

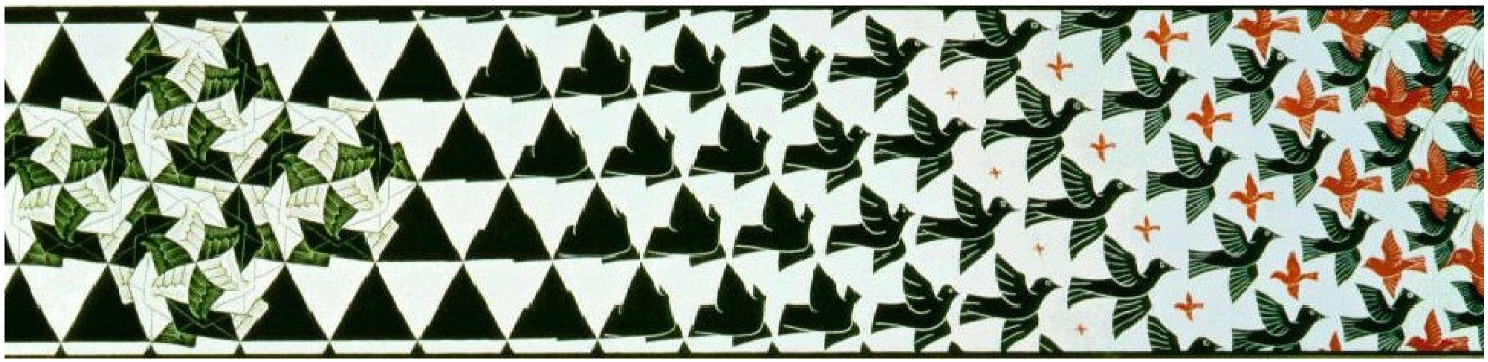

Figure 2–15: Excerpt from Metamorphosis II by M.C. Escher. Motion is suggested as change over "time" in this non-temporal work. (All M.C. Escher works 2016 The M.C. Escher Company - the Netherlands. All rights reserved. Used by permission. www.mcescher.com)

Figure 2–16: Frank Gehry: 8 Spruce Street, New York, and 888 W Bonneville Ave, Las Vegas. These buildings project an imaginary temporal axis by depicting a flexible substance in motion. (Top image © Ken Peterson https://www.flickr.com/photos/kenpete/5529589586; bottom image © D. Laird https://www.flickr.com/photos/dottieg2007/6085551117.)

Temporality is a *property* of a *physical* sensory medium. A *sense* of temporality (or motion) is a *quality* of perceptual experience. Unlike pattern and shape (and *like* other *qualities*), it is *not* directly transferable across perceptual axes – though the perceptual affordances underlying it may *involve* pattern and shape, and these affordances may have correlations across various axes (e.g. the kinds of shapes and patterns that portray acceleration).

The subjective sense of temporality can involve a process of change, a sense of going in some direction, or kinematic effects like bouncing, leaping, swooshing, tumbling, galloping, swirling, spiraling, floating, crashing, coasting to a halt, gaining momentum, belly flopping, dancing, gesticulating, sauntering, swaying, or falling to pieces.

The feeling or idea that a dimensional axis is *temporal*, or that something is in *motion*, can occur whether the physical medium is temporal or not.

Time and motion can be depicted in non-temporal media: Figure 2–15, by M.C. Escher, depicts transformation as spatial "change." A different way of showing motion on a spatial axis through patterned shape is apparent in Marcel Duchamp's "Nude Descending a Staircase No. 2" (not shown). Figure 2–16 shows buildings by Frank Gehry that project an imaginary temporal axis by depicting a flexible substance in motion.

Qualitative motion in music includes progressions, transformations, narrative sequences, grooving, rocking, dancing, and various other kinematics. Music also can convey phenomena that are less temporal (notwithstanding the fact that these occur *at* temporal moments in the music), including shapes, symbols, rhetorical or expressive statements, personalities, attitudes, atmospheres, and scenery.

The apparent sense of motion or temporality that is sometimes present in music is *correlated* to physical time through the system of perceptual and cognitive affordances that creates the musical experience – but it is not a *direct* result of the physical temporality of music. Motion in music is a *depiction* of motion. It is qualitative, subjective, and optional.

The view of temporality as a *property*, a *quality*, and something that is *depicted*, helps to explain why pattern and shape *are* coherent when translated to or from the physical temporal axis. It locates perceptual temporality as analyzable only *with respect to* the specific affordances of the senses and the mind, and not as directly resulting from either the physical medium or the particular patterns or shapes in evidence.

2.4.4 Texture

Texture occurs when pattern elements are too small, too many, too disorganized, or otherwise too undistinguished to be considered individually. A *statistical* or aggregate quality can result, in which case a structural or pattern/shape analysis somewhat misses the mark. But the boundary between structural and statistical perceptual modes is not simple, and a texture may become a pattern or vice versa dependent on context (e.g. when given more attention). A texture may have emergent shape or occasional individuated elements that form patterns. A texture also may depend on micro-patterns or micro-shapes in such a way that these are subliminal but qualitatively active.

In music, "texture" is often highly (structurally) organized and made of elements well above the limits of perceptual differentiation. Texture is visible in piano-roll depictions of scores (Figures 2–2–2–6).[21] *Texture* in music refers to a general, imprecise but definite, sense of how different elements are fitting together at one moment.[22] For example, the elements may be close to each other in pitch, or some may be very high and some very low. There may be only one note at a time, or several independent voices, or a primary voice with a background. All voices may speak in the same rhythm, or one faster than another, or some regular and others irregular, or two voices may alternate moving and holding. Texture in music is *designed* – *patterned* and *shaped*, and *made of* patterns and shapes. But yet it may be (partly) qualitatively constituted of statistical or aggregate effects.

Texture is a liminal perceptual mode, characterized by imprecision and generality, introspectable mostly out of the corner of the eye or ear, but producing definite qualitative effects. It is a transitional mode of the sense of design (pattern and shape) as it turns into something like *color*.

2.4.5 Color

Colors are characteristically *qualities*, or qualitative identities, as are instrumental timbres in music.

---

[21] If the pictures were tapestries, what kinds of *weaves* would they have? If they were braille, what would they present to the sense of touch?

[22] I.e. a very short non-instantaneous timespan, e.g. on the order of 2 seconds

Colors and timbres are *orientable* on different axes (redness, brightness), affording shading, gradation, and detail. But their underlying physical structure is hard to directly introspect, and as a result discontinuities in color are often perceived as *un*oriented contrasts, or differences.

Color can be *patterned* and *shaped* as part of a design, but since colors seem to have an irreducible and unintrospectable qualitative identity, it isn't normally available as a "designed" element itself, but as a *given* material for use in design. For example, there seems something "absolute" and given about *blue*, because we can't introspect it as a structure of metric properties of light (or *describe* it in any other way).

Another kind of color in music is *harmony* – for example the qualitative sense of a major triad in contrast to a minor triad. As in visual color perception, the harmonic quality of tones is determined in relation to their participation in a context with other tones. We know that visual color perception is *contextual* (i.e. relative) because the subjective identification of a color depends on its surrounding colors *without our being able to introspect this process.* Nonetheless, we have the *sense* that our color perception is "absolute" because we identify red *as* red – an absolute qualitative value. In contrast, most people don't have the sense that their harmonic pitch perception is absolute – the harmonic qualities of a score are invariant under transposition. A small minority of people have a sense that different pitches *do* have different absolute values, as well as contextual values.[23]

While we have some sense of how color is constituted out of measurable properties of light, and harmony of combinations and contexts of pitches, both of these phenomena display the opaqueness of the structure-semantics interface. This is the *mind-body problem* that we inevitably arrive at, since we are in a Cartesian universe.[24]

2.4.6 Expression

All perception, at *all* structural levels, comes with *quality*, and therefore with a mind-body problem. At a basic level, this is color, timbre, and harmony. At higher

[23] People with *absolute pitch* may have more or less "absoluteness" depending on how consistent the correlation is between their "absolute" sense and the physical signal. Absolute pitch is sometimes characterized as the ability to identify or reproduce physical pitches without context. But when describing the musical experience, what is more relevant is the *sense* of identity of the different pitches, since this offers a new qualitative level with cognitive advantages and aesthetic potentialities. These two characterizations are correlated, but may be present in different ratios.

[24] The kind of structural analysis we describe is based on Cartesian geometry. Descartes also posed the *mind-body problem*, (or "substance dualism"): mind and body are two different *kinds* of things, and we don't understand how they relate and interact with one another.

Descartes observed the mind-body problem in color, proposing that it is a *quality* that is modellable by physical *shape* (– before Newton described color in terms of physical properties) [Clarke2003, p.47].

Any structure-semantics interface (i.e. at which a *quality* or *human-level meaning* is created) is an example of the mind-body problem. These interfaces are everywhere when human consciousness is concerned; music has to do with a special class of them.

levels, texture, pattern, and shape have characteristic qualities and can produce qualitative effects such as moods and motion.

The *highest* qualitative level is that of *expression*. At this level, artifacts exist that are arbitrarily structurally complex or simple, that may be technically inventive (i.e. posing new mind-body problems), and that are contextually integrated into an expanding world with no boundaries. This is the level of (human) meaning, with all of its referential, resonant, linguistic, personal, cultural, and psycho-physical complications. *Expression* (whether real or supposed) is the bodying forth of these objects as actualized ideas. Expression may have its own purposes, e.g. aesthetic, artistic, pragmatic, or educational.

Medium and quality, structure and expression, interpenetrate each other at *all* levels. A structural analysis doesn't *explain* the phenomenon of music, and also can't motivate its production. It is an algebraic descriptive language for the investigation of musical artifacts at the level of the *medium*, the *means* of expression. It produces a map not of music's ideal territories, but of its *real* potentialities.

CHAPTER 3
Polyphones

3.1 Introduction and Definitions

If a *phone* is a temporal unit of sound (e.g. a note), *polyphony* is a multiplicity of phones. Melody, a *sequence* of phones, is a base case for polyphony; another base case is a chord in which all notes start and end together. A score can contain any number of notes not only sequentially or simultaneously, but in arbitrarily complex configurations of temporal inclusion and overlap. Any such configuration of temporal relations is called a *polyphone*. Polyphones generalize sequences and simultaneities. They also generalize *trees*, by representing both inclusion *and* overlap.

Temporal (or geometrically one-dimensional) coincidence is a basic and natural structural dimension. A potential non-musical application is *sensor fusion*, in which data is obtained from a number of different sensors, and the problem is to get an overall picture from the union of the sensors. In musical application, we need not stop at polyphones of notes: we can treat any structure as a phone (if it can generate a temporal interval), and then study the temporal relations between these.

To construct polyphones, each phone supplies its temporal interval as an (i, o) pair of open endpoints (i for "in", o for "out," with $i < o$). These are points on a timeline (or other dimensional line) in any measurement unit. Polyphones are *networks of temporal relations* between phones (– the *absolute* positions and lengths of the phones are not used in constructing these relations).

Monophones

The simplest relation between two (i, o) intervals is identity: the *monophone* relation $M(p, q) \leftrightarrow ((p_i = q_i) \wedge (p_o = q_o))$.

Because this is an identity relation, in a graph setting (i.e. taking phones as a set of *nodes* $\mathcal{N}$ and drawing edges $p \to q \ \forall (p, q) \in \mathcal{N} s.t. M(p, q)$), we obtain a set of (bidirectional) *cliques*, where each member of a clique has identical polyphonic relations to other phones on the score. It is therefore unambiguous to treat each monophone clique as a single *monophone node* in a graph, so that each node refers to a *set* of phones with the same (i, o).

Hold and Fold Relations

We define two relations on *monophones* p and q with open endpoints (p_i, p_o) and (q_i, q_o), corresponding to containment (H for "hold") and overlap (F for "fold"):

$$H(p, q) \leftrightarrow \text{"q is contained in p"} \leftrightarrow (p_i \leq q_i) \wedge (q_o \leq p_o) \wedge (p_{(i,o)} \neq q_{(i,o)})$$

$$F(p, q) \leftrightarrow \text{"p overlaps with q (on p's right)"} \leftrightarrow (p_i < q_i) \wedge (p_o < q_o) \wedge (q_i < p_o)$$

At this level of generality, we have dealt with three of the four possible relations (identity, containment, overlap); the fourth is the null, disjoint relation.[1]

[1] Students of music écriture (in a widely practiced tradition dating from the 16th century) learn to write polyphonic music using "species" of two-part counterpoint in which all of these relations are covered: first species concerns temporal identity (note against note), second and third species have hold relations (shorter notes contained in longer notes), fourth species uses fold relations (overlapping notes), and fifth species mixes all of the previous species.

A higher degree of specificity would handle shared endpoints, as when $(p_i = q_i) \wedge (p_o > q_o)$. These kinds of relations (which we return to briefly in Section 4 of this chapter), still count as "polyphone" concerns, while questions of absolute and proportional length do not.

Polyphone Graph

Any set of monophones $\mathcal{M}$ forms a graph where $\mathcal{M}$ are nodes and the edges $\mathcal{E} = \mathcal{F} \cup \mathcal{H}$ are defined by $\mathcal{F} = (\forall(p, q) \in \mathcal{M} * \mathcal{M}$ such that $F(p, q))$, and $\mathcal{H} = (\forall(p, q) \in \mathcal{M} * \mathcal{M}$ such that $H(p, q))$. These edges are directed and typed either F or H.

Any such graph on a set (or subset) of monophone nodes is a *polyphone graph*; given a set of nodes, the *full* set of its edges is required in order to faithfully represent the relations within the set. Therefore whenever we speak of a polyphone *subgraph*, this means a subset of *nodes*, and *all* of the edges among them.

The polyphone graph is a *sequential* graph, in which nodes can be ordered by a $<$ relation: $p < q \leftrightarrow (p_i < q_i) \vee ((p_i = q_i) \wedge (p_o > q_o))$. This can also be represented as ordering on $(p_i, -p_o)$. As a result, $H(p, q) \rightarrow (p < q)$ and $F(p, q) \rightarrow (p < q)$. Therefore the polyphone graph is a *directed acyclic graph*, or dag. The relation $<$ is a defined node-order, so that we can refer to the "leftmost" (i.e. "least") hold-child of a node, or the "next" node in a graph. This sequentiality relation is also a looser stand-in for the undefined adjacency relation $(p_o = q_i)$.

A polyphone graph consists of a *sequence* of its connected components, called *polyphones*. Polyphones are separable by an instant of time that passes through the body of no phone, though it may pass through the open endpoints. Equivalently, we can define the set of polyphones in a polyphone graph as a partition (i.e. a set of maximal subsets) of monophones such that $\forall X, Y \subset \mathcal{M}, \forall x \in X, \forall y \in Y, \neg F(x, y) \wedge \neg F(y, x) \wedge \neg H(x, y) \wedge \neg H(y, x)$. These polyphones form a sequence ordered by $<$, since for different connected components X, Y of a polyphone graph, $(x \in X < y \in Y) \rightarrow (\forall x' \in X, y' \in Y, x' < y')$

3.2 Visualizing Polyphones

We visualize polyphones by drawing ("abstract") graphs, and by superimposing graph drawings on musical notation.

Hold edges are drawn in blue, and are oriented downward. Only hold-edges that are minimal-length paths between two nodes are drawn – i.e. we draw parent-child relations but not grandparent-grandchild (etc.) relations. No information is lost by doing this, because $H(a, b) \wedge H(b, c) \rightarrow H(a, c)$. Therefore these undrawn hold-relations are implicit (and we must bear this in mind e.g. when thinking about subgraph relations). We call these contextual subrelations "direct" and "indirect" holds.

Graphs are read from left to right, with the sequence of hold-children appearing in $<$ order.

The graph of direct hold edges looks something like a sequence of *trees* (each consisting of one root and zero or more children, grandchildren, etc.): the difference is that these trees can *overlap* by *sharing nodes* – in particular, when two nodes *fold* with one another, they may share (direct or indirect) *hold* children. One way to think of this is that not only do the two folding *nodes* overlap one another, but they are parts of trees (or subtrees) that fold, and the shared children are an aspect

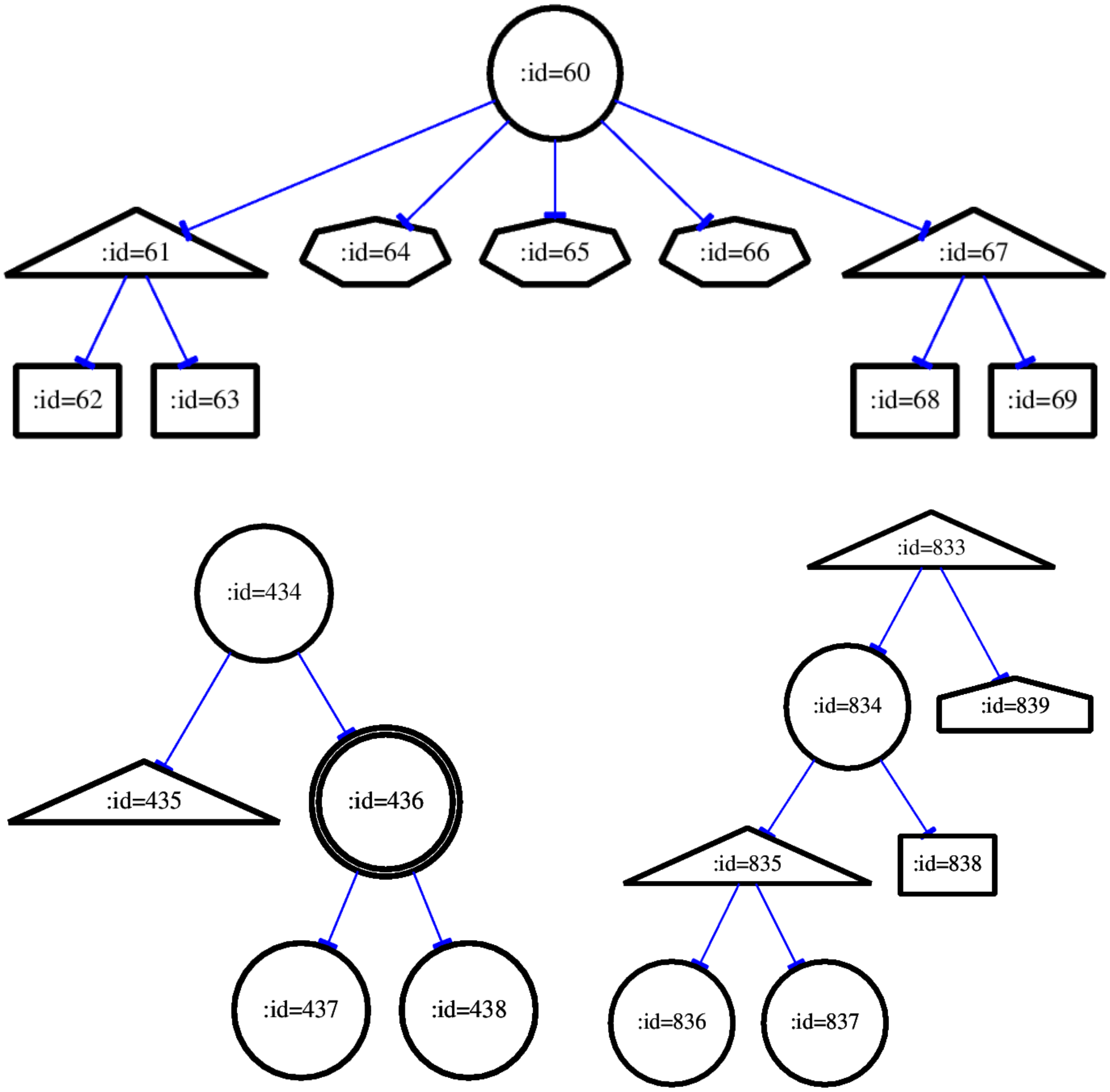

Figure 3–1: Polyphones. These polyphone components are hold trees, with no fold edges. Node shape indicates monophone cardinality.

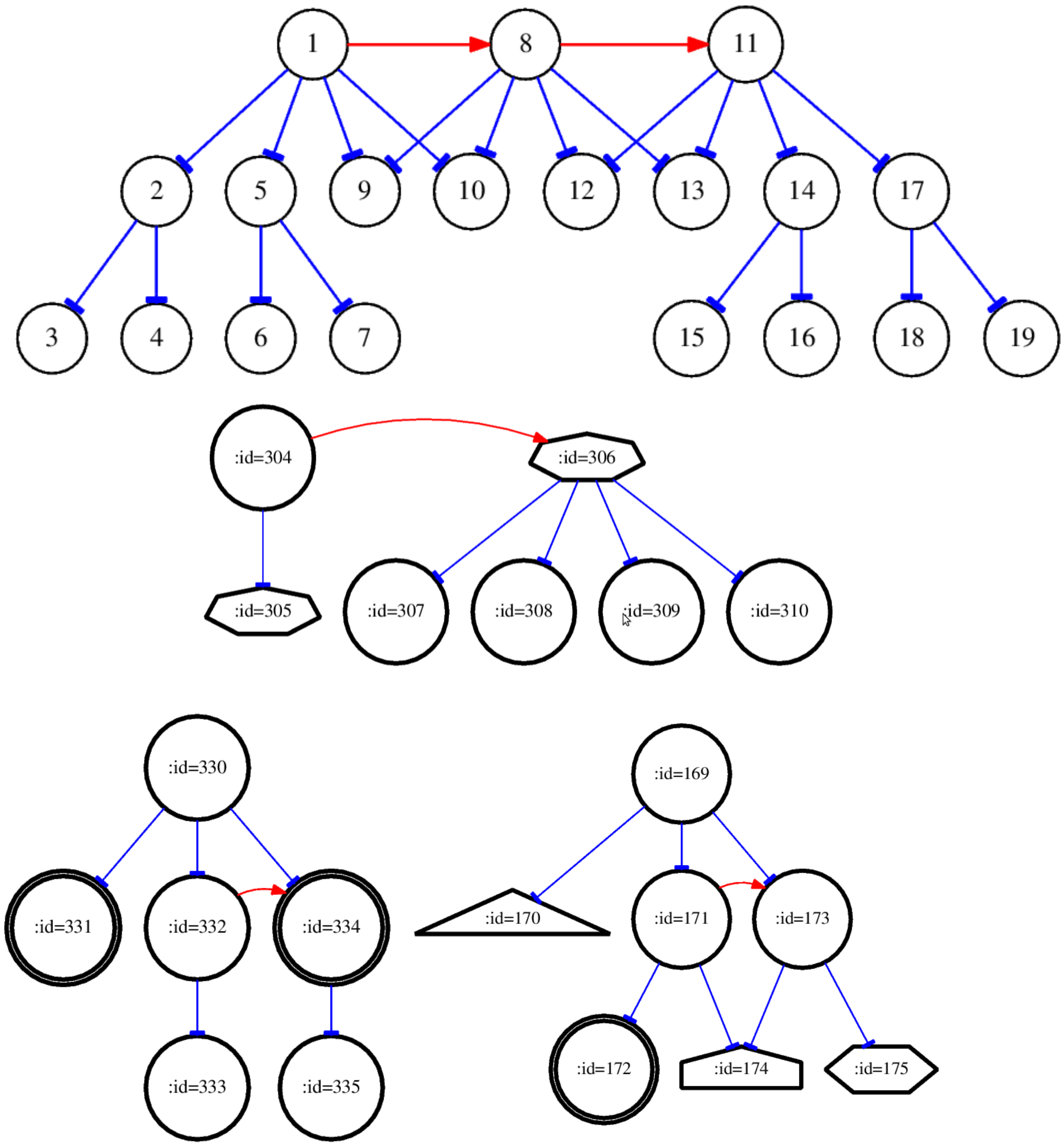

Figure 3–2: Polyphones. Blue edges are hold relations, red edges are fold relations. Hold *trees* that *fold* with one another may also share nodes (e.g. the top graph can be divided into three overlapping hold-trees with roots 1, 8, and 11).

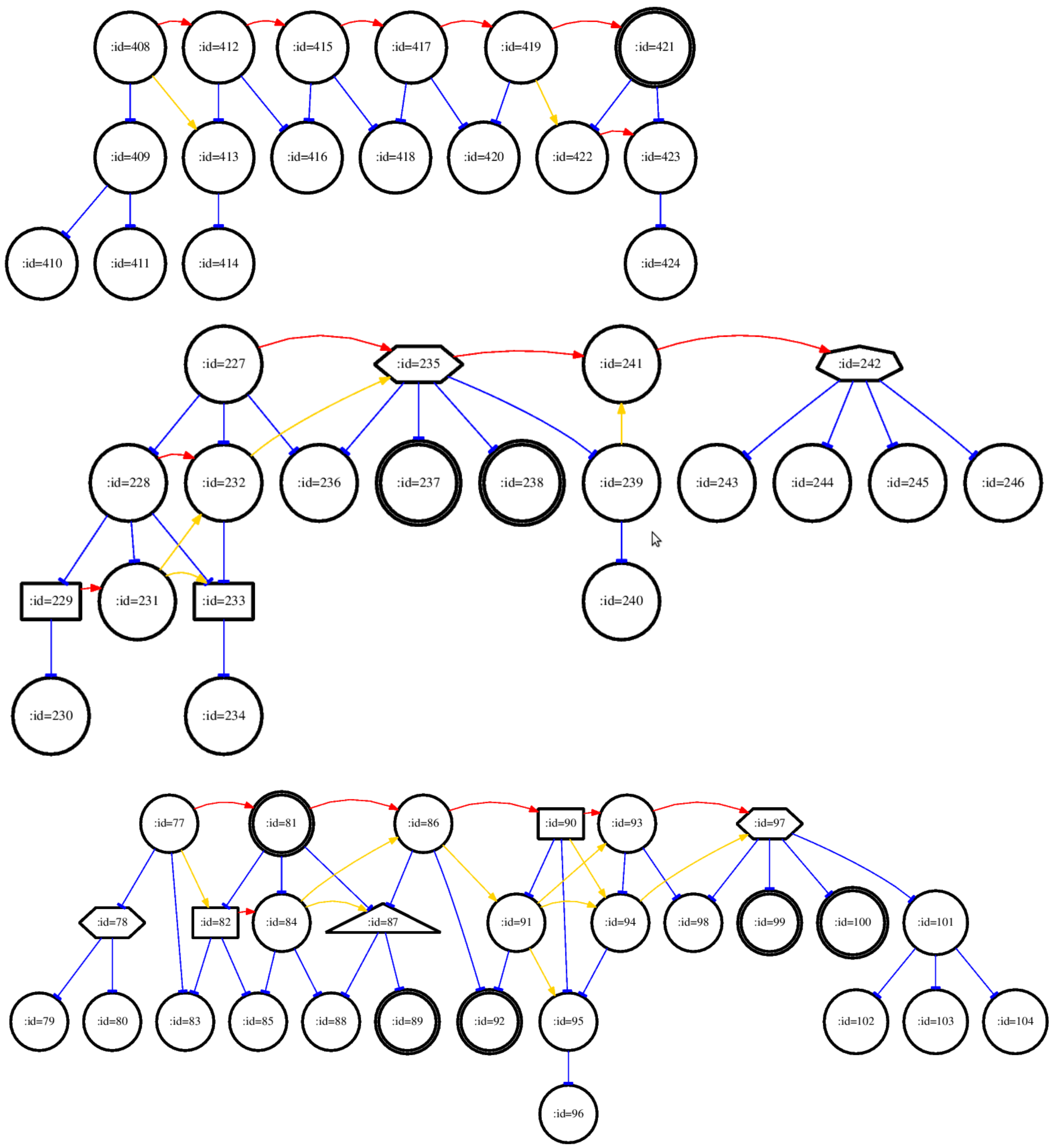

Figure 3–3: Polyphones. Blue edges are hold relations, red and gold edges are fold relations. Each red edge is the *head* of a fold family, with zero or more gold edges appearing beneath it. A fold family is the set of fold edges between two hold (sub)trees.

of this overlap. The graph of direct hold edges is a *Hasse diagram* on the nodes. A polyphone component with *no* fold edges is simply a *tree* of hold edges.

Whether folds are present or not, the structure of holds in the graph makes available many of the usual concepts for *trees* such as roots and leaves (– each hold-"tree" has *one* root, and each root has a corresponding unambiguous set of *leaves*), and accounting of *depth* and *height* for each node, based on its distance from the root and leaves.

Fold edges are oriented from left to right, and drawn in red and gold. A fold edge is drawn in gold if one of the two nodes also participates in a fold relation with a hold-ancestor of the other node. F(a,b) is gold if $\exists F(a, b')$ s.t. $H(b', b) \vee \exists F(a', b)$ s.t. $H(a', a)$. In this case we say that $F(a, b')$ or $F(a', b)$ is a *superior* fold edge to $F(a, b)$. If $F(a, b)$ has *no* superior fold edge, it is the *head of a fold-family*, and is drawn in red.

Fold edges can be partitioned into *families*, each consisting of a *clique* of red "head" edges, and zero or more gold edges below it such that each of the gold edges is inferior to at least one of the red edges. A fold family is the set of fold edges between two or more hold-*trees* (or subtrees), with the red head-fold(s) appearing between the roots of these trees. In the simplest case, a fold family between *two* (sub)trees has one *head* fold between the roots of the trees or subtrees involved (drawn in red), and zero or more other family members (drawn in gold), each going *from one tree to the other*.

The graph of F edges is not hierarchical and doesn't contain redundant edges. The presence of *cliques* (i.e. subgraphs that would be complete if we considered edges to be undirected) show one kind of fold "depth". For example if we have $(F(p,q) \wedge F(q,r) \wedge F(p,r))$, this implies that *all three* nodes are overlapping at some moment.

Despite the conceptual and geometric simplicity of fold-cliques, the simplest case of a fold-clique of size two (i.e. just an edge between two nodes) seems to be *by far* the most common in the *note-texture* of the polyphonic style of Palestrina and Victoria, as well as in the texturally simpler classical style (e.g. Mozart, Beethoven). Fold-cliques of size three occur occasionally (i.e. three nodes and three edges) – Figure 3–28 shows an example from Bach, who also uses them sparingly.

In defining how we will *draw* polyphones, we have characterized two secondary properties of the edges – the "directness" of a hold, and the "superiority" of a fold. In the case of holds, we can omit the subordinate edges, but for folds we can't do so without loss of information; in both cases the subrelations are not retained on subgraphs, because they refer to *contextual* properties of the edges. These contextual properties help to visualize and reason about the shape of polyphone graphs *beyond* the logical union of individual (pairwise) relations.

Node shape is sometimes used to visualize monophone cardinality, with 1 = circle, 2 = double-circle, 3 = triangle, 4 = box, 5 = pentagon, 6 = hexagon, 7 = heptagon, 8 = octagon, and > 8 = double-octagon. Nodes may by labeled by index (arbitrary integers in $<$ order), or by other features of the structures represented by the node.

Figures 3–1–3–3 show polyphone graph components. Figure 3–1 shows components consisting each of a single hold-tree; Figure 3–2 shows components with holds and simple (head-only) fold families; Figure 3–3 shows polyphones with red and gold fold edges.

3.2.1 Drawing Polyphones on Scores

One way to draw polyphones on scores is to draw colored hold and fold edges from note to note. With moderately complex scores, these become hard to read because edges may not be oriented in the vertical dimension if voice identity and/or pitch is being represented in this dimension. We can disambiguate these orientations with color, but this fails to organize them visually in a straightforward way. Monophones are not necessarily spatially grouped, complicating the visual picture. An example is shown in Figure 3–4.

In Figures 3–5, 3–7, 3–8, and 3–9, the scores are reorganized by redistributing the notes onto staves to clarify the structure of the polyphones. The colors of the notes represent their voice identity in the original score (information previously conveyed by which staff they were on). Monophones are grouped together on the same staff, and hold ancestors are located on a spatially higher staff than their descendants. On the abstract graphs, red head-fold edges are typically depicted as horizontal, but to avoid overlaps on the same staff, these form zig-zags on the permuted score (compare Figures 3–5 and 3–6).

3.3 Logical Structure of Polyphones

We observe some lawful regularities in the structure of polyphones.

If $F(p, q)$ is a head fold, then its family includes any folds $F(p^d, q^d)$ where p^d is p or an H descendant of p, and q^d is q or an H descendant of q. This is not a free-for-all between descendants of p and q, but follows a few rules which significantly restrict

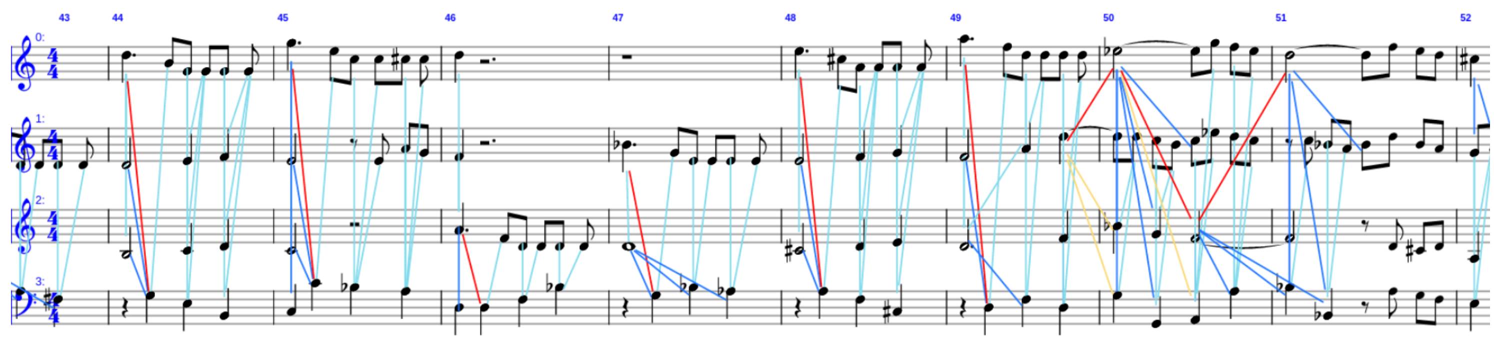

Figure 3–4: Mozart, String Quartet K.168, Mvt. I (excerpt). Polyphone edges are shown: dark blue lines are hold-relations oriented (spatially) downward, light blue lines are hold-relations oriented upward. Red and gold lines are fold edges (heads and non-heads of fold-families respectively). Score is unpermuted – compare with subsequent figures of scores permuted to clarify graph structure.

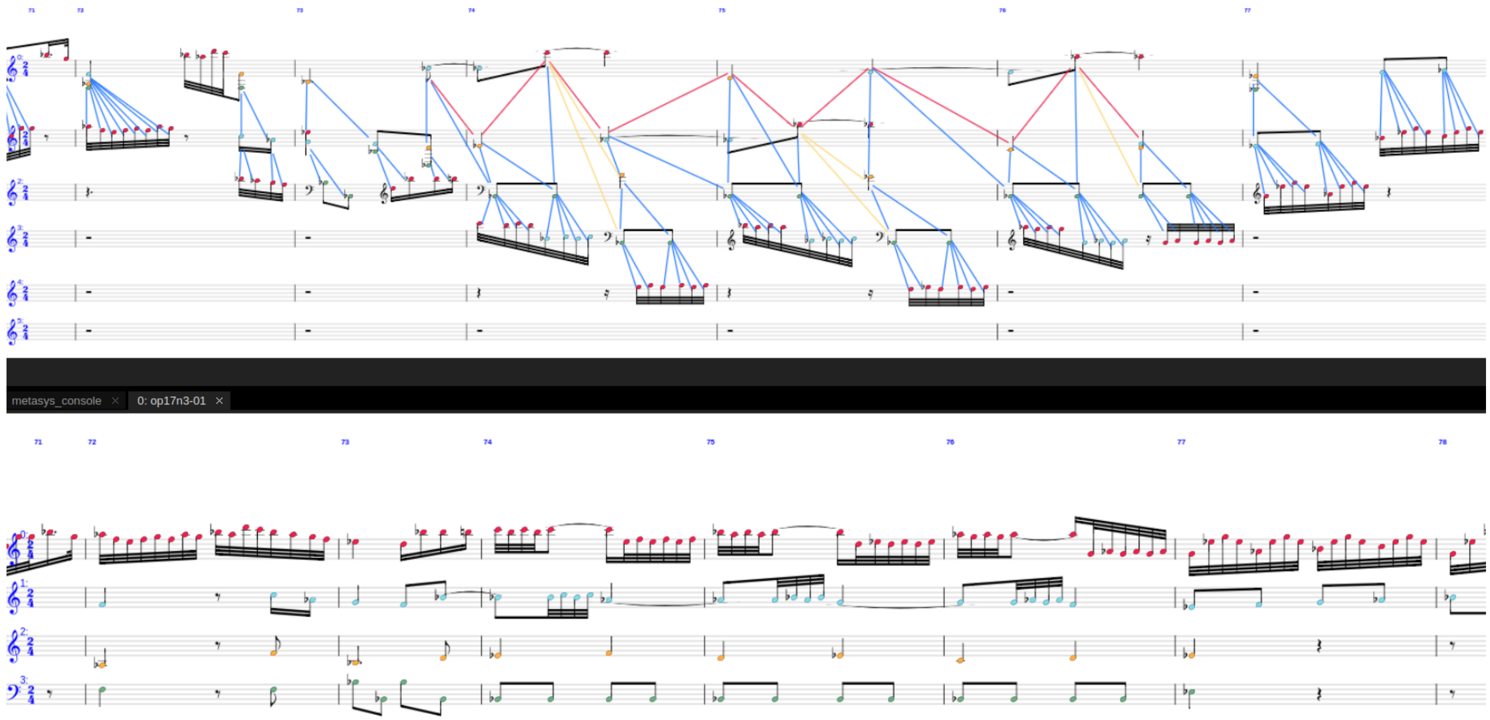

Figure 3–5: Polyphones on an excerpt from Haydn String Quartet Op.17 No.3 Mvt. 1. The bottom staff system is the score colored by voice. The top system shows the notes of the score *permuted* to show graph structure. Notes have the *same color* as in the unpermuted score below, and are arranged on the staves by polyphone hold-depth (and non-collision), with monophones grouped together on a staff. Figure 3–6 shows the abstracted version of the polyphone with the long chain of folds.

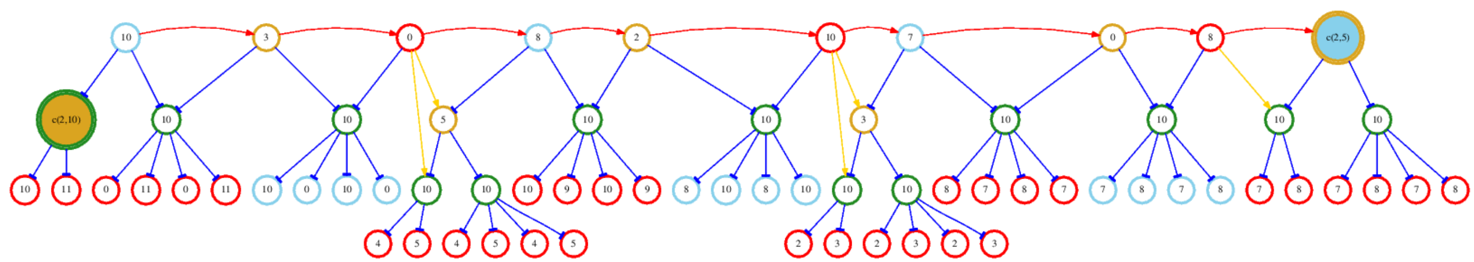

Figure 3–6: Polyphone component with red fold-path, from the Haydn excerpt in Figure 3–5. Node colors are voices (corresponding to note colors in Figure 3–5).

Figure 3–7: Polyphones on an excerpt from Mozart String Quartet K.168, Mvt. 1. Measures 78 and 79 show polyphone components with the same shape but *permuted* voice identities (colors). Compare measures 80 and 81 – the graph from measure 80 is slightly altered and expanded in mm. 81–82.

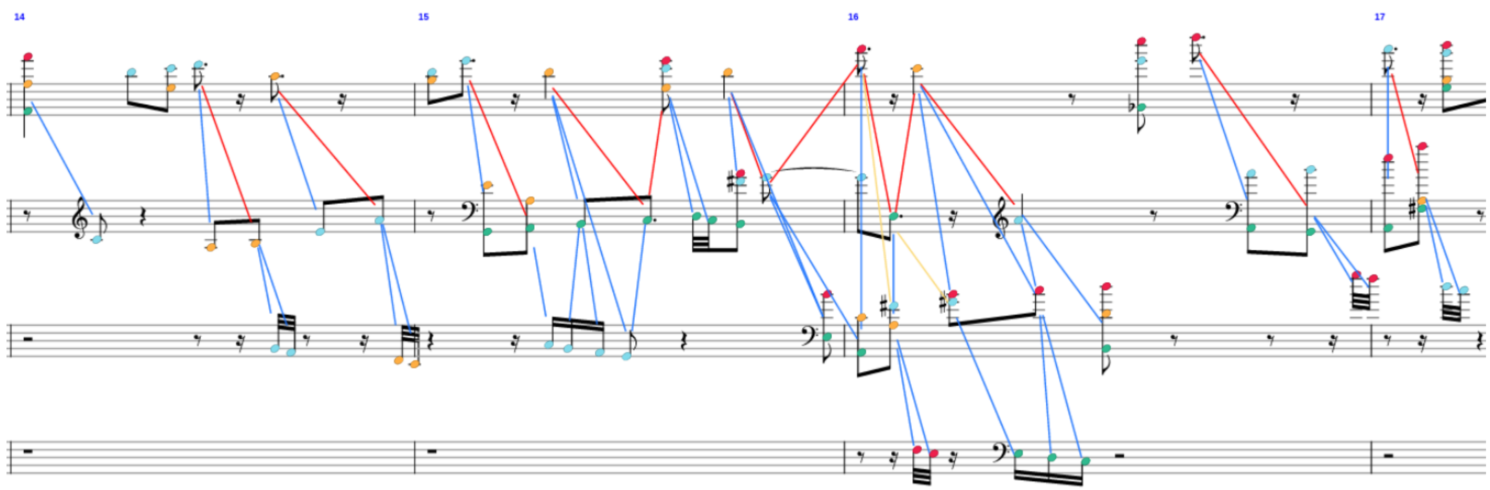

Figure 3–8: Polyphones on an excerpt from Bach, Well-Tempered Clavier Book I, Fugue 1, mm. 14–17.

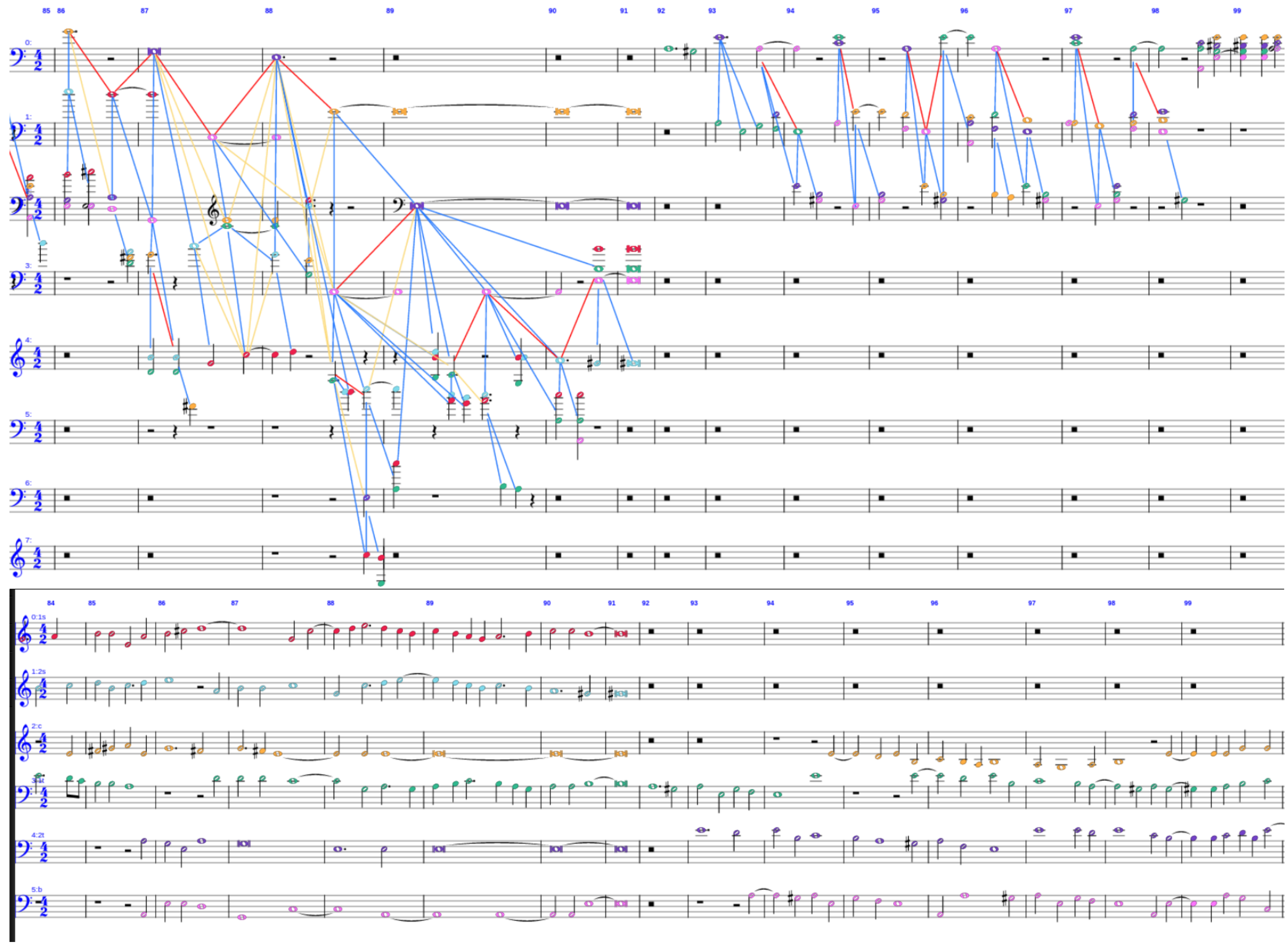

Figure 3–9: Victoria: Vadam et Circuibo Civitatem (excerpt mm. 86–100). Permuted score with polyphone edges; unpermuted score below. Polyphonic complexity, simplicity, and variation.

the configuration of the family. The following four rules describe logical restrictions on the edges participating in the family.

1. (Shared tree rule) If r is an H descendant of both p and q then r participates in a fold with neither p nor q. This is evident from the definitions of H and F. We mention it as a rule because in visualizing or constructing fold families within polyphone dags, it helps to remember that these nodes r can exist as a "middle" tree shared between p and q which doesn't participate in the fold family under head $F(p,q)$. This is illustrated in Figure 3–10.

$$H(p,r) \wedge H(q,r) \rightarrow \neg F(p,r) \wedge \neg F(r,q)$$

Proof: follows from $\neg(H(a,b) \wedge F(a,b))$. □

2. (Path rule) If p^d folds with q^d, then for any H ancestor p^a of p^d that is not an H ancestor of q^d, p^a folds with q^d. The symmetric case: if p^d folds with q^d, then for any H ancestor q^a of q^d that is not an H ancestor of p^d, p^d folds with q^a. This rule describes the behavior of folding H *paths*, such that descendants p^d of p fold with q or its descendant q^d only if the entire *path* from p to p^d folds with q or q^d (and vice versa for the symmetric case). Illustration in Figure 3–11.

$$F(p^d,q^d) \wedge H(p^a,p^d) \wedge \neg H(p^a,q^d) \rightarrow F(p^a,q^d)$$

Proof: We prove the equivalent statement

$$F(p^d,q^d) \wedge H(p^a,p^d) \rightarrow H(p^a,q^d) \vee F(p^a,q^d).$$

$$\cfrac{\cfrac{\cfrac{F(p^d,q^d)}{p_i^d < q_i^d < p_o^d < q_o^d} \qquad \cfrac{H(p^a,p^d)}{p_i^a \leq p_i^d < p_o^d \leq p_o^a}}{p_i^a < q_i^d < p_o^a}}{H(p^a,q^d) \vee F(p^a,q^d)} \qquad \square$$

3. (Inner tree rule) The formally-stated inner tree rule is slightly obscure, but the intuitive meaning of the rule is this: as a rule, the *innermost* nodes of the H trees are involved in the fold family (i.e. the rightmost nodes of the tree on the left and the leftmost nodes of the tree on the right) – in particular, innermost nodes to the *outside* of any shared tree are involved in the fold family. The formal statement of the inner tree rule characterizes the exceptional case where a non-innermost node is involved:

Suppose p^d and p^c are siblings with an H parent p, with $p^c < p^d$, and p folds with q. If p^c folds with q and p^d is not an H descendant of q, then p^c folds with p^d and p^d folds with q. (And similarly for the symmetric case.) Note that if p^c folds with p^d, this fold is the head of a separate fold family (since neither is a descendant of q). Figure 3–12 illustrates.

$H(p,p^c)\wedge H(p,p^d)\wedge(p^c<p^d)\wedge\neg H(p^c,p^d)\wedge F(p,q)\wedge F(p^c,q)\wedge\neg H(q,p^d)\rightarrow F(p^c,p^d)\wedge F(p^d,q)$

Proof:

We want to show $F(p^c,p^d)\wedge F(p^d,q)$, so we show
$(p^c_i<p^d_i)\wedge(p^c_o<p^d_o)\wedge(p^d_i<p^c_o)\wedge(p^d_i<q_i)\wedge(p^d_o<q_o)\wedge(q_i<p^d_o)$.

From $(p^c<p^d)\wedge\neg H(p^c,p^d)$ we get $(p^c_i<p^d_i)$ and $(p^c_o<p^d_o)$. $\surd\surd$

From $F(p,q)\wedge H(p,p^d)\wedge\neg H(q,p^d)$ we get $(p^d_i<q_i)$ and $(p^d_o<q_o)$. $\surd\surd$

From $F(p^c,q)$ we get $(q_i<p^c_o)$, and since we have $(p^c_o<p^d_o)$ and $(p^d_i<q_i)$,

we get $(p_i^d < p_o^c)$ and $(q_i < p_o^d)$. $\surd\surd$

$\square$

The proof of the inner tree rule doesn't require $H(p, p^c)$, and in fact the rule holds when $H(p, p^c)$ is replaced by $F(p^c, p)$ (i.e. removing $H(p, p^c)$ would necessitate adding $F(p^c, p)$ by the path rule). However, since these rules are meant to be descriptive of multiple F edges between two H trees, and with this replacement we would lose that structure, we keep the $H(p, p^c)$ assumption.

4. (Clique middle rules) Suppose $F(p, q)$ and $F(q, r)$ and $F(p, r)$ with $p < q < r$:

a) If p^d is an H descendant of p and also an H descendant of r, then p^d is an H descendant of q.

b) If p^d is an H descendant of p and $F(p^d, r)$, then either $F(p^d, q)$ or $H(q, p^d)$. And similarly the symmetric case.

c) If for some fourth node s, if $F(p, s)$ and $F(s, r)$, then either $H(q, s)$ or $H(s, q)$ or $F(q, s)$ or $F(s, q)$ (– in the latter two cases, we obtain a four-clique). Shown in Figure 3–13.

a) $F(p, q) \wedge F(q, r) \wedge F(p, r) \wedge (p < q < r) \wedge H(p, p^d) \wedge H(r, p^d) \rightarrow H(q, p^d)$

b) $F(p, q) \wedge F(q, r) \wedge F(p, r) \wedge (p < q < r) \wedge H(p, p^d) \wedge F(p^d, r) \rightarrow F(p^d, q) \vee H(q, p^d)$

c) $F(p, q) \wedge F(q, r) \wedge F(p, r) \wedge (p < q < r) \wedge F(p, s) \wedge F(s, r) \rightarrow H(q, s) \vee H(s, q) \vee F(q, s) \vee F(s, q)$

Proof:

a) We have $(p_i < q_i < r_i \leq p_i^d < p_o^d \leq p_o < q_o < r_o)$, therefore $H(r, p^d)$.

b) We have $(p_i < q_i < r_i < p_o^d)$, with $(p_i < p_i^d)$. Therefore, we get either $(p_i^d < q_i)$, in which case $F(p^d, q)$, or $(q \leq p_i^d)$, in which case $H(q, p^d)$.

c) We have $(p_i < q_i < r_i)$, and $(p_i < s_i < r_i)$ giving two cases $(q_i < s_i)$ and $(s_i < q_i)$. Likewise, we have $(p_o < q_o < r_o)$, and $(p_o < s_o < r_o)$, giving two cases $(q_o < s_o)$ and $(s_o < q_o)$. The product of these 2 x 2 cases gives four cases:

i) $(q_i < s_i) \wedge (q_o < s_o) \rightarrow ((s_i < q_o) \rightarrow F(q, s))$

ii) $(q_i < s_i) \wedge (s_o < q_o) \rightarrow H(q, s)$

iii) $(s_i < q_i) \wedge (q_o < s_o) \rightarrow H(s, q)$

iv) $(s_i < q_i) \wedge (s_o < q_o) \rightarrow ((q_i < s_o) \rightarrow F(s, q))$

From $F(p, q) \wedge F(p, s)$ we get $(p_i < q_i < p_o < q_o) \wedge (p_i < s_i < p_o < s_o)$; therefore $(s_i < q_o) \wedge (q_i < s_o)$.

□

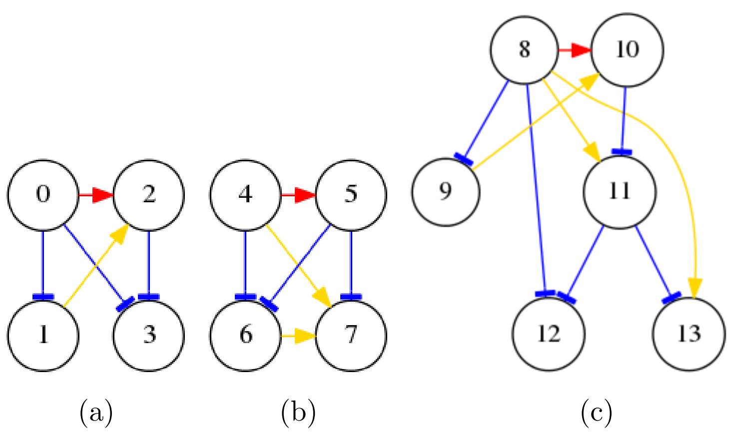

Figure 3–10: Shared "middle" trees. In (a), node 3 (and its descendants, if any) is a shared tree under F(0,2). In (b), node 6 is a shared tree under F(4,5). In (c), 12 is a shared tree under F(8,10).

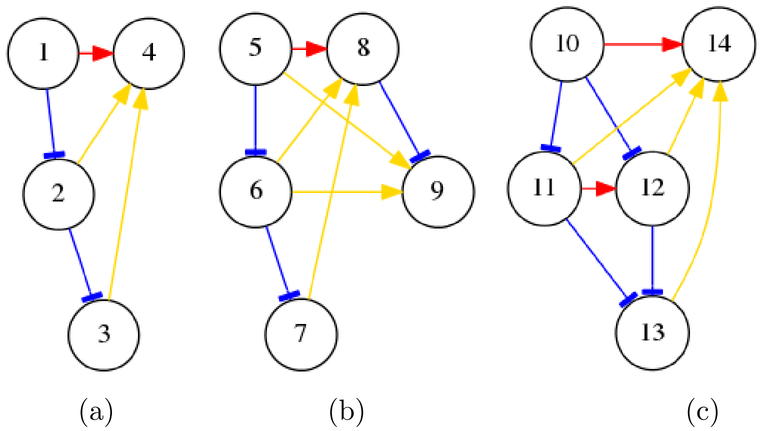

Figure 3–11: Path rule. In (a) F(3,4) requires F(2,4) and F(1,4). In (b) F(6,9) requires F(6,8), F(5,9), and F(5,8). In (c) F(13,14) requires F(11,14), F(12,14), and F(10,14). F(11,12) is not part of the fold family under F(10,14), since it only concerns members of the H tree under 10, and not the H tree under 14.

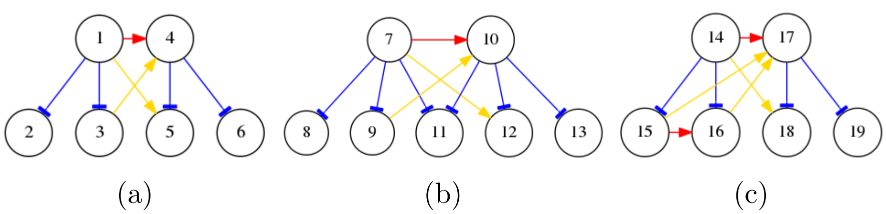

Figure 3–12: Inner tree rule. (a) As a rule, the *innermost* nodes of the H trees are involved in the fold family. (b) More precisely, innermost nodes to the *outside* of any shared tree are involved in the fold family. (c) The formal statement of the inner tree rule characterizes the exceptional case where a non-innermost node is involved – here F(15,17). The rule requires F(15,16) and F(16,17) in this case. This exceptional case involves the formation of an *F clique* of size 3 (but the clique edges are not all head-folds).

Because of these logical restrictions, we see fold configurations in which paths down from p and q participate, where descendants fold with a subset of the nodes folded with by their ancestors (path rule). These paths are to the *outside* of any

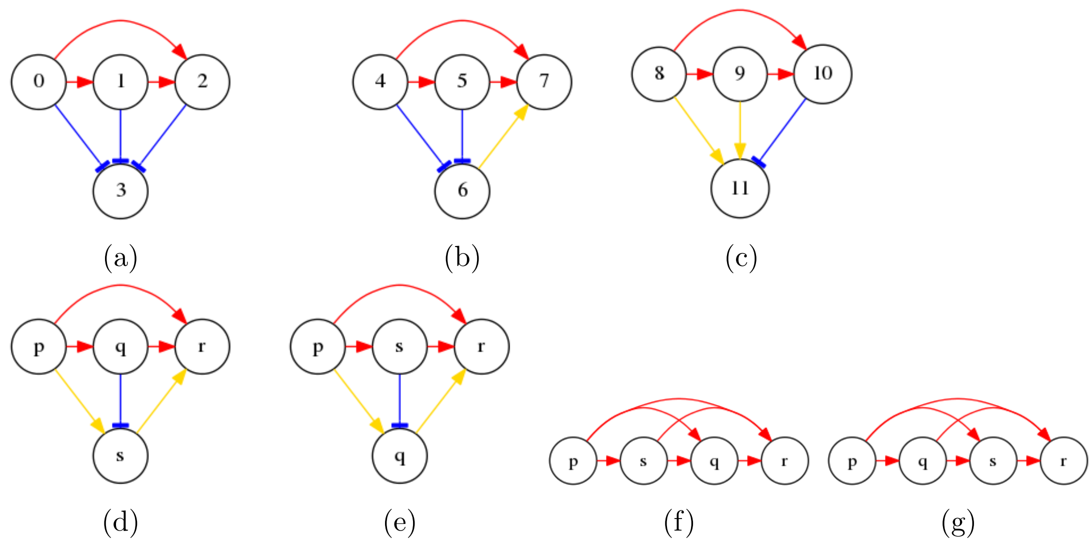

Figure 3–13: Clique middle rules. In (a), H(0,3) and H(2,3) requires H(1,3). In (b) and (c) the H edge and the F edge on either side require that the middle node of the fold clique has either an H or an F edge to the subordinate node (clique middle rule b). (d) through (g) show the four cases of clique middle rule c, when F(p,s) and F(s,r).

shared descendants of p and q (shared tree rule), but to the *inside* of any non-shared descendants of p and q, and any paths not *strictly* to the inside are folding with their inner siblings (inner tree rule).

3.4 Related Literature

3.4.1 Interval Temporal Logic

James Allen's *interval algebra* [Allen1983] describes 13 temporal relations, shown in Figure 3–14. These include *equality* (monophone relation), and six pairs of opposite relations, (i.e. they could be thought of as six *oriented* relations).

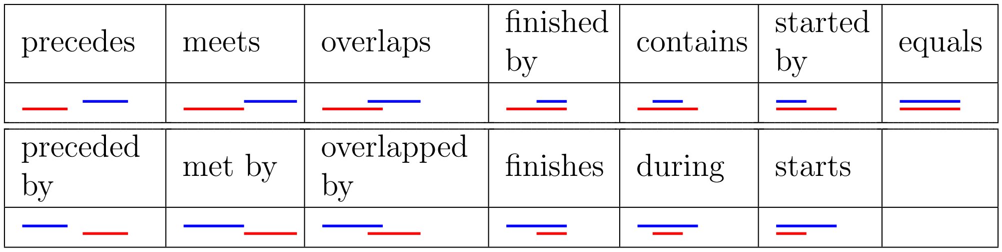

Figure 3–14: Allen's 13 basic relations for temporal intervals.

Of these six, "precedes" and "meets" do not correspond to polyphone relations, "overlaps" is equivalent to *fold*, and "finished by," "contains" and "started by" correspond to *hold* – with a shared starting point, no shared interval-boundary, or a shared end-point.

This set of relations is higher-informational than the smaller set we have used for polyphones. For some purposes, it may be necessary to *augment* polyphone hold-edges with their shared boundaries. For instance, if we want to use a polyphone graph to generate a corresponding set of (normalized) intervals, the options are either to use shared-boundary information, or to adopt an arbitrary rule (e.g. placing all hold-children strictly *inside* their parents). Despite this, we find it more convenient to treat this additional information as an augmentation or sub-type, rather than introduce more types of edges – one reason is that the subtypes of holds always occur at predictable locations (i.e. rightmost and leftmost hold-children), so if we are walking the graph, we always know *when* to check whether an endpoint is shared.

Allen's interval algebra is used for *interval temporal logic* – to *reason* about sets of intervals from partial knowledge. Given a set of intervals, the relation between each may be known, unknown, or *partially* known as a disjunction of several of the basic relations (e.g. "precedes *or* meets *or* overlaps"). Then deductions are made using logical operations (e.g. complements, converses, transitive inferences, intersections, etc.), in order to propagate the constraints of the system and deduce more information about the relations. A logical system can be represented as a *graph* of constraints, but the particular shapes that these graphs can take, and how these might correspond to the structure of a set of intervals, are not explored in the literature.

The *satisfiability* of an interval temporal logic problem (i.e. whether it has a completion without contradictions) is NP complete. A number of tractable sub-algebras have been identified, in which constraints on the form of the problem guarantee its efficient solvability [Krokhin+2003].

Interval temporal logic is applicable whenever partial knowledge is available about the temporal relations of events. A score, however, gives a *complete* account of temporal relations among events. Polyphones, in contrast with interval temporal logic, are not concerned with making logical deductions, but with describing the *shapes* of graphs and subgraphs, and how subgraphs of different shapes can interrelate to form larger shapes or patterns.

3.4.2 Interval graphs

An *interval graph* is a graph for which each node can be represented by a temporal interval, and there is an edge between two nodes if and only if there is some moment of time at which both intervals are active. Interval graphs are like polyphones in that they compare a graph structure to a corresponding set of temporal intervals. However, the edges of an interval graph are untyped and undirected, and there is no *monophone* concept. An interval graph is therefore more abstract (information lossy) in comparison with a corresponding interval set.

Interval graphs are well-studied, and there are many known characterizations: for example, *any* graph that is *chordal* (all cycles are triangulated), and contains no "asteroidal triple" is an interval graph [Golumbic1980]. An asteroidal triple consists of three nodes such that between any two of them there exists a path that doesn't pass through any node adjacent to the third – the asteroidal triple and the cycle of size > 3 represent minimal subgraphs that foil the linear dimension.

The linear characteristic of (connected) interval graphs also means that they can be represented as a *path of cliques* [McKee+1999]. A clique-path through an interval graph specifies a polyphone interpretation – each clique corresponds to the intervals active at some moment in time.

[Fishburn1985] characterizes the *family* of interval sets that correspond to a single interval graph – these are related by re-ordering. The family of interval orderings shows how a given interval graph could be temporally specified in a combinatorial number of ways (– an interval ordering, however, is as specific as a polyphone, since an intersecting ordered pair could correspond to either a hold or a fold).

In general, the graph-theoretic concerns related to interval graphs seem to *start* from a graph, and then make deductions about properties of either the families of corresponding interval sets, or (more often), the graph itself. This is in contrast to the concerns of polyphones, where we start with a set of intervals, and use graph structure to visualize, organize, think, and compute about them.

Interval graphs have been used in application to scheduling and resource assignment, as well as ordering problems in genetics and archaeology. Generally speaking, these applications use the interval graph to represent dependence constraints, and then use the quasi-linear properties of the graph to generate feasible orderings. One way that interval graphs have been used *descriptively* is in describing the structure of food webs (i.e. overlapping ecological niches in which living beings eat one another) [Cohen1977]. Interval graphs are used to characterize the shape of these interactions – but ecological niches don't have a natural temporal or physical *ordering*, so this kind of information isn't lost in the graph representation. The interval graph description is used to help characterize an abstract dimensionality for niche interactions.

3.5 Observations with Polyphones

In this section, we show how polyphones can be used to make various kinds of observations about the pattern and shape of texture on scores. Each observation takes a low-dimensional projection of polyphonic structure and uses it to make a sketch on the score.[2]

This *projection* phase is a complement to the structure-building process, since it allows us to "see" simple facets of a structural picture which is too complex to visualize at once, and also facilitates the recursion of structure-building *from* a low dimensionality.

[2] We give brief suggestions of what is to be observed in these sketches, but these images are meant to afford a *visual* description, and detailed *prose* descriptions of the visual shapes and patterns are not given.

A basic method for making projections of polyphones is to *compare* different polyphone subgraphs, by inventing summary functions that afford equivalence or measurement relations. We show a few such functions – it's easy to see how to construct more, relating and distinguishing parts of the score from different points of view.

3.5.1 Textural Sketches

In the first set of illustrations (Figures 3–15 to 3–17), the *shape* of each polyphone is taken as the equivalence relation, with *monophone* cardinality abstracted away such that monophones of any cardinality are equivalent. Since the graphs are dags (directed and acyclic), this kind of isomorphism is straightforward to detect. Each different polyphone shape is represented by a different color on a zoomed-out view of the score. Figures 3–15 to 3–17 show three piano sonata movements by Beethoven, from this point of view. The illustrations show how this low-resolution sketch of polyphonic texture (showing only *difference*), gives a large-scale viewpoint on musical pattern and shape. We can see sections or zones of colors, outlining an overall "form" for the movements: contrast, repetition and variation, pattern processes, uniformity and variety.

Figures 3–18 and 3–19 show the same kind of picture on two-part inventions by J.S. Bach, along with a further equivalence distinction made between the permutation of *voices* with respect to graph-shape – in these figures, the two equivalence relations are related by inclusion, giving a *simple* dimensionalization of difference.

Figures 3–20 to 3–22 again show polyphone component isomorphisms, along with a different kind of isomorphism on a different kind of polyphone subgraph. This time we take *hold-trees* as subgraphs to be compared (– these can overlap). Rather than taking simple isomorphism, we invent a looser relation by taking a reduced *sequence* of hold-heights of the (direct) children of each tree-root.

The *hold-height* of a node is calculated as the length of the *longest* path from a leaf *up* to the node. The sequence of hold-heights of the root's children is *reduced* by removing repetitions, so that if the sequence is (2 2 1 1 1) the comparison feature is (2 1) (– and that tree is therefore isomorphic to another tree with sequence (2 1 1) under this relation).

In the figures, the outer boxes show the color and number for the identity of the top-level tree; the same colors are also used to color the roots of inner *subtrees*. For example, in Figure 3–20, the red boxes have height-sequence (1), while the blue boxes have height-sequence (2 1) – therefore the blue boxes contain a red note at their beginning, which is the root of a subtree with height-sequence (1).

This hold-height-sequence isomorphism is a loose summary of tree-shape, tending to group together larger numbers of structures, showing a different level of detail. The images also show how trees overlap to constitute polyphone components with folds.

Figures 3–20, 3–21, and 3–22 also show differential treatment of polyphonic isomorphism and patterning, corresponding roughly to a stylistic difference in musical texture. Although Figures 3–20 and 3–21 are fugues from the same collection, Figure 3–20 (Bach, WTC Book I, Fugue 2) shows more isomorphism and tighter pattern structure at the level of relatively simple polyphone components, while the more flexible polyphony of Figure 3–21 (Bach, WTC Book I, Fugue 12) contains many polyphone components that are not comparable under the tight isomorphism, but yields more to comparison and patterning under the looser isomorphism on sub-components. Figure 3–22 (T. L. Victoria: "Date ei de fructu"), in an earlier polyphonic style with an alternate set of procedures and values on pattern and difference, also affords more comparison under the looser isomorphism, but without revealing obvious patterning.

Figures 3–23, 3–24, and 3–25 provide a visualization of *fold* structure on scores. Notes involved with folds are colored per voice, and fold lines are drawn in red and gold. The shapes thus produced show processes of repetition, variation, and difference, tight and loose organization, chaining, clustering and sparseness, internal counterpoints, and other details of shape made visible.

3.5.2 A Measure of Polyphonic "Complexity"

Just as there are many ways of inventing equivalence relations for polyphones, there are also many ways of *measuring* them in order to compare aspects of their size or complexity. For example, we could measure the depths of hold-trees, the number of edges involved in a fold family, fold path, or fold clique, the cardinality of a monophone, etc. In this subsection we propose another way of reasoning about polyphonic "complexity," by generalizing the concept of the *voice*.

