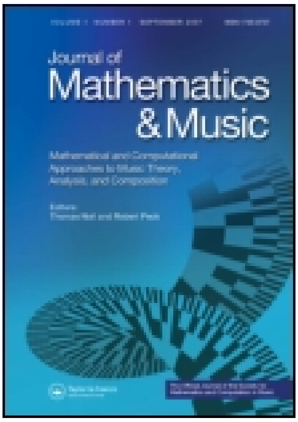


Journal of Mathematics and Music: Mathematical and Computational Approaches to Music Theory, Analysis, Composition and Performance
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tmam20

Subdivision schemes and multi-resolution modelling for automated music synthesis and analysis

Sigalit Hed [a] , Robert O. Gjerdingen [b] & David Levin [a]
[a] School of Mathematical Sciences , Tel Aviv University , Tel Aviv , Israel
[b] School of Music , Northwestern University , Chicago , IL , USA
Published online: 23 Apr 2012.

Journal of Mathematics and Music
Vol. 6, No. 1, March 2012, 17–47

Subdivision schemes and multi-resolution modelling for automated music synthesis and analysis

Sigalit Hed[a]*, Robert O. Gjerdingen[b] and David Levin[a]

[a]*School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel;* [b]*School of Music, Northwestern University, Chicago, IL, USA*

(*Received 19 May 2010; final version received 10 March 2012*)

Subdivision schemes are special multi-resolution analysis (MRA) methods that have become prevalent in computer-aided geometric design. This paper draws useful analogies between the mathematics of subdivision schemes and the hierarchical structures of music compositions. Based on these analogies, we propose new methods for music synthesis and analysis through MRA, which provide a different perspective on music composition, representation and analysis. We demonstrate that the structure and recursive nature of the recently proposed subdivision models [S. Hed and D. Levin, *Subdivision models for varying-resolution and generalized perturbations*, Int. J. Comput. Math. 88(17) (2011), pp. 3709–3749; S. Hed and D. Levin, *A 'subdivision regression' model for data analysis*, 2012, in preparation] are well suited to the synthesis and analysis of monophonic and polyphonic musical patterns, doubtless due in large part to the strongly hierarchical nature of traditional musical structures. The analysis methods demonstrated enable the compression and decompression (reconstruction) of selected musical pieces and derive useful features of the pieces, laying groundwork for music classification.

Keywords: algorithmic composition; music synthesis and analysis; subdivision schemes; computer-aided geometric design; multi-resolution analysis and synthesis; monophony; rhythmic patterns; polyphony; counterpoint; pitch-and-rhythm interrelationships

1. Introduction

1.1. *Background and motivation*

Hierarchy is a central concept in music theory. It can refer not only to idealizations of the interactions of similar musical patterns at different time scales or spans as in the work of Schenker [1] but also to traditional notions of the distinction between anchor pitches and their ornamental neighbours as taught in the crafts of counterpoint [2] and partimento [3].

In many studies, the hierarchical nature of music is either assumed or demonstrated. Cognitive models of music, such as the pioneering work of Lerdahl and Jackendoff [4, GTTM] or Longuet-Higgins and Lee [5], emphasize the perception of hierarchy, grouping and segmentation in musical artworks. Aspects of cognitive models have been harnessed in the area of music-information

*Corresponding author. Email: sigali6@post.tau.ac.il

ISSN 1745-9737 print/ISSN 1745-9745 online

http://dx.doi.org/10.1080/17459737.2012.680313
http://www.tandfonline.com

retrieval (MIR) by, for instance, Carter et al. Mathematical approaches that exhibit a fundamentally hierarchical nature, such as Zipf's [6] law, Mandelbrot's [7] fractals and Lindenmeyer's L-systems [8], have led to various methods for automatic music synthesis [9–11] and automatic music analysis and classification [12–15]. However, these methods do not explicitly address the hierarchical nature of cognitive models and musical schemata.

Our approach attempts to fill this gap by linking mathematical and musical hierarchies in order to achieve new music synthesis tools. Furthermore, these synthesis tools will serve as a basis for a new approach to music analysis. Our methods are based on subdivision schemes, which are refinement rules used traditionally for the generation of smooth curves and surfaces and for the construction of wavelets (see Section 1.2). Hed and Levin [16,17] proposed two new models generalizing the conventional subdivision operation and two inverted models for data analysis. In this paper, we show how these four models can be adapted for music synthesis and analysis, being applied in the time–pitch (time–frequency) domain. We also show that the intrinsic properties of subdivision schemes exhibit similarities to the hierarchical musical structures characteristic of monophony, polyphony, rhythm and polyrhythm, making the schemes suited for the generation and analysis of these kinds of musical patterns. Moreover, the synthesis and analysis methods proposed address pitch and rhythm simultaneously through a *unified* mechanism and hence consider their interrelationships. This mechanism suggests a new perspective on musical hierarchies and extends the notion of metric subdivision in music.

It is important to note that our methods deal with symbolic, rather than acoustic, music representations. Although notation does bear a close connection with the conceptions of performers, notated music can encode information not perceivable in the work as heard [18] and can, therefore, aid in the discovery and comparison of various patterns [19]. The multi-resolution analysis (MRA) methods proposed here for music *analysis*, being applied to notated music, are distinct from the existing MRA methods in music analysis applied to the audio signal (e.g. [14]). At the same time, the MRA methods proposed here for notated music *synthesis* are distinct from other MRA-related, notation-based works in algorithmic composition (e.g. [9]), as the latter may neglect pitch-and-rhythm interrelationships (although both pitch and rhythm may be treated) and generally do not make reference to advanced musical structures.

