Dictionary-Based Analysis/Synthesis and Structured Representations of Musical Audio

Graham Boyes

Music Technology
McGill University
Montreal, Canada

December 2011

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Arts.

2011/12/13

Abstract

In the representation of musical audio, it is common to favour either a signal or symbol interpretation, where mid-level representation is an emerging topic. In this thesis we investigate the perspective of structured, intermediate representations through an integration of theoretical aspects related to separable sound objects, dictionary-based methods of signal analysis, and object-oriented programming. In contrast to examples in the literature that approach an intermediate representation from the signal level, we orient our formulation towards the symbolic level. This methodology is applied to both the specification of analytical techniques and the design of a software framework. Experimental results demonstrate that our method is able to achieve a lower Itakura-Saito distance, a perceptually-motivated measure of spectral dissimilarity, when compared to a generic model and that our structured representation can be applied to visualization as well as agglomerative post-processing.

Abrégé

Dans la représentation du signal audio musical, il est commun de favoriser une interprétation de type signal ou bien de type symbole, alors que la représentation de type mi-niveau, ou intermédiaire, devient un sujet d'actualité. Dans cette thèse nous investiguons la perspective de ces représentations intermédiaires et structurées. Notre recherche intègre tant les aspects théoriques liés à des objets sonores séparables, que les méthodes d'analyse des signaux fondées sur des dictionnaires, et ce jusqu'à la conception de logiciels conus dans le cadre de la programmation orienté objet. Contrairement aux exemples disponibles dans la littérature notre approche des représentations intermédiaires part du niveau symbolique pour aller vers le signal, plutôt que le contraire. Cette méthodologie est appliquée non seulement à la spécification de techniques analytiques mais aussi à la conception d'un système logiciel afférent. Les résultats expérimentaux montrent que notre méthode est capable de réduire la distance d'Itakura-Saito, distance fondé sur la perception, ceci en comparaison à une méthode de décomposition générique. Nous montrons également que notre représentation structurée peut être utilisée dans des applications pratiques telles que la visualisation, l'agrégation post-traitement ainsi qu'en composition musicale.

Acknowledgments

First and foremost I would like to extend my sincerest gratitude to my supervisor Philippe Depalle. His insight and approach to problem solving has inspired me to always consider how an idea can be developed further.

My thanks to Marlon Schumacher, for his ability to ask a pertinent question and seemingly endless supply of interesting projects to work on. In particular, I am grateful to him for suggesting the collaboration featured in this thesis.

Thanks to all the members of my family. To my mother, whose encouragement and support has enabled me to pursue my interests and passions. To my father, who raised me to value education and (whether he did so intentionally or not) instilled in me an analytical spirit that I find myself continually drawing upon. To my brother, whose curiosity and penchant for takings things apart has always been an inspiration. To my sister, whose ambition and diligence continues to impress and motivate me. To Lisa, Dennis, and Jan for continued support and advice.

Finally, my thanks to Moana whose energy and enthusiasm helped motivate me over the course of this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis investigates the middle-ground between two common representations of musical information. The first is a system of symbols understood by musicians working in a particular idiom. In the Western tradition, a familiar example of such a system is the *score* where the fundamental unit of information is the note. The largely prescriptive nature of the score-based representation can be contrasted with that of music as an audio signal. This *waveform* representation is based on the measurement of an acoustic phenomenon, that is a representation of the information associated with the physical domain of musical sound. Put another way, it is principally descriptive, i.e. an artifact of a process. When discretized, the lowest level of information provided by the signal representation is a sampled value associated with the underlying process.

Depending on the context, each of the units addressed above can offer certain advantages. For example, a note-based representation facilitates the analysis and manipulation of musical elements at the symbolic level, i.e. in the language used by musicians. More generally, a symbolic representation lends itself to abstract comparison and classification. On the other hand, music represented as a sequence of samples can be treated (or transformed) using techniques from digital signal processing.

In this thesis, we integrate symbolic features into quantitative analysis in an effort to capitalize on the desirable properties associated with each representational paradigm. However, rather than attempt to determine pitch or other relatively high-level attributes from the signal itself, we look to sources of information from a different modality. In particular, we explore a *top-down* approach to signal analysis starting from the basic elements of a

score-based representation. In this investigation, we are motivated by a qualitative description of sound in terms of separable objects and we have treated a symbolic representation as a starting point for the dissection of sounds into invertible structures with varying degrees of abstraction.

In this regard, we believe that the representation of an object is inextricably linked to that which is used to represent it. Though this may appear to be a vague statement, from the perspective of computation, it is a practical consideration. In order to explore the topic of structured representations of audio, we have developed a similarly structured interface to data and functionality in a software framework. Here, the motivation was two-fold. First, to test the applicability of our conceptual model to software design. Second, to develop a working framework to facilitate experimentation both in an empirical sense and as part of a collaborative project oriented towards music creation. Further, software design in this context has constituted an interesting point of convergence between the opposed perspectives presented above. Here, it is common for signal and symbol representations to coexist and research in the area of computer-aided composition has highlighted some benefits of their interaction [9].

Aside from a viable approach to software design, the exploration of structured representations of musical audio requires versatile signal modeling techniques as, from the signal perspective, this class of signals features disparate behaviours.

Dictionary-Based Methods (DBMs) are a category of analysis techniques used to approximate a signal according to a set of elementary waveforms, or *atoms*. The result of such an analysis is a model consisting of discrete objects with interpretable attributes, in some sense reminiscent of notes.

The representation of musical signals by DBMs has many desirable properties. In particular, they are flexible with regards to the members of an analysis set and are thus able to represent a variety of sound classes more directly, i.e. they can yield a model consisting of fewer, more salient elements. In this thesis, we are motivated by the possibility that such models are more readily interpretable and lend themselves to further structured/mid-level representations.

One significant drawback to DBMs however, is their computational complexity. This facet is directly related to their flexibility in the sense that, in order to find the most representative set of elementary functions for a given signal, one must theoretically consider all combinations of elementary functions in all parameter configurations. Of course, this

is impossible in practice and we are required to make decisions regarding the contents of the dictionary in order to ensure the tractability of a decomposition while facilitating a desirable representation in terms of meaningfulness and sparsity. Therefore, where possible, it is advantageous to include as much information regarding the expected features of a given analysis target as possible. Furthermore, we consider that the addition of such information can contribute to structured representation. In accordance with our *top-down* methodology, we consider fundamental symbolic entities, namely sound source and notated musical context.

The aforementioned considerations associated with DBMs pertain to the members of the analysis set, i.e. the dictionary. Aside from this feature, DBMs are also concerned with strategies to amass atoms over the course of an analysis to *decompose* a signal. Among these techniques, a popular sub-optimal technique is Matching Pursuit (MP), which operates by extracting the atom most correlated with the signal at each iteration. In consideration of a structured extension to MP, we introduce a variant where we extract a set of atoms based on an inductive principle pertaining to the relatedness of components at each iteration. To evaluate our results, in addition to the typical measurement of the signal-to-residual ratio, we consider a perceptually-motivated distance measurement based on spectral similarity, namely the Itakura-Saito distance.

The organization of this thesis is as follows:

In Ch. 2, we examine crossdisciplinary approaches to the representation of sound that reconcile the two opposing descriptive modalities. As mentioned, this discussion is centred around an interpretation of sound in terms of separable entities, or *objects*. As such, we are motivated by an associated conceptual model developed by Pierre Schaeffer[57]. After introducing this method of qualitative analysis, we compare and contrast it with features of other perspectives, including quantitative *analysis/synthesis* and the representation of sound in computational contexts. As regards the tools of quantitative analysis, we draw particular attention to the relatively recent consideration of a *mid-level* representation.

In Ch. 3, we present DBMs. In doing so, we first address the associated theoretical foundations and characterize this class of techniques as a generalization of signal representation by means of orthogonal bases. Following this presentation, we turn our attention to examples in the literature where the various stages of a dictionary-based analysis procedure have been customized to consider features expected in a musical signal.

In Ch. 4, we present the software framework developed to serve as an intermediary

between dictionary-based methods and the quasi-symbolic representation of sound objects using an object-oriented data representation. In this chapter, we demonstrate the functionality of its core components, and present a collaborative project where it was featured in the context of computer-aided composition.

In Ch. 5 we present experiments where we have considered symbolically-oriented features at multiple stages of the analysis/synthesis process. The first series of experiments demonstrates the properties of our MP variant and compares its performance to a generic decomposition in the modeling of sound source-specific signals. In the second series of experiments, we use a score-based representation as a guide for atomic decomposition and model visualization. In the final series we explore how a structured model can be used in agglomerative post-processing.

Chapter 2

Perspectives on the Representation of Sound Objects

2.1 Introduction

In this chapter, we are concerned with perspectives that treat sound as being composed of separable entities with associated attributes. Generally speaking, through its respective approach to analysis, each perspective adds an aspect of identity to the physical phenomena it is concerned with and is indicative of a reconciliation between two descriptive poles.

The first pole is characterized by the representation of sound in *symbolic* terms, i.e. a sound itself is a signifier pointing to the concept of an object. This interpretation is appealing in that it constitutes a high-level description oriented towards abstract comparison and classification. On the other hand, it could be considered limiting in its description of the acoustic (or aesthetic) properties of a given sound. The second pole is characterized by the representation of sound as a *signal*, that is a measurement of a physical quantity. This representation is appealing in that it accounts for the specificities of a particular sound to the degree that a convincing facsimile can be rendered. On the other hand, the separation of a sound signal into easily interpretable or meaningful elements can be difficult.

In the sections that follow, we examine three general perspectives and the descriptive approaches associated with each. Here, we are motivated by the applicability of the approach to structured representations of audio.

In Sec. 2.2, we discuss descriptive systems associated with a qualitative analysis of

sound. Here, we present the *sound object* of Pierre Schaeffer and further extensions to this conceptual model.

In Sec. 2.3, we turn to a perspective that is not only concerned with the analysis of a given sound, but also the ability to modify and reconstruct it from the analytical result, that is an *analysis/synthesis* representational paradigm. Historically, the models associated with this perspective have tended towards a signal interpretation, however a more recent development in this area is the concept of a *mid-level* intermediate representation and this approach is presented in detail.

In Sec. 2.4, we consider the representation of sound in reference to data and computational paradigms where our general concept of *analysis* pertains to the form and interaction of the components in these systems. Here, we examine the interaction between the control of sound synthesis and the description of sonic entities in the contexts of domain-specific programming languages, computer-aided composition, and certain object-oriented implementations with a direct connection to the conceptual model of a sound object.

2.2 Conceptual Models

In this section, we approach the analysis of sound from conceptual models founded in qualitative description. In contrast to the perspectives and techniques presented in subsequent sections, the qualitative analysis of sound is oriented towards a symbolic interpretation. That is, it begins with a description of perceptually significant entities in a sound. In a sense, this analytical perspective can be characterized as a generalization of a score-based representation of music to sound.

2.2.1 Sound Objects

In the introduction to this chapter, we outlined two opposing interpretations regarding the representation of sound. Rather than consider signal and symbol representations as being indicative of a rigid dichotomy, here we characterize them as tendencies towards opposing ends of a spectrum. Naturally, this characterization is motivated by the notion of representations that fall between these poles. So, conceptually speaking, what is the basis for such intermediate models?

In [57], Pierre Schaeffer develops a taxonomy for the description of sounds built upon the notion of a *sound object*. The sound object can be considered a generalization of a

traditional note in the context of *musique concrête*, as it ascribes an identity to elements of sound with internal coherence that retain their character upon multiple auditions [11]. According to Schaeffer, these objects are determined by an intentional process of *reduced listening*, which operates without making reference to the supposed source or meaning of a sound, thus distancing it from a purely symbolic interpretation. Furthermore, though correlated with the physical signal, a sound object is considered as a separate and interpreted entity without an objective formulation.

The taxonomy described by Schaeffer is built upon *typo-morphological* description. From this perspective, a sound object retains an aspect of a symbolic representation, i.e. persistent typological identity, and sound is comprised of a heterogeneous collection of such elements [48]. On the other hand, the morphological facet of this descriptive approach is more analogous to signal-based representation. This aspect is concerned with a description of sound objects in terms of features that evolve over time, though the information represented in this case is of a psychoacoustic (rather than acoustic) nature. In the remainder of this section, we present two cases where typo-morphological description has been developed further.

2.2.2 Spectromorphology

The issue of time-dependent evolution was developed further by Smalley in his theory of sound representation based upon the application of morphological criteria to spectra, i.e. *Spectromorphology* [63]. It can be considered a furthering of Schaeffer's qualitative mechanism by means of perceptually motivated spectral descriptors, that is a vocabulary to describe the *shape* of a sound object [64].

The motivation behind this method was to further qualify the representation and listening experience of sounds without clear causal identities. In a practical setting, this methodology has been applied to musical analysis and 'score-like' visualization of electroacoustic music [70].

2.2.3 Microsound

Where spectromorphology is primarily concerned with the change of certain interpreted features over time, *microsound* is a conceptual model that considers the constituency of sound in terms of its elementary materials [48]. Tending towards a signal interpretation,

these elements exist at the limits of perception and microsound describes a time scale beneath a note but greater than the sample-level associated with a signal. In this way, it can be considered as a typology oriented towards the signal interpretation of sound. That is, where the spectromorphological perspective associates a signal-like morphology to a semblance of a symbolic identity, microsound associates a symbol to elements that approach the signal level using a taxonomy of *grains*.

However, it is noted that this examination was typically from the perspective of construction, as microsound arose from a particular creative aesthetic in the works of Xenakis [77] as well as the *granular synthesis* of Truax [71], Roads [47], and others [69]. This is to say that the granular approach to sound assembly and synthesis predated an invertible modeling technique[1] and, as such, can be considered a conceptual model stemming primarily from the perspective of synthesis.

2.3 *Analysis/Synthesis* and Intermediate Representations of Audio

In this section, we move from conceptual models of sound objects to the description of separable entities using the quantitative methods of the *analysis/synthesis* paradigm. These *intermediate representations* constitute a precise description of sound and, in contrast to the qualitative description of the previous section, add to the formulation of their analytical techniques the constraint of *invertibility*[2].

In this domain, the task of finding an intermediate representation is a problem of reconciling opposing principles towards a signal interpretation. That is, in order to satisfy the constraint of invertibility, it must be possible to infer a sample-level representation from a model. On the other hand, an intermediate representation consisting of considerably fewer components facilitates processing, e.g. by means of a targeted approach.

[1] It has been suggested, e.g. in [67], that dictionary-based analysis methods provide such a counterpart. In Chapter 3, these are presented in detail.

[2] As such, we limit our discussion of quantitative methods to those whose analyses enable reconstruction and set aside the wealth of descriptors used for classification in music information retrieval [30].

2.3.1 Low-Level Intermediate Representations

Though not directly concerned with the modeling of sound objects in the sense of Schaeffer, certain tools from the domain of signal processing provide a generic classification of sound phenomena. We characterize these as *low-level* intermediate representations, as they make little reference to symbolic entities and require additional interpretation for such association.

Though this may be considered a consequence of the analytical techniques involved rather than their goal, in the context of the current discussion, Spectral Modeling Synthesis (SMS) [62] is notable for the separation of a given signal into classes of phenomena. Originating from the sinusoidal representation of McAulay and Quatieri [38], SMS treats a signal as the combination of a slowly-varying deterministic component and stochastic component characterized by a time-varying filter applied to white noise [61]. Later developments included an improved method for obtaining cogent sinusoidal partials [16] and the addition of model for transient part [73].

Where SMS and extensions provide a model of sound object classes based on spectral properties, *sound segmentation* is concerned with the classification of signal excerpts based on their time-domain properties, e.g. as in [20] or [51].

2.3.2 Mid-Level Intermediate Representations

In [19], Rosenthal and Ellis develop the concept of a *mid-level representation* of audio and define a set of attributes that such a representation should have. Here, we review these attributes and relate them to the conceptual models of the previous section.

- **Sound source separation :** In contrast to the concept of reduced listening as articulated by Schaeffer, here sound source identity is considered a fundamental consideration of a signal representation. This is perhaps unsurprising as the study approaches the topic form the perspective of *computational auditory scene analysis*, which is an effort to apply the principles of the human auditory system, as described by Bregman [7], to machine listening.

- **Invertibility :** This attribute is characterized by the ability to synthesize a result from the representation that is at least similar, and ideally perceptually equivalent, to

the original. Here, the most important aspect is the separate synthesis of meaningful parts.

- **Component reduction :** Rather than consider each value of a signal (or its time-frequency representation) in isolation, a mid-level representation should reduce the number of components even if only by a grouping of elements such that its overall size remains unchanged. Both this feature and invertibility are congruent with the division of a sound into heterogeneous objects as specified by the conceptual models presented in the previous section.

- **Abstract salience of attributes :** This attribute pertains to the parameters of the representation and suggests that they should be indicative of their intended physical characteristics in terms of perception.

- **Physiological plausibility :** This principle states that, ideally, a mid-level representation should approximate the auditory system. Or, at least, that it should not contradict it. Though perhaps not explicitly stated as such, these two latter features allude to an aspect of qualitative description. Specifically, that a component of the representation should correspond to a perceptually valid and meaningful element in the signal.

Through an examination of these criteria, we have demonstrated that in many cases they can be related to aspects of the conceptual models presented in Sec. 2.2, where a noted exception is the significance given to sound source separation in the criteria of Ellis and Rosenthal. However, there is also a divergence from the conceptual perspective exhibited by the clear desire to express mid-level representations quantitatively (including their synthesis). Further, though concerned with a representation of sound that is more oriented towards a symbolic interpretation than prior analysis/synthesis techniques, here the construction of a mid-level representation was approached in a *bottom-up* manner. In fact, the remainder of the study is an evaluation of certain signal-oriented, quantitative sound representations (e.g. the Fourier transform, sinusoidal tracks, and the constant Q-transform) according to related criteria as well as the development of an alternate model called a *Weft*. That is, entities approaching the symbolic level are inferred from successive application of signal processing tools. Nevertheless, we shall see in Chapter 3 that the crite-

ria associated with a mid-level representation are featured prominently in the formulation of musically-oriented dictionary-based methods.

2.4 Data and Computational Paradigms

In this section, we address the representation of separable sound entities from the perspective of data organization, that is the building blocks used to generate sound. In our generalized view, we deem this an *analysis-by-construction* approach to sound. Here, we focus on three different contexts: languages for sound synthesis, computer-aided composition, and specific object-oriented data models notable for their abstract characterization of sound objects. As in previous sections, we highlight representational issues in the given context of the recurring theme of symbol/signal archetypes.

2.4.1 Data Representation in Sound Synthesis Languages

Superficially, sound synthesis languages tend towards a signal interpretation and sequences of symbolic musical events are generally characterized as a slowly evolving, relatively narrow-bandwidth signals, e.g. MIDI. A further examination of the predominant data model yields a familiar conceptual framework.

At the core of the MUSIC-N family of sound synthesis languages is a kind of low-level typo-morphological design principle. In a sense analogous to modular analog synthesizers, representation of sound in this context is typically framed as a high-level control of hidden processes essentially without an intermediate representation. This facet is evidenced by the division between *unit generators* [37] and the evolution of their control parameters over time. For example, this division is explicit in the syntax of Csound, with its definition of an *orchestra* (i.e. the specification of unit generators and their interaction) and a *score* (i.e. the control of the evolution of synthesis parameters). In fact, these two specifications are even supplied in separate file types [5].

Though we have provided an example relative to a widely used, more modern descendant of the MUSIC-N family, this particular typo-morphological design pattern is also evident in other implementations, such as the MPEG-4 Structured Audio Standard [58]. More generally, it is typical feature of modular (or *dataflow* oriented) sound synthesis languages whether they are text-based, e.g. Supercollider [39], or graphically-based, e.g. Max/MSP [45] [6].

To further demonstrate this organizational division, we mention that these disparate data features often operate at separate temporal rates, where unit generators function at the *audio rate* and the morphological control layer operates at a slower *control rate*. A notable exception is the ChucK language, which features a sample-synchronous, concurrent programming model where data elements operate in accordance with a single temporal reference [74].

Though the prevalence of this approach to data organization speaks to its ability to articulate certain sound synthesis routines, a considerable shortcoming associated with this approach is the representation of analysis/synthesis procedures. As regards sound synthesis languages, analysis and synthesis stages are often coupled, and a small number of established processes are offered in the form of integrated tools [75]. An exception to this tendency is demonstrated by more recent developments in the ChucK language. These have introduced a *unit analyzer*, i.e. an analysis building block analogous to the unit generator described above [75]. This feature is significant in that it offers direct access to analysis parameters and results (as they develop over time), thus permitting the construction of intermediate representations. Aside from this relatively recent approach, we notice that the typical coupling of analysis and synthesis stages results in a reduced modularity in the application of this technique.

2.4.2 Representation of Sound in Computer-Aided Composition

In contrast to the signal-oriented representation of sound synthesis languages, in environments for Computer Aided Composition (CAC) such as OpenMusic [1], the representation of sound objects tends towards a symbolic interpretation. Audio waveforms and intermediate representations built from audio (e.g. sound segmentations) are characterized as note-like objects and are treated according to compositional purposes [8]. In this context, sound synthesis is subject to a similar high-level of control with its parameters being subject to symbolic procedures [65].

2.4.3 (Sound) Object-Oriented Data Models

In this subsection we present Object Oriented (OO) models in the context of computer music. Here, the emphasis is on an OO data interpretation of sound objects. That is, other systems employ an OO software design strategy (e.g. Supercollider, ChucK, and

OpenMusic), however, here we address those that exhibit a direct relationship to the sound object of Schaeffer, as it is the application of the conceptual model to a programming paradigm that is of interest rather than the paradigm itself.

The defining data structure in OO programming expresses a connection between state (*attributes*) and functionality (*methods*). Further considerations specifically relevant to sound, are the management of time-evolution in an OO context as well as a coherent framework for multiple levels of sonic organization.

Referring to the latter of these aspects, aside from the difficulties associated with analysis/synthesis discussed in Sec. 2.4.1, a particular disadvantage of the typo-morphological setting provided in the MUSIC-N family of synthesis languages (and its associates) is with regards to the organization of higher-level musical sound structures such as phrases, sections, or other combinations of elements [56]. Towards this end, several projects have constructed systems based on an OO data model in conjunction with the conceptual model of a sound object, as this model makes no formal distinction between levels of sonic organization provided that they elicit a qualitative description, i.e. a component of an isolated instrumental sound is a sound object as is the recording of an entire piece of music.

FORMES [50] was a project concerned with sound synthesis in the context of musical composition. The most important aspect of FORMES to our current discussion is that of a *process*. A FORMES process is a named entity that groups together procedural rules, a scheduler, a set of local environmental variables, and subprocesses. Further, a process is subject to a precise duration. Together the properties of a process provide a hierarchical, structured approach to control of sound synthesis in the form of a precise morphological routines. These routines can provide a kind of internal coherence to the evolution of control parameters and consequently the resultant sound, i.e. the synthesis of a kind of sound object.

Kyma [56] is a system for sound manipulation that makes an explicit link to the conceptual model of sound object. In Kyma, these objects are defined as an *Atom*, (e.g. a stream of samples), a unary transform $T(s)$ of a sound object s, or an N-ary transfrom $T(s_0, s_1, \cdots, s_m)$ of multiple sound objects [54]. As in FORMES, a finite duration is associated with these fundamental objects. With this definition, it is evident that a sound object in Kyma can consist of an accumulation of sound objects, thus facilitating higher-level musical structures and subsequent grouped processing. In a sense, Kyma represents a kind of programmable audio workstation built on OO principles [55].

The CLAM framework is more recent application of OO principles to the concept of a sound object [3]. However, here the conceptual model is extended to subsume essentially all entities relevant to musical audio discourse. For example, a given audio track is an object, as is a musical note, or an instrument, or any other element that can be associated with a state and behaviour. One goal of this framework is to provide a user with pluggable objects in the context of analysis/synthesis and effects processing [2]. In this sense it is reminiscent of a proposed feature of extended SuperVP [17].

In this thesis, we are concerned with the building blocks of an analysis/synthesis system, that is a method of interacting with sound analysis data built on OO principles and the objects available in such a system. In this regard, CLAM is of particular interest as it includes an analysis stage, i.e. a *sound object identification* procedure. Though founded on an inclusive conceptual model and implemented according to OO principles, in consideration of what we believe to be a desirable formulation of the problem of music representation, we notice a shortcoming as regards the identification of sound objects in the system. Specifically, descriptive methods consider only *low level* or *high level* features. That is, those closely related to a pure signal interpretation, such as spectral descriptors, or those more analogous to sound classification. That is, a mid-level representation is not provided.

2.5 Conclusion

In this chapter, we have identified a dilemma with respect to the representation of sound, namely the reconciliation of physical and symbolic interpretations. Further, we have made reference to three general perspectives that address this issue with their respective analytical mechanisms. In comparing these methods, we have observed varying tendencies towards the symbolic or signal interpretation. However, in this observation, we have noted a lack of tools that fully integrate a mid-level, invertible representation into a modular and malleable software framework. That is, while tools exist to perform analysis/synthesis in a symbolic framework (for instance, the integration of SuperVP processes in OpenMusic [9]), these constitute an interaction between signal and symbolic aspects rather than a framework making explicit use of mid-level representations.

Chapter 3

Dictionary-Based Methods and Musical Audio

3.1 Introduction

Dictionary-based methods (DBMs) are a class of adaptive analysis techniques used to obtain a parametric representation of a signal according to a set of elementary functions, or a *dictionary* consisting of *atoms*. Highlighting two important features from this definition, a DBM is specified by the properties of its analysis set and one or more rules by which to combine its elements in order to approximate the target signal.

In Sec. 3.2, we address the theoretical aspects of DBMs and present them as a generalization of conventional signal representation by means of waveform bases. Further, we show how a set of analysis functions without the constraint of mutual independence can permit a sparse representation of a signal where a majority of its energy is accounted for by a small number of coefficients. However, a consequence of this generalized setting is the absence of a unique signal representation stemming from the redundant nature of the analysis set. This feature has led to the development of criteria related to the form of a desirable model.

The first criterion alludes to the strong relationship between DBMs and audio coding, where the ideal representation has been considered to be the model that is most sparse, i.e. consists of the least number of elements from the analysis dictionary [52].

On the other hand, researchers in the area of musical signal analysis have been motivated

by the flexibility afforded by DBMs to formulate the analysis process in consideration of expected (and targeted) features of the signal in question. These efforts represent the adoption of another criteria pertaining to the structured nature or 'meaningfulness' of the representation. This is taken to mean that the salient components of a signal are well-represented in the model.

Ideally, these two criteria would be equivalent. That is, the optimal representation in terms of sparsity would also be the most meaningful. However, the multifaceted nature of musical phenomena is such that determining what constitutes a signal 'component' and what makes it 'well-represented' in this context eludes a single objective formulation. As such, strategies concerned with different aspects of the analysis process have been developed to account for sound objects expected in a musical signal.

A typical dictionary-based signal analysis procedure can be expressed as a sequence of three stages. The first stage is concerned with the properties of the atoms contained within the dictionary. The second stage is concerned with how a decomposition takes place, i.e. the process by which atoms are selected and combined. In an optional third stage, the contents of the synthesis *book*, i.e. the model, can be grouped according to some structural metric. Throughout the literature, each of these stages has been re-formulated to incorporate knowledge about musical audio, e.g. [34][28][15][68]. Following the presentation of the theoretical foundation of DBMs, each of the stages discussed above is addressed in detail. In the sections that follow we restrict our presentation to discrete-time as we are primarily concerned with digital audio signals.

3.2 Foundations of Dictionary-Based Methods

In this section we provide an overview of the theoretical aspects central to dictionary-based methods. We first present time-frequency uncertainty as the motivation for waveform representations using overcomplete sets, which are presented in the second section.

3.2.1 Time-Frequency Uncertainty

The analytical foundation for an atomic representation of sound was developed by Gabor [23]. Dissatisfied with the counter-intuitive description of a signal in terms of either time or frequency, Gabor sought to express a signal as a sum of elementary functions localized in both domains, that is a *time-frequency* representation. Here, a result was the formal

expression of the reciprocal relationship between time and frequency. Stated concisely, a coordinate in the time-frequency plane can not be determined with arbitrary precision simultaneously in both domains. That is, increased specification in time results in increased ambiguity in frequency and vice versa. Using the notation of Heisenberg's uncertainty principle, this duality can be stated as

$$\Delta t \Delta f \geq 1 \tag{3.1}$$

where Δt and Δf represent the effective duration and bandwidth of the time-frequency quadrant, respectively.

In order to facilitate his time-frequency signal representation, Gabor developed a parametric function that satisfies the lower bound of the uncertainty relation defined in Eq. (3.1). This function, referred to as a *quantum of information* by Gabor, takes the form of a complex sinusoid whose amplitude is modulated by a Gaussian distribution, i.e.

$$g\left(t\right) = e^{-\alpha^2(t-u)} e^{2\pi j \omega t} \tag{3.2}$$

where t, u, ω, and α represent time, translation, frequency, and variance. These *Gabor atoms* have been used extensively in the literature for both general and domain-specific DBMs (see Sec. 3.3).