The concept of voice is this: imagine that each structure can be *sung* by a *voice* that can only sing one item at a time. In the most basic case, each *phone* (e.g. a note) is sung by a voice, and as with human voices, trumpets, flutes, etc., the normal situation is that one voice can sing one note at a time, but can sing any number of notes in sequence. The minimum number of voices needed to sing all of the *phones* gives a basic complexity measure.

The basic notion of the *voice* is complicated by the fact that some instruments can play more than one note at a time, and some *groups* of instruments or sounds are *timbrally* similar enough that it's interesting to consider them as one "voice" (or part) in a score. Therefore the number of "voices" from the point of view of a timbral or instrumental analysis may not correspond to the number of "phone voices" – voices that can sing one phone at a time.

From a polyphonic point of view, we can *generalize* the notion of a voice singing a *phone* at a time, so that a higher-level voice can sing a *monophone* at a time, and a yet higher-level voice can sing a *hold-tree* at a time so long as none of the nodes in the hold-tree are folding.

At the lowest level, the level of phones, we can ask *how many phone-voices* would we need in order to sing the polyphone? If phones are notes, then this is the number of human voices needed to sing the score: the maximum number of phones sounding at one time.

At the next level up, with phones grouped into monophones (simultaneous "chords"), we ask *how many monophone-voices* would we need in order to sing the polyphone; or: what is the maximum number of monophones sounding at one time? This is like having human voices that can sing a *chord* at a time, but only with a monophonic *rhythm*. If, for instance, we have four phone voices, but they are always singing together in pairs, then we will need two monophone voices – and this is true even if the phones switch their pairings. In a situation without any folds, the number of monophone voices is just the maximum height of a hold-tree. When we have folds in the graph, we can count the number of necessary monophone-voices by finding the node with the maximum number of in-edges and adding one. (This method also works for phone-voices, by summing the monophone *cardinalities* on the in-edges plus the cardinality of the node.)

At the third level up, we group the monophones into hold-trees, asking *how many hold-voices* would we need to sing the polyphone? If there are no folds in the graph, then only one hold-voice is required. In the general case, the number of hold-voices required corresponds to an optimal *coloring* of the fold-graph (i.e. such that each node has a color and two folding nodes never share a color).

Hold-voices cannot sing folds, so a fold anywhere in polyphone will break the graph into multiple trees to be sung by different voices. Here the *size* of a fold-family between any *two* hold-trees (i.e. the number of gold edges) doesn't make a difference, since the two trees can be sung by two voices, no matter how intricately they overlap.

If a fold-clique of size *three* occurs, then we need at least three hold-voices. If a *no* fold-clique of size three or greater occurs, then either two or three hold-voices are needed. The case in which *three* are needed occurs when there's an odd cycle in the fold-graph.[3]

Figures 3–26, 3–27, and 3–28 show three graphs (all from the Ricercare a 6 from the Musical Offering by Bach), which are three-colored such that each color

[3] [Olariu1991] gives a optimal, linear-time greedy algorithm to color interval graphs.

can be sung by a hold-voice. Figure 3–26, shows an odd cycle that requires a third color; Figure 3–27 shows a larger graph component that also has a few odd cycles, and 3–28 shows a 3-colored graph with 3-cliques (– in the general case we can't guarantee that a graph with 3-cliques is 3-colorable).

The number of phone-voices, monophone-voices, and hold-voices are three different ways of measuring the "complexity" of a polyphonic situation, in the sense that they're a rough measure of number of "independent" parts, for different ways of defining independence. Figures 3–29 to 3–33 show these measures of complexity or voice-independence on polyphone components of scores.

3.5.3 Hold-height and hold-depth "voices"

In Figures 3–34 – 3–37, polyphone structure is used to partition the notes of a score into textural components in two complementary ways: by taking hold-*depths* (i.e. distance from root) and hold-*heights* (i.e. distance from leaves) of each note. These each give a partition into "voices" which can contain *fold* edges but not *hold* edges.

This kind of partition offers a way of relating aspects of the texture that *aren't* related by the graph-edge relations, but are *contextually* related through their *positions* in the polyphonic structure. The "voices" are each represented by a color on the score (we can imagine this also as *instrumental* or *timbral* color). Each voice has its own pattern of occurrence and absence, projecting a rhythmic and pitch profile that cuts across the voices and registers of the score.

Figures 3–38 and 3–39 show a variant of the same idea in which each height or depth "voice" is made temporally continuous by allowing the next closest height or depth to stand in – for example, for the voice corresponding to depth 2, if no nodes of depth 2 occur in the current subtree, then nodes of depth 1 (or 0) are used. The result of this is an "orchestration" which is not a partition, but in which the voices sometimes converge to sing in unison.

3.6 Future Work

This chapter introduced polyphone graphs as a way of visualizing and reasoning about polyphonic relations on a score. There are many avenues available for future research on the formal and graph-theoretic properties of these graphs, as well as algorithmic techniques to be developed. Graph theoretic problems include coloring methods and proofs of coloring properties, as well as methods and tractability proofs for various kinds of subgraph isomorphisms (corresponding to polyphonic sub-score isomorphisms).

We briefly touched on augmenting the graphs to include end-point coincidence. In practice, it's useful to have various graph-walking algorithms to propagate information; one example is a walk that imitates a time-slice view of the score, so that each "chord" or partial chord is addressed (– this requires end-point augmentation or a simplifying assumption). Implicitly, we also associate data of *any* kind to the nodes: quantities such as pitch-height, codes such as voice identities, and so on. We have given a few sketches that integrate this data with polyphonic structure; these kinds of techniques are an open field for invention. We have suggested a few techniques for measuring and comparing polyphones and their subgraphs – there is room here for expansion and experimentation as well.

We have shown a few techniques for using polyphones to make sketches on scores in order to show properties of texture and how they project elements of large-scale form. These techniques are open to extension and innovation. We have shown how polyphones can be used to informally make distinctions between different types of texture, both between differing musical styles, and between sections of one score. These distinctions could be made clearer, by exploring how polyphones could be used to characterize different textural and stylistic procedures. Specific studies of musical works or corpora would also be of value, for example making a taxonomy or map of polyphonic types, or showing how polyphonic patterning projects formal structure.

We noticed that fold-cliques of size three or greater are rare in the complex polyphonic music we looked at, but it's not clear why this is, what other stylistic factors contribute to this, or whether they can be found in more abundance in other corpora. In this chapter, we have limited our example application to note-texture, but polyphones are applicable to *any* set of structures that project temporal intervals (– in the next chapter, we will apply them to pitch-class sets). Applications to other musical (and non-musical) structures offer themselves for study: for example, if we take a multi-voice (or multi-instrumental) score, and divide each voice into phrases, then the texture of overlap and inclusion of these utterances can be described polyphonically – and these kinds of polyphones may look very different from polyphones of note-textures.

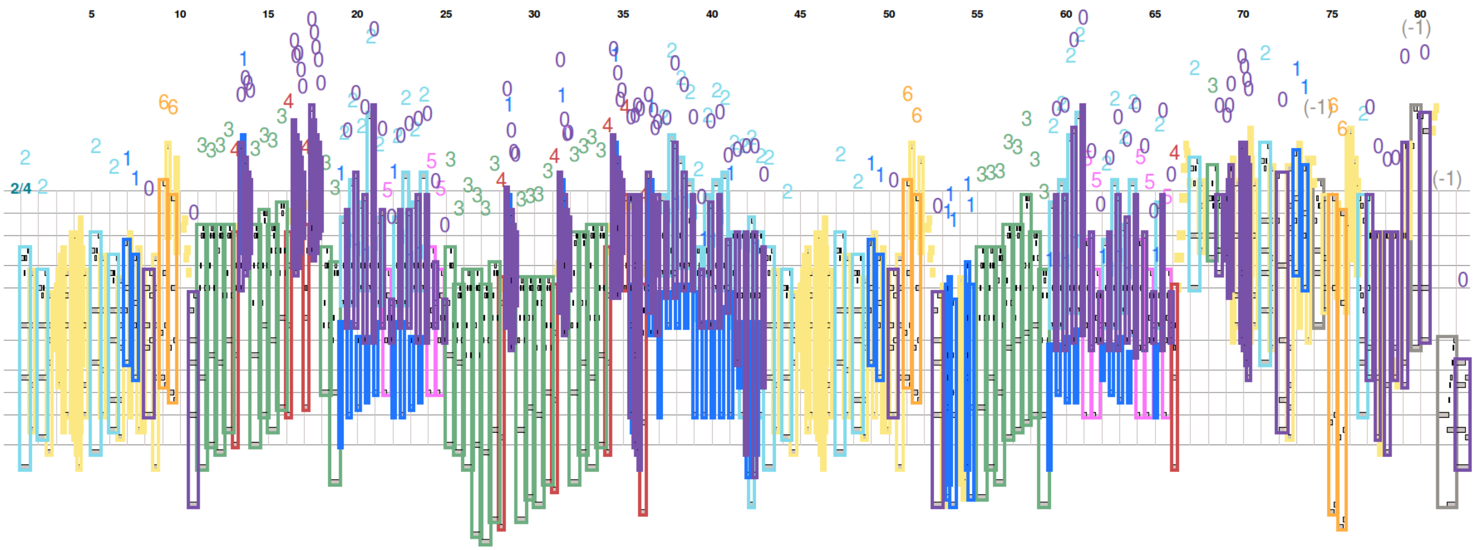

Figure 3–15: Beethoven, Piano Sonata No 3. (Op. 2) Mvt. 2 (*Adagio*). Box colors and numbers distinguish polyphone shape. Yellow boxes (no numbers) are monophone components. Grey boxes marked (-1) are unique occurrences. A sketch of textural *difference* gives an image showing large-scale shape and pattern. Zones of consistency and difference are observable, as are procedures of repetition and variation. The final section of the piece follows a looser textural procedure, without the density of repetition of the rest of the piece – but nonetheless uses previously seen material.

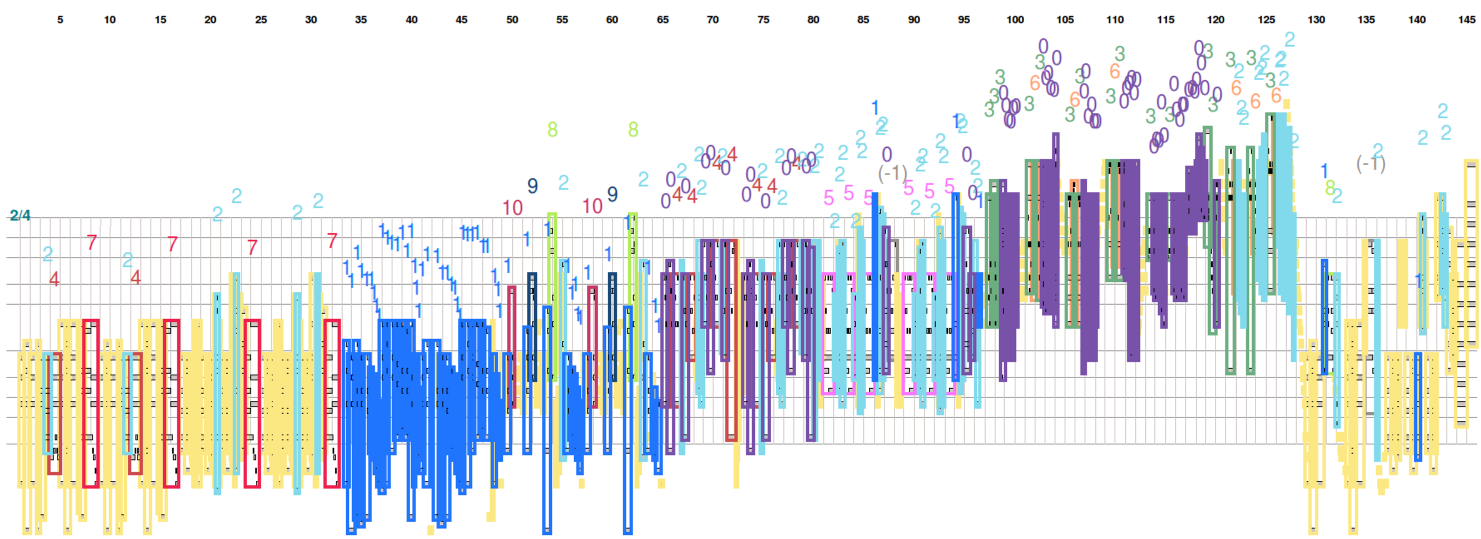

Figure 3–16: Beethoven, Piano Sonata No. 23, (Op. 57, “Appassionata”), Mvt. 2 (*Andante con moto*). Box colors and numbers distinguish polyphone shape. Yellow boxes (no numbers) are monophone components. Grey boxes marked (-1) are unique occurrences. This sketch of textural difference shows a formal procedure with zones of pattern, repetition, and difference.

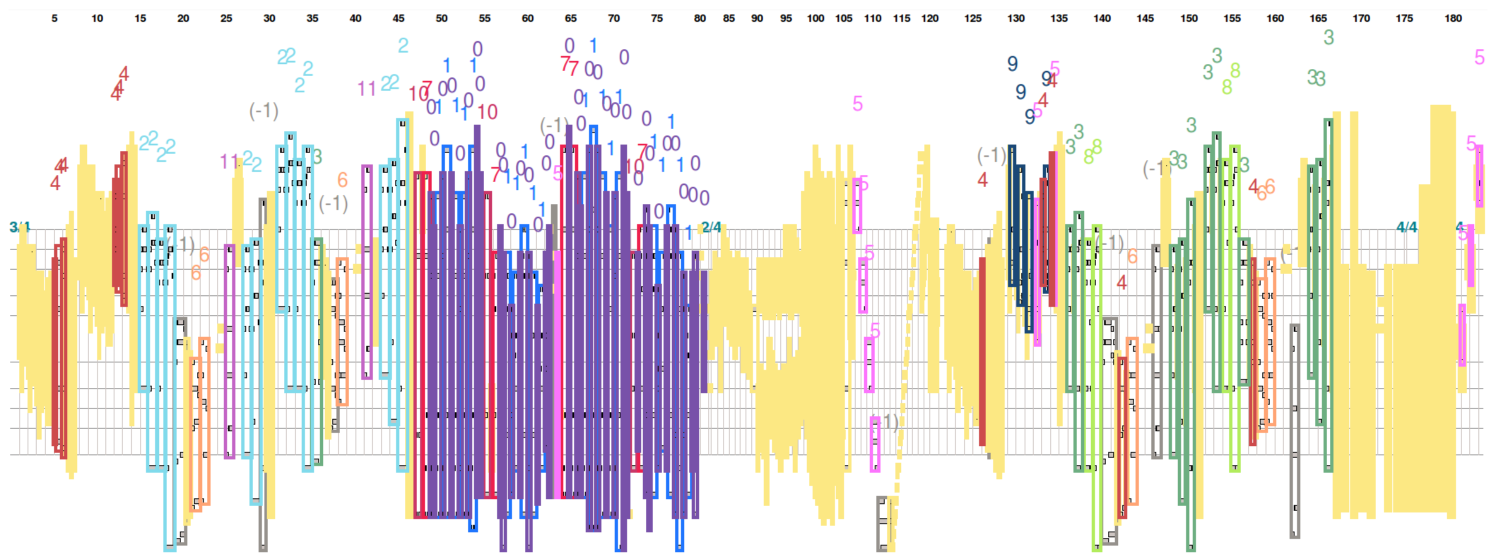

Figure 3–17: Beethoven, Piano Sonata No. 29 (Op. 106, “Hammerklavier”), Mvt. 2 (*Scherzo*). Box colors and numbers distinguish polyphone shape. Yellow boxes (no numbers) are monophone components. Grey boxes marked (-1) are unique occurrences. After the yellow diagonal sweep upward around the middle of the piece, the score is a texturally varied thematic repetition from the beginning of the piece (as can be roughly seen by the vertical placement of the boxes).

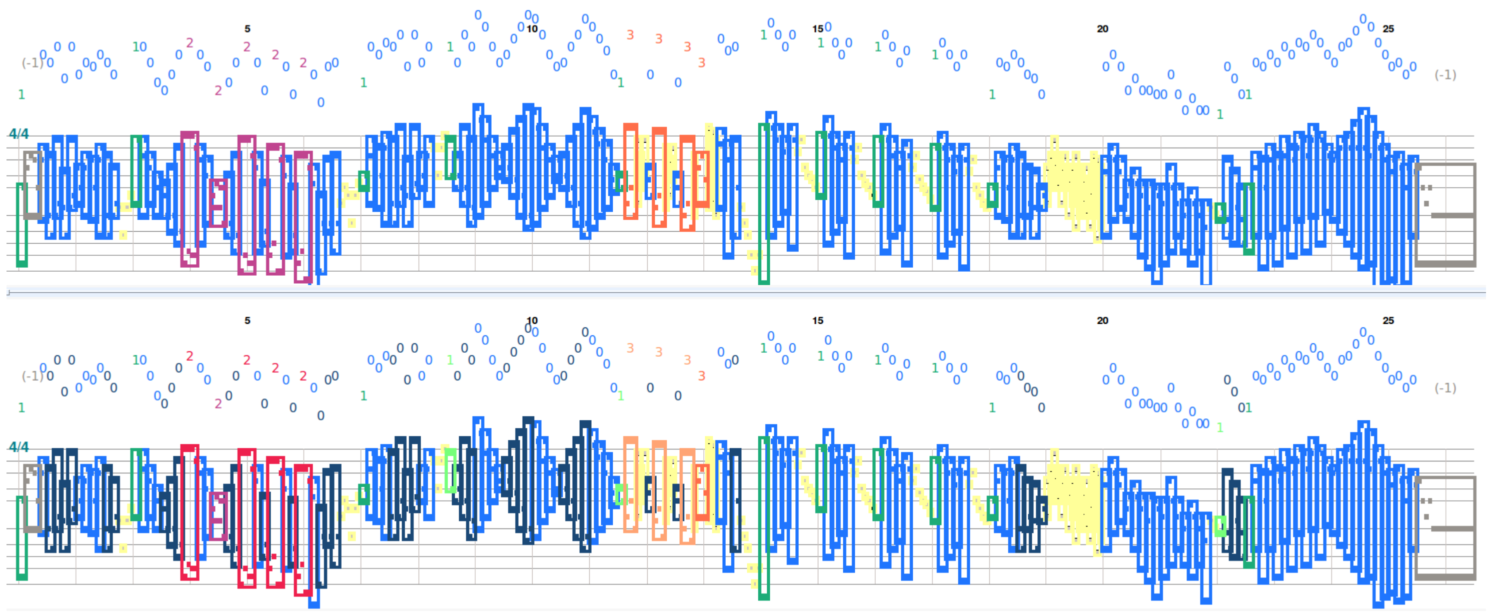

Figure 3–18: J.S. Bach, Invention 13. Top panel shows polyphone shape isomorphism as in previous figures, bottom panel distinguishes isomorphic graphs based on voice permutation – similar colors (light and dark blue, green, orange, red) show the same shape with different voice permutation.

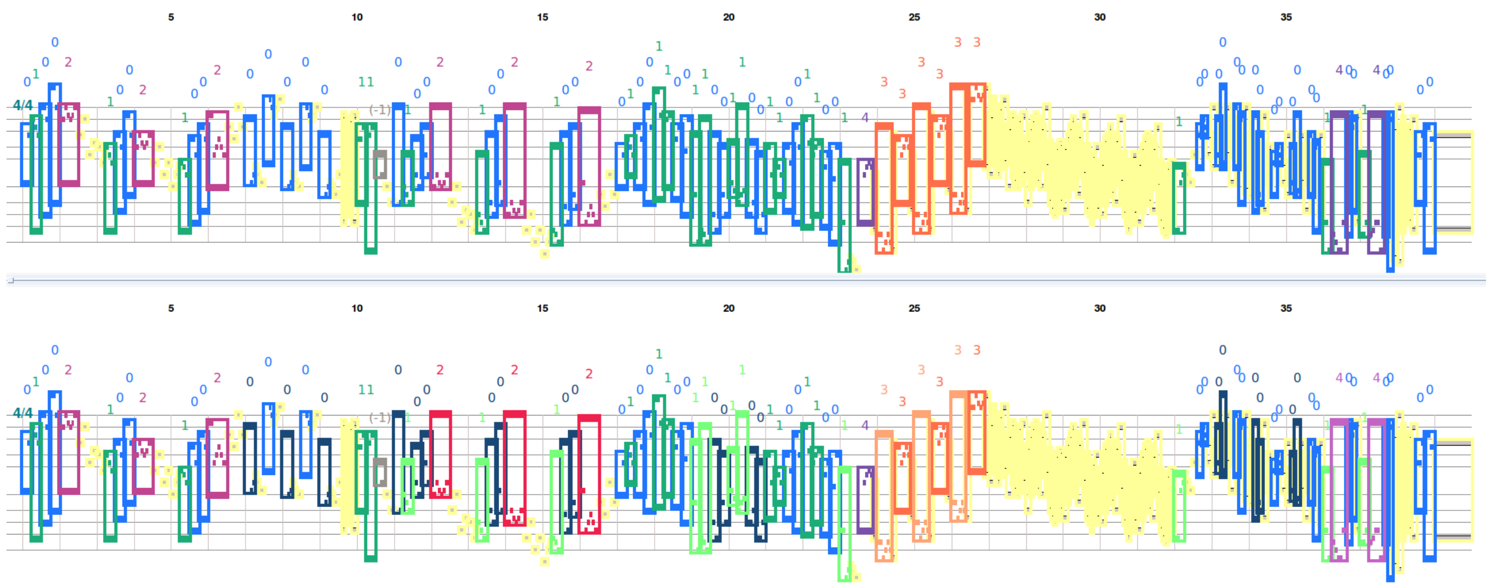

Figure 3–19: J.S. Bach, Invention 14. Top panel shows polyphone shape isomorphism as in previous figures, bottom panel distinguishes isomorphic graphs based on voice permutation – similar colors (light and dark blue, green, orange, red, purple) show the same shape with different voice permutation.

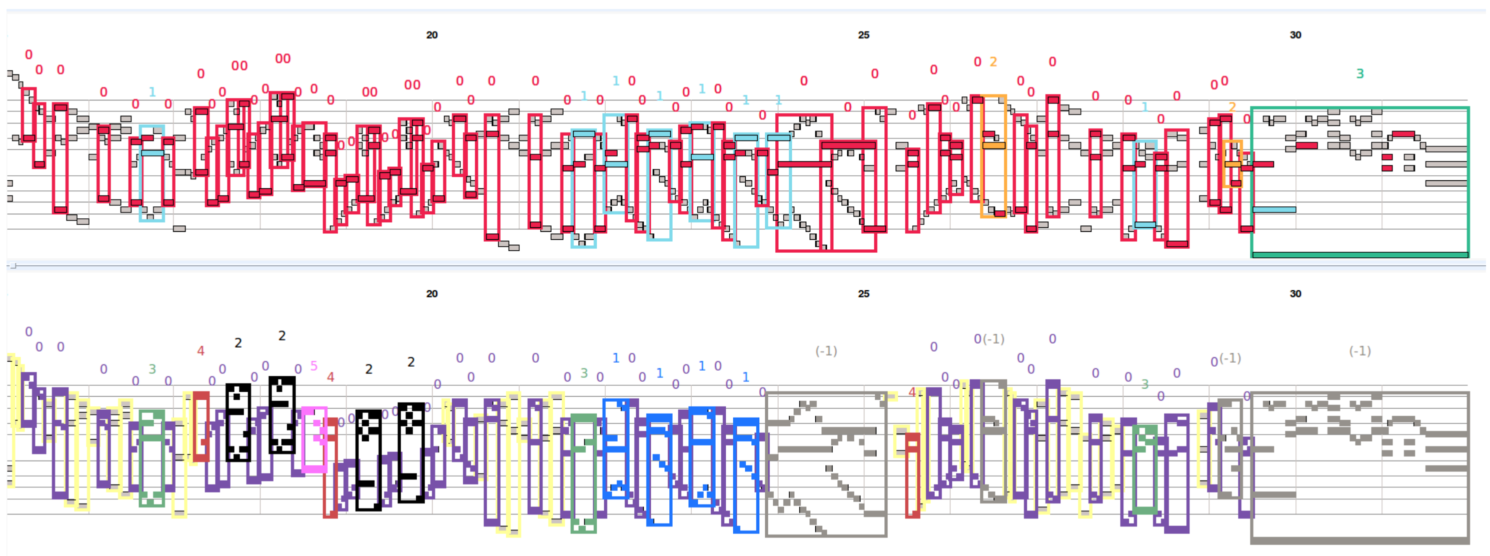

Figure 3–20: Bach: Fugue No. 2 from WTC Book I (excerpt). Bottom panel shows isomorphism on polyphone shape. Top panel shows looser isomorphism on *hold-tree* shape based on hold-height sequence of root children (described in text). Some boxes on the bottom panel are constituted of *overlapping* boxes on the top panel.

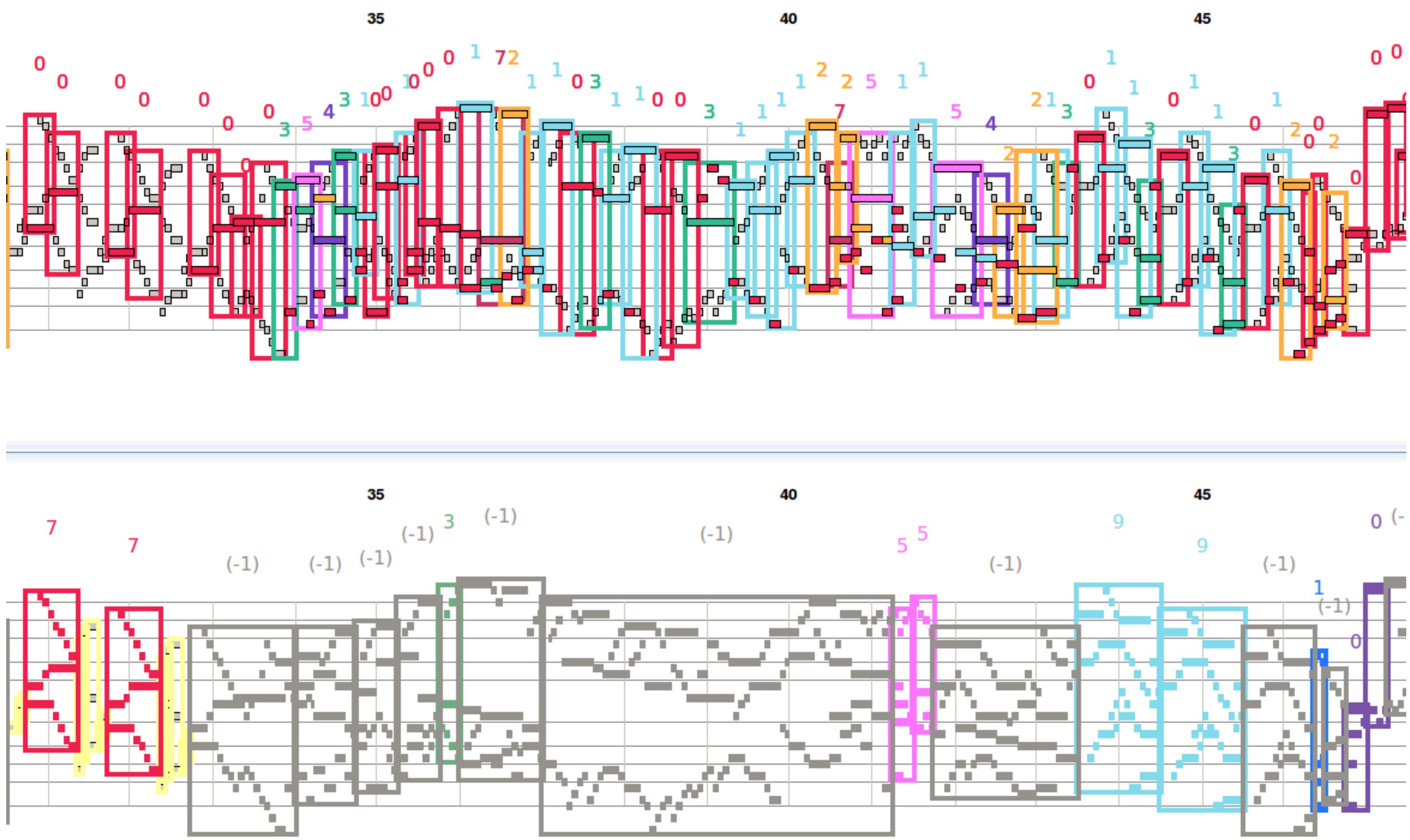

Figure 3–21: Bach: Fugue No. 12 from WTC Book I (excerpt). Bottom panel shows isomorphism on polyphone shape. Top panel shows looser isomorphism on hold-tree shape, based on hold-height sequence of root children. In this case, the top panel seems more informative (compare with Figure 3–20).

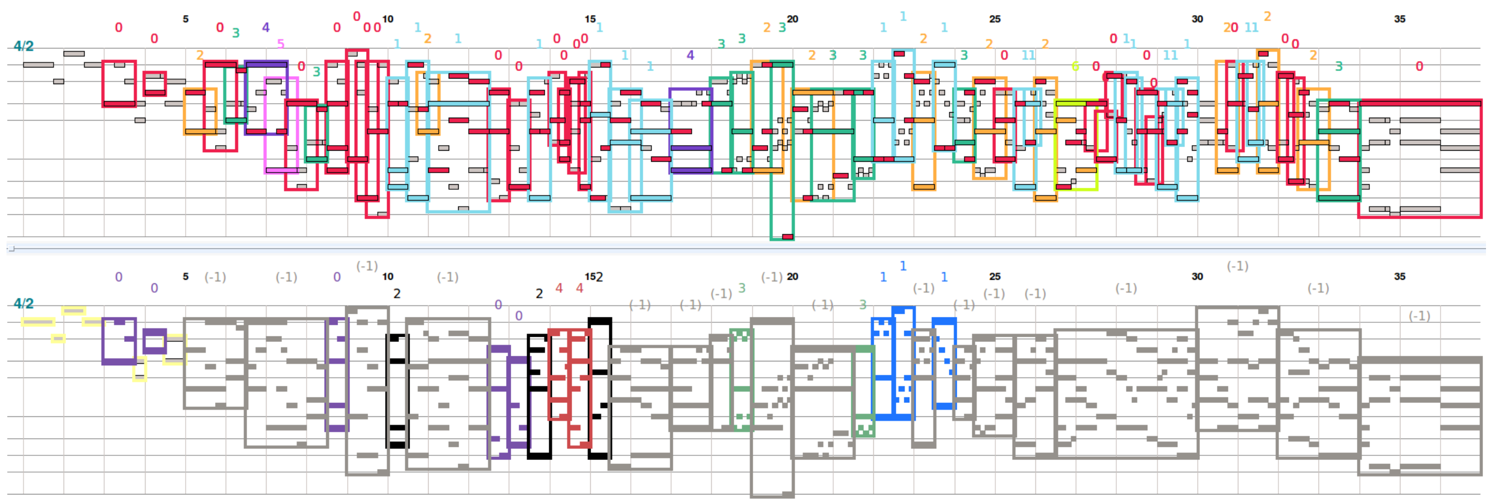

Figure 3–22: Victoria: "Date ei de fructu." Bottom panel shows isomorphism on polyphone shape. Top panel shows looser isomorphism on hold-tree shape, based on hold-height sequence of root children. The looser isomorphism affords more comparability between subgraphs; the style of polyphony in this score has an observably different valuation of "pattern" than the previously illustrated scores.

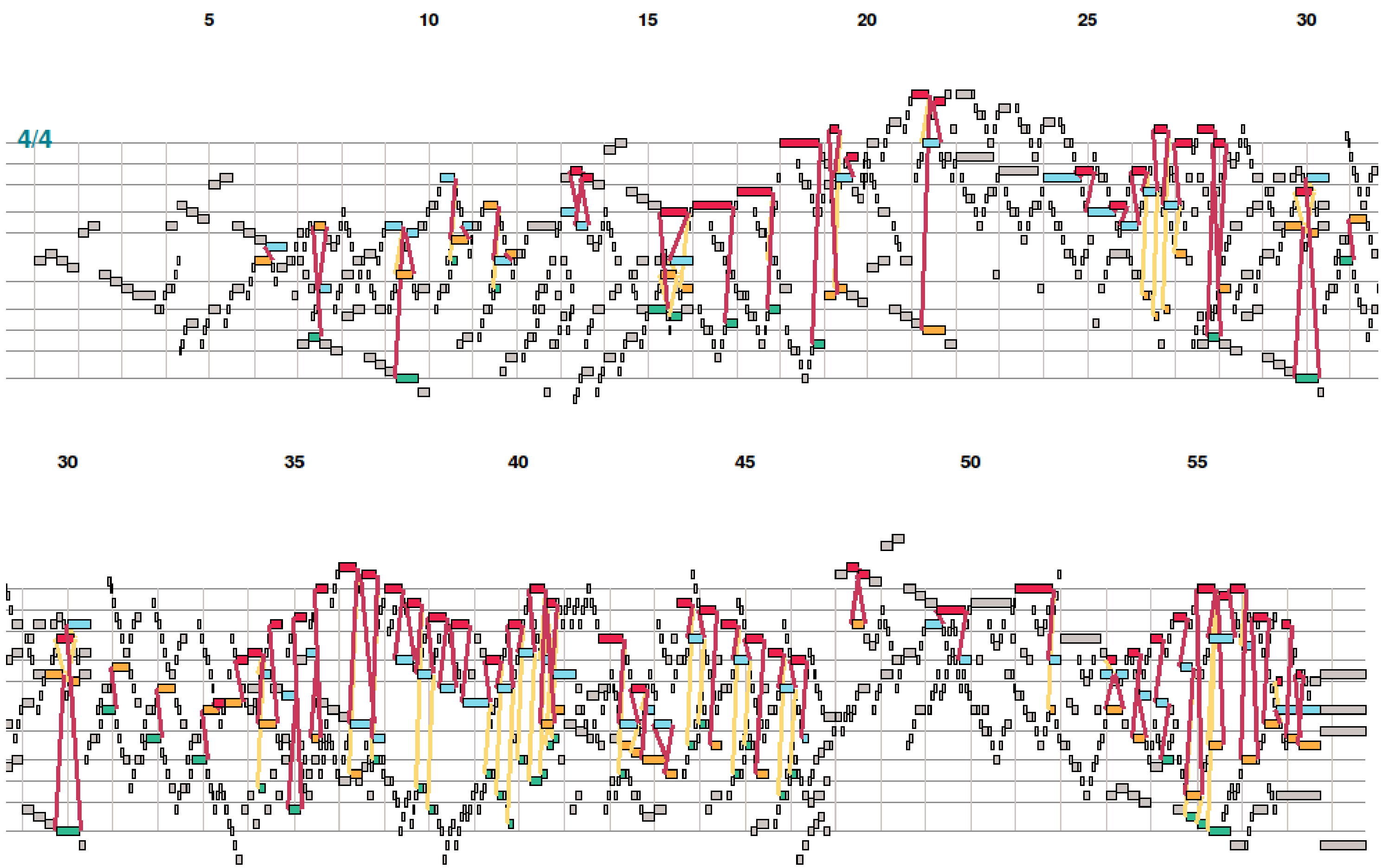

Figure 3–23: Bach: Fugue 12 from WTC Book I. *Folds* are shown with red and gold lines between the end of one note and the beginning of the next. Notes involved with folds are colored with one color per *voice*. Pattern and shape of fold families and chains are observable, revealing repetition and variation, differential textural process and contrapuntal shape.

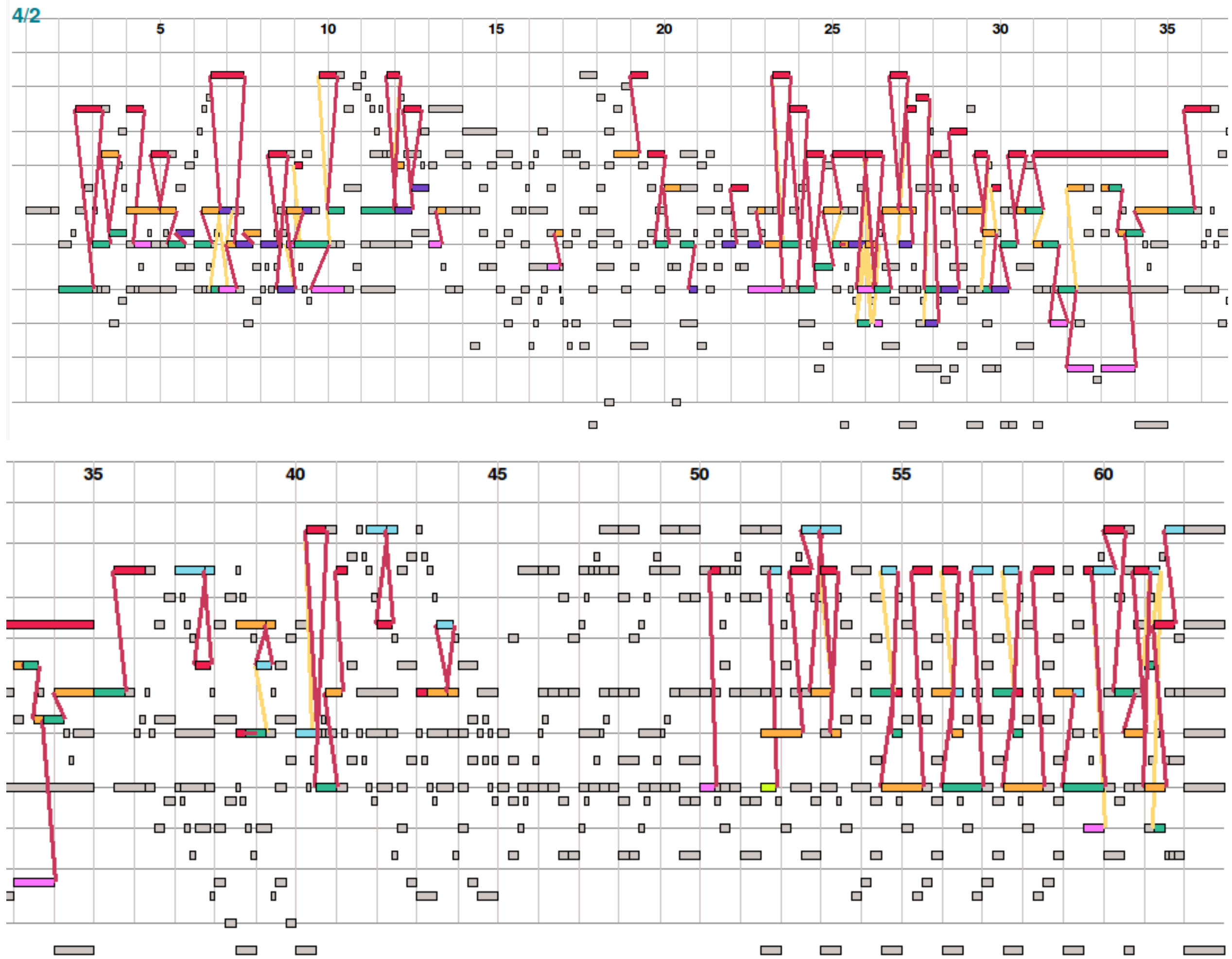

Figure 3–24: Victoria: “Quam pulchri sunt: Agnus Dei.” Folds are shown with red and gold lines between the end of one note and the beginning of the next. Notes involved with folds are colored with one color per voice. Textural and pattern difference is observable: the top panel shows dense, free organization of folds, in contrast to the bottom panel.

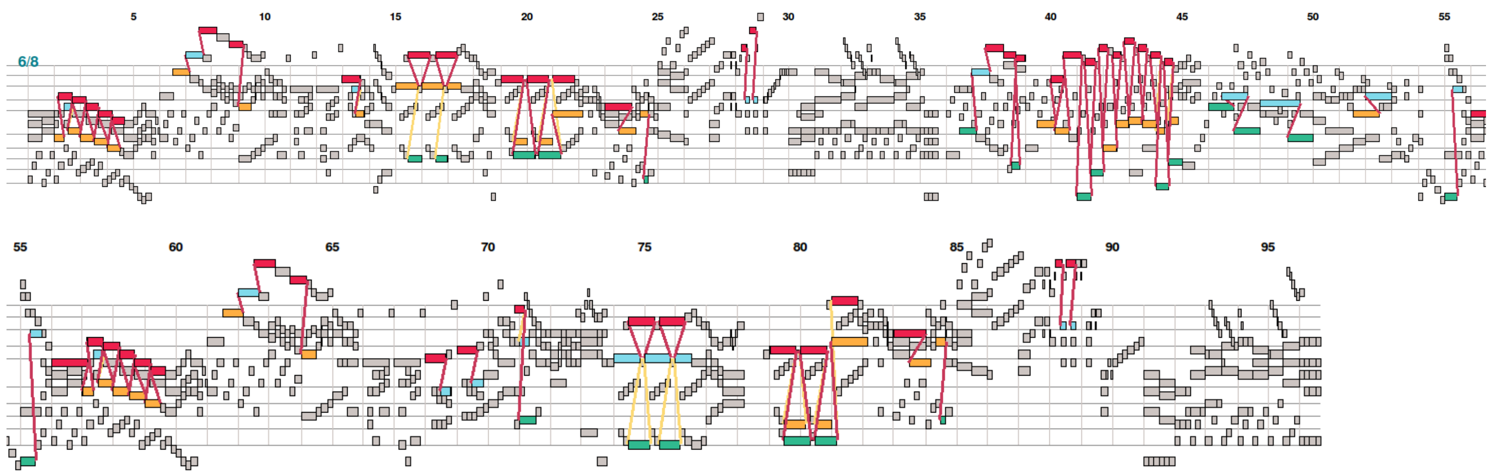

Figure 3–25: Mozart: String Quartet K. 428, Mvt. 2. Folds are shown with red and gold lines between the end of one note and the beginning of the next. Notes involved with folds are colored with one color per voice. Bottom panel recapitulates thematic material from the first half of the top panel, with folds highlighting some permutation of voice roles, but no great textural difference.

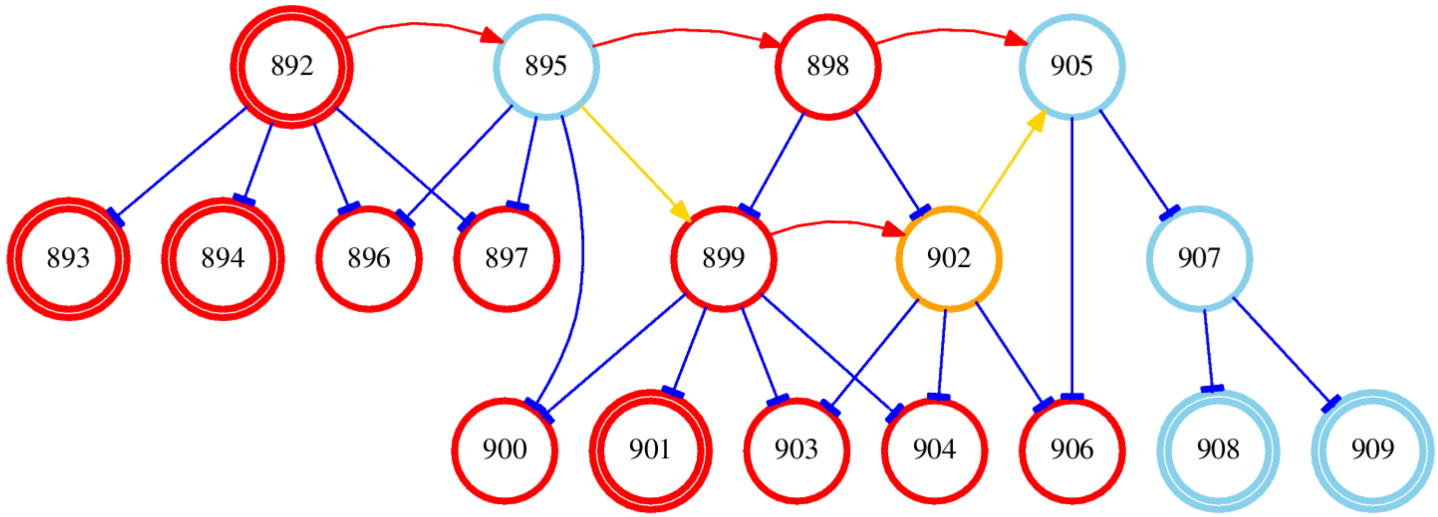

Figure 3–26: A polyphone component from Bach's Ricercare a 6 from the Musical Offering. A minimal coloring is given such that no two folding nodes share a color. The third color is required because of the odd cycle (i.e. 895, 899, 902, 898, and 905).

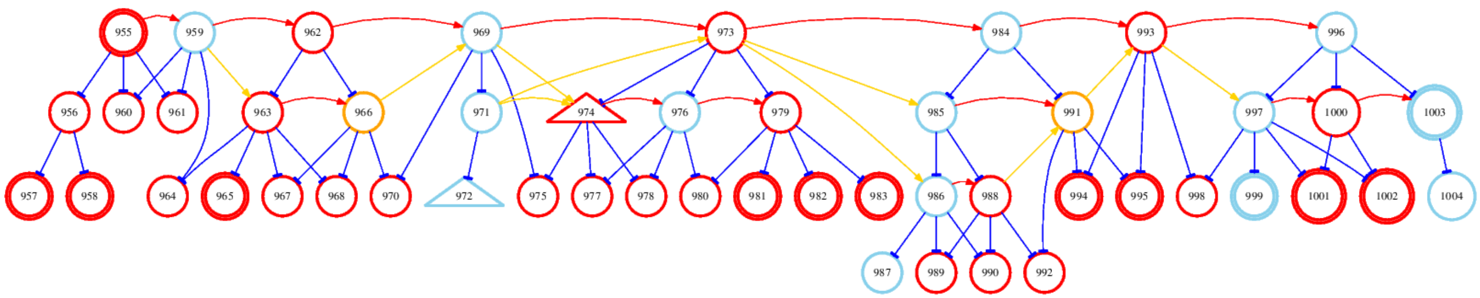

Figure 3–27: A polyphone component from Bach's Ricercare a 6 from the Musical Offering. A minimal coloring is given such that no two folding nodes share a color. Gold nodes (the third color) are required where there are odd cycles. A coloring on fold edges identifies a way of partitioning the graph into "hold-voices" where each "voice" can sing hold-trees but not folds.

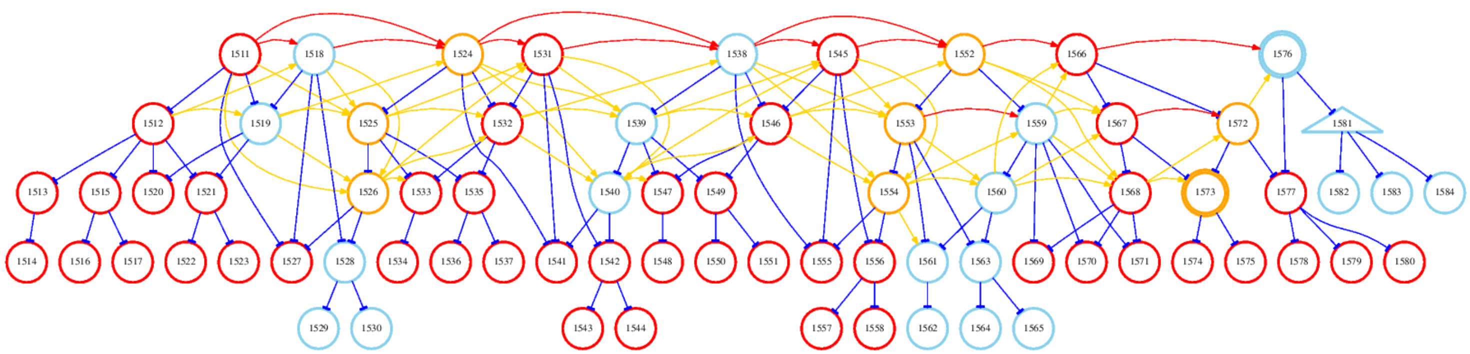

Figure 3–28: A polyphone component from Bach's Ricercare a 6 from the Musical Offering. A minimal coloring is given such that no two folding nodes share a color. This graph has a chain of cliques (leapfrogging red edges at the top left), which is an unusual polyphonic feature.

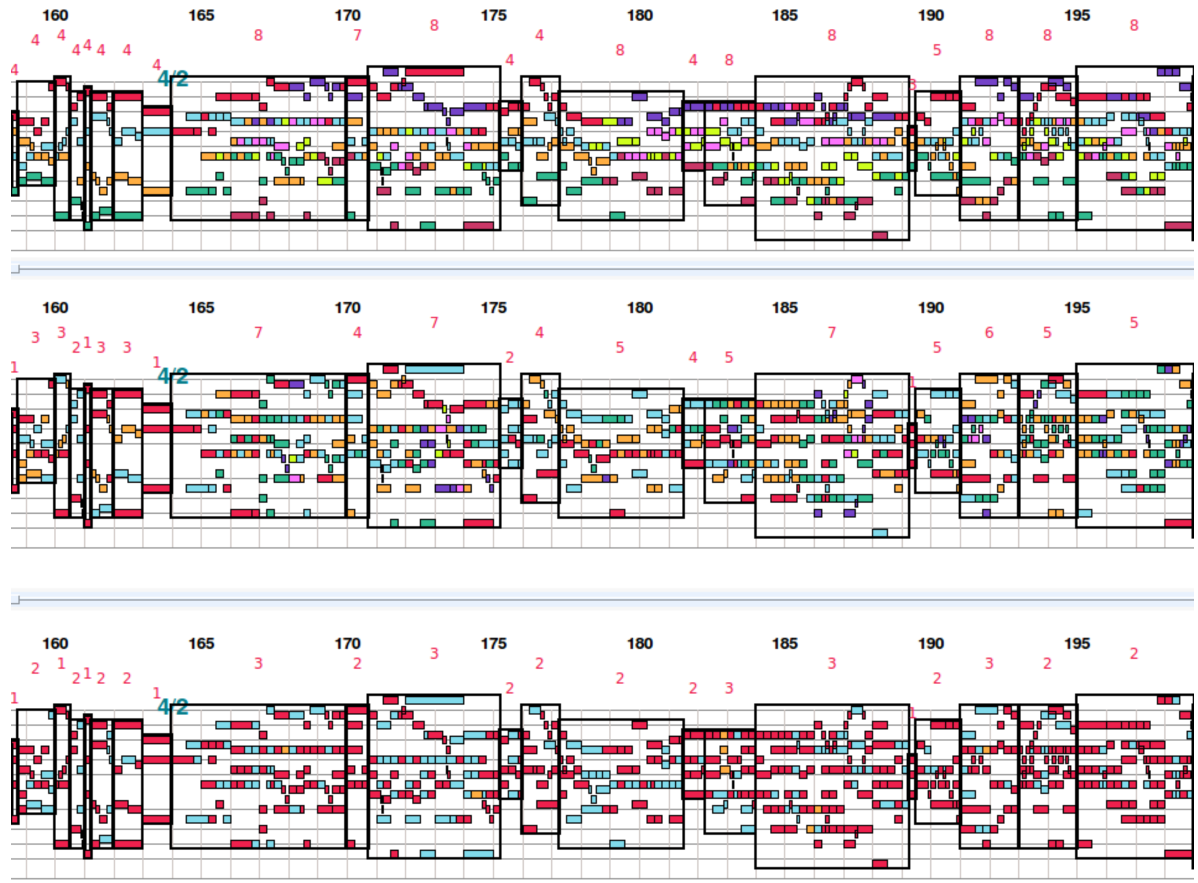

Figure 3–29: Palestrina: Missa *Confitebor Tibi*, Credo (excerpt). Top panel shows number of (phone) voices participating in each polyphone component. Middle panel shows number of *monophone* voices per polyphone component, giving an idea of the *independence* of the phone voices. Bottom panel shows *hold*-voices, giving another measure of independence. Note-colors correspond to “voices.”

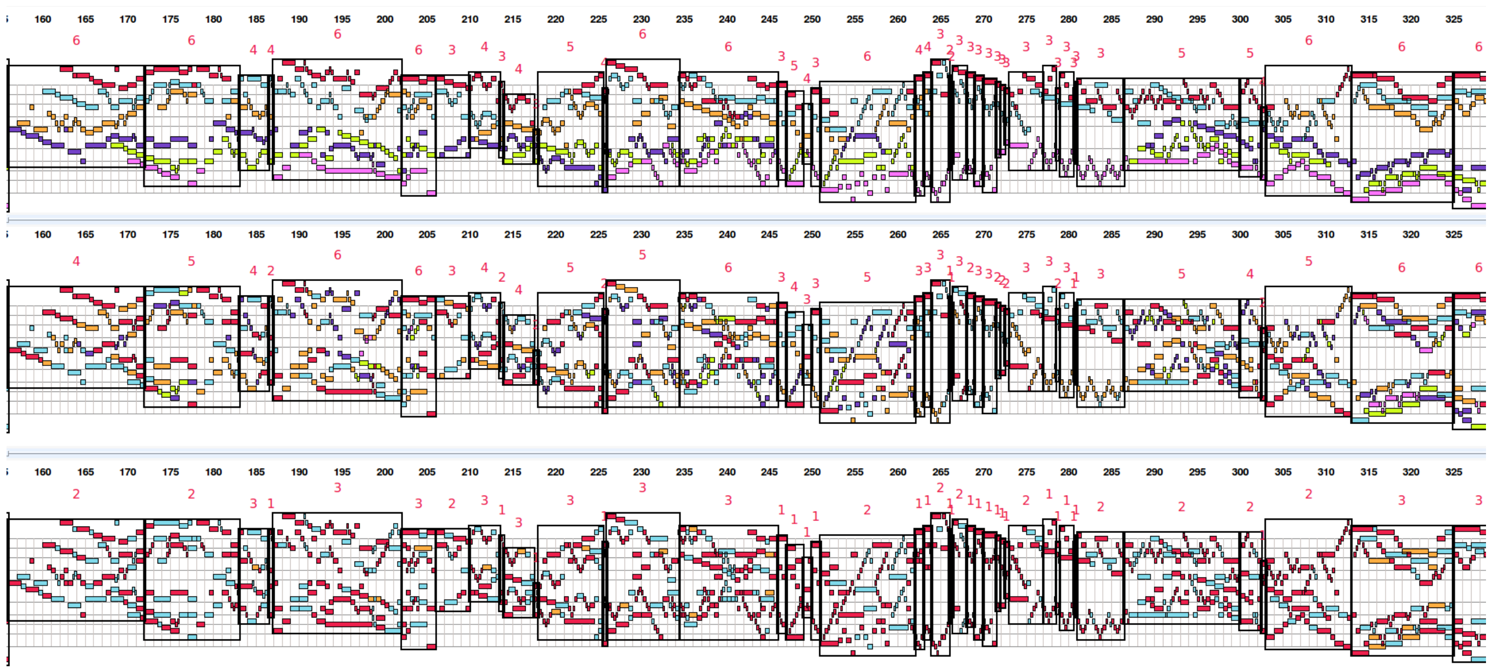

Figure 3–30: Bach: Ricercare a 6 from the Musical Offering (excerpt). Top panel shows number of (phone) voices per polyphone component; middle panel shows number of monophone voices, and bottom panel shows number of hold voices. These are different ways measuring voice *independence*.

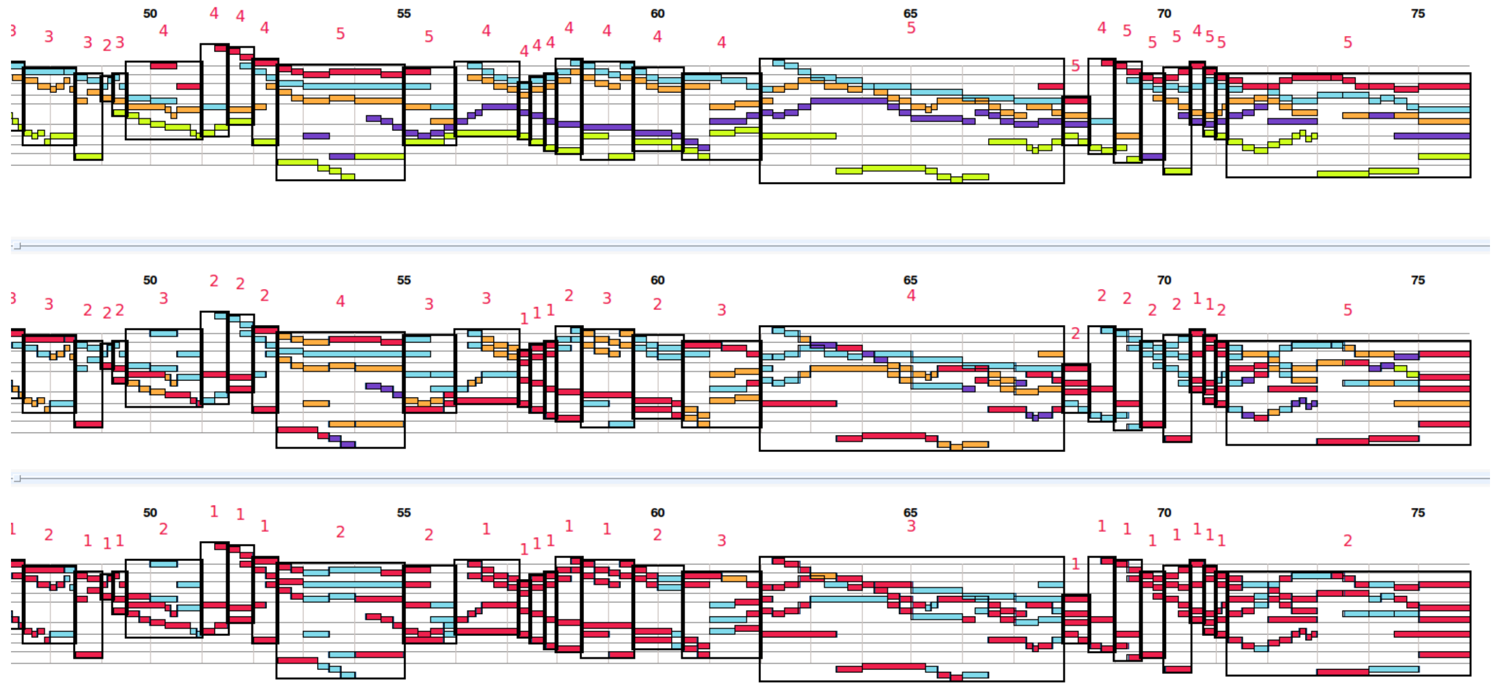

Figure 3–31: Bach: Fugue 22 from WTC Book I (excerpt). Top panel shows number of (phone) voices per polyphone component; middle panel shows number of monophone voices, and bottom panel shows number of hold voices. These are different ways measuring voice *independence* – these vary somewhat independently.

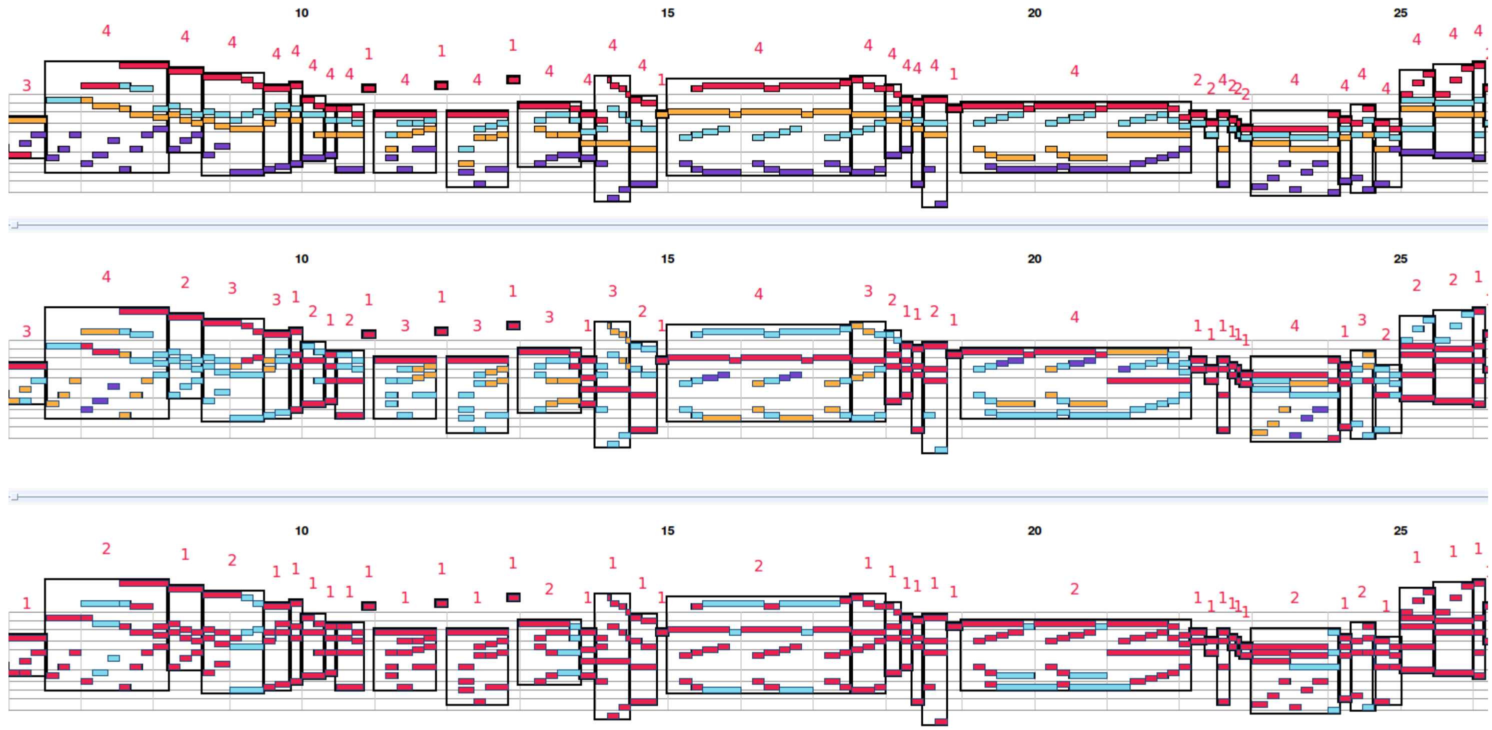

Figure 3–32: Mozart: String Quartet K.428, Mvt 2. Top panel shows number of (phone) voices per polyphone component; middle panel shows number of monophone voices, and bottom panel shows number of hold voices.

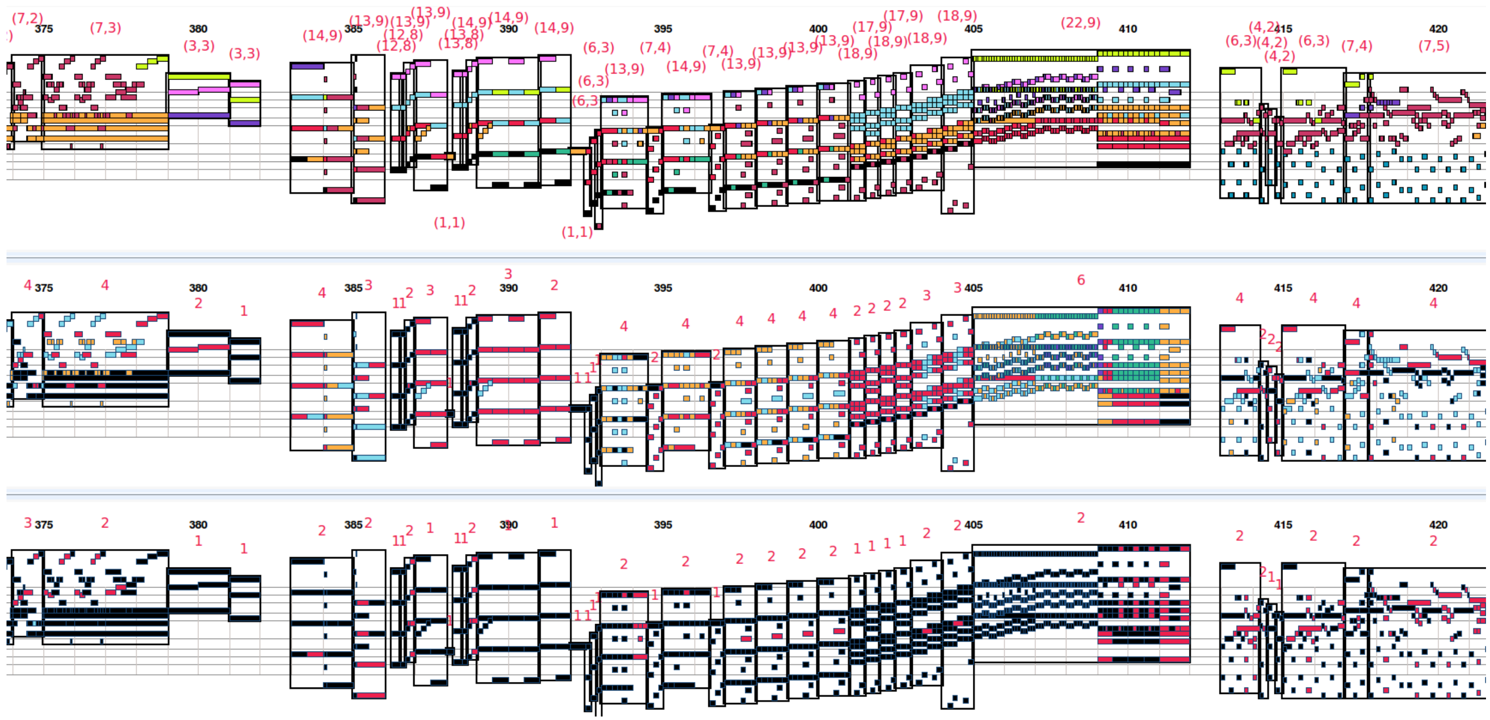

Figure 3–33: Bruckner: Symphony No. 4, Mvt. 4. This score has voices grouped together by timbre: in particular all string parts are grouped. First panel shows number of phone voices and number of timbral voices; second panel has number of monophone voices; third panel has number of hold-voices.

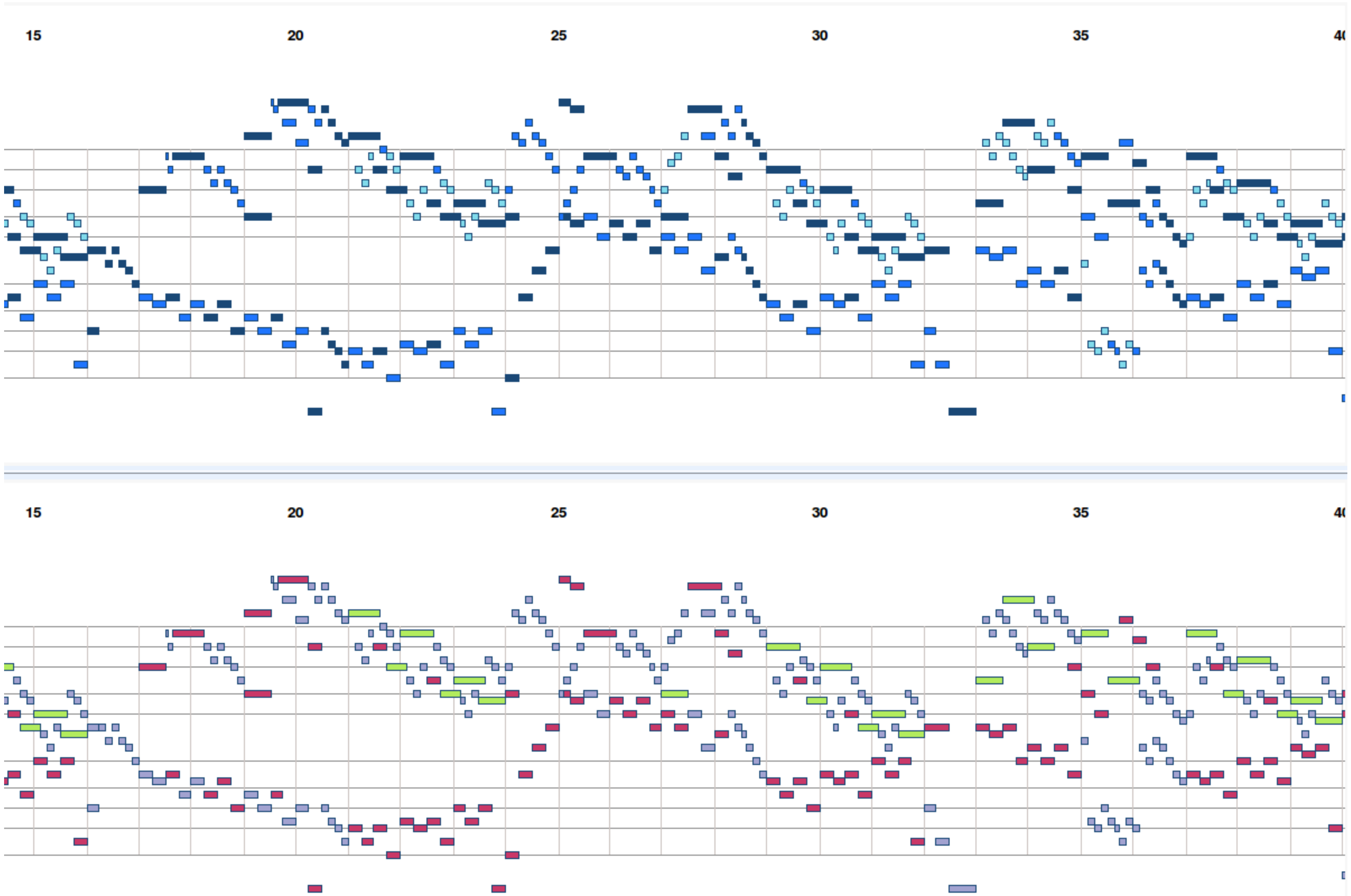

Figure 3–34: Bach: Goldberg Variation No. 2. Top panel has notes colored by hold *depth* and bottom panel by hold *height*. These are complementary views, with dark blue in the *top* panel showing the hold *roots*, and grey in the *bottom* panel showing the hold *leaves*. These two views project sub-scores based on polyphonic place, where each color has its own rhythmic and formal shape on the score.