All in all, this paper makes several new contributions. Section 2 provides in general terms the principles and the rationale for applying the subdivision refinement models proposed in [16,17] to music synthesis and analysis. Section 3 reviews the models proposed in [16,17]. Based on these models, Section 4 proposes new tools for music synthesis and analysis. The music synthesis methods have their own value, but at the same time they provide a didactic platform that is important for conveying the methods for music analysis, representation and reconstruction. The music analysis methods demonstrated have value for the representation and reconstruction of given musical pieces as well as for their classification and retrieval. Section 5 briefly describes future research directions for both the synthesis and analysis methods. Finally, Section 6 provides a summary.

1.2. *Subdivision schemes*

Here, we briefly review subdivision scheme rudiments pertinent to our methods. These basics are required in Section 2 to convey the infrastructure and the musical rationale of our methods.

Subdivision schemes are recursive methods in computer-aided geometric design (CAGD) and approximation theory (see [20, and references therein, 21]), closely related to the MRA [22, 23]. These schemes are used to *generate* and represent smooth curves and surfaces and for the construction of wavelets, by an iterative process of refinement. Here, we discuss *binary* subdivision schemes (BSS) and later mention the generalization for any subdivision number.

Here, we focus on *scalar* subdivision schemes that generate functions on $\mathbb{R}$.

Given a set of values on the integers $\boldsymbol{f}^0 = \{f_i^0\}_{i\in\mathbb{Z}} \subset \mathbb{R}$, a BSS operator S recursively generates values $\boldsymbol{f}^k = \{f_i^k\}_{i\in\mathbb{Z}} \subset \mathbb{R}$ at refined resolution levels $k \geq 0$ on the grid[1] $\{2^{-k}i\}_{i\in\mathbb{Z}}$. The scheme S refines $\boldsymbol{f}^k$, yielding a new set of values, $\boldsymbol{f}^{k+1} = S\boldsymbol{f}^k$, on a doubly finer grid, $\{2^{-(k+1)}i\}_{i\in\mathbb{Z}}$. The initial set of values, $\boldsymbol{f}^0$, are the 'control points' and recursively $\{\boldsymbol{f}^k = S^k\boldsymbol{f}^0\}_{k=1}^{\infty}$. An *interpolatory* subdivision scheme (ISS) additionally satisfies $f_i^k = f_{2i}^{k+1}, i \in \mathbb{Z}, k \in \mathbb{Z}_+$. Denote by $L\boldsymbol{f}^k$ the piecewise linear interpolant of the values $\boldsymbol{f}^k$. The uniform limit of the sequence $f = S_{\boldsymbol{p}}^{\infty}\boldsymbol{f}^0 = \lim_{k\to\infty} L\boldsymbol{f}^k$, if it exists, is the function generated by the scheme.

In the case of a linear subdivision operator, the scheme hereby, denoted by $S_{\boldsymbol{p}}$, is represented by a mask $\boldsymbol{p} = \{p_j : j \in \mathbb{Z}\} \subseteq \mathbb{R}$ and the operator is equivalently represented by the matrix $\mathbf{P} = \{p_{i,j} : i,j \in \mathbb{Z}\} \subseteq \mathbb{R}$, s.t $p_{i,j} = p_{i-2j}$.[2] The refinement is defined through the mask or the matrix by

$$f_i^{k+1} = (S_{\boldsymbol{p}}\boldsymbol{f}^k)_i = \sum_{j\in\mathbb{Z}} p_{i-2j} f_j^k, \tag{1a}$$

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k. \tag{1b}$$

Using the following notations, one can simplify Equation (1a). Denote by $[\boldsymbol{v}]^{\uparrow}$ the up-sampling-by-2 of the vector $\boldsymbol{v}$[3] and by $[\boldsymbol{v}]_r^{\downarrow} \equiv \{v_{2j+r}, j \in \mathbb{Z}\}$, $r = 0, 1$, the r-shifted down-sampling-by-2 of the vector $\boldsymbol{v}$.[4] The operation (1a) then has the alternative formulation:

$$\boldsymbol{f}^{k+1} = S_{\boldsymbol{p}}\boldsymbol{f}^k = \boldsymbol{p} \star [\boldsymbol{f}^k]^{\uparrow}. \tag{2}$$

$\boldsymbol{p} \star [\boldsymbol{f}^k]^{\uparrow}$ is the convolution of $\boldsymbol{p}$ with the data $\boldsymbol{f}^k$ after up-sampling-by-2. Equation (2) can be split into two convolution equations, by splitting $\boldsymbol{p}$ into two interlaced 'rules' (filters), $\boldsymbol{p}_{\{r\}}$:

$$\boldsymbol{f}^{k+1,r} = \boldsymbol{p}_{\{r\}} \star \boldsymbol{f}^k, \quad r = 0, 1, \quad \text{where } \boldsymbol{p}_{\{r\}} \equiv [\boldsymbol{p}]_r^{\downarrow},\ \boldsymbol{f}^{k+1,r} \equiv [\boldsymbol{f}^{k+1}]_r^{\downarrow}. \tag{3}$$

The two rules $\boldsymbol{p}_{\{0\}}$ and $\boldsymbol{p}_{\{1\}}$, which are responsible for a single refinement iteration, operate as follows: rule $\boldsymbol{p}_{\{0\}}$ transfers the values on the kth grid, $\{2^{-k}i\}_{i\in\mathbb{Z}}$, to those new values on the $(k+1)$th grid corresponding to the kth grid, $\{2^{-(k+1)}2i\}_{i\in\mathbb{Z}}$; rule $p_{\{1\}}$ transfers them to those new values on the $(k+1)$th grid corresponding to the in-between grid points of the kth grid, $\{2^{-(k+1)}(2i+1)\}_{i\in\mathbb{Z}}$. Subdivision schemes with general division number d (e.g. ternary) are described by d (e.g. three) rules by an appropriate extension of Equation (3) and the grid subdivisions.