3.2.2 Waveform Representations

The definition of DBMs given at the beginning of this chapter is reminiscent of more conventional signal representations by means of waveform bases, such as those of the time or frequency domain. In broader terms, waveform bases and dictionaries are collections that both belong to a general class of time-frequency analysis tools, referred to as *waveform representations*. Here, we outline the basic properties of these constructs and leave details regarding specific collections for music signal analysis to the subsequent section. In the following, we limit our discussion to the finite-dimensional Hilbert spaces $\mathbb{R}^N$ and $\mathbb{C}^N$, which we denote by $\mathbb{H}$.

A waveform representation of a signal is a model built from a linear combination of elementary functions, or *atoms*. Among this class of techniques, waveform bases find the most straightforward expression and can be considered the foundation from which further

techniques are developed.

The defining properties of waveform bases are completeness and orthogonality. Considering the first property, a set of waveforms $\Phi = \{\varphi_i\}_{i \in I}$ is complete with regards to $\mathbb{H}$ if, for any signal $x \in \mathbb{H}$, there exists an expansion in the form of

$$x = \sum_{i \in I} \alpha_i \varphi_i \tag{3.3}$$

where α_i represents the ith expansion coefficient, obtained by

$$\alpha_i = \langle x, \varphi_i \rangle \tag{3.4}$$

and the set I is an index over Φ. The property of orthogonality states that $\langle \varphi_i, \varphi_j \rangle = 1$ if $i = j$ and 0 otherwise.

As we are primarily concerned with the properties of the set $\mathbf{\Phi}$, we reconsider it as a matrix of size N^2 with the vectors φ_i as its rows and $\boldsymbol{x}$ as a column vector of length N. As such, we write the equivalence relation in matrix-vector form as

$$\boldsymbol{x} = \mathbf{\Phi}^\dagger \alpha \tag{3.5}$$

where $\dagger$ denotes Hermitian conjugation and

$$\alpha = \mathbf{\Phi} \boldsymbol{x} \tag{3.6}$$

Expressed as such, Eq. (3.6) is the expansion of a signal into a basis and Eq. (3.5) is the inverse operation.

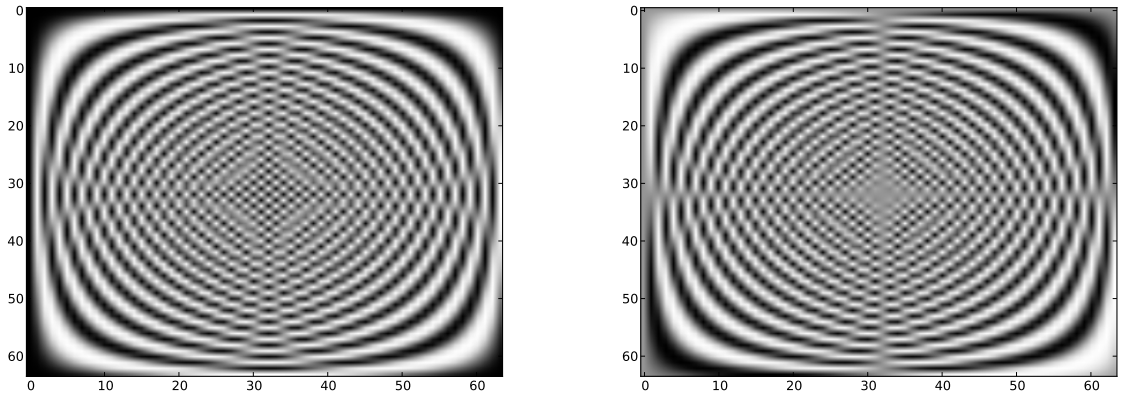
Following from the definition of a basis presented above, we observe that the set of time-domain basis functions built from a sequence of Dirac impulses is certainly the simplest example of a matrix $\mathbf{\Phi}$. In fact, it is merely the identity matrix of the N-dimensional vector space where all entries are zero except along the diagonal, i.e.

$$\mathbf{\Phi}_D = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{3.7}$$

In the context of time-frequency analysis, the other fundamental set of waveform bases are those that define the discrete Fourier transform (DFT). Though often depicted as a discrete sum, as in [42], here we focus on the expression of the DFT in matrix form [60]

$$\mathbf{\Phi}_F = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & z & z^2 & z^3 & \cdots & z^{N-1} \\ 1 & z^2 & z^4 & z^6 & \cdots & z^{2(N-1)} \\ 1 & z^3 & z^6 & z^9 & \cdots & z^{3(N-1)} \\ 1 & \vdots & \vdots & \vdots & & \vdots \\ 1 & z^{N-1} & z^{2(N-1)} & z^{3(N-1)} & \cdots & z^{(N-1)(N-1)} \end{bmatrix} \tag{3.8}$$

where $z = e^{\frac{2\pi j}{N}}$ and $\frac{1}{\sqrt{N}}$ is a scalar that ensures that the transform is orthonormal, i.e. that it preserves the inner product. These basis functions are shown in Fig. 3.1, for $N = 64$.



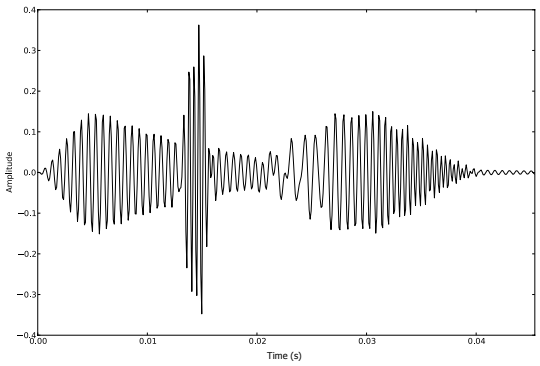
(a) Real part (b) Imaginary part

Fig. 3.1 DFT basis functions

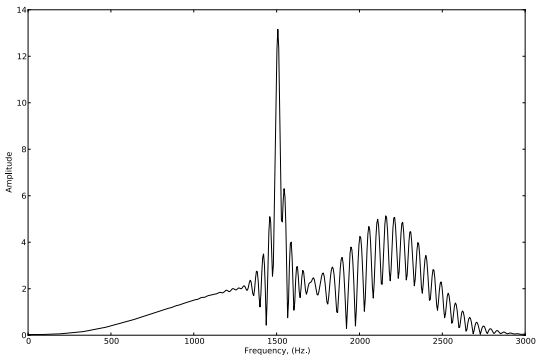
Waveform bases exhibit the attractive qualities of completeness and uniqueness, i.e. they are able to represent any signal in the vector space over which they form a basis in exactly one way. However, in practice it is possible (and indeed probable) that this representation will rely on a large number of elements, which is to say that the energy of the signal will be distributed over many coefficients. This feature is a consequence of the homogeneous nature of basis functions, which are generally obtained by strict translation, modulation, or dilation operations upon a single characteristic wave shape. As such, waveform bases

tend to represent signals consisting of heterogeneous components in a non-sparse manner, making use of the interaction between many elements of the analysis set. In terms of meaningfulness, this facet implies that the resultant representation may be difficult to interpret or even misleading with respect to underlying components in the signal.

As an illustration, consider a synthetic discrete-time signal x whose time-domain waveform is shown in Fig. 3.2a . This signal consists of three windowed sinusoids. The first is a formant-wave function (FOF) [49] with a duration of 93 milliseconds and a centre frequency of 1510.425 Hz. The second is a truncated Gaussian window with a duration of 5 milliseconds applied to a stationary sinusoid with a frequency of 2205 Hz. The third is a Hann window [29] with a duration of 23 milliseconds applied to a linearly-chirped sinusoid, which begins at 88.2 Hz and increases by 2976.75 Hz over its duration. The Fourier transform, or frequency-domain representation, of x is given in Fig. 3.2b. Though both of these representations are perfect in the sense that they exhibit no loss of information, the features of the underlying components are only well-depicted relative to their respective domains. In particular, the 'timeless' representation afforded by the Fourier transform has smeared the change in frequency of the chirped sinusoid across many coefficients[1]



(a) Time-domain representation



(b) Frequency-domain representation by DFT bases (real part)

Fig. 3.2 A synthetic signal x as represented by common waveform bases

A principal feature of DBMs that differentiates them from waveform bases is the relaxation of the constraint of orthogonality. Practically speaking, this can be thought of as a

[1] As mentioned by Gabor in [22], a 'change in frequency' is not defined in terms of the Fourier transform as it is a statement of both time and frequency.

consequence of the desire to introduce heterogeneous analysis functions that may be more well-suited to describing the underlying components of a signal. However, these functions may not be related to each other by a simple, orthogonality-preserving operation. As we will see, the lack of orthogonality renders the analysis expression more complicated. On the other hand, a well-chosen dictionary and decomposition strategy can facilitate a more sparse and/or meaningful representation.

If we begin by abandoning the constraints of orthogonality and homogeneity, we may turn to the issue of what properties an analysis set should have. In many situations, it could be advantageous to retain some of the formal properties of a basis, especially as regards completeness. In this context, the concept of a *frame* of a vector space is introduced.

A frame is a mathematical construct that generalizes a basis to sets of non-orthogonal functions that span a vector space [31]. If $\Phi = \{\varphi_i\}_{i \in I}$ is subset of $\mathbb{H}$, it is considered a frame if there exists two constants A, B such that $0 < A \leq B < \infty$ and

$$A||x||^2 \leq \sum_{i \in I} |\langle x, \varphi_i \rangle|^2 \leq B||x||^2 \tag{3.9}$$

for all $x \in \mathbb{H}$. Simply put, this *frame condition* implies that there is no $x \in \mathbb{H}$ that is orthogonal to all elements in Φ (the lower bound) and that sum has finite energy (the upper bound).

If the elements of Φ are linearly dependent and span $\mathbb{H}$, it is evident that Φ contains more items than are strictly required to represent a given signal. In this case, the set is said to be *overcomplete* and there may be an infinite number of signal representations built from a linear combination of its elements. The loss of a unique representation is in a sense the cost of using a redundant analysis set, which may be more suited to sparse/meaningful representations of a signal.

Here, we present frames in order to illustrate some of the benefits of redundancy in terms of the representation of signals. This is accomplished by demonstrating the relationship between frames and the discrete setting of the Short-Time Fourier Transform (STFT) [44].

As we have illustrated in Figs. 3.2a and 3.2b, a representation of a signal in the time or frequency domain can constitute a counter-intuitive description of the underlying components in a signal. The essential property of the discrete STFT is the localization of the DFT bases in time in order to describe spectral content or change.

Consider a finite-dimensional vector in $\mathbb{R}^N$ as the discretized signal $x[n]$. The STFT of

$x[n]$ can be expressed as

$$X\left[k,m\right]=\sum_{n=-\infty}^{\infty}x\left[n\right]w\left[n-m\right]e^{-j\frac{2\pi}{K}kn},\ k=[0,1,\cdots,K-1] \tag{3.10}$$

While it is a fundamental property of the DFT that the set $\left\{e^{-j\frac{2\pi}{K}kn}\right\}_{k=0\ldots K-1}$ constitutes a basis of $\mathbb{C}^K$, the STFT introduces constraints upon the time support of the analysis atoms[2] that must be satisfied in order to ensure reconstruction. Specifically, invertibility is also determined by the spacing, or *hop size*, of the analysis atoms as well as the properties of the analysis window $w[n]$. In order to satisfy this constraint, the upper bound on the hop size is determined by a window-specific feature such that the sum of evenly spaced windows is a constant [29], as in Fig. 3.3.

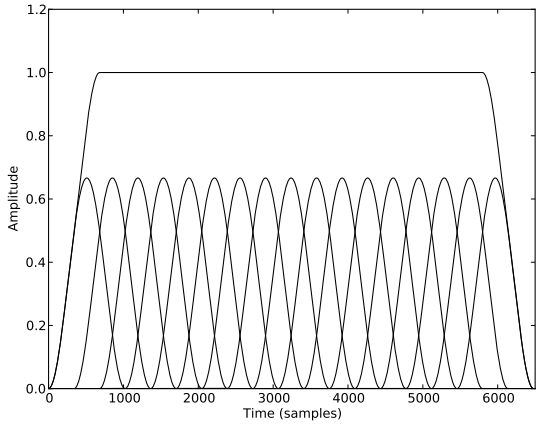
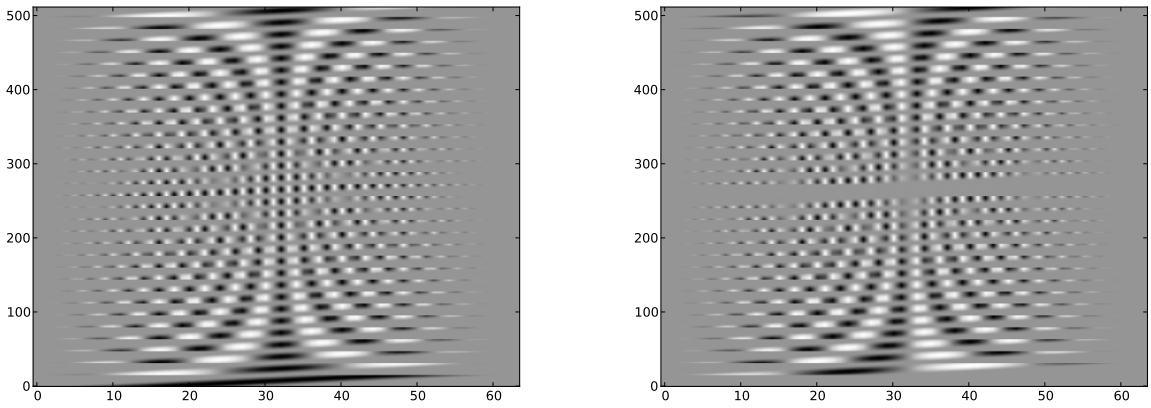


Fig. 3.3 Sum of overlapping Hann windows (length = 1024, hop size = 341)

Alternatively, the STFT can also be expressed in matrix form as is depicted in Fig. 3.4 where each row depicts a time and frequency localized sinusoid. Rephrased using the terminology of DBMs the minimum hop size is the limit that ensures the completeness of the collection, hence the possibility of reconstruction. A hop size smaller than the minimum increases redundancy in the analysis set and this overcomplete collection constitutes a frame

[2]In the literature, e.g. [4], it is common to encounter the term *grain* here in reference to the sample-wise product of window function and sinusoid. To emphasize the relationship, we favour the analogous terminology used in the context of DBMs.

of the space occupied by the signal. This property is evidenced by the rectangular shape of the STFT matrix.



(a) Real part (b) Imaginary part

Fig. 3.4 STFT Frame (N = 64, hop size = 2)

While bases and frames are both subject to strict mathematical definitions, a waveform *dictionary* is not specified with such precision. In [15], a dictionary is defined as a set of elementary functions D (typically more redundant than a tight frame) that spans some vector space $\mathbb{H}$ such that, for $x \in \mathbb{H}$, x can be expressed as a linear combination of elements in D. However, as we shall see in the context of music-specific techniques, even this relatively loose definition does not hold and D might be designed to be only complete for some subspace of $\mathbb{H}$. In the context of these application-specific techniques, it is perhaps more appropriate to relax this definition further and adopt a pragmatic view. Here, we consider a dictionary to be a collection of waveforms used to build some representation of x, be it oriented towards sparsity, meaningfulness in some setting, or some other criterion. Put another way, the specification of the contents of the dictionary is a parameter of the analysis procedure and the resultant representation is a function of this choice, which may be informed by knowledge about a class of signals. The *decomposition* of a signal according to DBMs can be characterized as a process of navigating the analysis set in order to find a desirable representation according to some guiding principle.

Returning to the signal x in Fig. 3.2a, it is evident that a set of analysis functions consisting of linearly-chirped Hann atoms, relatively brief Gabor atoms, and FOFs would

elicit a more concise representation than those of the time or frequency domain[3]. While we make no assurance that such a set is complete, in Fig. 3.5 we show the first three iterations of a decomposition of x using a collection of heterogeneous atoms and a Matching Pursuit algorithm, whose properties are discussed in detail in Sec. 3.4.1. In Fig. 3.6, we show a visualization of the time-frequency energy distribution of the model $\tilde{x}$ by means of a wivigram [67].

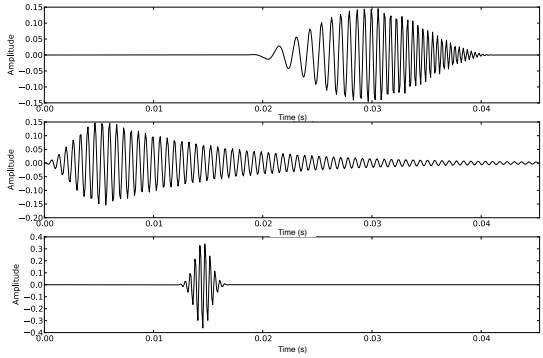


Fig. 3.5 First three atoms extracted from x

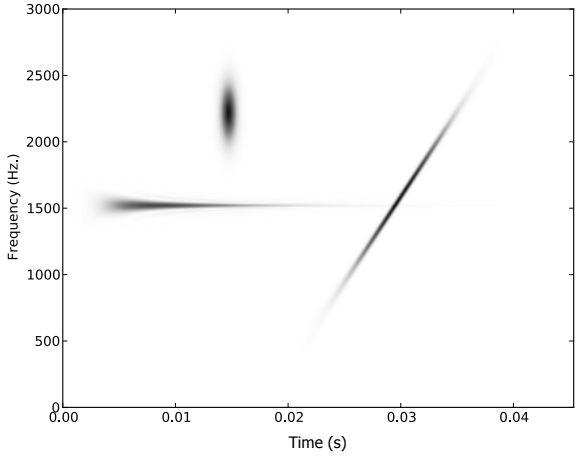


Fig. 3.6 Wivigram of the model $\tilde{x}$

[3]This is of course provided that the parameters of the underlying components in x are accounted for in the dictionary.

This example represents a significant simplification of the problem since, having created the synthetic signal, we can hazard a fairly omniscient guess about the properties of the elements in the analysis set. Of course, it is typically not possible to know *a priori* precisely what set of functions will yield a desirable representation. Furthermore, signals stemming from naturally occurring phenomena may not be so well-behaved with regards to the analytical expression of their underlying components. However, within a particular problem domain certain assumptions can aide the formulation of a DBM.

3.3 Dictionaries for the Analysis of Musical Audio

In this section we present elementary functions, and collections thereof, from the perspective of musical audio signal analysis. In this research area, assumptions based on expected features have been used in the selection of analysis functions for used in atomic decomposition. In the presentation that follows, we proceed from generally applicable analysis functions to those that incorporate specific information about instrument sources [4].

3.3.1 Generic Time-Frequency Atoms

In computer music applications, one of the most common *time-frequency representations* is the phase vocoder. While it is common to present the phase vocoder according to a filterbank or Fourier transform interpretation [18], it is equivalent to the STFT presented in the previous section. As such, the properties of the frame interpretation apply to all commonly encountered analysis window functions [29]. This set of windowed sinusoids can be considered the most often used collection of time-frequency atoms.

3.3.2 Atoms and Classes of Musical Signal Phenomena

Recalling the *sines + noise* paradigm [61], there has historically been a desire to express musical signals as being comprised of separable classes of phenomena. Though this is not structured in the sense of a mid-level representation, with regards to DBMs, a strategy has been to consider atoms with an analytical expression that approximates objects found in musical signals.

[4] Regarding variables, in the following s, t, u, ω are reserved for scale, time, translation, and frequency respectively. Additionally, the function g refers to the Gabor atom in Eq. (3.2)

The asymmetry of components found in musical signal has been approached in a number of ways. For example, Goodwin included damped sinusoids of the form [24]

$$d_{(\alpha,u,\omega)}(t) = e^{-\alpha(t-u)} e^{2\pi j\omega t} y(t-u) \tag{3.11}$$

where α corresponds to a damping coefficient and $y(t)$ is the Heaviside function, i.e.

$$y(t) = \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t \geq 0 \end{cases} \tag{3.12}$$

The *Formant-Wave-Function* (FOF), introduced in [49], can be used to provide more nuance to the description of asymmetrical signal phenomena. In essence, a FOF is a damped sinusoid where the initial discontinuity has been smoothed by a cosine shape of variable duration, i.e.

$$FOF_{(\alpha,\beta,\omega)}(t) = \begin{cases} 0, & \text{for } t \leq 0 \\ \frac{1}{2}\left(1 - \cos\left(\beta t\right)\right) e^{-\alpha t} \sin\left(2\pi\omega t\right), & \text{for } 0 \leq t \leq \frac{\pi}{\beta} \\ e^{-\alpha t} \sin\left(2\pi\omega t\right), & \text{for } t \geq \frac{\pi}{\beta} \end{cases} \tag{3.13}$$

Here, $\frac{\pi}{\beta}$ corresponds to the attack (or rise) time of the window and, in practice, the shape can be time-shifted as in Eq. 3.11. Though developed primarily for synthesis, this function has been employed in the context of atomic decomposition, e.g. in [28] where it was used for piano note-event detection.

In [35], Lewicki developed a model of the human auditory encoding system founded on an assumption of sparsity. Here, the shape of the optimal function was found to be variable in terms of the bandwidth associated perceptually-motivated sound classes. This function can be approximated by a gammatone of the form

$$f_{(\sigma,u,\beta,\omega)}(t) = (t-u)^{\sigma-1} e^{-2\pi\beta(t-u)} e^{2\pi j\omega t} y(t-u) \tag{3.14}$$

where β and σ represent order and bandwidth, respectively. These parameters tune the asymmetry of the time-frequency energy distribution.

To target signal components that change in frequency, a method using linearly-chirped

Gabor atoms of the form

$$g_{(s,u,\omega,c)}(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{j\left(\omega(t-u)+\frac{c}{2}(t-u)^2\right)} \tag{3.15}$$

was developed by Gribonval [27].

In order to account for the acoustic nature of many musical instruments, Gribonval introduced harmonic atoms of the form

$$h(t) = \sum_{k=1}^{K} \alpha_k g_{(s,u,\omega_k)}(t) \tag{3.16}$$

where $\omega_k \approx k\omega$ [28]. In practice, the relationship between components was strictly harmonic or specified by a function defined *a priori*.

3.3.3 Music-Specific Atoms

More recently, the method of harmonic atoms has been extended to instrument/context-specific atoms [34].

$$h_{(s,u,\omega,c,A,\Phi)}(t) = \sum_{m=1}^{M} a_m e^{j\phi_m} g_{(s,u,m\cdot\omega,mc_0)}(t) \tag{3.17}$$

where $A = \{a_m\}_{m=1\ldots M}$ the vector of partial amplitudes, $\Phi = \{\phi_m\}_{m=1\ldots M}$ the vector of partial phases. Here, A was obtained through a supervised learning procedure consisting of classical spectral peak estimation and vector quantization using a database of labeled exemplars. The elements of Φ were tuned during the decomposition.

Incorporating further contextual information, music scene-adaptive atoms were used in [10]. Here, the spectral characteristics relevant to each MIDI note were estimated from monotimbral recordings according to an unsupervised process for the purpose of note-event detection and automated transcription.

In [12], Cho and Kuo developed a formal expression of dictionaries comprised of source-specific subspaces. Working from an overcomplete set Φ that spans a Hilbert space $\mathbb{H}$, one can assume that only a subset of Φ is relevant to the consideration of audio signals in $\mathbb{H}$. In turn, only a subset of this subset is relevant to the analysis of particular instrument, that is

$$\Phi_{inst} \subset \Phi_{music} \subset \Phi_{audio} \subset \Phi \tag{3.18}$$

(see Fig. 3.7 for an illustration of this principle).

By conducting an atomic decomposition on a training database, one can obtain a relatively sparse description of the elements relevant to a particular analytical task. Furthermore, these can be grouped in order to exhibit structural characteristics, e.g. harmonic components.

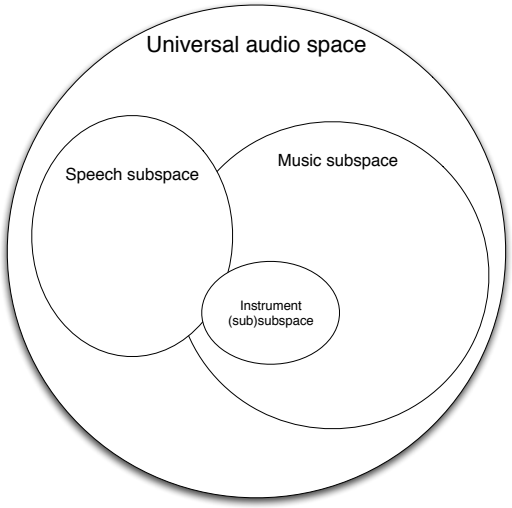


Fig. 3.7 Audio space and subspaces

3.4 Decomposition Algorithms

In this section we discuss decomposition strategies. As in the previous section we begin with generic decomposition strategies and progress to more musically-oriented techniques.

3.4.1 Generic Decomposition

Given an overcomplete set of elementary functions, finding the optimal signal representation using a finite subset represents a computationally intractable combinatorial maximization problem. As a result of this difficulty, the focus of algorithmic implementations has been the development of tractable sub-optimal solutions. The most popular of these, known as

Matching Pursuit (MP) [36], is an example of a *greedy* algorithm. As such, it follows the problem solving heuristic of making a locally optimal choice in the hopes of producing a globally optimal result [13]. In this case, the model is obtained via the iterative projection of the residual onto the maximally correlated atom in the dictionary.

MP has many desirable properties. For example, the convergence of the algorithm has been proven, its resulting model is typically sparse in comparison to the time-domain representation, and its typical usage is computable in a reasonable amount of time following [32]. Furthermore, the simple structure of the algorithm facilitates modification for particular analytical purposes. However, in its unmodified form, MP is essentially myopic, as it looks only to minimize the energy of the residual. As will be discussed at length in Chap. 5, there are salient components of musical signals that have relatively little energy (e.g. partials in the upper frequency range) and will be considered by MP only after many iterations.

3.4.2 Structured Decomposition

Though the convergence of MP was proven by Mallat and Zhang, the non-orthogonal nature of the dictionary contents implies that there is a complex interaction between atoms in the model. That is, the relationship between the components of a signal and the linear combination of a subset of atoms chosen from the dictionary may not be readily apparent.

Motivated to account for signal components that a human listener would classify as separable, several extensions to MP have been proposed in order to offset the greedy nature of the algorithm, which is ambivalent to intuitively structured elements.

In [15], Daudet used a variant of MP, referred to as *Molecular Matching Pursuit*(MMP) using a dictionary that consisted of the union of two sets of basis functions, namely modified discrete cosine and discrete wavelet bases. At each iteration of MMP, a cluster of atoms is classified as either tonal or transient and extracted.

Following the formulation of harmonic atoms, a variant of MP (HMP) was designed to obtain their specification over the course of a decomposition [28]. That is, where MMP makes an assumption about the continuation of tonal components in time this strategy expects harmonic components above the frequency of an atom.

Essentially as a combination of MMP and HMP, Meta Molecular Matching Pursuit was introduced in [33]. This procedure consists of two stages. First, a tonal component is identified, as in MMP. Second, a harmonic comb is fitted to the component. However,

in contrast to a typical application of MP, this technique is applied directly to a STFT matrix, i.e. with a single atomic scale/frequency resolution. Nevertheless, this technique was reported to have faired better than HMP in the analysis of signals that exhibit frequency modulation due to the chaining of atoms together to form tonal components. Though, in the analysis of piano sounds, components were missed as inharmonicity was not accounted for in the system.

3.5 Post-Processing

In contrast the techniques discussed above, domain knowledge has also been incorporated as a supplementary stage to a generic decomposition. Here, we outline two projects that featured this approach.

3.5.1 Agglomerative Clustering

In [68], Sturm describes a post-processing technique that clusters atoms according to a measure of similarity using as material a prior, generic decomposition. In this case, the metric employed was the complex correlation coefficient between each pair of atoms in the model. As in MMP, the targeted higher-order structures considered by this procedure were narrow band tonal components and broad band transient components.

3.5.2 Atom Prioritization

In a different application, MP decompositions on a database of instrument samples are used to obtain instrument-specific subspaces as a post-processing stage [12]. In this procedure, the contribution of each atom to the overall energy of the decomposition is tabulated and all atoms above a given threshold are considered members of a sub-dictionary to be used for the decomposition of signals featuring a particular instrument. A further step combined harmonically-related members of the sub-dictionary to yield instrument-specific atoms and these were used for musical signal separation.

3.6 Conclusion

In this chapter, we have presented a category of signal analysis tools. After reviewing the foundations of DBMs, we have presented the customization of generic DBMs to musical signal analysis. Here, assumptions about expected features of musical signals have been used to inform the choice of elementary functions used in analysis, structured decomposition, and post-processing.