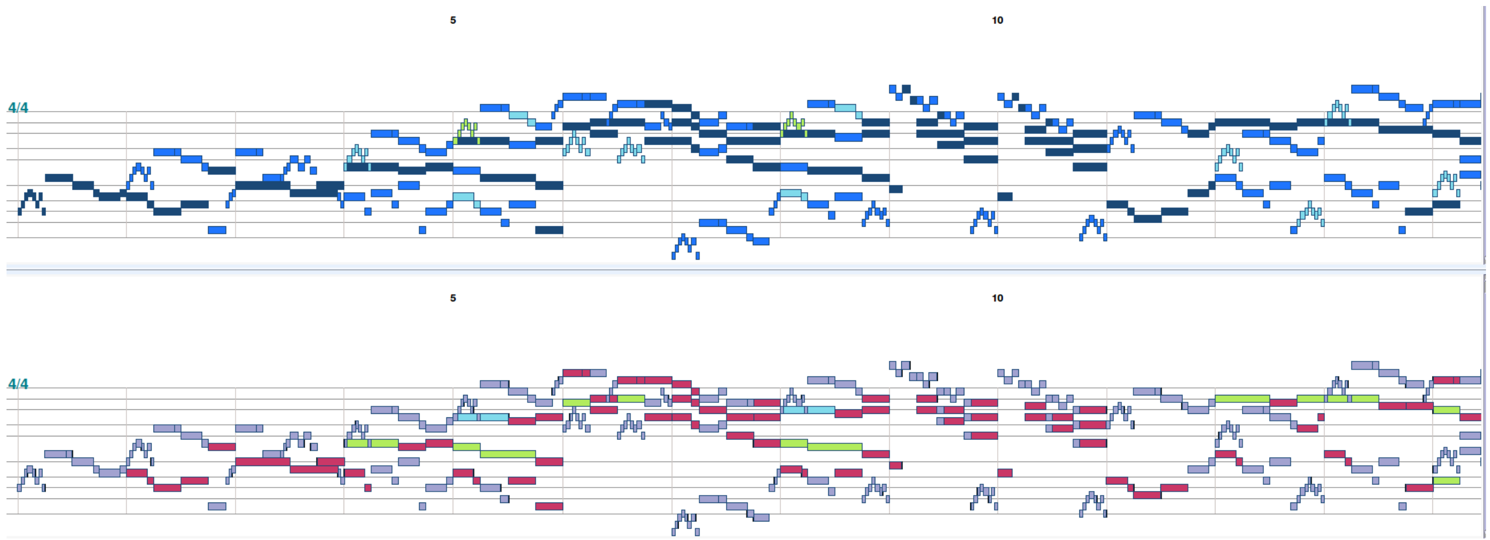

Figure 3–35: Bach: Fugue No. 5 from WTC Book I. Top panel has notes colored by hold *depth* and bottom panel by hold *height*. Color patterning brings out aspect of musical form. For example, the quick motive that begins the score is always grey from the point of view of the bottom panel (i.e. hold-*leaves*), but varies in color in the top panel based on local polyphonic context.

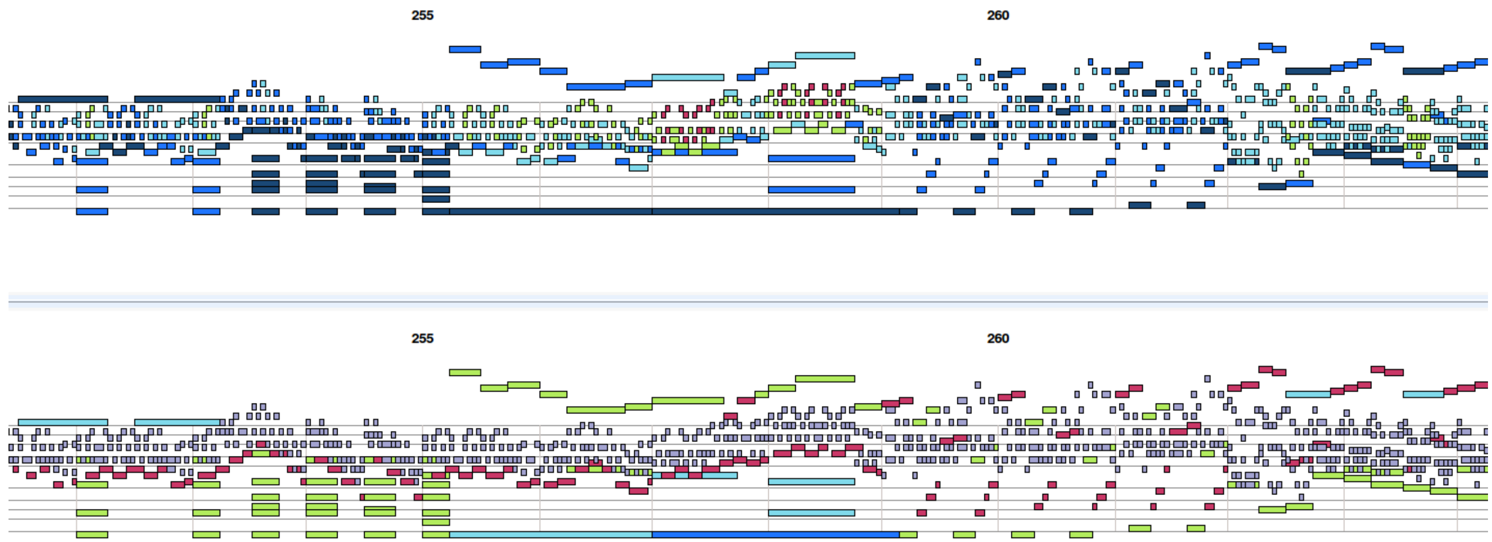

Figure 3–36: Schoenberg: *Verklärte Nacht* (excerpt). Top panel has notes colored by hold *depth* and bottom panel by hold *height*. Hold *height* produces a more “obvious” and continuous partition of textural components, while hold *depth* gives a more discontinuous point of view offering less straightforward and more differentiated structural patterning.

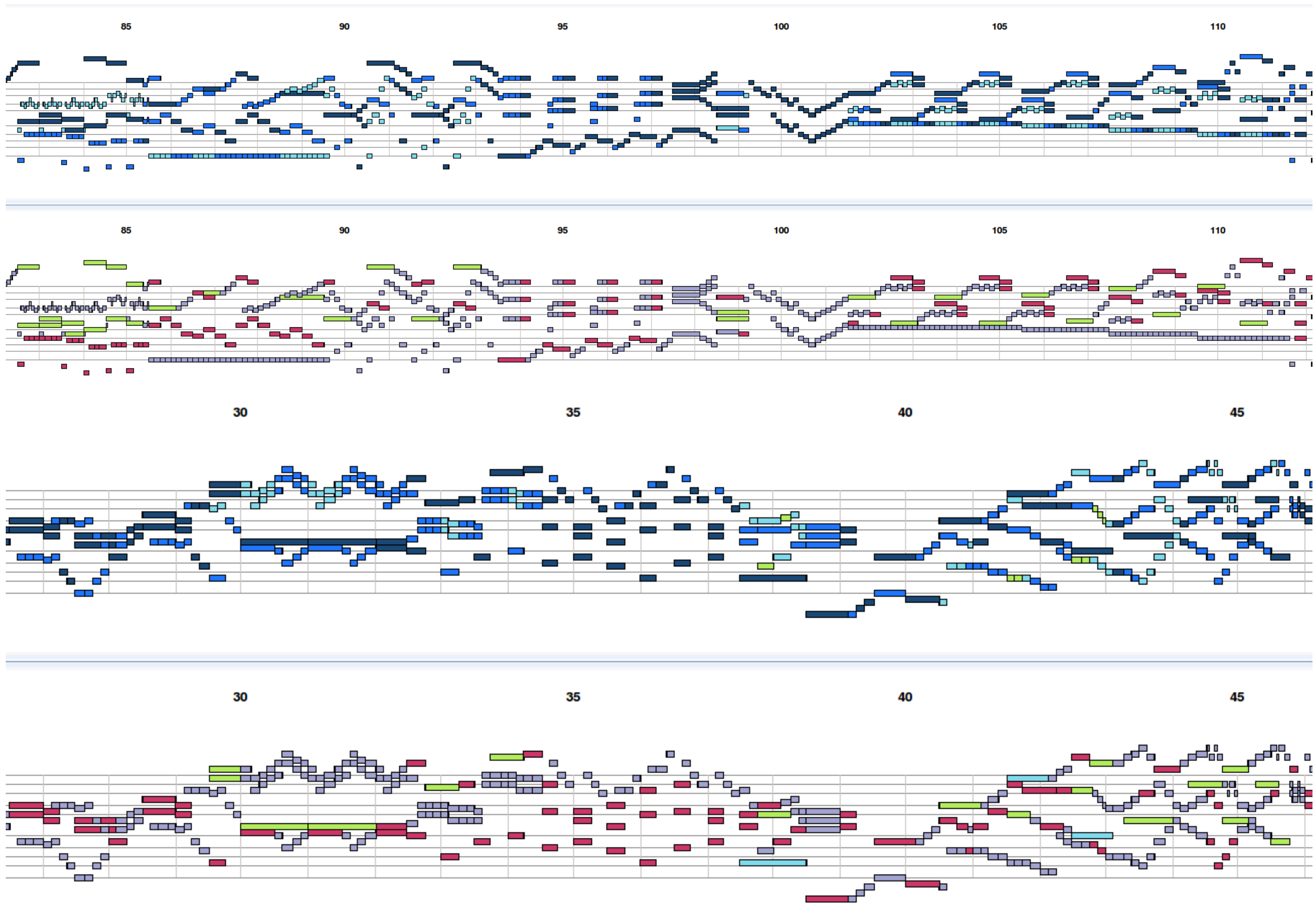

Figure 3–37: Mozart: String Quartet K. 465, Mvt. 1 (excerpts). Panels in blue show hold-depth, grey red and green show hold-height. These colorings show different points of view on textural patterning.

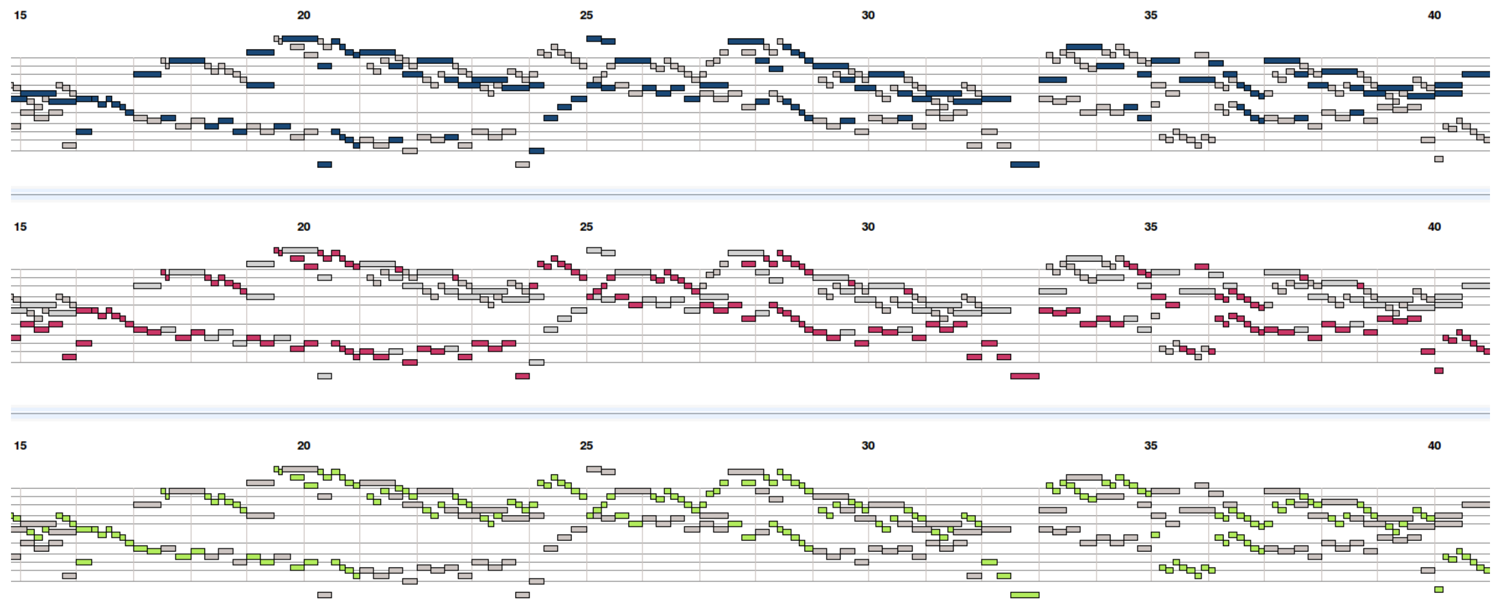

Figure 3–38: Bach: Goldberg Variation No. 2. Top panel is hold-depth 0, middle panel is hold-depth 1 (or 0 where 1 does not exist), and bottom panel is maximum hold-depth (– compare with Figure 3–34). These "voices," projected from polyphonic structure, are each continuous over time, converging and diverging when necessary.

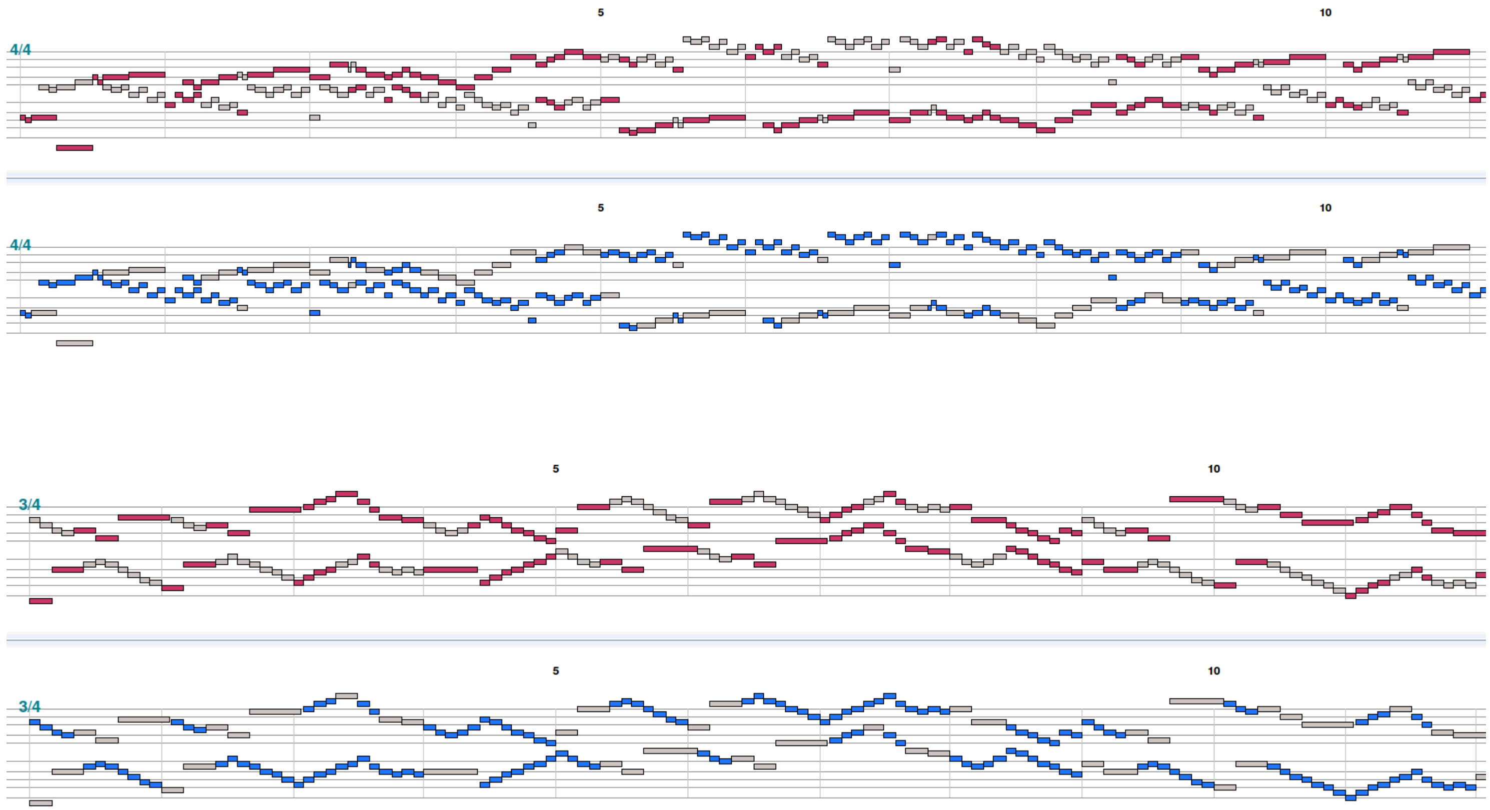

Figure 3–39: Bach: Inventions No. 5 and No. 9 (excerpts). *Hold*-height is used to determine voices, but voice *continuity* (over time) is assured. This gives voices with a pattern of unison and separation and patterns of incidence on the two score voices, as well as different rhythmic profiles – for example, in Invention 9 (bottom), the blue voice has a continuous rhythm of eighth notes, while the red voice has more rhythmic differentiation, while sometimes joining in the eighth notes.

CHAPTER 4
N-sets

4.1 Temporal (N-)sets

4.1.1 Temporal Sets

Temporal sets describe the *content*, in terms of discrete elements, of temporal intervals on a score. Given a score that consists of a set of structures, any temporal interval (t_i, t_o) on that score generates a corresponding subset of score structures that take place during (t_i, t_o). This set, together with its generating interval (t_i, t_o), is a *temporal set* on the score.

Each structure on the score is *tagged* with a label corresponding to the result of a function f of the structure's feature-set. f is any summarizing function, used to *partition* the structure set such that each subset S of the partition is defined by $s_i, s_j \in S \leftrightarrow f(s_i) = f(s_j)$. By tagging each structure with its tag $f(s_i)$, we obtain a reduction (or projection) of the score to a set of tags, each with a temporal interval corresponding to that of the structure it summarizes. The tags are subject to equality-checking, but need not be otherwise comparable.

f should be selected such that the tags are interestingly multi-incident – if all structures have different tags or all tags are the same, then the analysis is trivial. Two structures may have the same temporal interval *and* the same tag: e.g. if the structures are notes tagged by their pitch-classes, then a simultaneous chord containing an octave contains two structures with the same temporal interval and same tag.

Temporal sets on *tagged* structures afford identification of timespans with tag-sets that are set-related to one another (e.g. equal, included, intersecting, disjoint), or timespans that are set-related to sets of heuristic (e.g. semantic) interest.

4.1.2 Temporal N-Sets

If we suppose that structure *onsets* are of particular interest, and seek all temporal-sets containing a unique subset of the structure-onsets on a score, the result is a quadratic number of temporal sets – i.e. if there are m structure-onsets on the score, then there are $m + (m-1) + ...2 + 1 = m(m+1)/2$ temporal sets with unique structure-onset content.[1]

*Temporal **N**-sets* are a subset of these: intervals which are *informative* with respect to a selected tag function of the score structures. N-sets are those temporal-sets that are *locally maximal* with respect to their tag content: extending the temporal interval of an N-set increases the cardinality of its set of tags.

Suppose we have a temporal set $\mathcal{T}$ on a score, where $\mathcal{T}$ is defined by the temporal interval (t_i, t_o): ($t_i \leq t_o \in [t_0, t_1 ... t_m]$ of ordered structure-onsets), and generates a corresponding subset of tagged score-structures with tag-set of cardinality N (i.e. there are a total of N different *tags* occurring in the time interval). Then $\mathcal{T}$ is

[1] Using *both* structure endpoints for each structure, i.e. including sets of onsetless "tails" of structures as interesting temporal sets, we have $m' = 2m$ endpoints to consider. Including both endpoints requires only a trivial complication of the definitions, properties, and algorithms as they are presented here.

a temporal N-set if and only if it is *locally maximal*. $\mathcal{T}$ is locally maximal if the temporal interval (t_{i-1}, t_o) generates a temporal set with tag-set cardinality $> N$ (or if $i = 0$) and the temporal interval (t_i, t_{o+1}) generates a temporal set with tag-set cardinality $> N$ (or if $o = m$). Since expanding the interval of an N-set gives a larger set of tags, the interval of the N-set is locally maximal with respect to its tag-set.

On a score with m different structure-onsets and t different structure tags, there are at most $m * t$ temporal N-sets (justification for this to be found in the next section). N-sets are the most *informative* temporal sets because they express the local boundaries of a given set of N tags. The events to the outside ($i - 1$ and $o + 1$) are known to *break* the set by containing tags not contained in (i, o).

Temporal N-sets give a view of *what happens when*, what *doesn't* happen when, and what things happen more or less "together." From a global point of view, this can give a picture of (musical) *form* – does the score have different sections, marked by the presence and absence of particular structures? Are these obtained through contrasts, or gradual transformations? Is there a thematic or textural common element running through the whole piece? Whereas a traditional account of musical form consists of a single, hierarchical, top-down partition of the score into temporal segments, a more flexible view is afforded by a multidimensional picture with overlaps and contradictions included. N-set analyses of different structures (or structure-reductions) need not be reconciled; and a *single* (non-trivial) N-set analysis includes structured overlaps and inclusions.

The disposition of shapes and themes, and their relations, might give a discursive or narrative sense of "what's happening" in a score, where "what" are structurally identifiable musical objects or features. But the same process of identifying temporal sets can be used on the lowest-level structures. Our primary application takes *notes* as structures, summarized by their pitch-class, in order to build a sense of harmony and tonality, or of atonal pitch-structure.[2] This application is so central to music that pitch-class (pc) N-sets have their own name: PcNs.

The notion of *pitch-class sets* is common in theory and analysis of atonal (e.g. 20th-Century) music – music *without* the traditional notions of tonality (keys) and harmony (chords).[3] The extension of pitch-class sets to PcNs – a systematic, exhaustive structural (and structurable) analysis of the pitch-class sets on a score – is our innovation.

The remainder of this chapter provides algorithmic specification and analysis (Section 2), discussion of structural properties (Section 3), and illustrative applications of PcNs to both atonal *and* tonal music analysis, with attention to related music-theoretic and music-computational literature (Sections 4 and 5).

---

[2] *Pitch-class* is pitch with octave equivalence (i.e. such that middle C and high C have the same pitch-class: C). The common practice in Western music is to use 12 pitch classes, so this is taken as an assumed basis (which is generalizable). In standard notation, each pitch class can be "spelled" in a number of different ways (e.g. C♯ v.s. D♭), but these share a piano key and a midi number (per octave), and are considered to be the same pitch class. By convention, pitch classes are named $\{0...11\}$ with $C = 0$.

[3] The music-theoretic concept of pitch-class sets is due to [Forte1964, Forte1973].

4.2 PcN Definition and Algorithm

We discuss PcNs, with the understanding that the generalization to other temporal N-sets is straightforward.

A pitch-class N-set, or PcN for short, is a maximal temporal interval on a score, during which a specific set of N pitch classes occurs. *Maximal* means that both the *next* and *previous* (onset) events in the score contain a pitch class outside the set, or the next or previous events do not exist (i.e. at the beginning or end of the score). The pitch classes just outside the interval which are not in the PcN set *break* the PcN.

The score provides a *sequence* of time points at which one or more notes enter, where each note has a pitch class. Under the current definition, PcNs are restricted to include onsets, such that if we have two notes A and B with the same onset, and A ends before B, we *do not* obtain a set including *just* the tail end of B. However, if a third note C enters while the tail of B is still sounding, we *do* obtain the set of B and C. It is possible to include both onsets and note-endings, but the restriction simplifies matters and probably finds the most perceptually salient sets.

When several notes share an onset, they are found *together* in *all* sets in which they appear, since no temporal cut can separate these notes into different sets. Allowing chords to be divided arbitrarily would give an unwieldy combinatorial number of sets. It is common, however, for music theorists carrying out pitch-class set analyses by hand to select such divisions. We will approach this problem by using subset matching on the set of PcNs obtained: in this way we stand a chance of finding relevant relations between PcNs without generating a powerset in advance.

The maximum cardinality of any PcN is 12 – or, more precisely, the cardinality of the set of pitch classes on the entire score, which is at most 12 for the standard pitch-system we're assuming. In the general case for N-sets, the maximum cardinality is the cardinality of the total set of tags. The N-set of maximum cardinality is trivially the entire set of structures, with a temporal span equal to that of the score.

Figures 4–1 and 4–2 show all PcNs on a score. The first panel of Figure 4–1 shows the score with notes tagged and color-coded by pitch-class. The remaining panels in both figures show PcNs of cardinalities 1–6 (there are a total of 6 pitch-classes on this score).

If there are m structure-onsets on a score, then the number of PcNs on the score can be upper-bounded at $12m$; in the general case if there are m structure-onsets and t different structure-tags on the score, then the number of N-sets is bounded at $t * m$. This is because there can be at most *one* N-set of a given cardinality N that *starts* at each structure-onset. Given a cardinality N and a structure-onset where the N-set should *start*, it takes time at most $O(m)$ to determine where the N-set should *end* (i.e. by stepping through subsequent structure-onsets until a set of size N is obtained and the *next* step overshoots N). An N-set candidate discovered in this way has a correct end-point given its starting point; but we do not know whether its starting point is the "maximal" (i.e. earliest) one for the tag-set obtained. Supposing we had already obtained N-sets of cardinality N for the all of the previous starting-points on the score, we need only check that the current candidate is not temporally contained in any of these previous N-sets, and since there are at most m N-sets of cardinality N starting at an earlier time, this check takes $O(m)$ time. Since $O(m)+O(m) = O(m)$, it takes $O(m)$ time to discover *and* verify the maximality of each N-set. Therefore we will be able to find all of the N-sets on a score in $O(t * m^2)$ time.

Figure 4–3 shows this method (for PcNs), but with maximality-checking carried out after the candidate-enumeration phase (which could be less efficient unless cleverly organized).

It's possible to optimize the algorithm (without improving the asymptotic running time) by noticing that we need not attempt to build a N-set starting on *every* structure-onset; we can take advantage of the structure of *breaks*. Once we obtain an N-set p with cardinality N and endpoints (p_i, p_o) that *breaks* on some event p_{o+1}, we know that any N-set of cardinality N starting after p_i but *not* including p_{o+1} will have the same tag-set as p, and therefore will not be maximal. Therefore, we can begin by including p_{o+1} in the next potential N-set, work backwards to find out how many previous structures should be included (such that *not more than* N tags are obtained), and then work forward from p_{o+1} to find out how many subsequent structures should be included (such that *exactly* N tags are obtained). If no N-set of cardinality N is obtainable from p_{o+1}, we continue sequentially with p_{o+2}, and so on.

Although we look both backward *and* forward for each N-set, the total number of steps taken to find one N-set is still linear ($O(m)$), and is only one more than the number of steps taken by the forward-only search (i.e. since we seek to overshoot the bounds of the N-set on *both* sides). This modification to the algorithm is more efficient because it avoids having to start a search at every event, and also avoids finding redundant non-maximal N-sets, so the verification step is no longer needed. Pseudocode (for PcNs) is shown in Figures 4–4 and 4–5

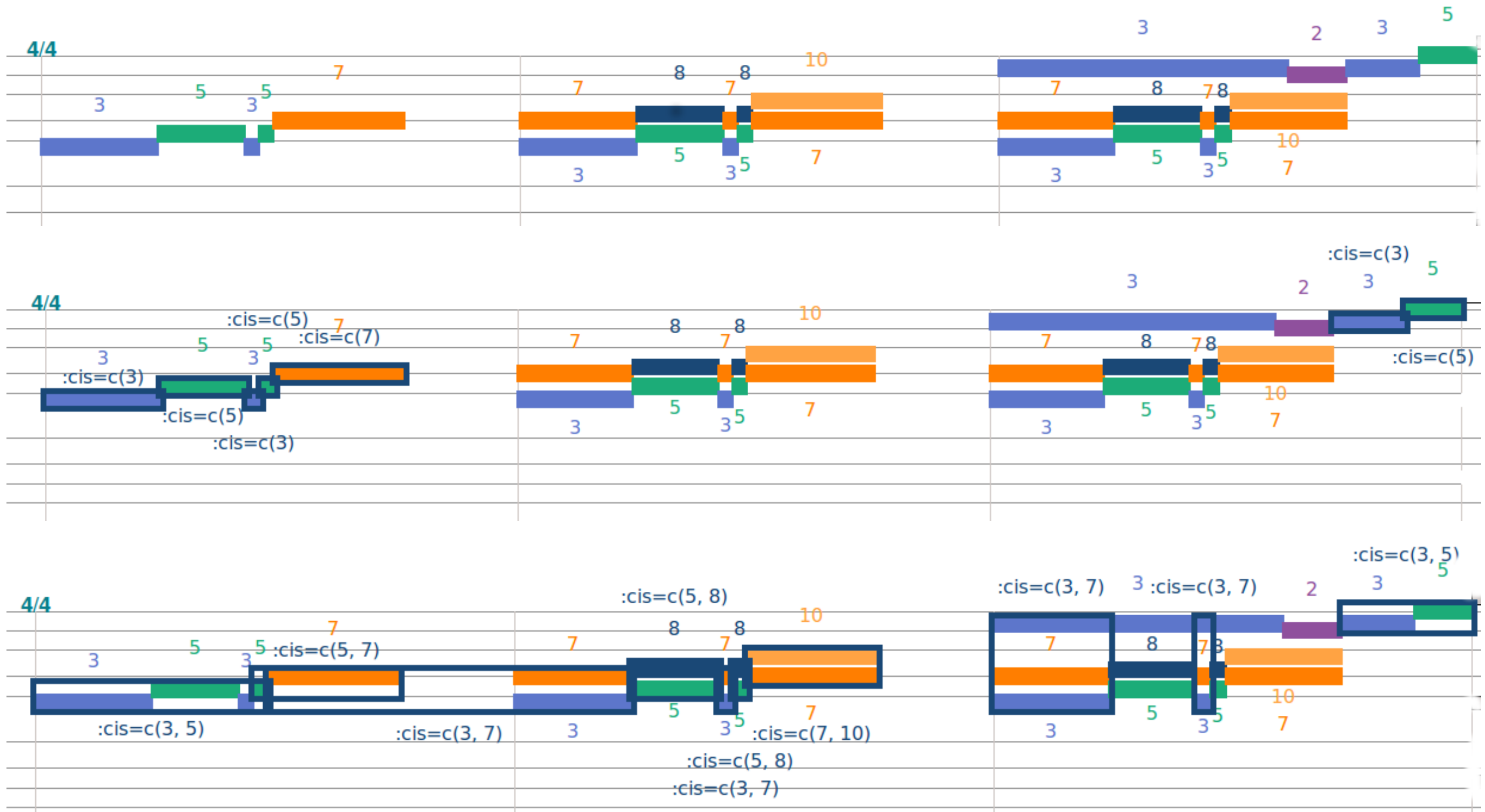

Figure 4–1: Notes-as-structures tagged by pitch-class number and color; PcNs of cardinality 1, and 2. PcNs are defined by the pitch-class content of maximal temporal spans containing exactly N pitch-classes. (Excerpt from Haydn: String Quartet Op. 1, No. 0, Mvt. 1).

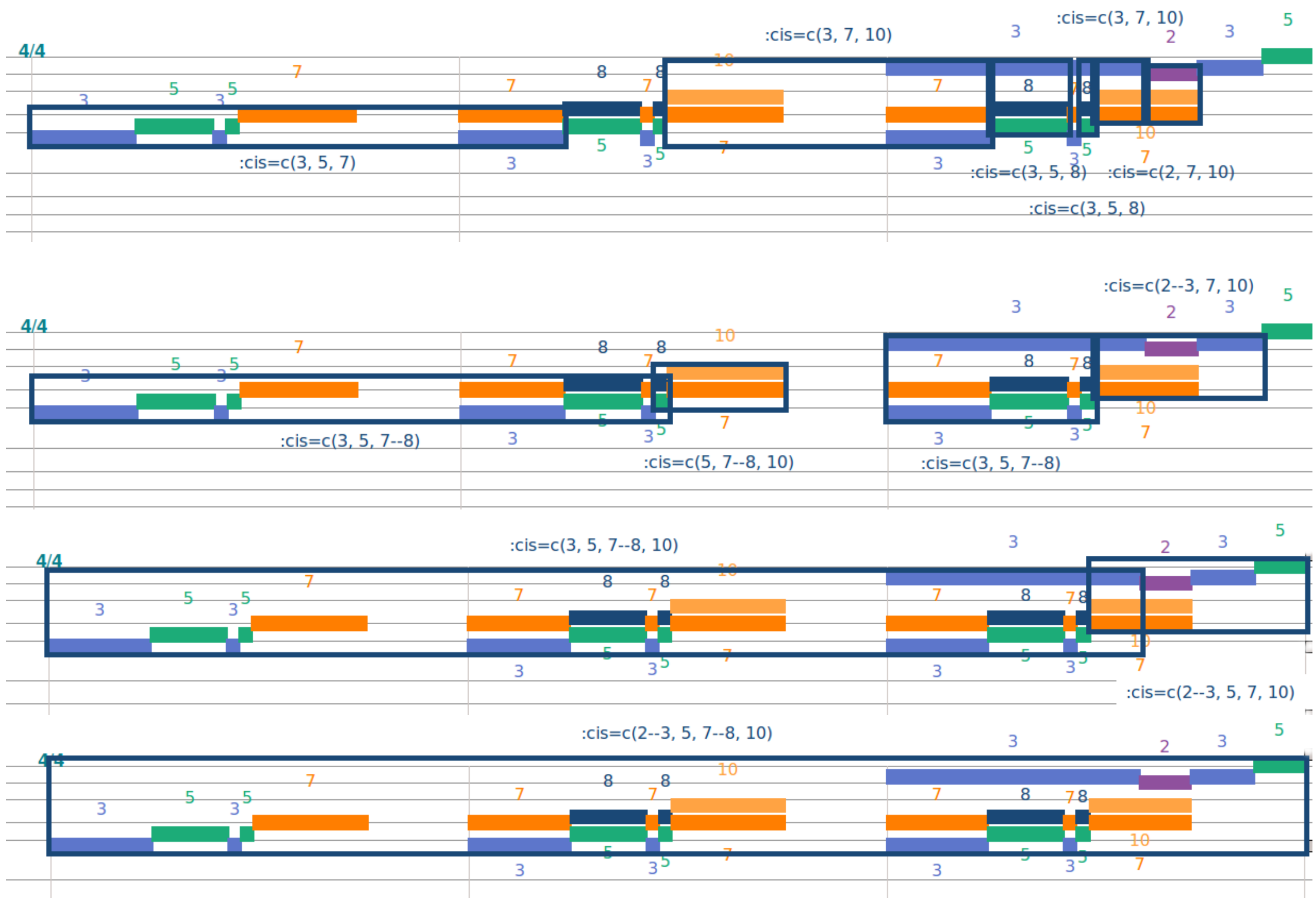

Figure 4–2: Continued from Figure 4–1; PcNs of cardinality 3, 4, 5, and 6.

Find One PcN

Input:

- E: a sequence of events $(e_1, e_2 ... e_m)$ such that each event is a set of pitch-classes.
- N: an integer $(0 < N < 12)$.
- S: a starting-point on E (i.e. e_s).

Method:

if $|e_s| > N$ **then** return Null **else**

```
let loop index pcn_set =
        if index = m then
                if |pcn_set| = N then return (S,index,pcn_set)
                else return Null
        else
        let next_union = ⋃ (pcn_set, e_(index+1)) in
        if |pcn_set| = N and |next_union| > N
        then return (S,index,pcn_set)
        else if |pcn_set| < N and |next_union| > N
        then return Null
        else if |next_union| ≤ N
        then loop (index + 1) next_union
in
loop S e_s
```

Find All PcNs

Input:

- E: a sequence of events $(e_1, e_2 ... e_m)$ such that each event is a set of pitch-classes.

Method:

1. `results` = $\{(1, m, \bigcup E)\}$;
   `total_card` = $|\bigcup E|$
2. **for** N = 1 to (`total_card` - 1)
   for S = 1 to m,
   `results` $\Leftarrow$ `find_one_PcN`(E, N, S)
3. For each result $p = (p_i, p_o, p_{set})$, if there exists some other result $q = (q_i, q_o, q_{set})$ such that $p_{set} = q_{set}$ `and` $q_i < p_i$ `and` $q_o \geq p_o$ then remove p from the result pool.

Figure 4–3: (Unoptimized) algorithm for finding all PcNs on a sequence. The `loop` is a tail-recursive iteration, not a true recursion.

Find One PcN (v.2)

Input:

- E: a sequence of events $(e_1, e_2 ... e_m)$ such that each event is a set of pitch-classes.
- N: an integer $(0 < N < 12)$.
- S: a point on E (i.e. e_s) that will be in the resulting PcN.

Method:

```
if |e_s| > N then return Null else

let forward_loop start index pcn_set =
      if index = m then
            if |pcn_set| = N then return (start,index,pcn_set)
            else return Null
      else
      let next_union = ⋃ (pcn_set, e_(index+1)) in
      if |pcn_set| = N and |next_union| > N
      then return (start,index,pcn_set)
      else if |pcn_set| < N and |next_union| > N
      then return Null
      else if |next_union| ≤ N
      then forward_loop start (index + 1) next_union
in

let backward_loop index pcn_set =
      if index = 1
      then forward_loop index S pcn_set
      else
      let prev_union = ⋃ (pcn_set, e_(index-1)) in
      if |pcn_set| ≤ N and | prev_union| > N
      then forward_loop index S pcn_set
      else if | prev_union| ≤ N
      then backward_loop (index - 1) prev_union
      else return Null
in

backward_loop S e_s
```

Figure 4–4: Inner loops for optimization (Figure 4–5) of Figure 4–3. The forward and backward loops are linear iterations written in tail-recursive style, not true recursive functions.

Find All PcNs (v.2)

Input:

- E: a sequence of events $(e_1, e_2 ... e_m)$ such that each event is a set of pitch-classes.

Method:

```
1. results = {(1,m,⋃E)};
   total_card = |⋃E|
2. let pcn_end_index (start,end,set) = end
3. let outer_loop N S =
        if S > m then return results else
        let pcn = find_one_pcn_v2(E, N, S) in     %(call to Figure 4–4)
        results ⇐ pcn;
        if pcn = Null then outer_loop(N, (S+1))
        else outer_loop (N,(pcn_end_index(pcn) + 1))
4. for N = 1 to (total_card - 1)
        outer_loop N 1
```

Figure 4–5: Optimization of Figure 4–3; likewise finds all PcNs on a sequence.

4.3 Structure of Sets of N-Sets

Running the N-set (or PcN) algorithm locates all maximal temporal sets on the score. These have a structure of inclusion and overlap with one another, since any one structure may participate in several N-sets. This section describes structural properties of a *natural* set of N-sets obtained from one application of the N-set algorithm (not a mixed set pulled from a union of structure-sets or tag functions).

We can visualize the structure of a natural set of N-sets as a *polyphone*, since polyphones describe the structure of temporal overlap and inclusion of a set of temporal intervals. Because of the way N-sets are defined, this polyphone has particular structural properties:

1. *Monophones of cardinality 1:* Monophones are sets of phones (polyphonic atoms) that have the same temporal interval. Each monophone in the N-set polyphone contains just one N-set: by definition, it is impossible for two different N-sets to have identical time intervals.

2. *One top:* The polyphone graph of N-sets contains just one connected component, with a single root at the top. This is because there is an N-set that includes all of the structures in the score, temporally including all of the other N-sets. This corresponds to a "top" node of the polyphone, with *hold*-edges to all of the other N-sets.

3. *Meet N-sets:* If two N-sets x and y *fold* (overlap) with one another, giving a polyphone edge $F(x, y)$, then their pivot (temporal intersection) is *also* an N-set in the polyphone. The pivot interval is $p = (y_i, x_o)$. Because of the construction of x and y as maximal intervals, we know that x_{o+1} contains an element not in p, and likewise that y_{i-1} contains an element not in p. Therefore p is a maximal interval for its set.

4. *Join N-sets:* If two N-sets x and y fold with one another $F(x, y)$, then there is another N-set z in the polyphone such that $tagset(z) = \bigcup(tagset(x), tagset(y))$, and there are polyphone edges $H(z, x)$ and $H(z, y)$ (i.e. x and y are temporally included in z). However, we do not know whether $z_i = x_i$, nor whether $z_o = y_o$, since it could be the case that $tagset(x_{i-1}) \subseteq tagset(y)$ or $tagset(y_{o+1}) \subseteq tagset(x)$. But since we know that the set of z is obtainable at minimum by (x_i, y_o), we can infer by construction that this interval or a temporal extension of it is an N-set with the set

z. This rule also holds if $x_{o+1} = y_i$ (i.e. x and y don't fold, but no structure occurs between the end of x and the beginning of y).

5. *Bottoms don't fold:* There are a set of *bottom* N-sets that are included in other N-sets (i.e. with hold edges $H(other, bottom)$), but do not include any other N-sets (i.e. $\neg\exists x : H(bottom, x)$). These bottom N-sets do not participate in any fold edges ($\neg\exists x : F(bottom, x)$; $\neg\exists x : F(x, bottom)$). Proof by contradiction: given two bottom N-sets that fold, then by 3 above there is a meet N-set that is included in both; therefore our givens are not bottom N-sets.

Figure 4–6 shows the polyphone of PcNs from Figures 4–1 and 4–2.

4.4 Analysis of Atonal Pitch-Structure

In this section we show how PcNs can be used in similar analyses to those done by music theorists, contributing increased formality, flexibility, and the availability of a large amount of organizable information.

While the pitch relations in tonal music are dominated by a system of chords and keys, the pitches in *a*tonal music may be organized in any number of ways. One way that analysts can approach atonal music is to investigate the use of one or several pitch-class sets on a piece. This has been a common practice since Allen Forte's introduction of the method in the 1970s.[4] Forte's essential contributions include the concept of the unstructured *set* of pitch classes as a musical object and a method for *normalizing* pitch-class sets (described in the next subsection).

[4] [Forte1973], predated by the less well-known [Forte1964].

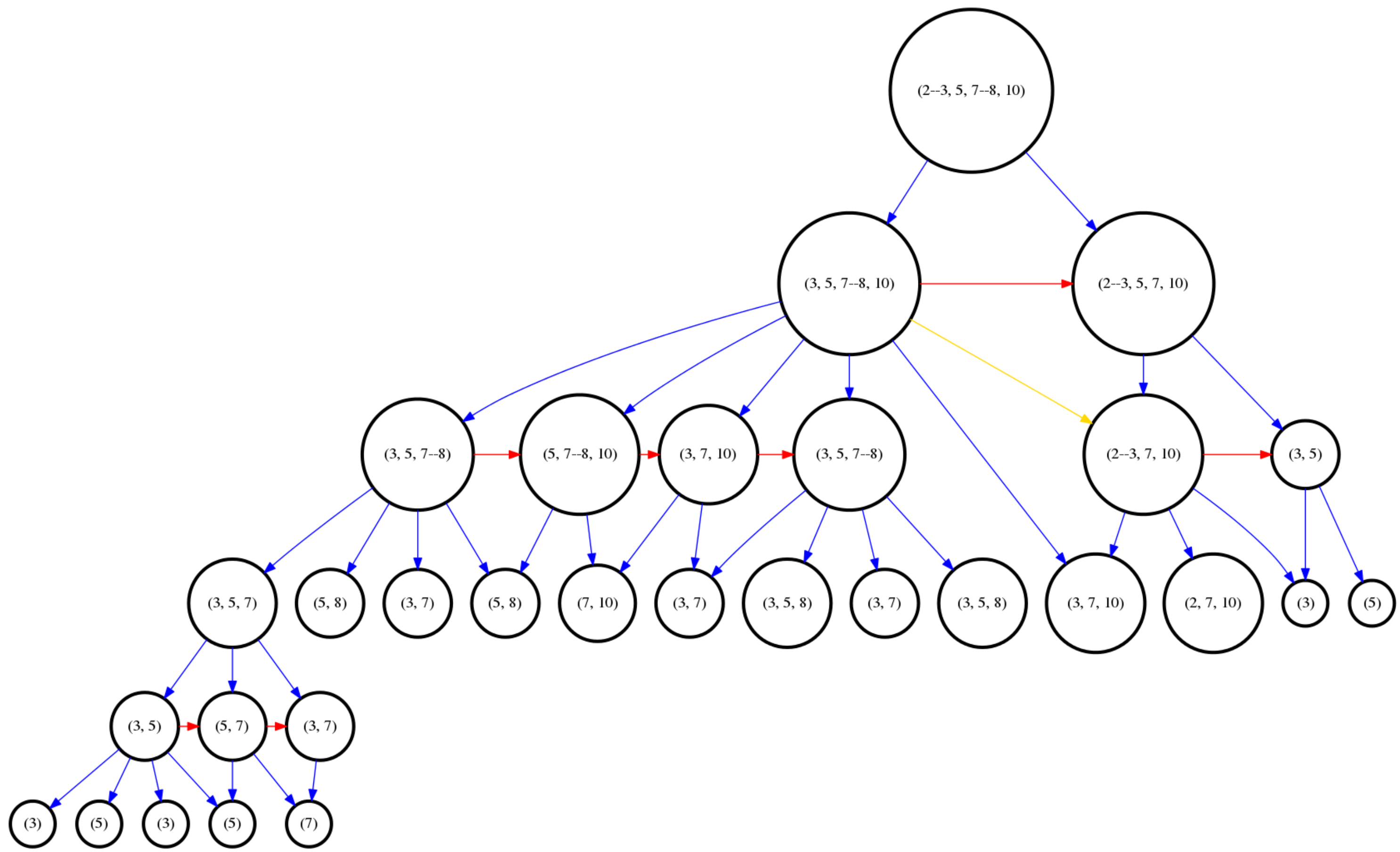

Figure 4–6: Polyphone graph for the PcNs from Figures 4–1 and 4–2 (from Haydn: String Quartet Op. 1, No. 0, Mvt. 1). Blue edges are holds (inclusions), red and gold edges are folds (overlaps); hold-children are ordered from left to right by onset time.

PcNs are a method to *locate* a set of pitch-class sets on a score. Music analysts usually find pitch-class sets "by hand," by looking at and listening to the score. PcNs offer the usual advantages of computation: convenience and scale. They also offer a mathematical formality, guaranteeing a particular *structural* meaning for each identified temporal interval by omitting temporal intervals that do not have informative boundaries, and by finding a *complete* set of informative intervals.

A pitch-class set analysis only *begins* by identifying pitch-class sets of interest on a score, then a musical interpretation takes place, taking into account the musical context of the pitch class sets, their relations to one another, etc. Illustrations in this section show sets of PcNs organized by simple functions of pc-set relations and temporal relations – *interpretation* is beyond our current scope.

Music analysts do not typically make any claims as to the *completeness* of their pitch-class set analysis – in any case a musical analysis is *always* incomplete, and there is no reason why a discursive interpretation of a musical score should include "all" available information. However, with PcNs, it *is* possible to make definite statements about a systematic accounting of pitch classes that is complete with respect to certain formal conditions: though it is possible to obtain other pitch-class sets (e.g. by slicing chords in different ways), PcNs show *all* locally maximal, temporally compact pitch-class sets on a score. In this sense, PcNs offer a more formal approach than the traditional method. In case of an aesthetic or interpretive selection of pitch-class sets, PcNs show a formal background against which these can be positioned – or *from* which they can be selected.

Finding pitch-class sets by hand is time-consuming; a computational methodology for finding, organizing, and displaying these reduces the manual labor that precedes a music-analytic discussion; it also affords a *quick* look at a piece of music to get a sense of how its pitch organization operates, even if no in-depth interpretation results from this, and this rapidity allows impressionistic observation of a larger number of scores.

Since PcNs make pitch-class sets computationally available, these can be made a part of a larger computational analytic process or system, and corpora of these structures (and structures of these structures) can be subject to searching, counting, comparison, and other data analyses.

4.4.1 Forte-normalization

Forte's normalization scheme affords the identification of different pitch-class sets that are the same under transposition (i.e. geometric translation modulo 12), for example {0; 1; 3} and {5; 6; 8} and {0; 2; 11} have the same normal form. Since pitch perception is typically *relative*, pitch-class sets with the same normal form sound very similar, and each normal form can be related to a harmonic "quality." Tonal harmonic units, such as triads, seventh chords, and scales, can be specified by pitch-class sets; Forte's system is a natural generalization of this.

Forte's original normalization scheme also identifies different sets that are the same under transposition *and/or inversion* (i.e. geometric reflection): for example $\{0;1;3\}$ and $\{0;2;3\}$. However, this normalization is too strong for some applications, since the inversionally related forms do not sound as similar as the transpositionally related forms: for example minor and major triads are inversionally related.

A commonly used variant of Forte's method therefore keeps inversions separate, and only uses transpositional normalization.

The Forte-normal form *without* inversional invariance of a pitch-class set can be obtained by finding the transposition of the set that includes 0 *and* has the minimum possible interval between 0 and the largest number in the set. For example, the set $\{4;8;9;11\}$ has transpositions $\{\{2;3;5;10\};\{1;6;10;11\};\ \{2;6;7;9\};\{0;1;3;8\};\ \{4;8;9;11\};\{0;4;5;7\};\ \{3;7;8;10\};\{1;5;6;8\};\ \{3;4;6;11\};\ \{1;2;4;9\};\ \{0;5;9;10\};\ \{0;2;7;11\}\}$; we select $\{0;4;5;7\}$ as the normal form.

The Forte-normal form *with* inversional invariance of a set can be obtained by finding the Forte-normal form of the set (as above) *and* the Forte-normal form of the set's inversion (as above), and selecting the one of these that is most "tightly packed" to the left, with smaller intervals first. For example, the Forte-normal form (no inversion) of {2; 6; 9} is {0; 4; 7} and the Forte-normal form of the inversion of {2; 6; 9} is {0; 3; 7}. Since {0; 3; 7} has the smaller interval to the left, it is the normal form (with inversion) of the set.

A pitch class set can be fully specified by a Forte-normal form, a transposition number, and an indication whether an inversion is required to get from the normal form back to the original set: a set is either in "inverted" (I) or "prime" (P) form compared to its normalization. Our convention is as follows: if the set is not inverted, then simply transpose each pitch-class in the normalized set by the transposition number to get the original set. If the set is inverted, then transpose the *normalized* inversion by the transposition number. For example, to decode the normalization

($\{0;3;7\}$, trans $= 2$, inv $= I$): begin by taking the normalized inversion of $\{0;3;7\}$, which is $\{0;4;7\}$. Then transpose this by 2 to obtain $\{2;6;9\}$.

4.4.2 Stockhausen: Klavierstück III

This subsection shows PcNs close-up on a small score, and the next subsection shows a more zoomed-out view on a larger one.

As a first example of an atonal PcN analysis, we take Stockhausen's Klavierstück III. This score is selected because it is very short, and because David Lewin's pitch-class set analysis of it [Lewin2007] provides a point of departure.[5] This is a difficult piece, however, and a pitch-class set analysis of it does not yield obvious through-going structure.

Lewin selects the normalized set $\{0;1;2;3;6\}$ and identifies some of the instances of it on the score, where instances are allowed to skip some notes. These selected sets are shown in Figure 4–7. Many of Lewin's $\{0;1;2;3;6\}$ sets will not show up as PcNs, because they contain one, two, or three extra notes. To find these kinds of sets, we can ask which PcNs are a *superset* of $\{0;1;2;3;6\}$ (i.e. where the subset is a Forte-normal form that can occur in the superset in any transposition/inversion).

The result of the Forte-superset function is an account of *how* the subset can be found in the superset: by inverting and transposing the subset, and by adding extra notes. In some cases, there are several ways of doing this. For example, given

[5] Lewin's discussion goes in many directions beyond identification of pitch-class sets. Rather than engaging with Lewin's full analysis, we only take his initial set analysis as an example of what such an analysis might look like.

the subset $\{0;1;2\}$ and the superset $\{0;1;2;3\}$, there are two instances of the subset within the superset: ($\{0,1,2\}$, trans=0, inv=P, extra=$\{3\}$), and ($\{0,1,2\}$, trans=1, inv=P, extra=$\{11\}$). The extra notes are normalized with respect to the transposition, to facilitate comparisons. For example, in the Klavierstück, when looking at different sets from the point of view of whether they are supersets of $\{0,1,2,3,6\}$, we obtain two different PcNs analyzed as ($\{0;1;2;3;6\}$, trans=4, inv=I,extra=$\{7\}$) and ($\{0;1;2;3;6\}$, trans=11, inv=I,extra=$\{7\}$). Since they have the same inversional identity and the same normalized extra notes, we can infer that the supersets themselves are transpositionally related.

Figures 4–8 to 4–10 show all of the PcN sets and supersets with one or two extra notes for $\{0;1;2;3;6\}$. These include some sets not appearing in Lewin's analysis (– he makes no claim to exhaustivity, since he selected a *particular* subset of pitch-class sets based on his interpretation of the score; it may nonetheless be of interest to observe what he omitted, since part of his narrative was about following the incidence of a particular pitch-class set). Since the score is *covered* by plus-two supersets, we haven't shown plus-three supersets (which include some of Lewin's sets). The sets, plus-one, and plus-two supersets together form a sort of "heat map" of the incidence of the set on the score.

In Figures 4–11 to 4–13, we carry out the same "heat map" process for several pitch-class sets on the score – in particular, all of the pitch-class sets of cardinality 3–7 that show up at least twice on the score (with no extra notes). The figures show an abstract of locations of sets and plus-one and plus-two supersets. These figures show that PcNs make available many different points of view, such that many alternatives to Lewin's starting set are available. We make no claims about the relative salience or theoretic interest of all of these different sets; but we offer an efficient and systematic computational method for finding these kinds of structures, such that a large amount of organized pitch-class set information is available for further analytic consideration.

The heat-maps for different sets in Figures 4–11 to 4–13 are related to one another by temporal as well as abstract set-theoretic overlap and inclusion relations. For example, all of the pitch-class sets shown in Figure 4–11 are subsets of {0;1;2;3;6} (Lewin's set), which is shown at the bottom of that figure. Therefore we can take this figure not as a set of alternative points of view, but as *one* picture of how some of the *fragments* of Lewin's set appear on the score.

Polyphones of PcNs offer a means of relating different pitch-class sets with respect to their interaction on the score. A simple way that PcNs are related is by being (temporally) included in one another. Figure 4–14 shows excerpts from the polyphone (holds only; folds not shown) of PcNs on the score; red nodes are normalized {0,1,2,3,6} sets, other colored nodes are those that appear in a hold-relation to these sets *more than once* on the score. This shows us a bit of the pitch-class-set context surrounding the set of interest. The purpose of this figure is to briefly show that we not only have used PcNs to find pitch-class sets on the score, but that their polyphonic relations on the score are also available.

```
                             [m.5]
score:    9e28t9|8e5324|(87)|6|54t|2(3e)|(859)|(1t)45|  ...
P:  [8]   9e28t9|8e             t|2( e)|(859)|( t)
p:  [5i]     8t9|8e5            t| ( e)|(859)|( t)
p6: [11i]        e5324        4 |2(3e)|( 5 )
P6: [2]           5324|(8 )   4 |2(3 )|(85 )
p9: [2i]            2 |(87)|6|5
P8: [4]               4|( 7)|6|54t
```

```
                        [m.10]                         [m.15]
score cont:   |(1t)45|637|4(e8t)0|9(714)853|6e43|2(1t)|8|709|e
P8: [4]           45|6 7|4(  t)
p8: [1i]       (1 )45|6 7|4      ; (714) 5 |6
P9: [5]             5|6 7| (e8 ) ; (7  )85 |6e
P1: [9]                3 | (e t)0|9
P2: [10]                   (e t)0| ( 14)
p6: [11i]                                 53| e43|2
p5: [10i]                                      43|2(1t)
Pe: [7]                                            (1t)|8|7 9|
p2: [7i]                                           (1t)| |70 |e
```

Figure 4–7: Lewin's selection of {0;1;2;3;6} sets from Stockhausen's Klavierstück III (copied from [Lewin2007]). The top and middle rows represent the score with numbers as pitch-classes (t = 10 and e = 11); parentheses enclose simultaneous onsets. Notations in square brackets give the transposition and inversion with relation to the normal form – e.g. Lewin's "P" is ($\{0;1;2;3;6\}$, trans=8, inv=P), notated as [8]; Lewin's (lowercase) "p" is ($\{0;1;2;3;6\}$, trans=5, inv=I), notated as [5i].

Figure 4–8: PcNs on Stockhausen's Klavierstück III with Forte-normal form {0,1,2,3,6} (including inversions). The score as notated here is a rhythmically simplified version of Stockhausen's score, serving as a rough background map for the sketch of pitch-class set locations depicted here and in the following figures.

:i=1, :norm=c(0–3, 6), :trans=8, :inv=P,
:i=2, :norm=c(0–3, 6), :trans=5, :inv=I,
:i=3, :norm=c(0–3, 6), :trans=11, :inv=I,
:i=4, :norm=c(0–3, 6), :trans=10, :inv=I

Figure 4–9: PcNs that are plus-one supersets of {0,1,2,3,6} sets. Some sets are supersets of {0,1,2,3,6} in two different ways. This figure includes supersets of those sets found in Figure 4–8.

:i=1, :forte_subs=((:norm=c(0–3, 6),:trans=8,:inv=P,:extra=c(9)), (:norm=c(0–3, 6),:trans=5,:inv=I,:extra=c(9))),
:i=2, :forte_subs=(:norm=c(0–3, 6),:trans=5,:inv=I,:extra=c(10))),
:i=3, :forte_subs=((:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(9)), (:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(9))),
:i=4, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(5)),
:i=5, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=I,:extra=c(2)),
:i=6, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=P,:extra=c(4)),
:i=7, :forte_subs=(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(11)),
:i=8, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=I,:extra=c(7)),
:i=9, :forte_subs=(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(7)),
:i=10, :forte_subs=(:norm=c(0–3, 6),:trans=10,:inv=I,:extra=c(1)),
:i=11, :forte_subs=(:norm=c(0–3, 6),:trans=10,:inv=I,:extra=c(10)),
:i=12, :forte_subs=(:norm=c(0–3, 6),:trans=7,:inv=P,:extra=c(5))

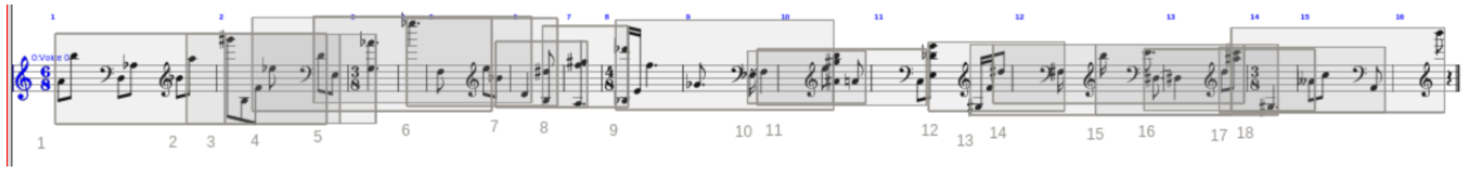

Figure 4–10: PcNs on Stockhausen's Klavierstück III which have a subset of {0,1,2,3,6} plus two extra pitch-classes. These include supersets of those sets found in Figures 4–8 and 4–9.

:i=1, :forte_subs=(:norm=c(0–3, 6),:trans=8,:inv=P,:extra=c(7, 9)),(:norm=c(0–3, 6),:trans=5,:inv=I,:extra=c(9–10)),
:i=2, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(7, 9)),(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(9–10)),
:i=3, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(5, 9)),(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(8–9)),
:i=4, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(4–5)),(:norm=c(0–3, 6),:trans=2,:inv=I,:extra=c(1–2)),
:i=5, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=P,:extra=c(4, 10)),(:norm=c(0–3, 6),:trans=2,:inv=I,:extra=c(2, 8)),
:i=6, :forte_subs=(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(7, 11)),
:i=7, :forte_subs=(:norm=c(0–3, 6),:trans=8,:inv=P,:extra=c(7, 9)),(:norm=c(0–3, 6),:trans=5,:inv=I,:extra=c(9–10)),
:i=8, :forte_subs=(:norm=c(0–3, 6),:trans=5,:inv=I,:extra=c(8, 10)),
:i=9, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=P,:extra=c(9, 11)),(:norm=c(0–3, 6),:trans=1,:inv=I,:extra=c(2, 9)),
:i=10, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=I,:extra=c(7, 11)),
:i=11, :forte_subs=(:norm=c(0–3, 6),:trans=4,:inv=I,:extra=c(7–8)),
:i=12, :forte_subs=(:norm=c(0–3, 6),:trans=1,:inv=I,:extra=c(2, 7)),
:i=13, :forte_subs=(:norm=c(0–3, 6),:trans=2,:inv=P,:extra=c(4, 9)),(:norm=c(0–3, 6),:trans=11,:inv=I,:extra=c(7, 9)),
:i=14, :forte_subs=(:norm=c(0–3, 6),:trans=10,:inv=I,:extra=c(1, 8)),
:i=15, :forte_subs=(:norm=c(0–3, 6),:trans=10,:inv=I,:extra=c(1, 10)),
:i=16, :forte_subs=(:norm=c(0–3, 6),:trans=1,:inv=P,:extra=c(7, 9)),(:norm=c(0–3, 6),:trans=10,:inv=I,:extra=c(9–10)),
:i=17, :forte_subs=(:norm=c(0–3, 6),:trans=7,:inv=P,:extra=c(5, 7)),
:i=18, :forte_subs=(:norm=c(0–3, 6),:trans=7,:inv=P,:extra=c(4–5)),(:norm=c(0–3, 6),:trans=7,:inv=I,:extra=c(1–2)))

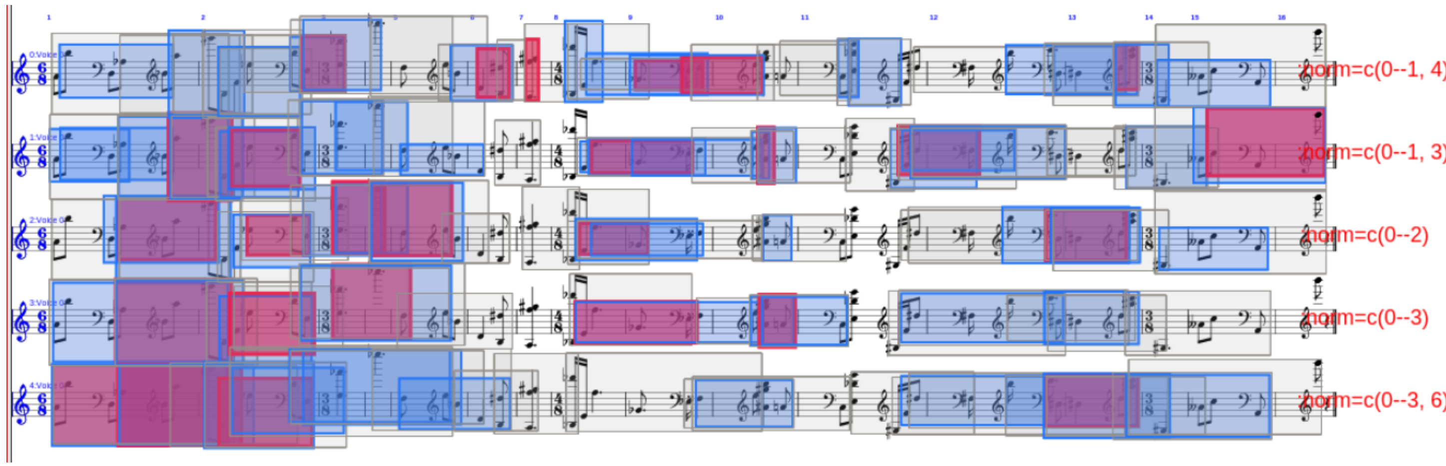

Figure 4–11: These figures show plus-zero, plus-one, and plus-two supersets of all normalized pitch-class sets of cardinality 3–7 which appear in plus-zero form at least three times on the score. The bottom staff is Lewin's set. (1 of 3)

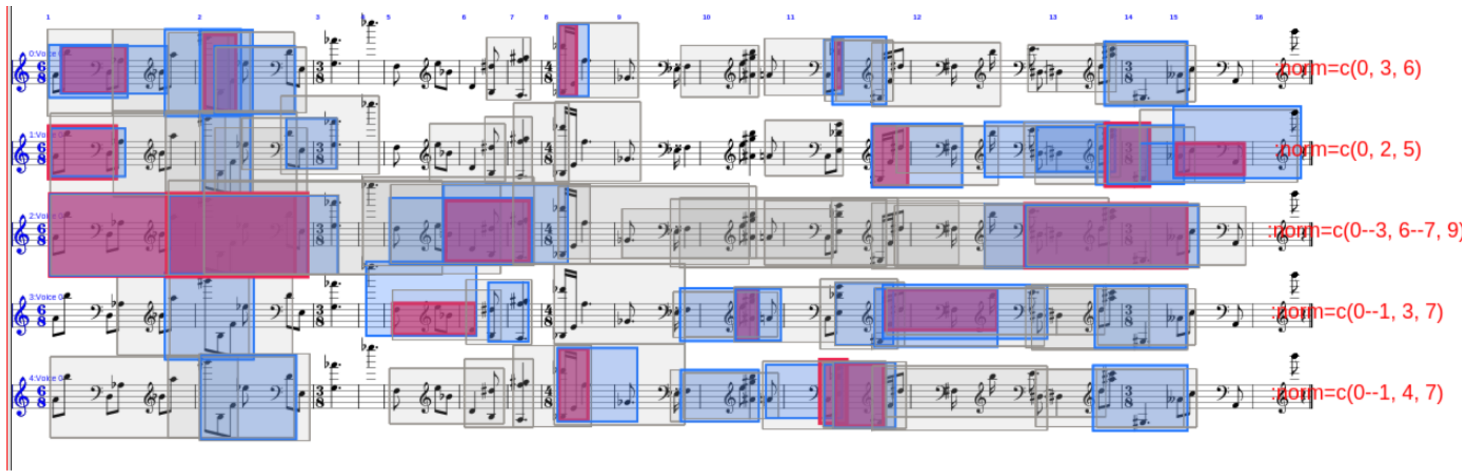

Figure 4–12: These figures show plus-zero, plus-one, and plus-two supersets of all normalized pitch-class sets of cardinality 3–7 which appear in plus-zero form at least three times on the score. (2 of 3)

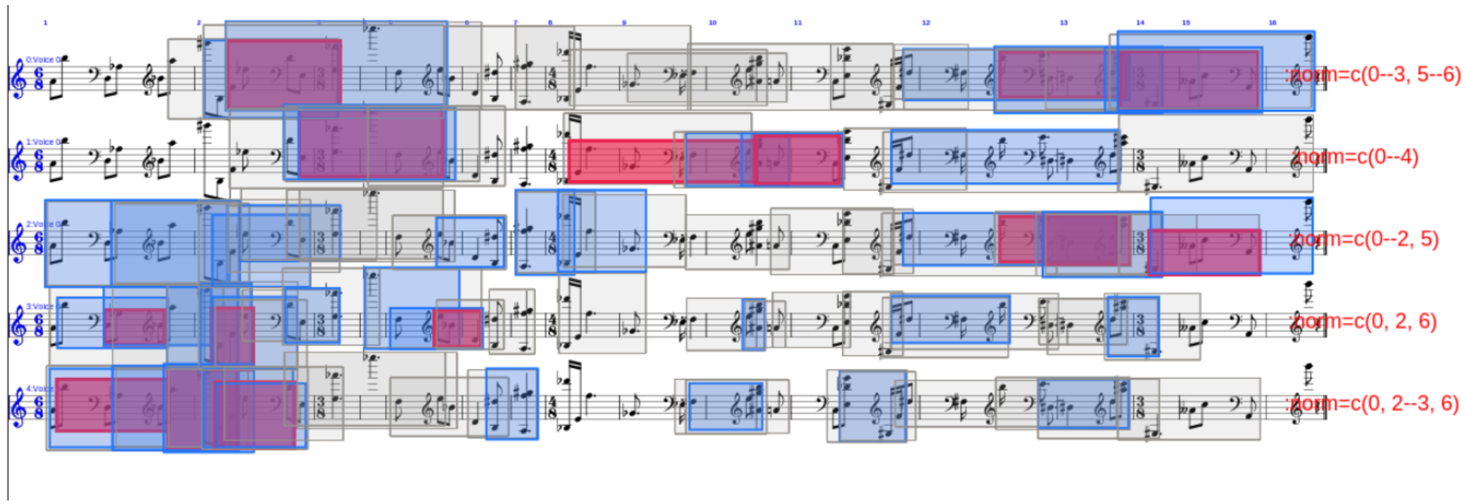

Figure 4–13: These figures show plus-zero, plus-one, and plus-two supersets of all normalized pitch-class sets of cardinality 3–7 which appear in plus-zero form at least three times on the score. (3 of 3)

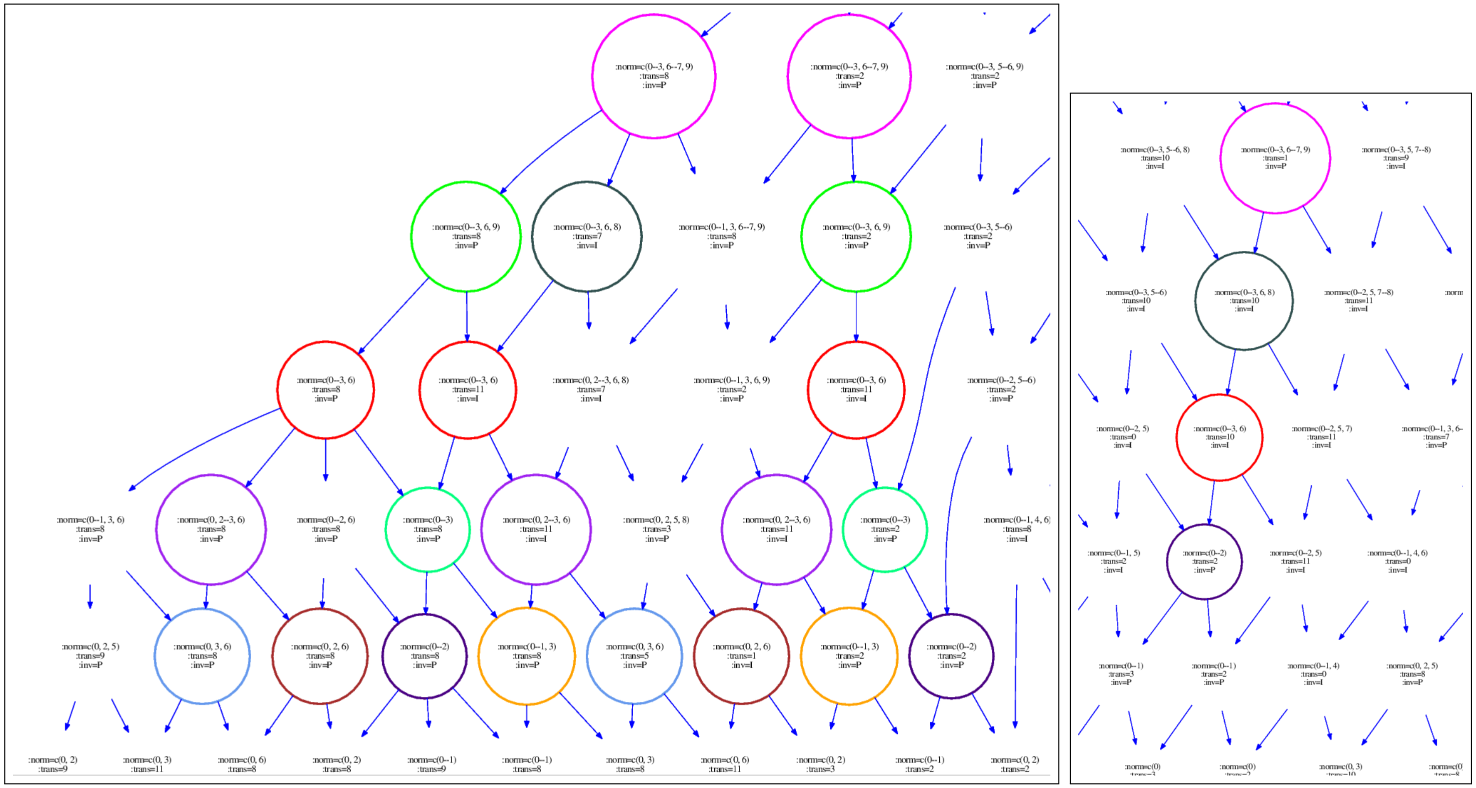

Figure 4–14: Excerpts form the polyphone (holds only; folds not shown) of PcNs on Stockhausen's Klavierstück III; red nodes are normalized {0,1,2,3,6} sets, other colored nodes are those that appear in a hold-relation to these sets *more than once* on the score.