Remark 1.1 For interpolatory scheme (with general division number), $\boldsymbol{p}_{\{0\}} = \boldsymbol{\delta} = \{\delta_{i,0}\}_{i\in\mathbb{Z}}$.

The L-iterated operator, $S_{\boldsymbol{p}}^L$, satisfies $\boldsymbol{f}^L = S_{\boldsymbol{p}}^L\boldsymbol{f}^0$ with the matrix $\mathbf{P}^{(L)} \equiv \mathbf{P}^L$ s.t. $\boldsymbol{f}^L = \mathbf{P}^{(L)}\boldsymbol{f}^0$. This operator can be represented by the L-iterated mask $\boldsymbol{p}^{(L)} = \{p_j^{(L)} : j \in \mathbb{Z}\} \subseteq \mathbb{R}$ s.t. $\boldsymbol{p}^{(L)} = \boldsymbol{p} \star [\boldsymbol{p}^{(L-1)}]^{\uparrow}$. The L-iterated mask is defined by an extension of Equation (2) and is composed of 2^L rules that extend Equation (3), as explained in Section 3.1.2.

The scheme $S_{\boldsymbol{p}}$ is said to be C^nSS if the limit function, $f = S_{\boldsymbol{p}}^{\infty}\boldsymbol{f}^0$, is C^n (has n smooth derivatives). The necessary and sufficient conditions for C^nSS have been discussed, for instance, in [20,24]. The relation of subdivision schemes and fractals has been discussed in [25].

In an *interpolatory* scheme, $f_i^k = f(2^{-k}i), i \in \mathbb{Z}$. In a *non-stationary* scheme, the mask is dependent on the iteration, that is, $\mathfrak{p} = \{\boldsymbol{p}_k\}_{k\in\mathbb{Z}_+}$ [20], and the L-iterated operator is then $S_{\mathfrak{p}}^{(L,k)}$, with the matrix $\mathbf{P}^{(L,k)}$ s.t. $\boldsymbol{f}^{k+L} = \mathbf{P}^{(L,k)}\boldsymbol{f}^k$. In a *non-uniform* scheme, the mask is additionally dependent on the location, that is, $\{\boldsymbol{p}_{(k,j)}\}, k \in \mathbb{Z}_+, j \in \mathbb{Z}$ [20]. A *symmetric* scheme $S_{\boldsymbol{p}}$ satisfies $p_j = p_{-j},\ \forall j \in \mathbb{Z}$. A *perturbed* scheme has a *tension* parameter that controls the smoothness of the scheme [24]. Examples are the four-point C^1SS interpolatory scheme

$\boldsymbol{p}(\omega) = \{-\omega, 0, \frac{1}{2}+\omega, 1, \frac{1}{2}+\omega, 0, -\omega\}$ [21], which perturbs the linear interpolation scheme $\boldsymbol{p} = \{\frac{1}{2}, 1, \frac{1}{2}\}$, or the six-point C^3SS interpolatory scheme $\boldsymbol{p}(\theta) = \{\theta, 0, \frac{-1}{16} - 3\theta, 0, \frac{9}{16} + 2\theta, 1, \frac{9}{16} + 2\theta, 0, \frac{-1}{16} - 3\theta, 0, \theta\}$ [26], which perturbs the four-point scheme with $\omega = \frac{1}{16}$.

We concentrate on masks with finite support $\sigma = [m_1, m_2]$, that is, $\boldsymbol{p} = \{p_j\}_{j=m_1}^{m_2} \subseteq \mathbb{R}$, in which case the matrix $\mathbf{P}$ can be replaced by a finite bisection.

Recall the relation (1b), $\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k$. The extension of the subdivision operation and its reverse operation to the MRA [27] is accordingly

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k + \mathbf{Q}\boldsymbol{d}^k \quad \text{reconstruction (synthesis),} \tag{4}$$

$$\left.\begin{array}{ll} (a) & \boldsymbol{f}^k = \mathbf{P}^\dagger \boldsymbol{f}^{k+1}, \\ (b) & \boldsymbol{d}^k = \mathbf{Q}^\dagger \boldsymbol{f}^{k+1}. \end{array}\right\} \text{orthogonal decomposition.} \tag{5}$$

$\mathbf{M}^\dagger$ denotes the Moore–Penrose pseudo-inverse of the matrix $\mathbf{M}$, $\boldsymbol{f}^k$ (low resolution) is a least-squares solution to Equation (1b) with minimum norm and $\boldsymbol{d}^k$ (high resolution) is the differences or the least-squares error vector. The operators $\mathbf{P}$ and $\mathbf{Q}$ are the matrix representations for the MRA filters corresponding to the *scale* and *wavelet* functions [23], respectively.