Recalling the concept of a mid-level representation presented in Sec. 2.3.2, it is interesting to apply the associated criteria to DBMs. Of these, sound source separation, invertibility, and component reduction are principal motivating features behind the formulation of music-specific DBMs. Physiological plausibility could also be interpreted in the application of structured decomposition. However, with the exception of source-specific atoms, these structured techniques have been oriented towards the representation of signal objects, i.e. partials and transients, through *bottom-up* construction.

Chapter 4

pydbm

4.1 Introduction

In Ch. 2, we presented perspectives on the analysis, both qualitative and quantitative, and synthesis of (musical) sound objects as well their representation in data. In the examination of these perspectives, we observed a lack of structured intermediate representations in the context of the analysis/synthesis paradigm. In Ch. 3, we presented dictionary-based methods as a set of signal processing tools that exhibit many of the properties considered desirable for a mid-level representation. Furthermore, in the literature, these methods have been applied to the task of musically-oriented analysis and structured representation.

`pydbm` is a library, written in the python programming language, which we have developed to apply dictionary-based analysis/synthesis methods to the structured representation of musical signals. In particular, `pydbm` is an Object-Oriented (OO) framework designed to obtain structured models of audio and provide methods to visualize, synthesize, and otherwise manipulate these representations.

In the sections that follow we outline the architecture of `pydbm`, demonstrate the functionality of its core components, and present a collaborative project where it was featured in the context of computer-aided composition.

4.2 System Architecture

In comparison to structured DBMs, OO software design represents a similarly structured approach to data. Specifically, a fundamental aspect of OO programming is the union

of data and functionality in a single data structure, called an *object*. Another principal feature is *inheritance*, which is characterized by a hierarchical propagation of functionality amongst objects, typically progressing from a relatively abstract description to further specification. Perhaps due to the conceptual link between musical elements and this abstract hierarchical classification, OO software design has been employed frequently in musical applications [43]. In `pydbm`, we have applied this design philosophy to the development of a modular framework for the structured and invertible representation of musical audio.

4.2.1 Modules and Classes

In python, *modules* are collections of definitions, e.g. classes and/or functions , which can be imported into an environment recognized by the python interpreter. In this subsection, we present the modules of `pydbm`.

`atom`

In this module, we apply the concept of a unit generators (discussed in Sec. 2.4.1) to classes of atom generators, see Fig.4.1 for an illustration of the classes and inheritance relations found in the module. In `pydbm` we deal exclusively with time-frequency atoms and, as such, each generator inherits from the `window` and `sinusoid` base classes. The `window` class is particularly important in the context of signal analysis, as it features methods to calculate window-specific attributes described in [29], such as equivalent noise bandwidth and minimum overlap percentage.

As the union of attributes and methods, a generator class is equipped with a function that returns a parametrically specified atom in the form of a one-dimensional array. Due to the extent to which these functions are called during analysis procedures, they have been optimized and compiled as extensions in the C programming language using `cython` [72].

In `pydbm`, each atom genre is associated with specific multidimensional data types that are defined by the associated set of generating parameters and the context in which they appear. For example, a (chirped) FOF atom has generating parameters: frequency, phase, chirp rate, rise time and decay time. In the context of an analysis set, i.e. a dictionary, it also has an onset time and if part of a model, i.e. a book, it has a scaling coefficient. As regards class attributes, each generator contains the relevant metadata associated to its data types. In terms of structured analysis, the harmonic classes are for use in dictionaries

and books consisting of instrument-specific atoms. Here, the generating functions are the same, but the data type has additional fields pertaining identifying the properties of the an individual component in a structure.

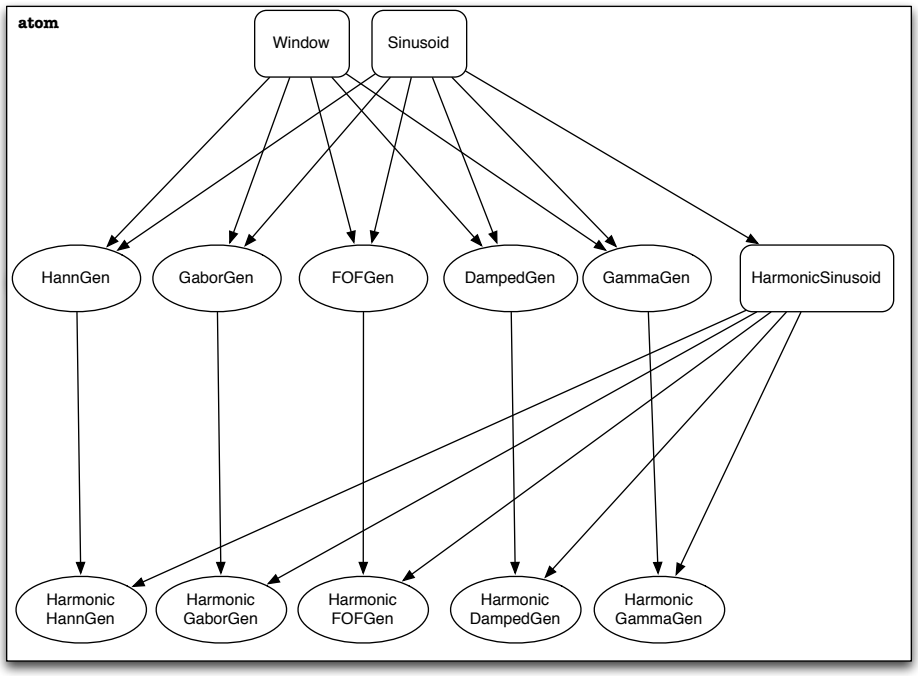


Fig. 4.1 Classes and inheritance in the `atom` module

In `pydbm` the `atom` module is self-contained, that is it does not inherit from objects in other modules. This is not the case of the remaining modules and in order to provide a system-wide context before presenting each of the remaining components in detail, we draw attention to Fig. 4.2, which illustrates the `pydbm` class tree.

`base`

The `base` module consists primarily of abstract base classes, which are not intended to be instantiated directly but contain some elementary functionality that is inherited across several modules. Here we briefly mention the key properties of the constituent classes: `Types`, `Group`, `Spectral`, and `IO`.

The `Types` class is essentially a meta-level description of the data types considered in `pydbm`. This includes the atom types discussed above as well as SDIF[76] types considered.

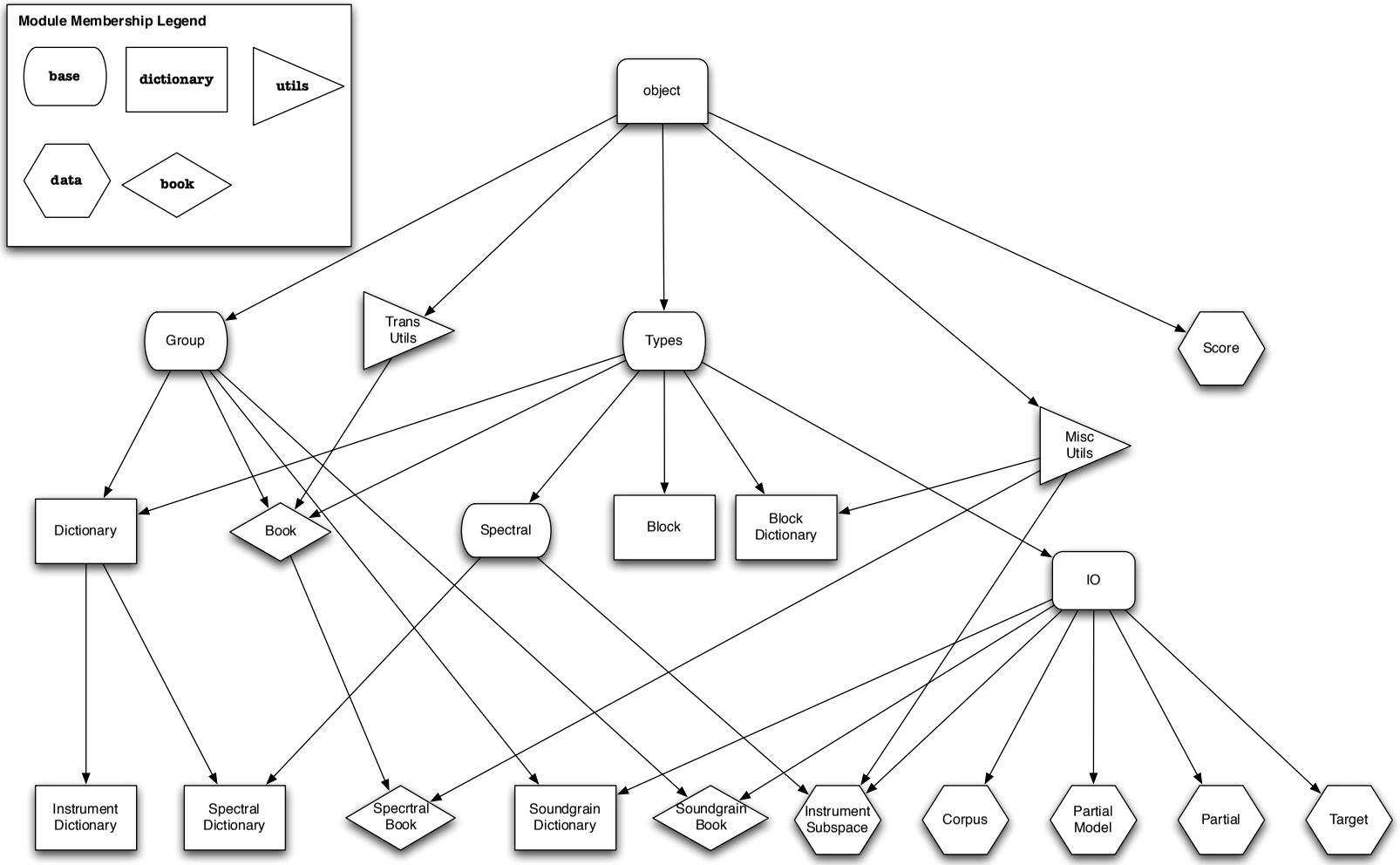


Fig. 4.2 Inheritance relations in `pydbm`

With regards to atom generator classes, type descriptions are stored in an associative way. That is, the name of a type, e.g. 'hann' in the case of a Hann generator, points to an object that has been instantiated along with the initialization of the parent class. This process is done only once, again avoiding overhead during computationally expensive analysis procedures. This associative organization also permits the use of dictionaries containing arbitrarily mixed atom types and is also important if one wanted to edit the contents of a model as regards atom type.

`Group` is an abstract base class inherited by any object that contains a set of atoms, that is any dictionary or book. It provides methods to count, partition, or parametrically alter a group of atoms. Also, it overloads the addition operator to provide an intuitive way to merge dictionary and book objects. For two dictionary/book objects $A + B = C$, the result being a new dictionary/book object whose atoms are the union of A and B. In this way we offer direct access to the data contained within a group in order to permit logical filtering, mixing operations, and the creation of subgroups.

`Spectral` is a base class with methods to perform conventional estimation and interpolation of spectral peaks. Additionally it also defines methods to compute spectral shape statistics, i.e. centroid, spread, skewness, and kurtosis. It inherits from `Types` to facilitate computation of window properties, e.g. coherent gain.

The `IO` base class is merely a generic wrapper for read/write capabilities in terms of audio, afforded by the scikits audiolab package [14] and SDIF, using python bindings developed by Moguillansky [40].

`data`

This module contains objects which describe some intermediate data structures that can be used in conjunction with the more central dictionary-based analysis/synthesis objects. In particular, they describe components using a different modality than the weighted linear combination that is at the foundation of atomic decomposition. In this sense, they are included to suggest components (from outside of the system) to be modeled as targets or act as unconventional elements incorporated in a modeling procedure.

In the context of the experiments presented in Sec. 5.3, and more generally the interaction between signal and symbolic representations of music explored in this thesis, the most important class in this module is the `Score`. This class has methods to parse and store a

MusicXML file in the associated hierarchical manner, which is depicted in Fig. 4.3.

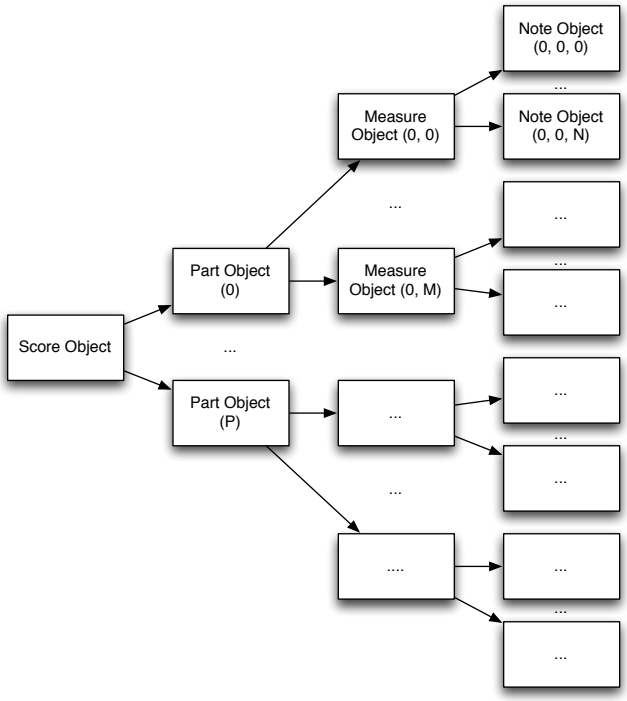


Fig. 4.3 Hierarchical organization of a `Score` object.

We have also included a `Corpus` class that can be used to reference a database of sound files and obtain information associated with them. This class is important to the project described in Sec. 4.3. `Target`, is a class that can be used in relation to the target of a particular analysis. In practice, it facilitates the storage of an audio waveform and associated metadata. A `PartialModel` class is also defined in this module. This class is initialized with an analysis in the form of a SDIF partial tracking file and converts each partial into a `Partial` object. In certain situations, a partial model can be useful as a guide for atomic decomposition, that is it points to regions of interest. Of course, this is only the case if an atomic decomposition can offer some benefit in terms of representation that a partial model can not, for example the broadband region often associated with the onset of a partial. Similarly, spurious partials could be indicative of a region in a signal more appropriately modeled by a multi-resolution DBM.

dictionary

In pydbm, a dictionary is an object that unifies the functionality and data associated with an analysis procedure. Here, *functionality* refers to the methods used to construct a set of atoms and the decomposition algorithm(s) associated with each class. On the other hand, *data* refers to the storage of analysis function parameters, here represented as a structured array, and the metadata associated with a given dictionary instance. The classes within the dictionary module have been designed differently in terms of these two features with varying degrees of specificity with regards to musical signals.

The Dictionary class is the most generic setting of a dictionary object in the sense that its functionality is not specifically tailored to the analysis of musical signals. As such, its constructive methods are rooted in a signal interpretation. For example, it contains methods to add atoms that constitute STFT frames and time-frequency regions. As far as higher-level identities are concerned, these are also based in signal structures. That is, methods are available to add atoms to tile a region associated with a partial or transient. Similarly, the constituent decomposition methods are of a general nature. Here, we have implemented a standard Matching Pursuit (MP) as well as a version of MP that features a gradient descent optimization of an arbitrary set of atom parameters.

The BlockDictionary class is similarly generic, in fact there are methods to convert a BlockDictionary into a Dictionary and vice versa. Here, the difference is with regards to the MP implementation where, in a BlockDictionary the computation of the inner products is accomplished in the frequency domain. In some situations, this can speed up computation, however this is not always the case since it typically involves a relatively large Fourier transform. Nevertheless, one significant improvement in this implementation is that it is computationally equivalent to check all time locations. That is, the decomposition is *time-shift invariant*.

The SpectralDictionary class is an example of a structured dictionary. Its decomposition strategy incorporates an assumption that elements occurring simultaneously are potentially related and extracts a group of atoms at each iteration. Here, one can further refine this structural relationship with an arbitrary degree of preference towards harmonicity[1]. In light of this assumption, it features a constructive method that adds atoms to target approximately harmonic components above a particular fundamental frequency over

[1]This variant of MP is the subject of Sec. 5.2.1.

a span of time.

With further specificity, the `InstrumentDictionary` class also contains constructive methods to target a particular fundamental frequency. However, in this case the set of atoms is itself structured, as the parameters are determined from a database of labeled exemplars. At each iteration, the source-specific MP extracts a set atoms whose frequencies and amplitudes are representative of the instrument in question.

Stemming from a different motivation, the `SoundgrainDictionary` considers a collection of sound files (found in an associated `Corpus object`) as atoms. At each iteration, its 'decomposition' strategy looks for the most suitable match between the residual and the members of the `Corpus`. The goal here is a particular aesthetic, rather than analysis.

`book`

Similar to a dictionary, the attributes of a book object describe a set of atoms. However, rather than being specified directly, these are the result of an associated analysis. In this way, a particular decomposition strategy can be considered a morphism that maps a dictionary object to a book object. Amongst members of this module, functions for synthesis, agglomeration, and visualization are the principal differentiating features.

A `Book` object is the product of a decomposition using a `Dictionary` or `BlockDictionary` object. In this case, synthesis is a straightforward process and includes options to add frequency modulation. In terms of clustering operations, a method is defined to compute the cross-correlation of the atoms in the book and agglomerate them into 'molecules' as described in [68]. This object also includes a method to visualize the model using a wivigram.

A `SpectralBook` object is the product of a decomposition using a `SpectralDictionary` or a `InstrumentDictionary`. It inherits from class `Book` but redefines its synthesis methods to facilitate frequency modulation appropriate to a quasi-harmonic structure. In addition to the cross-correlation operation, the `SpectralBook` class defines a routine to compute the spectral self-similarity of its structures and agglomerate these into molecules. In terms of visualization, it also adds a piano roll-like display that is the subject of Sec.5.3.1.

A `SoundgrainBook` object is the result of the `SoundgrainDictionary` matching procedure. In this case, there is no synthesis function, but rather a process of scaling and assembling appropriate sound files.

`utils`

This module defines two classes with more generic tools that are inherited by multiple classes.

The `MiscUtils` class includes miscellaneous tools, for example methods to perform conversions and make measurements of local energy. However, most important here are functions to quantify the effectiveness of a model. We have implemented measures of entropy and the perceptually motivated Itakura-Saito spectral distance, which will be discussed in more detail in Chap. 5.

As methods, the `TransUtils` class has common transforms such as the short-time Fourier transform, and the discrete-cosine transform. Most important in the context of DBMs is the implementation of the discrete Wigner-Ville distribution. The superposed distribution of atoms, sometimes referred to as a wivigram [67], is a common way to visualize a sparse approximation.

4.2.2 Illustration of Work Flow

In the code listing below, we illustrate the functionality of `pydbm` in a simple case. Here, we demonstrate the basic stages of a dictionary-based analysis/synthesis procedure, namely the definition of a dictionary, the application of a decomposition algorithm, and the optional post-processing of a synthesis book. In the listing, italicized text are comments that outline the steps of the procedure.

```
#import the necessary modules
import pydbm.dictionary
import pydbm.base

#useful python library of numeric tools
import numpy as np

#get audio
I = pydbm.base.IO()
x, fs = I.readAudio('/Users/grahamboyes/Desktop/test.wav')

#instantiate a Dictionary object
```

```
D = pydbm.dictionary.Dictionary(fs)

#specify three sets of STFT-like analysis functions
#with associated window-specific arguments where necessary
durations = [64, 256, 2048]
hops = [8, 32, 256]
atom_types = ['damped', 'gabor', 'FOF']
win_args = [{'damp' : 0.1}, {}, {'rise' : 32, 'decay' : 2016}]
min_time = 0
max_time = [len(x) - d for d in durations]

#iterate through the list and add atoms
for i, duration in enumerate(durations):
    D.addSTFT(atom_types[i],
              duration,
              hops[i],
              min_time,
              max_time[i],
              **win_args[i])

#perform analysis of signal x using a Matching Pursuit algorithm
#the MP algorithm returns signal vectors of the model and residual
#as well as a Book object
max_iterations = 10
SRR_thresh = 35.
model, residual, Book = D.MP(x,  max_iterations, SRR_thresh)

#partition and synthesize the model
#(if it contains gabor atoms)
if any(Book.atoms['type'] == 'gabor'):
    inds = np.where(Book.atoms['type'] == 'gabor')[0]
    B1, B2 = Book.partition(inds)
    #...apply some changes to the models here...
    Book = B1+B2
```

```
#synthesize the book and write audio
x_hat = Book.synthesize(synthtype='default')
I.writeAudio(x_hat, '/Users/grahamboyes/Desktop/test_hat.wav', fs)
```

In contrast to other dictionary-based software libraries, such as MPTK [32], in `pydbm` we sought to provide a high-level and malleable interface to DBMs. Here, we were especially interested in the design of a framework that could be integrated into environments for musical creation.

4.3 Application of `pydbm` in Computer-Aided Composition

In this section we discuss the application of `pydbm` to computer-aided composition in the context of a specific collaborative project with composer/researcher Marlon Schumacher around the composition of his piece *Ab-Tasten*. Its premiere was given as part of a live@CIRMMT concert at McGill University in April 2011. Binaural renderings of this performance are available at `http://www.music.mcgill.ca/~marlon/audio/Abtasten`.

4.3.1 OpenMusic and SDIF

To promote interactivity, dictionary definition and decomposition processes were bundled as command-line executables. These were then wrapped in a LISP interface by Marlon Schumacher in order to facilitate control in a compositional context. An example of the resulting environment is shown in 4.4. Here, we note the `gabor-params` and the `gabor-decomp` elements. These correspond, respectively, to the definition of a dictionary of Gabor atoms and a subsequent decomposition of a target signal (located in the top-left of the image).

The Sound Description Interchange Format (SDIF) [76] was adopted to facilitate OM access to the low-level `pydbm` features, e.g. the set of atom parameters associated with a dictionary or book. An OM interface to this data is shown in 4.5.

4.3.2 Corpus-Based Decomposition

Though DBMs are typically expressed in the formal setting of sparse and/or structured approximation, the flexibility of the components of the system also render DBMs suitable

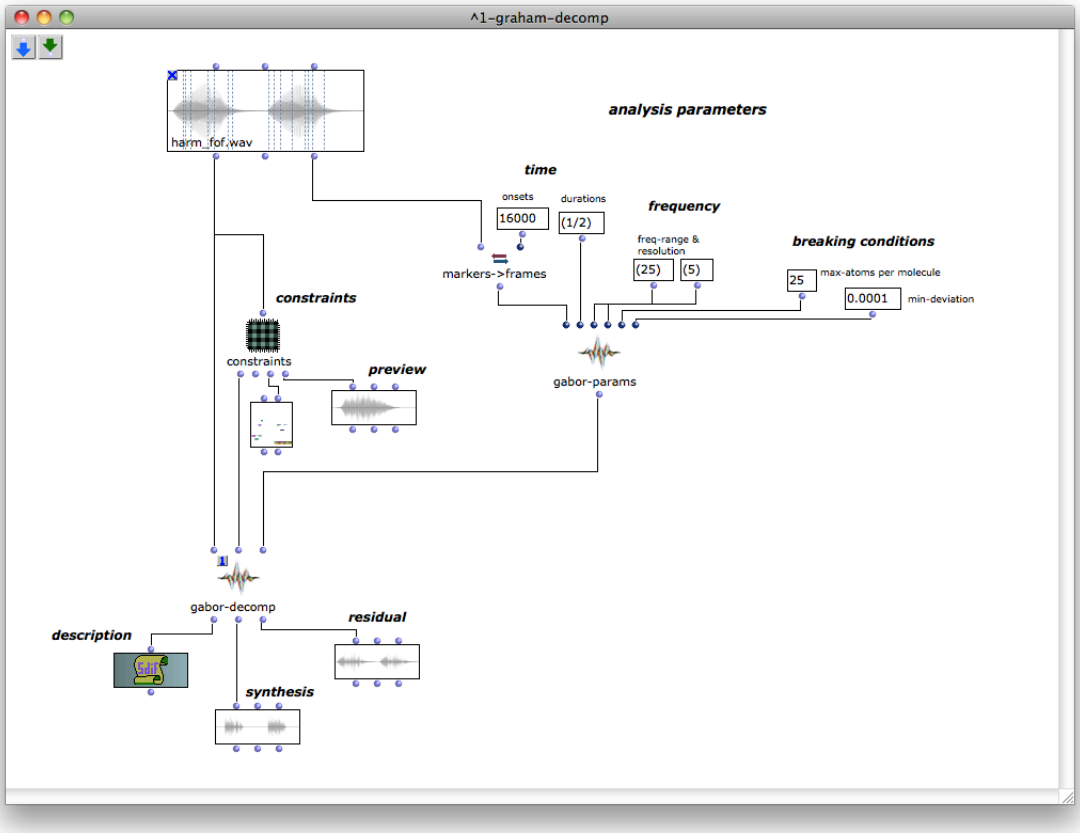


Fig. 4.4 `pydbm` components wrapped in OM (image provided by Marlon Schumacher)

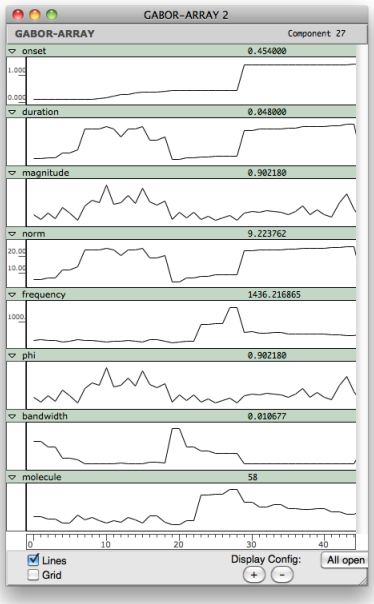


Fig. 4.5 Access to `book` parameters via SDIF (image provided by Marlon Schumacher)

for creative purposes. Here, we demonstrate this principle in their application to corpus-based synthesis [59].

A set of directories containing sound material can be used to define a `Corpus` object. This object is subsequently supplied to a `SoundgrainDictionary` along with a set of onsets. These samples are then used to approximate a sound target. In this sense, it is similar to the *adaptive concatenative synthesis* described in [66]. However, instead of an assembly process based on signal descriptors, here we use a variant of a Matching Pursuit algorithm, i.e. we apply the comparison at the signal level.

In contrast to sparse signal approximation, the goal of this technique is not to obtain a low-order model that is perceptually identical to the target, but is rather to assemble existing material in a way that retains features of both the corpus and the target. Here, the breaking condition assigned to the maximum number of pursuit iterations and to a certain extent the density of onset locations become synthesis control parameters, which can be set to yield a result closer to the target[2].

[2]Audio examples illustrating this principle are available at `http://mt.music.mcgill.ca/~boyesg/thesis_examples/4.3.2.html`. See Appendix A for details regarding the sound examples website.

In this collaborative project, this technique was used to suggest an approximation of a sound using piano tones. In turn, this model could be manipulated by the composer and used as control data for a Disklavier, which realized this 'score' in concert.

4.4 Conclusion

In this chapter, we have presented a software framework designed for dictionary-based analysis/synthesis with an emphasis on structured representation of musical signals. Further, we have detailed its modular design and demonstrated its applicability to computer-aided composition.

Chapter 5

Experiments

5.1 Introduction

In this chapter we present experiments wherein we consider two sources of *a priori* information stemming from symbolic entities, namely sound source and notated musical context. Recalling the points where outside information can be introduced into a dictionary-based method (DBM), as presented in Chap. 3, we sought techniques that included high-level information in the design of targeted dictionaries, decomposition strategies, and post-processing procedures. In this way, we have adopted a *top-down* approach to signal analysis where we were motivated to obtain structured and mid-level representations of musical audio.

In the techniques and experiments described in the first section, we are chiefly interested in the application of knowledge about sound sources. In Sec. 5.2.1, we introduce an extension to Matching Pursuit (MP), which iteratively redefines the effective analysis dictionary during the pursuit according to an estimate of spectral peaks. Motivated by the modeling of sound sources, this algorithm features a parameterized tolerance for inharmonicity. Following the description of this algorithm, we demonstrate some of its properties through the decomposition of synthetic signals. In Sec. 5.2.2, we describe our application of this technique to the identification of source-inspired dictionaries to be used by a standard MP. In the subsequent experiment, Sec. 5.2.3 these structured elements are compared to decomposition by a conventional MP. In Sec. 5.2.4, we apply our technique to the determination of instrument-specific analysis structures and apply them for the task of separable sound source modeling for synthesis.

In Sec 5.3, we consider a symbolic representation of the musical context associated with a given target. Here, we demonstrate how such high-level information can be used to inform structured atomic decomposition and the visualization of mid-level representations.

In Sec 5.4, we demonstrate how post-processing in the context of structured decomposition can be used to construct intermediate representations for visualization and synthesis.