4.4.3 Stravinsky: Augurs of Spring

Allen Forte's *The Harmonic Structure of the Rite of Spring* is a detailed account of the most important pitch-class sets on that score and how they relate to each other, structuring harmonic and melodic material. This is not a book that one would *read* – it's a book that demands that its reader have a few scores of *The Rite of Spring* (both the orchestral score and piano reduction), some colored pencils, and a piano. Or, a computer system that can find pitch-class sets, organize and display them from different perspectives, and play them back.

PcNs generate a large number of these kinds of points of view, which can be interactively explored – but this document is constrained (like Forte's book) to show a small amount of static information. It's hard to generate a small number of low-dimensional projections that effectively summarize the amount of structural information generated by a PcN analysis on a score, and so the flexible, programmable context is an essential aspect of the medium – rather than generating an analytic picture, PcNs are part of an *environment* for exploration.

PcNs could be used to follow Forte's text, or to script a curated tour through some of his main points, or to produce a new curated analysis of any score. They can also be used more casually to take a quick, impressionistic look at a score in order to see generalities about its organization.

The illustrations in this section offer zoomed-out, informal impressions that nonetheless give intuition into harmonic and formal aspects of this score, showing the utility of pitch-class sets and PcNs. We show a few pictures of *The Augurs of Spring*, the second movement of the *Rite*. Each panel of Figures 4–15–4–18 shows all of the PcNs for a given Forte-normalized set, with the transpositions of the prime form in blue and transpositions of the inverted form in green. Twelve different sets are shown, out of a total of 185 such pictures generated for pitch-class sets of cardinality 3–8.

In Figure 4–15, observe that PcNs with the same set seem to cluster in temporal zones, designating formal sections by their harmonic material. In Figure 4–16, we can see the "evolution" of a pitch-class set (i.e. the use of three similar sets) over part of the score (mm. $\sim$120–150). Figures 4–17 and 4–18 show a few other sets, showing partial coincidences of related sets.

Figures 4–19, 4–20, and 4–21 show some of the same sets as the previous figures, this time differentiated by their non-normalized pitch-class set. This gives a view of *how* each set is being used: often one version of the set is predominant, but there is one section in particular which is less harmonically static, using multiple versions of the same sets in close proximity.

From this brief, impressionistic point of view, we can surmise that the disposition of pitch-class sets on the *Augurs of Spring* is structure-bearing, and that the repetition and emphasis of pitch-class sets predicts that they strongly inform qualitative harmonic character and contrast of different sections.

The *Rite of Spring* is an *a*tonal work, in the technical sense that it does not use the system of harmony and *keys* used in tonal music. A casual glance at PcNs in a score can quickly show whether or not a score is *tonal*. For comparison with the PcN organization of the *Augurs of Spring*, we show a few pictures of two tonal scores: Mozart's String Quartet Op. 90 (mvt. I), and Verklärte Nacht by Schoenberg

(Figures 4–22–4–27). Although tonal scores often contain *some* differentiable zones of pitch-class sets, they are also covered or almost covered by a few *specific* sets related to the major and minor keys. Another approach to key-related sets is discussed in the next section.

Flexibility of PcNs

Forte's analysis of the *Rite of Spring*, like most other pitch-class set analyses, does not strictly employ temporal sets as we have defined them (i.e. *all* of the pitch-classes with onsets during a particular timespan). For instance, Forte separates out melodic lines in the different instrumental parts. Because the PcN or N-set algorithm is part of a programmable analysis environment, it can be run on different subscores, and we can use different approaches to finding pitch-class sets on a score. For example, we can run the PcN algorithm separately on the different voices (or groups of voices), and compare the results. Figure 4–28 shows three different sets of PcNs on three different instruments on a section of the *Augurs of Spring*.

In the PcN algorithm as described above, if a long note is sustaining it will be included in all sets occurring during its duration. Supposing we want to include *onsets only*, so that a long pedal tone doesn't get included in all harmonies: instead of altering the PcN algorithm itself, we simply have to define the *structures* upon which the algorithm is being run as the *onsets* of the notes rather than their entire durations.

Another way of making PcNs is to break a score into subscores (e.g. with a heuristic such as splitting at long-enough rests), and perform the PcN algorithm on each section separately so that e.g. if a voice plays "ABC (long rest) DEF", we won't obtain a segment "C (long rest) D" as a PcN.

While the function of PcNs on all of the notes in the score is a useful harmonic sketch, this is just one application of N-sets – a general structuring method. The contextual flexibility of N-sets (and the principle of structural relativity) means that their work is never done.

4.5 Tonal Pitch-Structure: Key Assertions[6]

Tonality is a complex and flexible musical phenomenon in which pitch-classes take on particular, asymmetric *relations* to one another within a dynamic musical context. A *key* is a particular set of such relations.

Furthermore, a *key* involves a distinction between pitch-classes that are "in" the key, and those that are not. This distinction is not straightforward, however, since music in a given key can "alter" its notes, "borrow" from or "tonicize" other keys, or "ornament" with out-of-key notes, while still being considered (music-theoretically and arguably cognitively) to be *in* the given key.

Key induction – determining the keys of a score – is a much-researched topic, since it is first step toward a more detailed harmonic and formal analysis, as well as being of interest in music cognition and having practical applications (e.g. music transcription).

In this section, we give an overview of how PcNs can be used to gain detailed information about the keys of a tonal score. This is done through a heuristic *reduction*

[6] Part of this section is adapted and expanded from [HandelmanSigler2013].

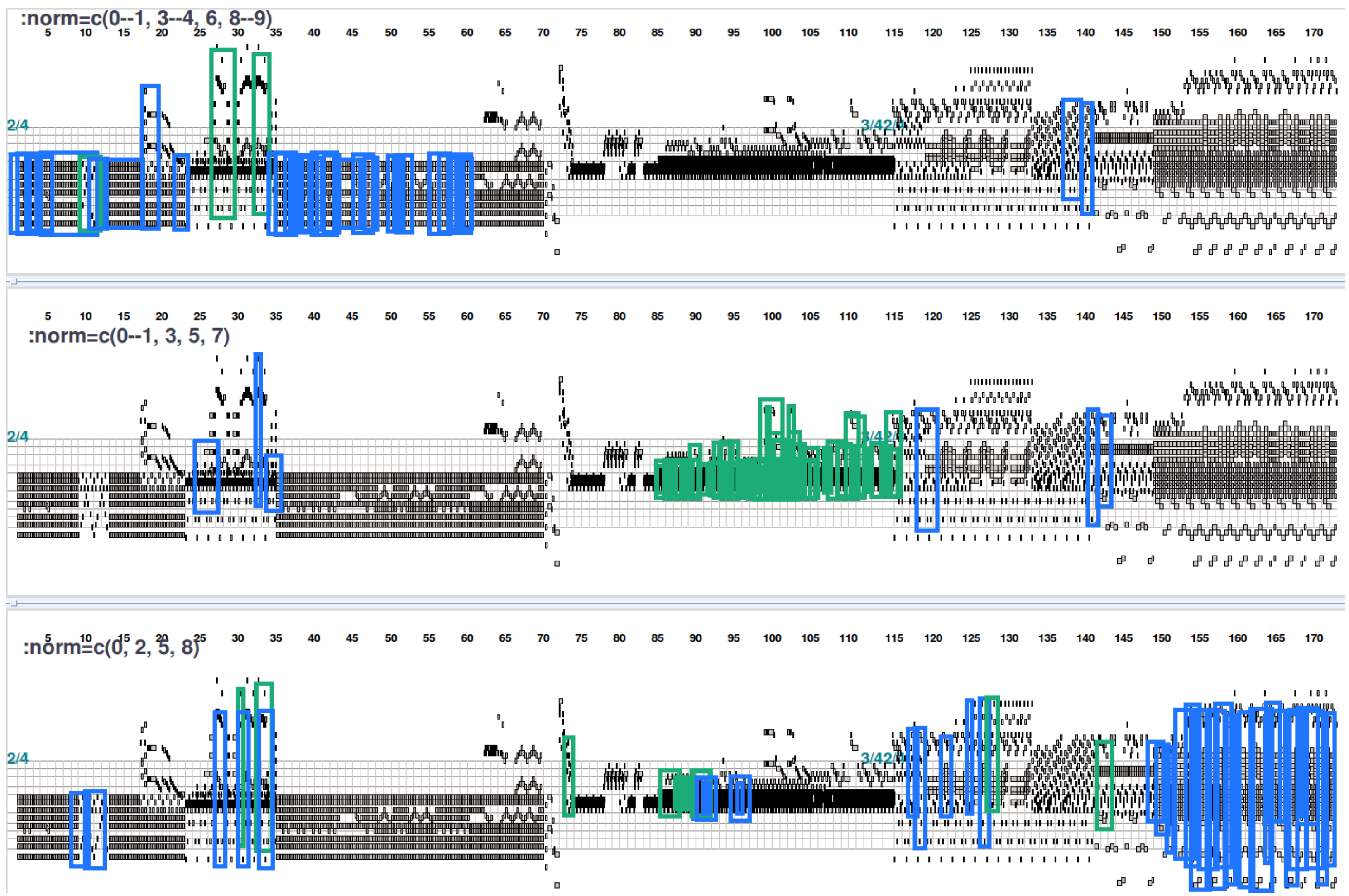

Figure 4–15: Stravinsky, *Augurs of Spring*. PcNs showing sectional zoning. {0,1,3,4,6,8,9}, appearing prominently at the beginning of the score, is Forte's "Augurs chord." Blue boxes are "prime" form (any transposition) and green boxes are inverted form.

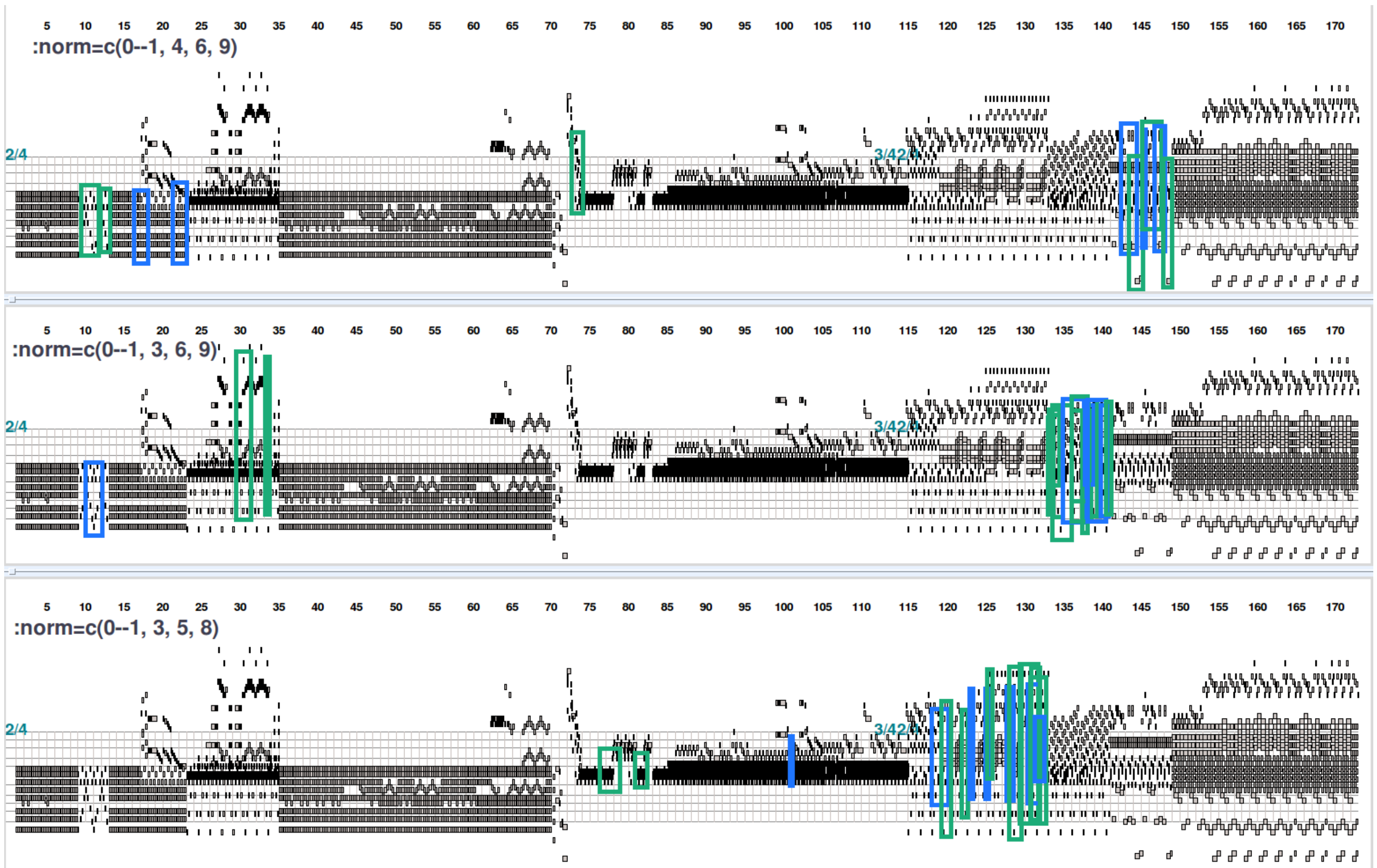

Figure 4–16: Stravinsky, *Augurs of Spring*. Pitch-class sets (PcNs) clustered and “evolving” over the course of a section (mm 120-150)

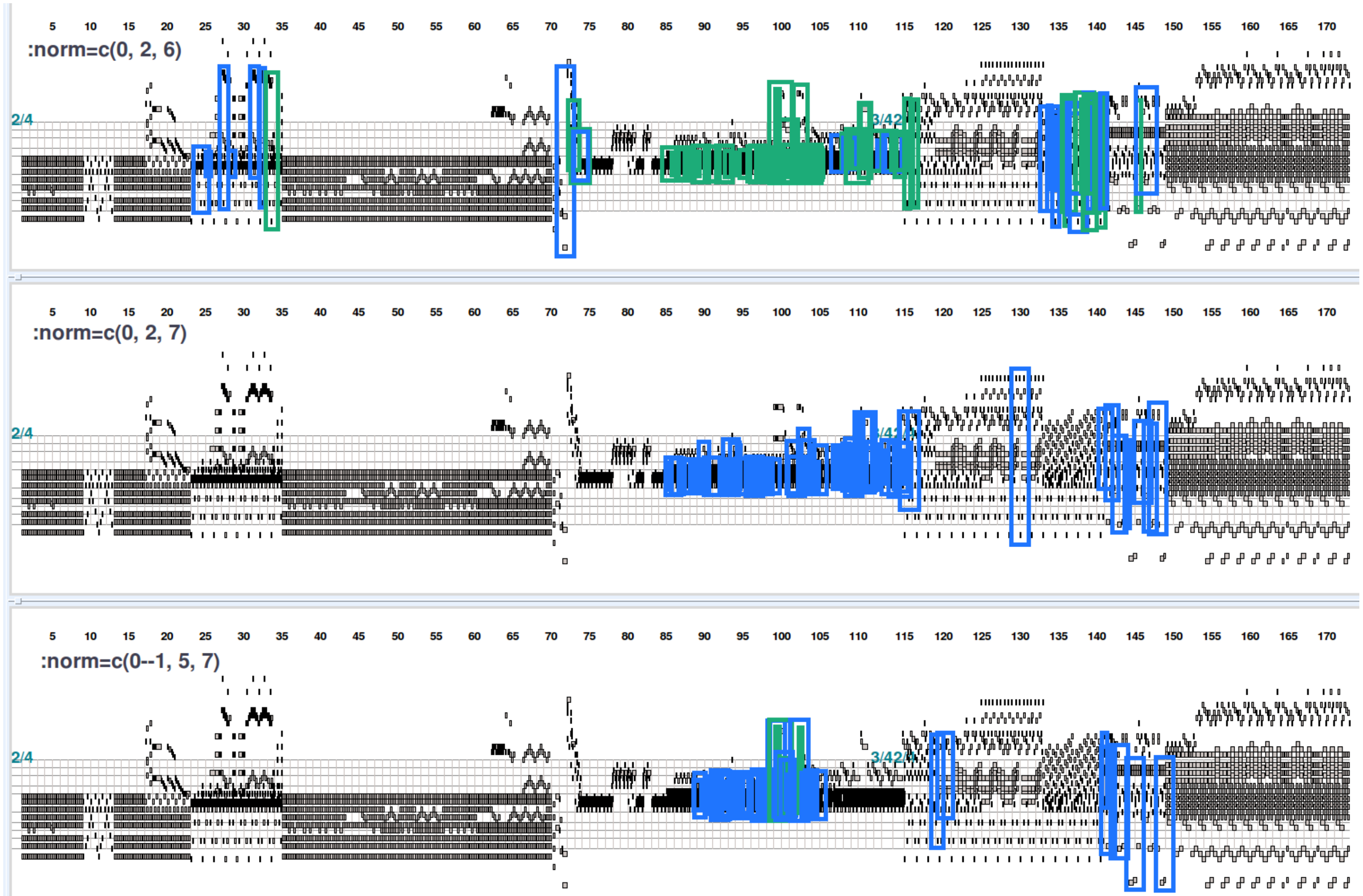

Figure 4–17: Stravinsky, *Augurs of Spring*. Related pitch-class sets (PcNs) sometimes coincident, other times not.

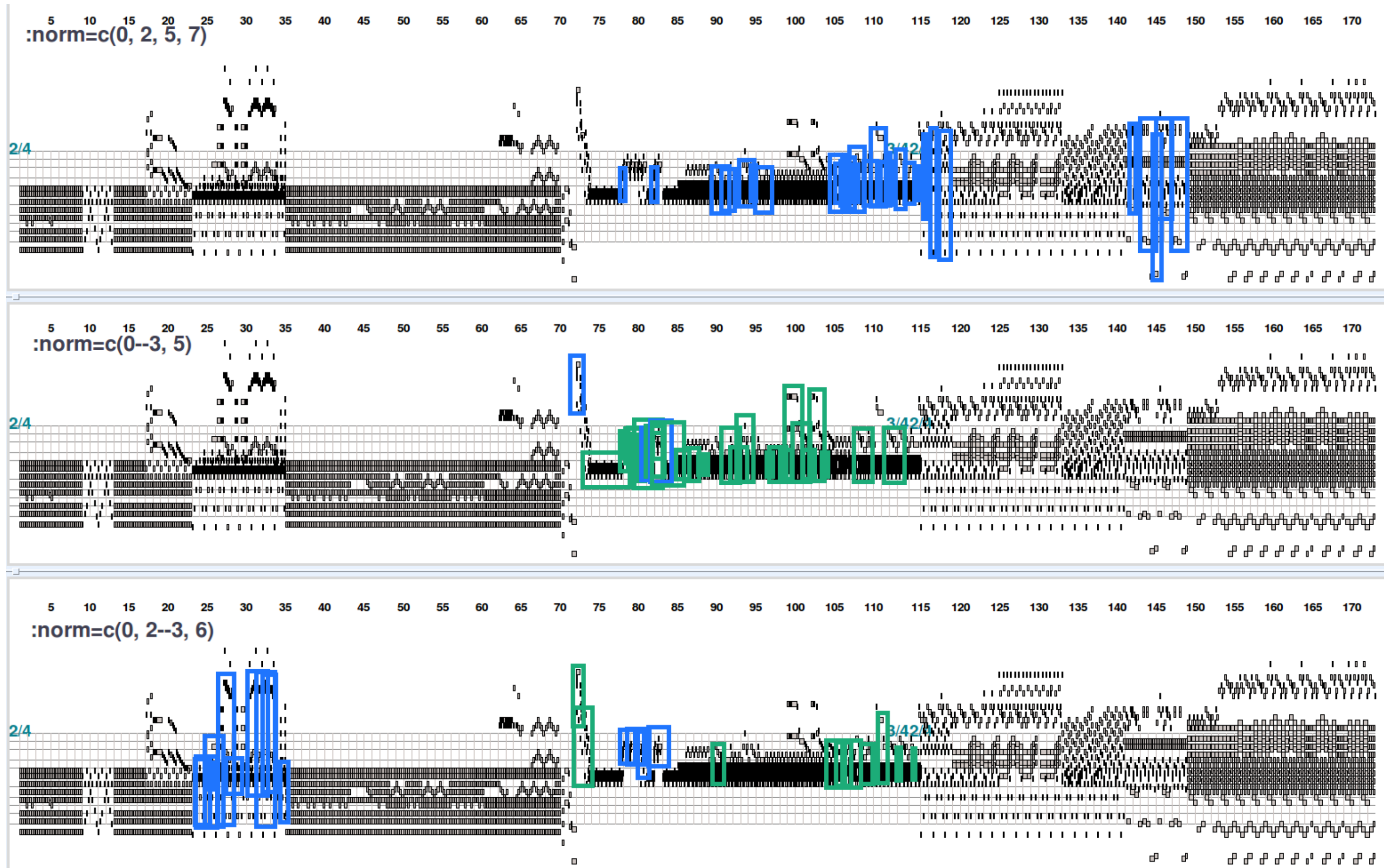

Figure 4–18: Stravinsky, *Augurs of Spring.* More PcNs.

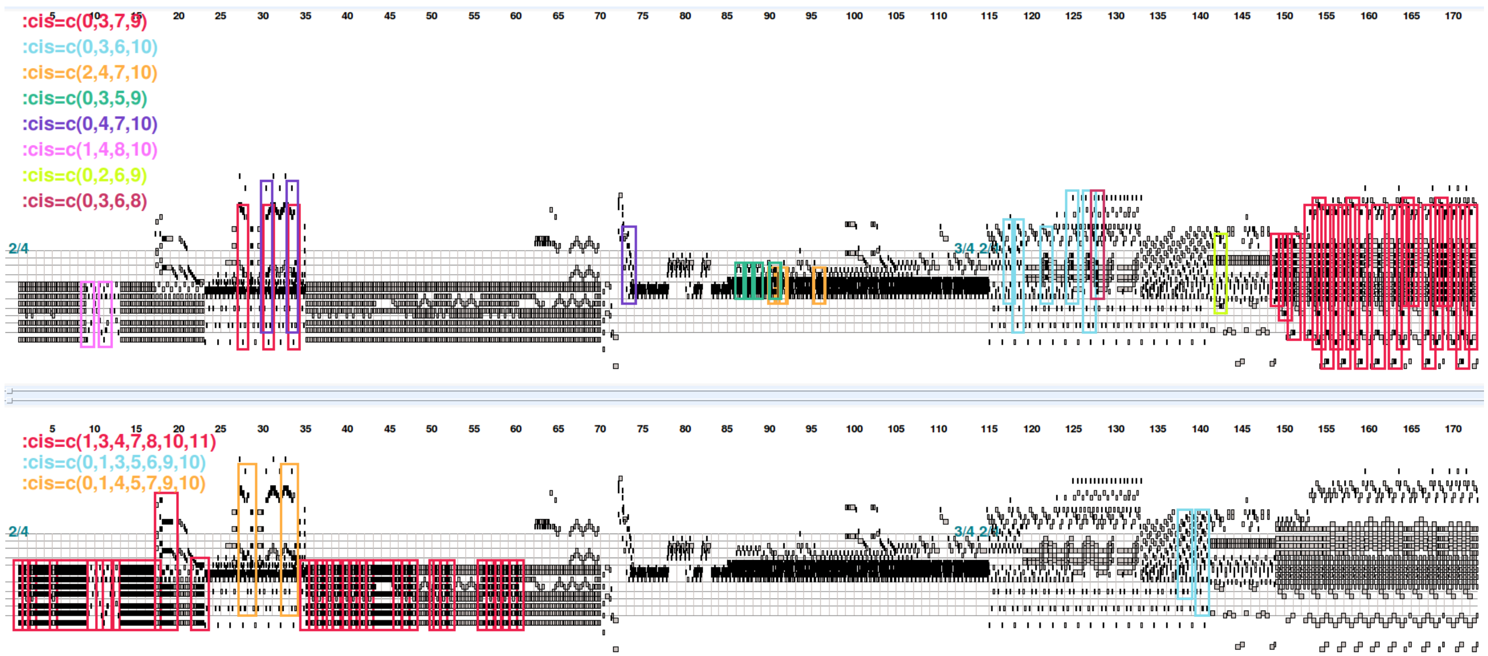

Figure 4–19: Stravinsky, *Augurs of Spring*. Each panel shows PcNs of the same normalized form, differentiated by non-normalized pitch-class set. The normalized sets are {0,2,5,8} and {0,1,3,4,6,8,9} (shown also in Figure 4–15). The latter set is Forte's "Augurs chord." The larger set occurs predominantly in one form (in red) at the beginning of the score, and the smaller set occurs in one consistent form at the end of the score (also in red). Although the smaller set is a *normalized* subset of the larger, the red *version* of the smaller set is *not* a subset of red version of the larger set.

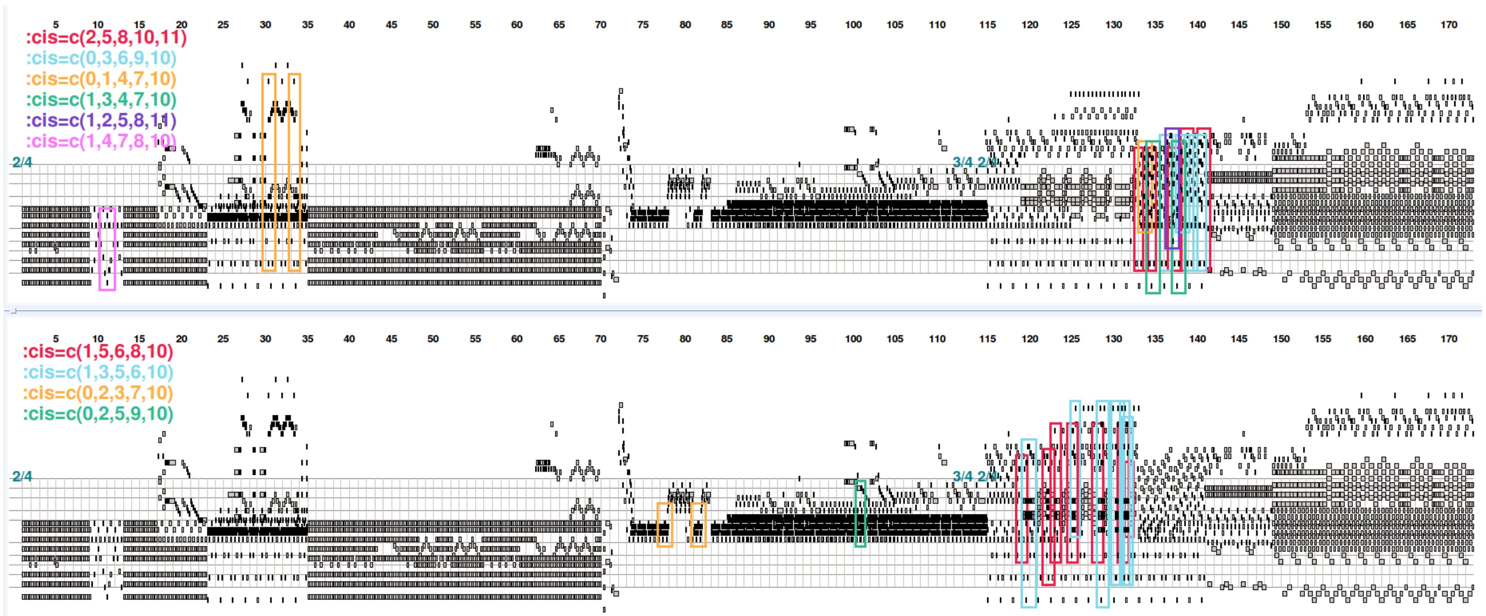

Figure 4–20: Stravinsky, *Augurs of Spring*. Each panel shows PcNs of the same normalized form, differentiated by non-normalized pitch-class set. The normalized sets are {0,2,3,6,9} and {0,1,3,5,8} (shown also in Figure 4–16) The cluster of PcNs in the top panel is made of five different versions of the normalized set, while the cluster in the second panel is made of two versions.

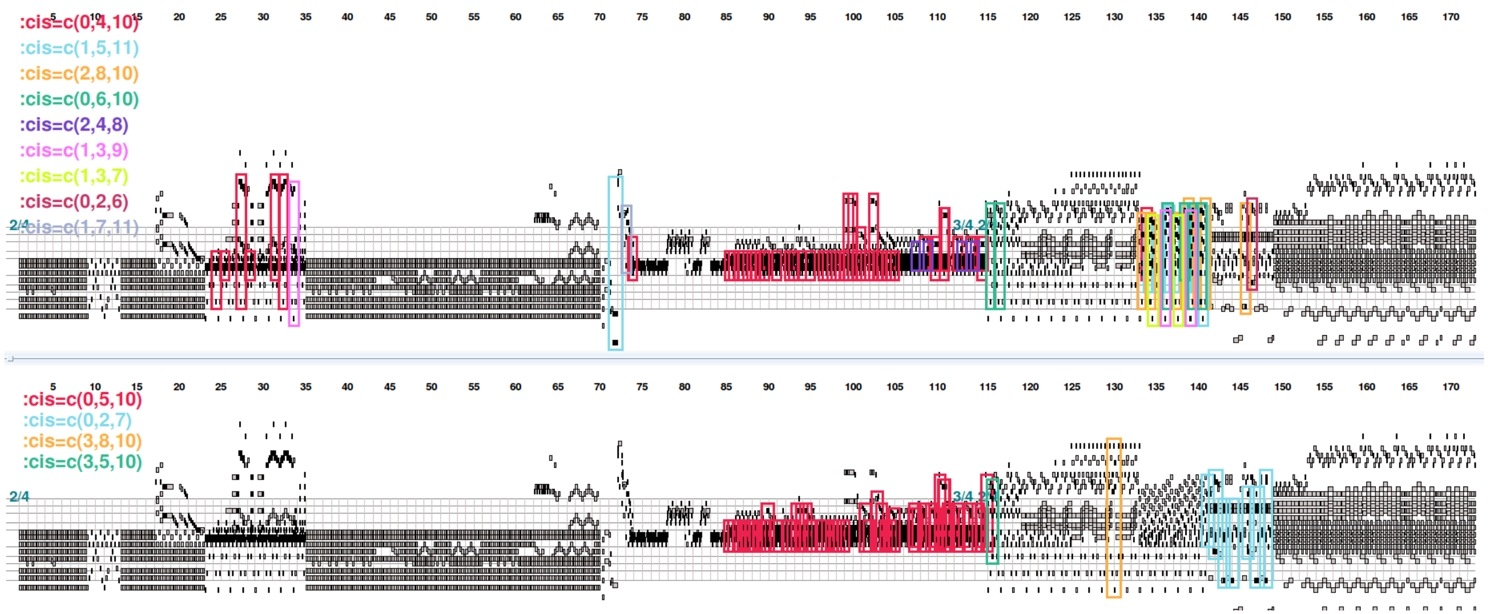

Figure 4–21: Stravinsky, *Augurs of Spring*. Each panel shows PcNs of the same normalized form, differentiated by non-normalized pitch-class set. The normalized sets are {0,2,6} and {0,2,7}. Both occur predominantly in one (red) form around the middle of the piece – though there are also some purple versions of the first set around measure 110. The last (large) cluster in the first panel is made of six versions of its set, and is followed by a cluster of only one version of the other set, on the second panel. A cluster of the same set in different color versions means the same *kind* of harmony or sonority in variety or transformation, while the same version means literal repetition of the same pitch-content.

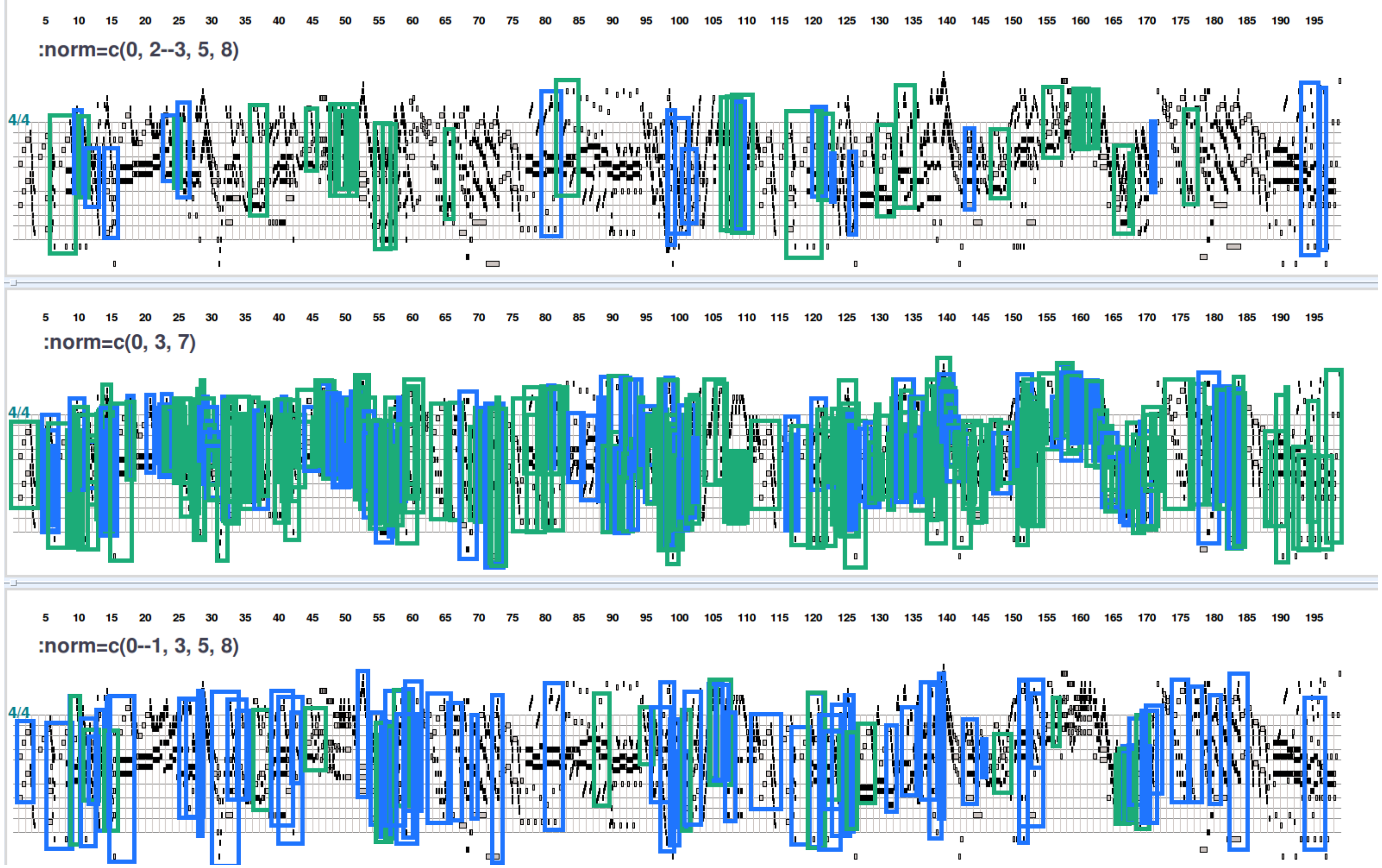

Figure 4–22: Mozart, String Quartet K.590, Mvt 1. Tonal works are pervaded by key-related pitch-class sets. Blue boxes are uninverted forms; green boxes are inversions (– on the middle panel, green boxes are major triads and blue boxes are minor triads).

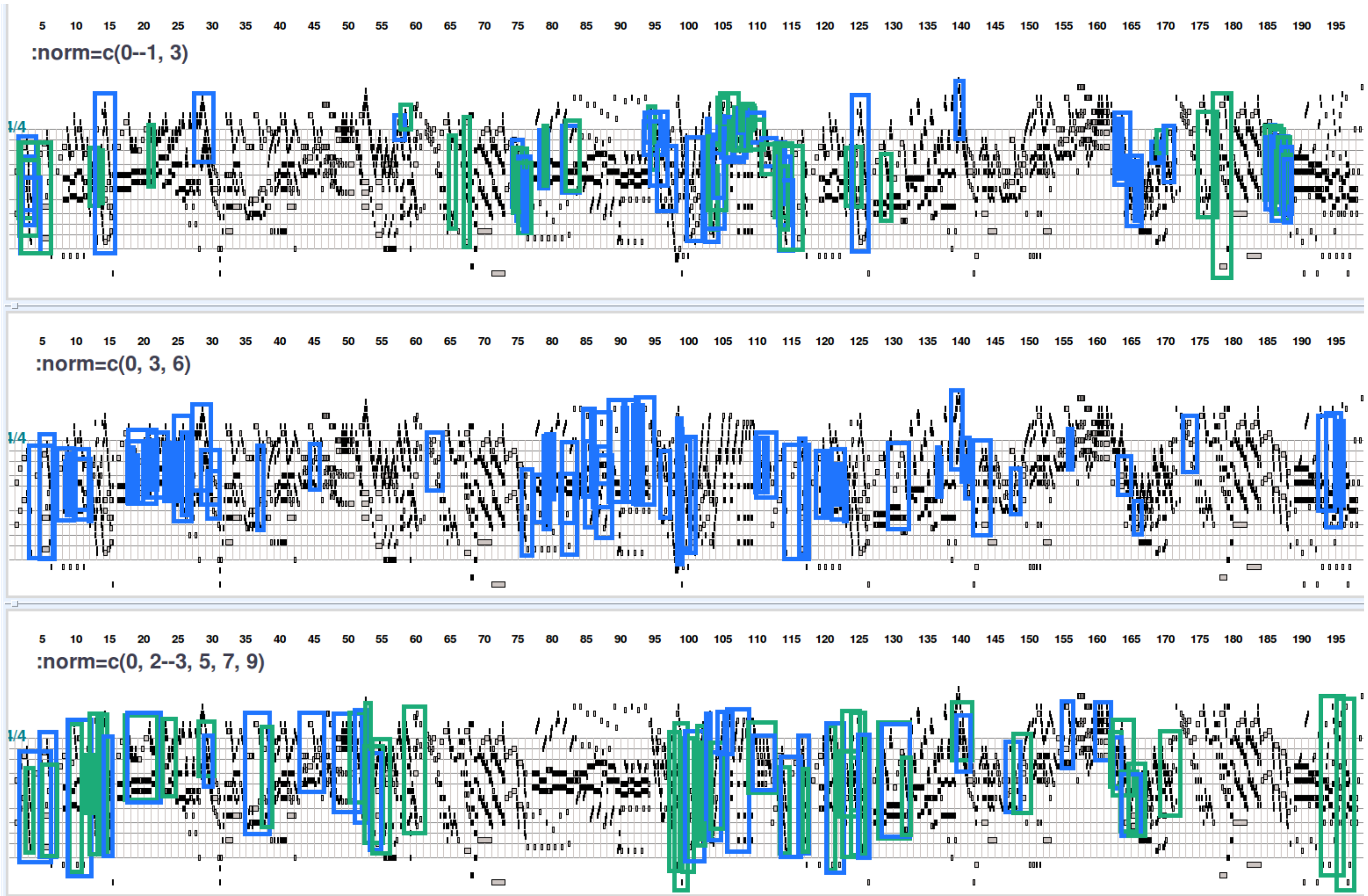

Figure 4–23: Mozart, String Quartet K.590, Mvt 1. Some key-related PcNs offer sectional differentiation.

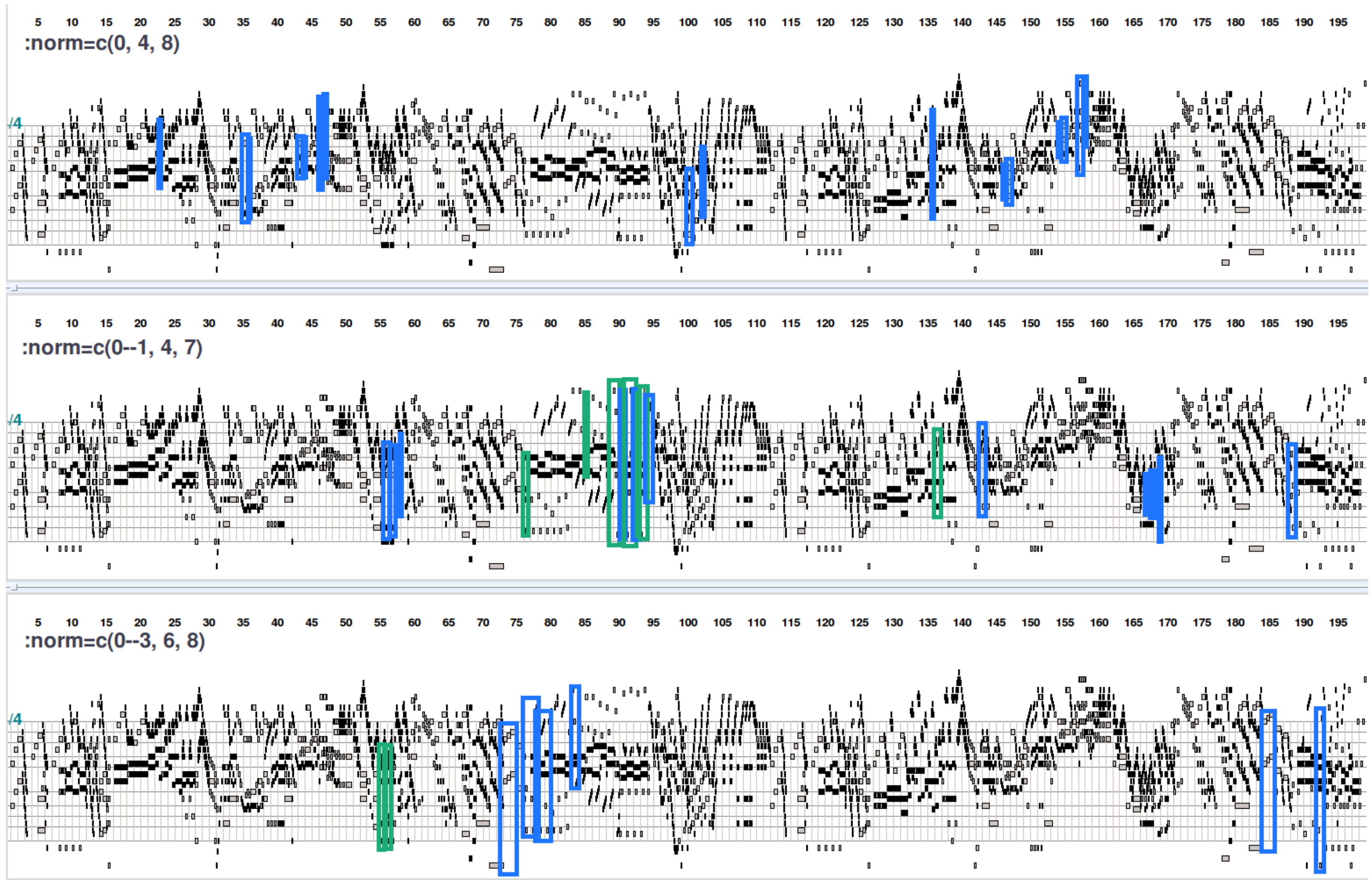

Figure 4–24: Mozart, String Quartet K.590, Mvt 1. A view of some less pervasive sets.

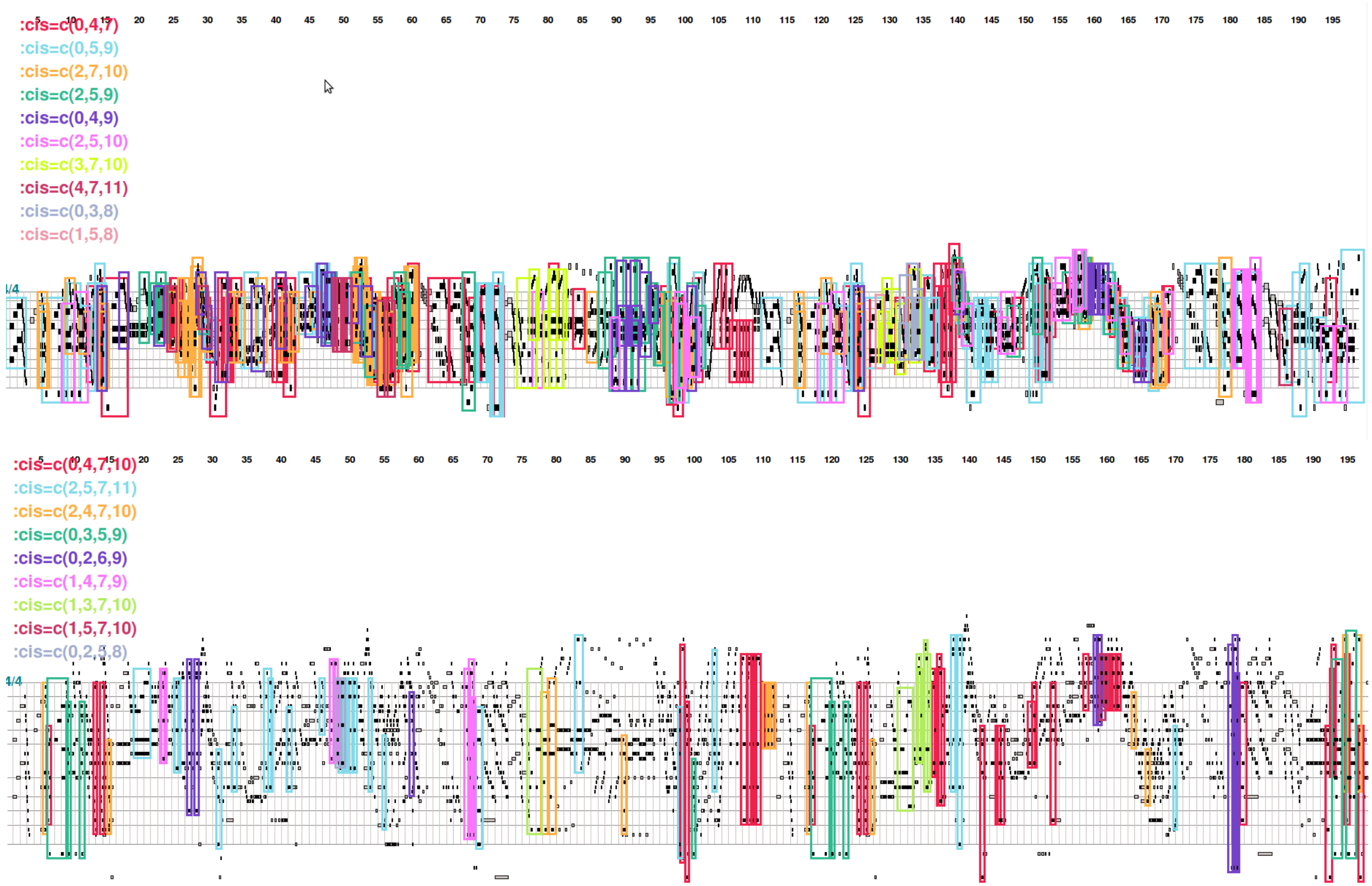

Figure 4–25: Mozart, String Quartet K.590, Mvt 1. Key-related sets {0,3,7} (major / minor triads) and {0,2,5,8} ("dominant" seventh and half-diminished seventh chords), differentiated by their non-normalized versions.

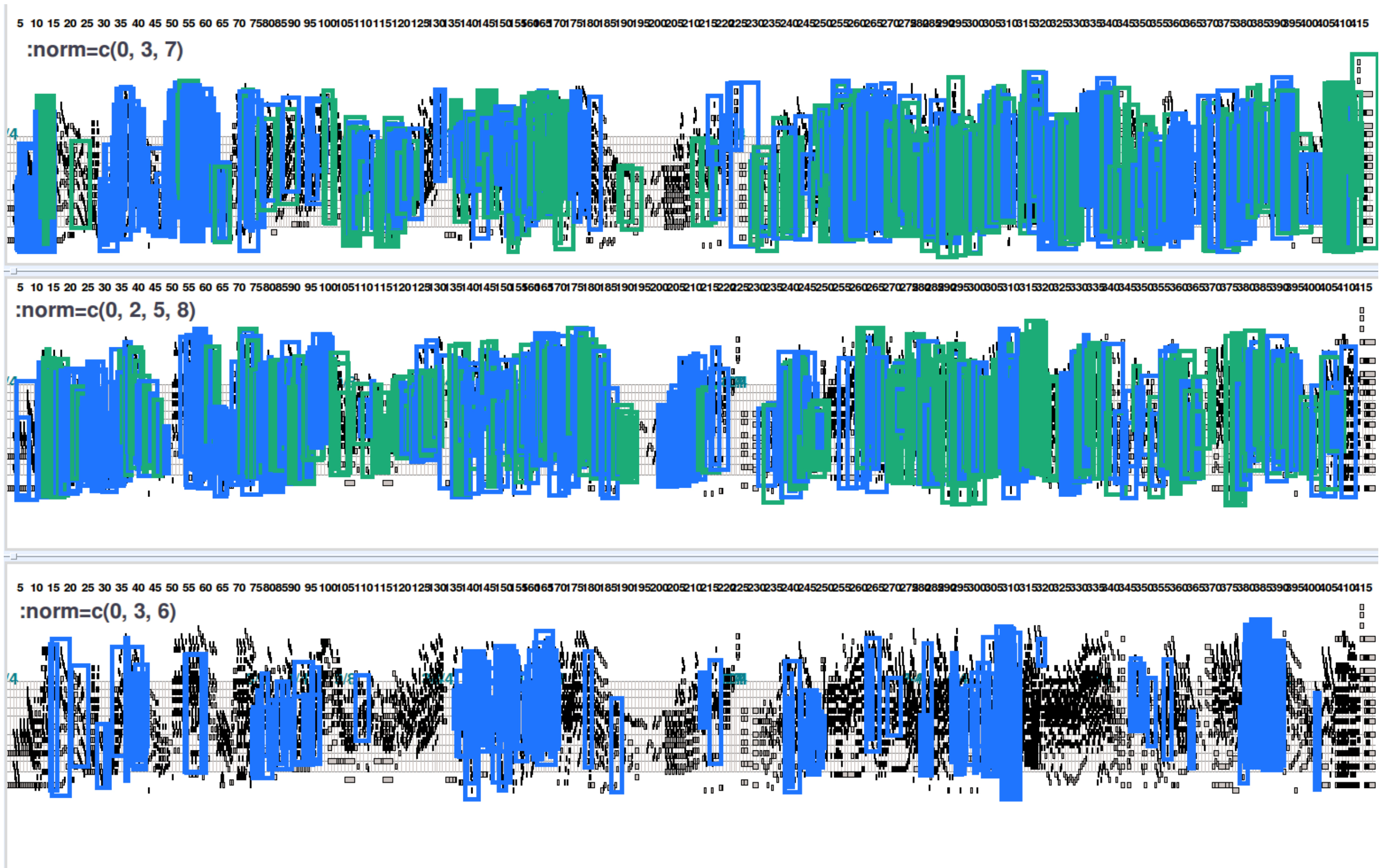

Figure 4–26: Schoenberg, *Verklärte Nacht*. A tonal score, pervaded by key-related pitch-class sets.

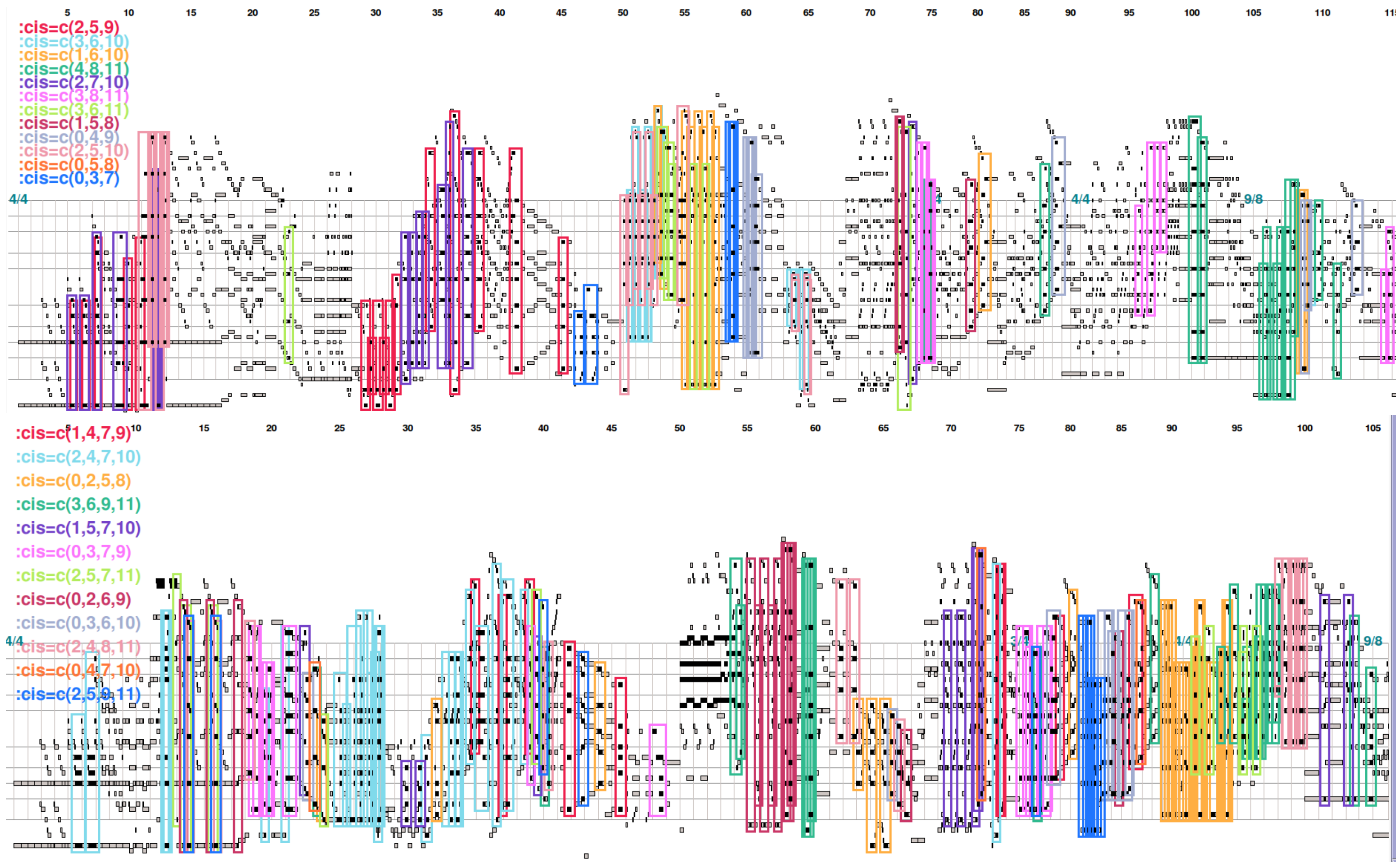

Figure 4–27: Schoenberg, *Verklärte Nacht* (excerpt). Key-related sets {0,3,7} (major / minor triads) and {0,2,5,8} (“dominant” seventh and half-diminished seventh chords), differentiated by non-normalized versions.

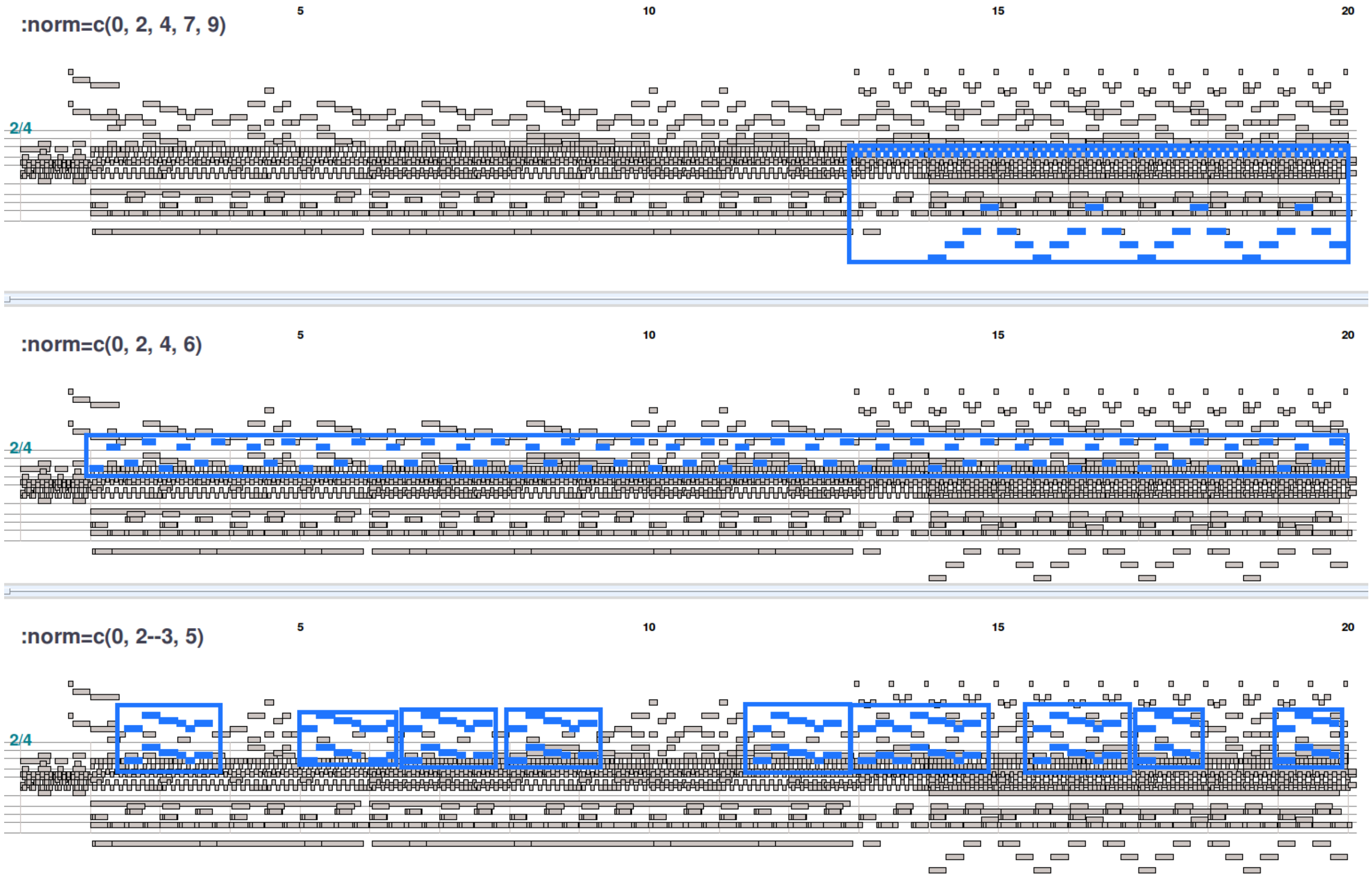

Figure 4–28: Stravinsky, *Augurs of Spring* mm. 116-135. PcNs in three different voices: violas, oboes, and horns.

of the set of PcNs to a subset of pitch-class sets hypothesized to be informative about the semantic notion of *key*. The particular set of pitch-class sets we take to be of interest is one hypothesis among potentially many – for instance another analysis might focus on particular harmonic progressions rather than diatonic key areas. The purpose of this section is to show what it looks like when we *start* with a structural analysis (PcNs), and push these toward a more traditional music-theoretic view. This affords a comparison with an existing literature of computational methods for locating key areas on a score and a discussion of the mathematical and empirical assumptions inherent in different kinds of computational analysis.

4.5.1 Key-Asserting PcNs

In this experiment, we define a set of "key-asserting" PcNs based on set-theoretic relations to the diatonic sets. We define the major and minor keys by their pitch-class sets. C or 0 major is {0,2,4,5,7,9,11}, and C or 0 minor is {0,2,3,5,7,8,11}.[7] The other 11 major and 11 minor keys are transpositions (translations) modulo 12.

If a pitch-class set is a subset of *exactly one* key, then it "asserts" that key. If a pitch-class set is a subset of exactly *two* keys, it asserts the intersection of those keys:

[7] The major set is unproblematically defined, but minor keys are more complex. The set we have selected for this definition, called "harmonic minor," was selected for its maximal intervallic differentiation from the major set. The "natural" minor (and other "modes" or rotations of the major set) are not distinguished from the major by pitch-class sets, so any set-based assertion of a "major" key subsumes these possibilities, which require a different structural approach to be distinguished. The ascending upper tetrachord of the "melodic" minor is likewise a subset of the major set.

a double-key assertion. There are also triple-key (etc.) assertions, as well as other set-theoretic relations describing e.g. intersection or superset relations to keys, but for this study we use only single and double assertions, making a (heuristic) *selection* of PcNs that are the *most* unambiguously telling about diatonic key areas, rather than creating a key-related label for every PcN.

Starting from the major and minor generating sets above, there are 39 (normalized) single-key asserting sets (12 for each major and 27 for each minor key), and 23 double-key asserting sets.

We add to these the set of pitch-class sets that are subsets of the minor set *plus* the major sixth, in such a way that the set cannot be interpreted as a plus-one superset of a subset of any other key – there are 6 such sets for each minor key (– this helps cover parts of the "melodic" version of the minor scale.)

Tables showing all of these sets are in the appendix to this chapter. There are a total of 351 normalized pitch class sets; 283 of these are *not* key-asserting under this model.

Double-key assertions

Double-key pitch-class sets are defined as being a subset of exactly two keys. There are not a Cartesian number of double-keys, because not all intersections are *only* included in two keys. There are five intersections that yield double-keys – (0 maj, 9 min), (0 maj, 0 min), (0 min, 9 min), (0 maj, 5 maj), and (0 min, 4 min). Each of these has a dual – except (0 maj, 0 min) which is its own dual – so that 0 major can also be paired with 7 major and 3 major, and 0 min can also be paired with 3 minor, 3 major, and 8 major.

These relations have music-theoretic names: (0 maj, 0 min) is *parallel* major/minor; (0 maj, 9 min) is *relative* major/minor, (0 maj,5 maj) is the (major) subdominant/dominant. (0 min, 3 min) is the parallel minor of the relative major, and its dual is the relative minor of the parallel major. (0 min, 4 min) is the relative minor of the dominant, its dual is the relative minor of the subdominant. *Minor* subdominants and dominants are not double keys.

Figure 4–29 compares the double-keys with Schoenberg's "regions" [Schoenberg1969]. Double-keys involving the major are identical to Schoenberg's closest regions of major, but double-keys in minor are different from Schoenberg's closest regions of minor: Schoenberg selects the minor dominant and subdominant, which are not double-keys, but *not* the minor keys on the minor and major mediants and submediants, which *are* double-keys. From this we can conclude only that the minor dominant and subdominant are closely related but fully distinguishable from the tonic minor, and that the sub/mediant minor keys may be theoretically (or "functionally") *less* related, but share a unique pitch-class intersection.

4.5.2 Structure of Key-Asserting PcNs

Once we find all PcNs on a score, it's trivial to label each one that asserts a single or double key. For this analysis, these will be the only PcNs used. Figure 4–30 shows key-asserting PcNs of four different keys on a score.

Each key-asserting PcN can tell us something about what key area or intersection of keys we are in from a pitch-class set-theoretic point of view. This does *not* directly correspond to any music-theoretic or cognitive notion of key, because it is comparatively much too sensitive to detail, giving new key labels to brief digressions

E	e	G*	g	B♭	b♭
A	a*	**C**	c*	E♭	e♭
D	d	F*	f	A♭	a♭
				D♭	

	G	e	E	c♯*	C♯
c*	C*	**a**	A*	f♯*	F♯
f*	F	d	D		
	B♭				

Figure 4–29: Schoenberg's regions of the major and minor keys, with the reference key in green and the most closely related keys in yellow. Double-key relations are starred. Figures copied from [Schoenberg1969], p.20 and p.30 (stars added).

that would normally be taken as ornaments, borrowings from other related keys, "passing" keys, or tonicizations (brief, inconclusive, formally subordinate excursions into other keys). Figure 4–31 shows a sequence of brief key areas, giving a tonal description that is more detailed than a typical key-labeling.

We do not make it a priority to heuristically reduce the set of PcNs to represent the smoother, more summarized analytic level that is required by the typical definition of "key" – instead we examine the relations between these diatonic assertions, and what they show on scores. We would like to be able to *structure* sets of PcNs into higher-level configurations, in order to obtain a more global, contextual perspective on how key asserting PcNs may behave together to form the outlines and the details of tonality. Since PcNs have time intervals that overlap and are included in each other, they can be structured *polyphonically*.