2. Idea and rationale

Before getting into more details, in this section, we introduce the idea in general terms. Generally speaking, in this paper, we explore the associations between the reconstruction (4) and the composition of notated music and those between some modification of decomposition (5) and the analysis of notated music. We relate to some recently developed generalizations [16,17] of the subdivision operation that yield fine points in different resolutions (synthesis) or that extract the scheme $S_{\boldsymbol{p}}$ from given fine and coarse points (analysis). Essentially, we show that these 'geometrical' subdivision models lend themselves specifically to music generation, and eventually to music analysis, by assigning the time dimension to the subdivision's free parameter (x-axis) and assigning pitch, time[5] or other musical dimensions to the value (y-axis).

Recall that a subdivision (forth) operation $S_{\boldsymbol{p}}^L$ is the mapping $\mathbf{P}^{(L,k)} : \boldsymbol{f}^k \to \boldsymbol{f}^{k+L}$, from level k to level $k+L$, and its reverse (back) operation maps from level $k+L$ to level k. In the case of the existence of displacements, $\mathbf{Q}\boldsymbol{d}^k$, the operation and its reverse are L consecutive operations of Equations (4) and (5), respectively. The music synthesis proposed in this paper, in its basic mode, uses the forth operation: it renders some given coarse notes $\boldsymbol{f}^k$ by operating a given scheme $\mathbf{P}^{(L,k)}$ to yield a refined musical piece $\boldsymbol{f}^{k+L}$. However, the music analysis described in this paper does *not* use the back operation (which recovers $\boldsymbol{f}^k$). Instead, the music analysis, in its basic mode, extracts the scheme $\mathbf{P}^{(L,k)}$ from some given coarse notes $\boldsymbol{f}^k$ and fine notes $\boldsymbol{f}^{k+L}$ and uses it for both music analysis and reconstruction. In the advanced mode of both music synthesis and analysis, the underlying scheme and fine notes are of varying resolutions (not a constant L) and include perturbations, using the models proposed in [16,17].

Example 2.1 Figure 1 shows a preliminary example of the musical input and output of the synthesis and analysis before the incorporation of varying resolutions (small part of Figure 13 in Example 4.11). This example encompasses both the synthesis and analysis operations because it shows the reconstruction (synthesis) of a given piece based on its analysis. The scheme (masks) extracted by *analysing* the original piece is applied on the coarse control notes to *synthesize* the reconstruction (approximation) of the piece. In Example 4.11, we present the specific scheme being used by the synthesis or extracted by the analysis and more details on these operations. We also discuss the potential of the scheme extraction not only for reconstruction, but also for

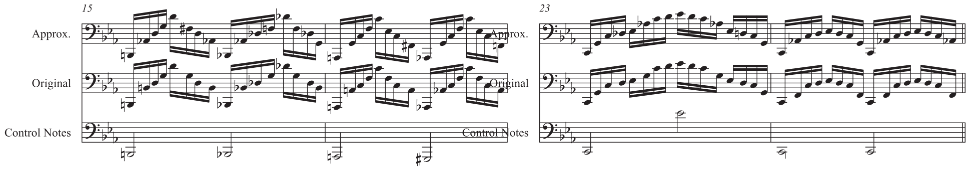


Figure 1. A preliminary example for music analysis and synthesis: reconstructing Chopin, Op. 10, No. 12, 'Revolution', bass part, Bars 15, 23. Middle staves: original piece; bottom staves: coarse control notes; upper staves: reconstructed (approximated) piece (Audio13_1, Audio13_2, Audio13_3). The reconstruction is *synthesized* by operating the masks, extracted by *analysing* the original, on the coarse notes. The full reconstruction is shown in Figure 13.

music analysis and classification. More generally, the discussions throughout the paper regarding the synthesis precede those regarding the analysis and reconstruction because the synthesis is needed to convey the analysis and both synthesis and analysis are required for the reconstruction.

As will be shown, this assignment of musical values into the geometrical models yields analogies between subdivision schemes and musical structures, emphasizing note approximation, perturbations, permutations, pitch-and-rhythm interrelationships and polyphony, giving rise, in turn, to additional features such as varying tempo, compound meters, accentuation and unexpectedness. In the following, we first emphasize the relevance of some of these musical operations and structures to rationalize the conversion of the geometric models proposed in [16,17] to different musical domains. These geometric subdivision models are reviewed in Section 3. Their conversion to the musical domain and the resulting procedures for music synthesis and analysis are elaborated in Sections 4.1 and 4.2.

2.1. *Note approximations*

Note approximation, in general, preserves a basic property of traditional melodies, according to which small melodic intervals (normally steps) are more prevalent than large ones (skips, etc.). This is reinforced by perceptual findings, such as the first finding of van Noorden [28] (discussed later). In addition, the input for the approximation allows a simple control over the final result. In particular, the masking mechanism of the subdivision schemes provides a means to 'nonlinear composition', in which the current musical segment is generated (approximated) not only according to the preceding notes (its 'past'), but also according to the constraints on the following notes (its 'future').

Such approximations, with various degrees of complexity, are generated in Section 4.1 (synthesis) and are assumed in Section 4.2 (analysis).