The availability of sound examples corresponding to the sections of this chapter is detailed in Appendix A.

5.1.1 Terminology for Structured Representation

Prior to the presentation of experiments, we define our use of terminology for structured representation in terms of a hierarchical organization based on atoms and groups thereof. This terminology is used in reference to both analysis and synthesis elements.

We recall that in the context of dictionary-based methods an *atom* refers to the lowest level of organization, and each structured representation can be reduced to a 'flattened' collection of such units. In the experiments that follow, a *structure* refers to a first-order group, i.e. a set of atoms. Similarly, a *metastructure* is a second-order group, i.e. a structure of structures. At the limit of our formalism an atom could be regarded as a structure of order 0. We favour this terminology as it can be extended arbitrarily to accommodate further grouping procedures.

From our perspective on structured representation, the collection of atoms associated with a generic synthesis book[1] constitutes a first-order group and is structurally equivalent to any other set of atoms. In this sense we are inspired by the descriptive language of Schaeffer's sound object, which makes no formal division between a recording of an entire piece of music and any excerpt from it [57]. By our terminology, any hierarchical organization built from atoms is uniformly regarded as a structure.

We contrast our interpretation to examples in the literature, e.g. [28] or [34], where a group of harmonically related elementary functions is deemed an *atom* and [15], where transient and tonal structures are assigned the term *molecule*. We notice that, though motivated by a description of a different character, the objects of their respective models are of the same structural order.

[1] Here, we give the presentation in terms of the synthesis construct, but it applies analogously to a dictionary

In terms of our implementation, we emphasize that the differentiating feature between a structure and a book is the functionality attached to the latter. That is, in addition to being associated with a set of atoms, a book also defines methods for operating upon this collection. We note that our descriptive formalism is paralleled in the behaviour of the `Group` base class in `pydbm` as each of its derived classes is initialized with a corresponding order N and each of its atoms is specified by a unique N-dimensional index. Further, the addition of `Group` derived classes, is handled such that they retain their internal structure.

5.2 Structured Analysis of Sound Sources

In this section we consider the symbolic notion of a *sound source identity* and examine how this feature can be applied to structured signal analysis. We introduce a variant of MP that can be used to model a signal, but can also yield a vertically-structured intermediate representation suited to the definition of targeted sets of analysis functions. After the presentation of this method, we demonstrate its application to the determination of source-inspired dictionaries, offer a comparison of MP based techniques for the modeling of monotimbral sounds, and apply it to the task of separable sound modeling for synthesis.

5.2.1 Spectrally-Inductive Matching Pursuit

As we were motivated by the development of analysis functions tailored to the spectral properties of specific sources, we begin our presentation by recalling two similar studies discussed in Chap. 3 (namely those of Leveau et al [34] and Cho et al [12]). In each, instrument-specific atoms are determined from analyses of a labeled database. However, the techniques involved in this process differed. In [34], collections of partial amplitudes are obtained according to a conventional spectral peak extraction procedure on the contents of a database of labeled samples. Afterwards, for each pitch label, the vectors of partial amplitudes are quantized in order to reduce the number of elements considered in the subsequent decomposition.

On the other hand, in [12], a MP decomposition is applied to a labeled database. The atoms determined from the decompositions are then prioritized, as discussed in Sec. 3.5.2, and a suitable subspace is determined for each instrument in consideration.

In comparing these two strategies, we notice appealing features of each and have implemented a hybrid strategy in an attempt to integrate the positive features of both. The

result is a variation of MP that iteratively decomposes a signal according to an estimate of spectral peaks.

Firstly, an advantage of the strategy employed by Cho et al is the multi-scale (and thus multi-bandwidth) description of the target sound. On the other hand, in [34], Leveau et al only use a single scale with a duration of 46 ms in their applications. Though not stated explicitly, the limitation of using a single scale could be related to the parameters of the peak extraction learning procedure, that is partial amplitudes were estimated relative to a single scale.

Second, is the issue of frequency resolution, which is in fact tied to another basic difference between these strategies. The technique described by Leveau et al features a built-in assumption of harmonicity. In this case, they can construct a dictionary that considers any fundamental frequency, assigning to each element the most appropriate vector of partial amplitudes. Of course, the cost of this assumption is that they can not model inharmonic sounds using this technique. On the other hand, Cho et al make no assumption about harmonicity. However, this implies that they must specify *a priori* which frequencies they will consider. For their preliminary subspace estimate, Cho et al report a constant frequency sampling of 800 points. This is possibly related to the computational complexity associated with atomic decomposition.

Having considered the advantages of the techniques discussed above, we developed a vertically-oriented extension to MP stemming from a supervised approach, which we refer to as *Spectrally-Inductive Matching Pursuit* (SpecIMP). In its application, a structure characterized by a set of time-localized spectral peaks is extracted at each iteration. Owing to established techniques in this domain, we improve the estimate of peak frequencies using the interpolation strategy presented in [61]. The amplitude and phase values for each peak are obtained as in a standard MP. This procedure is depicted in Fig. 5.1, where the 'update region' refers to the elements of the dictionary whose temporal location overlaps with the extracted structure. In the two segments that follow, we outline some properties of our approach and illustrate their utility through the analysis of synthetic signals.

Inharmonicity Tolerance and Effective Dictionary Size

In general, the motivation behind the development of this variant of MP was to help the technique identify quasi-harmonic structures while also permitting a description without a

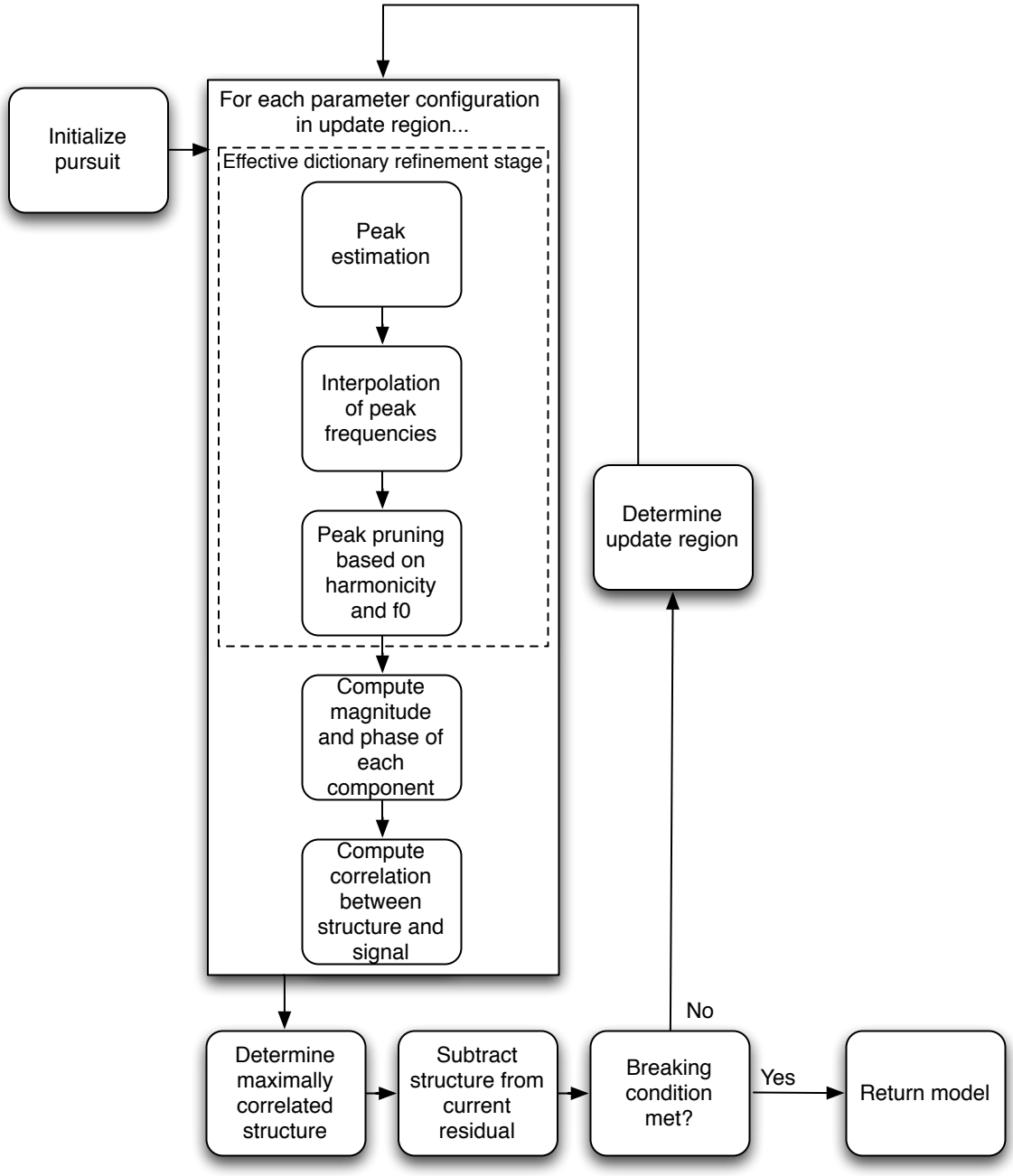


Fig. 5.1 Flow diagram of SpecIMP procedure

strict assumption of harmonicity. In the structured analysis of sound sources this feature was thought to be desirable, as the partials of many instruments occur at roughly integer multiples of a fundamental frequency, though in practice there is often deviation from the theoretical location. In this sense, it is a more general setting of harmonic MP [28].

To account for variability, we specify a tolerance as regards the degree of inharmonicity permissible as an additional parameter of the decomposition. In this way, the materials of the associated dictionary suggest a fundamental frequency and the inharmonicity tolerance is used to apply a constraint on the harmonic nature of the components above this fundamental. A high inharmonicity tolerance implies that a larger region of the spectrum is considered around the theoretical location of a harmonic component (at a specific time, with a specific analysis bandwidth)[2]. Though more accepting of deviation, the cost of a high inharmonicity tolerance is that the description of a harmonic source within a mixture could be obscured if an unrelated component is located within the bounds defined by the tolerance.

Generally speaking, it is not necessary to supply an extensive estimate of frequency parameters in the specification of the dictionary, thus reducing the complexity of the pursuit. This feature is a consequence of the fact that the peak estimation procedure is adapted to the residual at each iteration. This implies that the number of atoms considered by the decomposition is substantially larger than what is specified prior to analysis. Here, we contrast the specification of a *sparse* dictionary with the *effective* size of the dictionary used over the course of the decomposition.

As a variant of MP, the successive refinement the dictionary becomes important particularly when it is not necessarily complete, as the process of decomposition can introduce artifacts whose properties are difficult to predict. However, there is also a benefit associated with SpecIMP in an overcomplete setting, in the sense of the structured nature of the resultant representation. Furthermore, in comparison to a standard MP implementation, components in the upper region of the frequency spectrum are modeled in early stages of a SpecIMP decomposition to its vertical orientation. In practice, components in this region are often initially neglected by the *greedy* nature of MP as they typically account for relatively little signal energy though they are salient in terms of perception.

To illustrate the functionality related to inharmonicity tolerance, consider the signal

[2]Here, we note there is a built-in minimum spacing assigned in the peak estimation procedure relative to the bandwidth of a given window function and atom duration.

depicted in Fig. 5.2a and the corresponding sonogram in Fig. 5.2b (where the magnitude is given in dB). This signal contains a harmonic structure consisting of three components located at 220 Hz, 440 Hz, and 660 Hz. In addition to this harmonic group, it contains three inharmonic components located at 363 Hz, 589 Hz, and 910 Hz. A SpecIMP decomposition using a dictionary with defined fundamental frequencies and a low tolerance for inharmonicity is able to extract the harmonic group. An example of this capability is illustrated in Figs. 5.3a and 5.3b. The associated residual is depicted in Figs. 5.4a and 5.4b.

Structured Description of Transients with Harmonic Behaviour

Congruent with other dictionary-based decomposition techniques, SpecIMP facilitates multi-scale, i.e. multi-resolution, analysis. However, in a structured sense, this type of decomposition is more readily able to account for the broadband phenomena encountered in the attack portion of asymmetrical signal phenomena. That is, it provides a pseudo-harmonic description of a transient that might be encountered in the analysis of an instrument that can be characterized as a struck resonator, e.g. a piano or marimba.

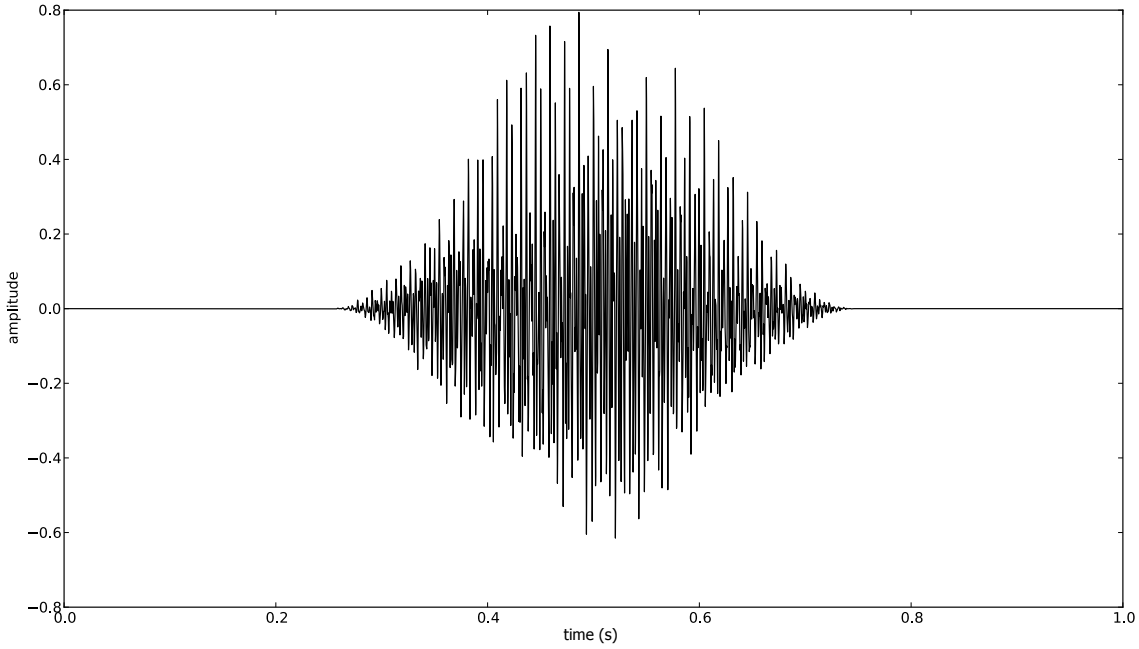
As an illustration, consider the signal in Fig. 5.5a, which is the result of a FOF window (with a brief rise time) being applied to a harmonic sinusoid with five components. As depicted in the corresponding sonogram, Fig. 5.5b, the brevity of the attack implies that there is energy distributed across many frequencies in this region of the signal. Though this behaviour is congruent with the concept of a transient, we would also like to provide a description of the internal harmonic relationship between the components of the signal in our model.

Using a multi-scale SpecIMP decomposition consisting of Hann atoms, the transient region is modeled by quasi-harmonically ordered, brief atoms. This feature is illustrated by the first five iterations of the decomposition, as depicted by the wivigram[3] in Fig. 5.6a. We measure the approximation of the model according to the signal-to-residual ratio (SRR) between the nth order model and residual, which is defined as

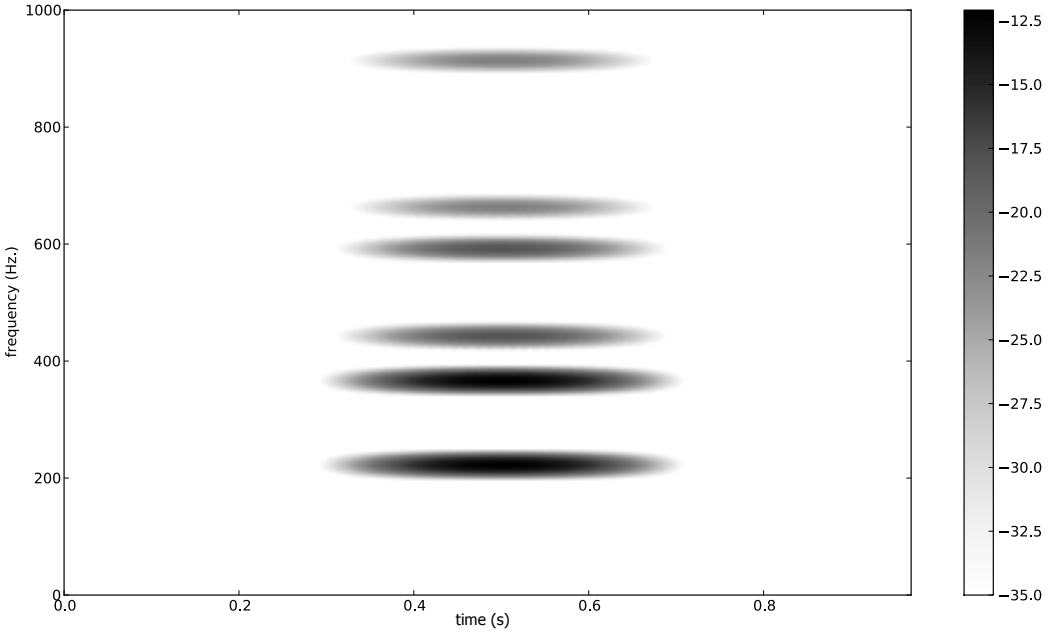
$$SRR_n = 10log_{10}\frac{||m_n||^2}{||r_n||^2} \tag{5.1}$$

After 53 iterations a total of 212 individual atoms have been extracted, see Fig. 5.6b, and

[3]In this section, wivigrams show the magnitude of the scaling coefficient obtained by the decomposition in dB.

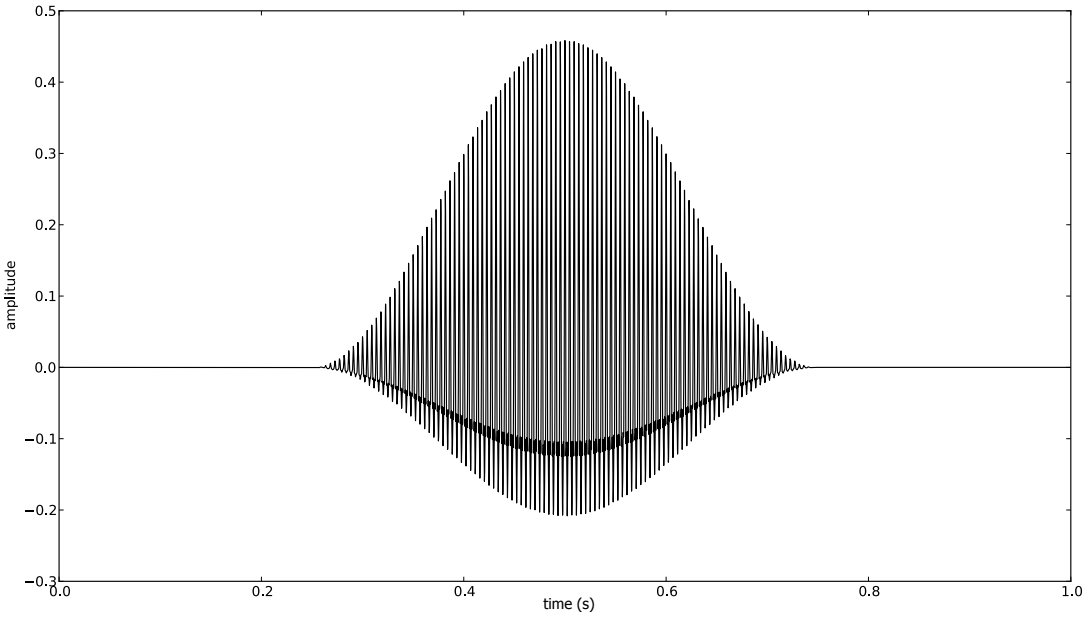


(a) Time-domain waveform

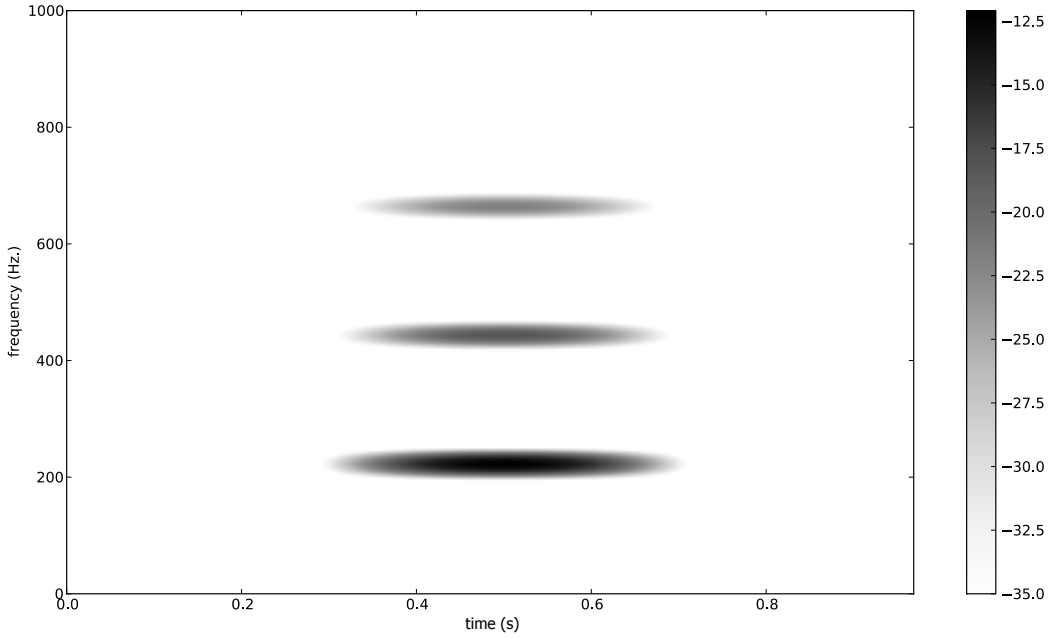


(b) Short-time Fourier transform

Fig. 5.2 Synthetic signal containing a harmonic group and inharmonic components.

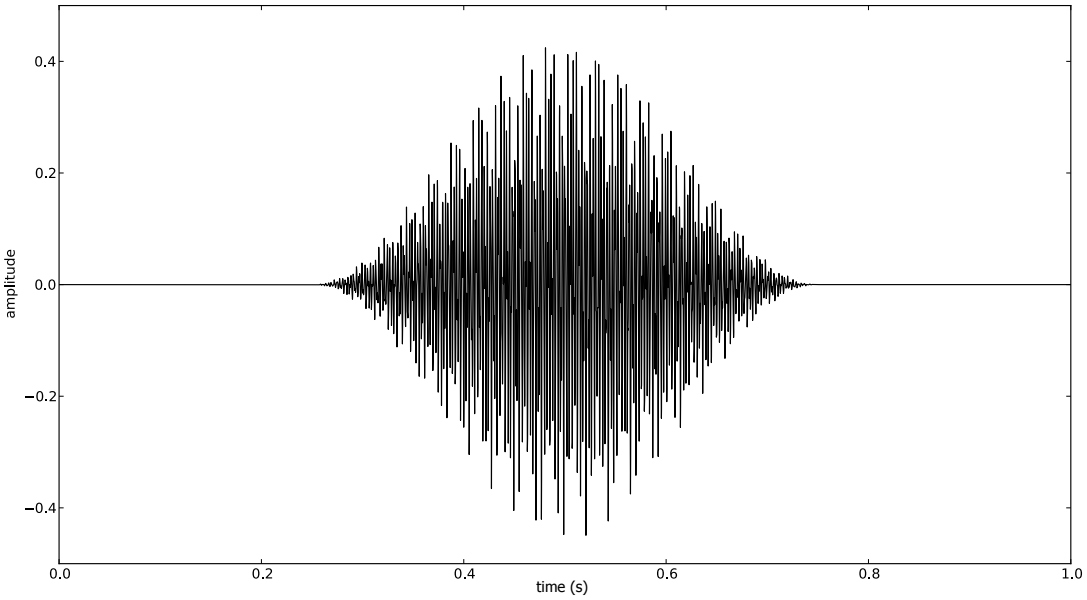


(a) Time-domain waveform

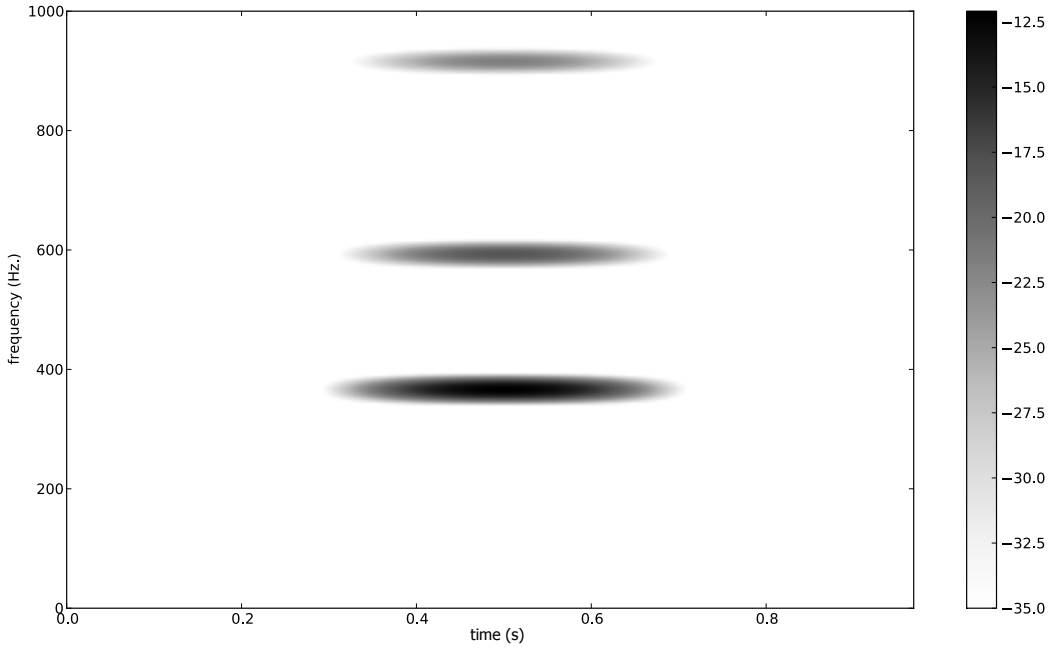


(b) Short-time Fourier transform

Fig. 5.3 SpecIMP model (low inharmonicity tolerance)



(a) Time-domain waveform



(b) Short-time Fourier transform

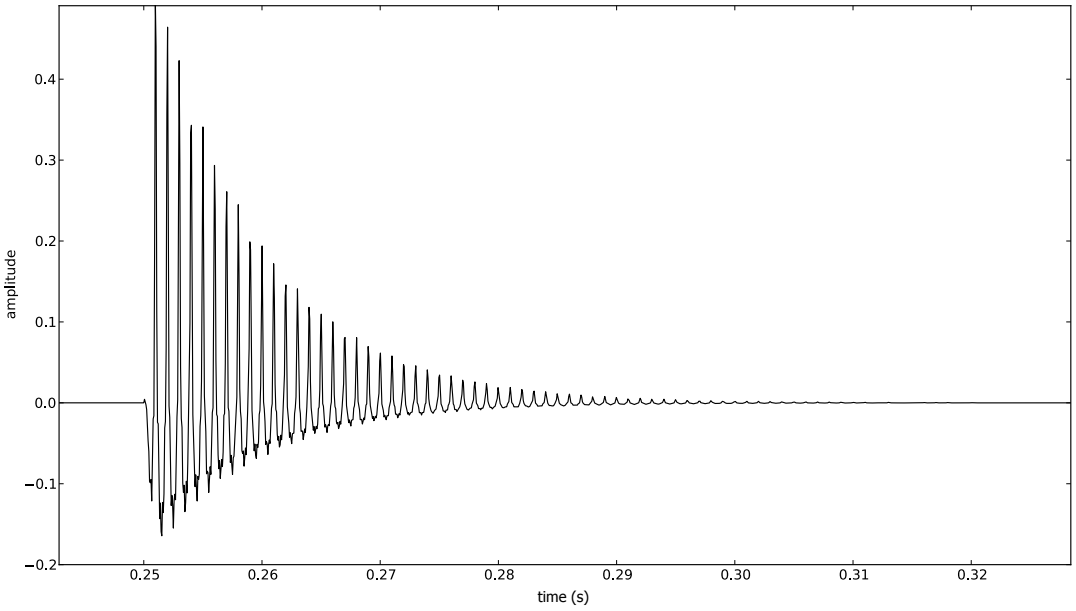
Fig. 5.4 SpecIMP residual (low inharmonicity)

the SRR has reached 40 dB. Audibly, this approximation is judged to be quite close to the original.