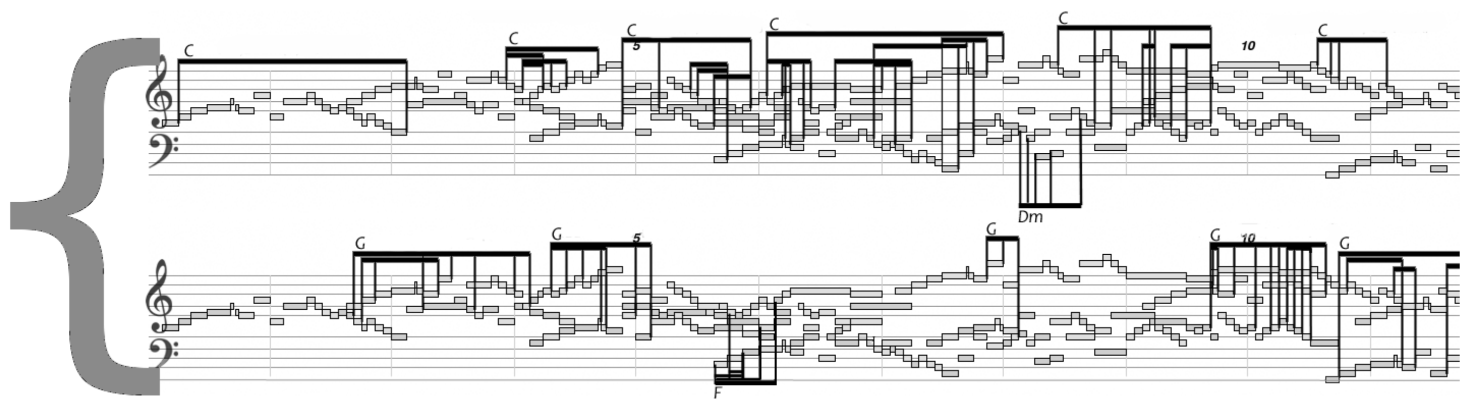

Figure 4–30: Beginning of Fugue No. 1 in C major from WTC I by J.S. Bach; two staff-systems to be read simultaneously. Key assertions in C major (top, stems up), D minor (top, stems down), G major (bottom, stems up), and F major (bottom, down) cover the excerpt. Many key-assertion regions contain multiple key assertions, of which one (the "top" one) includes all of the others. "Top" PcNs of each key may (temporally) include other PcNs of the *same* key, and may overlap with PcNs of different keys.

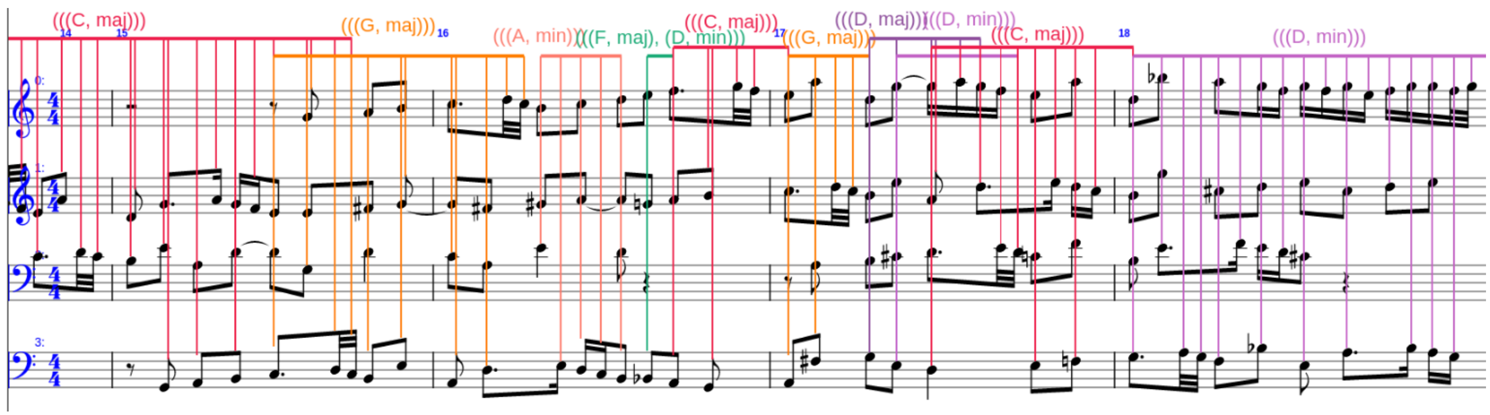

Figure 4–31: Excerpt from Fugue No. 1 in C major from WTC I, showing detailed passage through a series of brief key areas. Only *top* key assertions are shown – those that are not included in another key assertion with the same label.

A polyphonic analysis of the set of *single*-key PcNs shows the following. A PcN from one key may only be *included* (polyphone *hold*) in another PcN in the *same* key. This is evident by construction, since given PcNs X and Y and a key-set K, if $X \subset Y$ and $Y \subseteq K$, then $X \subset K$.

Therefore, if we reduce the polyphone graph to *hold* edges only, we obtain a sequence of connected components where each component refers to *one* key. Since hold graphs are rooted trees, each of these connected components has a *top* PcN which contains all of the others. These *top* key PcNs are useful for summarization, significantly reducing the number of PcNs needed to describe key.

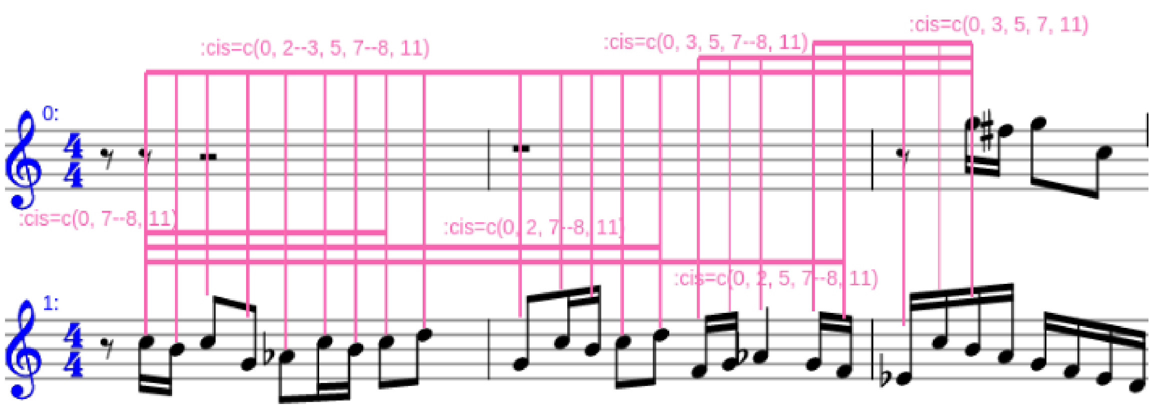

Figure 4–32: Bach: Fugue No. 2 from WTC I (excerpt). Overlapping and included PcNs asserting C minor. The “cis” (“compact integer set”) of each is its pitch-class content. The largest, with set {0, 2, 3, 5, 7, 8, 11}, includes all of the others.

Figure 4–32 shows a hold tree of key PcNs: asserting segments in the same key overlapping and included in one another, and all included in a top PcN.

Included PcNs within one of these one-key hold trees may fold (overlap) with one another. But two *top* PcNs that are subsets of the same key sets never fold with one another. This is because if two PcNs from the same key set fold with one another, then there is a third PcN in the same key which includes them both (by the "Join N-set" rule in Section 3 above).

Top PcNs in different keys may fold with one another (– they may be the heads of fold families connecting their respective hold-trees).

Including the minor key plus-major-sixth PcNs in this analysis of the polyphonic structure of single-key assertions does not change these inferences.

Now consider the polyphones of *double-key* PcNs on a score. The polyphones of double-key assertions has the same structure as the polyphone of key assertions, with an intersection of keys standing in for a key. Therefore, we can obtain a set of double-key *top* PcNs, and proceed with those.

Double-key tops may be included in zero, one, or two single-key tops on the score. *Included* double-keys imply a "region" of the superordinate key(s). They do not provide new information about *which* key we are in, but they provide even more fine-grained detail about what "part" of the superordinate single-key is being expressed. Figure 4–33 shows double-key assertions within single-key assertions. While this is of interest for a more detailed harmonic analysis, we do not pursue it further here.

Double-key PcNs that are *not* included in a single-key PcN can contribute to an analysis of key by offering information about events that are not covered by single-key assertions. Figure 4–34 illustrates.

We can reduce the total graph of key-asserting PcNs to a graph of tops (i.e. with no hold-edges). On the score, this translates into a set of temporal segments that

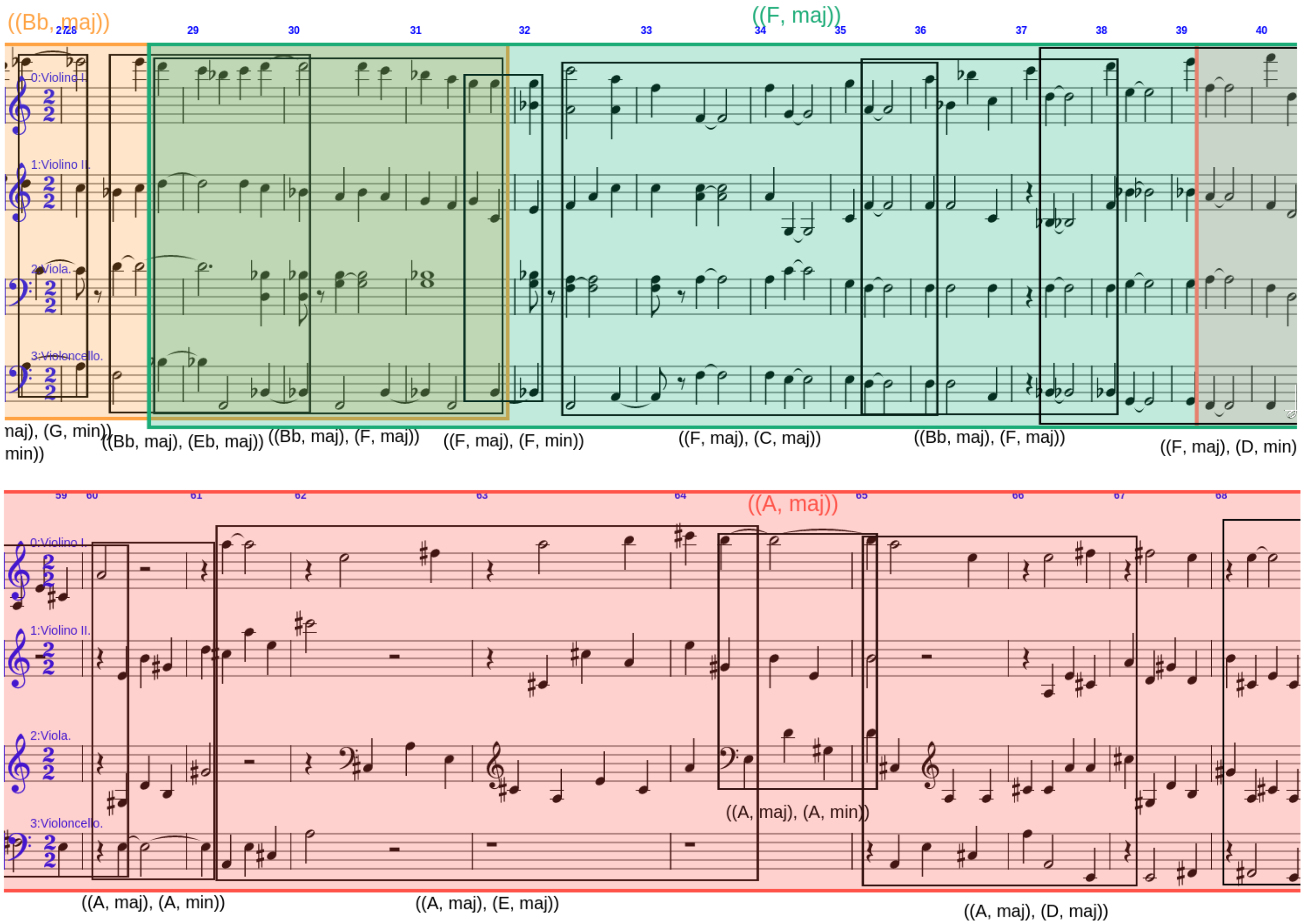

Figure 4–33: Double-key assertions within single-key assertions, giving detail about key “regions.” Excerpts from Beethoven, String Quartet No. 16, Mvt. 4.

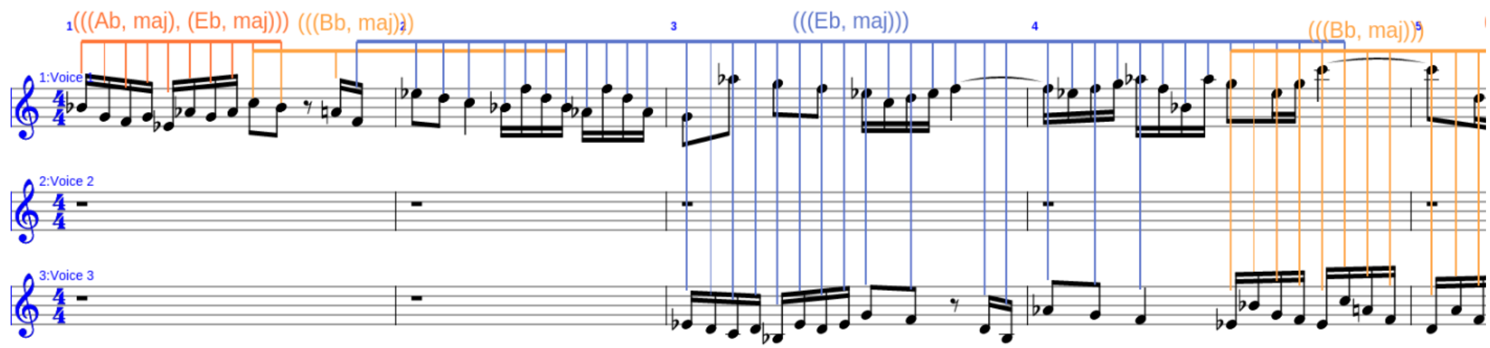

Figure 4–34: Top single and double key assertions at the beginning of Fugue No. 7 in E♭ major from WTC I (Bach). The opening segment is in the double-key (intersection) of E♭ major and A♭ major, and is not described by any single-key assertion.

are sometimes overlapping, sometimes abutting, and sometimes have gaps between them, but are never contained in one another. Each segment is labeled with a single or double key label, summarizing its pitch-class content with respect to inclusion in the sets defined for the keys.

Coverage and Correctness

Figure 4–35 shows segments that are *not* covered by key assertions. For tonal music, a great majority of a score is covered by key assertions.[8] For uncovered moments, one approach would be to develop further key-assertion-style labels (triple-keys etc.) to obtain greater coverage; but it seems that a contextual analysis relating these segments to surrounding keys would be more straightforward.

Since key assertions are structurally *defined* as PcNs with pitch-class sets that are subsets of the defined pitch-class set for a diatonic key, they are always “correct” in terms of the given diatonic definition. In some cases, however, the diatonic key-assertion label is different from the music-theoretic (usually harmonic) analysis. An alternative or augmentation to key-assertions might consider the location of *triads* (and other chords) in PcNs, in order to capture this point of view.

[8] A corpus of 83 string quartet movements (152175 notes) by Mozart was found to be 97.25% covered by single-key assertions and 99.17% by single- and double-key assertions together. A more chromatic tonal corpus of 52 piano pieces (206720 notes) by Liszt was 87.79% covered by single keys and 94.59% covered by both. For comparison, Stravinsky’s Rite of Spring, an atonal work (35572 notes) is 56.48% covered by single-key assertions and 71.32% covered by single and double assertions. Schoenberg’s atonal Opus 33a (761 notes) is 65.44% covered by single-key assertions and 84.01% covered by both.

Figure 4–35: Haydn Quartet Op. 54, No. 2, Mvt 3. Some segments are not covered by key assertions. In the top system, the uncovered segment by itself is included in four different keys *not* including the surrounding key G major. In the bottom system the first uncovered slice consists of a set of 4 notes not included in any key. In both cases it's easy to guess a contextual key for the uncovered segments by observing surrounding key structure.

Figure 4–36: Prelude No. 1 in C major from WTC I, by J.S. Bach (excerpt). The “A minor” segment contains a diminished seventh chord borrowed from C minor, resolving to C major. The diminished seventh chord by itself is not enough to assert any key, and its surroundings do not include other material from C minor; so it falls under an A minor assertion.

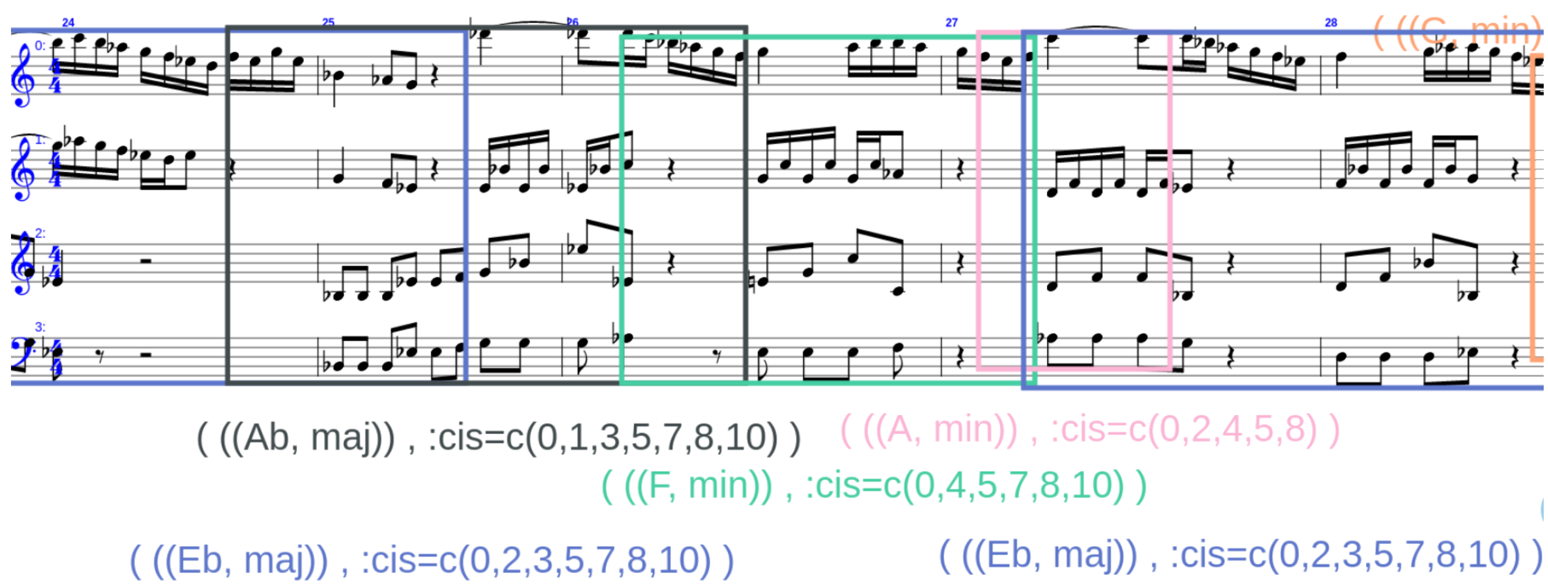

Figure 4–37: Haydn: String Quartet Op. 1, No. 2, Mvt. 1 (excerpt).The “A minor” segment is a case of “overlabeling” – the segment does not contain the tonic A, and is completely overlapped by (better) key assertions on either side. This segment isn’t a good example of A minor – showing that *diatonic* unambiguity isn’t sufficient for the semantic notion of *key*. Further analysis for harmonic content would help.

When a diminished seventh chord is followed by a major triad, the diminished seventh chord is usually considered to be a borrowing from the parallel minor. The diminished seventh chord is a symmetric chord that occurs in four minor keys, and so by itself does not assert any key. Therefore its context determines the key assertion – Figure 4–36 shows an instance where the diminished seventh chord's preceding diatonic context puts it in A minor, while a harmonic analysis would put it in C minor.

Figure 4–37 shows "overlabeling" of keys, where the temporal zone intersecting two diatonic keys produces a subset of a third (unnecessary and incorrect) key. In this case, the key-assertion at the intersection *doesn't contain it's own tonic note*. Some of the key-asserting sets as defined don't express their key very well – a refinement of the heuristic labeling of key-asserting sets might could ensure that certain tones, intervals, or triads are present. Or, a reduction of the labeling of "top" key assertions could leave out those that are completely covered by stronger assertions on their left and right.

Subordinate Key-Assertions and a Heuristic for "Overall Key"

Only the *top* key-assertions for each key – those not included in other key-assertions with the same label – are needed to see which key is where on the score. We have not used the subordinate PcNs so far in our key analysis. But having several subordinate asserting segments could mean that a key is better established, or more certain. For example, a long PcN might be ambiguous for most of its extent, becoming a key-assertion only towards the end. Subordinate segments show that the key is asserted and re-asserted during the extent of the top PcN.

Key-Profile	Derivation	Errors /120
Krumhansel-Kessler	Psychological test data	12
Bellman-Budge	Chord freq., 18th-19th C.	9
Temperley-KP	Scale deg. freq., 17th-20th C.	8
Aarden-Essen	Scale deg. freq., Ger. folk song	6
Sapp-Simple	Theoretical model	4

Figure 4–38: Performance of key-profiles for guessing overall key in WTC I and II and Chopin's Preludes Op.28 [Sapp2011]. The key assertion method does just as well as the "Sapp-Simple" key profile. Key profiles are from [Krumhansl1990, Bellman2005, Temperley2007, Aarden2003, Sapp2011]

The total number of key assertions in each key is a heuristic for guessing the "overall" key of a piece: it turns out that the key with the maximum number of key-assertions, including subordinate PcNs, is often key of the piece. This heuristic gives a correct result on 97.9% of the pieces in the Well Tempered Clavier, Books I and II combined; and 88.4% correct on the corpus of Chopin's Preludes and Nocturnes.

Another obvious heuristic for guessing the overall key is to take the key covering the largest number of *events* in the piece. The results using this heuristic are much worse, with 91.7% correct on the Bach corpus, and 83.7% correct on the Chopin corpus. Since counting key-assertions in each key is a better heuristic than the reasonable heuristic of counting number of events in each key, this is evidence that included assertions may *not* be redundant, and that they seem to have some bearing on the importance of different keys within a piece.

Counting key assertions also does well compared to *key-profile* methods for guessing the key of a piece (described in more detail below). Correlating formulated key-profile histograms with a histogram of the pitch-class content of a piece by duration, [Sapp2011] compares five sets of key-profile histograms for overall-key guessing. The test corpus includes both books of the Well-Tempered Clavier as well as Chopin's Preludes Op. 28: a total of 120 pieces. The results are shown in Table 2. Of the five key-profile sets, one is based on psychological data, three on music data, and one on a simple theoretical model. The theoretical model performs best out of the five, with 4/120 incorrect. The method of counting asserted segments does just as well as the correlation method with the simple profiles, also getting 4/120 incorrect on this test corpus.

4.5.3 Literature Review and Comparison with Key-Induction Methods

Current methods for key induction tend to divide the problem into two parts: deciding on a (single) key to describe a musical *segment*, and deciding *how* to segment a score into different key areas – *key mapping*.

Deciding the key of a musical segment

A variety of mechanisms for deciding the key of a musical segment have been designed using theoretical models [Chai+2005, Chew2006], analysis of music data [Temperley2002] or the results of psychological testing [Krumhansl1990, Noland+2006]. It turns out that despite their differences, these operate on some of the same fundamental principles.

A model of key based on psychological data is developed in [Krumhansl1990]. The model consists of a set of *histograms* representing the perceived stability of each chromatic scale degree in the major and minor keys. The histograms are called *key-profiles*, and the key of a musical segment can be decided by finding the key-profile which is most similar (i.e. most proximate) to the (weighted) pitch-class content of the segment. While histogram key-profiles can be obtained by other methods (theoretical models, data analysis); the essential concept for us is the measure of similarity or proximity from the pitch-class content of a target musical segment to each of the key-profiles, with the decision procedure of taking the closest.

In [Temperley2002], A Bayesian approach to data analysis is used to build a model for key. Each chromatic scale degree is assigned a probability P_i of occurring in a major or minor key segment, based on its frequency of occurrence in the relevant key in a labeled corpus.[9] The probability of the scale degree *not* occurring in a given segment is taken to be $(1-P_i)$. Given the set of pitch-classes in an unlabeled segment, a likelihood score for each key is determined by taking the product of the probabilities P_i for chromatic scale degrees in the set and $(1 - P_i)$ for each of the scale degrees not in the set. The key with the highest likelihood score is chosen as the key of the segment. While this takes into account absent as well as present pitch classes, the paradigm is essentially similar to the key-profile method described above – maximal statistical similarity of the pitch-class content of a segment to one of a set of model distributions.

An example of a theoretical model, [Chew2006] designs a geometric model of tonal space in which pitches, triads, and keys are represented by points on an array of spirals (shown in Figure 4–39). The model first places pitch-classes on a spiral which sets them in a "spatial" relation to one another, in which more theoretically

[9] Since the keys are assumed to be identical under rotation – as they also are in [Krumhansl1990] described above as well as in the key-assertion method, the scale degree representation (i.e. pitch-classes normalized relative to key) is equivalent to the pitch-class representation.

"related" pitch classes are closer together. Then, major and minor triads are defined as triangular planes with their vertices at the three pitch classes of the triad. A *weighted* center for each triangle is defined, based on the theoretic importance of the three pitches in the triad. Finally, *keys* are defined as triangular planes between the three most theoretically important triads of the key (tonic, subdominant, and dominant), and a central point is defined to represent the key, based on the relative theoretic importance of the triads.

Segments of music are represented geometrically by taking the convex combination of pitch-points weighted by their durations (or metric weights) – the "center-of-effect" of the segment. The key-finding method is simply to find the geometrically closest key-representing point to the center-of-effect.

This model is superficially different from the last two models discussed, offering a visualization of a "space" of pitch-classes, triads, and keys – but it is similar to the other models, because the points assigned to each key have some set of distances to the points assigned to each pitch; the center-of-effect for a segment likewise has a set of distances to the pitch-points (this is how the center of effect is calculated). If we made histograms of these distances and used the right similarity metric, we would have the *same* algorithm as in [Krumhansl1990] (with differently defined histograms).

These key-induction methods are differently derived and implemented, but they all treat key in an essentially similar way. They treat key as a *distribution* of pitch-classes, with different in-key notes having different weight of importance, and with

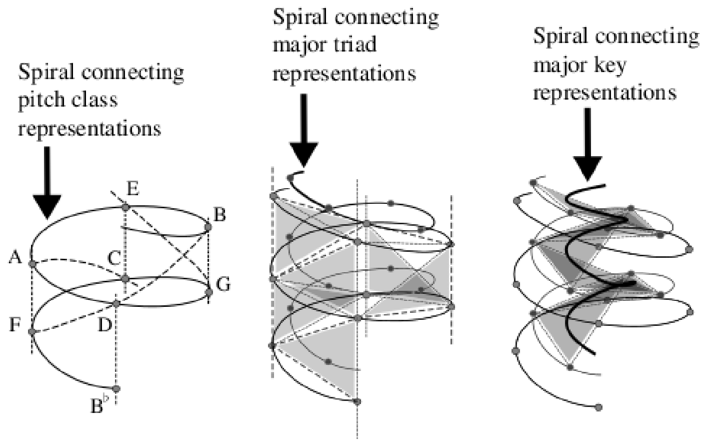

Figure 1 **The Array of Spirals in the Spiral Array**

Figure 4–39: Figure from [Chew2006]. (Reproduced with permission. Copyright, INFORMS, http://www.informs.org.) The first panel shows pitch classes represented as points on a spiral; in the second panel, triads are represented as points on the triangular plane between three pitch classes and a second spiral is drawn connecting these points; in the third panel, keys are represented as points on the triangular plane between three triads (the tonic, dominant, and subdominant triads), and a third spiral is drawn connecting the keys.

some of out-of-key notes expected or tolerated. Key-induction, for these computational methods, means finding the key distribution from a pre-defined set that most closely matches a distribution found in a musical segment.

Comparison with PcNs

Like the methods above, PcNs operate in the reduced space of pitch-class content. But unlike the above, PcNs use exact, *set-theoretic* relations to establish a correspondence to key, rather than comparisons with distributions. One advantage to this is that the step of *summarizing* the relation of the segment to a key does not require the loss of information, since none of the content has to be smoothed away as accidentals are treated as "noise." The relation between the pitch-class content of a musical segment and the keys-as-pitch-class-sets is maintained as a concrete set-theoretic relation, suitable as a foundation for building further structural relations, while bearing useful heuristic information.

The statistical models in the literature are designed to work for any pitch-class set: no matter how atonal a segment is, there must be some *closest* key(s); and furthermore a measure of the "distance" should be available. On the other hand, while it's possible to state the set-theoretic relations of inclusion or intersection between *any* pitch-class set and the keys-as-pitch-class-sets, we propose to consider primarily the PcNs that have simple such-relations: those that are included in just one or two keys. We are proposing, therefore, to proceed without a key-decider that works on *any* pitch-class set. It may be, after all, that a very "distant" recognition of a key by a statistical model is not really tonal in a music-theoretic or cognitive sense, and that a heuristic distance-limit should be set, in order to detect the *absence* of key. In these areas, statistical methods may hazard a guess, whereas PcNs will tell us that the segment can't be summarized using the simplest set-theoretic relations.

With tonal music, in any case, the problem is generally not to detect atonality, but to provide a *map* of the keys of a score, guessing a *segmentation* as well as a key-label for each segment.

Literature on key-mapping

The methods described above select a (single) key for a musical segment; these can be used together with algorithms that decide on a *mapping* (i.e. segmentation) of a score into different key areas.

One such method is to decide on a probability (or, equivalently, a penalty) for *changing* keys, and then use *dynamic programming* to find the most likely sequence of keys for a sequence of segments in a piece of music. The piece therefore must be pre-segmented, resulting in a low-resolution key-map.

Dynamic programming can be visualized as an efficient way of finding the *least-cost path* from the beginning to the end of a piece, where the cost for assigning a segment to a key is based on how well the segment fits the model of the key, plus an additional cost for changing keys between consecutive segments. Dynamic programming is used in [Temperley2002]; it is also the same algorithm used by researchers designing "hidden Markov models" (HMMs) for key induction [Chai+2005, Noland+2006].[10]

A more flexible approach to key-mapping is described in [Chew2006]. Unlike the dynamic programming method, this "Argus" method can construct segmentations at any point in the music. The geometric model described in [Chew2006] is used as the key-selection method, although another method could be substituted.

The Argus method works by looking at the centers-of-effect of segments before and after a potential key-boundary. The idea is that there will be very *different*

[10] In this context dynamic programming is often called "the Viterbi algorithm." Hidden Markov models, as [Noland+2006] points out, are ill-suited for key induction, since the principle behind the model being "hidden" is that it should be induced algorithmically. Suppose we take the *keys* as "hidden states" that generate the pitch-class set data. If we use expectation-maximization to try to induce models for these states, we may end up with hidden states that match the data well, but do not correspond to the notion of *keys*.

centers-of-effect before and after a key-boundary. When charted over time, considering each successive note as a potential boundary, the distance between the centers-of-effect will grow for a while as the boundary approaches, and then start to decrease. The (large) peak will be a good estimate for the boundary. Settings for threshold peak-height (distance between centers-of-effect) and size of window will determine sensitivity to local key-change.

Formalizing sensitivity to key change

The above methods for key mapping have a few things in common. First, they all model key-change as a *boundary*, whereas music-theoretic discussions of key-change emphasize a common method of modulation where an intermediate *pivot* segment is analyzed simultaneously in two keys, providing a transition between them – resulting in *overlapping* key segments. Since PcNs overlap with each other, we obtain overlapping labeled segments (though sometimes with *larger* overlap than the typical modulatory pivot). As well, PcNs have a *natural* size, based on where the pitch-class set is broken on the score. This obviates the need to predetermine a universal segment-size, or to use a heuristic threshold to find good boundaries, or to simply take *all* possible sizes.

Another similarity between the simple segmentation method and the argus method is that *sensitivity* to key-change is modeled by two constants: segment size, and key-change probability/penalty or threshold peak height. Settings for these constants can be determined in relation to an evaluation corpus of hand-labeled music. However, even an optimal constant setting will usually result in "errors" in rate-of-modulation with respect to the test corpus: this parameter of (human) key-labeling just isn't suited to being modelled by an inflexible constant, because sensitivity to key-change is dependent both on musical context (including rhythm, harmony, form, etc.) and on subjective difference (how someone "hears" or *conceives* of key).

In the literature under review, where human-labeled data is used, there is no discussion of the problem of sensitivity to key change, and no mention of whether the expert-labelers were instructed to adhere to a particular notion of key which tries to formalize what counts as a key change.

Given the contextual factors, modelling the parameter of human sensitivity to key-change is not immediately an option within the framework of pitch-class content. An algorithm's sensitivity to key-change can nonetheless be *formalized*. This is important because if the output of an algorithm is to be useful as a structural building block for a more complete analysis, it must behave consistently.

It is possible to formalize the parameter of sensitivity to key-change in two simple ways, by either *minimizing* the parameter, or *maximizing* it. Minimizing sensitivity to key-change is equivalent to applying a key-decision model to a whole piece as one "segment" with no key-changes recognized (as in [Krumhansl1990, Sapp2011]). Maximizing sensitivity would mean that all key areas, no matter how small and unimportant, are recognized.

The PcN key-assertion method maximizes sensitivity to key-change. As well as being more formal and parsimonious (no "magic number" parameters) than a constant setting for sensitivity to key-change, maximizing sensitivity preserves a maximal amount of *information* about key areas large and small. This information can be used in later stages of analysis, which could include using more musical context to model the more complex sensitivity to key-change shown by human labelers.

Differing goals

It's impossible to decide *how well* the PcN method performs in comparison with the statistical methods for key induction. The difference in the methodology arises because the problem statement and the goal for each is defined in a different way.

The statistical modellers are trying to build a heuristic model that imitates or approximates *semantically*-labeled data. In this model, the judgments *of people* are a "ground truth" or a *target* – comparative data for empirical validation and/or the data from which the model is inferred.

The PcN method, in contrast, tries to build a detailed analytic picture of how pitch-class sets on a score are related to the pitch-class sets of keys – in a *more* fine-grained way than people would ordinarily describe key. The formality and determinacy of the method are a priority, since the PcNs are designed to be part of a larger structural analysis.

We do not deny that semantic interpretation might be the *most* interesting problem. But the *semantic gap* marks a current limitation of computation – we can't *compute* semantics, we can only *guess* at them using rules of thumb. The statistical methodology takes a low-informational stab at a semantic category. Rather than tackling a semantic distinction right off the bat, our general *structural* program is to avoid decisions for as long as possible, and work on *building information.* Therefore, we make an approach to key that *doesn't* heuristically *target* musical perceptions or music-theoretic conceptions, but one that nonetheless generates information that can be used to *address* semantic-level concerns.

One More Approach to Key Mapping

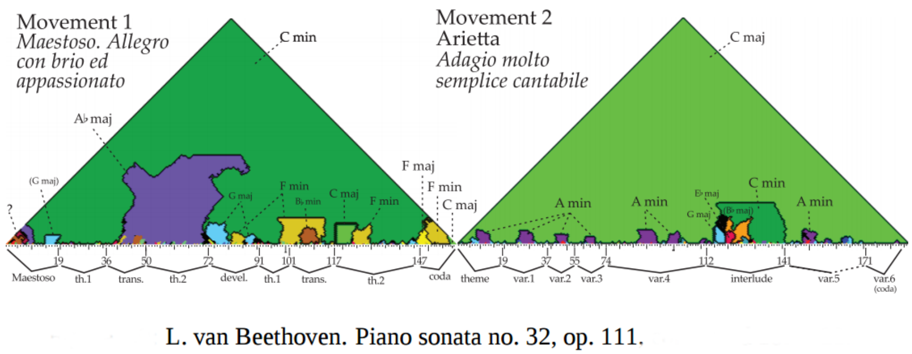

Figure 4–40: Figure from [Sapp2011]. Triangular "keyscape" drawings: the score (for each of two movements) runs along the base of the triangle. The color of each point on the triangle is determined by the guessed key of the segment beneath it, such that the "overall" key is at the apex, the key of the first half of the piece is found halfway up the left side and the key of the second half of the piece is halfway up the right side, and so on.

Another approach to key-mapping is proposed by [Sapp2011], who proposes *not* to decide where one key ends and the next begins, but instead guesses keys (using the histogram method) for all possible segmentations (i.e. all temporal pitch-class sets on the score). The result is a triangular map of keys called a *keyscape*, with the score running along the base, the "overall" key at the apex, the key of the first half of the piece found halfway up the left side and the key of the second half of the piece found halfway up the right side, and so on (shown in Figure 4–40).

This study was not performed using labeled key data (it includes a comparative study on different key-profile histograms). The problem of modelling the sensitivity to key change of human labelers is avoided because labeled data is not used, and also because all possibilities are mapped. This is another way of formalizing sensitivity to key change by refusing to parametrize it.[11]

The triangular keyscape method almost seems *structural*, because it is high-informational, taking all possibilities without a heuristic decision process (– the triangular segmentation method itself, not the statistical key-guessing method applied to the segments). It also gives a polyphonic, multi-scale analysis, placing local key areas within more global ones.

But part of structural analysis is finding "natural" (structural) places to cut data, (i.e. cuts that are based on properties of the data itself), and this analysis simply cuts *everywhere* – an example of a "powercut" (i.e. n-grams).

The triangular keyscape method therefore contrasts with PcNs: the triangular method uses all of the temporal sets on the score; PcNs take a structurally defined subset of these. Key-assertions make a further *heuristic* reduction.[12] While structural analysis tends to be high-informational in comparison with heuristic and statistical methods based on decisions and classifications, it does not generally operate by taking *all* possibilities. The triangular method finds *all* temporal sets and

[11] [Martorell+2016] extend the method given in [Sapp2011] to "class-scapes" rather than keyscapes – they use the triangular method to find the pitch-class sets for all segments on the score (i.e. they omit the heuristic key-guessing step in order to afford a pc-set analysis rather than a tonal analysis).

[12] Key-assertions are a structurally defined heuristic cut – heuristic because the key sets are defined heuristically (i.e. selected by hand), but structural because the principle defining single and double key assertions is set-theoretic, not metric.

uses a statistical method to guess their relation to a key; the PcN method finds those temporal sets that are both informative in relation to their temporal interval, and have a particular formal relation to a pitch-class set of interest. While the triangular method offers polyphonic relations of overlap and inclusions between its sets, the *shape* of this polyphone is regular and predetermined, whereas the polyphonic shape of PcNs bears structural information about the score.

4.5.4 More Key-Assertion Illustrations

In this section, we show diatonic sketches on scores given by key-asserting PcNs, showing different kinds of formal overviews and transitional processes. These heuristic labelings of PcNs, giving a detailed picture of the diatonic areas covered, are just one way among potentially many for reasoning about the key-relatedness of PcNs. The PcN approach offers a set of overlapping and included segments (here reduced to *overlapping* key-referencing segments), displaying the *natural* boundaries between pitch-class sets on the score.

Figure 4–41 shows a string quartet movement by Mozart and Figures 4–42 – 4–45 show scores by Bach, giving a variety of relatively simple tonal overviews. Figures 4–46 and 4–47 from Bruckner, and 4–48 and 4–49 from Chopin show some more intricate and less traditional tonal patterning. Figure 4–50 shows zoomed-out sketches of some of Bach's Goldberg Variations, which share a common broad tonal schema; key-asserting PcNs offer a an overview of some of the variety and complexity available within this schema from a diatonic point of view.

4.6 Future Work

We defined N-sets as a systematic method for structuring temporal co-locations of items on a score. While *polyphones* give precise temporal relations, *N-sets* offer a looser sense of what occurs "together" within a temporal interval. A polyphonic set of temporal intervals are selected for their *informativeness*, with their (natural) boundaries determined by difference, or set *exclusion*.

We showed an application to pitch-class sets, producing "PcNs" – future work would explore other musical (and non-musical) applications, such as the incidence of instrumental sounds on a score, or the locations of different melodic shapes.

We briefly discussed ways of obtaining PcNs on *sub*scores, such as taking each instrumental voice separately, or segmenting by phrase. The analytic challenges and opportunities afforded by crossing PcNs with other dimensions (such as rhythm or timbre), offer an open field for research.

We aimed to show the potential utility of PcNs in investigating tonal and atonal works, but the construction of any single careful, detailed analysis was beyond our scope. An in-depth discussion of the PcNs in a score and their participation in the story of the music would be a possible avenue for an analytic music theorist. An alternative, *quantitative* approach to investigating PcNs might also be of interest – quantitative and statistical approaches to pitch-class sets on a score are discussed in [Martorell+2016, Huovinen+2008]).

Statistical approaches to *subset* relations are given in [Martorell+2016, Huovinen+2008]; we showed how *polyphones* offer a *structural* view of the temporal relations of these sets, mapping relationships between subsets and supersets as well as folding transitions. Polyphonic analysis could assist in giving contextual *meaning* to PcNs, for example mapping forms and procedures such as subkey areas, modulations, and types of harmonic progressions, and assisting in making contextual or structured *selections* of PcNs. Polyphonic pitch-class set analyses and methods are a domain requiring further development.

We developed set-theoretic heuristics to investigate the relation of pitch-class sets to key-related sets – our *key-assertions* are just one of many possible approaches to key. For example, using triads or tetrachords as a basis would give a different view of harmony. Similar methods could also be developed for sets associated with other kinds of tonality or atonality, or to induce productive supersets and subsets over a score.

PcNs could also be suitable as a basis for heuristic musical applications such as *pitch-spelling*, in which a process decides for each note whether to "spell" it using a sharp or a flat (or neither) – in particular the inferences available about diatonic and harmonic regions and the transitions between them would suggest ways of constructing normative or smooth spellings.

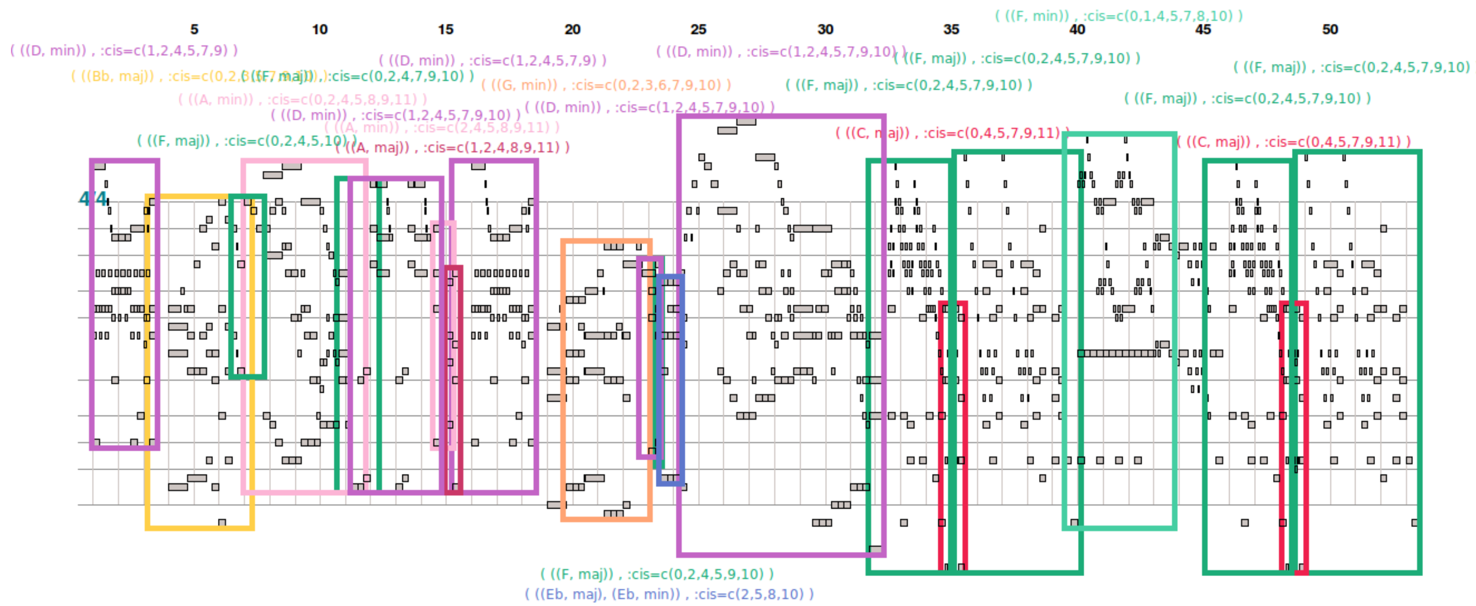

Figure 4–41: Mozart: String Quartet K. 173, Mvt. 3. "Key-asserting" PcNs give a summary of diatonic key areas (both important and transitory ones). The largest keys in this score: D minor (purple), B♭ major (yellow), A minor (pink), D minor, G major (orange), D minor, and F major and minor (green).

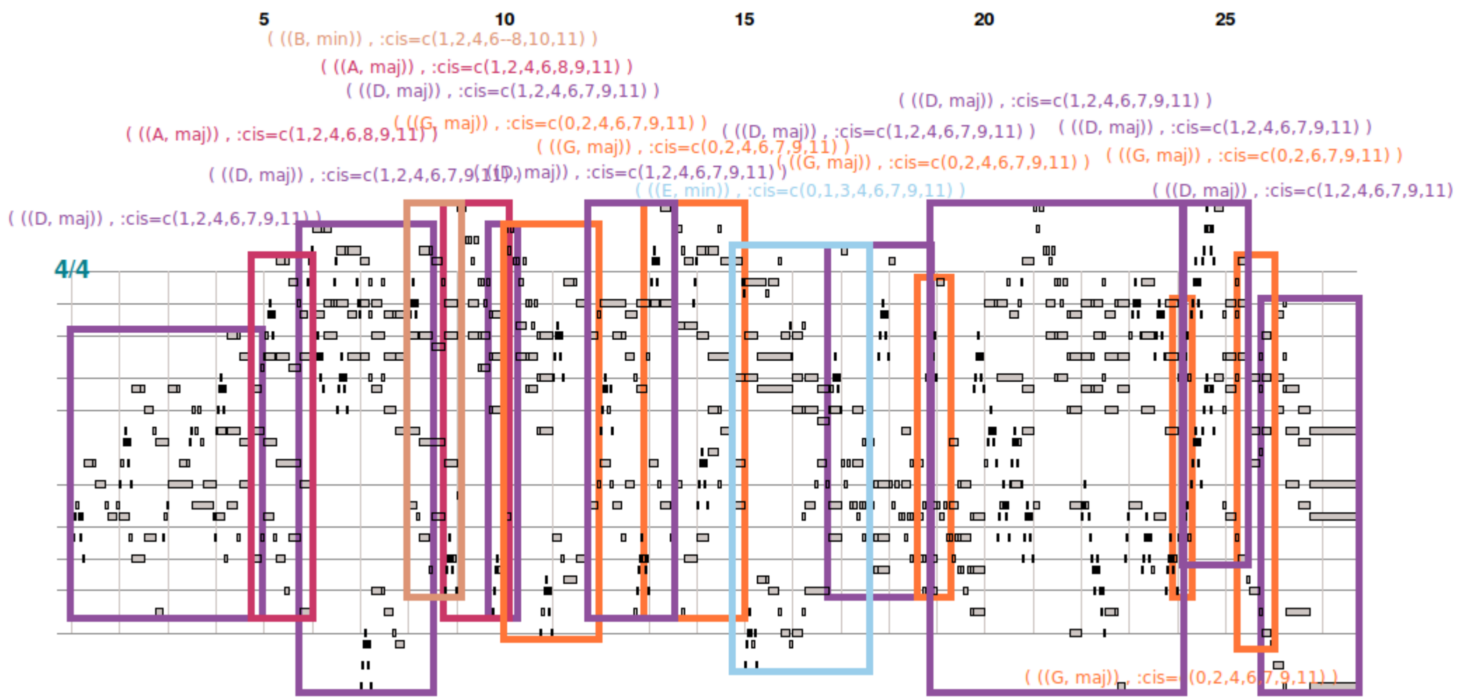

Figure 4–42: Bach: Fugue No. 5 from WTC Book I. The score is predominantly in D major, with excursions in A major (dark red), B minor (brown), G major (orange) and E minor (light blue).

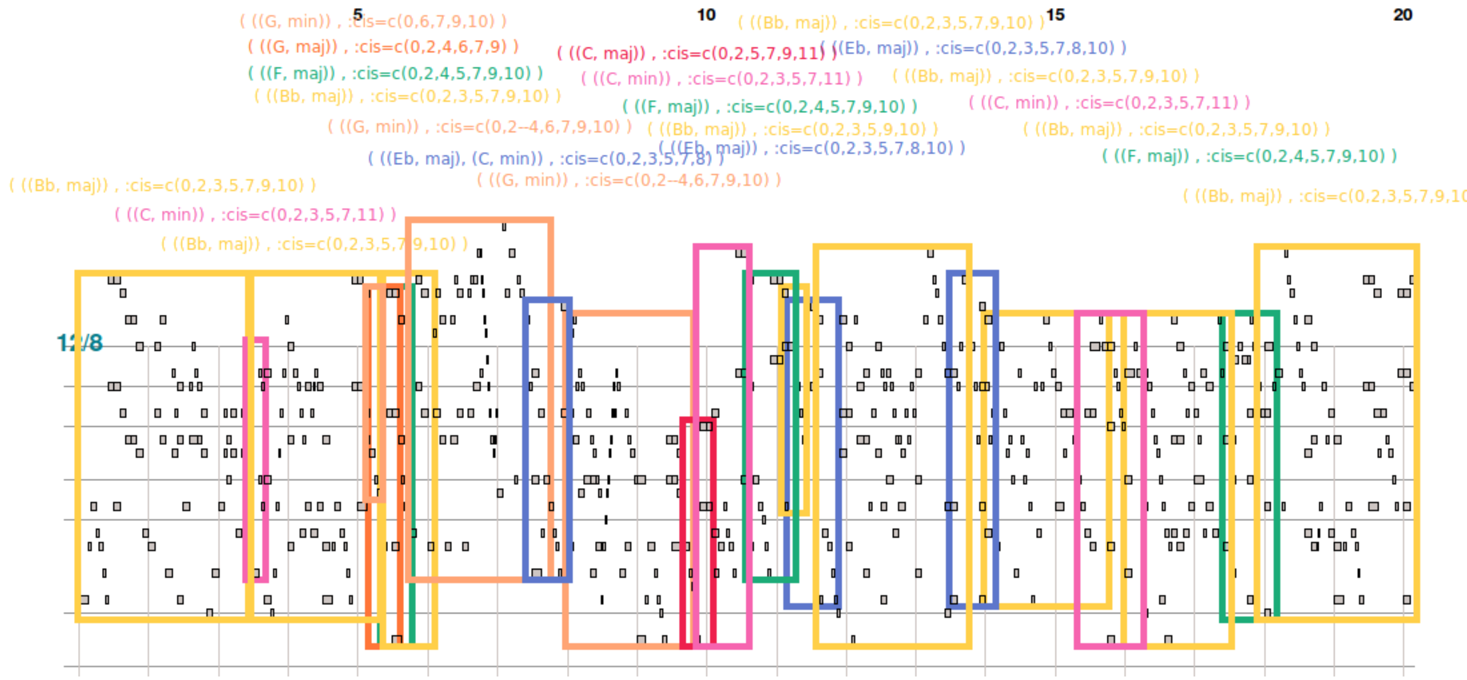

Figure 4–43: Bach: Siciliano for solo violin (from Sonata No. 1). Key-assertions give a summary of important and transitory key areas. The score is mostly in B♭ major, with C minor (pink), G minor (orange), E♭ major (blue), and F major (green).

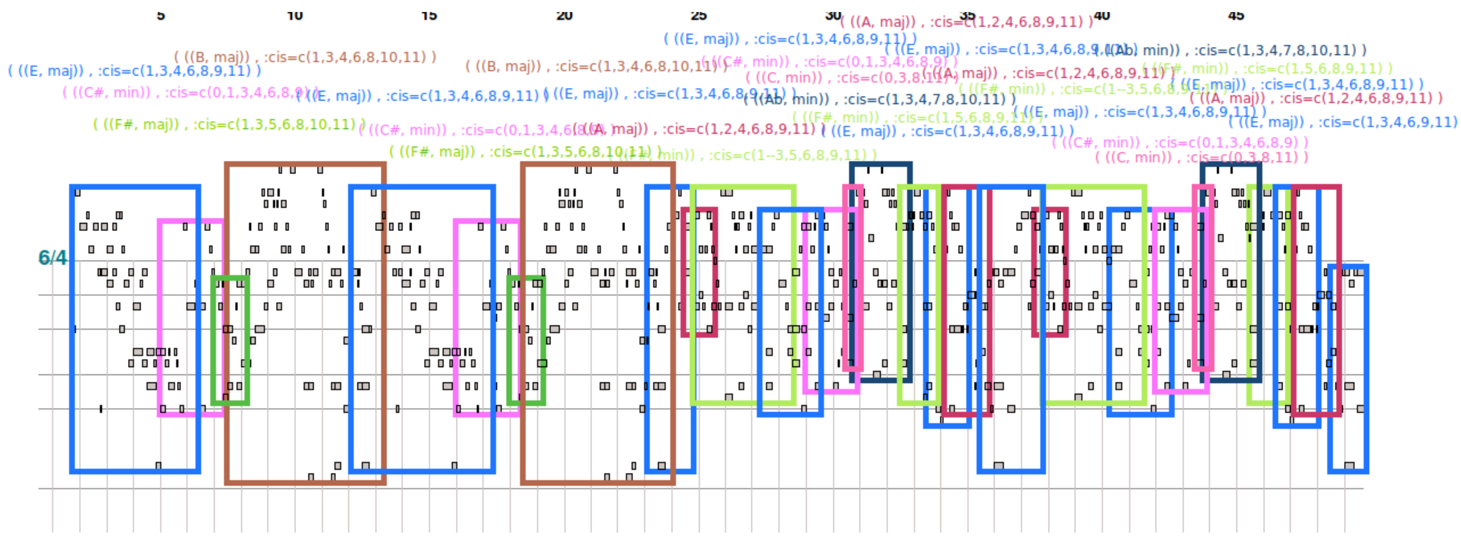

Figure 4–44: Bach: Louré for solo violin (from Partita No. 3). Key-asserting PcNs sketch diatonic key areas. Blue is E major, pink is C♯ minor, greens are F♯ major and minor, brown is B major, dark red is A major, and dark blue G♯ minor.

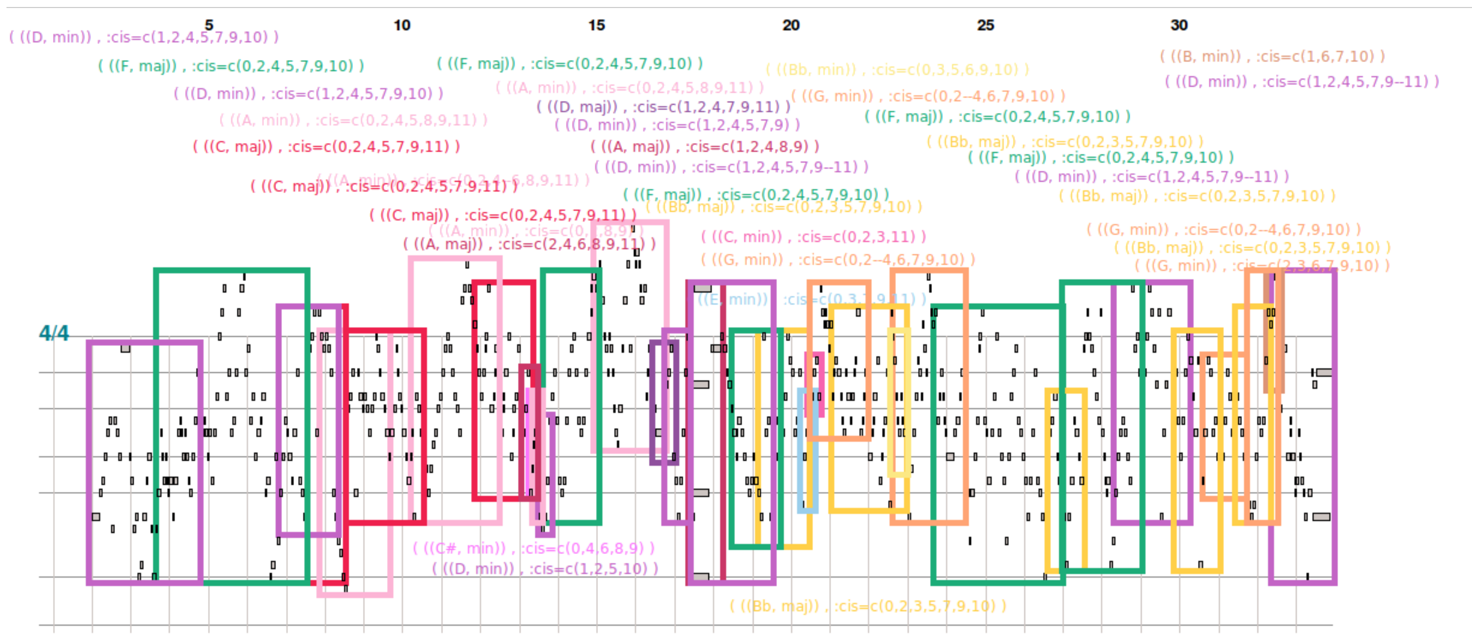

Figure 4–45: Bach: Allemande for solo violin (from Partita No. 2). Key-asserting PcNs show diatonic key areas. Purple is D minor, green is F major. The section in red and pink is C major and A minor; yellow and orange is B♭ major and G minor.

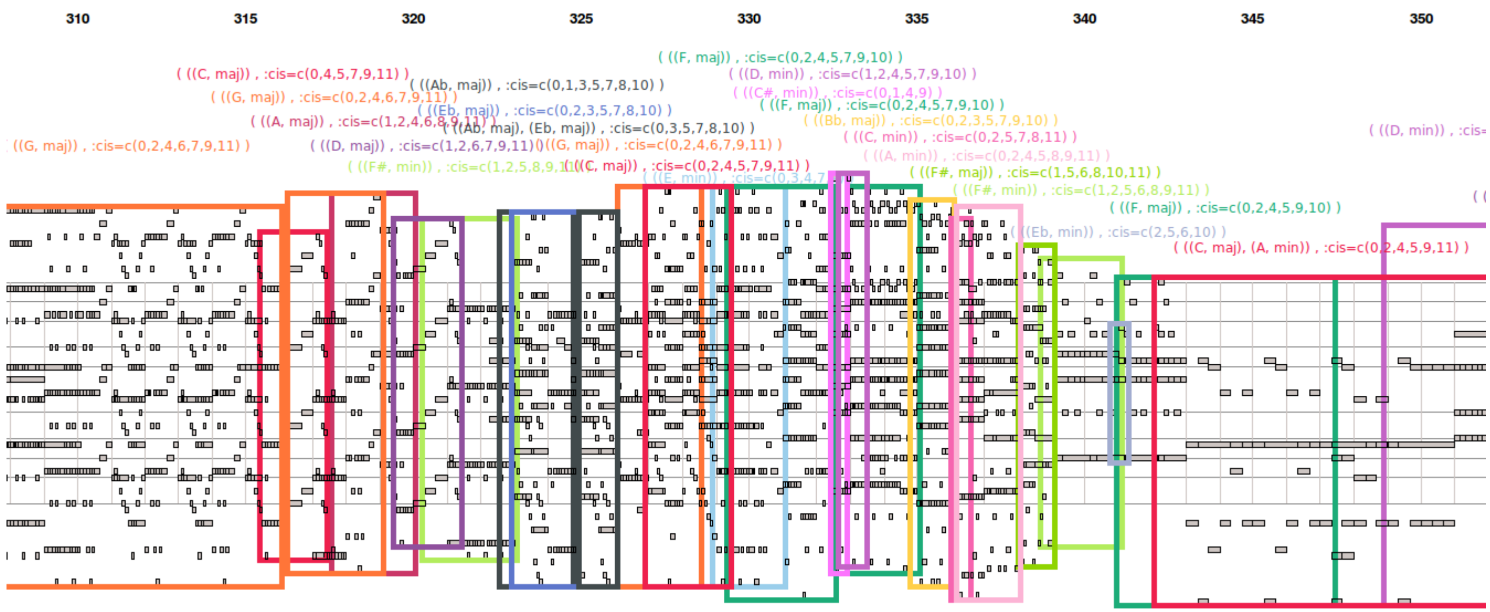

Figure 4–46: Bruckner: Symphony No. 4, Mvt. 4 (excerpt). Transitions between many different (and often not-too-closely related) keys. (Key labels are "unspelled" i.e. "C♯" may be either C♯ or D♭.) The most important keys here are (in chronological order) G major (with C maj), A major, F♯ minor, A♭ and E♭ major, G, C, F major (with D minor), B♭ major, A minor, F♯ major, F♯ minor, F major / C major / A minor.

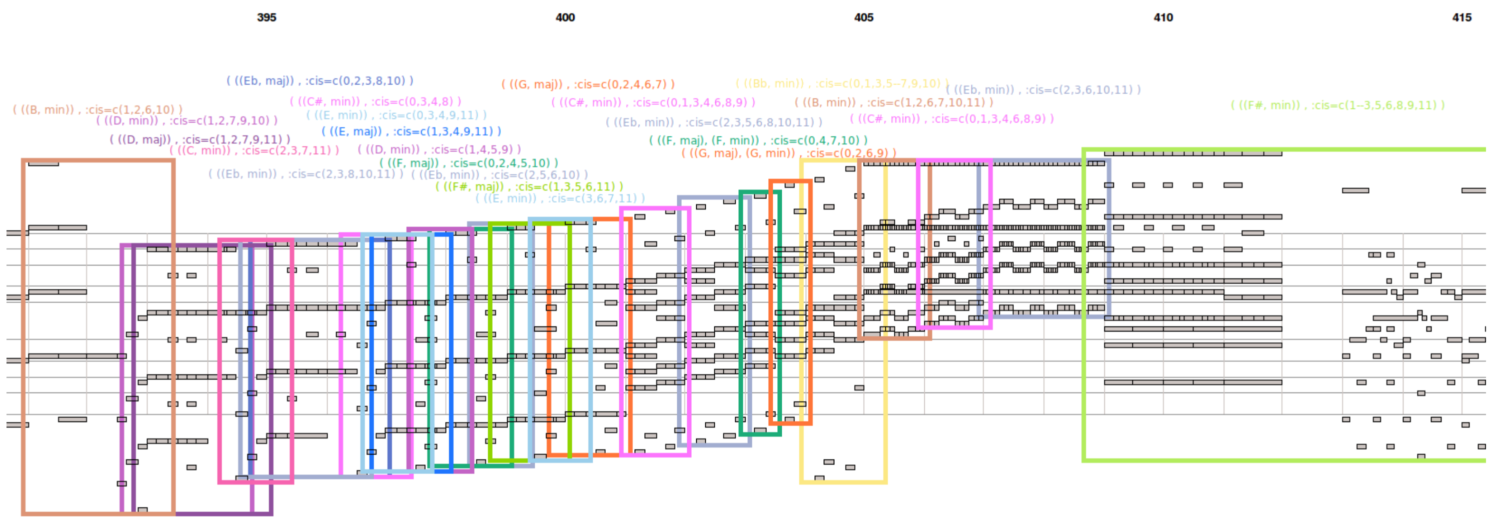

Figure 4–47: Bruckner: Symphony No. 4, Mvt. 4 (excerpt). Transitions between many different keys. The middle part of this excerpt is almost monophonic, and several of the key labels don't have their tonic notes – this segment could be reduced to the more confirmed labels: E♭ major, E major, F major, F♯ major, G major. The passage continues with C♯ (or D♭) minor, E♭ minor, F maj/min, G maj/min, B♭ minor, B minor, D♭ minor, E♭ minor. This is in contrast to standard tonal practice of tending to move between keys related by dominant or modal relations (as shown in 4–29), instead moving stepwise.

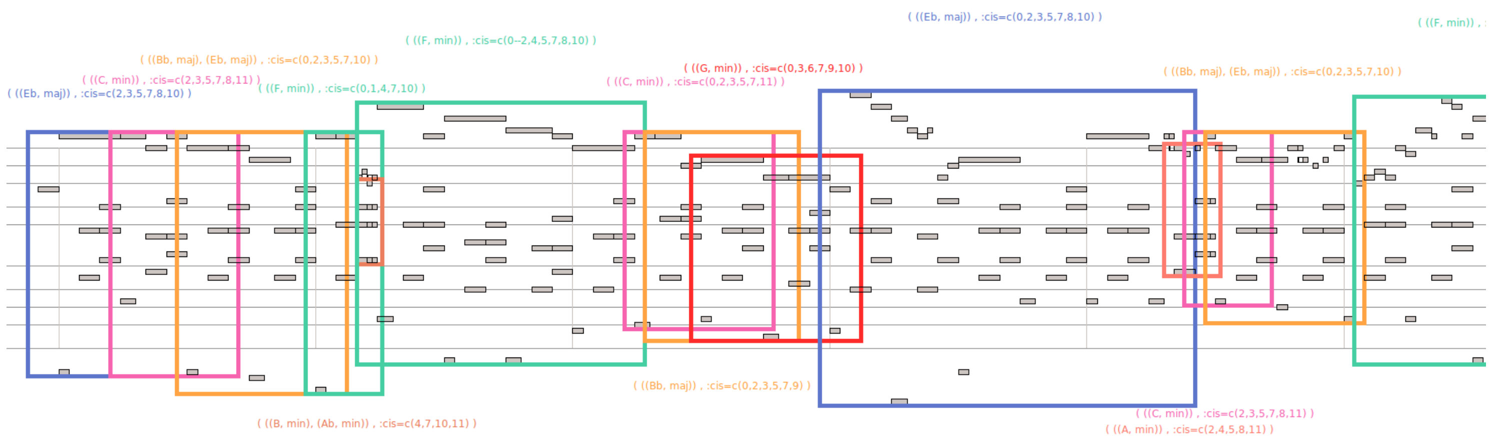

Figure 4–48: Chopin: Nocturne No. 2, first few measures. Chopin's fluid tonality gives chromatic inflections and borrowings from many keys; in this passage they have the character of smooth transitions back and forth between E♭ major and F minor. The opening measure has boxes for E♭ major, C minor, B♭ / E♭, and then into F minor with an ornamental moment in B minor / A♭. Measure 3 has overlapping diatonic boxes in C minor, B♭ and G minor, before going back into E♭ major.

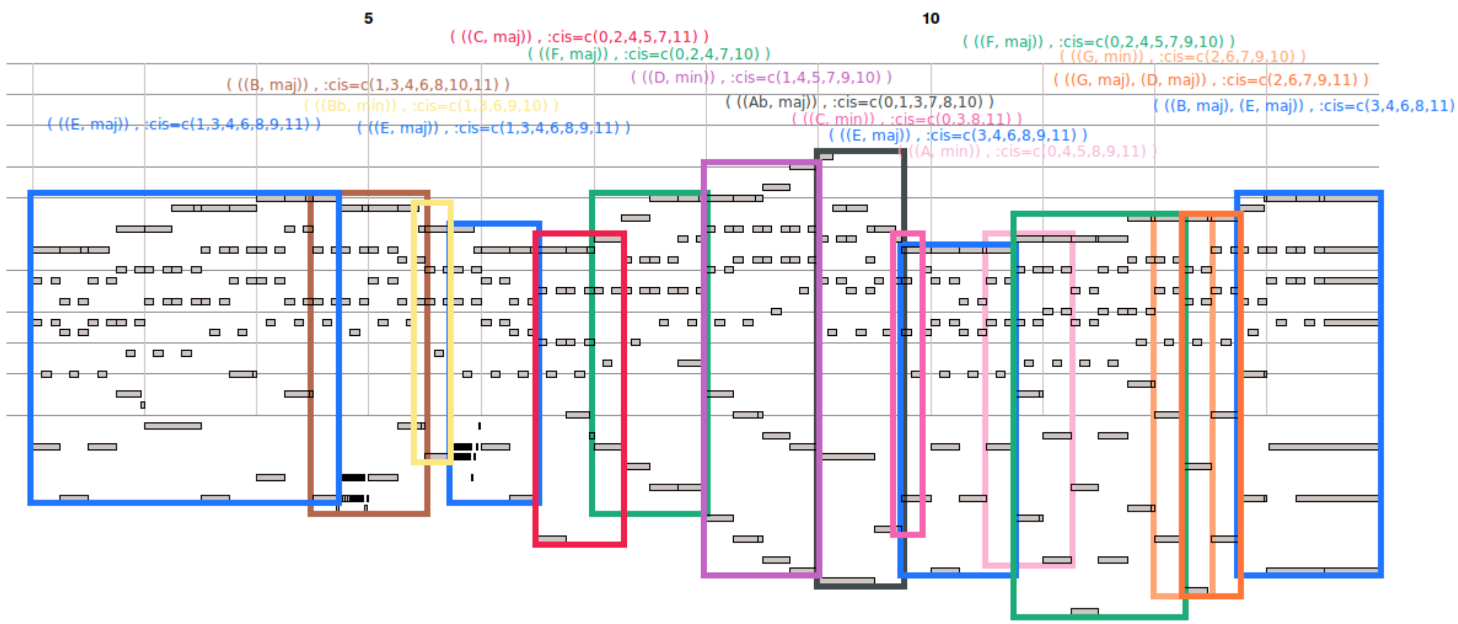

Figure 4–49: Chopin: Prelude No. 9. This small score shows a large variety in the kinds of transitions between keys. E major, to B major and back is a traditional “dominant” move E major to C major is a chromatic “third relation.” C to F to D minor is a standard subdominant-relative-minor move, followed by the unusual transitions to A♭ and E major, and then the *stepwise* transitions E, F, G minor and major, and another chromatic third relation to back to the key of E major.