2.2. *Perturbations*

As will be described in Section 3, the 'generalized perturbed schemes' synthesis model and its inversion, the 'subdivision regression' analysis model, represent a scheme's mask by a linear combination of 'template masks'. This approach allows representing a subdivision as an MRA reconstruction (4) and decomposition (5) that omit the outer differences $\boldsymbol{d}^k$ and depend only on the initial data $\boldsymbol{f}^k$. Musical data appear to fit this model. That is, music could be a result of a smooth subdivision that introduces deviations or noise expressible in terms of the initial data, much as rapid movements in a Baroque melody were described as 'diminutions' of a slower, simpler melodic template. This analogy is reinforced by the L-systems approach [8], the fractals–subdivision relations [25] and the evidence for fractals in music ($P(f) = 1/f$ in [12] and MIR in [15]). The examples given in Section 4.1 that include perturbations (e.g. Procedure 2) and the smoothness-and-noise decomposition in our music analysis methods in Section 4.2 appear to reinforce the suitability of these generalized perturbation models for music generation and analysis.

2.3. *Pitch-and-rhythm interrelationships model*

Using the generalizations of the subdivision operation in the musical context yields certain correlations between pitch and rhythm. These correlations are based on a correspondence that we propose between the rhythm tree and the 'subdivision rules tree', as follows.

The rhythm tree, on the one hand, is a pruning of the tree representing the meter, as shown in Figure 2. Longuet-Higgins and Lee [5] explained several phenomena of rhythm cognition through the rhythm tree while neglecting tonal and rhythmic relationships.[6] The rhythm tree has also been used for melodic comparisons in [29] and for rhythm generation in [9]. The latter work further used fractals (a sort of perturbation) and interpolations (a sort of approximation) to generate the pitches of the piece, independently of the rhythm.

The 'subdivision rules tree', on the other hand, is a tree organization of the subdivision rules describing the L-iterated operators, as described in Section 3 in more detail.

Equipped with these, we suggest a parallelism between the rhythm tree and the 'subdivision rules tree' (or its permutations) or an analogy between beat subdivision and subdivision schemes. As will be shown, this tree analogy allows unifying the above three musical mechanisms – rhythm trees, note approximations and note perturbations – to deal with rhythm and pitch simultaneously and to handle pitch-and-rhythm interrelationships.

The correspondence between the rhythm tree and the 'subdivision rules tree' is achieved by transferring the 'varying-resolution' synthesis model and the 'tree regression' analysis model, which rely on the 'subdivision rules tree', from a geometric space into a musical space. On top of that, perturbations are incorporated by the 'generalized perturbed schemes' synthesis model and 'subdivision regression' analysis model, as mentioned in Section 2.2.

As will be explained in Section 3, the 'varying-resolution' synthesis model and the 'tree regression' analysis model represent the subdivision operation and its reverse by the pair [*masks, tree*], where the 'tree' specifies a pruning of the 'subdivision rules tree', to achieve varying levels of refinement resolutions. Nonetheless, the rhythm tree may itself be considered to be a pruned (meter) tree. We, therefore, correlate the rules tree with the rhythm tree by applying the [*masks, tree*] representation in the musical context. This approach leads to music synthesis and analysis tools that control or extract pitch-and-rhythm interrelationships and give a new perspective

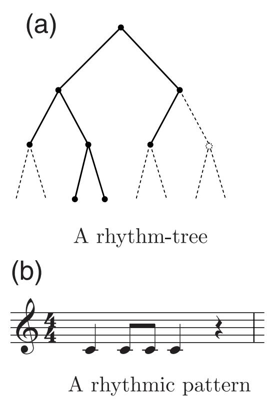


Figure 2. Rhythm tree [5]: the pruned tree (continuous line) in (a) represents the rhythm in (b). Its terminal nodes (full circles) represent the played notes. The rests (hollow circles) are terminal nodes of the extension of the pruned tree to an admissible tree.

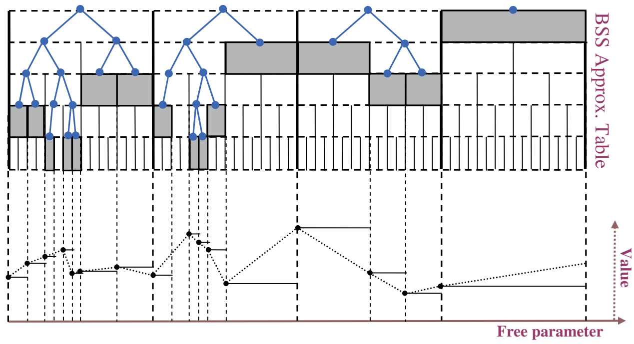


Figure 3. Use of the models 'varying-resolution' [16] and 'tree regression' [17] for melody refinement and analysis, respectively. Top: A 'full approximation table' of depth $L = 4$, exemplified by rectangles proportional to $2^{-l}, l = 1, \dots, L$, and an example for varying-resolution traverse (grey rectangles) confined by the leaves of the trees (blue). The coarse points $\boldsymbol{f}^k$ are represented by the top (widest) rectangles. The grey rectangles represent the resolution and shift (l, r) of the approximation values $\boldsymbol{f}^{k+l,r}$ corresponding to the leaves. Bottom: The melody composed of all refined values $\boldsymbol{f}^{k+l,r}$ for all possible shifts r (black dots) as achieved by Equation (11). Synthesis: The pitches are defined by the masks and the trees (of the pitch dimension), and the rhythm is defined by the (masks and) trees of the time dimension. Analysis: The masks are defined by the coarse values and the tree in each dimension. As a special case, the tree is the rhythm tree. (colour online only)

to musical hierarchy and to some *perceptual* findings (e.g. the second finding of van Noorden [28] discussed later). The method is depicted schematically in Figure 3, showing the leaves of the pruned trees selecting data from the different refinement levels. Sections 4.1 and 4.2 provide details on this approach to music synthesis and analysis and more explanations in Figure 3.