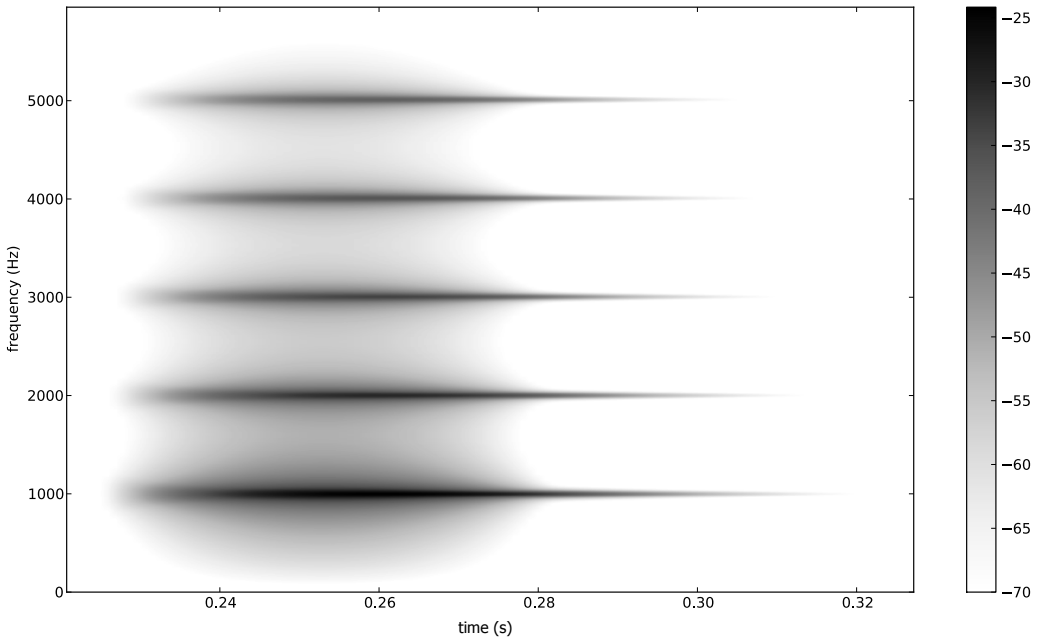
We contrast the previous strategy with a multi-scale Hann atom decomposition using standard MP. As MP does not target harmonic structure, the algorithm proceeds by extracting individual atoms in a greedy manner, i.e. initially favouring the non-transient part of the signal that accounts for the majority of the energy (Fig. 5.7a). The standard MP model achieves a SRR of 40 dB, however only after 440 iterations (Fig. 5.7b). Further, we notice the considerable number of atoms of a corrective nature. In fact, stemming from the greedy nature of the algorithm, the refinement of the low-frequency components with higher energy occurs prior to the modeling of high-frequency transient region. Again, the result is audibly quite close to the original. However, we argue that the model yielded by SpecIMP is more easily interpretable without any further organization, owing to the structured nature of the decomposition process.

The end result of such a decomposition is stored in a 'flattened' manner. That is, each spectral component is accessible by a 2-tuple index indicating its membership within a structure and the structure within a decomposition. As such, we can consider isolated atoms or the embedded spectral structure. Such a structure can lend itself to the application of audio effects, e.g. adding vibrato to a signal for an essentially harmonic model. To illustrate in a simple case, we refer again to the signal in Fig. 5.2a and the subsequent extraction of its harmonic structure, we are able to apply vibrato (whose depth is scaled logarithmically owing to the structured nature of the model) and add the result to the residual of the signal. The sonogram of the resulting signal is shown in Fig. 5.8.

In terms of computational complexity, the adaptive process of redefining the effective dictionary due to estimates of peaks typically implies that the initial dictionary has considerably fewer elements than a strictly overcomplete MP dictionary. In the example above, the former contained 20361 elements compared to the 741031 elements of the latter, which was constructed as a union of STFT frames with functions of three durations. This factor is not trivial in light of processes involving a database of sounds, such as the determination of relevant source-specific subspaces detailed below, or musical signal analysis in a practical setting.

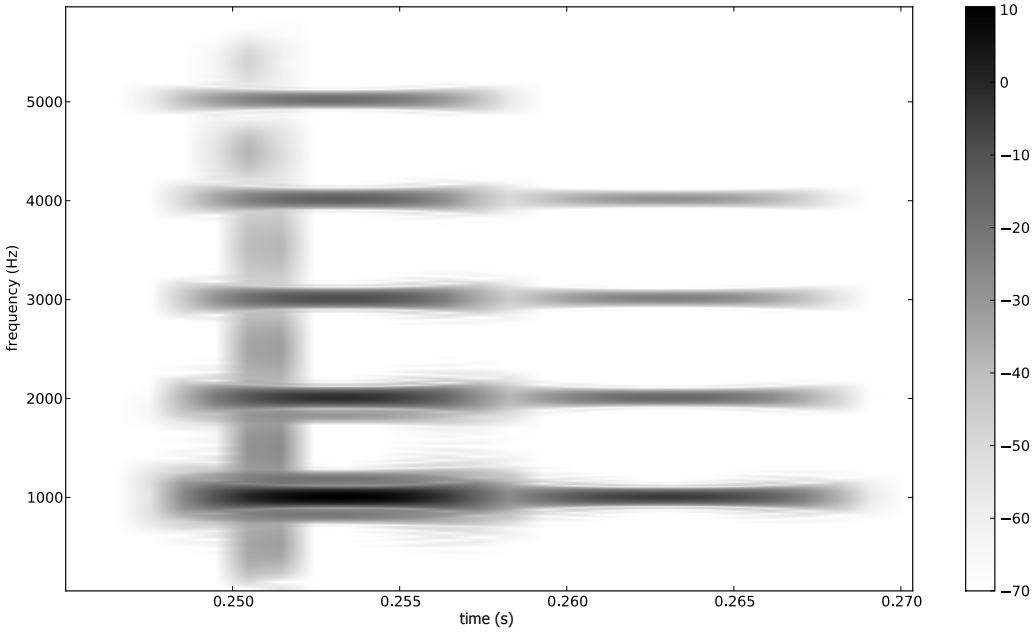


(a) Time-domain waveform

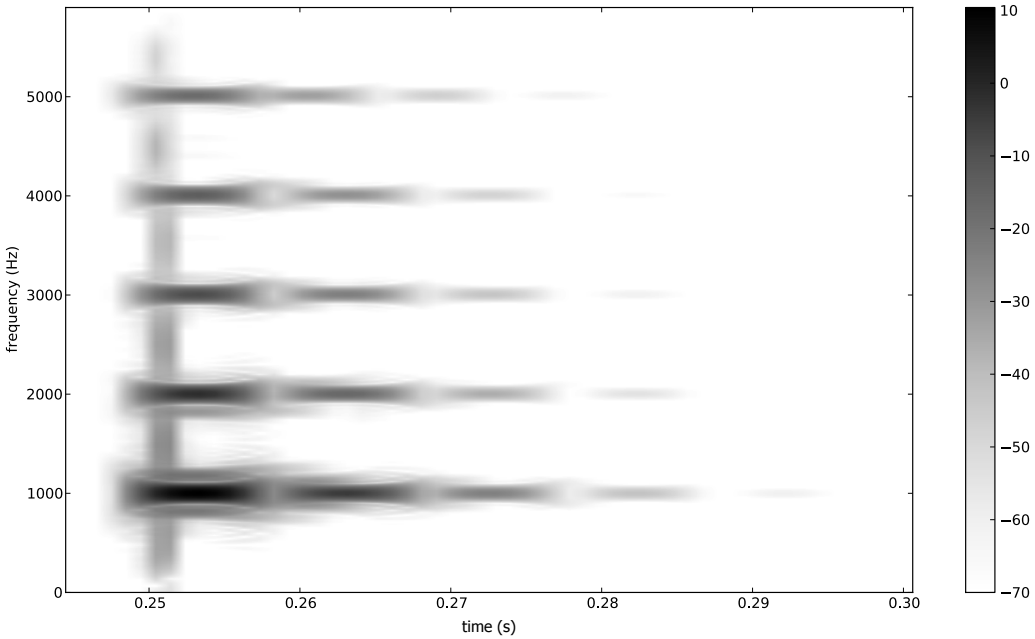


(b) Short-time Fourier transform

Fig. 5.5 Asymmetric harmonic test signal

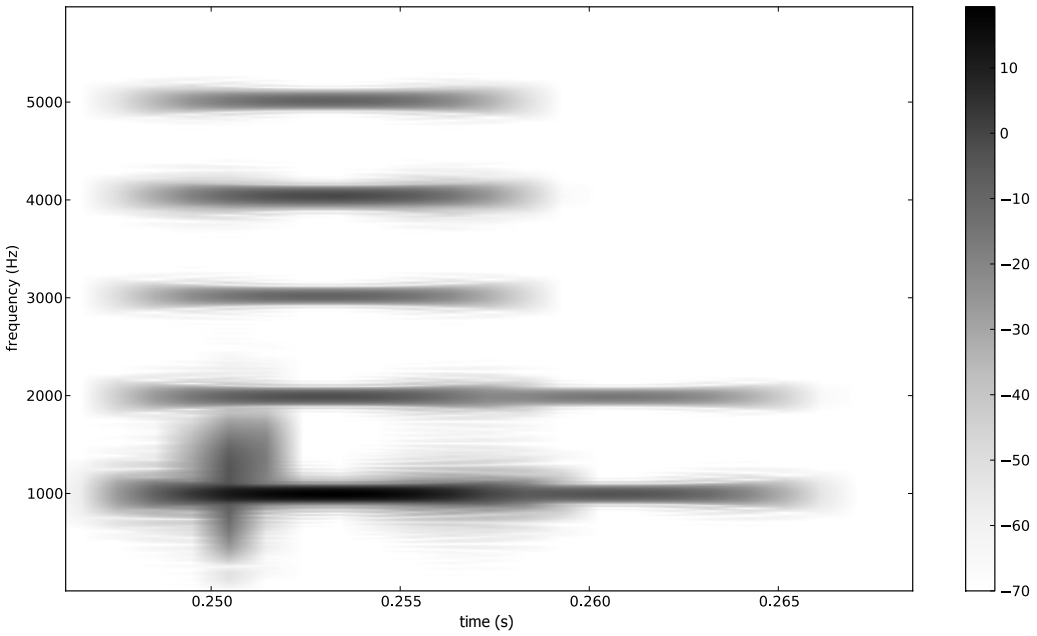


(a) Wivigram (5 iterations/20 atoms)

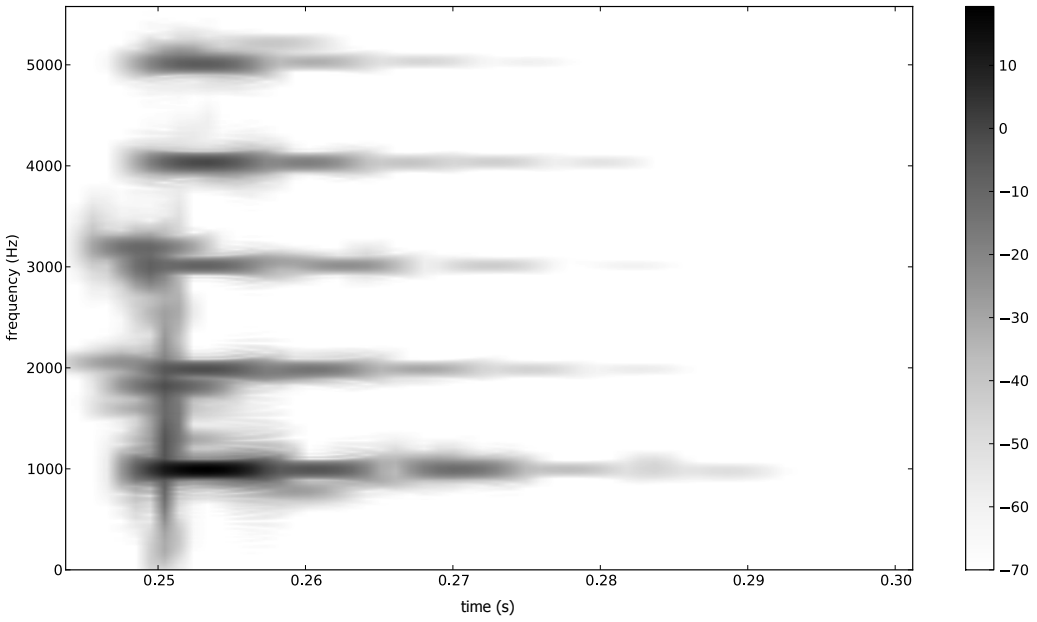


(b) Wivigram (53 iterations/212 atoms)

Fig. 5.6 SpecIMP model of asymmetrical harmonic test signal



(a) Wivigram of the standard MP model (20 iterations/atoms)



(b) Wivigram (440 iterations/atoms)

Fig. 5.7 Standard MP model of asymmetrical harmonic test signal

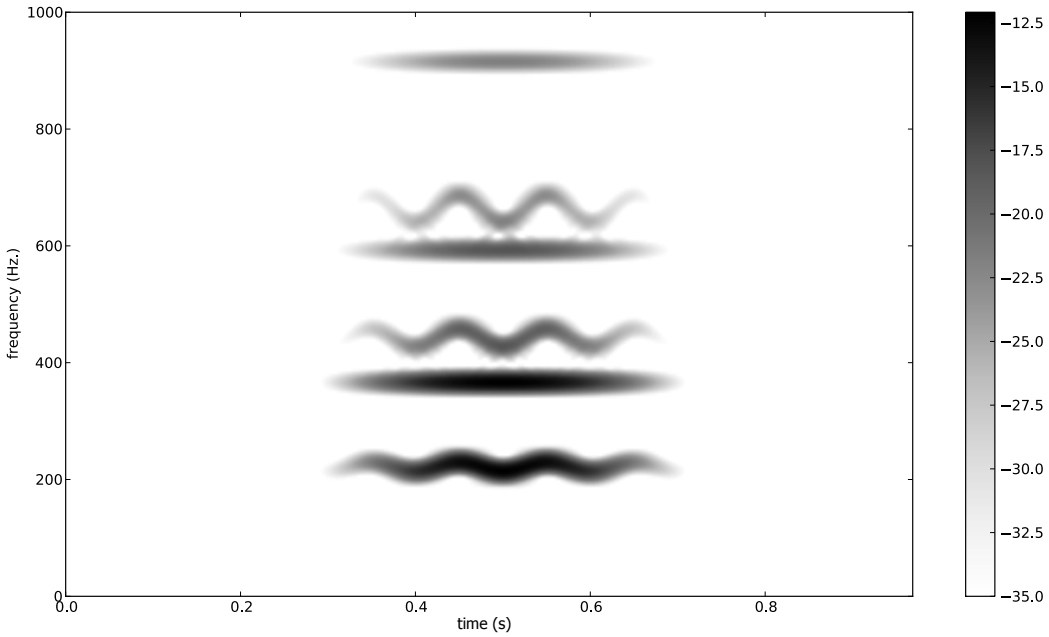


Fig. 5.8 Sonogram of resynthesis after extracting the harmonic structure and applying vibrato

5.2.2 Source-Inspired Sub-Dictionaries

In a different approach, we were concerned with the pre-processing capacity of SpecIMP. Here, with respect to the specification of dictionaries suited to the decomposition of signals featuring a particular sound source. The further partitioning of such a dictionary be accomplished to target a particular note of an instrumental source. We refer to a construct of this type as a Source-Inspired Sub-Dictionary (SISD)

In order to obtain such dictionaries, we applied SpecIMP to a database containing a selection of files from the McGill University Master Samples [41], where each is labeled with the appropriate note name. In the experiments described here, we consider cello, piano, and clarinet audio files that have been downsampled to a rate of 11025 Hz.

For each sound sample, a multiscale SpecIMP decomposition was conducted until a SRR of 35 dB was achieved or the model ceased to improve[4]. Each decomposition consists of atoms of three durations: 23 ms, 92 ms, 371 ms. Spectral peaks were deemed acceptable if their magnitude was above -90 dB and they fell within 3 semitones of their expected harmonic location. In this procedure, we were relatively generous with thresholds in order to obtain a rich model as elements deemed undesirable could be pruned afterwards. This process of pruning can be equated to a sampling of effective dictionary used by SpecIMP.

5.2.3 Comparison of Methods for Monotimbral Signal Modeling

In this subsection, we compare the performance of selected MP-based analysis strategies in the modeling of signals consisting of isolated cello and piano notes. Here, we are interested in how directly a particular strategy can represent a sound source and we a favour a model that accounts for a monotimbral signal in few ideally structured and salient components.

In this experiment, we consider standard MP, SpecIMP, and standard MP using a SISD. With regards to the two latter strategies, we examine the difference between an adaptive estimate of spectral components and a static dictionary that constitutes a sampling of predetermined model. Further, these extensions are compared to a typical MP decomposition.

To be congruent with the sub-dictionary determination procedure, we have considered decompositions using Hann atoms with three possible durations: 23 ms, 92 ms, 371 ms.

[4]This can be the case if the residual contains no viable spectral peaks, e.g. if the peak frequency location lies outside the bounds defined by the inharmonicity tolerance or if peaks fall below the assigned dB threshold.

The generic dictionaries, to be utilized by standard MP, are specified as a union of sampled STFT frames relative to each duration. So, regardless of the analysis target, the frequency resolution in these dictionaries is a constant 2.691 Hz.

The dictionaries for SpecIMP were centered around the fundamental frequency corresponding to the note in question. Multiple configurations related to the dB threshold for spectral peak estimation and inharmonicity tolerances were attempted throughout the course of this experiment.

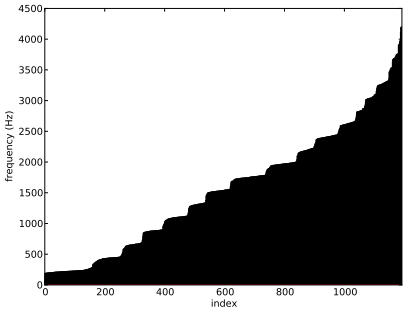
Source-specific dictionaries consist of atoms suggested by prior SpecIMP decompositions. These atoms have been quantized in terms of frequency, where the choice quantization factor represents a tradeoff between frequency resolution and dictionary size (consequently the associated computational complexity). In contrast to the frequency resolution of dictionaries for standard MP, the sampling of the spectrum in source-inspired dictionaries is not done at fixed bins and they exhibit a greater resolution around expected components (see Fig. 5.9).

In the decompositions considered here, we quantized the frequencies of the SpecIMP decomposition using a baseline of 0.11025 Hz, that is 0.001% of the sampling rate. Furthermore, the magnitude coefficients determined in the preceding SpecIMP decompositions were used to further prune the members of the dictionary. Here, we assigned a threshold of -60 dB to the L^2 norm of the atoms determined by SpecIMP.

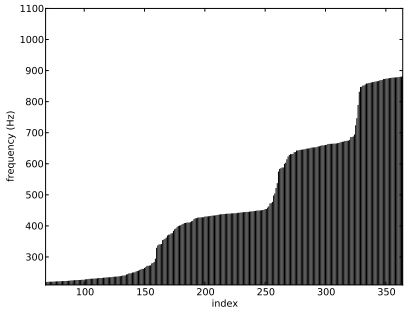
In this experiment, the decomposition targets were selected piano and cello samples from the RWC database[25]. Each represents a different configuration with the following parameters: instrument, note, performer, playing style, and playing dynamic. These configurations are summarized in Tab. 5.1.

For each sound sample and each decomposition strategy we initially conducted a low order approximation of 100 individual components. We chose to stop at this point, as the decay of residual energy has been proven to decay exponentially in MP and, in this experiment, we are primarily concerned with how a given dictionary/decomposition strategy behaves towards highly-correlated signal components. However, for MP and SpecIMP we later compared a higher order estimate. Here, the latter is in fact 100 iterations the spectral pursuit, that is the resultant models consist of 100 ordered structures.

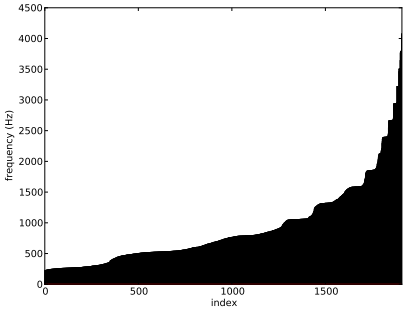
The results of the experiment are displayed in Tabs. 5.2 and 5.3. Here, the upper table shows decompositions with a breaking condition at 100 individual components (atoms) and below decompositions where the breaking conditions were set at a SRR of 35 dB, as well



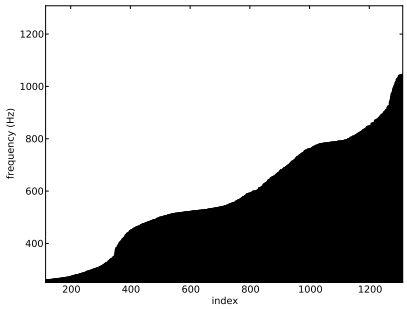
(a) inst=cello, f0=220 Hz



(b) inst=cello, f0=220 Hz (detail)



(c) inst=piano, f0=261.63 Hz



(d) inst=piano, f0=261.63 Hz (detail)

Fig. 5.9 Unique frequencies in source-specific dictionaries.

Index	Instrument	Note	Performer	Playing Style	Dynamic
1	cello	C2	1	Normal	Forte
2	cello	E2	1	Normal	Forte
3	cello	G2	1	No Vibrato	Mezzo
4	cello	F#5	1	No Vibrato	Mezzo
5	cello	A3	3	Normal	Forte
6	piano	A#1	1	Normal	Forte
7	piano	G7	1	Normal	Forte
8	piano	F5	1	Stopped	Mezzo
9	piano	F2	3	Normal	Forte
10	piano	C6	3	Normal	Mezzo
11	piano	C4	2	Normal	Forte

Table 5.1 Summary of sound sample parameters

	MP				SISD-MP				SpecIMP (100 comp.)			
Index	D	B	SRR	ISD ($\times 10^3$)	D	B	SRR	ISD ($\times 10^3$)	D	B	SRR	ISD ($\times 10^3$)
1	155228	100	12.99	18100.75	199254	100	13.15	20329.21	5480	100 (5)	7.13	4410.84
2	92852	100	19.24	29073.03	104714	100	19.76	21157.80	3320	100 (5)	5.65	8843.39
3	175635	100	18.375	9967.12	31617	100	17.06	10340.64	6210	100 (5)	2.17	415.35
4	143109	100	17.76	359.09	70809	100	17.03	224.26	5070	100 (20)	16.635	361.82
5	169258	100	10.84	3959.70	62037	100	10.81	6180.12	5980	100 (13)	6.25	4252.01
6	166709	100	5.98	11834.54	147528	100	6.41	9355.42	5870	100 (5)	2.73	8278.57
7	28683	100	12.71	142.30	5124	100	5.78	63528.72	1170	100 (34)	12.9	70.68
8	20268	100	28.87	25.34	20386	100	25.11	286.74	840	100 (30)	27.32	18.04
9	149105	100	8.57	5896.62	165392	100	9.20	243102.54	5270	100 (4)	3.29	5029.70
10	31749	100	22.92	8.00	40386	100	15.46	723.69	1230	100 (58)	19.28	6.14
11	112494	100	17.01	846.41	111041	100	16.38	2714.44	4020	100 (26)	17.51	562.54

Table 5.2 Low order models
(Index=corresponding sample in Fig. 5.1, D=dictionary size, B=book size (with no. of structures in parentheses where appropriate), SRR=signal-residual-ratio, ISD=Itakura-Saito distance)

	MP				SpecIMP (100 struct.)			
Index	D	B	SRR	ISD ($\times 10^3$)	D	B	SRR	ISD ($\times 10^3$)
1	155228	2011	35.01	26.06	5480	4133 (100)	35.02	7.71
2	92852	1358	35.00	157.26	3320	3515 (100)	35.03	23.77
3	175635	908	35.00	65.06	6210	1978 (100)	35.27	15.08
4	143109	1649	35.00	0.38	5070	1075 (100)	27.62	22.35
5	169258	1728	35.00	0.32	5980	2314 (100)	30.49	0.12
6	166709	3367	35.00	5.36	5870	4344 (100)	32.57	1.06
7	28683	1065	35.01	0.0004	1170	1562 (100)	33.47	0.01
8	20268	202	35.01	7.88	840	221 (100)	34.10	3.173
9	149105	3352	35.01	0.92	5270	3439 (100)	28.67	0.19
10	31749	358	35.01	0.74	1230	168 (100)	23.71	5.27
11	112494	740	35.01	2.26	4020	1370 (100)	26.45	0.24

Table 5.3 High order models

as a maximum of 100 structures in the case of SpecIMP. In terms of the book size in the SpecIMP cases, the numbers in parentheses denote the number of structures, i.e. groups of atoms.

The measurements used to gauge the effectiveness of the model are the SRR (specified in Eq. 5.1) and the Itakura-Saito Distance (ISD), which is defined as

$$ISD\left(P\left[k\right],\hat{P}\left[k\right]\right)=\frac{1}{K}\sum_{k}\frac{P\left[k\right]}{\hat{P}\left[k\right]}-log\frac{P\left[k\right]}{\hat{P}\left[k\right]}-1 \tag{5.2}$$

for discrete power spectra $P[k]$ and $\hat{P}[k]$. We found the ISD to be of interest for our purposes here, as it was developed as a perceptual measure of the distortion between a spectrum and its approximation[46].

Interpreting the results of this experiment, the measurements (i.e. SRR and ISD) contrast the decomposition strategies employed by MP and SpecIMP. MP, which extracts one atom per iteration and looks only to minimize the energy of the residual as much as possible, was typically able to achieve higher SRR compared to SpecIMP in both low and high order models. In the high order model, MP also yields a more sparse representation relative to SRR. On the other hand, SpecIMP, which extracts a vertical structure at each iteration was almost always able to achieve a lower ISD. For our purposes, this feature is significant as we are primarily interested in meaningful intermediate representations for analysis/synthesis and we consider the ISD (along with the SRR) to be a better indication of the effectiveness of a model in this regard than the SRR alone. So, in the application of our technique, this experiment has demonstrated that we sacrifice a relatively high SRR for spectral similarity and a structured representation.

The exceptions where MP exhibited a superior performance in terms of ISD are observed in the case of piano notes that exhibit noise-like characteristics, e.g. the sound of the hammer, or the dense spectral content associated with sympathetic resonance below the fundamental. Though able to account for broadband signal features around partials, these features were difficult for SpecIMP to account for, e.g. sample nos. 7 and 10. Similarly, sample no. 4 contains a substantial noise-like component, which was not efficiently accounted for by SpecIMP, explaining its relatively poor performance in modeling this sample.

As a follow-up experiment to test whether or not SpecIMP is able to account for noise-

like features in a transient if the analysis was configured differently, we added sets of Hann atoms with relatively short durations (1.5 ms and 6 ms) to the SpecIMP dictionary. Further, we added a set of FOF atoms with a relatively brief rise time (3 ms). We performed two decompositions of sample nos. 7 and 10. First, with breaking condition of 100 individual components and second with a breaking condition of a 35 dB SRR. The results of this experiment are shown in Tab. 5.4. We observe that the decomposition of these sound samples using SpecIMP with augmented dictionaries was superior than their MP counterparts in the previous experiment in terms of ISD and comparable in terms of SRR. Also, we notice that these samples are more effectively described by structures containing relatively few broadband elements. This is unsurprising, as these sounds are representative of a brief decay time, i.e. there is minimal resonance or energy concentration around partials.

	SpecIMP (100 comp.)				SpecIMP (35 dB)			
Index	D	B	SRR	ISD ($\times 10^3$)	D	B	SRR	ISD ($\times 10^3$)
7	23380	100 (25)	12.6	30.2	23380	1244 (320)	35.01	0.00001
10	24400	100 (39)	17.64	1.31	24400	1441 (846)	35.01	0.007

Table 5.4 Results of follow-up experiment

The results associated with SISD-MP were inconsistent, not showing any clear advantage in terms of SRR or ISD. There were instances where a given model performed better in terms of a measurement, however there were also examples where it performed quite poorly (especially in terms of the spectral measure). This deficiency is possibly related to the sampling of the prior models, or their analysis parameters. For instance, we notice that the performance of SISD-MP was comparable (and occasionally superior) to MP for the cello samples. In this way, the difficulty associated with noise-like features in SpecIMP was passed to the SISD. Further, as demonstrated by the follow-up experiment, the minimum scale of 23 milliseconds was not sufficiently short to span the transient region of the piano notes in an effective way.

Another possible shortcoming pertaining to the SISDs could be the material from which the prior models were built, if for example the database was not adequately representative. In light of these limitations, we conducted the remainder of the experiments in this chapter taking advantage of the adaptive and structured nature of SpecIMP.