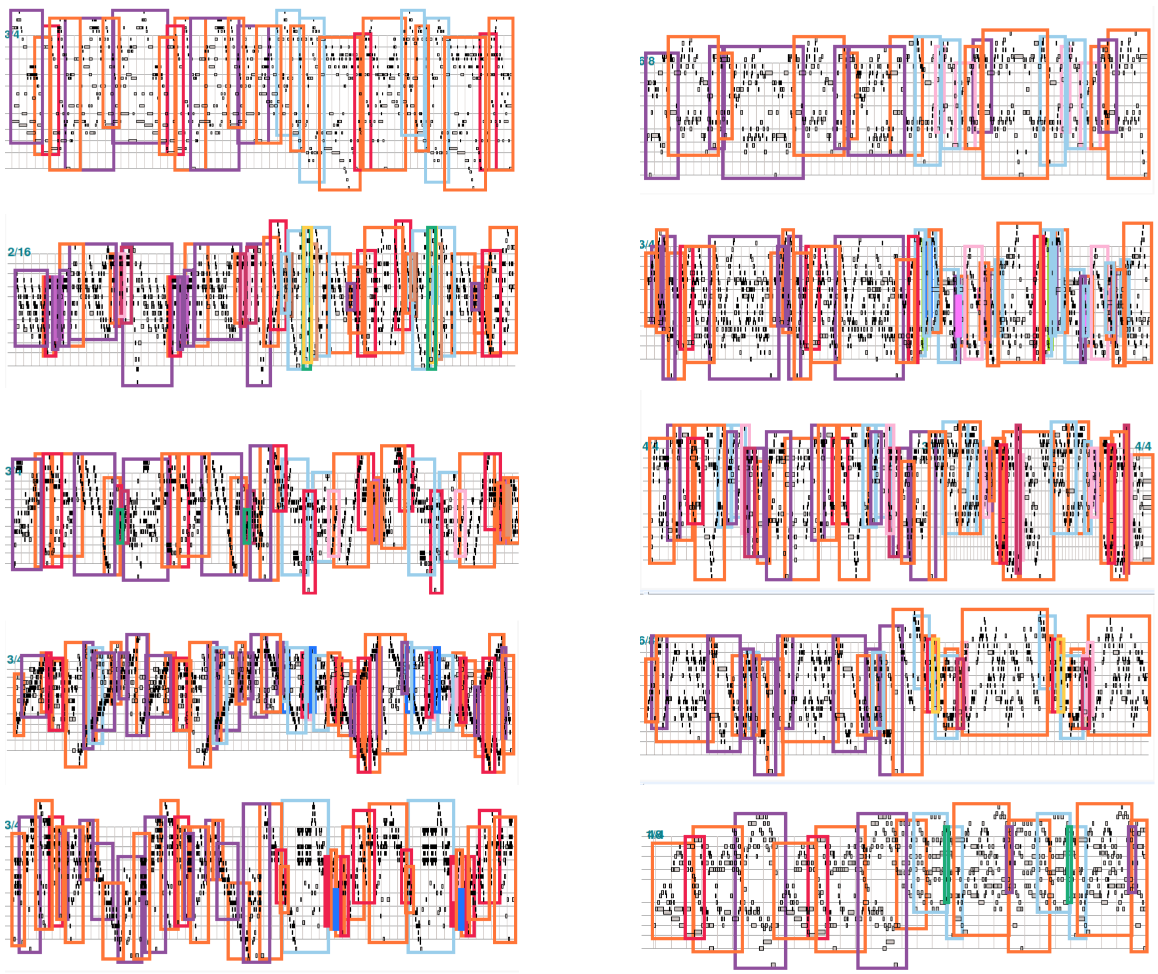

Figure 4–50: Key-asserting PcNs give diatonic sketches of some of the Goldberg variations by Bach (in order reading left to right on each line: the Aria, and variations 7, 11, 12, 14, 16, 26, 27, 29, and 30). These have roughly the same tonal outline, with alternations of D major and G major (purple and orange) in the first half (with borrowings from C major – red) , and E minor (light blue) and G major in the second half (again with C major borrowings). The harmonic tonic at th beginning of the score is G major, but often the diatonic key of the dominant D major is confirmed before the tonic key (i.e. some variations start with purple, other with orange). These sketches give a rough overview of the the variations are treated differently with respect to the density of diatonic shifting and the presence of different tertiary key areas (e.g. pink for A minor, dark red for A major, green for F major, yellow for B♭ major, blue for E major), within the same broad tonal outline.

Appendix:
Tables of Key-Asserting Pitch-Class Sets

N	0 (C) Major	0 (C) Minor
4	{11; 7; 5; 4} {11; 9; 7; 5}	{11; 8; 7; 5} {11; 8; 7; 2} {11; 8; 7; 0} {11; 8; 3; 0} {11; 7; 5; 3} {11; 7; 3; 2} {11; 5; 3; 0} {11; 3; 2; 0}
5	{11; 9; 7; 5; 4} {11; 9; 7; 5; 2} {11; 9; 7; 5; 0} {11; 7; 5; 4; 2} {11; 7; 5; 4; 0}	{11; 8; 7; 5; 3} {11; 8; 7; 5; 2} {11; 8; 7; 5; 0} {11; 8; 7; 3; 2} {11; 8; 7; 3; 0} {11; 8; 7; 2; 0} {11; 8; 5; 3; 0} {11; 8; 3; 2; 0} {11; 7; 5; 3; 2} {11; 7; 5; 3; 0} {11; 7; 3; 2; 0} {11; 5; 3; 2; 0}
6	{11; 9; 7; 5; 4; 2} {11; 9; 7; 5; 4; 0} {11; 9; 7; 5; 2; 0} {11; 7; 5; 4; 2; 0}	{11; 8; 7; 5; 3; 2} {11; 8; 7; 5; 3; 0} {11; 8; 7; 5; 2; 0} {11; 8; 7; 3; 2; 0} {11; 8; 5; 3; 2; 0} {11; 7; 5; 3; 2; 0}
7	{11; 9; 7; 5; 4; 2; 0}	{11; 8; 7; 5; 3; 2; 0}

Figure 4–51: Key-asserting pitch class sets for C major {0; 2; 4; 5; 7, 9; 11} and C minor {0; 2; 3; 5; 7; 8; 11}. A key-asserting set is one that is a subset of exactly *one* of the 24 pitch-class-sets representing the keys.

{11; 9; 8; 7; 3; 2}
{11; 9; 8; 7; 5; 3}
{11; 9; 8; 7; 3; 2; 0}
{11; 9; 8; 7; 5; 3; 0}
{11; 9; 8; 7; 5; 3; 2}
{11; 9; 8; 7; 5; 3; 2; 0}

Figure 4–52: Minor-key plus-major-sixth assertion sets for 0 (C) minor. The major sixth in 0 minor is 9. These sets are selected such that they cannot be construed as a subset of another key plus an extra pitch-class.

Maj 0 / Min 9 (dual: Min 0 / Maj 3)	{11; 5; 4} {11; 5; 4; 0} {11; 5; 4; 2} {11; 9; 5; 0} {11; 9; 5; 4} {11; 5; 4; 2; 0} {11; 9; 5; 2; 0} {11; 9; 5; 4; 0} {11; 9; 5; 4; 2} {11; 9; 5; 4; 2; 0}
Maj 0 / Min 0 (dual: same)	{11; 7; 5} {11; 7; 5; 0} {11; 7; 5; 2} {11; 7; 5; 2; 0}
Min 0 / Min 9 (dual: Min 0 / Min 3)	{11; 8; 0} {11; 8; 2; 0} {11; 8; 5; 0} {11; 8; 5; 2; 0}
Maj 0 / Maj 5 (dual: Maj 0 / Maj 7)	{7; 5; 4; 2; 0} {9; 7; 5; 4; 0} {9; 7; 5; 4; 2; 0}
Min 0 / Min 4 (dual: Min 0 / Min 8)	{11; 3; 0} {11; 7; 3; 0}

Figure 4–53: Double-key asserting pitch class sets involving 0 (C) major and 0 minor. These sets are in the intersection of exactly two keys. The *duals* involve duplication of normalized sets, but the duals show the full set of keys that have productive intersections with 0 major and minor.

CHAPTER 5
Z-chains

5.1 Introduction

The structures in this chapter (Z-chains and Z-shapes) operate on melodies or sequential (monophonic) subscores. Their purpose is to take *low*-dimensional data and *increase* its apparent dimensionality. The intuition is that although we can picture a melody as a pitch-time *line*, we can also find internal processes and parallels that cut through the superficial linearity of the medium.

With Z-chains, we are trying to illustrate a way of thinking and of computing. Z-chains are perhaps the clearest illustration of the structural-analysis concept in this thesis. The algorithm works bottom-up, taking local shapes and linking them to form supershapes. Shapes and supershapes are built using low-dimensional projections, but rather than taking these as low-informational *summaries*, we keep the structures and their labels *on* the score as a new set of descriptive dimensions, with the melody as their intersection. We have complexified the picture. This doesn't give us any *answers* about the melody, but it gives us more ways of thinking about its parts.

Section 5.2 of this chapter reviews the literature on computational melodic analysis. Section 5.3 introduces Z-chains, providing an algorithmic definition, illustrations, and a comparison with a method from the literature. Section 5.4 discusses technical extensions to the Z-chain concept, and analyses properties of a natural Z-chain set.

In section 5.5 we introduce Z-shapes, a more "local" and somewhat simpler (and perhaps more intuitive) derivative of Z-chains. Section 5.6 is a summary and conclusion, tying back in with the existing literature and proposing future directions.

5.2 Review of the Literature

This section reviews the literature on the computational analysis of melody. A few main sub-areas are covered: *contour analysis*, *melodic similarity*, *segmentation*, *tree-structuring*, and *prediction*.

Contour analysis is conceptually closest to the Z-chain approach, since it is concerned with the *shape* of the melody in pitch-time "space." The literature on *similarity* is considerably more voluminous than the other topics, partially because of an annual competition that generates a lot of interest.[1] While the study of melodic similarity has its own discussion here, we find that it's a concept that turns up repeatedly with regard to the other categories – where, for example, a contour analysis or a tree-structuring is used to construct a notion of similarity, or similarity is used to construct a segmentation.

Descriptive methodology is most apparent in the literature on *contour*, while *segmentation* and *tree-structuring* are "structural" in the sense of looking at a score

[1] The Music Information Retrieval conference (ISMIR) issued a challenge task to produce a program that imitated the similarity judgments of a panel of experts. This challenge was run 9 times between 2005 and 2015.

in terms of its parts and their relations. The *similarity* concept involves the measurement of a notional distance between (sub)scores, *without* necessarily saying *anything* else about them (the *methods* for obtaining a measurement may or may not be descriptive). Likewise, the goal of "predicting" what comes next in a melody may not be based on description.

The literature here is drawn from a few different academic fields. The earliest (non-computational) work on melodic contour was done by ethnomusicologists, who considered songs as cultural artifacts. Some of the later work on melodic contour was done by music theorists studying works of modern art. Much of the work on parsing and segmenting is also done by music theorists, often with interest in European art music from the extended classical period (roughly 16th to early 20th Centuries), but also making use of the available folk-song corpora which are more extensive, reliable, and easier to work with than existing classical corpora. While musicologists focus on musical culture and music theorists are trained to produce technical interpretations of artworks, music information retrievalists use computation as their *primary* mode of investigation, and are interested in how to work with *music (corpora) as data*. The (computational) study of music has also been strongly influenced by the *cognitive* revolution of the late 20th Century, as well as the quantitative methodologies of recent scientific scholarship including the AI training and testing methodology of "ground truth."

5.2.1 Contour

"Contour" is the musicological term for the abstraction of relative pitch-height in melody. The concept was alive for ethnomusicologists by the early 20th Century.

By the 1960s and 70s, approaches were generally typological, describing melodies as sequences of small shapes such as *level*, *rising*, *falling*, *undulating*, *two-plane*, and *scalar*, or as more global shapes: *rising wave*, *falling line*, *arch-shaped*, *undulating* around a central pitch. At this stage, classification was not computational, and relied on hand annotation by musicologists.[2]

An early systematization, [Adams1976] considers the first, last, highest and lowest notes, categorizing melodic segments by the relative heights of these. A further set of *measurements* are proposed, such as the height of the first pitch as a percentage of the total range, the number of beats between the first pitch and highest pitch, etc. The goal is to provide a typology for melodic descriptions, where means and ranges of different values can be considered. An application is given, comparing songs from two different Native American tribes.

In the 1980s and 90s, music theorists worked to further systematize and generalize the study of melodic contour initiated by ethnomusicologists. [Friedmann1985] works with sequences of orientations (*up* and *down*, but not *be*),[3] and normalized pitch contours, (such that the lowest pitch is 0, the second-lowest is 1, etc.) The application is to transformational relations (e.g. inversion, rotation) between melodic fragments (segmented by hand) in atonal music by Schoenberg.

In [Morris1987], a melody is described by a 2-d matrix of orientations (up, down, and be). The Schoenberg "school set" of transformations (retrograde, inversion, and

[2] A review of this kind of work is found in [Adams1976].

[3] *Be* is a name for the orientation = or "same."

retrograde-inversion – but not rotation) are definitive of contour equivalence classes; a combinatorial analysis for small n is undertaken. The application here is also to Schoenberg.

[Marvin+1987] provides a similarity measure for orientation matrices based on proportions of shared values, and a similarity measure for normalized pitch contours based on subsequence matching. Applications are to the music of Schoenberg's students, Berg and Webern.[4]

[Morris1993] gives an algorithm to generate nested summaries of a melodic contour, with local maxima and minima at each level providing points of salience. The result is a detailed and visually informative sketch. Since this method is somewhat similar to the Z-chain concept, we make a more detailed comparison below (Section 5.3.4).

[Huron1996] summarizes the contour of a folk song or folk-song phrase by its first and final pitches, along with an average of all the pitches in between. This provides a classification of 9 basic 3-point contours, which are used for a quantitative study on the Essen collection of German folk songs, showing which contour types are more common, and that there is no evident correlation between the song contour and the contours of its phrases.

[4] Schoenberg's own published discussions of melodic contour consist of pitch-time "graphs" of melodies by Bach, Haydn, Mozart and Beethoven, so that their overall shape can be *seen*; there is no classification, measurement, or transformational analysis [Schoenberg1967].

[Quinn1997] uses a similarity measure on "averages" of sets of orientation matrices to determine "fuzzy" set membership for melodies (– the application is to segments from a piece by Steve Reich – a composer characterized by postmodern minimal formalism).[5]

[Schmuckler1999] uses Fourier analysis to estimate long-range and local contour oscillations in melody. These are used in an experimental to check the correlation of a similarity measure with the judgment of test subjects.

[Juhasz+2009] samples each melody at equidistant temporal points, and applies a *self-organizing map*[6] to organize the contours of folk songs by geometric similarity. Clustering relations within and between regional folk-song corpora are shown.

This brief survey shows a conceptual evolution from description, through algebraic systematization and formal transformation relations, toward an increasing interest in similarity with less focus on descriptive methods. The Z-chain method is primarily descriptive, and differs from many of the methods mentioned above in that it makes available a set of overlapping and interleaving partial (recursive) descriptions, offering many facets rather than a summary.

---

[5] "Fuzzy" set membership is determined by a measure of "how close" an item is to a definition for a set, along with a parameter saying how close an item has to be to be "in" the set.

[6] A dimensionality-reducing unsupervised neural net – the dimensionality reduction allows high-dimensional data to be visualized on a low-dimensional plane of summarized similarity.

5.2.2 Similarity

One of the main topics of current melody research is the definition and detection of *similarity*. Similarity is sometimes framed as a subgoal in service of another application: e.g. retrieval and classification, studying how folk songs vary over time and by region, or the articulation of a score into similar and contrasting parts in order to better understand its structure. Similarity recognition is often stated as an explicit analytic goal, with the premise that understanding similarity is an essential part of understanding music.

The conceptual basis contains a few key ideas. The two basic ways of thinking about similarity are *measuring* a distance between (projections of) two items, and finding projections under which the two items are identical or contain identical parts.

The ways of *measuring* tend to be general and standard (with the *projection* techniques being more specific to music): Supposing the melody is represented by a contour plot, two such plots can be compared using standard measurement techniques such as correlation distance, city-block distance, or euclidean distance [Jannsen+2015]. If the melody is represented by a symbol string, standard techniques include edit distances [Rolland2001, Jannsen+2015, Grachten+2005, Marsden2012], substring matching [Risk+2015, Silva+2016], and alignment optimization techniques including dynamic programming methods [Kranenburg2010] and dynamic time warping [Juhasz+2009, Gulati+2015, Silva+2015].

Similarity can also be measured by comparing *global* features of the melody [Eerola2007, Kranenburg+2013]. [Kranenburg+2013] tests 88 global features such as "average melodic interval," "amount of stepwise motion," "fraction of melodic intervals that are an ascending major sixth" (etc.), and computes which features are most discriminative for different tune families – but ultimately concludes that "local" (sequence-based) comparisons are more useful for estimating similarity than are global features.

Projections and Abstractions

[Cambouropoulos2001] gives a review of some of the basic decisions to be made in representing a melody as a sequence of pitches or pitch-intervals, with or without durations or duration-ratios. This includes normalizing with respect to chromatic or diatonic intervals.[7] [Gulati+2015] uses a tetrachord normalization, while [Kim+2000] experiments with different normalizations, simplifying contours to have 3, 5, or 7 types of intervals.

To make sequence-processing techniques more powerful and less dependent on individual choice of representation, many methods make use of *multiple* representations (or "viewpoints") [Conklin+2001]. These can be used as individual, parallel sequences, or in different tuple combinations. Some viewpoints don't include all notes of the melody, e.g. by focusing on specific meter places, such as quarter-note beats. These kinds of features can be used as meter-based reductions, or to further *specify* the alignment of segments with respect to meter. [Lartillot2004, Lartillot2005, Lartillot+2007] builds suffix trees with mixed specificity, showing how segments can

[7] I.e. equidistant intervals, or intervals within an asymmetric musical scale such as the major scale.

be repeated with respect to different subsets of viewpoint features. Multiple viewpoints are used for classification of folk-songs [Goienetxea+2016, Conklin+2011], prediction [Pearce2005, Cherla+2014], and motivic analysis [Conklin2010].[8]

Apart from defining initial representations for sequences, further abstraction techniques can be used to compare melodic segments. One common example is the problem of finding similarity between a simpler melodic segment and an "ornamented" or elaborated variation. [Ganguli+2016] takes the approach of "time averaging" – essentially smoothing or blurring the contour to accentuate the most important pitches. [Gulati+2016] includes a complexity weighting factor in the similarity measurement, so that high-complexity segments can be more easily matched to low-complexity segments. [Risk+2015] uses meter information to guess which notes are the most salient, and give a higher similarity weighting to to matching strong-beats. [Cambouropoulos2000b] addresses elaborations by allowing a single melodic interval to match with a sequence of intervals with the same sum ("filling and thinning"); [Pikrakis+2006] abstracts away repetitions and neighbor-notes, while [Cambouropoulos2000b, Adiloglu+2006, Knopke+2009, Buteau+2000] allow matches that are invariant under retrograde and inversion.

Other kinds of abstractions and analyses have been proposed as the basis for similarity measurements (– in general, it seems that almost any kind of interpretation

[8] A *motive* is a brief melodic (or musical) idea; typically one that is repeated and varied in a score. A motivic analysis is a discussion of a score focused on motives (e.g. identifying, relating, interpreting them).

could be used as the basis of a similarity comparison). [Valero+2002] uses meter information to (deterministically) structure pitches as a *tree*, rather than a sequence. A tree-edit distance is used to measure similarity. [Tojo+2013, Matsubara+2014] make a different kind of parametrically controlled tree structures (based on [LerdahlJackendoff1983], and discussed below), and develop a similarity measurement for these trees based on edits and elaborations. [Grachten+2005] parses a melody using Narmour's "implication/realization" model ([Narmour1990]). In this model, three-note local shapes are categorized and labeled. The result of the parse is a sequence of symbols, each covering a segment of the melodic pitch-sequence. Similarity is then measured using an edit distance on the symbol-sequence.

Motive Significance

Since there are a powerset of possible subscores, in many applications an overabundance of possible *motives* (repeated or varied ideas) are found. In order to make the analysis more tractable and more informative, it's often desirable to be able to highlight the most interesting or *significant* motives.

Some notions of significance are "a priori" (– the motive is a "good" paradigm; significance is based on the characteristics of the motive itself), and others are "a posteriori" (– the motive *occurs* in a score or corpus in a way that suggests its importance) [Cambouropoulos2000b].

"A priori" significance involves a description of what is "motive-like," including concepts of temporal *compactness* [Meredith+2002] and reference to (other) Gestalt principles such as figure/ground [Lartillot2004], immediate repetition [Cambouropoulos2003], and an analysis of the redundancy inherent in embeddedness (subset relations) [Lartillot+2007].

"A posteriori" significance can be conceived in a few different ways, including *coverage*, *typicality*, *distinctiveness*, and *surprisingness*.

A simple means of obtaining a "significant" set is to make a greedy cover of a score [Meredith2016]. This corresponds with the principles of favoring those paradigms which are longer, more frequent, and less overlapping [Cambouropoulos+2000, Cambouropoulos2003, Adiloglu+2006]. [Silva+2015] defines significant excerpts as those that occur most frequently in a song, using these to identify different versions of the same song.

"A posteriori" methods can also be used on a corpus. [Knopke+2009] uses simple counting techniques to determine which motives are "common" in a corpus. [Conklin+2001] builds a zeroth- and first-order Markov model of a corpus, and judges significance of each motive found in contrast to how likely it is to be generated by the model – a motive that occurs more often than "expected" by the model is more significant. [Goienetxea+2016] likewise uses a zeroth-order Markov model to determine the "interestingness" (surprisingness) of motives in a corpus, and then determines "similarity" of two scores based on the summed "interest" of their common motives.

[Conklin2008, Conklin+2011] use "anticorpora" – a motive that appears more often in corpus (or opus) A than in a comparison corpus (or opus) B is considered to be *distinctive* for A. [Gulati+2016b] uses topic-modelling techniques to determine *characteristic* melodic segments for different rāgas, based on the frequency of their presence (and absence) in a corpus.

[Juhasz+2009] describes and compares the kinds of contours that are "typical" of 22 different folk-song cultures by plotting their occurrence on a self-organizing map, and observing where clusters occur.

Similarity and Compression

The concept of *compression* comes up a few times in the literature on similarity measurement. [Cilibrasi+2004] uses bzip2 string compression on MIDI files to calculate "distances" between pairs of scores: the bzip2 algorithm derives a compression-dictionary for each score, and then the distance between two scores can be conceived as a function of the amount each score can be compressed using the dictionary of the other.

Another take on compression occurs in [Meredith+2002, Meredith2016]. A geometric approach is given, in which note-onsets are represented as points in a 2-d space, and (exact) repetitions are detected as subsets of points that are invariant under translation (– this method works on non-monophonic scores). In order to measure similarity, a greedy covering of geometric shapes is used as a dictionary to "compress" a score (by describing it as shapes * translators), and compression using the shapes from a different score predicts similarity of the scores (– the greedy covering acts an estimation of "significance" of shapes, and the "compression" algorithm quantifies the overlap in significant shapes). In [Loubouin+2016], this geometric compression method is compared with general purpose compression algorithms, and while traditional compressors compress *more*, the geometric compression method makes better predictions of human similarity judgments.

[Pearce+2017] discusses compression, and offers another way to measure relative information content between pairs of scores. The IDyOM model (a multi-viewpoint, variable-order Markov model, described in Chapter 2 of this thesis) is used to "learn" a model from one piece of music, and then use the model to express the shared information content (and therefore similarity) in terms of the summed surprisingness of the second piece with respect to the model.

Classification and Organization

Methods are used to classify or categorize motives, or to otherwise organize their relatedness. In this literature, the classification is often by nearness or clustering of *similarity measurement* – a one-dimensional, *non-metric*, heuristic quantity. *Non-metric* means that the triangle inequality doesn't hold: i.e. if melody A and melody B are each one similarity unit away from melody C, we can't be sure that A and B are no more than two similarity units away from each other.

A simple approach: [Risk+2015] offers retrieval of related tunes *ranked* by similarity to the query.

[Cilibrasi+2004] uses a randomized method to build "phylogeny" trees, which branch to show similarity and difference. [Crawford+2001, Buteau+2000] show progressive similarity, illustrating how a motive can "evolve" over the course of a piece of music.

In order to induce categories from pairwise similarity measurements on a set of melodic segments, a network similarity threshold can be determined that maximizes a function of similarity among pairs in the same group, while minimizing similarity between groups [Cambouropoulos+2000, Gulati+2016b]. Similarly, [Pinto2012] uses a spectral method to induce a clustering on a matrix of pairwise similarity measurements.

Another approach to clustering is to define a set of representative paradigms (either automatically or by hand), and then to relate each motive found to the closest representative [Silva+2015, Kranenburg2010]. [Gulati+2016, Goienetxea+2016] train on a set of labeled data, and then take the labels of the k-nearest-neighbors as a classification for a new item.

The use of self-organizing maps [Juhasz+2009] to cluster and relate melodic contour is not clustering by a similarity *measurement*, but by the higher-dimensional contour vectors.

[Wattenberg2002] generates "arc diagrams," visualizing (exactly) repeated subsequences connected by translucent arcs over a timeline. Arcs on multiple scales are used to construct a visual sketch of motive organization on the score.

Ground Truth and Evaluation

Much of the literature refers the concept of "similarity" to a set of human-labeled data, which is used to design and test the algorithms. In particular, the MIREX challenge is a benchmark for many researchers, who try to achieve the ground-truth of similarity as labeled by human experts.

[Kranenburg2010] designs a multi-dimensional similarity labeling scheme, having experts evaluate the similarity of folk-songs from several points of view; [Eerola2007, Schmuckler1999] likewise ask humans for their judgments, but framed as an experimental psychology procedure rather than a request for an "expert" opinion.

[Mullensiefen+2007] runs a psychology experiment for "expert" listeners (musicology students), testing the results against several similarity measurements.

While most researchers aim to maximize the correspondence of their results to their ground truth target, [Risk+2015] finds that unexpected results suggest areas of interest for further musicological study.

[Buteau+2000] suggests that mathematical definitions of similarity carry more definite meaning than heuristic approximations to human intuitions.

[Marsden2012] interrogates the practice of using human similarity labels, pointing out that declaring how similar two melodies are is a subjective and *creative* human act – and one that is somewhat musically unnatural. He points out that there are many ways of thinking about musical similarity – this is evidenced in e.g. [Mullensiefen+2007], where many dimensions and points-of-view are considered, but all with the goal of finding the *best* way to crush all considerations into a single dimension. [Marsden2012] proposes that the complexity, contextuality, and creativity of musical interpretation don't benefit by dimension-crushing analyses.

5.2.3 Segmentation

Segmenting a melody is a goal in some of the literature: the melody is split into phrases (and sometimes further into subphrases) such that phrases don't overlap.[9]

[9] Common music theoretic concepts such as *elision* and *pivot* (both involving overlap) are not handled by this paradigm. An alternative model, *grouping*, is compared in [Lartillot2004].

The *local boundary detection model* [Cambouropoulos2001b] works by using heuristic rules (concerning similarity and proximity) to estimate the discontinuity between each pair of notes in a melody. Local peaks in discontinuity are considered to be local boundaries, with the strength of discontinuity indicating the strength (or hierarchical position) of the boundary.

Many of the melodic similarity researchers cite segmentation as a potential use of similarity detection. [Cambouropoulos2006] uses repetition (similar melodic subsequences) to determine boundaries, with the idea that a motive, heard as a melodic unit, starts a segment, in "parallel" with its repetitions. Results are compared with the local boundary detection model. [Ahlbäck2007] develops a segmentation scheme with an emphasis on similarity and metrical parallelism, and secondary principles including (dis)continuity and symmetry.

[Temperley2001] segments by preferring to segment at a long duration or pause, to have phrases close to a given length, and to have phrases that begin at metrically parallel places. Dynamic programming is used to optimize a set of boundaries.

[Bod2001] uses a data-based approach, modelling the phrase boundaries in the Essen corpus of German folk songs (which are given in the database). A Markov grammar technique is used to estimate the likelihood (i.e. frequency-in-corpus) that a given suffix is followed by a phrase boundary, along with a factor for total phrase length.

The IDyOM model has also been used to produce a segmentation [Pearce+2010], by hypothesizing that boundaries could occur at points of "expectancy violation" (i.e. high surprisal according to the Markov model).

The main idea of much of this research is that there may be *normative* or *obvious* points at which to segment a melody. Although these papers are by musical thinkers and not psychologists, there is an emphasis in their writing on *cognition* and *perception*, rather than creative, cultural, and musical-contextual (e.g. song text, performance indication) treatment. However, [Thom+2002] demonstrates with an experiment that segmentation is "ambiguous" – i.e. that there is *not* one obvious answer agreed upon by musicians. Some of the tunes tested are from the Essen corpus.[10] The phrase marks in the Essen corpus don't show how a continuous tune is segmented, but show how performance gestures and textual phrases help to create the musical picture – all of which are a product of a specific musical culture. Using a model of this corpus as a stand-in for a cognitive model is therefore conceptually slippery.

5.2.4 Tree Structuring

As in the segmentation literature, the methods for parsing a melody into a *tree* structure tend to conceptualize a perceptual or cognitive basis for this operation, inspired by the Chomskyan zeitgeist as well as by the nested music-analytic structure proposed in the early 20th Century by Heinrich Schenker.

[LerdahlJackendoff1983] (*A Generative Theory of Tonal Music*, known as "GTTM"), co-authored by a music theorist and a linguist, offers a (non-computational) system of rules for parsing a score into a binary tree in which each split is weighted to the right

[10] A casual attempt to guess where the phrase marks are in a few songs from the Essen corpus suggested to Dr. Handelman and me that these aren't obvious.

or left, indicating which side is "stronger." There are necessary "well-formedness" rules, as well as "preference" rules which need not be fulfilled, but can help make decisions. The preference rules may conflict and so many interpretations are possible according to this method (– helping to account for the variety of opinions and interpretations of how a score could be parsed). Because of this ambiguity, the large number of rules, and the informality of many of the terms in the book, the system has been the subject of less computational research than its popularity among music theorists would predict.

One group, however, has succeeded in formalizing and parametrizing many of the preference rules, automatically generating trees for melodies [Hamanaka+2005] and automatically tuning the parameters [Hamanaka+2007]. The software can also be used interactively [Hamanaka+2009] to build custom interpretations. [Groves2016] uses a corpus of melodies parsed into trees by humans (according to the *GTTM* method) to induce a probabilistic context-free grammar.

[Gilbert+2007] offers a method of parsing a melody into tree structures by using probabilistic (i.e. weighted) context-free grammar rules for melodic shapes with meter information (e.g. describing passing and neighbor notes). Since there may be more than one possible parse, the weightings express the "probability" of different parses. [Marsden2001] uses a more extensive set of grammatical rules to parse a melody into a graph of elaborations. These graphs can also be used to modify or generate melodies.

5.2.5 Prediction

Another kind of approach to melodic analysis is *prediction*. While many similarity projects try to "predict" which melodies are most similar (to human judgment), *prediction* models try to predict the music directly (e.g. given the beginning of a score, how will it continue).

One example of this is the IDyOM Markov model ([Pearce2005] described in Chapter 2). Other projects build models based on similar conceptual premises of memory-based prediction, but use different algorithms to implement these models. For example, [Cherla+2014, Cherla+2015, Boulanger-Lewandowski+2012] build different kinds of neural networks. While the IDyOM project is based on the music-perception concept of expectation, and judges its success by how the results match up with human predictions of the music the neural-net projects judge their success on their ability to predict new scores.[11] One way of thinking about this is that in the IDyOM concept, a "surprising" melodic moment is expected to surprise people, but in the neural-net model, a "surprising" melodic moment means that there's something the neural-net model hasn't accounted for – and in an ideal model this would be minimized.

What these models have in common is that they build predictors that use "known" music to predict how "unknown" music will go. While they have long-

[11] The perceptual-expectation concept is also invoked by [Grachten+2005], who use Narmour's descriptions of short-term melodic expectations to parse a melody into symbols – but the symbols are not used here to model expectation, but to make similarity judgments using an edit distance on the symbol string.

and short-term memories, the structure as described is one of linear entailment, that X having happened *predicts* that Y will happen – their descriptions are of correlations that recur in a corpus, rather than of the shapes or stories of individual pieces. The common language of this kind of procedure is that the musical data is being "modelled," but it's unclear in what sense a "model" *of a score* can result – i.e. this is not a *descriptive* method, but a statistical one.

5.2.6 Contrast with Z-chains

The methods to be described in this chapter contrast with much of the existing research in a few different ways. While Z-chains and Z-shapes (like any description) can be used to say something about similarity, the immediate interest of a Z-chain analysis is in describing the shape of *one* melody, without concern for *other* melodies (– neither from a similarity point of view, nor from the point of view as a context of memory with which to *predict* things about the melody). Many of the projects for similarity detection, segmentation, and tree-structuring try to imitate a labeled ground-truth of human interpretation, while Z-chains involve deterministic geometric descriptions. Segmentation and tree-structuring both propose a hierarchical structure for melody, while Z-chains show shapes and processes that overlap, cutting across one another from different points of view. Much of the literature tries to compare, segment, structure, or predict the melody *without* first *describing* particular shapes (or processes, patterns, events, etc. *Taxonomies* have been avoided since the mid 20th Century). Z-chains and Z-shapes give an algebraic description language for a geometric typology that's *pervasive* in all data.

5.3 Z-chains[12]

5.3.1 Algorithmic description

Z-chains provide an analysis of (melodic) "contour" or *shape*, entirely based on *orientation*. *Z-chains* are a structural analysis on a *sequence* of totally orderable terms – terms that generate the *orientation* relations $\{<, >, =\}$, called *up*, *down*, and *be*. Without loss of generality, such a sequence can be represented by a sequence of integers. A *chain* is a (sub)sequence in *one* orientation, and a *Z-chain* is, recursively, a chain of chains or chain of Z-chains, in which the lower-level chains or Z-chains all have the same orientation.

In music, orientation is available on pitch, duration, loudness, cardinality, etc., and on constructed or heuristic measurements like density or dissonance. These parameters don't necessarily give a *sequence*, but we assume that one has been generated from the score in advance of a Z-chain analysis.

Our primary application will be to melodic pitch, with this example standing in for the general principle. With orientation on pitch, abstracting away *absolute* pitch, Z-chains address the up-and-downness – the *zigzagginess* – of a melody.

A *chain* is a consecutive subsequence in *one* orientation – it is therefore the *simplest* orientation shape. We use the *chain* as a structural base-case. We hypothesize the musical importance (or basic-ness) of chains in *one* orientation, delimited by *changes* in orientation (– as we have seen, orientation or contour is a common

[12] Say it American style: Zee-chains. Z stands for zig-zag.

projection of melody in the literature). This corresponds to the basic notion of musical pitch "going up and down," and can be generalized to "getting *more* or *less* x" for arbitrary x.

The recursive step is motivated in the same way, with the additional notion that *local* musical structures (e.g. chains) can be chained *like-to-like*, *across temporal gaps* (in which other material intervenes). The *perceptual* salience and coherence of these higher-order chains varies, but even when they are distant and disparate enough to be "non-obvious," they can highlight large-scale structures that tell a different kind of musical story.

The *chaining algorithm* finds all (maximal) chains in a sequence. Pseudocode is given in Figure 5–1. In one pass through a sequence, the chaining algorithm finds all contiguous subsequences that just go *up*, or just go *down*, or consist just of *be* (repetition).[13] *Any* sequence of orderable terms is *exhaustively* composed of chains overlapping at their endpoints; these intersections are called *pivots*.

We take the sets of chains in *each* of the three orientations as three separate "dimensions," in the sense that the chains are partitioned by orientation and processed separately in the recursion step – chains in the same orientation are *parallels*. In the recursion, the idea is to take *similar* things together, rather than taking things

[13] More formally: given a sequence $(t_1, t_2 ... t_n)$ and a function F such that $\{F(t_1), F(t_2) ... F(t_n)\}$ are linearly orderable, a *chain* is is a consecutive subsequence $c = (t_i, t_{i+1}, t_{i+2} ... t_{i+j})$ such that $\forall t_k \in c$,
$F(t_k) < F(t_{k+1}) \leftrightarrow F(t_{k+1}) < F(t_{k+2}) \wedge$
$F(t_k) > F(t_{k+1}) \leftrightarrow F(t_{k+1}) > F(t_{k+2}) \wedge$
$F(t_k) = F(t_{k+1}) \leftrightarrow F(t_{k+1}) = F(t_{k+2})$.

together because they are sequentially conjunct (e.g. as in n-grams). The principle of the recursion therefore will be to *chain like to like*.

Given a sequence of chains in *one* orientation, run the chaining algorithm again. Since the initial chaining orientation relation may not be well defined on chains, a new orientation relation must be defined for the recursive process – the *Z-relation*. Possibilities include top (i.e. max), bottom (min), span (top minus bottom), and cardinality. This relation will be retained for the rest of the recursion.[14] The recursive algorithm is the *Z-chain algorithm* (– pseudocode in Figure 5–2).

The result of the first recursion step (running the chaining algorithm on a sequence of *chains* of the same orientation) is a set of *second-order* Z-chains, with second-order orientations – e.g. a *Z-chain down-up* in feature *top* is a sequence of chains down with respective top terms going up. The chain of top terms going up is a second-order *superchain*. The base chains could also be called first-order Z-chains, and the terms in the initial sequence could be thought of as zeroth-order Z-chains.

In keeping with the principle of "chaining like to like," the second-order Z-chains are again partitioned by orientation and the chaining algorithm is run again. Since each iteration of the chaining algorithm takes a sequence of structures and chains some of them together to form higher-order structures, fewer structures are obtained at each successive step. When only one Z-chain remains in some orientation – a

[14] Mixed-relation Z-chains are also possible.

top-order or highest-order Z-chain – that branch of the process naturally terminates. The number of iterations to find all Z-chains in a sequence is quadratic.[15]

5.3.2 Illustrations

To get an intuition for what Z-chains look like and what kind of shapes they can show, we take a look at "Happy Birthday," a short and familiar melody (Figure 5–3). Each staff in the figure shows a highest-order Z-chain on the melody, in Z-relation top-pitch (shown in red and orange, with beams above the staff) or bottom-pitch (in blues, with beams below the staff). Here are a few observations about "Happy Birthday" using Z-chains.

The first four staves of Figure 5–3 show Z-chains with base orientation *up*. The first staff, *up-up-up* in top-pitch, shows how the opening four-note figure of the melody, a Z-chain *up-up*, is subsequently continued and expanded, with a longer superchain (two chains are followed by three), and a higher top-pitch. The third staff shows a Z-chain on *bottom*-pitch, containing the same set of base chains as those in staff 1. The Z-chain is of second-order, with a super-orientation *be*. The bottoms of these chains form a grounded and stable counterpoint to the bouncing motion of the tops. In the second half of the tune, the bottoms climb up (staff 4) while the tops

[15] Proof that the Z-chain algorithm (pseudocode in Figure 5–2) is quadratic: The chaining algorithm takes linear time since it runs in one pass, marking changes of direction. Therefore, if the Z-chain algorithm runs the chaining algorithm a linear number of times, then the Z-chain algorithm is quadratic. Each time the chaining algorithm runs on a sequence with length $n > 1$, the number of output chains is $< n$. Since the number of things to be chained decreases each time the chaining algorithm runs, it can run at most n times before the Z-chain algorithm terminates.

CHAINING ALGORITHM:

Input:

- S: sequence of terms $(t_1, t_2 ... t_n)$ of any type (type `term`).
- F: function of type `(term -> integer)`

Method:

1. **define** get_ori(x,y) {
   if $x < y$ **then** *up*
   else if $x > y$ **then** *dn*
   else *be*}
2. **let** $chains\_found = empty\_list()$;
   let $current\_chain = [t_1; t_2]$;
   let $current\_ori = get\_ori(F(t_1), F(t_2))$;
3. **for** $i = 3$ **to** n **do**:
   let $new\_ori = get\_ori(F(current\_chain.last), F(t_i))$;
   if $new\_ori = current\_ori$
   then concatenate t_i onto $current\_chain$
   else
   (concatenate $current\_chain$ onto $found\_chains$;
   $current\_chain := [current\_chain.last; t_i]$;
   $current\_ori := new\_ori$)
   done;
4. concatenate $current\_chain$ onto $found\_chains$;
   return $found\_chains$

Output: A sequence of *chains* $\mathrm{C} = (c_1, c_2, ... c_n)$ such that:

1. $\forall c_k \in \mathrm{C}, \exists i$ such that $c_k = (t_i, t_{i+1}, t_{i+2} ... t_{i+j})$ – each chain is a *consecutive subsequence* of S.
2. Each chain has a single orientation with respect to the function F:
   $\forall c_k \in \mathrm{C}, \forall t_i, t_{i+1}, t_{i+2} \in c_k,$
   $F(t_i) < F(t_{i+1}) \leftrightarrow F(t_{i+1}) < F(t_{i+2}) \wedge$
   $F(t_i) > F(t_{i+1}) \leftrightarrow F(t_{i+1}) > F(t_{i+2}) \wedge$
   $F(t_i) = F(t_{i+1}) \leftrightarrow F(t_{i+1}) = F(t_{i+2})$
3. Chains are of maximal length – extending the subsequence in either direction does not produce a valid chain.

Figure 5–1: Chain finding algorithm (pseudocode).

Z-CHAIN ALGORITHM

Input:

- S: sequence of terms $(t_1, t_2 ... t_n)$ of any type (as `term`).
- F: function of type `(term -> integer)`
- Z(F): function of type `((term -> integer) -> sequence -> integer)` – we use the same F as above. W.l.o.g this can be a (recursive) function on Z-chains (as structured sequences). Example functions: top (i.e. max), bottom (min), span (max - min), cardinality.

Method:

1. Run `chaining_algorithm` (S,F) – returns a sequence of chains C.
2. Partition C by orientation (preserving sequence order for each subsequence), obtain $C_<, C_>, C_=$.
3. For each C_{ori},
   (a) `if` $|C_{ori}| \leq 1$, save contents of C_{ori} for output as a top-level Z-chain; branch terminates.
   (b) `else if` $|C_{ori}| > 1$,
      i. run `chaining_algorithm` $(C_{ori}, Z(F))$ – returns a sequence Z of Z-chains (i.e. chains of chains...).
      ii. Recurse to step 2 with Z as C.

Output: Set of top-level Z-chains.

Figure 5–2: Z-chain finding algorithm (pseudocode).

Figure 5–3: Z-chains in “Happy Birthday” (originally “Good Morning to All” by Mildred Hill).

descend (the final second order Z-chain in staff 2 is synchronized with the Z-chain in staff 4), giving a second-order converging counterpoint. All of the chains up in "Happy Birthday" have the same length (two notes).

The bottom four staves of the figure show Z-chains with base orientation *down*. Staff 5 shows the same supershape as staff 1, from another perspective. From the point of view of chains down, the chain *lengths* are growing in synchronization with the top-pitch supershape. Staff 6 shows how this synchronization is continued with the second-order top-pitch descent in the second part of the tune. In staff 6: while the predominating top-pitch motion in the second half of the piece is different from that in the first part, there is nonetheless a parallel. In fact the first second-order Z-chain here is an exact transposition of the last two chains of the piece. Staves 7 and 8 show the bottom-pitches from the point of view of chains down, showing that the *be* superchain from the beginning is reflected at the end of the tune. Together, the Z-chains on chains down cover all of the notes except the first – unlike the chains up, which are gapped.

Z-chains can be arbitrarily big, working over large-scale forms as well as on short melodies. Figures 5–4, 5–5, and 5–6 show some large Z-chains on the Presto for solo violin by J.S. Bach. Oriented structure is sketched as parallels on multiple larger-scale levels, sometimes with long gaps between parallel instances. These can show partial formal parallels and resonances; the gaps signify a zone of *absence* of the relevant orientation.

Figure 5–7 shows a different way of looking at Z-chains on the Presto. A subset of Z-chains creates a compact contour-sketch on tops and bottoms of chains. *Compact*

Z-chains are those which do not skip chains in their own base-orientation.[16] In the figure, *superchains* at all recursive levels are drawn with dotted lines – red for top-up; orange for top-down; dark blue for bottom-up, light blue for bottom-up, and black for be. The number of superchains present on top or bottom at any given time shows the depth of the compact Z-chains at that point – if there is only one contour line, it is second-order (i.e. one level of superchain), further tracings show higher-orders of compact Z-chains.

5.3.3 What are Z-chains?

Each Z-chain is a *partial motion-sketch* for the melody. Z-chains are a way of taking a sequence apart to find subsequences that each afford a recursively simple point of view. These subsequences don't *partition* the sequence, since each term can play multiple roles in a complex, multidimensional shape. While the principles that build Z-chains include contiguity, hierarchicality, and parallelism, the total set of Z-chains foils these principles to provide a dimensionally deep, intricate picture which is nonetheless made of simple, well-defined components.

Z-chains aren't attributes or measurements of a melody, they are *parts* of a melody – not independent, atomic parts, but *dependent*, interactive parts showing ways in which a melody is *not* partitionable – ways in which a sequence has structure *beyond* the concatenation of its elements – because it contains lines of force which cross one another. It is composed not of units, but of dimensions, or rather directions in motion.

[16] Compactness is formalized and generalized later in this chapter.

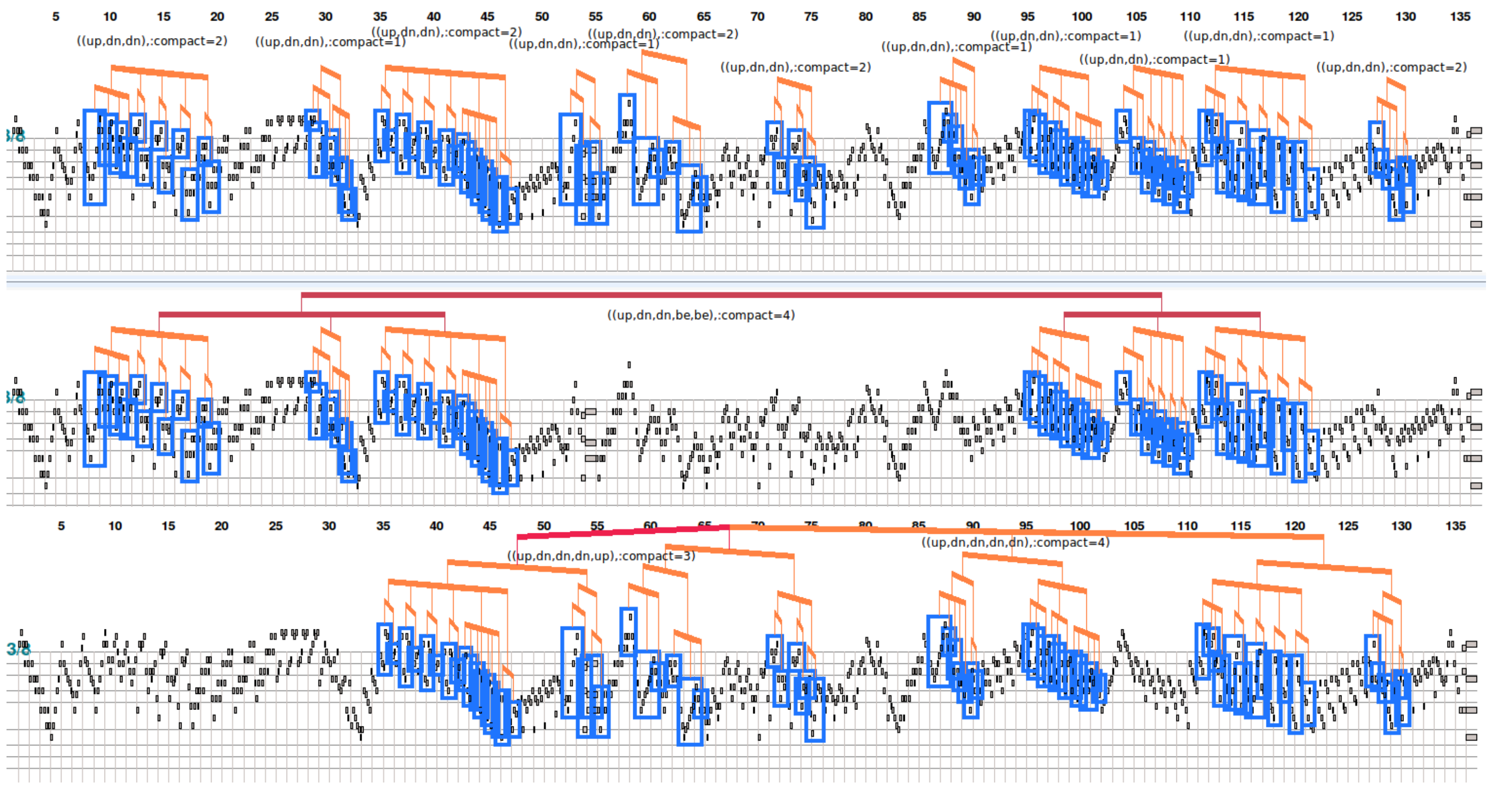

Figure 5–4: Large scale Z-chains on the Presto by J.S. Bach. Z-chains up-down-down (Z-feature *top*) on the top panel are chained on the lower panels. Groupings show recursively parallel orientations on different parts of the piece.

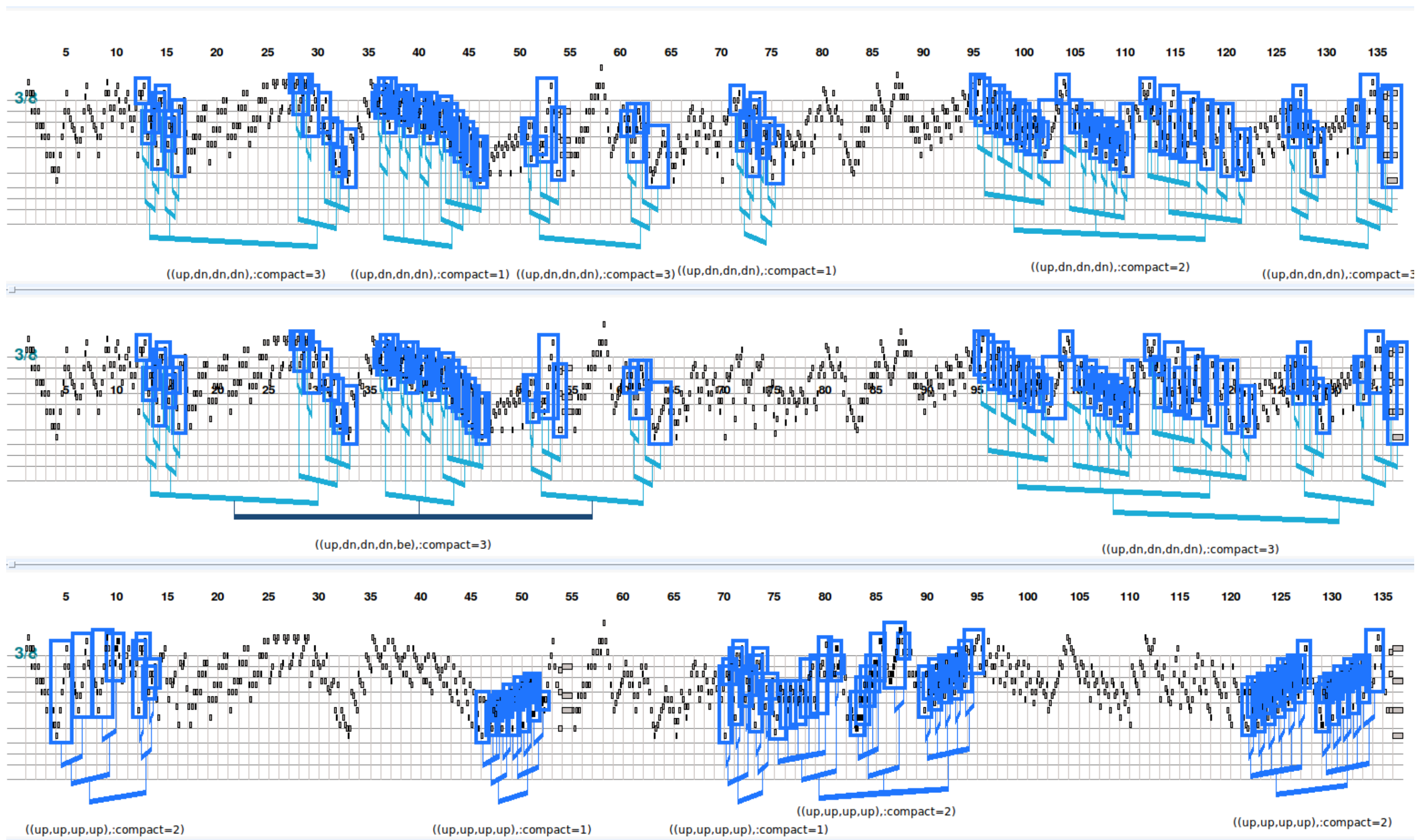

Figure 5–5: Large scale Z-chains on the Presto by J.S. Bach. Z-chains up-down-down-down (Z-feature *bottom*) on the top panel are chained on the middle panel. Bottom panel shows contrasting shape up-up-up-up.

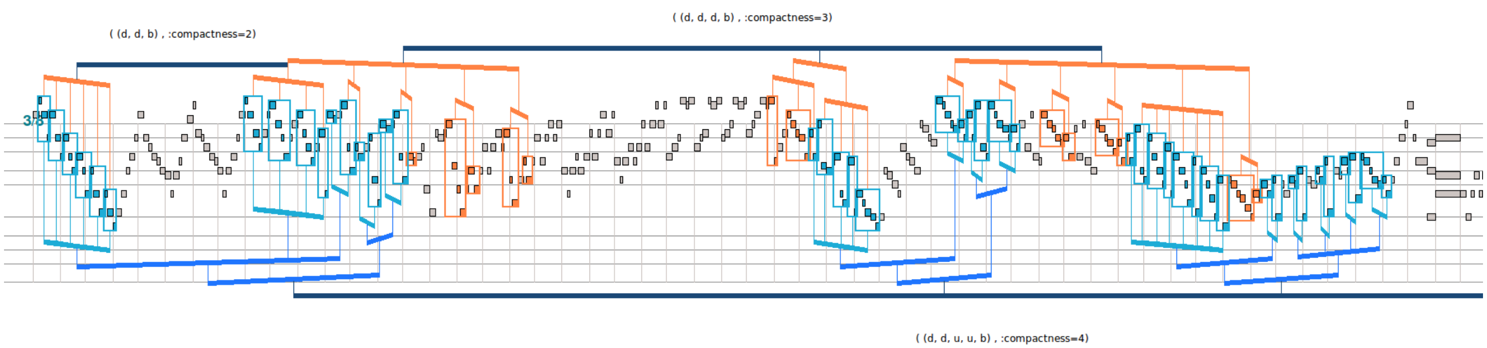

Figure 5–6: Large-scale Z-chains on the Presto for solo violin by J.S. Bach (excerpt). Shown are Z-chains down-down-be and down-down-down-be on top pitch (beams going upward), and down-down-up-up-be on bottom pitch (beams going downward). Each Z-chain shows long-range parallels. The Z-chains on top and bottom pitches are *sometimes* synchronized.

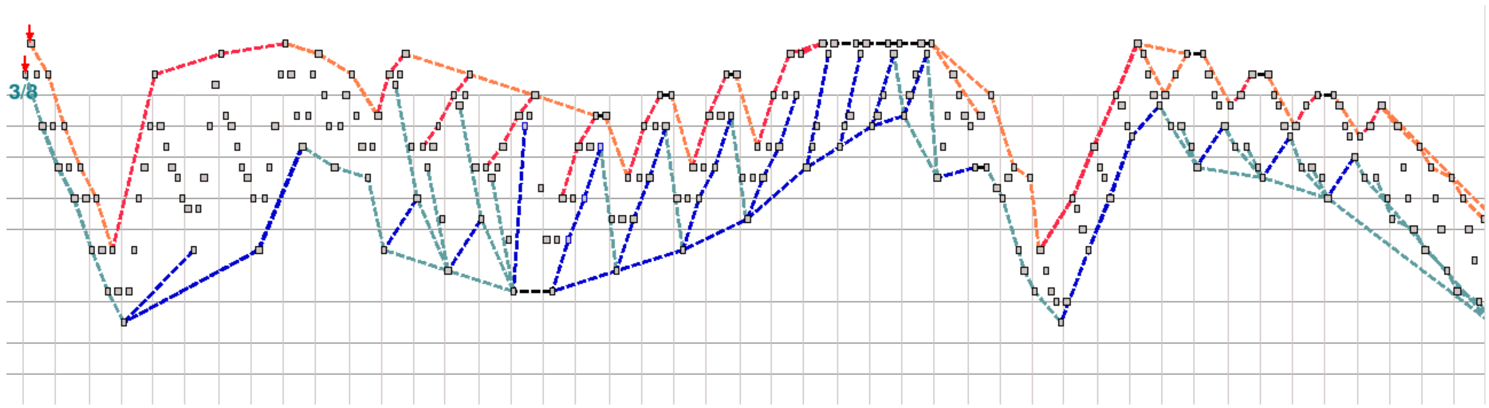

Figure 5–7: J.S. Bach, Presto for solo violin (excerpt). Z-chain superchain sketch showing compact contour on tops and bottoms. *Compact* Z-chains are those which do not skip chains in their own base-orientation. *Superchains* at all recursive levels levels are drawn with dotted lines – red for top-up; orange for top-down; dark blue for bottom-up, light blue for bottom-up, and black for be. The number of superchains present on top or bottom at any given time shows the depth of the compact Z-chain contour.

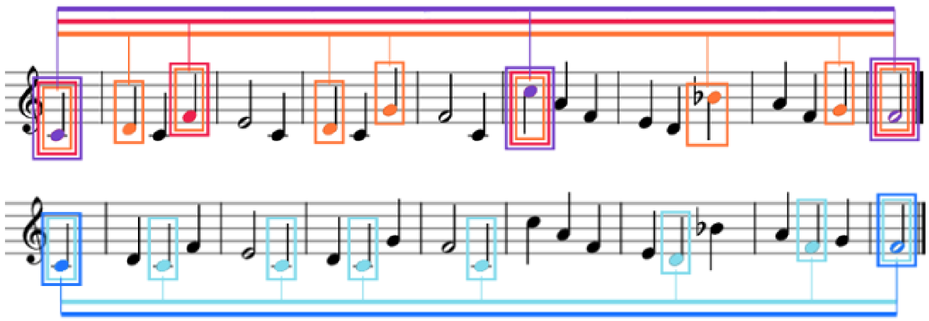

Figure 5–8: Morris' contour reduction algorithm on "Happy Birthday"

Z-chains illustrate one way in which *a sequence of numbers is structurally multi-dimensional*. Musically speaking, the pitch-time Cartesian grid, here reduced to the "1.5-D" pitch-event sequence, is sufficient to *denote* but not to *describe* a melody – the sequence is not *two-dimensional*, a concatenation of magnitudes fluctuating over time: it's an architecture of parallels, developments, recollections, and reversals.

5.3.4 Z-chains and Morris' Contour Reduction Algorithm

Most previous work on melodic contour is very different from Z-chains; there is one method, however, that is superficially similar enough to make a suitable foil. Morris' contour reduction algorithm [Morris1993] works by taking the local maxima and minima of the melody, then recursing to take the maxima of the maxima and minima of the minima. The result gives multiple hierarchical levels, with each level summarizing the previous one. Figure 5–8 shows the result on "Happy Birthday."

At each level (most often taken at the highest level), Morris' reduced contour can be *normalized* to afford comparison with other contours. This is done by taking the union of the maxima and minima, and normalizing these starting at zero for the lowest note and using successive integers. In "Happy Birthday," taking the highest level, we get the first note, the central high note, and the last note, giving a normalized contour of $< 0, 2, 1 >$. If we took the second-highest level in "Happy Birthday," offering a bit more specificity, we would add the fourth note of the tune, giving a normalized contour of $< 0, 1, 2, 1 >$. Taking the lowest level of reduced contour, we would end up with almost all of the notes in the tune.

We discuss a few points of comparison between Morris' contour reductions and Z-chains. While Morris's algorithm and Z-chains both concern maxima and minima, the methods turn out to be *essentially* different, with neither method reducible to a special case of the other.

Morris' result gives several nested levels of contour, each *smoother* and describing *less* structure than the last. The goal seems to be to *reduce* a complex structure (the melody) get a kind of "big picture" top-down sketch. Information is deliberately removed at each level, in order to obtain a simpler summary. In contrast, Z-chains are *less* smooth and more zig-zaggy at each higher level, describing more and more structure. Z-chains *start* with simplicity (the chains) and work toward complexity, *increasing* the amount of available information – a bottom-up approach to describing structure, resulting in a fractal view where the *inside* counts.

Z-chains describe *recursive* orientation, in which e.g. a third-order Z-chain is a *chain* of second-order Z-chains. Morris contours are *not* recursive (although his algorithm is) – "reduced contours" are given in the form $< 0, 2, 1, 3 >$, which can be read as "first go up, then go down but not as far down as where we started, then go up even higher than before" – this is sequential and linear, describing *one* level of zig-zagging at a time. A simultaneous reading of Morris contours on different hierarchical levels still does not produce a recursive structure; it produces a linear structure in which some zig-zagginess at lower levels is abstracted away at higher levels.

The structures Morris obtains at each level are hierarchically nested in one another. While *each* Z-chain is a hierarchical structure, the total set of Z-chains are related to each other in a non-hierarchical and complex way. Whereas finding "parallels" (i.e. matching reduced contours) within a piece with Morris' algorithm requires pre-segmentation of the piece, Z-chains, since they work bottom-up, find parallels anywhere in the piece without pre-processing, also finding different sets of parallels that would "conflict" if we assumed a segmentation model. Furthermore Z-chains not only *find* parallels, but recursively *structure* the relations between these by chaining them together into higher-order Z-chains.

In sum, Morris offers a method for obtaining a fixed, simplified, linear perspective on a contour, to facilitate identification of different contours through their reductions. Z-chains provide a non-linear set of structured perspectives as partial motion sketches, not reducing to a single "answer," but instead *increasing* the amount of information available about the structural relations within a melody.

5.4 Structure of Z-chains and Z-chain sets

5.4.1 Gaps and compactness

Gappedness

Z-chains of any order > 1 may be *gapped* with respect to the original sequence – they can skip elements. For example, on the sequence (1,3,5,2,1,3,4), we get chains up (1,3,5) and (1,3,4), chain down (5,2,1) and second-order Z-chain in top up-down

((1,3,5),(1,3,4)). The second-order Z-chain skips the 2 because it doesn't participate in a chain in the orientation *up*. *Any* number of elements may be skipped by this process. In particular, a Z-chain skips the inside of any chain or Z-chain in any opposing orientation.

Gaps in Z-chains also bear orientation information about the sequence. For example, it's possible to find a Z-chain up-up-up at the beginning of a long sequence and again at the end; these chain together with an arbitrarily long gap. The gap in the Z-chain asserts a negative property of the sequence, since no Z-chain up-up-up can occur in the gap.

Compactness

Compactness is defined as a property of Z-chains in partial (i.e. dimensional) opposition to gappedness. More compact Z-chains can facilitate a tidier local analysis (at the expense of long-range parallels) and are generally easier to reason about.

A Z-chain that doesn't skip any elements is *zero-order compact*. A Z-chain that doesn't skip any chains *in its own base orientation* is first-order compact, and so on. For example, turn once again to Figure 5–3, the Z-chains on "Happy Birthday." Staff 1 shows a Z-chain that is *not* zero-order compact, because it skips some notes. However, since the gaps do not contain any chains-up (the relevant base orientation), the Z-chain *is* 1-compact. The Z-chain in Staff 2 is not 0-compact, because it skips some notes. It also skips a chain-up (the 8th and 9th notes of the tune, boxed in Staff 1 as a chain up) – therefore it is *not* 1-compact. It doesn't skip any Z-chains up-down, so it's 2-compact.

Lower-order compactness is stricter than higher-order compactness, allowing less skipping and gapping, and therefore describing simpler and more local structure. If a Z-chain is 2-compact it's therefore also 3-compact, but saying it's 2-compact is more descriptive. Figure 5–9 shows a 3-compact 4th-order Z-chain with relevantly oriented 2nd-order Z-chains in the gap. Figure 5–10 shows a 1-compact 3rd-order Z-chain up-be-up – while a few notes are skipped (because they are on the inside of chains *down*), no chains *up* are skipped.

Since this definition of compactness only counts skipped elements in its *own* sub-orientation, it's possible for a 1-compact Z-chain to nonetheless contain a large gap (e.g. as in Figure 5–11). It's also possible to formulate an "all-dimensional" definition of compactness, where no Z-chains of the relevant order in *any* orientation can be skipped.

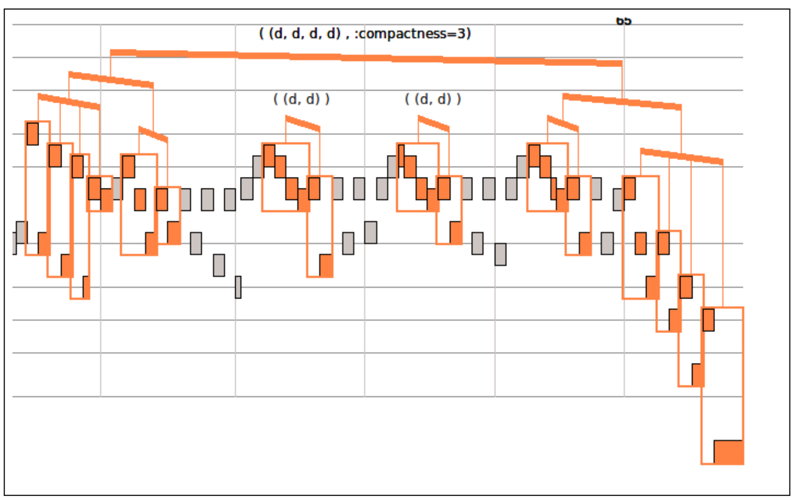

Figure 5–9: A 3-compact 4th-order Z-chain (down-down-down-down) with relevantly oriented 2nd-order Z-chains (down-down) in the gap.

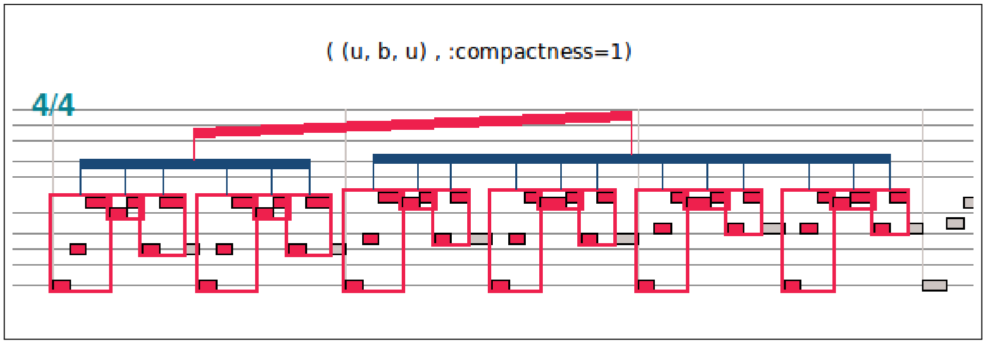

Figure 5–10: A 1-compact 3rd-order Z-chain (up-be-up) – no chains up are skipped.

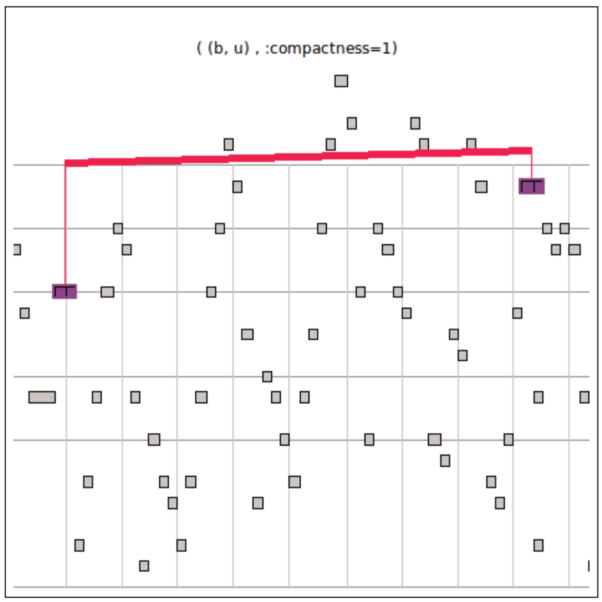

Figure 5–11: A 1-compact Z-chain with a gap – there are no *be*-chains (i.e. repeated notes) in the gap.

The basic Z-chain algorithm as described above finds all and only Z-chains that are (n-1)-compact – which is as loose as it can get without skipping Z-chains in the *relevant* (n-1) orientation, (which would lead to a combinatorial number of structures with a more complex set of interactions).