To understand these music–mathematics analogies, we first review the mathematical models.

3. Subdivision-based models for data synthesis and analysis: review

As mentioned above, our infrastructure and musical motivation are based on the geometric synthesis and analysis models proposed in [16,17]. For self-containedness purposes, we review these models below. These models are harnessed later in this paper to music synthesis and analysis.

3.1. *Subdivision-based models for data synthesis*

In the following, we review the synthesis models proposed in [16]. These models generalize the subdivision operation and can yield non-smooth schemes resembling the original schemes in a controllable way.

3.1.1. *The 'generalized perturbed schemes' model*

The 'generalized perturbed schemes' model represents a scheme's mask by a linear combination of predetermined 'template masks' $\{\boldsymbol{\phi}^{[j]}\}_{j=1}^m$, by a 'tension vector parameter' $\boldsymbol{\omega}$. The corresponding 'template matrices' $\{\boldsymbol{\Phi}^{[j]}\}_{j=1}^m$ span by the same $\boldsymbol{\omega}$ the subdivision matrix $\mathbf{P}$. These relations are formulated by

$$\boldsymbol{p} \equiv \boldsymbol{p}(\boldsymbol{\omega}) = \sum_{j=1}^{m} \omega_j \boldsymbol{\phi}^{[j]}, \tag{6a}$$

$$\mathbf{P} \equiv \mathbf{P}(\boldsymbol{\omega}) = \sum_{j=1}^{m} \omega_j \boldsymbol{\Phi}^{[j]}. \tag{6b}$$

Let $S_{\boldsymbol{p}}$ and $S_{\boldsymbol{q}}$ be two subdivision schemes, such that $S_{\boldsymbol{p}}$ is C^nSS for some $n > 0$. Let $\mathbf{P}$ and $\mathbf{Q}$ be the corresponding subdivision matrices. The subdivision operation of $S_{\boldsymbol{p}+\boldsymbol{q}}$ can then be written as

$$\boldsymbol{f}^{k+1} = \tilde{\mathbf{P}}\boldsymbol{f}^k = \mathbf{P}\boldsymbol{f}^k + \mathbf{Q}\boldsymbol{f}^k. \tag{7}$$

Equation (7) is then an MRA reconstruction (4) that avoids the outer differences $\boldsymbol{d}^k$.

The following two examples will also be used later in the paper.

Example 3.1

$$\boldsymbol{\phi}^{[1]} = \{\tfrac{1}{2}, 1, \tfrac{1}{2}\}, \quad \text{the linear interpolation scheme,} \tag{8a}$$

$$\boldsymbol{\phi}^{[2]} = \{-1, 0, 1, 0, 1, 0, -1\}, \quad \text{four-point perturbation mask,} \tag{8b}$$

$$\boldsymbol{\phi}^{[3]} = \{1, 0, 0, 0, 0, 0, 1\}, \quad \text{a perturbation mask.} \tag{8c}$$

$\boldsymbol{p} \equiv \boldsymbol{p}(\boldsymbol{\omega_p}) = \sum_{j=1}^{2} \omega_j \boldsymbol{\phi}^{[j]}$ with $\boldsymbol{\omega_p} = \{1, \omega_2\}$ is the four-point scheme. To comply with Equation (7), one may pick ω_2 in the smoothness range of the four-point scheme, and $\boldsymbol{q} \equiv \boldsymbol{q}(\boldsymbol{\omega_q}) = \omega_3 \boldsymbol{\phi}^{[3]}$.

Remark 3.1 Using $\boldsymbol{p}(\boldsymbol{\omega}) = \sum_{j=1}^{3} \omega_j \boldsymbol{\phi}^{[j]}$, $\boldsymbol{\omega}_p = \{1, \omega_2, \omega_3\}$, with the 'template masks' in Equation (8) and $\omega_2 \neq 0$ or $\omega_3 \neq 0$ yields a disturbance to the odd rule $\boldsymbol{p}_{\{1\}}$ and so, according to Remark 1.1, the corresponding $\boldsymbol{p}$ remains interpolatory. The perturbed and unperturbed subdivision results will, therefore, share control points (notes).

Note: The 'four-six-point' scheme proposed in [16] is built using $\boldsymbol{\phi}^{[1]}$ and $\boldsymbol{\phi}^{[2]}$ in Equation (8) and $\boldsymbol{\phi}^{[3]} = \{1, 0, -3, 0, 2, 0, 2, 0, -3, 0, 1\}$.