In the experiment presented in this section, the models obtained via SpecIMP were less sparse than those of a traditional MP. However, in the contexts of targeted analysis

stemming from symbolic sources and building perceptually-motivated intermediate representations of audio we have chosen to develop SpecIMP further in the experiments that follow. This is due to its structured nature, its performance in terms of ISD, and its reduced computational complexity.

5.2.4 Sound Source Modeling in Mixed Signals

While SpecIMP is able to model structures with varying degrees of harmonicity (or mixtures thereof) it is not able to resolve spectral components that fall within the bounds defined by an inharmonicity tolerance that could have originated from different sources. In this subsection we demonstrate that the SISDs, though not demonstrably superior for sound modeling, can be used to obtain instrument-specific structures for analysis. We have tested these constructs in the task of separable source modeling for synthesis in a simple case where the partials of the sources do not overlap substantially.

In this experiment, SISDs were obtained according to the method described in Sec. 5.2.2. However, in this case, a combination of k-means clustering and threshold-based pruning was used to define instrument-specific analysis structures. A selection of instrument-specific structures are shown in Figs. 5.10 and 5.11.

As in SpecIMP, a set of atoms is extracted at each iteration. However, in contrast to that approach, in this case the dictionary is static (i.e. not adapted to the residual) and each component of an analysis structure is associated with a an amplitude scalar α. In order to permit the comparison of structures in a decomposition using the greedy heuristic of MP, the energy of each group has been normalized prior to analysis such that $\sum_k \alpha_k^2 = 1$, where k is an index over the components of a structure.

Two mixtures containing single clarinet and cello notes were made using materials from the McGill University Master Samples. The simple nature of these targets is indicative of our motivation, where we are concerned here with an invertible model of separable parts as described in Ellis and Rosenthal's criteria for desirable mid-level representations [19], rather than only classification.

In this experiment, each mixture was modeled with a dictionary specified as the union of the appropriate elements for each source and note. This is to say that the cello and clarinet analysis structures were considered over the course of the same decomposition.

For each mixture, we extracted a total of 100 structures. The results of this procedure

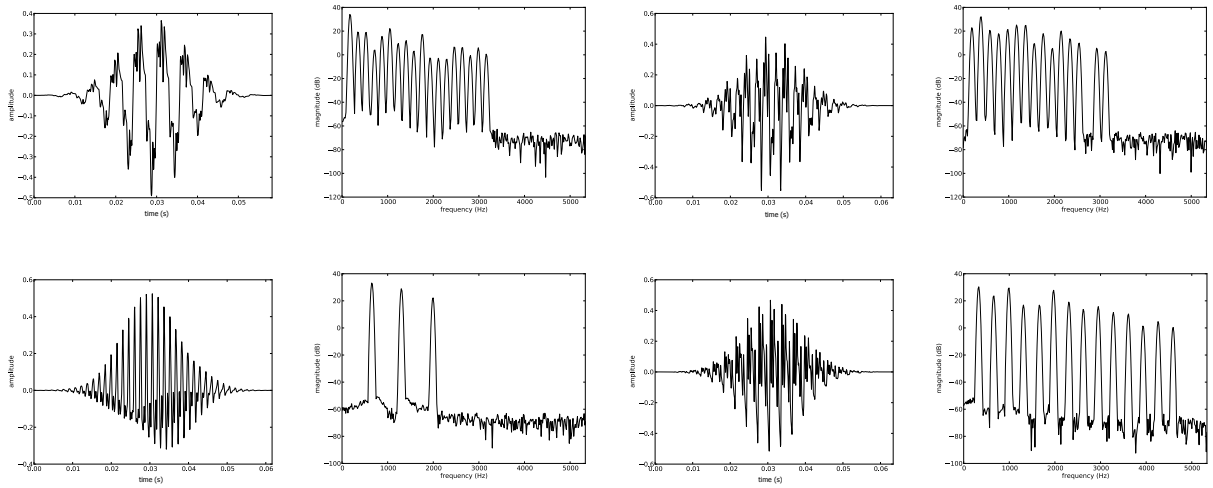


Fig. 5.10 Selected cello analysis structures and their spectra

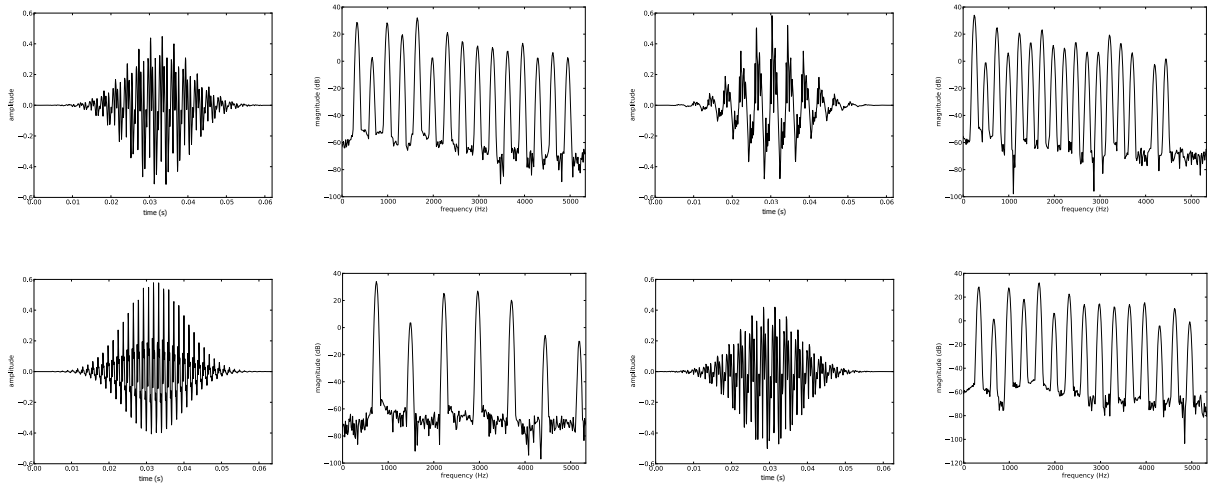
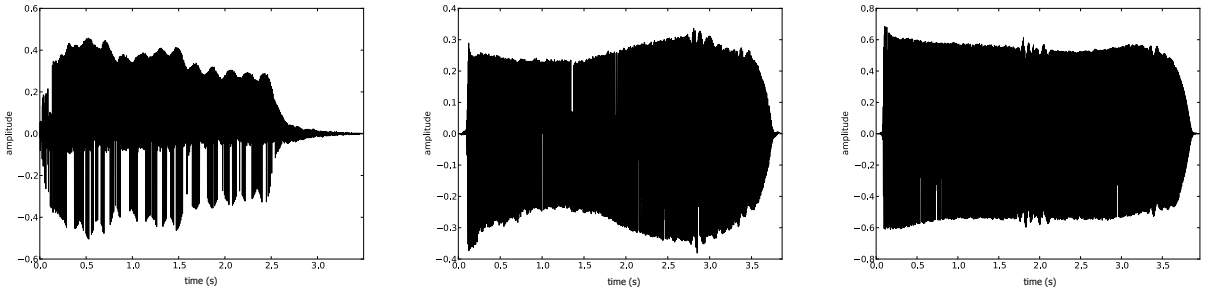


Fig. 5.11 Selected clarinet analysis structures and their spectra

	Mixture			Cello Part			Clarinet Part		
Mix. Index	Book	SRR	ISD ($\times 10^3$)	Book	SRR	ISD ($\times 10^3$)	Book	SRR	ISD ($\times 10^3$)
1	1079 (100)	17.34	715.25	564 (52)	1.2	0.8	515 (48)	-3.62	1.32
2	1123 (100)	17.39	485.121	528 (36)	-5.24	1.775	595 (64)	-3.03	0.54

Table 5.5 Results of separation by instrument-specific analysis structures



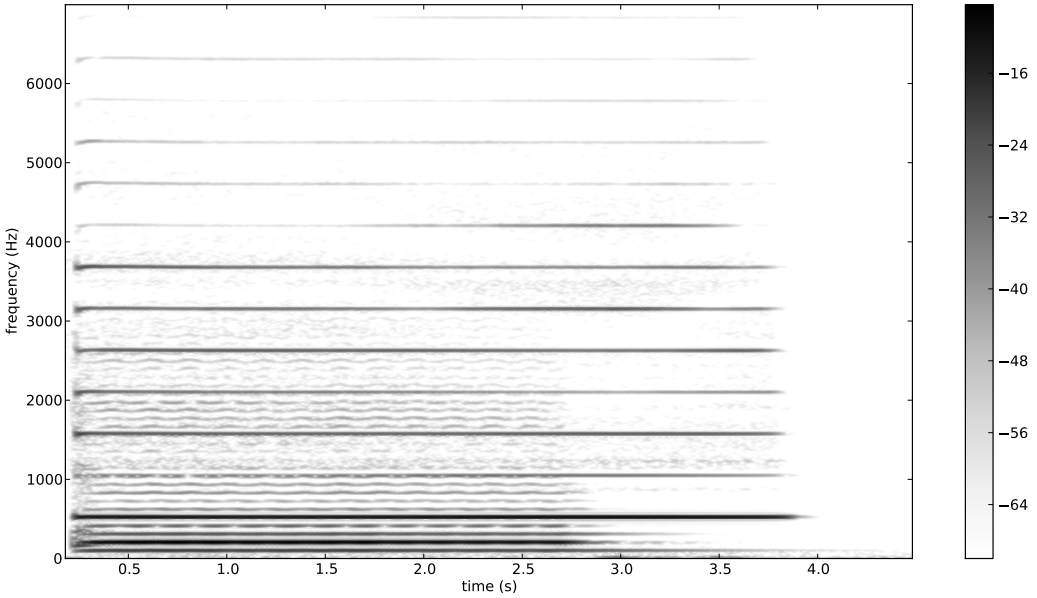
(a) Cello (G#2, 103.826 Hz) (b) Clarinet (C5, 523.25 Hz) (c) Clarinet (D5, 587.329 Hz)

Fig. 5.12 Components of target mixtures

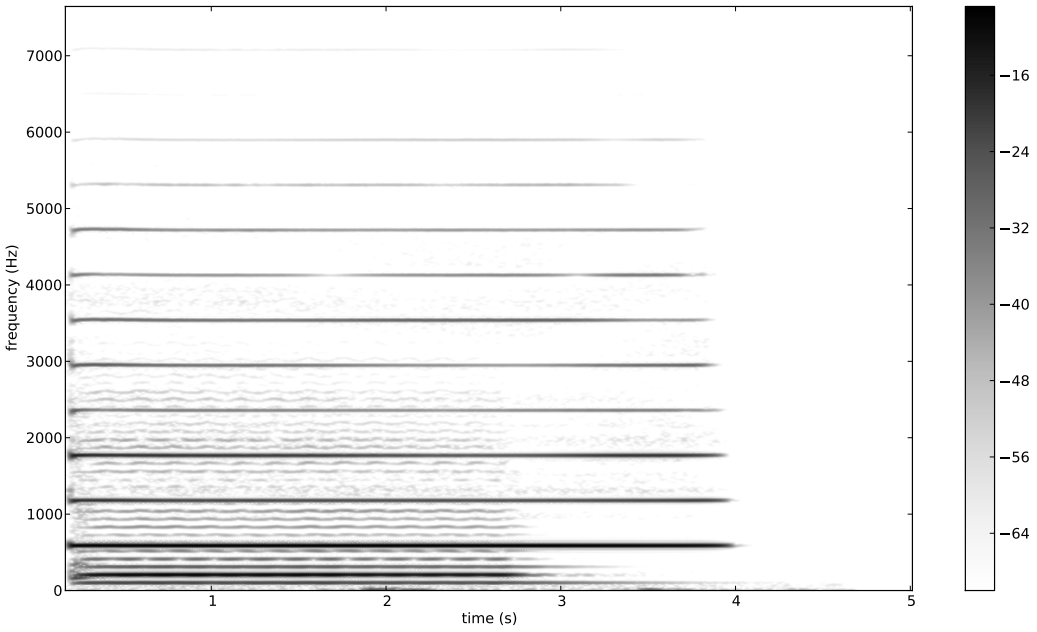
are summarized in Tab. 5.5. Interpreting these figures, and in comparison to the previous experiment, we note that the ISDs related to the extracted parts are relatively low, implying that there is a fair degree of spectral similarity between separated source and the unmixed target. On the other hand, the SRR associated with the extracted sources is quite poor. Furthermore, the time-domain waveforms of the extracted sources are noticeably different from the originals, substantially so in the case of the cello. This feature points to an interesting and somewhat unexpected feature of these models. In the case of the cello extraction, the vibrato of the target has been largely neutralized. We attribute this feature to the clustering stage of the preprocessing procedure, which acted to flatten any variation in the prior model. Also, we notice that essentially all aspects associated with noise profile of a sound source are left in the residual.

As with the performance of the SISDs in the previous experiment, we suspect that the instrument-specific analysis structures were overly rigid in practice. In this sense, it may be preferable to assume a purely harmonic model, as in [34], so that the fundamental frequency could be specified arbitrarily. However, this implies the use of a large dictionary, which is a detrimental feature associated with the technique due to the computational complexity involved.

In this subsection we have demonstrated the application of SpecIMP to the definition of instrument-specific analysis structures. Further, these structures have been tested in the separable analysis/synthesis of sound sources in two simple mixtures. Though the resultant models fall short of being audibly convincing, they constitute a second-order structured and invertible representation. That is, the model is divisible into sound source metastructures,

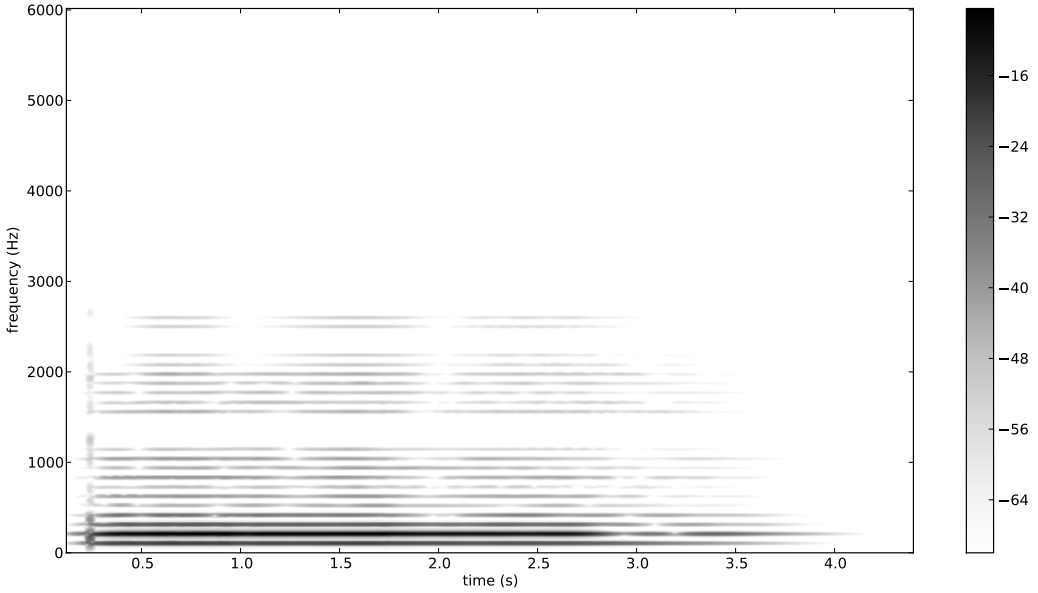


(a) Sonogram of Cello G#2 and Clarinet C5 mixture no. 1

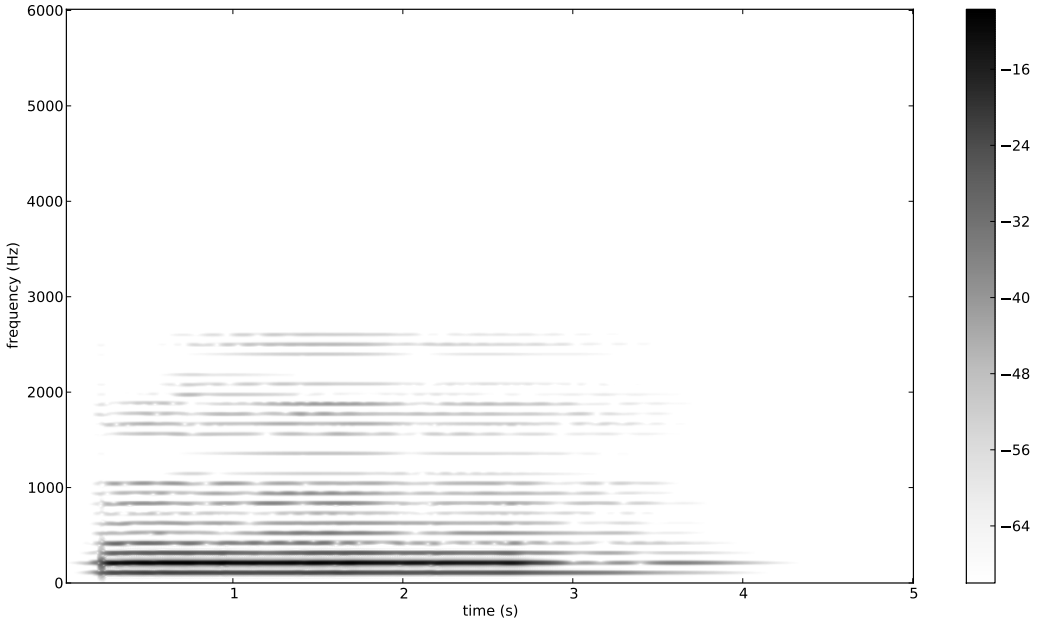


(b) Sonogram of Cello G#2 and Clarinet D5 mixture no. 2

Fig. 5.13 Target mixtures

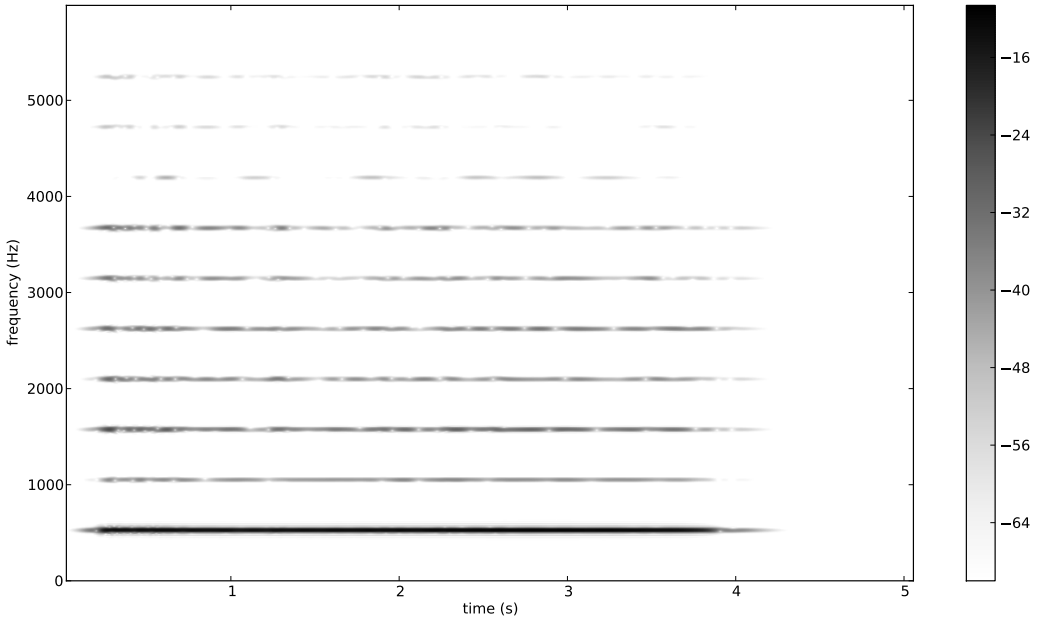


(a) Mixture no. 1

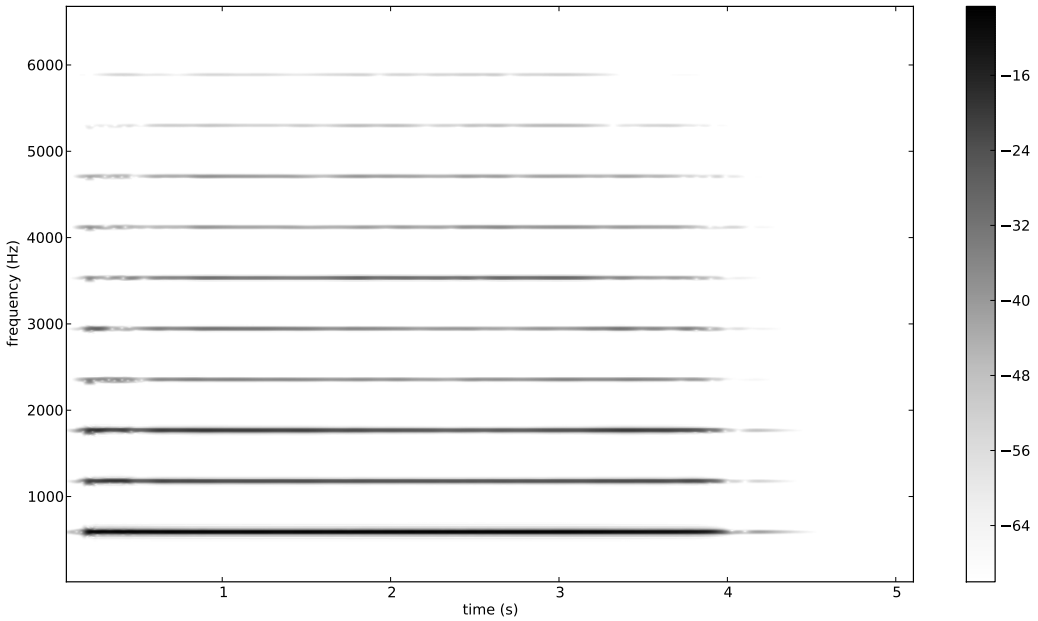


(b) Mixture no. 2

Fig. 5.14 Cello signals extracted from mixtures



(a) Mixture no. 1



(b) Mixture no. 2

Fig. 5.15 Clarinet signals extracted from mixtures

each consisting of ordered sets of atoms. It is our belief that even low fidelity models of this nature could present opportunities in the application of effects processing, e.g. spatialization.

5.3 Symbolic Representations for Dictionary-Based Signal Analysis

In the experiments presented in the previous section, we assumed knowledge of the fundamental frequency and source associated with the target sounds. However, in a practical setting, how can this high-level information be obtained and incorporated applied to musical signal modeling?

Rather than attempt to determine pitch or other relatively symbolic attributes from the signal itself, here we discuss a *top-down* approach to signal analysis starting from a symbolic representation. In this section, we are motivated by the qualitative description of sound in terms of separable objects and we have treated a symbolic representation as a starting point for the dissection of sounds into invertible structures with varying degrees of abstraction. In this section, we have tailored sets of analysis functions according to a given musical context using a score-based representation.

5.3.1 Visualization of Structured Decompositions

Prior to the experiments, we present our method for visualizing structured models developed in conjunction with the analytical techniques discussed in this subsection.

In contrast to the predominantly signal-based visualization of a decomposition by wivigram, we have developed a technique that retains some of the properties of the former but is more oriented towards a symbolic interpretation of musical sound. This technique is accomplished by the superposition of the sum the scaling coefficients associated with each structure in the model around its fundamental frequency. That is, this method of visualization is made possible due to the embedded structure of our decomposition strategy and is reminiscent of a *piano roll* MIDI editor found in many commercial music sequencing programs.

On the other hand, our method also depicts several signal-based characteristics of a model. For example, rather than discrete notes, we observe clusters of spectral structures

that are specified below the note level [5]. Typically, these structures have a more precise time-frequency location than a score-based representation. In our implementation, the resolution of these axes has been parameterized. So, in order to obtain a visualization closer to a MIDI representation, the frequency resolution can be reduced to a semitone. Conversely, a high frequency resolution yields a visualization similar to a log-scale wivigram. In a related feature, unlike a purely symbolic depiction, the approximate bandwidth of the fundamental atom of a structure is shown with the vertical dimension of the rectangle. Following the log-scale convention of musical notation, the vertical dimension of each rectangle is scaled appropriately.

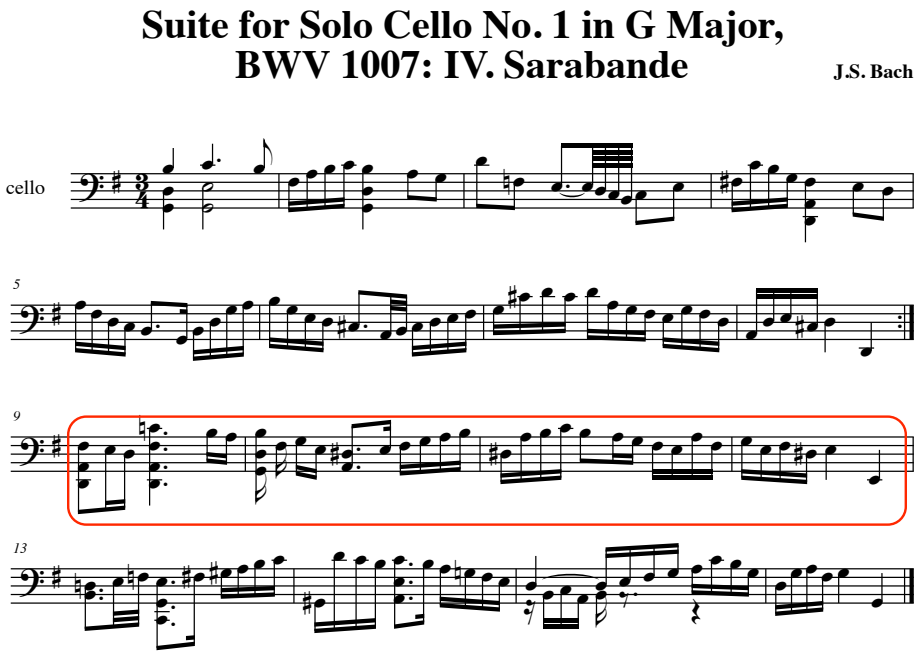
To demonstrate our method of visualization, we performed a score-informed decomposition of an excerpt from the first cello suite (henceforth referred to as BWV1007) [21]. The score representation of this excerpts is given in Fig. 5.16a. By way of contrast, we show the audio waveform associated with the cello suite excerpt in Fig. 5.16b and emphasize that, in this section, we seek a visualization that falls somewhere between the two. The application of our technique to a model of this excerpt is shown in Figs. 5.17a and 5.17b. For comparison, a wivigram of the same model is shown in Fig. 5.18. Here, we believe that our method represents a midpoint between commin signal and symbol visual depictions and that through configuration of parameters its orientation can be tuned towards a desired descriptive pole.

5.3.2 Modeling of Monophonic Musical Signals

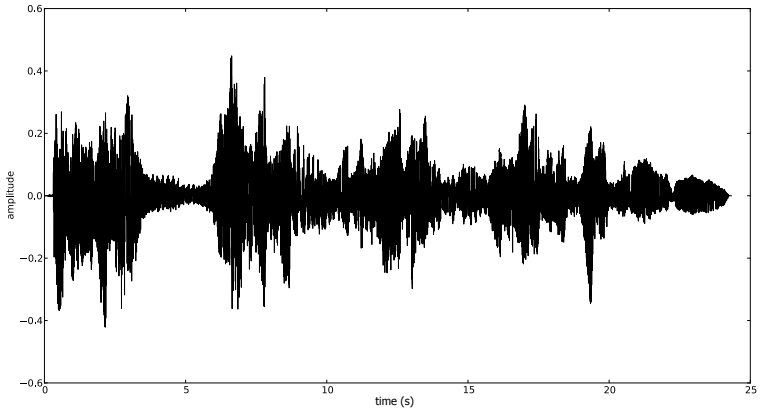
In this subsection, we present the use of a symbolic representation to facilitate the structured atomic decomposition of a musical signal with a single sound source and melodic line. We believed that this knowledge, along with our inductive decomposition algorithm, would contribute to a structured and invertible musical signal representation.

To test this hypothesis, we chose to model the subject of the C minor fugue from the first book of the Well-Tempered Clavier (BWV847) [26]. The corresponding score (in the format of a MusicXML file) was obtained from the Kern score database [53] and is depicted in 5.19. In our system, MusicXML was chosen as the format for score representation due to its presence in various programs related to music creation, such as OpenMusic and Finale, as well scholarly projects, such as Kern.

[5] Though, setting a threshold on the display leaves only the spectral structures with most overall energy, thus permitting a more MIDI-like visualization if desired.