Forcing Compactness

A Z-chain of order n that's *not* (n-2)-compact can be easily *broken* into a set of (n-2)-compact Z-chains of orders n and $(n-1)$. This can be done by walking the superchain of (n-1)-order Z-chains and checking whether any (n-2)-order Z-chains fall into the gap between any consecutive pair – if so, snip the superchain at that point – for example, it's easy to see how to break the superchain in Figure 5–9, giving a 2-compact 3rd-order Z-chain on each side of the gap. Using this principle, Z-chains can be broken to guarantee any order of compactness. In the figure, the 3rd-order Z-chain on the left is already 1-compact, while the one on the left would have to be broken again to achieve 1-compactness. It's also possible to specify a minimum compactness requirement at the generation stage, running the Z-chain algorithm with a constraint allowing only compact chaining. This can be done by keeping track of indices for each recursive generation, and making sure that indices at the prescribed level are compact (i.e. no indices are skipped). Pseudocode is shown in Figure 5–12.

Figure 5–7 (discussed above), shows a cover of 1-compact Z-chains giving a more "linear" sketch of motion throughout a score; Section 4 below is about using 1-compact Z-chains again to discover local schematic shapes.

5.4.2 Pivots and intersection

Running the Z-chain algorithm for a single Z-relation (e.g. top pitch, bottom pitch, chain length, *or* chain span) gives a *set* of Z-chains in different orientations.

N-COMPACT Z-CHAIN ALGORITHM

Input:

- S: sequence of terms $(t_1, t_2 ... t_n)$ of any type (as `term`).
- F: function of type `(term -> integer)`
- Z(F): function of type
  `((term -> integer) -> sequence -> integer)`
- N : integer compactness constraint

Method:

1. Run `chaining_algorithm` (S,F) – returns a sequence of chains C^1.
2. Partition C^i by orientation (preserving sequence order for each subsequence), obtain $\{C^i_<, C^i_>, C^i_=\}$.
3. For each C^i_{ori}
   (a) Index the chains in order, so $C^i_{ori} = (c^i_{ori_1}, c^i_{ori_2} ...)$
   (b) `if` N $\geq$ (i - 1), partition C^i_{ori} such that each part contains a *compact* sequence of c^N_o chains. *Compact* means that the sequence of c^N_o chains in the partition is indexed $(c_{o(k)}^N, c_{o(k+1)}^N, c_{o(k+2)}^N ...)$ for some k.
   Since all incoming Z-chains are guaranteed to be N-compact, this partition is always possible.
   Obtain partitions $\{C^i_{oria}, C^i_{orib} ...\}$
4. For each C^i_{orij},
   (a) `if` $|C^i_{orij}| \leq 1$, save contents of $C_{ori}{}^i_j$ for output as a top-level N-compact Z-chain; branch terminates.
   (b) `else if` $|C^i_{orij}| > 1$,
      i. run `chaining_algorithm` $(C^i_{orij}, Z(F))$, returns a sequence Z^{i+1} of (i+1)-order-Z-chains.
      ii. Recurse to step 2 with $Z^{(i+1)}$ as C^i.

Output: Set of top-level N-compact Z-chains. Since the output Z-chains are *indexed* at each level, it's easy to test them for M-compactness.

Figure 5–12: Pseudocode for a version of the Z-chain algorithm that guarantees N-compactness, and indexes Z-chains at all levels such that it's easy to test them for M-compactness (for any M).

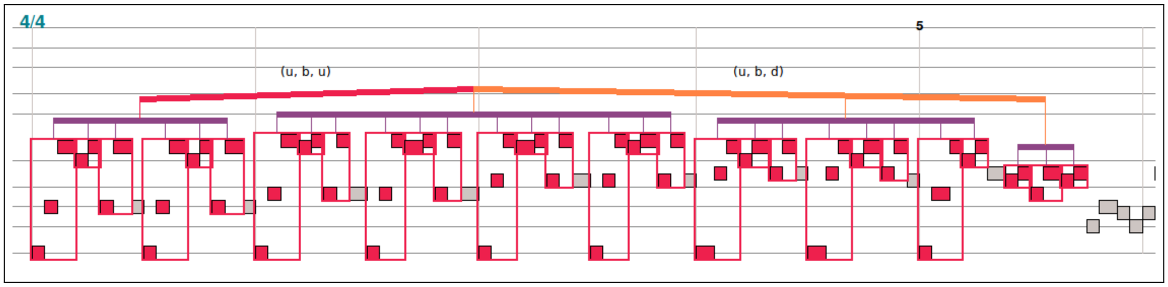

Figure 5–13: Two Z-chains sharing (n-1) order orientation (here *up-be*) may pivot with one another on that (n-1) order Z-chain. The red-beamed supershape is a Z-chain up-be-up, and the orange supershape is up-be-down. The second purple beam is the shared Z-chain up-be.

In this section we discuss some of the ways Z-chains relate to each other temporally, and how they may share structure.[17]

1) Z-chains of higher order *contain* Z-chains of lower order in their sub-orientations – e.g. Z-chains up-dn-up contain Z-chains up-dn; therefore they also contain chains up. This is evident from their construction, since higher-order Z-chains are built out of lower-order Z-chains.

2) Z-chains of the same (n-1) orientation *pivot* on Z-chains of that (n-1) orientation – so a Z-chain up-dn-up pivoting with a Z-chain up-dn-dn has a Z-chain up-dn as the second-order pivot. This statement characterizes any two n-order Z-chains sharing (n-1) orientation that *fold* (i.e. overlap) with one another.

[17] Since Z-chains have gaps, a simple application of *polyphonic* analysis (as discussed in Chapter 2) is insufficient to describe the temporal relations of these structures (though projections of each Z-chains to a temporal interval or set of temporal intervals affords polyphone treatment).

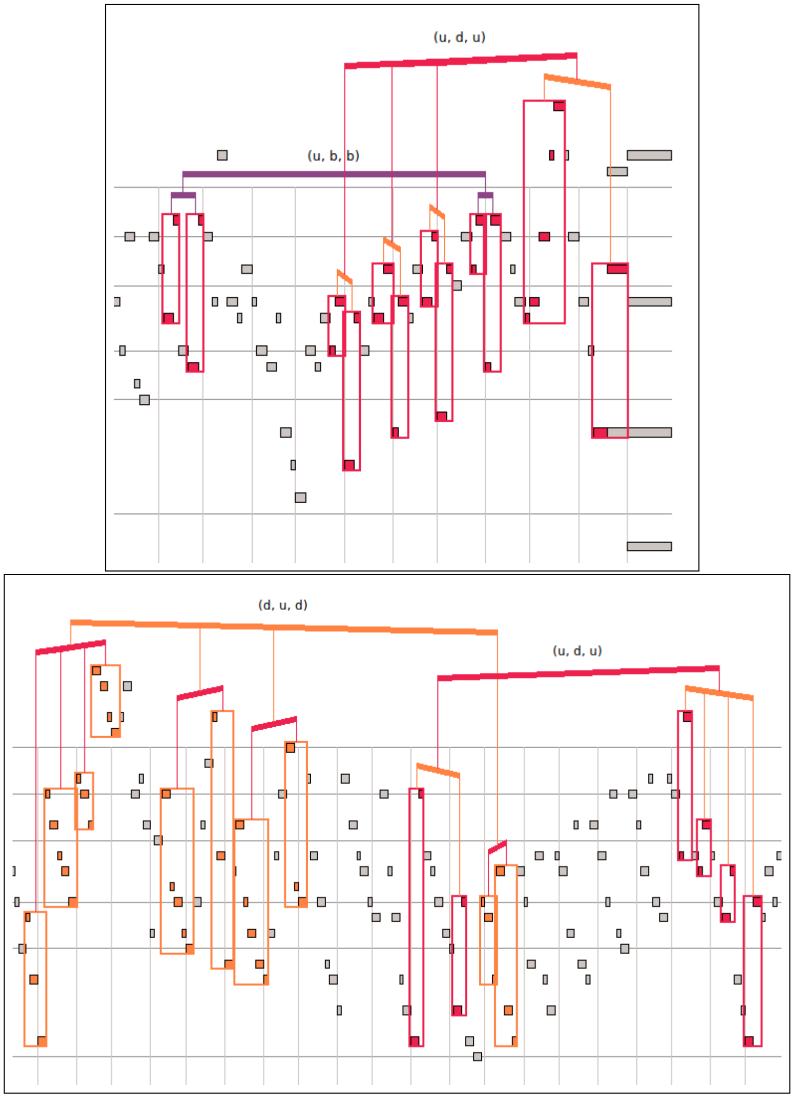

Figure 5–14: Interleaving Z-chains that *fold* but don't *pivot*.

All first-order pivots (endpoints of first-order chains) appear in the intersection of two different oriented chains. Likewise second-order pivots (end-chains of second-order Z-chains) appear in two different second-order chains, and so on. This is evident from the way Z-chains are constructed. An example is shown in Figure 5–13

3) It's possible for two Z-chains to *fold but not pivot* – they can overlap temporally, but not share *any* elements. This is because a Z-chain may contain gaps, and parts of Z-chains may fall into each others' gaps such that no elements are shared. Illustrations in Figure 5–14. It's also possible for a Z-chain to be temporally included in another, but not share any structure, the smaller falling in a gap of the longer Z-chain; e.g. as in Figure 5–9.

4) Different parts of large, gapped Z-chains can simultaneously pivot and interleave in different ways, giving a complex picture which is not easily summarized (e.g. as in Figure 5–15). Reasoning about Z-chains using polyphones (temporal identity, overlap and inclusion relations), is therefore not completely straightforward.

We can take advantage of the fact that gapped Z-chains are *made of* compact Z-chains, treating each Z-chain therefore as a sequence of temporal intervals corresponding to a sequence of compact lower-order Z-chains. We can obtain a *polyphony* (i.e. a *sequence* of polyphones) between two or more Z-chains much as we would obtain a polyphony between two or more voices each singing a sequence of notes.

This analysis can be simplified if we know in advance that the Z-chains being compared share a base orientation to some higher recursive level. In this case, the dimension of the base orientation can act as a temporal ground on which the polyphony occurs, i.e. so that we only pay attention to gaps (skipped Z-chains) *in this dimension*. So if the shared recursive orientation is of order m, then we can consider m-compact Z-chains to be compact for our purposes, rather than 1-compact Z-chains. This is the case in Figure 5–15. In the figure, the first 3rd-order Z-chains of each of the 4th-order Z-chains are 2-compact, and so in the shared 2nd-order orientation dimension (down-up), they constitute a compact interval. Therefore if these two 3rd-order Z-chains *fold given the underlying ground-dimension of down-up*, then they must fold on a pivot (not falling into each other's gaps), since they are compact in this dimension. This also follows from point 2 above.

5) Suppose an alternation only of chains up and chains down, and using top or bottom pitch as the Z-relation. Then, since the chains up and down share maximal and minimal pivots, their higher-order Z-chains have the same higher-order orientations (though they may have other structural differences, e.g. compactness). This property holds at all recursive levels (i.e. if there is no be-superchain at a given level). It is *not* guaranteed wherever be-chains disturb the alternation, or when the Z-relation doesn't obtain equally to both sides of the pivot, as with chain length or chain span. Some data naturally has the property of only alternating chains up and down; otherwise it's possible to reduce be-chains to a single point to give a simplified, two-orientation Z-chain picture. The phenomenon of shared supershape on opposite subshapes was evident in our analysis of "Happy Birthday;" in Figure 5–16 it is shown on a larger scale.

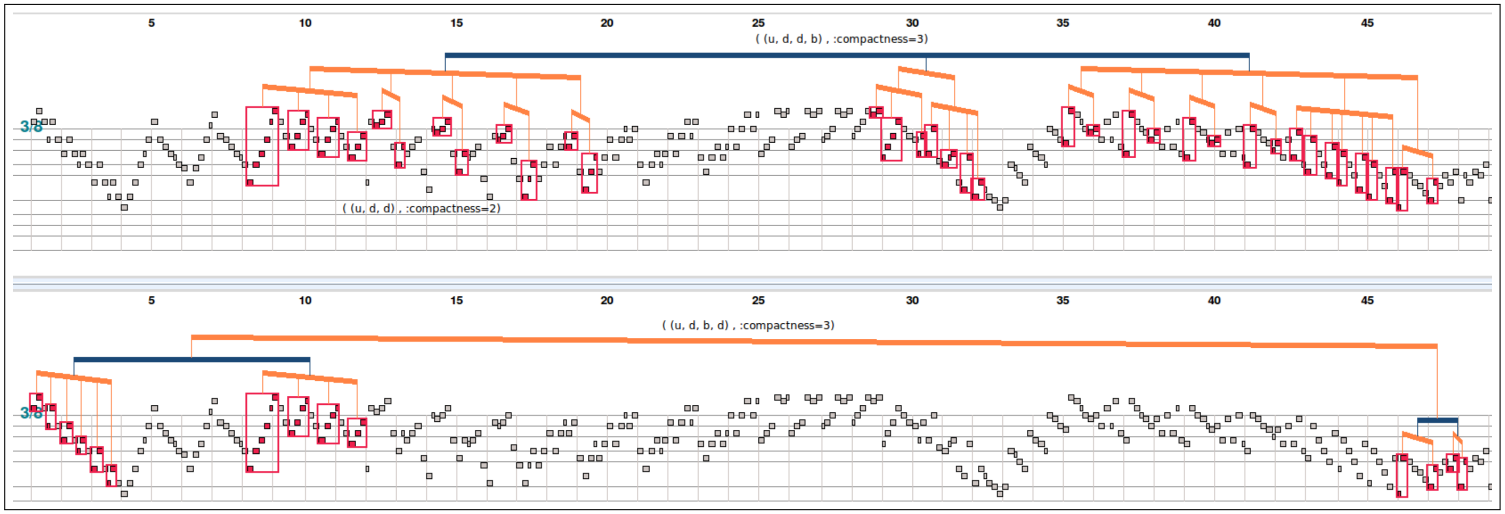

Figure 5–15: Two 4th-order Z-chains share a 2nd-order base orientation – therefore if two of their 3rd-order sub-Z-chains *fold*, then they *pivot* on their overlap. Top panel is up-down-down-be and is 3-compact; bottom panel is up-down-be-down and is 3-compact. The shared 2nd-order base orientation is up-down. The first *3rd* order Z-chains of each *fold* with one another, and *pivot* on a 2nd-order chain.

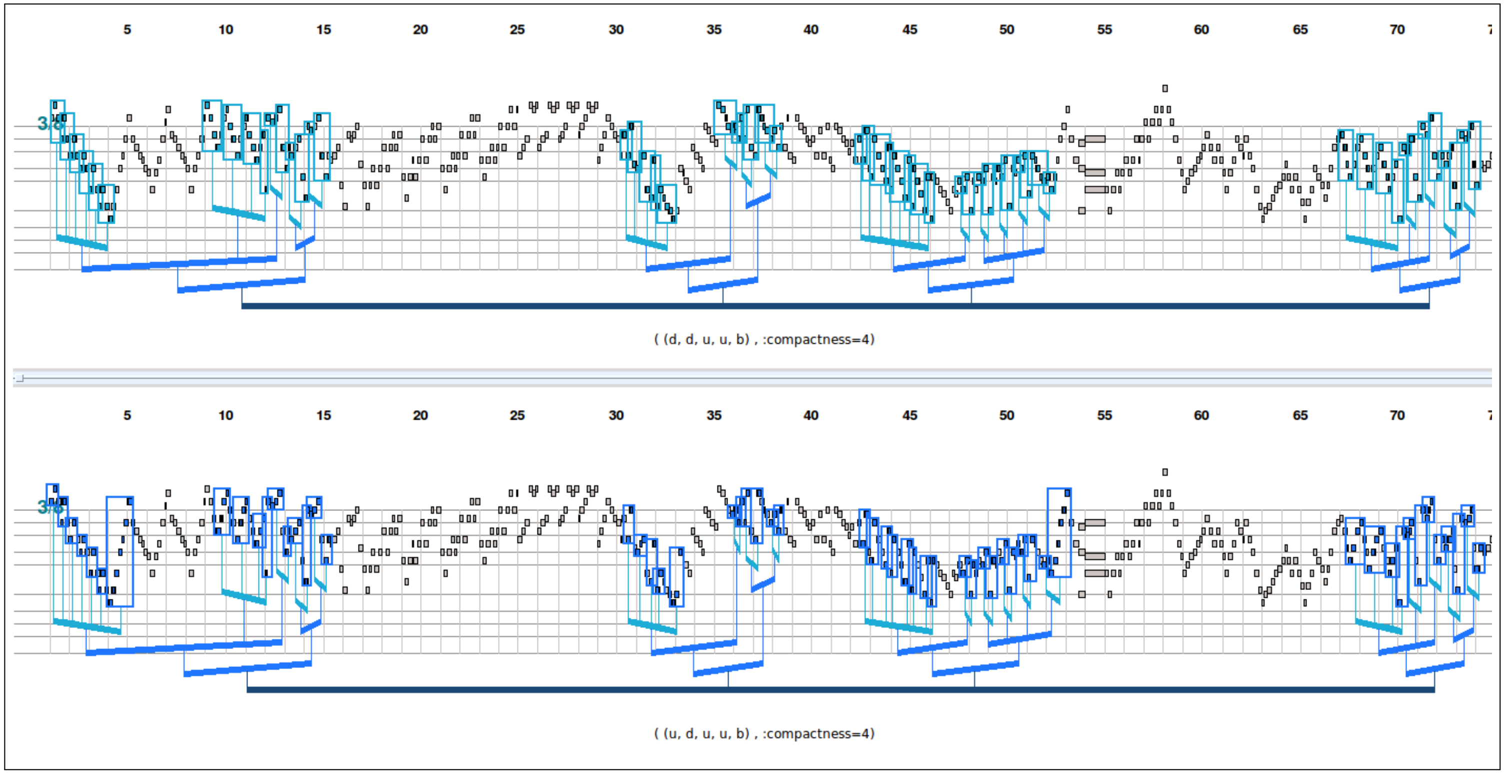

Figure 5–16: Z-chains with differently oriented base-chains that have the same superchains.

5.5 Z-shapes

5.5.1 Definition and Algorithm

In this section, intersections of 1-compact Z-chains are used to locate *Z-shapes* as compact sequences of *base* chains in *one* orientation, characterized by *supershapes* (of any recursive depth) on *both* top and bottom: a schematic two-sided "contrapuntal" shape. Z-shapes are defined as conjunctions (intersections, synchronizations) of Z-chains in different Z-relations on the same base-chains.[18] In contrast to the structural *building* paradigm of Z-chains, with increasingly large structures, finding intersections of such structures tends to generate smaller structures.

Taking the base-chain intersection of two Z-chains results often in a shape that is *not* characterized by the supershapes of the Z-chains, but is *contextually defined* by these. For example if two Z-chains intersect on a single chain, their intersection is not expressive of their recursive Z-orientations; likewise if two third-order Z-chains overlap on just two base-chains, it's impossible for the third-order superchains to be fully expressed. In these cases the superchains refer to the *context* of the intersection, but don't describe its shape. Although contextually defined schemas may have some utility, we will avoid complications and abstractions by ensuring that the intersection *is* characterized by the superchains. To do this, we accept an intersection only if it contains at least two subchains from each level of each Z-chain.

[18] Intersections of Z-chains in any combination and number of Z-relations are possible; here we use *top* and *bottom*.

To obtain Z-shapes, we first compute the sets of 1-compact Z-chains on top and bottom pitches, and then find the intersections of their cartesian product according to a rule which ensures that the recursive orientation ("Z-orientation") of both Z-chains is fully expressed.[19] Pseudocode is given in Figures 5–17 and 5–18 Since the Z-chains are 1-compact, the resulting intersection subsequence of base-chains is consecutive.

5.5.2 "Motivic" analysis

Z-shapes are smaller and more specific than Z-chains, providing a concentrated description of local shape. These can be used for something like a "motivic" analysis.

A *motive* (i.e. "motif") is a short musical fragment that is used several times within a score, possibly varied or transformed. There is no definitive notion of what counts as a motive, or of what counts as a variation, leaving an open field for creativity in both composition and analysis.

One way to approach computational motivic analysis is as a search for *repetition* of sequences (with or without gaps), allowing for specific transformations and/or for "error" (i.e. limited arbitrary difference – "similarity").[20] This kind of approach begins with a score as a plot of unanalyzable points, and tries to find repetitions of arbitrary (and often unanalyzed) shapes. Z-shapes afford another point of view.

---

[19] Since many of the compact Z-chains don't intersect at all, the size of the cartesian product is effectively reduced.

[20] E.g. [Conklin2010, Buteau+2008, Lartillot2004, Cambouropoulos+2000, Quinn1997]

one_sided_schematic_intersection
Input:

- base_chain_set
- z_chain

Method:

```
if z_chain is a base chain (a 1-st order Z)
then
  (if z_chain ∈ base_chain_set
  then return z_chain
  else return Null)
else
1. for child_zchain in z_chain do
  one_sided_schematic_intersection (base_chain_set,
                                     child_zchain)
2. count remaining (non-Null) child_zchains:
  if > 2 then return a version of z_chain with same
  z-orientation and remaining child_zchains
  else return Null
```

Output: A Z-chain that:

- expresses the same z-orientation as the input `z_chain` (i.e. has the same depth)
- is a subtree of (or equal to) the input `z_chain`
- contains only base-chains from the input `base_chain_subset`
- is the maximal such Z-chain, or if no such Z-chain exists, `Null`

Figure 5–17: Pseudocode. This function takes a set of base chains and a Z-chain, and returns the maximal subsequence of the Z-chain intersecting with the base-chain set, such that the same Z-orientation is expressed, or else Null if there is no such subsequence. The recursion expresses a tree-iteration bottom-up on the Z-chain, so it terminates naturally when it reaches the top of the tree.

two_sided_schematic_intersection
Input:
`top_z`: a 1-compact Z-chain on top pitch
`bot_z`: a 1-compact Z-chain on bottom pitch

Method:

```
let top_inter = one_sided_schematic_intersection
                       (base_chains(bot_z),top_z);
let bot_inter = one_sided_schematic_intersection
                       (base_chains(top_inter),bot_z);
if top_inter = Null or bot_inter = Null then
return Null
else
  if base_chains(top_inter) = base_chains(bot_inter)
  then return (top_inter,bot_inter)
  else
   two_sided_schematic_intersection(top_inter,bot_inter)
```

Output:
Well-formed top and bottom Z-chains such that they contain the same set of base-chains, or `Null`.

Figure 5–18: Pseudocode. This function takes two Z-chains and finds their mutual intersection, such that both express their Z-orientation (i.e a “Z-shape”). The recursion is linear in the number of base chains (and converges quickly).

If we start by obtaining a set of Z-shapes, then we can look at a score not as a plot of *points*, but as a plot of shapes which already have a structural identity, and *each* of which can be further analyzed. Z-shapes are a way of identifying small-scale *universal* (pervasive) shapes, instead of moving directly from the universal point (a "note") to the *particular* shapes or subsequences that characterize a particular score.

The smallest and simplest Z-shapes are defined by second-order Z-chains on top and bottom. These small Z-shapes are pervasive (as are base chains and second-order Z-chains), covering *any* sequence that has more than one chain in each orientation. A sequence consisting of *one* chain doesn't have any second order chains, and therefore no Z-shapes; a sequence consisting of *one* first-order pivot (e.g. a chain up and then a chain down) likewise doesn't have any second order shape. But any sequence with (at least) second-order Z-chains *necessarily* is covered by Z-shapes, since second-order Z-chains always have a productive intersection, with top and bottom Z-chains forming a sequence of *quadrilaterals* with either parallel, convergent, or divergent supershapes.

Z-shapes of higher order Z-chains have more complex shapes than quadrilaterals, with recursive orientations on top and/or bottom. These orientations need not be of the same order, and need not be synchronized in their subshapes.

Before showing some pictures of Z-shapes as motivic cells, methods are described to further relate and differentiate these cells, as each schematic Z-shape definition admits of an open number of varied realizations.

5.5.3 Relations on Z-shapes

A set of Z-shapes with the same schematic definition are "the same" with respect to this definition (and this kind of sameness is, furthermore, easy to *hear*), but they may otherwise differ. Unlike a simple point, each of these small shapes bears further analysis, and they can be identified or differentiated one from another from any number of points of view.

One way of analyzing shapes is to use a normalization, measurement, or other summary function to compare different instances of a schematic shape. These kinds of functions can be freely invented. The function can then be used as an equivalence relation, to partition the shapes into subsets – i.e. for a function f, each subset S is such that $s_1, s_2 \in S \leftrightarrow f(s_1) = f(s_2)$. Since shapes may be the same in some ways and different in others, different functions offer different partitions – different points of view.

For instance, we could identify subsets of shapes that are made up of the same sequence (or set, or normalized sequence...) of durations. Or we could count the *number of chains* in each shape: a shape with schema (top: down-down, bot: down-down) can be fulfilled by a minimum of two chains, but it can also be arbitrarily long.

In many of the illustrations that follow, shapes have been related by their sequence of diatonically normalized intervals – if two shapes have the same kinds of intervals from note to note as seen from the perspective of a major or natural minor scale, then they are labeled with the same number (and color) – the number itself bears no information.[21] If a shape is not matched with any other – if it is in a subset on its own – it is labeled with -1, so that shapes that *are* repeated are easier to see.

5.5.4 Illustrations

Figure 5–19 shows three sets of Z-shapes on the Presto for solo violin by J.S. Bach. These drawings on the score give an idea of how Z-shapes can show repetition, variation, and zones of occurrence and non-occurrence.

Figure 5–20 shows a close up of three Z-shapes at the end of the Presto, giving more detail on how different Z-shapes interact. The same passage is described from different oriented points of view. The top panel shows that the chain of similar shapes in orange is related to what came immediately before.[22] The second panel shows how the chain of similar shapes in green (covering the same territory as orange above) is related to another chain of purple shapes that come soon after it – but these purple shapes don't have a counterpart in the first panel. The two chains of green and purple shapes both are terminated by a *different* version of the shape, showing a technique of more gradual chain-breaking rather than sudden contrast. The bottom panel shows that the chains of smaller shapes in the second panel are included in larger oriented shapes. These larger shapes show a longer-scale, higher-order orientation, but are more specific and not universal (ubiquitous, as the small shapes are).

---

[21] More precisely, this normalization takes minor and major seconds to be identical, minor and major thirds, minor and major sixths, minor and major sevenths, and fourths, fifths, and tritones.

[22] The term *chain* here has the same underlying meaning as in *pitch chain* and *Z-chain* – a consecutive sequence linked by an aspect of sameness.

Figures 5–21 and 5–22, show how small schematic shapes that often overlap or concatenate can form larger group repetitions. A motivic analysis could identify *groups* (e.g. temporal sets or n-grams) of Z-shapes as motivic. Figure 5–21 shows three shapes on Fugue No. 11 from Book I of the Well Tempered Clavier by Bach. A fugue is a musical form with several distinct voices, in which an initial short melody (the *subject* of the fugue) recurs several times throughout the piece. Z-shapes were taken on each of the three voices independently, and then a union of these was made, so that each panel shows the relevant shape in all voices.

The first panel gives a clear view of how one fragment of the subject recurs the same way 10 times; at measure 15 recurs overlapping with itself (in "stretto")[23] , but for the remainder of the piece it is absent. The second panel shows another shape that occurs twice (in forms 0 and 1) as part of the subject. Form 1 is only present for the first five instances of the subject; some of the other forms of this shape are in a position to be variations of the original. The original 0 form in the second panel occurs a few times after the 15 instances of the subject found in the fist panel, and further variations of this part of the subject are present right up until the end of the piece. The third panel shows further versions of identity and variation within parts of the subject. Figure 5–22 shows three different *trigrams* of shapes, which include some of the shapes from the previous figure.

[23] A *canon* is a repetition that overlaps with itself. A *stretto* is a canon of several voices with a characteristically short interval between onsets of the different voices. It's common to have a stretto in a fugue.

Figure 5–23 shows a 10-gram in canon in a mass movement by Victoria (Veni Sponsa Christe). These melodies are quite different variations of one another, but the sequence of small shapes is the same. N-grams of this length in canon are very rare in this corpus, suggesting that it's specific and improbable enough not to be an entirely "accidental" structure. Figure 5–24 shows some bigrams and trigrams on the same score. Some of these n-grams are included in the 10-gram canon from the previous figure, and parts of the canon are involved in imitation with the other voices. Although all of these n-grams are different from one another in terms of the diatonic normalization, their proximity and distribution (i.e. local specificity) on the score indicate that their similarity is not accidental – these are instances of inexact imitation.

Figure 5–25 shows 2-grams on an excerpt from Beethoven's Grosse Fuge for string quartet, showing incidence, variation, and patterning of a few motivic fragments.

Figures 5–26, 5–27, and 5–28 show all Z-shapes *per phrase* on a few German folk tunes from the Essen corpus. Three panels show Z-shapes on base chains up, down, and be. Some phrases are uncovered when they consist only of a first-order pivot (a chain down followed by a chain up); phrases consisting of only *three* chains (e.g. up, down, up) also have the middle chain uncovered. Z-shapes on these figures afford description of the shapes of these phrases, as well as patterning and contrast from phrase to phrase.

Figures 5–29 and 5–30 show Z-shapes on excerpts from J.S. Bach's Suites for solo cello. These pieces contain large Z-shapes, showing passages with deeply patterned orientation shapes.

5.6 Future Work

Z-chains and Z-shapes are structures on sequences of orderable terms. While we showed an application to melodic pitch, many other musical (and non-musical) applications are possible. For example, future work could construct Z-chains of note loudness, duration, or the cardinalities, densities or other numerical properties of other kinds of structures.

On pitch chains, we primarily worked with top and bottom pitches, but other Z-relations such as chain *length* or *span* can also be used, and these can be intersected to form different kinds of Z-shapes.

The description of shape made here is one of many possible ways of looking at melodic shape – a different analysis could be made starting from a basis of first-order pivots, or second-order Z-shapes, or first-order chains that can be broken to show different kinds of parallels. The bigger picture of what we've suggested with Z-chains is a general method for making an overlapping, interleaving, *covering* set of algebraic shape-elements.

We've suggested with a few sketches how Z-chains and Z-shapes can be used to look at scores; a detailed analysis of a score was beyond our scope, as was a corpus study. Future work might ask such questions as: Can the kinds and structures of Z-chains occurring on a score be used to characterize different kinds of stylistic procedures? Can the Z-shapes on phrases of the Essen corpus suggest a set of phrase-types?

Future work could try to characterize or quantify the regularity, depth, and patterning of Z-chains and Z-shapes. Studies could be made into pivots and higher-order pivots, or orientation *opposites*, or the incidence of "one-sided" passages, where e.g. chains up by themselves cover entire segments. Contrapuntal texture could be studied by examining the polyphony of Z-chains and Z-shapes occurring in different voices.

We've suggested that the *polyphone* concept can be used on projections of Z-chains to temporal intervals (or sequences of temporal intervals). A kind of *fractal* polyphony suggests itself by the union of these possibilities (i.e. a polyphony where intervals have *gaps* on multiple hierarchical levels); the working out of this kind of geometry may be the subject of future research.

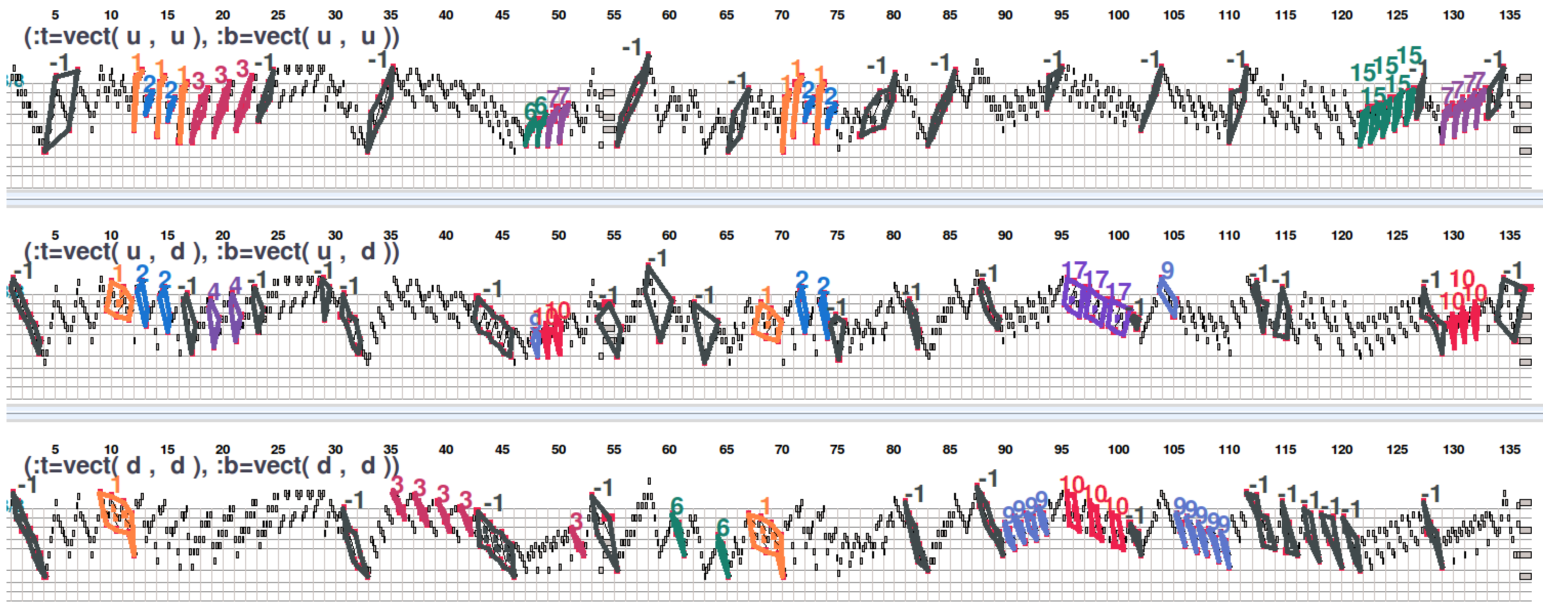

Figure 5–19: Three sets of Z-shapes on the Presto for solo violin by J.S. Bach. Top panel: shapes are made of chains up with supershape tops and bottoms both going up. Middle panel: chains up with tops and bottoms going down. Bottom panel: chains down with tops and bottoms going down. Shapes with the same color and number (per panel) have the same diatonic intervals. Gray shapes marked -1 are unique occurrences under this relation. Each of these simple shapes occurs throughout the piece with zones of occurrence and non-occurrence, with repetition and variation.

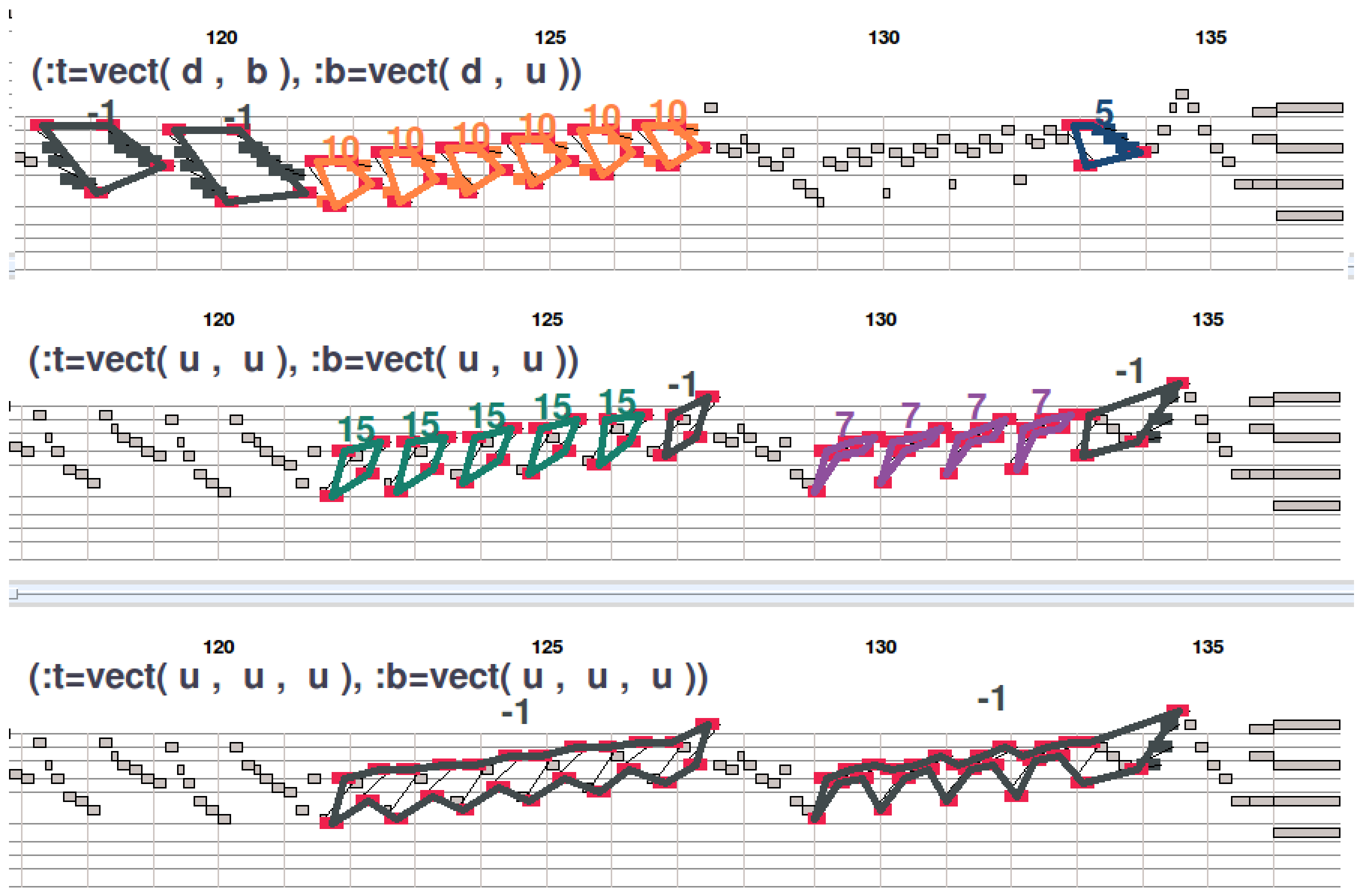

Figure 5–20: Detail from the Presto by Bach. A single passage is covered by a few different Z-shapes, showing relations from different oriented points of view. The top panel shows how the chain of orange shapes are related to what came before, in a way that the other points of view do not reveal, while the second panel shows a relation to what comes *after* the same passage. The green and purple chains are finished by variation, suggesting a technique of gradually releasing a process rather than ending it suddenly by contrast. The bottom panel shows higher-order orientation, showing how the compact Z-shape chains from the middle panel form larger Z-shapes.

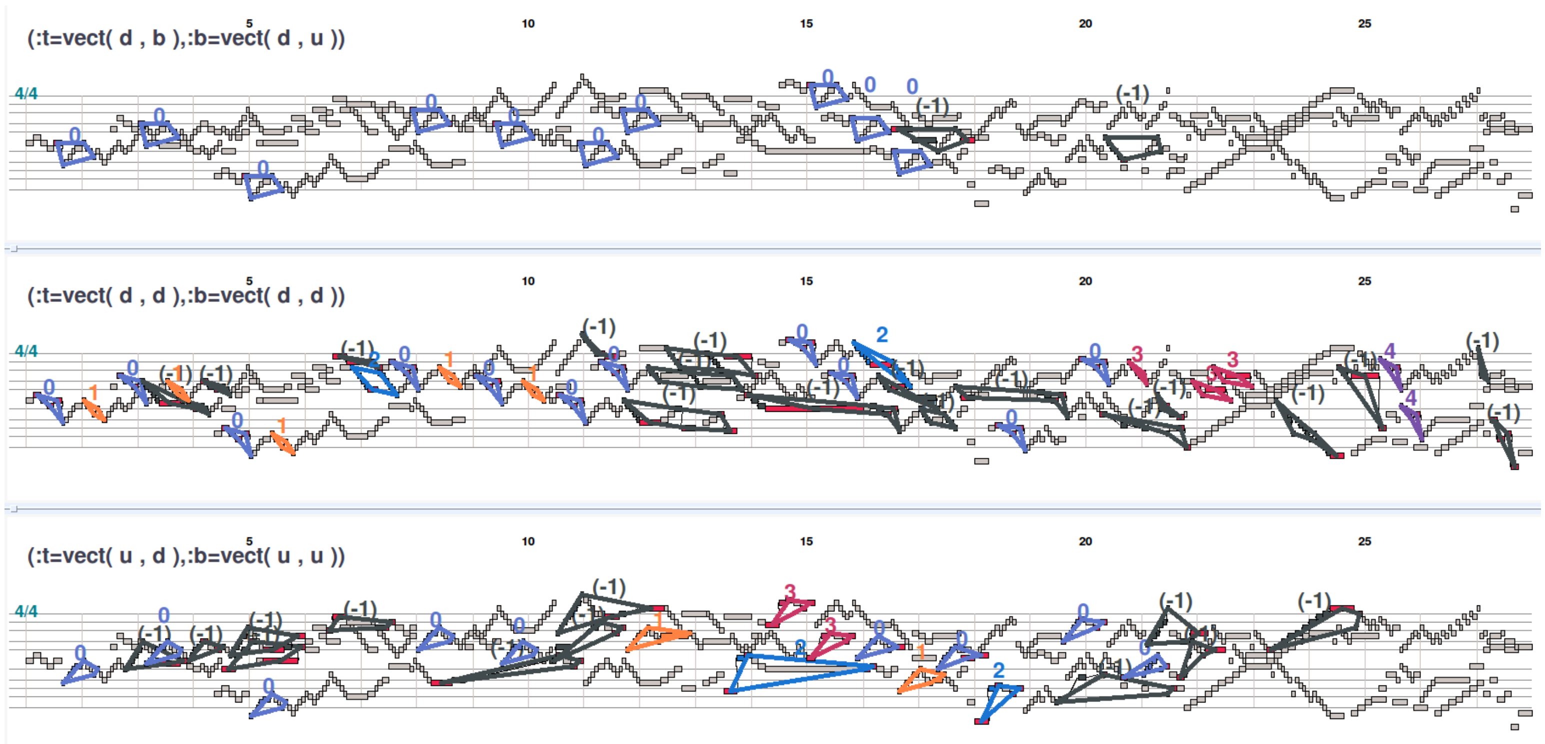

Figure 5–21: Fugue No. 11 from WTC I, Bach. Parts of the fugue's subject, recurring throughout the score, are highlighted by the Z-shapes. Shapes with the same color and number (per panel) have the same diatonic intervals (in their own orientation).

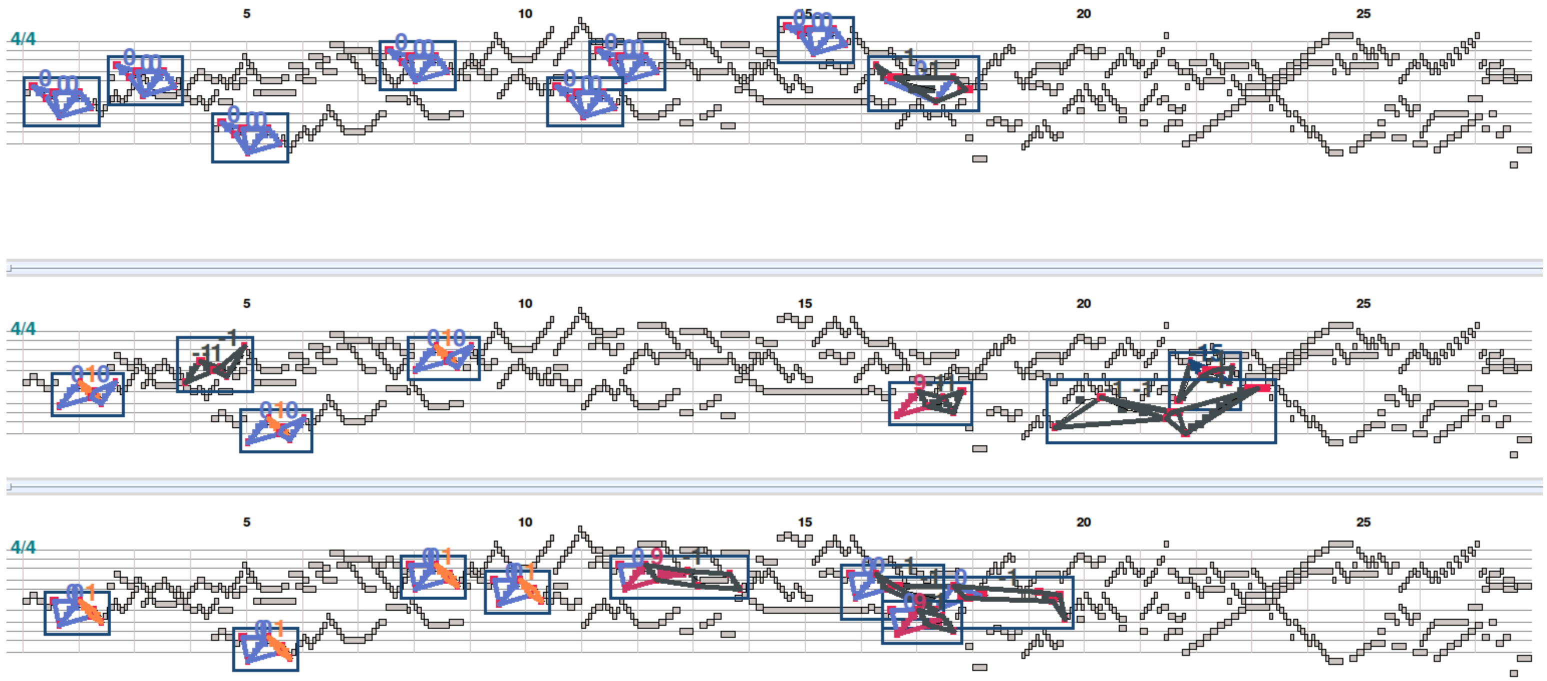

Figure 5–22: Fugue No. 11 from WTC I, Bach. Parts of the fugue subject are highlighted by *trigrams* of Z-shapes. The top panel shows sameness on part of the subject throughout the first portion of the piece; the second and third panels show how other parts of the subject are varied, omitted, repositioned, and reworked throughout the piece.

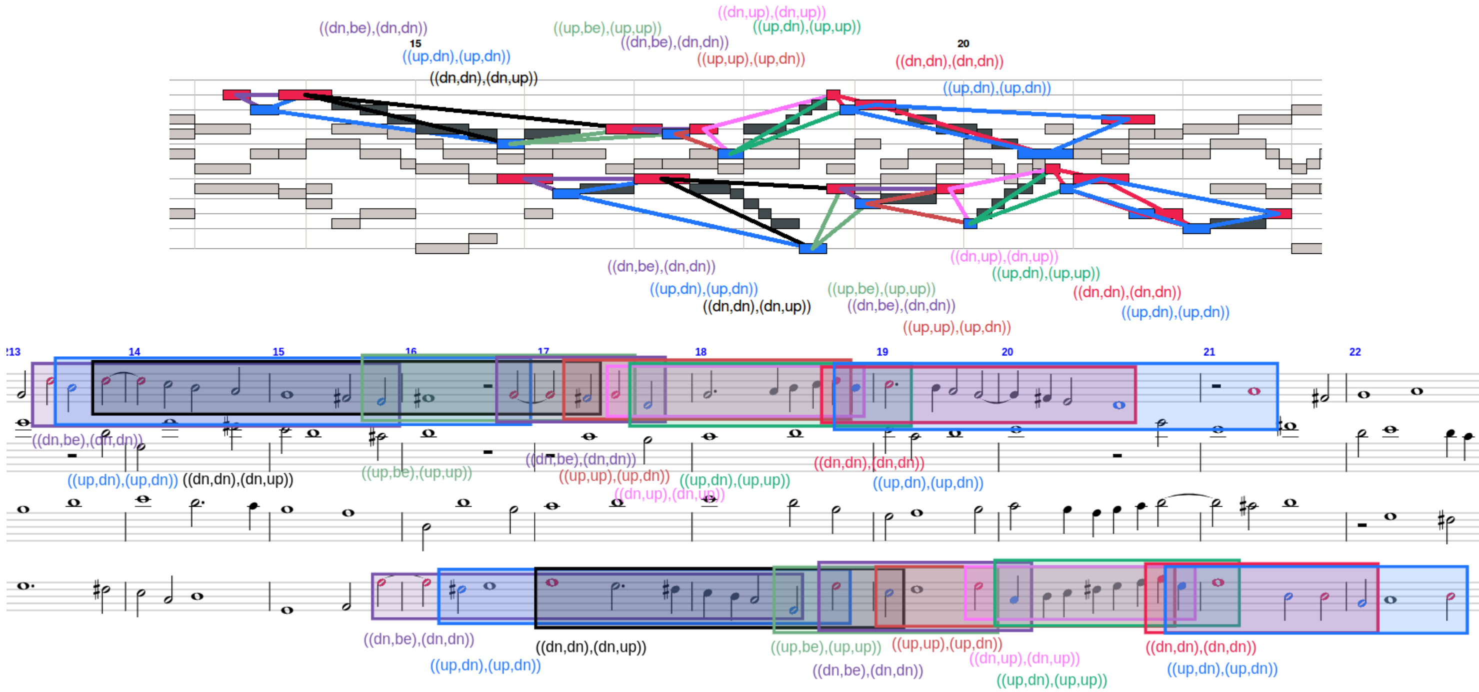

Figure 5–23: Victoria, Veni Sponsa Christe: a 10-gram in canon; piano-roll and music-notation views.

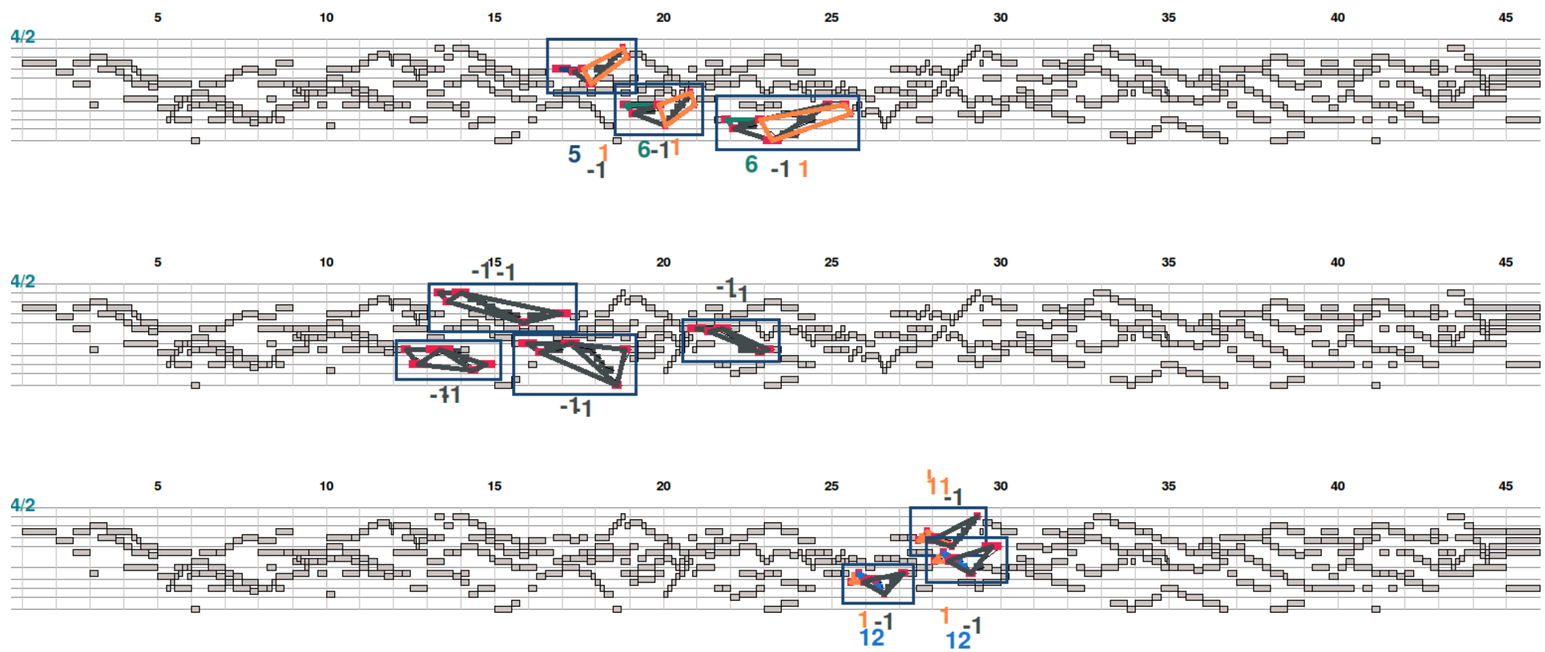

Figure 5–24: Victoria, Veni Sponsa Christe: 2- and 3-grams showing imitation.

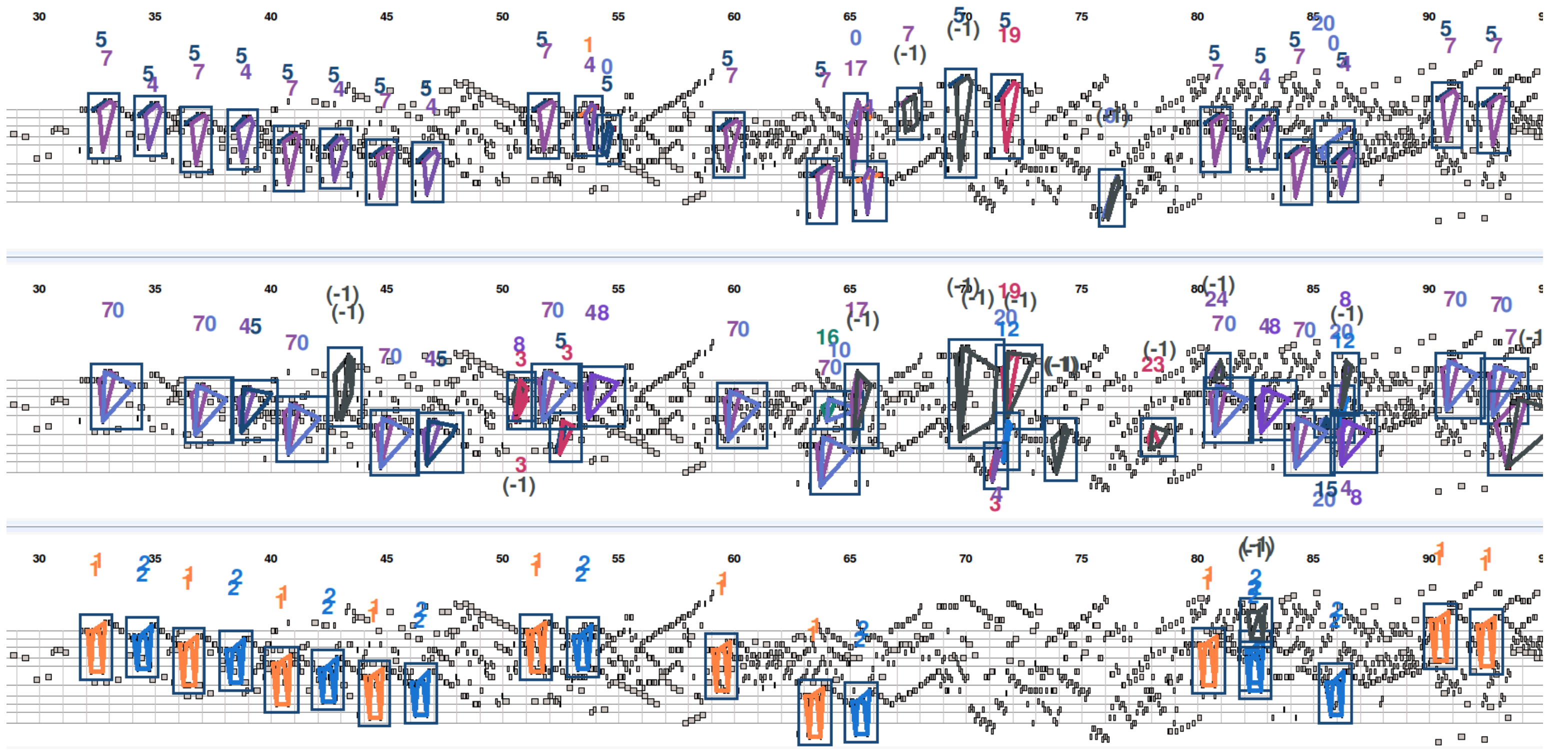

Figure 5–25: Beethoven: Grosse Fuge (Op. 133) for string quartet (excerpt). Z-shape 2-grams showing the incidence of motivic fragments and their variations.

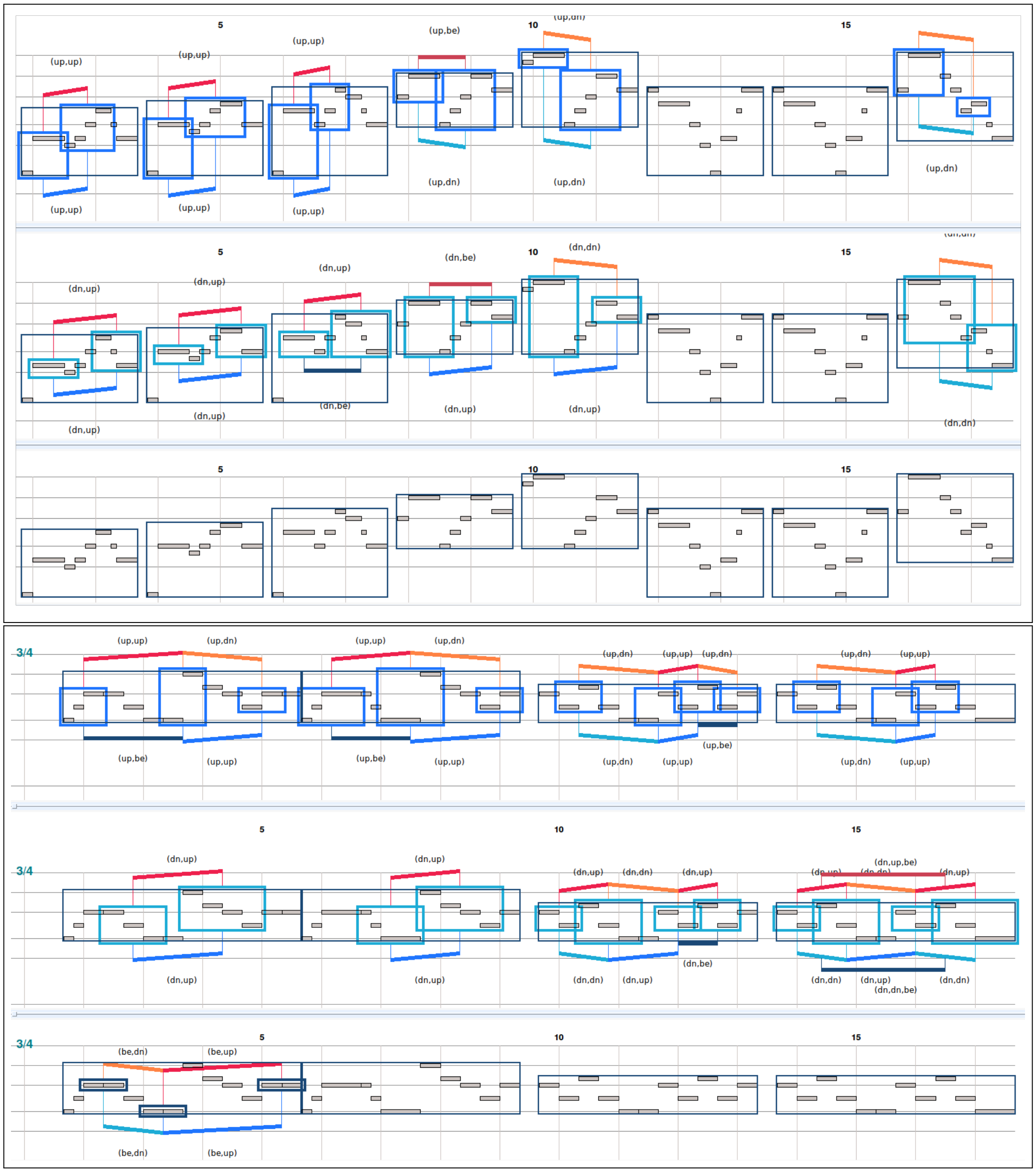

Figure 5–26: Two-sided Z-shapes on Essen tunes (1st of 3). Three panels per tune, for three base-chain orientations.

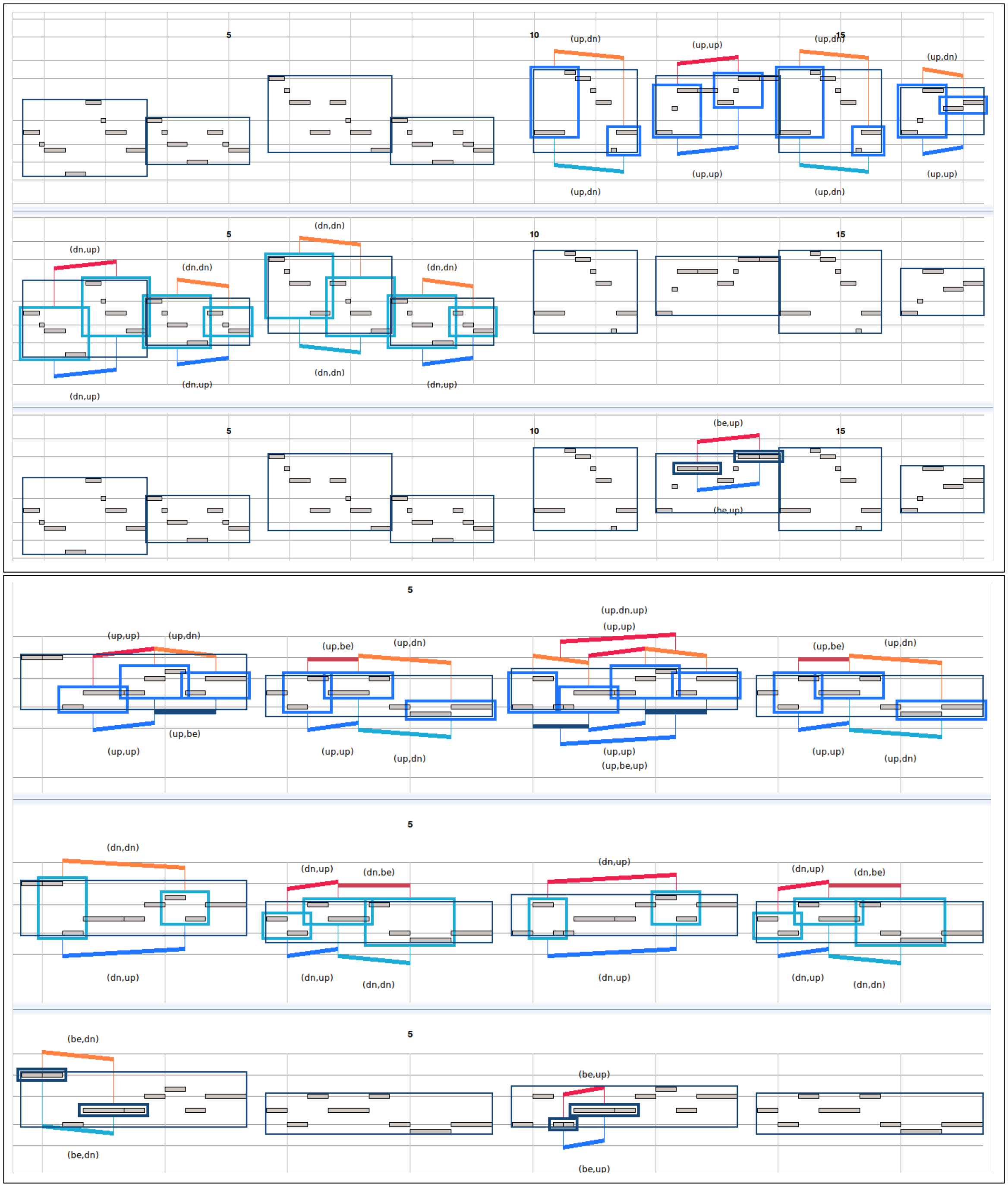

Figure 5–27: Two-sided Z-shapes on Essen tunes (2nd of 3). Three panels per tune, for three base-chain orientations.

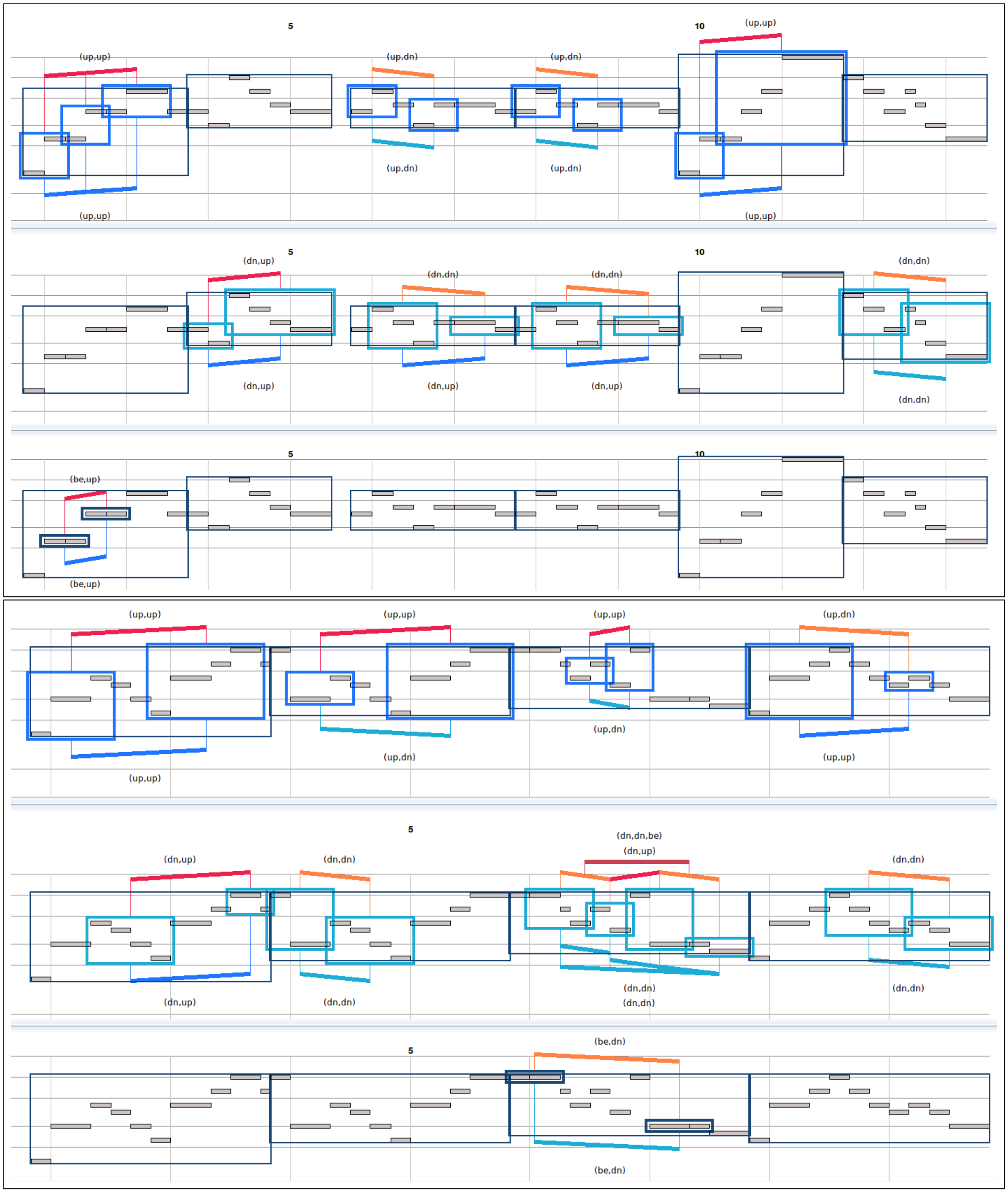

Figure 5–28: Two-sided Z-shapes on Essen tunes (3rd of 3). Three panels per tune, for three base-chain orientations.