Example 3.2 The following 'template masks' set is used later in the paper:

$$\boldsymbol{\phi}^{[1]} = \{\tfrac{1}{3}, \tfrac{2}{3}, 1, \tfrac{2}{3}, \tfrac{1}{3}\}, \quad \text{the ternary linear interpolation scheme,} \tag{9a}$$

$$\boldsymbol{\phi}^{[2]} = \{1, 0, 0, 0, 0, 0, 1\}, \quad \text{a perturbation mask.} \tag{9b}$$

Remark 3.2 Using $\boldsymbol{p}(\boldsymbol{\omega}) = \sum_{j=1}^{2} \omega_j \boldsymbol{\phi}^{[j]}$, $\boldsymbol{\omega}_p = \{1, \omega_2\}$, with the 'template masks' in Equation (9) and $\omega_2 \neq 0$ yields a disturbance to $\boldsymbol{p}_{\{0\}}$ and so, according to Remark 1.1, the corresponding $\boldsymbol{p}$ is a non-interpolatory scheme, and hence the control points (notes) are not generally part of the subdivision result.

3.1.2. *The 'varying-resolution' model*

The 'primal rules tree': Denote by $[\boldsymbol{v}]_L^{\uparrow}$ the up-sampling-by-2^L of $\boldsymbol{v}$ and by $[\boldsymbol{v}]\downarrow_{(L,r)} \equiv \{v_{j2^L+r}, j \in \mathbb{Z}\}$, $0 \leq r < 2^L$, the r-shifted down-sampling-by-2^L of $\boldsymbol{v}$.[7] Given the L-iterated operator $S_{\boldsymbol{p}}^L$, the following extends Equation (2) from one iteration to any number of iterations L:

$$\boldsymbol{f}^{k+L} = S_{\boldsymbol{p}}^L \boldsymbol{f}^k = \boldsymbol{p}^{(L)} \star [\boldsymbol{f}^k]_L^{\uparrow}. \tag{10}$$

$\boldsymbol{f}^k$ are the coarse values and $\boldsymbol{f}^{k+L}$ are the fine values at level $k + L$, achieved by L iterations of $S_{\boldsymbol{p}}$. The terms 'coarse points' and 'fine points' relate to the coarse and the fine values, respectively, coupled with their parameter (grid) value.[8] The parameter intervals between the coarse points are of size 2^{-k} and are called the *initial intervals*. The parameter intervals between the fine points are of size $2^{-(k+L)}$.

The relation in Equation (3) is extended to

$$\begin{aligned} \boldsymbol{f}^{k+L,r} &= \boldsymbol{p}^{(L,r)} \star \boldsymbol{f}^{k}, \quad 0 \le r < 2^L, \\ \text{where } \boldsymbol{p_r} \equiv \boldsymbol{p}^{(L,r)} &\equiv [\boldsymbol{p}] \downarrow_{(L,r)}, \boldsymbol{f}^{k+L,r} \equiv [\boldsymbol{f}^{k+L}]^{\downarrow}_{(L,r)}. \end{aligned} \tag{11}$$

The ‘subdivision rules tree’ is the binary tree containing rules $\boldsymbol{p_r}$, such that rule $\boldsymbol{p_r} \equiv \boldsymbol{p}^{(L,r)}$ resides in path $\boldsymbol{r} = \{r_1, r_1, \dots, r_L\}$ ($r = \sum_{l=1}^{L} r_l 2^{L-l}$, $r_l \in \{0, 1\}$) of the tree.[9] By convolution (11), $\boldsymbol{f}^{k+L,r}$ is a data fragment in shift r of $\boldsymbol{f}^{k+L}$ generated by applying rule $\boldsymbol{p_r}$ directly to the original coarse points $\boldsymbol{f}^k$. The ‘subdivision rules tree’, $\boldsymbol{p}^{(l,r)}$, $0 < l \le L$ (levels), $0 \le r < 2^l$ (shifts), therefore, expresses the rules for a level(l)–shift(r) (‘frequency–phase’) hierarchical grouping/decomposition of the whole refined data $\{\boldsymbol{f}^{k+l}\}_{l=0}^{L}$[10] into their fragments $\boldsymbol{f}^{k+l,r}$. The ‘primal rules tree’ satisfies the following: (a) rules $\boldsymbol{p}_{\{\boldsymbol{r},0\}}$ and $\boldsymbol{p}_{\{\boldsymbol{r},1\}}$ ($\boldsymbol{p}^{(L+1,2r)}$ and $\boldsymbol{p}^{(L+1,2r+1)}$) are the children of rule $\boldsymbol{p}_{\{\boldsymbol{r}\}}$ ($\boldsymbol{p}^{(L,r)}$). (b) The evolution of the rules from one level to the next (father to children) is ‘indirect’ (‘global’) as follows:

$$\boldsymbol{p}_{\{\boldsymbol{r},s\}} = [\boldsymbol{p}_{\{s\}} \star \boldsymbol{p}^{(L)}]^{\downarrow}_{(L,r)}, \quad s \in \{0, 1\}. \tag{12}$$

(c) The rules at each level and their corresponding data fragments are ordered by their location on the parameter (x) axis.