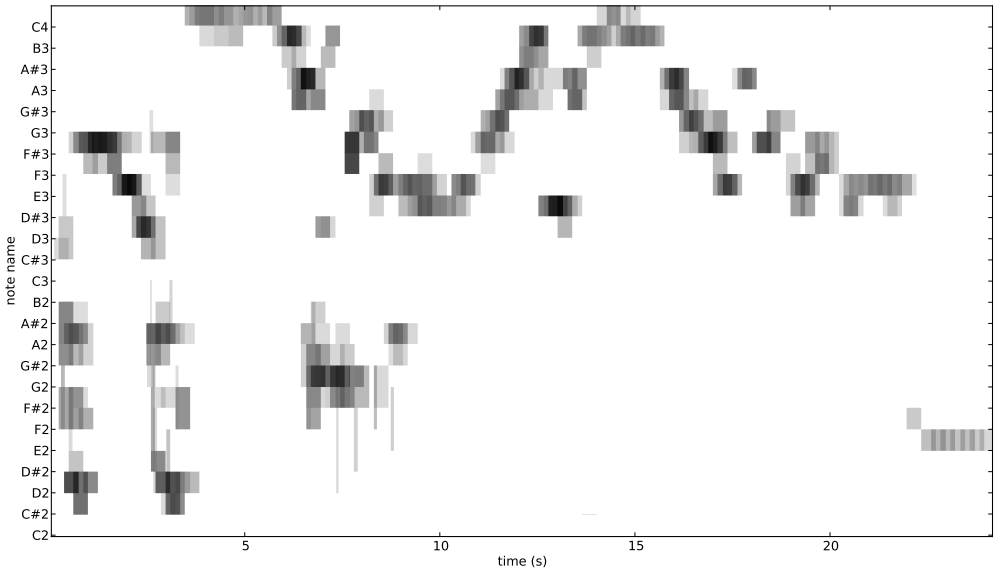


(a) Score, with analysis target outlined

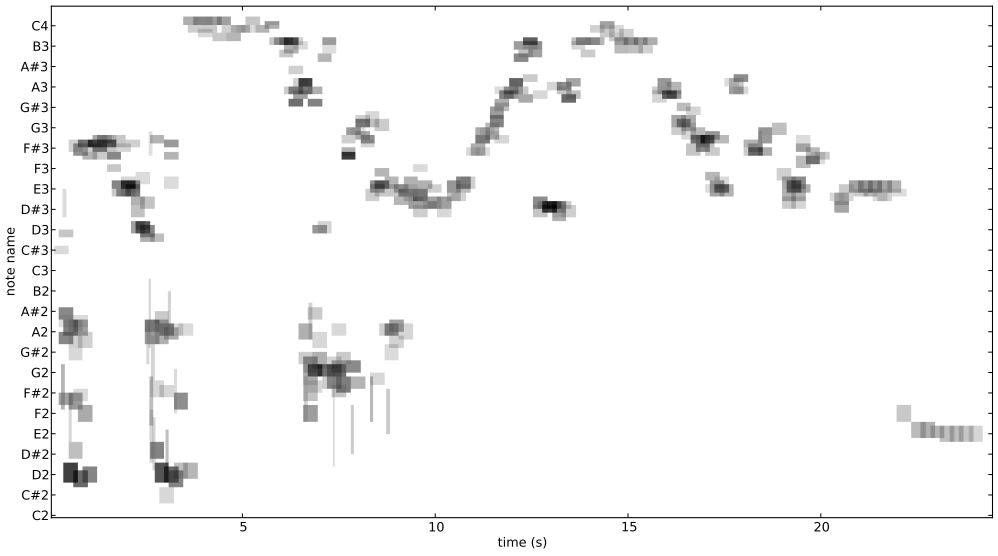


(b) Waveform of analysis target

Fig. 5.16 Two common representations of BWV1007



(a) Semitone frequency resolution



(b) 1/5 of a semitone frequency resolution

Fig. 5.17 Pseudo piano roll representations of BWV1007 demonstrating different frequency resolutions

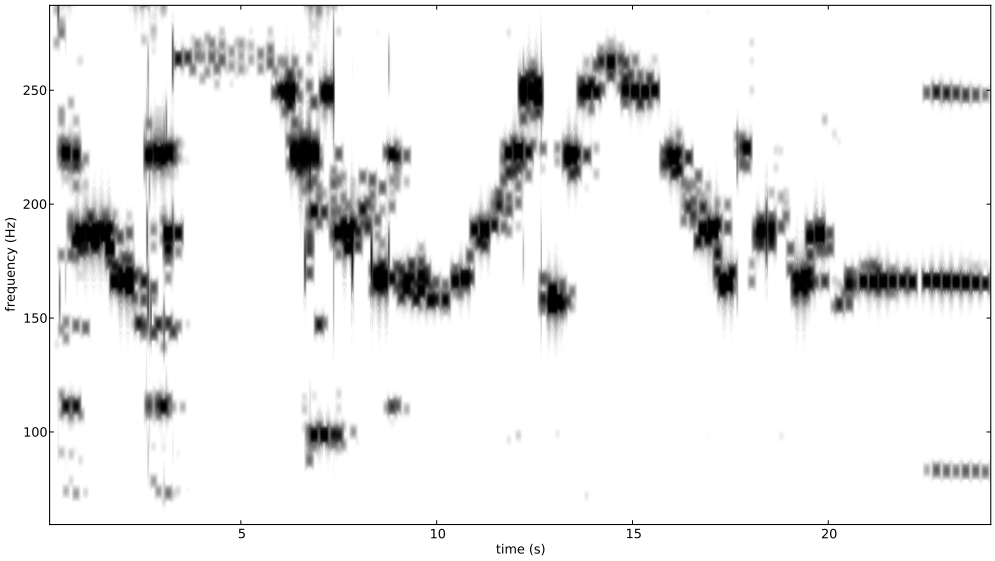


Fig. 5.18 Wivigram of BWV1007 model (detail)

After being converted into `pydbm Score` objects, the approximate time-frequency locations of the relevant notes were used to build dictionaries consisting of Hann atoms. Here, the durations and onsets were suggested using the conjunction of note length information from the score and the output of an adaptive marker generation algorithm developed for this task. However, we did not treat these theoretical temporal features as hard constraints since there is bound to be temporal variation when observed at the signal level. Variation in fundamental frequency was accounted for, in part, by specifying atoms around the theoretical location. However, the adaptive nature of SpecIMP also accounts for this variation within the bound of the tolerance.

In our experiment, we compared a decomposition informed by a symbolic representation to a baseline decomposition by standard MP.

Results for the score-specific technique are shown in Tab. 5.6, where we note that the bottom two entries contained FOF atoms (with short rise times) placed around expected note onsets. These were included in light of the follow-up experiment summarized in Tab. 5.4. The results of the standard MP decompositions are shown in Tab. 5.7.

In examination of these results, we notice that they are congruent with the experiment

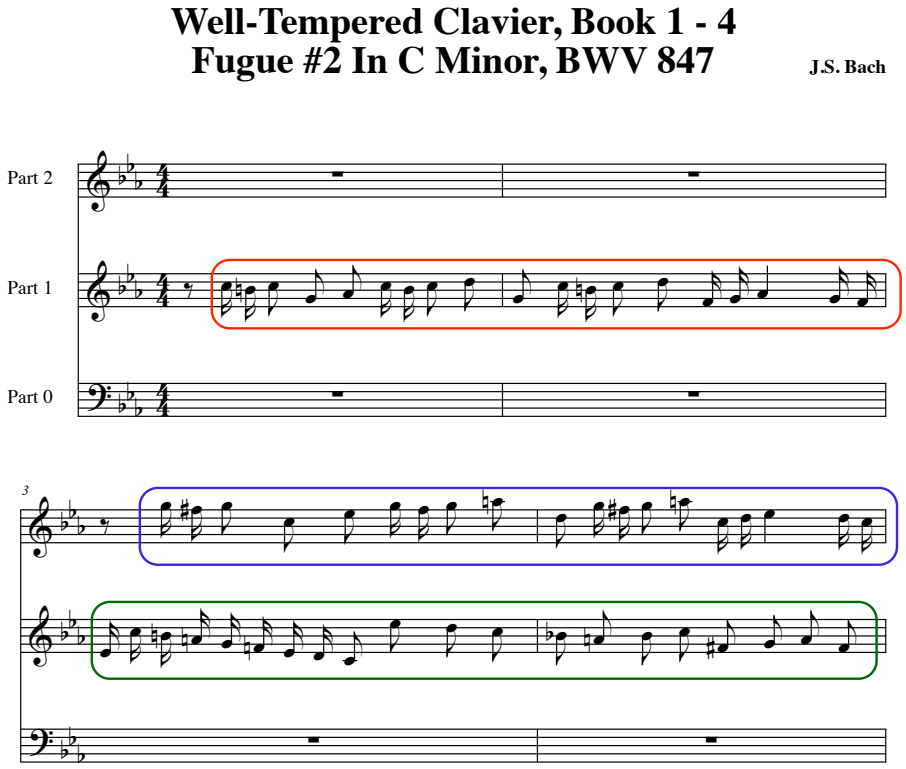


Fig. 5.19 Score representation with elementary musical analysis. The fugue subject outlined in red, the subject (transposed to the G min in blue), and the countersubject in green.

	Score-Informed SpecIMP						
Index	Dictionary	Book	durations	tol.	peak thresh. (dB)	SRR	ISD ($\times 10^3$)
1	23874	3553	(64, 256, 1024, 2000)	100	-80	15.2	2.13
2	10050	3592	(256, 1024, 2000)	100	-80	10.87	7.42
3	10030	1970	(256, 1024, 2048)	100	-90	19.67	8.52
4	19400	738	(128, 512, 2048)	100	-90	16.44	6.66
5	43090	8262	(128, 256, 1024)	1000	-100	29.54	0.37
6	16670	1962	(128, 1024)	200	-100	21.08	3.57
7	28930	680	(256, 1024, 2048)	100	-70	15.62	5.56
8	200	2005	(1024)	100	-70	-6.66	2.44

Table 5.6 Score-informed procedure

	Standard MP				
Index	Dictionary	Book	durations	SRR	ISD ($\times 10^3$)
1	184665	1000	(128, 2048)	20.45	23.1
2	276629	1970	(256, 1024, 2048)	25.51	4.83

Table 5.7 Baseline MP results

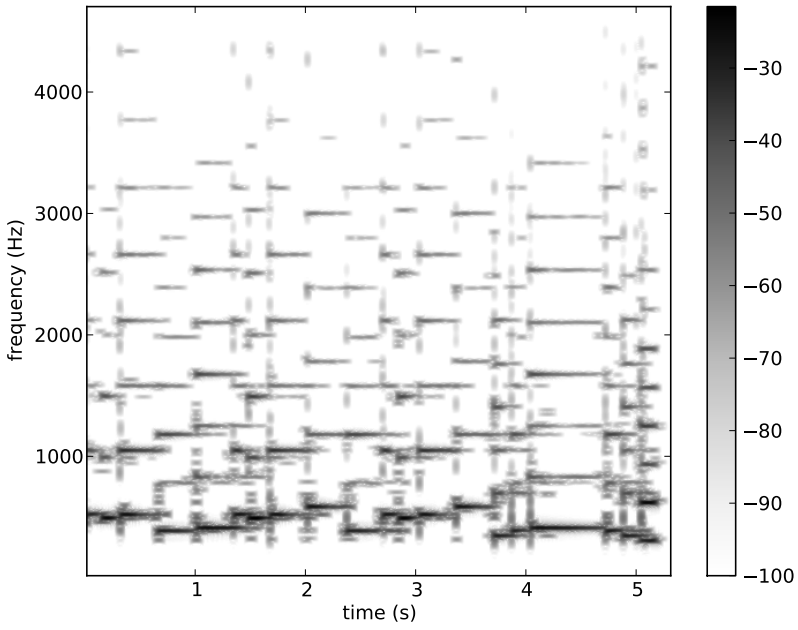
in Sec. 5.2.3 in that MP is able to achieve a higher SRR in direct comparison (Index 3 of 5.6 and Index 2 of 5.7. In this case the standard MP was also able to achieve a lower ISD. However, we refer to the sonograms in Fig. 5.20 to illustrate how MP favours the refinement of low-frequency components. That is, fewer of the upper partials are accounted for by MP.

Again, we believe this to be related to the durations used and SpecIMP was not able to account for the energy of the transient. This can be contrasted with decomposition no. 8, where the dictionary consisted of only a single duration of FOF with a rise time of 3 ms. This decomposition only modeled the attack portion of each note, see Fig. 5.21, and it achieved a relatively low ISD. Similarly, decomposition no. 1, was able to account for the attack regions and obtained a low score.

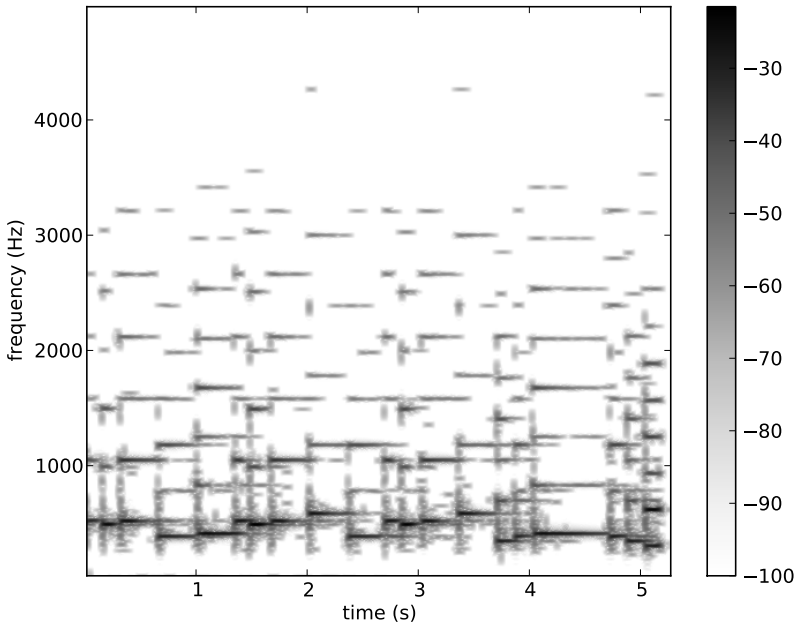
Further, we notice that increasing the inharmonicity tolerance (here given in midicents) improved the performance of the SpecIMP in the sense that more of the signal energy was accounted for. However, in this case the correspondence to the score is obscured. That is, many structures are used to account for energy around a partial. Here, we note that, in this experiment, a high inharmonicity tolerance was required to achieve a synthesis that was a close approximation to the original.

However, overall we prefer to develop the SpecIMP models further, as they constitute structured representations and lend themselves to intuitive visualization, Fig. 5.22. Data oriented in this ways also suggests further post-processing procedures, e.g. those presented in Sec. 5.4.

In this experiment we have demonstrated that a symbolic representation associated with a musical signal can guide dictionary-based analysis. Similar to how knowledge about the time-frequency characteristics of sounds and window functions can facilitate parameter configuration in low-level analysis methods such as the STFT, knowledge about the expected features of sources and musical context can facilitate analysis methods for mid-level representation. In terms of computation, it is certainly possible to proceed without this knowledge. However, the use of spectrally-oriented analysis structures, which can be tuned



(a) Score-informed SpecIMP



(b) Standard MP

Fig. 5.20 Sonogram comparison of decomposition strategies (1970 individual atoms each)

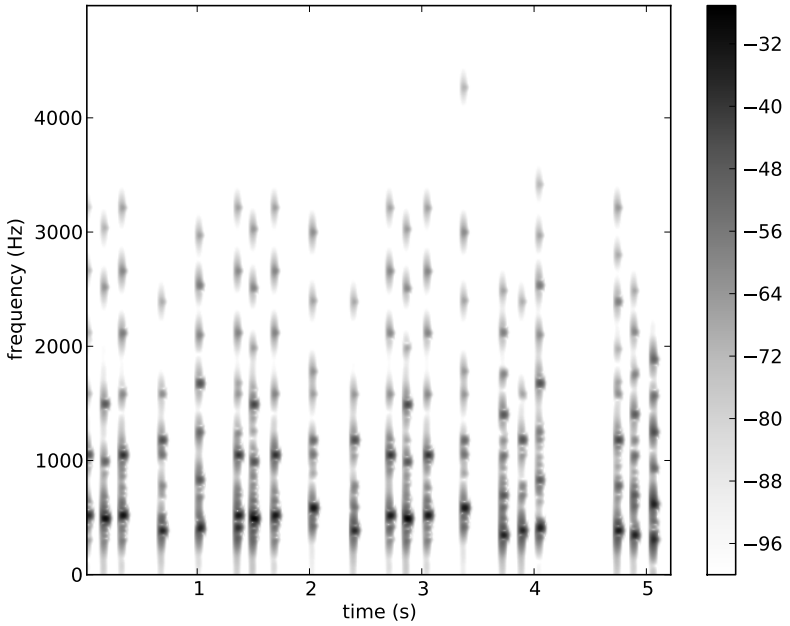


Fig. 5.21 FOF-only decomposition

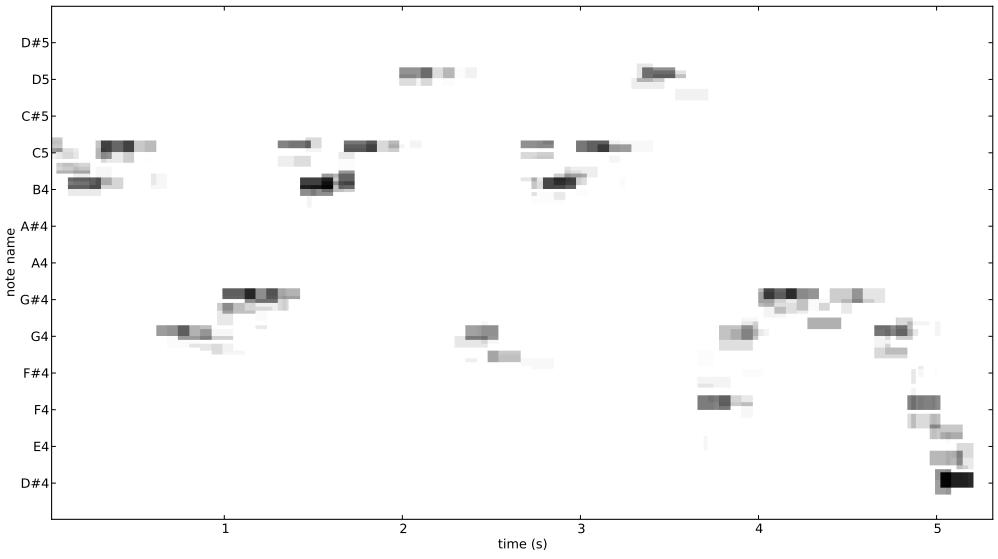


Fig. 5.22 Pseudo piano roll representation of BWV847 fugue subject mode (frequency resolution of 1/10 of a semitone)

to approximate quasi-harmonic structures, can benefit from a symbolic representation as it limits the scope of the search. Furthermore, considering this type of model from the perspective of musical symbols can yield an intuitive visualization of note-like regions while depicting signal characteristics such as energy and bandwidth.

5.3.3 Application to Polyphonic Musical Signal Analysis

Following the results of the previous experiment, we applied the use of a symbolic representation to the analysis of polyphonic audio. Preliminary tests showed that similar results to the previous section could be obtained. That is, with a sufficiently wide tolerance, the score-informed procedure could obtain a model of varying quality depending on parameters. So, we turned instead to a more specialized and difficult task: the separable modeling of individual parts in a musical signal[6].

In this test, we focused on BWV847 (measures 3 and 4 in Fig. 5.19) and the goal was to obtain a separable model of the transposed fugue subject and the counter subject. However, in this case we were not able to model each part separately with a high degree of quality since the region associated with the inharmonicity tolerance was either too wide, as to extract components of both parts in individual structures, or too narrow, as to exclude components that should be in a structure. However, we were able to obtain a separable low order model consisting of 1092 individual atoms, where the measurements associated with the model are an SRR of 16.14 and an ISD of 57.37×10^3. The piano roll visualization of this model is shown in Fig. 5.23.

As regards SpecIMP, we have determined two difficulties encountered in the task of part separation. First, is the presence of overlapping elements, notes (in the symbolic representation) and partials (in time-frequency). A related difficulty is the tuning of the inharmonicity tolerance such that the spectral components are assigned to the appropriate fundamental frequency, a problem if the tolerance is too high, and spectral components are not missed, a problem if the tolerance is too low.

[6]Here we use *part* to indicate a melodic line in a polyphonic monotimbral mixture

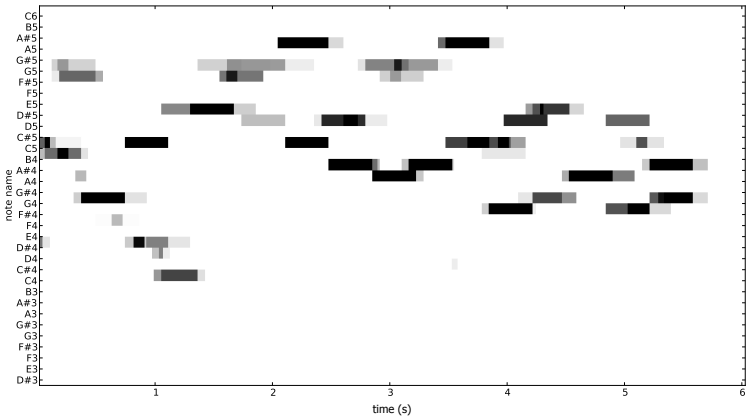


Fig. 5.23 Pseudo piano roll representation of BWV847 countersubject region

5.4 Agglomerative Clustering of Spectral Structures

In this section we apply agglomerative clustering stage to first-order structured models obtained by SpecIMP decomposition. In these techniques, we follow the procedure presented in [68]. However, in contrast to this study where the comparison occurred at the atomic level, we apply further grouping procedures to predetermined sets of atoms. Here, we identify two features of our approach that we believe to be advantageous in comparison to the atom-level technique. First, is a practical detail about implementation. This agglomerative family of techniques is built upon the computation (and storage) of N^2 (dis)similarity measurements[7], rendering them unsuitable for large problems. Though typically consisting of considerably fewer elements than a time-domain representation, a model obtained by a DBM can still be comprised of a sufficiently large number of atoms to render this computation impractical. On the other hand, comparison at an intermediate layer of the model can greatly reduce the computational load[8]. As regards what we consider to be the second overall advantage, we notice that the metastructural level describes elements that approach symbolic entities.

[7]Though, as mentioned in the next subsection, typically only half of these need to be computed.

[8]As an illustration, we refer to the difference between individual structures and components in the SpecIMP models in Fig. 5.3.

5.4.1 Agglomerative Algorithm

The two techniques described below are based on a single algorithmic foundation where the differentiating feature is the method of comparing structures. Here, we have implemented an algorithm that traverses a dissimilarity matrix and clusters structures deemed similar into metastructures, which is likely quite similar to the algorithm used in [68].

This algorithm is recursive over a binary dissimilarity matrix M where the lower triangle is identical to the upper triangle and/or not considered[9].

A metastructure is formed by a process of chaining together elementary structures. That is, for each element, each element deemed similar to it is added to the group, as is each element deemed similar to the second element and so on until no elements remain. It is important to note that an element appears only once in the resulting set of metastructures, i.e. each structure retains its uniqueness and a concatenation of the metastructures is equivalent to the initial model.

5.4.2 Note Region Metastructures

As mentioned in the previous section, in the context of a SpecIMP decomposition, a *note* is a symbolic entity that can be associated with a set of structures. Operating under the assumption that these structures are comprised of a number of quasi-harmonically related components above a fundamental frequency, we are able to agglomerate them according to a measure of dissimilarity. In the technique described here, we reduce each structure to a point in two-dimensional time-note space. We then compute the dissimilarity matrix as the Euclidean distance between each pair of structures. Optionally, a weighting coefficient can be applied to each parameter. An example of the result of this process is shown in 5.24.

Through the tuning of weight coefficients we are able to consider the neighbourhood around a cluster with a varying degree of tolerance for deviation in fundamental frequency and temporal location. For example, the metastructures have been constructed with a localized tolerance for a fundamental frequency within roughly a whole tone. As such, the recurring two and three note motives of the fugue subject are agglomerated into a relatively high-level construct. Applying more weight to the frequency parameter results in metastructures that resemble individual notes. Though we observe some distortion when notes overlap. Selected metastructures from each procedure are shown in Fig. 5.25.

[9] In the case of the ISD employed in Sec. 5.4.3 the measurement is asymmetrical.

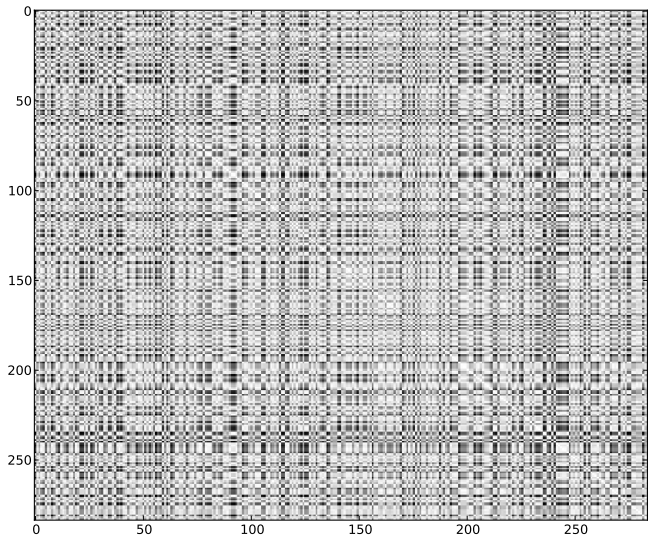


Fig. 5.24 Binary dissimilarity matrix of note-like properties in a BWV847 model

In this subsection we have demonstrated how the conjunction of a structured representation of audio and a symbolically-inspired agglomerative process can be used to obtain second-order structures exhibit note-level organization.

5.4.3 Metastructures Based on Spectral Similarity

In the literature, structured representations have often considered source as a fundamental identity, e.g. it is a criterion of a mid-level representation. In some cases, a source (in the causal sense) may not exist, e.g. electronic music where it has been suggested that a type of symbolic representation could be of use [64]. Alternatively, even if there is a causal source, perhaps an organization stemming from a more abstract description could be of interest, e.g. to the targeted application of audio effects.

In [68], the phase-invariant complex correlation coefficient is used to judge the similarity of atoms in the decomposition. Here, we agglomerate the spectral structures determined by a SpecIMP decomposition according to a perceptually-motivated measure of spectral similarity, the ISD. In order to compare this technique to that of the previous section, we begin with a model of BWV847. The associated distance matrix is shown in Fig. 5.26 and

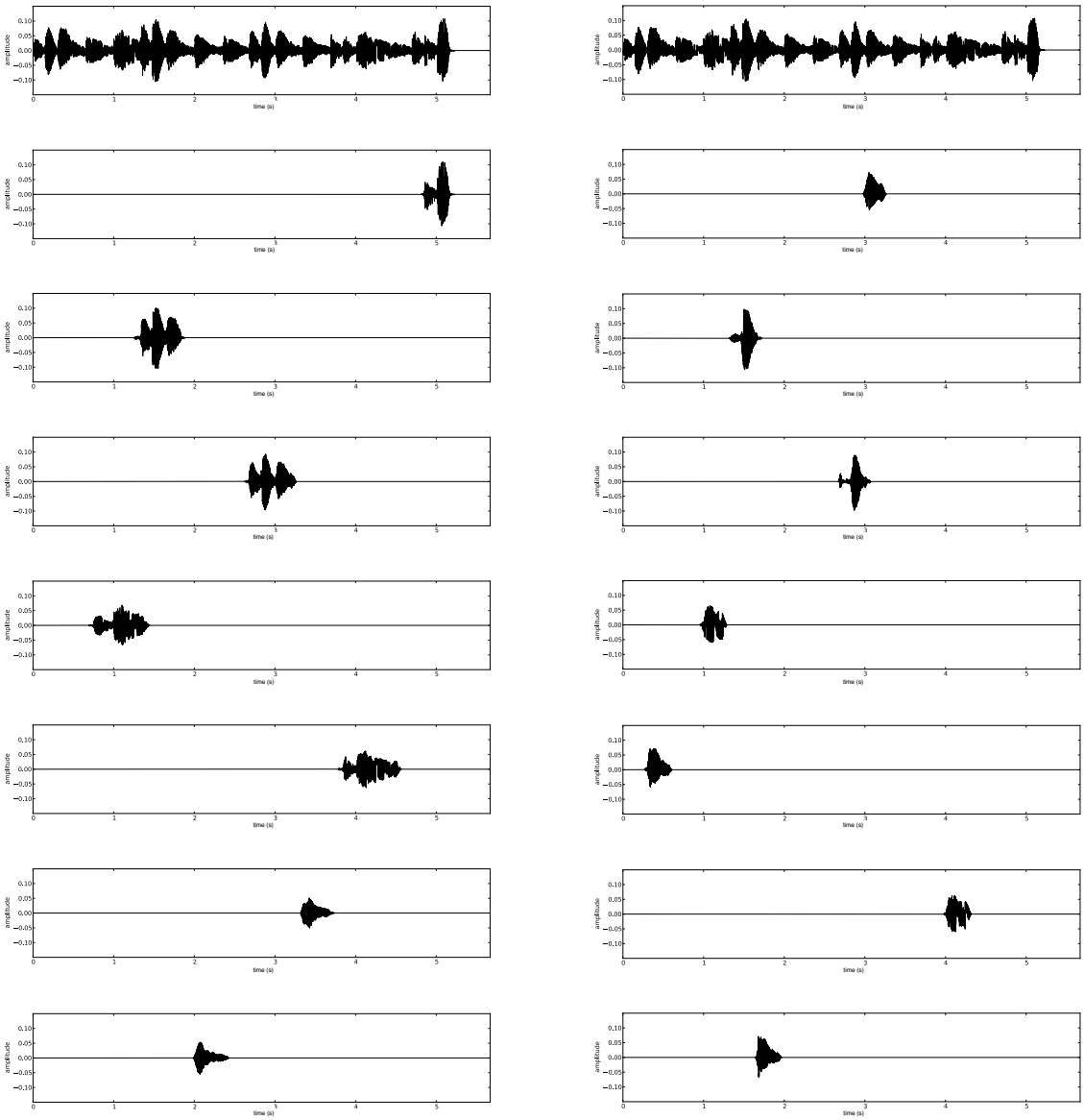


Fig. 5.25 Selected metastructures, where the agglomerative process associated with the left column targeted adjacent note groups and the process associated with the right column targeted individual notes. The first row shows the concatenation of all metastructures in the respective process, though they are identical, to illustrate the relationship between the extracted components and the model.

selected metastructures in Fig. 5.27.