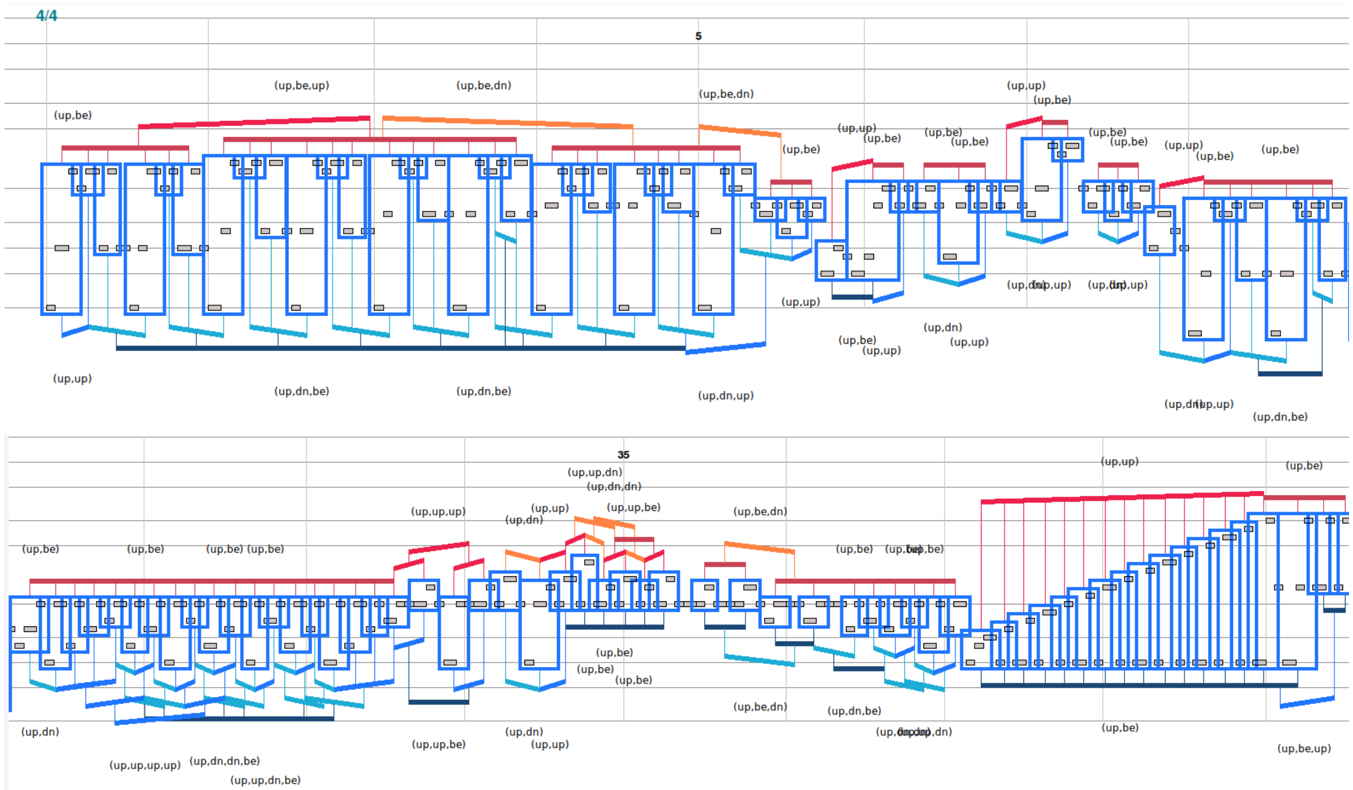

Figure 5–29: Z-shapes on Cello Suite No. 1, Prelude (excerpts), by J.S.Bach.

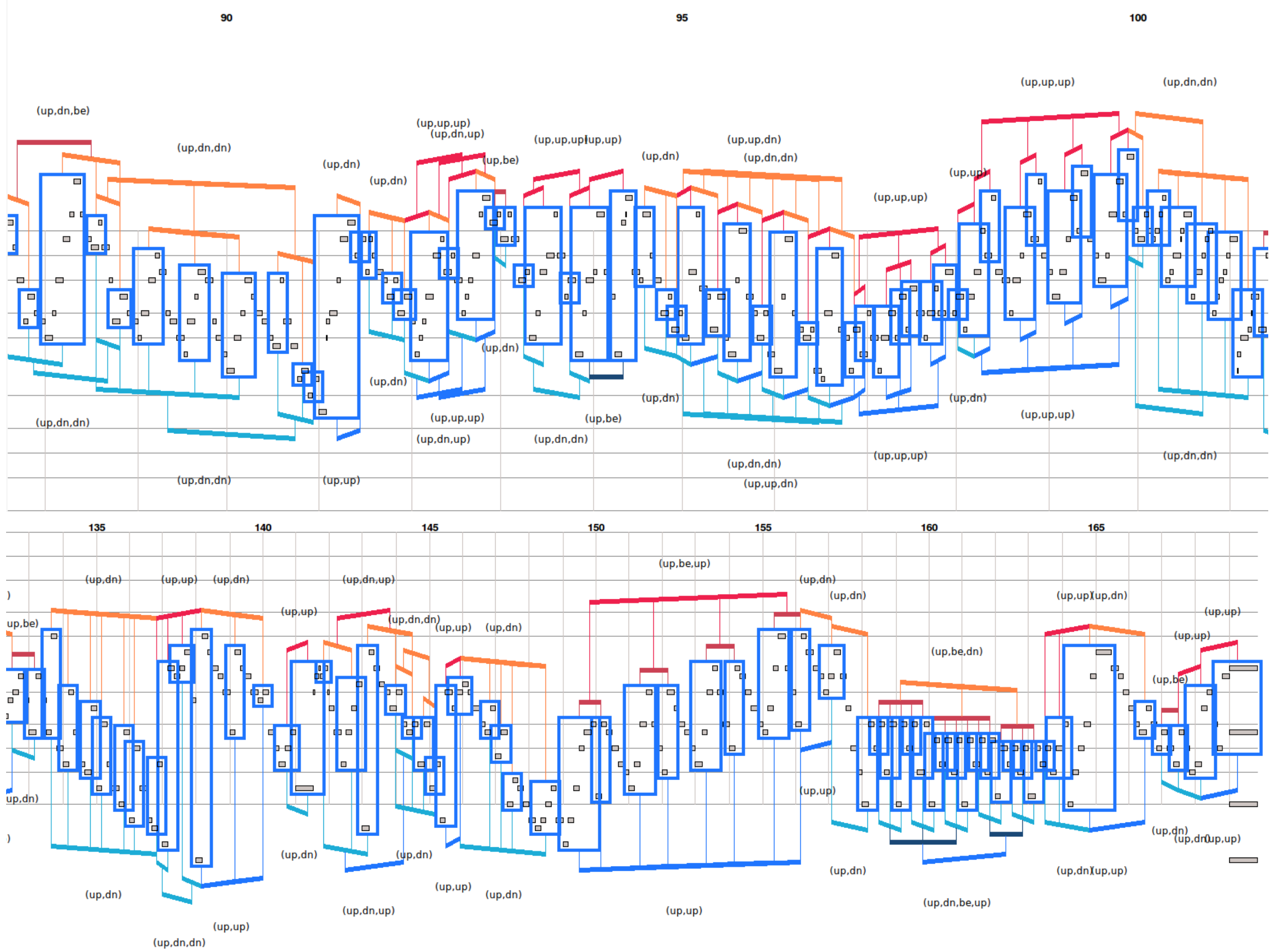

Figure 5–30: Z-shapes on Cello Suite No. 3, Courante (excerpts), by J.S.Bach

CHAPTER 6
Music and McLuhan's Evolution of Media

This concluding chapter is about a place and role for computing in the world – AI in general, with attention to music computing in particular. It's also an introduction to Marshall McLuhan's theory of media. McLuhan's media theory is an analysis of cultural psychology, premised on the observation that the material and intellectual *technical* possibilities available to individuals are a primary force driving cultural evolution. McLuhan's observations are used in this essay to inspire a picture of a world in which computation participates in an active art culture, offering new technical means through which to observe, enjoy, interact with, and create meaningful objects.

Section 1 explains McLuhan's division between two pseudo-sensory modalities, the *visual* (or *linear*), and the *auditory-tactile* (or *simultaneous*), and traces their historical incidence in media technology. Section 2 outlines some of the particular effects on music of the historical evolution of media. Sections 3 and 4 deal with the effects of media technology on the practices of art and of science, with attention to musical issues and our current (and immediate future) media age.

Section 5 positions the specific technical contributions from the previous chapters with respect to McLuhan's analysis, as part of the general transformation of media from "linearity" toward "simultaneity." This section functions as a summarizing conclusion for the main content of the thesis, re-linking the three structuring methods described in Chapters 3, 4, and 5 and relating them to the conceptual background of Chapter 2.

Finally, Section 6 explores how computational technology creates a new environment through which works of art from past ages (historical art) can continue to generate meaning and knowledge.

6.1 Media: Extensions of Our Senses

Media are environments that alter patterns of perception and sensibility.[1]

The basic premise in Marshall McLuhan's media theory is that media and technology are extensions of our bodies and our senses. Just as they extend our capabilities, they mediate our experiences and perceptions. As a result, they form an extended environmental body and sensorium, as transparent to us in its influence as our natural senses. These environments are not containers but processes *through* which we perceive, think, and exist. McLuhan seems close to Merleau-Ponty's thesis that all consciousness is perceptual consciousness (which is not to say that all consciousness is mediated by *external* sensing organs).[2] The balance of the senses mediates consciousness, and alteration of our technological environment alters the balance of the senses.

[1] *Marshall McLuhan Unbound [MMU]*, Vol. 3, p.23. Quotations from McLuhan are in this special font.

[2] [Merleau-Ponty1945]

In the broadest outline, McLuhan contrasts the "auditory-tactile" modality of pre-literate culture with the "visual" orientation of literacy, beginning with writing and accelerating through the invention of the printing press and the ensuing mechanical age. With electric media starting from the telegraph, we enter into a *post-literate*, auditory-tactile epoch.

McLuhan's sensory terminology (auditory-tactile vs. visual) can be difficult to intuit at first, since these don't always correspond to the use of our physical senses – electric media such as the telegraph-newspaper and the internet are *auditory-tactile*, forming a mosaic of unrelated fragments from various places and times. As a rule, *visual* orientation is associated with *literacy*, *linearity*, and *analysis*, while the *auditory-tactile* mode is characterized by *simultaneity* and *integration*.

When new forms of media are invented, the balance of the senses is altered, along with cultural and personal consciousness. The major transitions from auditory/simultaneous to visual/linear and back to auditory/simultaneous are only the major shifts in a constantly evolving media landscape. These major transitions are gradual, accumulating over hundreds of years. And the opposite or alternative modality is always available in some measure.

What I am saying is that new media may at first appear as mere codes of transmission for older achievement and established patterns of thought. But nobody could make the mistake of supposing that phonetic writing merely made it possible for the Greeks to set down in visual

order what they had thought and known before writing. In the same way printing made literature possible. It did not merely encode literature.[3]

McLuhan elaborates a set of tendencies of visual literacy or linearity. These include *centralization* of power (with extension of organizational bureaucracy), emergence of the *individual* (as well as the author, perspective, and *self*-expression), fragmentation and analysis (starting with the analysis of visually abstracted speech, and including the analytic methods of science and statistics).

McLuhan traces the beginnings of literate culture from the birth of Greek philosophy and democracy, through Roman organization, to medieval scholasticism. Linearity is dramatically accelerated with the invention of print, heralding the mechanical age. With the printing press, the remnant of the verbal contained in manuscript

[3] [MMU] Vol. 17, p.17.
McLuhan distinguishes the affordances and tendencies of "oral literature" (a contradiction in terms for him), manuscript culture, and print, as well as the changes brought by the typewriter. Manuscript culture sustained an *oral* procedure of disputation (scholasticism); the accessibility of the printed book introduced new notions of reading, expression and authorship – creating a *different* form ("literature"), rather than merely containing the old forms of linguistic expression ([McLuhan1964)] Ch.18: "The Printed Word: Architect of Nationalism").

culture is overturned, and new notions of interchangeability and modularity are invented, leading to the assembly line and specialism.[4] A culture of even more centralization and standardization is developed; in writing this is symbolized by the development of spelling, punctuation, and the dictionary. Around this time, the individual and the public are born, and perspective is initiated in the technique of painting – perspective in painting, seen by print-natives as natural and realistic, in fact had to be *invented*, and arose relatively lately in the long history of art. In the 20th Century, as we began the transition out of the visual age, what was still natural to the populace became old-guard to the artists.

Electronic media reverse the visual tendencies of linearity and specialization by reintroducing *simultaneity*. McLuhan's descriptions of the telegraph and of xerography are anticipations of the internet. With the telegraph news, we at once receive information from around the globe, without a unifying editorial perspective. In contrast to the mass-produced book, xerography allows private, specialized, on-demand publication. ***The electronic form of information service permits not only decentralizing of organizations but a wide diversity of products without additional expenditure.***[5]

[4] The verbal in the manuscript is partially reflected in the oral culture surrounding it and its resistance to *silent* reading, but also in the sculptural plasticity of the lettering of the illuminated manuscript itself. The manner of linguistic expression tended toward compression, allegorical and aphoristic, demanding interlocution, whereas print began to "spell out" and expand meaning. [McLuhan1964)]

[5] [MMU] Vol. 5, p.13.

With electric media, all places, times, and people become accessible everywhere, simultaneously – McLuhan coined the phrase ***"global village."*** It's easy to see that our current forms of electronic communication, commerce, and manufacturing are an acceleration of the tendencies already visible when McLuhan wrote in the 1960s and 70s. ***Such an instantaneous network of communication is the mind-body unity of each of us."***[6]

With electric simultaneity, the balance of our senses is no longer tuned to linearity in the same way. If products of literate culture like specialism and the traditional divisions of the arts and sciences, successive analysis, classifiable data, statistics, cause and effect, the linear concept of history, of producer vs consumer, figure vs ground, are "obsolesced" by the *fusion* caused by the simultaneity of electric media, it is not to say that they can no longer be *used*, only that they are no longer the whole picture: they are now *contained* in a larger world.

The electronic age finds it both natural and necessary to be aware of every kind of situation from many points of view simultaneously. ... This insistence on an inclusive image or consciousness is strikingly at variance with the exclusive and specialist focus of the literary mind."[7]

---

[6] [MMU] Vol. 6, p.6.

[7] [MMU] Vol. 1, p.21.

6.2 Music in a Visual Age

In an oral, tribal, "auditory-tactile" culture, music is oral, physical, unifying – as is everything else. The beginning of *writing* brings the mind-body split of the physical externalization of thought, a (further) binding of time.

Writing affords written music, and the separation of music from the physical and temporal. The "parts" in oral music are like *people* making music together, superimposed or parallel, simultaneously present. With music writing, the temporal "cross-section" becomes apparent and dissonance and consonance are recognized in a new way (see the early history of written polyphonic music). Music writing leads to the abstraction of music from poetry, dance, and performance. ***[P]rinted scores would seem to have made possible the maximal freedom of expression of 'pure music.' The visual, printed form permitted the release of the formal aspects of sound from the oral and verbal ground of music.***[8]

New conscious control of musical materials afforded by music literacy leads to choral and then purely instrumental works of increasing complexity (– the unwritten symphony is unthinkable). Written and printed music-theoretic treatises solidify aspects of the practice further into the domain of consciousness – we can think of these books not as encoding "rules" of practice, but as unifying *ways of thinking* about certain aspects of musical practice, while leaving other aspects unexamined

[8] [MMU] Vol. 15, p.14. While oral music cultures continued in many areas of the world, in this essay we will follow the "literate" Western thread.

and therefore open to "illiterate" sensate intuition. Where literacy creates the "mind-body" divide, it also creates the "conscious-unconscious" divide.

As words and letters become *objects* in relation to one another and subject to analysis, so do notes, scales, and chords. But it's no surprise that a high degree of literacy and technical knowledge is not sufficient to create art of any import, especially not art in a natively auditory-tactile medium. Music as an auditory-tactile affordance of the human sensorium relies on senses like pattern, shape, and motion, which have never been resolved (fixed) into a literate/conscious analysis – perhaps they can't be. Musical works of any interest are those in which pattern, shape, and motion are kept alive (in contrast to mere musical "écriture" – "competent," literate musical writing that somehow seems uninspired, or doesn't seem to "say" anything). Pattern, shape, and motion have always been in the domain of the illiterate – even pre-verbal – senses. It seems, for example, that Bach could not explain what he was *really* up to, even to his most talented children.

For the amateur and professional musician, literacy and print had the effect of reifying concepts of note, chord, various idiomatic and quasi-grammatical ideas, the opus, and the composer. This is parallel to the reification of the letter, word, grammar and punctuation, the book and the author.

In the scientific era, we observe further *linear* principles used in (computational) music analysis. These include the basic Cartesian principles used to build the structures in this thesis, as well as other linear concepts such hierarchical *parses* of segments or harmonic progressions, the idea of creating music out of "re-combinable" bits of other music, *classification* into genres or types, and analytics based on statistical properties.[9] In Section 5 of this chapter, we will explore what it might look like when these linear tendencies are reversed by the continued development of electric simultaneity – a change that doesn't involve *stopping* linear thinking, but *accelerating* it to the point that its opposite is produced.

One of the most obvious areas of change in the arts of our time has not only been the dropping of representation, but the dropping of the story line. In poetry, in the novel, in the movie, narrative continuity has yielded to thematic variation. Such variation in place of story line or melodic line has always been the norm in native societies. It is now becoming the norm in our own society and for the same reason, namely that we are becoming a non-visual society.[10]

Where literate culture *released* music from poetry, song, and dance, the popular culture of music in the electric age is deeply *involved* with these, as well as with other media such as film. Instead of musical specialism (absolute music as *just* music), we get musical relationship and the musical structuring of an otherwise poetic, physical, social, visual, and/or narrative experience.

---

[9] Classical statistics is a "linear" analysis because it supposes individuals in *populations*, as opposed to a ("mass") simultaneity of *interrelations*.

[10] [MMU] Vol. 4, p.16.

6.3 Electric Art and the Public

The electric age not only demands new ways of thinking about music, but a whole new formation of the arts in general, within which music, new and old, will operate. According to McLuhan, the shift into the electric causes several major shifts in the function of art.

McLuhan observes the shift of art from "anti-environment" to environment. Anti-environments (traditionally art, science, philosophy, etc.) are environmental *contents* which function to allow us to *encounter* our environment, which is otherwise invisible to us: ***Art and education become new forms of experience, new environments, rather than new anti-environments. ... Under electric conditions the content tends ... toward becoming environmental itself. This was the paradox that Malraux found in*** **The Museum Without Walls*****, and that Glenn Gould finds in recorded music. Music in the concert hall had been an anti-environment. The same music when recorded is "music without halls," as it were.***[11]

Malraux's concern in *The Museum Without Walls* has to do with the reproducibility and familiarity of a simultaneity of art from all eras and areas. The simultaneity has dramatically increased since Malraux wrote in the 1940s. We are confronted with the universal accessibility of all music, including the ability to add new music at will to this aggregate. Furthermore, works of music no longer stand in relation just to other works of music, but they appear alongside *everything else* on

[11] [MMU] Vol.4, p.12.

the internet, without the boundaries that defined the earlier media – so what used to be “a piece of music” is now “content,” an element of the omnipresent internet environment.

McLuhan theorizes the changing role of the audience. The print-tech “public” is an *environment* made up of individuals with varying points of view, versus the electric-tech “mass” as the *content* of the electric environment, and consisting of individuals ***“involved in depth in one another and involved in the creative process of the art or educational situation that is presented to them. Art and education were presented to* the public *as consumer packages for their instruction and education. The members of the mass audience are immediately involved in art and education as participants and co-creators rather than as consumers.”***[12]

The concept of the audience as the “content” is evident in the current phase of the internet. *Involvement* is of great relevance; applications that “autonomously” generate music work on a *production-consumption* model, whereas apps that facilitate interaction and co-creation are ascendant in the electric age. So far, some of the most publicly important aspects of music computation have been precisely those that allow people to *create* – applications for audio editing and transformation, score typesetting, and signal processing. Applications that involve interaction at the *musical* level offer another kind of metacreative setting.

[12] [MMU] Vol. 4, p.12.

McLuhan notices the changing nature of "the individual." From the beginning of print culture, we have a notion of the producing artist as an *individual* with a particular perspective to express, versus a consuming public. We can observe that we are no longer in an age of "the great individual" – a mythic figure of the Beethoven type is anachronistic. On the other hand, "individuality" has been spread out to everyone.[13]

[A] paradoxical aspect of this change is that when music becomes environmental by electric means, it becomes more and more the concern of the private individual. By the same token and complementary to the same paradox, the pre-electric music of the concert hall (the music when there was a public instead of a mass audience) was a corporate ritual

---

[13] Lewis H. Lapham writes in his (1994) introduction to *Understanding Media* [McLuhan1964)]: "Just as the advent of print placed the means of communication in the hands of a good many people previously presumed silent (prompting an excited rush of words from, among others, Rabelais, Cervantes, and Shakespeare) so also the broad dissemination of the electronic media invites correspondence from a good many more people presumed illiterate."

Russell's *A History of Western Philosophy* [Russell1945] has the notion of the "individual" in philosophical thought (and surrounding political and cultural trends) as one of its throughgoing themes (– see especially especially Ch. XII, "Philosophical Liberalism"). Russell makes the point that the *scientific* method, though it seems largely unconcerned with the subjectivity of the individual, depends on the modern individualistic (Protestant, post-Gutenberg) position that the authority of Aristotle and the Church are not as cogent the evidence of the senses. He also points out the fragmenting of individualism from the subversion of early "liberal" nationalism (and its aesthetic continuation in romanticism) with its "heroic" individual leaders, toward Marxism.

***for the group rather than the individual. This paradox extends to all electric technology. ... The age of the mass audience is thus far more individualistic than the preceding age of the* public.**[14]

Mechanized specialism permitted high virtuosity in the shaping of the art object, but such objects were denied any real role in the social life. They were classified as "art" and made peripheral to society and to individual consciousness alike. ... [A]rt as a classified activity dissolves with the advent of electric circuitry. The art object is replaced by participation in the art process.[15]

These days "everyone" can make music, just as they can have their own web presence in any expressive capacity. But the universalization of production is misleading if we think we are headed for a democratic paradise. While participation in art is now accessible to all, this is facilitated by "programmed" environments. In effect, the power of production has been lifted to a higher level of creation.

Today it is not the idle rich but the busy rich who are hastening to acquire squatters' rights all along the art frontiers. Why? Why should the top brass of industry and bureaucracy invade the penurious domain of the solitary artists? Perhaps because technology has itself begun to approach the mystery of the creative process? Because technology has plunged us collectively into the uncharted primitive terrors of individual

[14] [MMU] Vol. 4 p.13.

[15] [MMU] Vol. 20, pp.13–14.

artistic intuition? Has technology adopted as its province the entire human psyche and the earth which it inhabits? Are there sufficient signs that technological man is prepared to manipulate, as his matter, both earth and spirit? Have the ancient boundaries between art and nature been erased?[16] ***The artist leaves the Ivory Tower for the Control Tower, and abandons the shaping of art objects in order to program the environment itself as a work of art.***[17]

While "everyone" becomes involved with making (what used to be called) art (and what now is sometimes called "content"), the *specialists* are now "programmers" generating the environment – this itself is an environmental and artistic task. We might think of the internet media networks and the smartphone as being the "great works of art" of the past few years, as these are our current environment; and their creators are the major architects of our age.

Musical metacreation, or the "AI" pursuit of music, will have to be about musical environments not for consumption but for participation, exploration, and co-creation. On the other hand, we can even imagine "users" (content-creators or content-beings) specifying their works in a "meta" fashion, so that they can be contextually re-realized or so that a "piece" becomes a whole region of musical space that visitors can enter into and explore. In this case, environmental music programming gets an extra meta-level.

---

[16] [MMU] Vol. 13, p.10.

[17] [MMU] Vol. 20 p.14.

[T]he poetic process has become the subject, plot and action of works of art. No more divisions of form and content, meaning and experience.[18]

6.4 Electric Science and Education

For McLuhan, art and science have functioned as *anti-environments* that allow us to *encounter* our environment – like a fish in water, most of the time we are not aware of the effect of media on our sensibilities. But in the electric age, our environment is a sphere of simultaneous information and communication which functions as a extension of our nervous system. ***To create an anti-environment for such electric technology would seem to require a technological extension of consciousness itself.***[19]

This calls for a kind of artificial intelligence that helps us navigate and structure the information environment – but it must be the kind of AI that is highly interactive and promotes open exploration and probing of the environment. For example, a musical AI shouldn't (just) *generate* music, or answer *questions* about music, it should be a means of (creatively) exploring the history and potential of musical possibility. In this vision, we treat AI as a means not just of musical expression or creation, but as a means of scientific or educational probing, of actively *improving* our human musical abilities and understanding – extending our musical consciousness. In general, the goals of this kind of AI might be to facilitate virtual exploration, rather than

[18] [MMU] Vol. 16, p.9.

[19] [MMU] Vol. 4 p.8.

more "linear" paradigms like classification, prediction, quantitative optimization, and goal-achievement.

By programming our environment, McLuhan predicts that we can turn our environment into a *probe*, launching an "age of experimenters." Internet search is a primitive experiment, but McLuhan predicts a shift from "matching" to "making" – away from a search for the template of truth, and toward ***"faking as a legitimate feature of human consciousness."***[20] *Matching* refers to outer fact, while *making* outers inner fact – naturally *making* seems the more essential musical act. *Making* refers not only to an act of generation, but also to the notion that receptive communication is a creative, interpretive act.

With electric simultaneity, McLuhan predicts the closing of the gap between literary and scientific sectors. We move from fact to artifact, from rationality to integration and application, from analysis to probing and grasping. ***Under these conditions, prediction and evaluation are merely substitutes for observation.***[21]

6.5 Structural Analysis and Simultaneity

6.5.1 Linearity and its Acceleration

McLuhan predicts that in the computational age, the practice of data analysis proceeds from the *linear* toward the *simultaneous*. This happens by *acceleration* of linear media. An example of the *acceleration* process is the way that the Internet

[20] [MMU] Vol. 12, p.8.

[21] [MMU] Vol. 16, p.9.

accelerates the telegraph, which accelerates the printing press, which accelerates writing. Each successive stage has *more* simultaneity than the previous, but the earlier technologies are not *superseded*, nor are they simply *contained* in the newer media – they are *consumed* to produce the acceleration.

The way a *computer* works and the way *programs* are written are based on "linear" technology – but their acceleration has produced a cultural environment of increasingly deep simultaneity. Computational acceleration of data analysis produces the current state of AI – *deep learning* deepens data-dimensionality by accelerating statistical analysis. Structural analysis as we have described it also deepens the dimensionality of data by accelerating different kinds of linear analysis.

The kind of structural analysis we have described is based on linear and "literate" thought. Approaching music through the *score* (as data object) is a literate and linear-scientific point-of-view.[22] The structures we build are based on basic Cartesian geometric concepts, and formed from linear comparative relations such as $<$ and $=$. We use *visualizations* prominently in our methodology.

---

[22] One direction from the literature of the score toward more simultaneity is toward *sound* – where we notice that "notes" as "letters" don't begin to capture the actual variety of our phonemes, accents, and expressions. But we have continued to discuss *scores* as musical expressions discrete enough to get something of a medium-time-scale view (a level on which "music" occurs, as distinct and abstractable from sonic perception), and yet open enough to afford experimental elaboration. Since the linear-analytic view of the score is convenient, we continue to use it – just as we use "letters" and "writing" (typing) in the electric age: the simultaneity of information in the current age *depends* on the "letter."

The visual sense, alone of our senses, creates the forms of space and time that are uniform, continuous and connected. Euclidean space is the prerogative of visual and literate man. With the advent of electric circuitry and the instant movement of information, Euclidean space recedes and the non-Euclidean geometries emerge. Lewis Carroll, the Oxford mathematician, was perfectly aware of this change in our world when he took Alice through the looking glass in to the world where each object creates its own space and conditions. To the visual or Euclidean man, objects do not create time and space. They are merely fitted into time and space. The idea of the world as an environment that is more or less fixed, is very much the product of literacy and visual assumptions.[23]

6.5.2 Structural Dimensionality

From a music-analytic point of view, our hypothesis is that the informational concept of *simultaneity* is naturally present and effective in music (as an auditory-tactile artform), and therefore we are motivated to explore ways in which the structure of a musical score can be described *as* non-linear and non-hierarchical.

Polyphones describe geometric situations that are neither sequential nor hierarchical, but *polyphonic* – consisting of temporal inclusions and overlaps. A polyphonic situation is not *much* higher-dimensional than a tree, but by describing a relational

[23] [MMU] Vol. 4, p.15. “Cartesian” space is like “Euclidean” space with coordinates – more “literate,” and affording more acceleration.

dimensionality *just* beyond hierarchicality, it opens up a larger, more general and more flexible space of organizational possibilities.

The shape of a *tree* is generally understood and widely used, and the non-hierarchical temporal relation (overlap) is also easy to understand. Polyphones contribute formal descriptions and visualizations of the combination of these. By understanding the shape of polyphones, we increase our ability to *observe* non-hierarchical temporal organization.

Polyphones are a description *of* simultaneity as found (from a temporal-dimensional point of view) in the musical score – i.e. where and how do the (temporal) simultaneities of things occur, and in what way can a picture of the set of these simultaneities be organized.

The shapes formed by the graph of pairwise such relations among a set of temporal intervals are descriptive of the polyphonic "textures" among the *notes* of a musical score. Polyphones can also be used to describe relational structures among any other constructions or demarcations of temporal intervals, such as sets of segments describing melodic, harmonic, rhythmic, phrasal, or timbral content – a polyphone graph can be generated among sets of segments of the *same* structural type, or among *different* types.

N-sets are another non-sequential, non-hierarchical way of addressing the temporal configuration of structures on a score. Whereas polyphones are specific about the relations of temporal intervals, N-sets take a more aggregate approach, finding larger timespans on a score that contain sets of (kinds of) structures, that can be thought of as occurring more or less "together" within the larger timespan. We can see this as a looser (but still formal) definition of "simultaneity," giving a different kind of structural description.

The boundaries and content-description of N-sets are defined by the *set* of structures inside (rather than e.g. a statistical measure) in order to maintain structural specificity – though statistical methods can be used to take *measurements* of the content of each N-set.

N-sets take a sequence of sets and produce a polyphonic *set* of sets (superset-unions) – in doing so they take a low-dimensional projection of the score and produce a geometrically higher-dimensional superstructure. The set of N-sets obtained has a *polyphonic* organization, because N-sets have both inclusion and overlap relations.[24] Like polyphones, N-sets are applicable in various ways on any set of score structures that have temporal extent.

Z-chains, like N-sets, are a means of taking a low-dimensional projection (a sequence of orderable terms) and producing a higher-dimensional shape: a fractally-polyphonic set of recursively oriented hierarchical structures.

The temporal interactions of Z-chains are more complex than *polyphony* because Z-chains can be represented not as taking up temporal intervals, but recursively structured, gapped sequences of temporal intervals. Z-chains are therefore more difficult to reason about. But they are the result of a basic fact about musical

[24] And the polyphonic shape of a set of N-sets is descriptive of their structure in a way that the polyphonic shape of a "triangular" powercut is not (since we know its shape in advance of seeing the score).

structure: *that it can be discontinuous.* Z-chains structure parallels *across* temporal discontinuity.

One way of looking at Z-chains "polyphonically" (in order to structure aspects of their simultaneity) would be to take polyphonic projections at different structural levels (e.g. make *one* analysis by treating second-order Z-chains as temporal intervals, etc.).

Another approach to dealing with the complexity of a Z-chain picture is to analyze and restrict the "gappedness" of Z-chains (– we showed one way of doing this *structurally*, rather than e.g. using a *measurement* for the maximum gap-length). *Z-shapes* use simultaneities of Z-chains to produce more detailed shapes by using a simple method of identifying temporal relations of interest *without* analyzing the total (fractal-)polyphonic situation.

Each of the structuring algorithms presented can produce a *complete* combinatorial number of structural descriptions (i.e. no description is invalid as in a grammatical parse, or ignored as in a dictionary search), and operates by linking local structures into superstructures. These structuring algorithms are illustrative *examples* of an open-ended class of such methods.

These kinds of structuring algorithms are *general* in the sense that they can be used on any number of projections of a score, based on different featural dimensions. Such *projections* may themselves be produced by structuring algorithms (– or by any other method). The kinds of projections required e.g. for N-sets and Z-chains function using low-dimensional *relations*: N-sets require an equivalence relation on input structures, and Z-chains an ordering relation. There is an open number of such equivalence relations (and these may be based on structural properties, or may include heuristic semantics, e.g. injecting knowledge about diatonic scales as in key-assertions and diatonic equivalence classes for Z-shapes).

The total analytic map is therefore *very* open and flexible, containing any number of iterations and branchings of constructions and projections – this is *structural relativity*.[25]

Understanding is neither a point of view nor a value judgment ...effects as causes that relate modes of dynamic perception. They are neither definitions of concepts nor expressions of opinion, since all patterns of perception merge and metamorphose in the very act of exploration and discovery. They avoid value judgments, and serve as guides to insight and comprehension through* re-cognition *of the* dynamic structures *that occur in all processes. In replaying such patterns we are not taking any side but many sides, also the* inside*.[26]

6.5.3 Rhizomatic Geometry and Map-Making

In *A Thousand Plateaus*, Deleuze and Guattari describe two pseudo-sensory, information-analytic modes similar to McLuhan's – they describe *striated* (i.e. linear, analytic) space, versus *smooth* space (with an open number of dimensions), corresponding to "optic" versus "haptic" sensory modes [DeleuzeGuattari1980].

---

[25] [Resnik1997] (– discussed in Chapter 2).

[26] [MMU] Vol.3, p.30.

Deleuze and Guattari describe the *geometry* available in smooth space as being "rhizomatic" (contrasted with the hierarchical "tree" or "root").[27] In a *rhizome*, anything can (and "must") be connected to everything else.[28] A rhizome consists of a *multiplicity* of entities, types, and relations – a *growing* multiplicity.

The "dimensionality" of a multiplicity is $(n-1)$, signifying an unfinished, open dimensionality (– there is always another way of looking at things). The one dimension that is never attained (the -1) is the dimension of Unity. In the case of a score analysis, this means that we can take the score apart and put it back together in any number of different ways, making observations and discoveries about its structure, but we cannot expect to find any dimension that somehow makes sense of or explains the entirety.

Deleuze and Guattari contrast two methods for analysis: the *map* and the *tracing*. The *map* is rhizomatic in form, while a *tracing* is a linearization or dimensional reduction of a rhizome. The map is an open-ended analytic *process* (or field of potential), whereas a tracing is an analysis. Chomsky's grammatical trees are given as an example of a tracing – they are tracings not just because they are low-dimensional, but because they proceed from a sense of "unity" (the root of the tree is the sentence S), and enforce a grammatical shape.

> *Make a map, not a tracing. ... What distinguishes the map from the tracing is that it is entirely oriented toward an experimentation in contact*

[27] The rhizome is a conceptual geometry, rather than a mathematically formal one.

[28] [DeleuzeGuattari1980, p.7]

> *with the real. ... It fosters connections between fields ... The map is open and connectable in all of its dimensions; it is detachable, reversible, susceptible to constant modification. ... A map has multiple entryways, as opposed to the tracing, which always comes back 'to the same.' The map has to do with performance, whereas the tracing always involves an alleged 'competence.'...* [29]

The map is a virtual *place* on which connections are *traced*. A tracing is a *picture* of a *selected aspect* of the map (or of the rhizome that it maps).

Deleuze and Guattari say that we should "make a map." While we have described an open, rhizomatic dimensionality of constructions and projections (tracings), we don't know how to *make* them into a map.

> *Have we not, however, reverted to a simple dualism by contrasting maps to tracings, as good and bad sides? Is it not of the essence of the map to be traceable? Is it not of the essence of the rhizome to intersect roots and sometimes merge with them? Does not a map contain phenomena of redundancy that are already like tracings of its own? Does not a multiplicity have strata upon which unifications and totalizations, massifications, mimetic mechanisms, signifying power takeovers, and subjective attributions take root? ... It is a question of method:* ***the tracing should always be put back on the map.*** [30]

[29] [DeleuzeGuattari1980, p.12]

[30] [DeleuzeGuattari1980, p.13]

Here is what it might mean to *put the tracing back on the map*: suppose that a tracing is a low-dimensional summary of (for example) a score. One way that a summary is useful is as a generalization which can be used to categorize or compare the score with other scores, or to make a decision or prediction about the score. We might be able to represent this kind of summary abstractly as a code, or a sequence, tree, or graph of terms – this is the *tracing*. We could think of using the tracing to *annotate* or comment *on* the score as a way of putting the tracing back on "the map." This method means the *addition* of information to the score, rather than abstracting the tracing *from* it.

The "map" is not a high-dimensional *object*, it's an activity: the process of *discovering* and *relating* aspects of structure. The map is the principle that *affords* tracings.

In practice, we can't access or represent the simultaneity of information all at once – McLuhan's *simultaneity* doesn't imply that we can e.g. see all of the internet at once, only that we can see any part of it at any time, and therefore *relate* any set of parts of it in any way we can think of.[31]

Using linear information methods, or tracings, we relate to the global, simultaneous information environment. When the tracing is put back on the map, it becomes

[31] This is also what the philosopher Markus Gabriel means when he says "the world does not exist" – the world (for Gabriel) is not a *thing*, it's a simultaneity of ontological perspectives [SteinbauerGabriel2016].

part of the sphere of simultaneously *available* information. The "map" corresponding to a score is the virtual (i.e. potential) structural linking of any information that it can generate, as situated *on* the score itself.

The *score itself*, as the initial, unanalyzed data-object, functions as what Deleuze and Guattari call a "body without organs." If we think of a body *with* organs, this is conceived as a hierarchical, functional, fixed structure. A body *without* organs, by contrast, can be *organized* in a rhizome of different ways.

> *A body without organs is not an empty body stripped of organs, but a body upon which that which serves as organs ... is distributed according to crowd phenomena, in Brownian motion, in the form of molecular multiplicities. The desert is populous. Thus the body without organs is opposed less to organs as such than to the organization of the organs insofar as it composes an organism. The body without organs is not a dead body but a living body all the more alive and teeming once it has blown apart the organism and its organization. ... The full body without organs is a body populated by multiplicities.*[32]

6.5.4 Music-Analytic Tracings

The "map" of a score is a high-dimensional virtual "space" in which "all" tracings are representable and interrelatable. Because of the principle of structural relativity, this is a space with an *open* number of informational dimensions. The only

---

[32] [DeleuzeGuattari1980, p.30]

way we know how to visualize (or otherwise represent) any part of it is to make *projections* into a low dimensionality (tracings).

The kinds of analytic visualizations shown in this thesis are *tracings* showing a low-dimensional picture of structures selected and organized through low-dimensional comparisons. Using a few simple structuring algorithms, we very quickly ended up with more tracings than we could begin to show.

The tracings in this thesis are *visualizations*; another medium for observing tracings is *sonification*. For example, if we start with a score with a small number of note-features (e.g. pitch and timing, and maybe voice identity), we can maintain these features without alteration, and *add* new sonic features such as instrumental timbre, loudness, or articulation. This adds an analytic layer to a *musical* realization of the score.[33]

These kinds of sonifications are in some sense "performances" of scores. One approach to computational performance is to sound like an (expressive or natural) human (physical) performance (e.g. the approaches found in [Kirke+2012]), while the contrasting approach of structural sonification is based on making structural tracings of the score *into* the musical medium. A similar idea was expressed by Adorno:

[33] In [HandelmanSigler+2012] we sonified a selection of Z-chains using different instrumental timbres to create orchestrations, and in [HandelmanSigler2014] we took N-sets on pitch-class and rhythm-class, made projections through subsets of these to determine loudness and articulation (i.e. smooth vs detached notes).

> ...structural instrumentation would be to use every timbre and above all the mode of orchestration to make real all the structural elements that are indispensable to the articulation of the musical meaning: a procedure, then, that does not cloak music in orchestral garb, as the critics would say, but that translates its own articulation into that of sound. The principle of structural instrumentation is not one of a color calculus but of compositional clarity. ... Instead of treating the instrumentation as one parameter among others to which it is only abstractly related, the composition should develop the instrumentation from the meaning of the musical events. In that way it would become an authentic parameter, a concrete function of the music. The music would be the beneficiary of this treatment of instrumentation, since it would be one of the means of objectification music has needed ever since it ceased to lie cocooned inside the traditional formal scheme. ... every other treatment of the orchestral palette is just playing around.[34]

The habitual contemplation of the media of communication as art forms necessarily invokes the principle that the instruments of research are also art forms, magically distorting and controlling the objects of investigation.[35]

---

[34] [Adorno1999, p.212]

[35] [MMU] Vol.15, p.15

McLuhan was perhaps thinking of the distorting and controlling power of the instruments of investigating e.g. physics. In this case we are wistful about perhaps never "really" knowing the outside world except through the distortions of our senses and our instruments. But music is a thing of an essentially different nature: an *inner* world. It's *inherently* subjective – at the limit, it's like a secondary consciousness. In this case, "magically distorting and controlling the objects of investigation" may be just what is desired.

6.6 Music and History

We have touched on the implications of electric media for art and science. Here we examine how the electric age conceives of history in a different way from the linear age, and how we might approach historical music through metacreative probes.

The modern world abridges all historical times as readily as it reduces space. Every**where** ***and every*** **age** ***have become*** **here** ***and*** **now*****. History has been abolished by our new media. If prehistoric man is simply preliterate man living in a timeless world of seasonal recurrence, may not posthistoric man find himself in a similar situation? May not the upshot of our technology be the awakening from the historically conditioned nightmare of the past into a timeless present? Historic man may turn out to have been literate man. An episode.***[36]

We have seen that all of music history (short history though it is) is simultaneously available. However, historical music seems to lose its cultural importance, as

[36] [MMU] Vol. 6, p.7.

"classical music" audiences decline and we witness the obsolescence of the orchestra. We may be living in a "museum without walls," but to the wider culture, "literate" music is museum music. While there's little danger of the archive of scores and recordings being lost, we have already lost the technique and understanding of the great age of literate music. It is *not*, therefore, correct to say that every age of music is *here* and *now*. This state of affairs still lies in the desired future of metacreative musicology.

We will examine the kind of relationship we might have with the older music, if we can ***"approach tradition through the door of technological awareness."***[37] But McLuhan cautions us about what might happen when we pursue "linear" *goals* as a means of encountering the older music.

"Information overload leads to pattern recognition." This dictum falls under McLuhan's general principle that a process accelerated beyond a certain limit tends to *reverse* itself ("metamorphosis by chiasmus"). For example, "musical information" (i.e. music as such) has a certain *meaning* in the sense that encountering it produces a certain kind of *experience*. However, when we are taking a "big data" approach to music, it's *impossible* to generate the relevant musical meaning or experience. The same applies to language, images, etc. If we reach a point of *information overload*, we can no longer encounter music *as music*, but as "data" – leading to "pattern recognition." ***The ... visual [thinker], ... will naturally assume as a metaphor of any art form, that of a receptacle. Once this basis has***

---

[37] [McLuhan+1967, p.55]

been established, it is difficult to resist the need to use the receptacle as a waste basket. ... junking the classics by classification. Categories numb perception.[38]

Naturally, there's no real way to understand music except *as music*, so we must reverse the reversal and approach music *on its own terms* if any understanding is to be obtained. This shows why McLuhan thinks we must extend "consciousness itself" in order to make sense of the overwhelming information environment; it is this extension that would allow the second reversal.

Up until the advent of electro-magnetism the Western world had merely externalized and exchanged the products of thought and experience. We are quite unprepared for the much higher-educational demands of the present situation in which we must share the actual process of thought and experience of all mankind. But this new necessity not only compels us to inspect our own processes of thought and perception more carefully than in any previous age, but it urges us to the most earnest inquiry into all past art, literature and culture in order to benefit from all past discoveries about the processes of insight. ... Instead of merely establishing a perspective for past cultures, our tendency today is to reconstruct entire past cultures from within. ... the total reconstruction

[38] [MMU] Vol. 12, pp.11–15.

procedure is less visual than empathetic. It is a structure of simultaneous and organic inter-relationships.[39]

The vision is that we can approach the lost past of literate music technique through expanded music-technological consciousness. Naturally it will never be "the same" – it may be that we will have to drop some of our attachment to historical authenticity and veneration of composers-as-authors in order to revive some of their discoveries.

In the electric age, we see that sampling, mashups, and remixes are common musical practices, legitimizing creative appropriation and musical material as common property. In the metacreative future, we can expect to see reworkings that probe the classics, not only by sampling and signal processing, but through transformation of pattern, shape, harmony, and other inner-dimensions, as well as new works and spaces that use techniques and processes abstracted from earlier music. These works could be at once musical and music-educational in a technical as well as a historical sense.

As we have already seen, the concepts of "individual" composer and of "historical time" are obsolesced in the electric age – still available for consideration, but now *within* a sphere of electric simultaneity. In "The Death of the Author," Barthes says that the author as expressive individual has receded in favor of language and writing itself, and the simultaneity of potentials in its *reading* – we can no longer rely upon

[39] [MMU] Vol.7 pp.14–15.

the unifying perspective of the person and the intentions of the author.[40] In fact, as Foucault points out in "What is an Author?," the very notion of a *work* or body of work entails authorial assumptions.[41] So reliance on authorship for e.g. definition of a "style" opens up problems discussed by Barthes and Foucault: the supposed unity of expression, technique, and intention of an author is, after all, not certain.

Foucault discusses the differences in cultural notions of *authorship* between the arts and sciences in the manuscript and print ages: in the earlier age, literary narratives were often anonymous, while scientific texts relied on "authority"; in the print age, the literary author was valorized while "scientific discourses began to be received for themselves, in the anonymity of an established or always redemonstrable truth." In the age of art-science fusion, can we anticipate another change? Will we start to think of musical techniques not as individual styles of expression, but as technologies for creation?

But in the same essay, Foucault cautions: "It would be pure romanticism...to imagine a culture in which the fictive [or music] would operate in an absolutely free state, in which fiction [or music] would be put at the disposal of everyone and would develop without passing through something like a necessary or constraining figure."[42] Therefore we back off from an ultimate "formalism," knowing that a semantic, cultural, and contextual level is always in effect. Whereas we can conceive of music as

[40] [Barthes1968]

[41] [Foucault1979]

[42] [Foucault1979]

an open material, individual culture-creators (or meta-creators) will continue to innovate and drive the larger culture onward. As we develop new technological ways of navigating musical possibility space, *map-making* may be based on human meaning and value.

The promise of posthistoricity, of "seasonal recurrence" is that the older artists can *continue* to contribute. "The great work of art belongs to history," wrote Malraux, "but it does not belong to history alone."[43] This is Malraux's notion in *Metamorphosis of the Gods*: that art cannot be limited to characterization by its *historical* location, nor does it endure because it is timelessly *eternal*, but rather that it transcends time through constant resuscitation, metamorphosis, and transformation in meaning.[44] "Metamorphosis is not an accident, it is the very law of life of the work of art. ... and what the great work of art sustains is not a monologue, however authoritative, but an invincible dialogue."[45]

6.7 Conclusion

McLuhan's celebrated aphorism ***The Medium is the Message*** means that ***the* form *of communication [is] the basic art situation, which is more significant than the information or idea 'transmitted.'***[46] This is because

[43] [Malraux1951], [Allan2013].

[44] [Allan2013].

[45] [Malraux1957], [Allan2013].

[46] [MMU] Vol. 15 p.6.

media extend our senses; they form a process-environment through which we encounter the world. For example, the expressive potential of *language* is more fundamental than any particular thing we might say to one another. For McLuhan, language was our ***first technology for letting go of the environment in order to grasp it in new ways***.[47] As we develop new media, what at first seem to be containers for the old media (writing as encoded speech, "horseless carriages," TV as "movie-in-the-home," etc.) ultimately turn out to be nothing of the kind. Computation, then, is *not* a "tool" for (among other things) analyzing and producing music. Computation, like language, extends consciousness. As a virtual environment, musical metacreative technology will function to extend human musical consciousness, expanding our musical understanding and ability.

It is even possible that understanding of the musical sense will be of broader utility in AI. Music has the distinction of being a meaning- and experience-bearing structural medium that can operate *without* reference to external knowledge e.g. about objects in the world, human motivations, etc. Therefore we have a chance of building an abstract semantic technology before AI research solves the hard general problem of human meaning. This technology could be applied to many areas of design, perception, and organization. ***"All the arts aspire to the condition of music," said Walter Pater, and under conditions of instant information***

[47] [MMU] Vol. 11, p.2.

the only possible rationale or means of order involves us in the musical structuring of experience.[48]

[48] [MMU] Vol. 5, p.5.

REFERENCES

[Aarden2003] Aarden, Bret. *Dynamic Melodic Expectancy.* PhD Thesis: School of Music, Ohio State University. 2003.

[Adams1976] Adams, Charles R. "Melodic Contour Typology." *Ethnomusicology.* 20.2, 179-215. 1976.

[Adiloglu+2006] Adiloglu, Kamil, Thomas Noll and Klaus Obermayer. "A paradigmatic approach to extract the melodic structure of a musical piece." *Journal of New Music Research*, 35.3, 221–236. 2006.

[Adorno1999] Adorno, Theodor W. *Sound Figures.* Rodney Livingstone, trans. Stanford University Press. 1999.

[Ahlbäck2007] Ahlbäck, Sven. "Melodic similarity as a determinant of melody structure." *Musicae Scientiae*, 11.1_suppl, 235-280. 2007.

[Allan2013] Allan, Derek. *Art and Time.* Cambridge Scholars Publishing. 2013.

[Allen1983] Allen, James F. "Maintaining knowledge about temporal intervals." *Communications of the ACM,* 26.11, 832-143. 1983.

[Barthes1968] Barthes, Roland. "Death of the Author." Richard Howard, trans. UbuWeb: http://www.ubu.com/aspen/aspen5and6/threeEssays.html#barthes 1968 (accessed 2015).

[Bellman2005] Bellman, Héctor. "About the Determination of Key of a Musical Excerpt". *3rd International Symposium on Computer Music Modelling and Retrieval*, 187–203. 2005.

[Bod2001] Bod, Rens. "Memory-based models of melodic analysis: challenging the gestalt principles." *Journal of New Music Research*, 30.3, 27–36. 2001.

[Boulanger-Lewandowski+2012] Boulanger-Lewandowski, Nicolas, Yoshua Bengio, and Pascal Vincent. "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription." *International Conference on Machine Learning.* 2012.

[Buteau+2000] Buteau, Chantal, and Guerino Mazzola. "From Contour Similarity to Motivic Topologies." *Musicae Scientiae* 3.2, 125–149. 2000.

[Buteau+2008] Buteau, Chantal and Guerino Mazzola. "Motivic analysis according to Rudolph Réti: formalization by a topological model." *Journal of Mathematics and Music*, 2.3, 117–134. 2008.

[Cambouropoulos+2000] Cambouropoulos, Emilios, and Gerhard Widmer. "Automated Motivic Analysis via Melodic Clustering." *Journal of New Music Research*, 29.4, 303-317. 2000.

[Cambouropoulos2000b] Cambouropoulos, Emilios. "Extracting 'Significant' Patterns from Musical Strings: Some Interesting Problems." *London String Days 2000 workshop*, 2000.

[Cambouropoulos2001] Cambouropoulos, Emilios. "Pattern Processing in Melodic Sequences: Challenges, Caveats and Prospects." *Computers and the Humanities*, 35, 9–21. 2001.

[Cambouropoulos2001b] Cambouropoulos, E. "The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing." *Proceedings of the International Computer Music Conference (ICMC)*, 2001.

[Cambouropoulos2003] Cambouropoulos, Emilios. "Musical pattern extraction for melodic segmentation." *Proceedings of ESCOM5.* 2003.

[Cambouropoulos2006] Cambouropoulos, Emilios. "Musical parallelism and melodic segmentation." *Music Perception*, 23.3, 249-268. 2006.

[Chai+2005] Chai, Wei and Barry Vercoe. "Detection of key change in classical piano music." *Proceedings of the 6th International Conference on Music Information Retrieval.* 2005.

[Cherla+2014] Cherla, Srikanth, Tillman Weyde, and Artur d'Avila Garcez. "Multiple Viewpoint Melodic Prediction with Fixed-Context Neural Networks." *ISMIR.* 2014.

[Cherla+2015] Cherla, Srikanth, Son N. Tran, Tillman Weyde, and Artur d'Avila Garcez. "Hybrid Long- and Short-Term Models of Folk Melodies." *ISMIR*. 2015.

[Chew2006] Chew, Elaine. "Slicing It All Ways: Mathematical Models for Tonal Induction, Approximation, and Segmentation Using the Spiral Array." *INFORMS Journal on Computing*, 18.3, 305-320. 2006.

[Cilibrasi+2004] Cilibrasi, Rudi, Paul Vitányi, and Ronald de Wolf. "Algorithmic Clustering of Music Based on String Compression." *Computer Music Journal*, 28.4, 49–67. 2004.

[Clarke2003] Clarke, Desmond M. *Descartes's Theory of Mind*. Oxford University Press, 2003.

[Cohen1977] Cohen, Joel E. "Food webs and the dimensionality of trophic niche space." *Proceedings of the National Academy of Sciences,* 74.10, 4533-4536. 1977.

[Conklin+2001] Conklin, Darrell, and Christina Anagnostopoulou. "Representation and Discovery of Multiple Viewpoint Patterns." *Proceedings of the 2001 International Computer Music Conference*. 2001.

[Conklin2008] Conklin, Darrell. "Discovery of Distinctive Patterns in Music." *International Workshop on Machine Learning and Music*. 2008.

[Conklin2010] Conklin, Darrell. "Distinctive patterns in the first movement of Brahms' String Quartet in C minor." *Journal of Mathematics and Music*, 4.2, 85-92. 2010.

[Conklin+2011] Conklin, Darrell, and Christina Anagnostopoulou. "Comparative pattern analysis of Cretan folk songs." *Journal of New Music Research*, 40.2, 119-125. 2011.

[Crawford+2001] Crawford, Tim, Costas S. Iliopoulos, Russel Winder, and Haifeng Yu. "Approximate musical evolution." *Computers and the Humanities*, 35.1, 55-64. 2001.

[DeleuzeGuattari1980] Deleuze, Gilles, and Felix Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. Brian Massumi, trans. University of Minnesota Press. 1987.

[Eerola2007] Eerola, Tuomas, and Micah Bregman. "Melodic and Contextual Similarity of Folk Song Phrases." *Musicae Scientiae*, Discussion Forum 4A, 211–233. 2007.

[Fishburn1985] Fishburn, Peter C. "Interval Graphs and Interval Orders." *Discrete Mathematics*, 55, 135-149. 1985.

[Friedmann1985] Friedmann, Michael L. "A methodology for the discussion of contour: Its application to Schoenberg's Music." *Journal of Music Theory*, 29.2, 223-248. 1985.

[Forte1964] Forte, Allen. "A Theory of Set-Complexes for Music." *Journal of Music Theory*, 8.2, 136-183. 1964.

[Forte1973] Forte, Allen. *The structure of atonal music.* Yale University Press, 1973.

[Foucault1979] Foucault, Michel "What is an Author?" Kari Hanet, trans. *Screen*, 20.1, 13–34. 1979.

[Ganguli+2016] Ganguli, Kaustuv Kanti, Sankalp Gulati, Xavier Serra, Preeti Rao. "Data-driven exploration of melodic structures in Hindustani music." *ISMIR*. 2016.

[Gilbert+2007] Gilbert, E. and Darrell Conklin. "A Probabilistic Context-Free Grammar for Melodic Reduction." *International Workshop on Artificial Intelligence and Music, IJCAI*, 83–94. 2007.

[Goienetxea+2016] Goienetxea, Izaro, Kerstin Neubarth, and Darrell Conklin. "Melody classification with pattern covering." *International Workshop on Machine Learning and Music*. 2016.

[Golumbic1980] Golumbic, Martin Charles. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York. 1980.

[Grachten+2005] Grachten, Maarten, JL Arcos, and Ramon Lopez De Mántaras. "Melody retrieval using the implication/realization model." *ISMIR*. 2005.

[Groves2016] Groves, Ryan. "Automatic Melodic Reduction Using a Supervised Probabilistic Context-Free Grammar." *ISMIR*. 2016.

[Gulati+2015] Gulati, Sankalp, Joan Serrá and Xavier Serra. "Improving Melodic Similarity in Indian Art Music Using Culture-Specific Melodic Characteristics." *ISMIR*. 2015.

[Gulati+2016] Sankalp Gulati, Joan Serrá, Kaustuv K. Ganguli, Sertan Şentürk and Xavier Serra. "Time-delayed melody surfaces for Rāga recognition." *ISMIR*. 2016.

[Gulati+2016b] Gulati, Sankalp, Joan Serra, Vignesh Ishwar, Sertan Şentürk, and Xavier Serra. "Phrase-based Rāga recognition using vector space modelling." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016.

[Hamanaka+2005] Hamanaka, Masatoshi, Keiji Hirata, and Satoshi Tojo. "ATTA: Automatic Time-Span Tree Analyzer Based on Extended GTTM." *ISMIR*. 2005.

[Hamanaka+2007] Hamanaka, Masatoshi, Keiji Hirata, and Satoshi Tojo. "Fatta: Full Automatic Time-Span Tree Analyzer." *International Computer Music Conference (ICMC)*. 2007.

[Hamanaka+2009] Hamanaka, Masatoshi, and Satoshi Tojo. "Interactive Gttm Analyzer." *ISMIR*. 2009.

[HandelmanSigler+2012] Handelman, Eliot, Andie Sigler, and David Donna. "Automatic Orchestration for Automatic Composition" *Proceedings of the International Workshop on Musical Metacreation*. AAAI Press. 2012.

[HandelmanSigler2013] Handelman, Eliot and Andie Sigler. "Key Induction and Key Mapping Using Pitch-Class Set Assertions." *International Conference on Mathematics and Computation in Music*. Springer. 2013.

[HandelmanSigler2014] Handelman, Eliot and Andie Sigler. "Artik: Articulation and Dynamics from Structural Pattern Analysis" *Proceedings of the International Workshop on Musical Metacreation*. AAAI Press. 2014.

[Hiller+1966] Hiller, Lejaren and Calvert Bean. "Information Theory Analyses of Four Sonata Expositions." *Journal of Music Theory*, 10.1, 96-137. 1966.

[Huovinen+2008] Huovinen, Erkki, and Atte Tenkanen. "Bird's-Eye Views of the Musical Surface: Methods for Systematic Pitch-Class Set Analysis." *Music Analysis*, 26.1–2, 159–214. 2007.

[Huron1996] Huron, David. "The Melodic Arch in Western Folksongs." *Computing in Musicology*, 10, 3–23. 1996.

[Huron2006] Huron, David. *Sweet Anticipation: Music and the psychology of expectation*. MIT Press, 2006.

[Jannsen+2015] Janssen, Berit, Peter van Kranenburg, Anja Volk. "A comparison of symbolic similarity measures for finding occurrences of melodic segments." *ISMIR*. 2015.

[Juhasz+2009] Juhász, Zoltán, and János Sipos. "A Comparative Analysis of Eurasian Folksong Corpora, using Self Organizing Maps." *journal of Interdisciplinary Music Studies*, 4.1, 2010.

[Kim+2000] Kim, Youngmoo E., Wei Chai, Ricardo Garcia, and Barry Vercoe. "Analysis of a Contour-Based Representation for Melody." *ISMIR*. 2000.

[Kirke+2012] Kirke, Alexis, and Exuardo Reck Miranda, eds. *Guide to Computing for Expressive Music Performance*. Springer. 2012.

[Knopke+2009] Knopke, Ian, and Frauke Jürgensen. "A System for Identifying Common Melodic Phrases in the Masses of Palestrina." *Journal of New Music Research*, 38.2, 171–181. 2009.

[Kranenburg2010] van Kranenburg, Peter. *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*. PhD Thesis: SIKS, the Dutch Research School for Information and Knowledge Systems, Utrecht University. 2010.

[Kranenburg+2013] van Kranenburg, Peter, Anja Volk, and Frans Wiering. "A Comparison between Global and Local Features for Computational Classification of Folk Song Melodies." *Journal of New Music Research*, 42.1, 1-18. 2013.

[Krokhin+2003] Krokhin, Andrei, Peter Jeavons, and Peter Jonsson. "Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra." *Journal of the ACM*, 50.5, 591-640. 2003.

[Krumhansl1990] Krumhansl, Carol. *Cognitive Foundations of Musical Pitch*. Oxford University Press. 1990.

[Lartillot+2004] Lartillot, Olivier and Emmanuel Saint-James. "Automating Motivic Analysis through the Application of Perceptual Rules." *Music Query: Methods, Models, and User Studies*, 13, 37–92. 2004.

[Lartillot2004] Lartillot, Olivier. "A Musical Pattern Discovery System Founded on a Modelling of Listening Strategies" *Computer Music Journal*, 28.3, 53–67, 2004.

[Lartillot2005] Lartillot, Olivier. "Multi-dimensional motivic pattern extraction founded on adaptive redundancy filtering." *Journal of New Music Research*, 34.4, 375-393. 2005.

[Lartillot+2007] Lartillot, Olivier and Petri Toiviainen. "Motivic matching strategies for automated pattern extraction." *Musicae Scientiae*, Discussion Forum 4A, 281–314. 2007.

[Leeuwenberg+2013] Leeuwenberg, Emanuel and Peter A. van der Helm. *Structural Information Theory: The Simplicity of Visual Form*. Cambridge University Press. 2013.

[LerdahlJackendoff1983] Lerdahl, Fred, and Ray Jackendoff. *A Generative Theory of Tonal Music.* MIT press. 1983.

[Lester1999] Lester, Joel. *Bach's Works for Solo Violin: Style, Structure, Performance*. Oxford University Press. 1999.

[Lewin2007] Lewin, David. "Making and Using a Pcset Network for Stockhausen's Klavierstück III." *Musical Form and Transformation: Four Analytic Essays*. Oxford University Press. 2007.

[Loubouin+2016] Louboutin, Corentin and David Meredith. "Using general-purpose compression algorithms for music analysis." *Journal of New Music Research*, 45.1, 1-16. 2016.

[Malraux1951] Malraux, André. *The Metamorphosis of the Gods*. Stuart Gilbert, trans. Secker and Warburg. 1960 (1951).

[Malraux1957] Malraux, André. *The Voices of Silence*. Stuart Gilbert, trans. Princeton University Press. 1978 (1957).

[Marsden2001] Alan Marsden. "Representing Melodic Patterns as Networks of Elaborations." *Computers and the Humanities*, 35, 37–54. 2001.

[Marsden2012] Alan Marsden. "Interrogating Melodic Similarity: A Definitive Phenomenon or the Product of Interpretation?" *Journal of New Music Research*, 41.4, 323-335. 2012.

[Martorell+2016] Martorell, Augistín and Emilia Gómez. "Contextual Set-Class Analysis." In *Computational Music Analysis*, David Meredith, ed. Springer, 2016.

[Marvin+1987] Marvin, Elizabeth West and Paul A. Laprade. "Relating Musical Contours: Extensions of a Theory for Contour." *Journal of Music Theory*, 31.2, 225-267. 1987.

[Matsubara+2014] Matsubara, Masaki, Keiji Hirata, and Satoshi Tojo. "Distance in Pitch Sensitive Time-span Trees." *Proceedings of International Computer Music Conference / Sound and Music Computing*. 2014.

[McKee+1999] McKee, Terry A. and F. R. McMorris. *Topics in Intersection Graph Theory*. Society for Industrial and Applied Mathematics. 1999.

[MMU] McLuhan, Marshall. *Marshall McLuhan Unbound*. Eric McLuhan and Terrence Gordon, eds. Gingko Press, 2005.

[McLuhan1964)] McLuhan, Marshall. *Understanding Media: The Extensions of Man*. Cambridge: MIT Press. 1994 (1964).

[McLuhan+1967] McLuhan, Marshall, and Quentin Fiore. *The Medium is the Massage: An Inventory of Effects*. Penguin. 1967.

[Meredith+2002] Meredith, David, Kjell Lemstr om, and Geraint A. Wiggins. "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music" *Journal of New Music Research*, 31, 321–345. 2002.

[Meredith2016] . Meredith, David. "Analysing Music with Point-Set Compression Algorithms." In *Computational Music Analysis*, David Meredith, ed., Springer. 2016.

[Merleau-Ponty1945] Merleau-Ponty, Maurice. *Phenomenology of Perception*. Colin Smith, trans. Routledge, 2005 (1945).

[Meyer1956] Meyer, Leonard B. *Emotion and Meaning in Music*. University of Chicago Press. 1956.

[Morris1987] Morris, Robert. *Composition with Pitch Classes.* Yale University Press. 1987.

[Morris1993] Morris, Robert. "New Directions in the Theory and Analysis of Musical Contour." *Music Theory Spectrum*, 15.2, 205-228. 1993.

[Mullensiefen+2007] Müllensiefen, Daniel, and Klaus Frieler. "Modelling experts' notions of melodic similarity." *Musicae Scientiae*, 11.1_suppl., 183–210. 2007.

[Narmour1990] Narmour, Eugene. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press. 1990.

[Noland+2006] Noland, Katy and Mark Sandler. "Key Estimation Using a Hidden Markov Model." *ISMIR*. 2006.

[Olariu1991] Olariu, Stephan. "An optimal greedy heuristic to color interval graphs." *Information Processing Letters*. 37, 21-25. 1991.

[Pearce2005] Pearce, Marcus Thomas. *The Construction and Evaluaion of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD Thesis: Department of Computing, City University, London. 2005.

[Pearce+2010] Pearce, Marcus T., Daniel Müllensiefen, and Geraint A. Wiggins. "The role of expectation and probabilistic learning in auditory boundary perception: A model comparison." *Perception*, 39.10, 1367-1391. 2010.

[Pearce+2017] Pearce, Marcus and Daniel Müllensiefen. "Compression-based Modelling of Musical Similarity Perception." *Journal of New Music Research*, 46.2, 135-155, 2017.

[Pikrakis+2006] Pikrakis, Aggelos, and Sergios Theodoridis. "Classification of Musical Patterns Using Variable Duration Hidden Markov Models." *IEEE Transactions on Audio, Speech, and Language Processing*, 14.5. 2006.

[Pinto2012] Pinto, Alberto. "A spectral clustering method for musical motifs classification." *International Symposium on Computer Music Modelling and Retrieval (CMMR)*. 2012.

[Quinn1997] Quinn, Ian. "Fuzzy Extensions to the Theory of Contour." *Music Theory Spectrum*, 19.2, 232-263. 1997.

[Resnik1997] Resnik, Michael D. *Mathematics as a Science of Patterns*. Oxford University Press. 1997.

[Risk+2015] Risk, Laura, Lillio Mok, Andrew Hankinson and Julie Cumming. "Melodic Similarity in Traditional French-Canadian Instrumental Dance Tunes." *ISMIR*. 2015.

[Rolland2001] Rolland, P-Y. "FIExPat: Flexible extraction of sequential patterns." *IEEE International Conference on Data Mining*. 2001.

[Russell1945] Russell, Bertrand. *A History of Western Philosophy*. Simon and Schuster. 1945.

[Schmuckler1999] Schmuckler, Mark A. "Testing Models of Melodic Contour Similarity." *Music Perception*, 10.3, 295-326. 1999.

[Sapp2011] Sapp, Craig Stuart. *Computational Methods for the Analysis of Musical Structure*. PhD Thesis: Department of Music, Stanford University. 2011.

[Schoenberg1967] Schoenberg, Arnold. *Fundamentals of Musical Composition*. Gerald Strange and Leonard Stein, eds. Faber and Faber. 1967.

[Schoenberg1969] Schoenberg, Arnold. *Structural Functions of Harmony*. Leonard Stein, ed. Norton. 1969.

[Sider2011] Sider, Theodore. *Writing the Book of the World*. Oxford University Press. 2011.

[Silva+2015] Silva, Diego Furtado., Vinícius Mourão Alves de Souza, and Gustavo Enrique de Almeida Prado Alves Batista. "Music Shapelets for Fast Cover Song Recognition." *ISMIR*. 2015.

[Silva+2016] Silva, Diego F., Chin-Chia M. Yeh, Gustavo E. A. P. A. Batista, and Eamonn Keogh. "SiMPle: Assessing Music Similarity Using Subsequences Joins." *ISMIR*. 2016.

[SteinbauerGabriel2016] Steinbauer, Anja and Markus Gabriel. "New Realism: Markus Gabriel." *Philosophy Now*, 113. 2016.

[Temperley2002] Temperley, David. "A Bayesian Approach to Key-Finding." *Music and Artificial Intelligence*, C. Anagnostopoulou, M. Ferrand and A. Smaill, eds., 195-206. 2002.

[Temperley2007] Temperley, D. *Music and Probability*. MIT Press. 2007.

[Temperley2001] Temperley, D. *The Cognition of Basic Musical Structures*. MIT Press. 2001.

[Thom+2002] Thom, Belinda, Christian Spevak, and Karin Höthker. "Melodic segmentation: evaluating the performance of algorithms and musical experts." *Proceedings of the International Computer Music Conference (ICMC)*. 2002.

[Tojo+2013] Tojo, Satoshi, and Keiji Hirata. "Distance and Similarity of Time-Span Trees." *Journal of Information Processing*, 21.2, 256-263. 2013.

[Valero+2002] Valero, David Rizo, and Jose Manuel Inesta. "Tree-structured representation of melodies for comparison and retrieval." *International Workshop on Pattern Recognition in Information Systems (PRIS)*. 2002.

[Wattenberg2002] Wattenberg, Martin. "Arc Diagrams: Visualizing Structure in Strings." *IEEE Symposium on Information Visualization*.

[Whitehead1925] Whitehead, A.N. *Science and the Modern World*. Macmillan Publishing Co. 1925.

[Whitehead1929] Whitehead, A.N. *Process and Reality*. Macmillan Publishing Co. 1978 (1929).

[Youngblood1958] Youngblood, Joseph E. "Style as Information." *Journal of Music Theory*, 2.1, 24-35. 1958.