A similar technique was applied to a segment of electroacoustic music, namely an excerpt from *Poème Électronique* by Varèse. Selected metastructures from this procedure are shown in

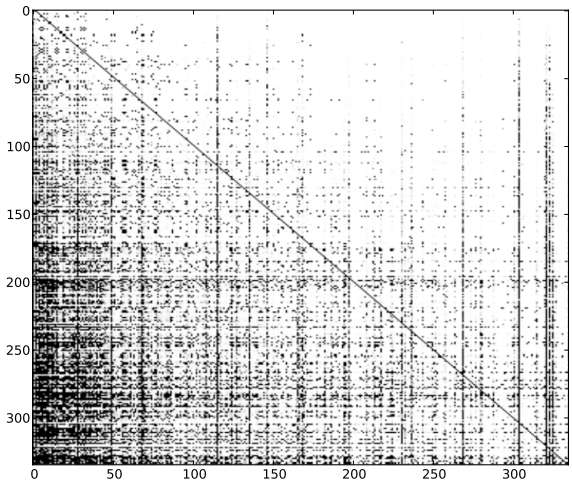


Fig. 5.26 Binary self-similarity matrix for a model of BWV847 (ISD $> 500 \times 10^3$)

The result of this procedure is a structured representation that is not tied to sound source or a specific note, but rather a 'transcription' based on spectral units. In terms of the examples produced here, we contrast the metastructures obtained from BWV847 to their counterparts in the previous section. Specifically. those determined by the measure of spectral similarity have a more synthetic character. This is owing to the fact that the structuring method points to the recurrence of similar spectra rather than the approximation of a symbolic entity whose definition is found outside of the system. Depending on the task at hand, we believe that both of these intermediate representations could be useful in sound signal processing. In our second example, there is little reference to constituent symbolic entities and our technique provides an invertible description of the organization of materials with similar spectra.

To underscore the relationship to the conceptual frameworks presented in Ch. 2 and our OO framework, each metastructure is represented in data as a new `SpectralBook` object that is the product of partitioning of the initial model. In this way, we construct a

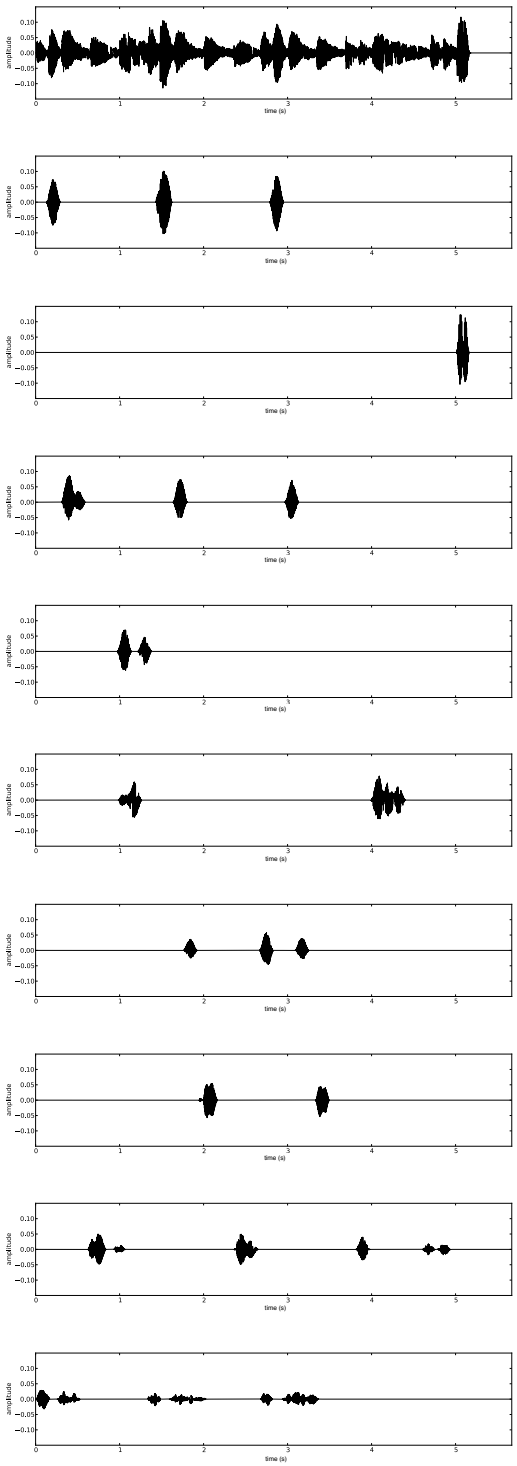


Fig. 5.27 Selected spectrally-similar metastructures obtained from a model of BWV847

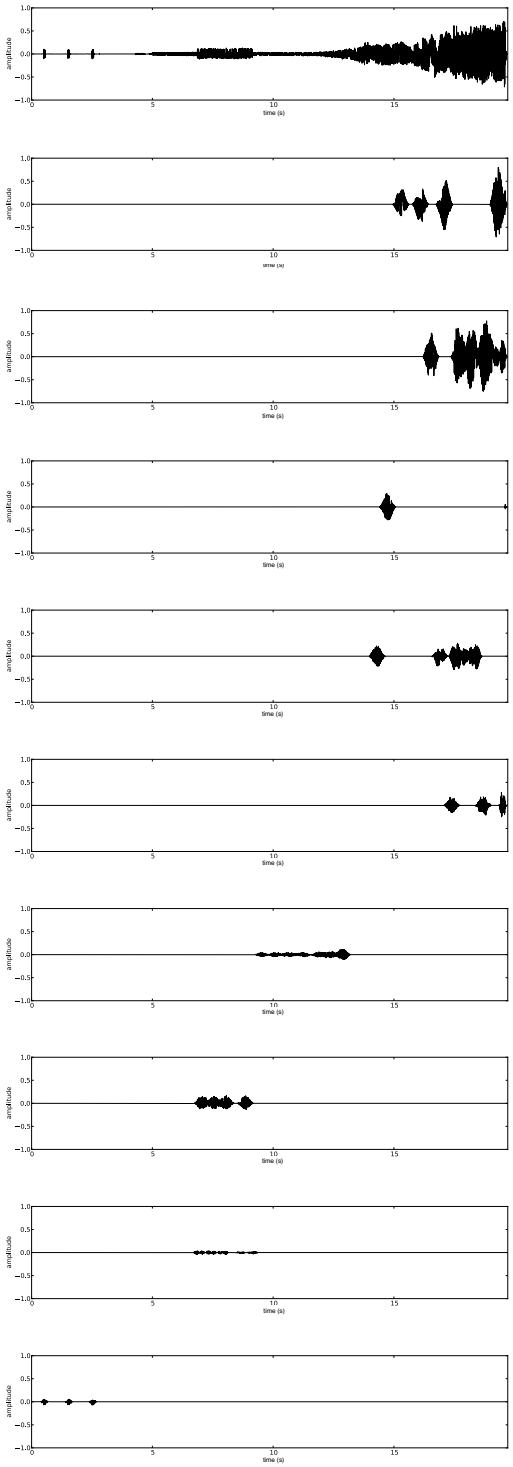


Fig. 5.28 Selected spectrally-similar metastructures obtained from a model of an excerpt from *Poème Électronique*

hierarchical representation that is in some sense analogous to a score, albeit less referential in terms of its constituent symbols.

5.5 Conclusion

In this chapter we have presented experiments concerned with structured and invertible representations of musical signals. Over the course of this study we have observed that an adaptive technique, based on a simple inductive principle, typically resulted in a superior performance by a perceptual measure of spectral similarity than a generic decomposition/-dictionary or a standard decomposition algorithm using a dictionary obtained by sampling the effective dictionary of the adaptive process. Furthermore, when combined with information about a musical context in score-like form, this approach to analysis was able to yield a structured intermediate representation in between a signal and a score. This feature was demonstrated through the development of a method of visualization that mixes representational modalities as well as the further agglomeration of structured models into sound objects that approach (or overlap with) a conventional symbolic representation of music. Moreover, an example was given where an excerpt of electro acoustic music, without clear causal features, was divided into classes of spectral units.

Chapter 6

Conclusion

In this thesis, we have approached the reconciliation of *symbolic* and *signal* interpretations of musical audio by customizing the stages of dictionary-based analysis/synthesis procedures. Further, we have provided an operational set of tools in an integrated environment called `pydbm`.

Through the consideration of two external *high-level* features, namely sound source and notated musical context, we have shown that a *top-down* approach to musical signal processing can yield a structured intermediate representations of sound based on separable objects. Moreover, each method suggests future avenues of inquiry.

We have demonstrated that a supervised strategy, built upon a general principle regarding the relatedness of simultaneous quasi-harmonically components and iteratively refinement of the effective dictionary, can be integrated into a Matching Pursuit-derived algorithm. In our experiments, this technique was consistently able to achieve a lower Itakura-Saito distance than a comparable generic technique.

An extension to this spectrally-inductive approach would further parameterize the *structure* rather than the *atom*. That is, in the approach reported here, we specify the parameters of an atom that we believe would be a viable *seed* from which to obtain a vertical structure. Other parameters that influence the character of the structure; specifically the inharmonicity tolerance, maximum number of spectral components, and minimum energy threshold are set as global variables of the pursuit. In examination of the results of the experiments presented here, and working with the software framework in general, we note that our approach was at times inflexible and could be generalized by specifying the pa-

rameters of the structure for each seed atom in the dictionary. For example, different structures could be specified using different inharmonicity tolerances. We believe that this could be particularly useful to specify structures oriented transients in the same pursuit as those oriented towards tonal elements. For example, in our experiments the effective modeling of transients typically called for a higher degree of inharmonicity tolerance and fewer components in comparison to tonal features. Further, we believe this extension to be conceptually appealing in that all stages of the technique would occur at a higher level of abstraction in comparison to a generic method of atomic decomposition as well as being a reasonable tradeoff between adaptive and static strategies.

From a set of spectrally-inductive pursuits we were able to define source-inspired dictionaries and instrument-specific analysis structures. Though not a perfect model in the perceptual sense, the latter of these was able to provide a separable and invertible model of sources in our simple mixtures. In this regard, we speculate that a technique built solely on spectral peak frequencies and amplitudes learned from a data base of examples will be insufficient to completely model a sound source in a meaningful and malleable way. Indeed, these structures describe but one facet of instrumental sounds and we imagine that they could be used in conjunction with other techniques, e.g. some source-specific description of noise profile.

One motivation behind a structured approach to analysis/synthesis was to facilitate abstract classification of separable objects in an audio signal. Stemming from a structured model, we employed two measurements of distance to agglomerate elements into metastructures that approach symbolic entities. Here, the first was more oriented towards a score-based representation and could be extended to consider other symbolic musical features, e.g. chords. In general, we believe that a symbolically-oriented post-processing procedure could be developed to re-evaluate the accuracy of a decomposition informed by a score and perhaps that this strategy could be employed to help the system in a polyphonic setting.

The second agglomerative approach made less reference to symbolic considerations, but rather constructed metastructures based on spectral similarity. This technique could be extended to consider temporal organization as well, i.e. to produce a time-localized measure of spectral similarity between structures. We believe such a model could be used to construct a state-transition network between spectral classes and provide a kind of invertible counterpart to the spectromorphological conceptual model, perhaps suitable for

a score-like visualization of electroacoustic music. Further, having determined metastructures, we believe that (spectral) similarity metrics could also be applied to that level of organization in order to add further levels to the hierarchy. That is, a hierarchical representation of the form: individual time-frequency atoms, spectral structures, metastructures, 'metametastructures', and so on until no further divisions can be made.

The development of these techniques suggests a symbolic method of manipulating separable sound objects. From the outset, it was our belief that a model of this type called for a similarly structured high-level interface in data. Toward this end we developed `pydbm` as an object-oriented framework for structured dictionary-based analysis/synthesis. In Sec. 2.4.3 we presented the FORMES project as an innovative computational approach to the control of sound synthesis[1]. Through the `Group` class (and its derived classes, i.e. dictionaries and books) in `pydbm` we have applied this methodology to analysis/synthesis, where the time-localized synthesis control parameters associated with a book approximate an analyzed sound. We note that, although this was not the goal of this thesis, `pydbm` could also be used for granular synthesis (not rooted in analysis), taking advantage of the structured and flexible approach to manipulating sets of atoms.

However, we believe the most interesting extensions to our framework would be a further development of the functionality and generality associated with the `Group` class.

As regards functionality, we imagine the utility of further operator overloading especially as dictionary-based audio effects are developed, e.g. pitch-shifting and time-stretching. In terms of generality, we are interested in a continuation of the formalized approach to Group structuring and conversion between hierarchical levels. For example, a *reductive* class method to 'flatten' a Group-derived object to a hierarchical level below returning a new, more generic object with associated class strategies for decomposition (or synthesis). Conversely, a *chunking* method to apply and shape hierarchical levels. Toward this end we envision dictionary and book metaclasses. These would serve as class factories able to define an Nth order hierarchical object at run time, redefining class methods as necessary. We believe that this approach would facilitate further structured agglomerative techniques, e.g. a nested process that continually spawns and re-organizes books of an arbitrary order until some breaking condition is met. In terms of analysis, perhaps a method of this character could also be applied to dictionary learning.

[1] We recall that FORMES provided an object-oriented and hierarchical control paradigm consisting of time-dependent parent and child *processes*.

In terms of integration into a musically-oriented software environment, the efforts reported in this thesis were exploratory. That is, through participation in a collaborative project, we have demonstrated that a framework founded on DBMs can be used in the context of computer-aided composition. Extensions in this area would provide more transparent access to analysis and synthesis constructs, e.g. facilitating algorithmic manipulation of the sound objects defined at various structural levels.

In short, our efforts constitute a relatively early exploration of the application of dictionary-based methods to the representation of musical audio. As these methods develop further, they are bound to suggest new approaches to the representation of sound objects. Conversely, musical sound as a rich and highly structured phenomena is likely to remain an interesting subject for analytical techniques, thus influencing their development and highlighting a symbiotic nature between the two disciplines.

Appendix A

Sound Examples

A website featuring audio examples of the material presented in this thesis is available at

`http://mt.music.mcgill.ca/~boyesg/thesis_examples/frontmatter.html`

To facilitate presentation, the sound examples have been divided according to the sections of the thesis text. These correspondences are detailed below.

4.3.2 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/4.3.2.html`
5.2.1 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.2.1.html`
5.2.3 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.2.3.html`
5.2.4 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.2.4.html`
5.3.2 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.3.2.html`
5.4.2 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.4.2.html`
5.4.3 : `http://mt.music.mcgill.ca/~boyesg/thesis_examples/5.4.3.html`

References

[1] Agon, C. *OpenMusic: Un langage visuel pour la composition assistée par ordinateur.* PhD thesis, University of Paris 6, 1998.

[2] Amatrian, X., Arumi, P., and Garcia, D. Clam: a framework for efficient and rapid development of cross-platform audio applications. In *Proc. of the 14th annual ACM international conference on Multimedia* (Santa Barbara, USA, October 2006).

[3] Amatrian, X., and Herrera, P. Transmitting audio content as sound objects. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio* (Espoo, Finland, 2002).

[4] Arfib, D., Keiler, F., and Zölzer, U. Time-frequency processing. In *DAFX : Digital Audio Effects*, U. Zölzer, Ed. John Wiley & Sons, Ltd, West Sussex, England, 2002, pp. 237–297.

[5] Boulanger, R., Ed. *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming.* MIT Press, Cambridge, USA, 2000.

[6] Bray, S., and Tzanetakis, G. Implicit patching for dataflow-based audio analysis and synthesis. In *Proc. of the International Computer Music Conference* (Barcelona, Spain, September 2005).

[7] Bregman, A. S. *Auditory Scene Analysis: The Perceptual Organization of Sound.* The MIT Press, 1994.

[8] Bresson, J. Sound processing in OpenMusic. In *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)* (Montréal, Canada, September 18-20 2006), pp. 325–330.

[9] Bresson, J. *La synthèse sonore en composition musicale assistée par ordinateur.* Phd dissertation, Université Paris IV - Pierre et Marie Curie, November 2007.

[10] Carabias-Orti, J. J., Vera-Candeas, P., Cañadas-Quesada, F. J., and Ruiz-Reyes, N. Music scene-adaptive harmonic dictionary for unsupervised note-event detection. *IEEE Trans. Audio, Speech, Lang. Process. 18*, 3 (March 2010), pp. 473–486.

[11] Chion, M. *Guide des objets sonores: Pierre Schaeffer et la recherche musicale*. Buchet Chastel, April 1994.

[12] Cho, N., and Kuo, J. Sparse music representation with source-specific dictionaries and its application to signal separation. *IEEE Trans. Audio, Speech, Lang. Process. 19*, 2 (February 2011), pp. 337–348.

[13] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.

[14] Cournapeau, D. "audiolab, a python package to make noise with numpy arrays". Internet, http://cournape.github.com/audiolab/, July 23, 2010 [August 13, 2011].

[15] Daudet, L. Sparse and structured decompositions of signals with the molecular matching pursuit. *IEEE Trans. Audio, Speech, Lang. Process. 14*, 5 (September 2006), pp. 1808–1816.

[16] Depalle, P., Garcia, G., and Rodet, X. Tracking of partials for additive sound synthesis using hidden markov models. In *Proc. of the International Computer Music Conference* (Tokyo, Japan, 1993), pp. 94–97.

[17] Depalle, P., and Poirot, G. Svp: A modular system for anlysis, processing and synthesis of sound signals. In *Proc. of the International Computer Music Conference* (Montréal, Canada, 1991).

[18] Dolson, M. The phase vocoder: A tutorial. *Computer Music Journal 10*, 4 (Winter 1986), pp. 14–27.

[19] Ellis, D., and Rosenthal, D. Mid-level representations for computational auditory scene analysis. In *Computational Auditory Scene Analysis*. Lawrence Erlbaum Associates, Inc., Mahwa, USA, 1998, pp. 257–272.

[20] Foote, J. Automatic audio segmentation using a measure of audio novelty. In *International Conference on Multimedia and Expo* (New York, USA, August 2000).

[21] Fournier, P. *Bach: 6 Suiten für Violoncello solo*. CD, Archiv Produktion, 1997.

[22] Gabor, D. Theory of communication. *J. Inst. Elect. Eng 93, Part III*, 26 (November 1946), pp. 429–457.

[23] Gabor, D. Acoustical quanta and the theory of hearing. *Nature 159*, 4044 (May 1947), pp. 591–594.

[24] Goodwin, M. Matching pursuit with damped sinusoids. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Munich, Germany, 1997), vol. 3, pp. 2037–2040.

[25] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. Rwc music database: Music genre database and musical instrument database. In *Proc. Int. Conf. Music Inf. Retrieval* (Washington, DC, October 2003), pp. 229–230.

[26] Gould, G. *The Glenn Gould Edition - Bach: The Well-Tempered Clavier, Book I.* CD, Sony, 1994.

[27] Gribonval, R. Fast matching pursuit with a multiscale dictionary of gaussian chirps. *IEEE Trans. Signal Process. 49*, 5 (May 2001), pp. 994–1001.

[28] Gribonval, R., and Bacry, E. Harmonic decomposition of audio signals with matching pursuit. *IEEE Trans. Signal Process. 51*, 1 (January 2003), pp. 101–111.

[29] Harris, F. J. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE 66*, 1 (January 1978), pp. 51–83.

[30] Kitahara, T. Mid-level representations of musical audio signals for music information retrieval. In *Advances in Music Information Retrieval*, Z. W. Ras and A. A. Wieczorkowska, Eds., vol. 274. Springer, 2010, pp. 65–91.

[31] Kovačević, J., and Chebira, A. An introduction to frames. *Foundations and Trends in Signal Processing 2*, 1 (2008), pp. 1–94.

[32] Krstulović, S., and Gribonval, R. MPTK: Matching pursuit made tractable. In *Proc. Int. Conf. Acoust., Speech. Signal Process (ICASSP)* (Toulouse, France, April 2006), vol. 3, pp. 496–499.

[33] Krstulović, S., Leveau, P., and Daudet, L. A comparison of two extensions of the matching pursuit algorithm for the harmonic decomposition of sounds. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (New Paltz, USA, October 2005).

[34] Leveau, P., Vincent, E., Richard, G., and Daudet, L. Instrument-specific harmonic atoms for mid-level music representation. *IEEE Trans. Audio, Speech, Lang. Process. 16*, 1 (January 2008), pp. 116–128.

[35] Lewicki, M. Efficient coding of natural sounds. *Nature Neuroscience 5*, 4 (2002), pp. 356–363.

[36] Mallat, S., and Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process. 41*, 12 (December 1993), pp. 3397–3415.

[37] Mathews, M. V. *The Technology of Computer Music*. MIT Press, Cambridge, USA, 1969.

[38] McAulay, R. J., and Quatieri, T. F. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Audio, Speech, Lang. Process. ASSP-34*, 4 (August 1986), pp. 744–754.

[39] McCartney, J. Rethinking the computer music language: Supercollider. *Computer Music Journal 26*, 4 (Winter 2002), pp. 61–68.

[40] Moguillansky, E. "gesellkammer/pysdif-github". Internet, https://github.com/gesellkammer/pysdif, August 9, 2011 [August 13, 2011].

[41] Opolko, F., and Walpnick, J. MUMS-McGill university master samples. Tech. rep., McGill University, 1987.

[42] Oppenheim, A. V., Schafer, R. W., and Buck, J. R. *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, New Jersey, USA, 1999.

[43] Pope, S. T., Ed. *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology*. MIT Press, Cambridge, USA, 1991.

[44] Portnoff, M. R. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Trans. Audio, Speech, Lang. Process. 24*, 3 (June 1976), pp. 243–248.

[45] Puckette, M. Combining event and signal processing in the MAX graphical programming environment. *Computer Music Journal 15*, 3 (Fall 1991), pp. 68–77.

[46] Rabiner, L., and Juang, B.-H. *Fundamentals of Speech Recognition*. PTR Prentice Hall, New Jersey, USA, 1993.

[47] Roads, C. Granular synthesis of sound. In *Foundations of Computer Music*, C. Roads, Ed. MIT Press, Cambridge, MA, 1985, pp. 145–159.

[48] Roads, C. *Microsound*. MIT Press, Cambridge, MA, 2001.

[49] Rodet, X. Time-domain formant wave function synthesis. *Computer Music Journal 8*, 3 (Autumn 1984), pp. 9–14.

[50] Rodet, X., and Cointe, P. Composition and scheduling of processes. *Computer Music Journal 8*, 3 (Autumn 1984), pp. 32–50.

[51] Rossignol, S., Rodet, X., Soumagne, J., Collette, J., and Depalle, P. Features extraction and automatic segmentation of acoustic signals. In *Proc. of the International Conference on Computer Music* (Ann Arbor, USA, 1998).

[52] Rubinstein, R., Bruckstein, A., and Elad, M. Dictionaries for sparse representation modeling. *Proceedings of the IEEE 98*, 6 (2010), pp. 1045 –1057.

[53] Sapp, C. "kernscore". Internet, http://kern.humdrum.net/, February 19, 2001 [August 13, 2011].

[54] Scaletti, C. The kyma/platypus computer music workstation. In *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology*, S. T. Pope, Ed. MIT Press, Cambridge, USA, 1991, pp. 119–134.

[55] Scaletti, C., and Hebel, K. An object-based representation for digital audio signals. In *Representations of Musical Signals*, G. De Poli, A. Picialli, and C. Roads, Eds. MIT Press, Cambridge, USA, 1991, pp. 372–389.

[56] Scaletti, C., and Johnson, R. E. An interactive environment for object-oriented music composition and sound synthesis. In *Proc. on Object-oriented programming systems, languages and applications* (San Diego, USA, 1988).

[57] Schaeffer, P. *Traité des objets musicaux: essai interdisciplines*. Seuil, Paris, 1966.

[58] Scheirer, E. The MPEG-4 structured audio standard. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Seattle, USA, May 1998), vol. 6, pp. 3801–3804.

[59] Schwarz, D. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine* (March 2007).

[60] Selesnick, I., and Schuller, G. The discrete fourier transform. In *The Transform and Data Compression Handbook*, K. R. Rao and P. Yip, Eds. CRC Press LLC, Boca Raton, 2001, pp. 57–98.

[61] Serra, X. *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. Phd dissertation, Stanford University, October 1989.

[62] Serra, X., and Smith, J. O. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal 14*, 4 (Winter 1990), pp. 14–24.

[63] Smalley, D. Spectro-morphology and structuring processes. In *The Language of Electroacoustic Music*, S. Emmerson, Ed. MacMillan, London, 1986, pp. 62–92.

[64] Smalley, D. Spectromorphology: explaining sound-shapes. *Organized Sound 2*, 2 (1997), pp. 107–126.

[65] Stroppa, M. Paradigms for the high-level musical control of digital signal processing. In *Proc. of the COST G-6 Conference on Digital Audio Effects* (Verona, Italy, December 2000), pp. 1–6.

[66] Sturm, B. Adaptive concatenative sound synthesis and its application to micromontage composition. *Computer Music Journal 30*, 4 (Winter 2006), pp. 46–66.

[67] Sturm, B., Roads, C., McCleran, A., and Shynk, J. Analysis, visualization, and transformation of audio signals using dictionary-based methods. *J. of New Music Research 38*, 4 (December 2009), pp. 325–341.

[68] Sturm, B., Shynk, J., and Gauglitz, S. Agglomerative clustering in sparse atomic decompositions of audio signals. In *Proc. Int. Conf. Acoust., Speech. Signal Process.* (Las Vegas, Nevada, April 2008), pp. 97–100.

[69] Thomson, P. Atoms and errors: towards a history and aesthetics of microsound. *Organized Sound 9*, 2 (2004), pp. 207–218.

[70] Thoresen, L., and Hedman, A. Spectromorphological analysis of sound objects: an adaptation of pierre schaeffer's typomorphology. *Organized Sound 12*, 2 (2007), pp. 129–141.

[71] Truax, B. Real-time granular synthesis with a digital signal processing computer. *Computer Music Journal 12*, 2 (1988), pp. 14–26.

[72] Various. "cython: C-extensions for python". Internet, http://cython.org/, August 5, 2011 [August 13, 2011].

[73] Verma, T., and Meng, T. Extending spectral modeling synthesis with transient modeling synthesis. *Computer Music Journal 24*, 2 (Summer 2000), pp. 47–59.

[74] Wang, G., and Cook, P. ChucK: A programming language for on-the-fly, real-time audio synthesis and multimedia. In *Proc. of the International Computer Music Conference* (Singapore, 2003).

[75] Wang, G., Fiebrink, R., and Cook, P. Combining analysis and synthesis in the ChucK programming language. In *Proc. of the International Computer Music Conference* (Copenhagen, Denmark, 2007).

[76] Wright, M., Chaudhary, A., Freed, A., Khoury, S., and Wessel, D. Audio applications of the sound description interchange format standard. In *Audio Engineering Society 107th Convention* (New York, USA, September 1999).

[77] Xenakis, I. *Formalized Music*, revised ed. Pendragon, Hillsdale, New York, 1992.