The ‘dual (adjoint) rules tree’: The ‘dual (adjoint) rules tree’ is defined by a permutation applied on each level of the ‘primal rules tree’, such that each rule (node) $\boldsymbol{r} = \{r_1, r_1, \dots, r_L\}$ is mapped to the reversed path node $\boldsymbol{\Pi}(\boldsymbol{r}) = \{r_L, \dots, r_1\}$. In the ‘dual rules tree’: (a) rules $\boldsymbol{p}_{\boldsymbol{\Pi}(\{\boldsymbol{r},0\})}$ and $\boldsymbol{p}_{\boldsymbol{\Pi}(\{\boldsymbol{r},1\})}$ ($\boldsymbol{p}^{(L+1,\Pi_L(r))}$ and $\boldsymbol{p}^{(L+1,\Pi_L(r)+2^L)}$) are the children of rule $\boldsymbol{p}_{\boldsymbol{\Pi}(\boldsymbol{r})}$ ($\boldsymbol{p}^{(L,\Pi_L(r))}$). (b) The evolution of the rules from one level to the next is ‘direct’ (‘local’) as follows:

$$\boldsymbol{p}_{\boldsymbol{\Pi}(\{\boldsymbol{r},s\})} = [\boldsymbol{p}_{\boldsymbol{\Pi}(\boldsymbol{r})} \star \boldsymbol{p}]^{\downarrow}_{(1,s)}, \quad s \in \{0, 1\}. \tag{13}$$

(c) The rules are not ordered geometrically, according to their sequence on the parameter axis.

The ‘dual (adjoint) scheme’: Given a subdivision scheme $S_{\boldsymbol{p}}$ (the ‘primal scheme’), its ‘dual (adjoint) scheme’, $S_{\boldsymbol{p}}^{\boldsymbol{\Pi}}$, is defined as the subdivision scheme whose ‘primal rules tree’ is the ‘dual (adjoint) rules tree’ of $S_{\boldsymbol{p}}$. By convolution (11), the refined data achieved by the ‘dual scheme’, $\boldsymbol{f}^{k+L,\boldsymbol{\Pi}} = S_{\boldsymbol{p}}^{L,\boldsymbol{\Pi}} \boldsymbol{f}^k$, are the reordering by the permutation Π_L of the original (primal) data fragments $\boldsymbol{f}^{k+L,r}$. The rules of the ‘primal scheme’ and the rules of the ‘adjoint scheme’ coincide on symmetric paths and specifically in the first and last rules at each level. This means that if the original scheme is interpolatory, then the ‘adjoint scheme’ (not interpolatory but) preserves the original coarse points.

Combining it all : the ‘varying-resolution’ model: Recall that traditionally the operation of a stationary subdivision scheme $S_{\boldsymbol{p}}$ is represented by the L-iterated operator $S_{\boldsymbol{p}}^L$ using Equation (10), and the operation of a (uniform) non-stationary subdivision scheme is represented by the L-iterated operator $S_{\mathrm{p}}^{(L,k)}$ generalizing Equation (10). The operator $S_{\mathrm{p}}^{(L,k)}$ transforms the coarse points at level k to the fine points at a *constant level* $k + L$ using the [content, structure] pair [*masks*, L] – the mask(s) (stationary or not) and a number L specifying the number of iterations (levels). This operation can be summarized by the relation: [*coarse points*, [*masks*, L]] $\rightarrow$ *fine points*.

The ‘varying-resolution’ model generalizes this transformation by a generalized iterative operation $S_{\mathrm{p}}^{\boldsymbol{T},\,\boldsymbol{k}}$, where $\boldsymbol{T}$ specifies a subtree, including the root, of the ‘subdivision rules tree’ of some non-stationary operator $S_{\mathrm{p}}^{(L,k)}$. The operator $S_{\mathrm{p}}^{\boldsymbol{T},\,\boldsymbol{k}}$ transforms the coarse points to *varying-resolution* fine points using the [content, structure] pair [*masks, tree*] (uniform scheme) or, more generally, the pair [‘*subdivision rules tree*’, *tree*] (non-uniform scheme) specifying a pruned ‘subdivision rules tree'. The tree structure pruning the 'subdivision rules tree' replaces the iteration number L to yield, through its leaves, varying levels of refinement resolutions:

(a) The *structure* (the pruning, regardless of the content) yields the x-axis (free parameter) locations and intervals: leaf (l, r) represents intervals of size $2^{-(k+l)}$ (2^{-l} of the initial intervals sized 2^{-k}) and shift r (specifying all the rth $2^{-(k+l)}$-sized intervals within the 2^{-k}-sized initial intervals).[11]
(b) The *content* yields the y-axis values: rule at leaf (l, r), $l \leq L$, in the pruned rules tree, is convolved according to Equation (11) directly with the coarse values $\boldsymbol{f}^k$ to compute the data fragment $\boldsymbol{f}^{k+l,r}$ (the rth in $\boldsymbol{f}^{k+l}$). The final varying-resolution fine values are composed of all these disjoint fragments.

Combining these (x, y) results achieved by the leaves, a $2^{-(k+l)}$ interval (at shift r) is assigned to the values of fragment $\boldsymbol{f}^{k+l,r}$.[12] The varying-resolution fine points are, therefore, associated with intervals of varying lengths depending on the path lengths of the leaves generating these fine points. These relations are portrayed in Figure 3. Overall, the tree leaves induce the fine points' values, their approximation level and shift, and the size of their free parameter interval by the relation [*coarse points*, ['*subdivision rules tree*', *tree*]] $\rightarrow$ *fine points*. Note that choosing an L-depth tree (pruned at a constant level (depth) L) yields the special case ['*subdivision rules tree*', L] (or [*masks*, L] if uniform), with 2^l fine values at each initial interval for each data level $k + l$, where $l \leq L$, which is the standard subdivision L-iterated operation ($S_{\mathrm{p}}^{(L,k)}$). It is the freedom to pick any subtree of the 'subdivision rules tree', using the operator $S_{\mathrm{p}}^{\boldsymbol{T},\boldsymbol{k}}$, that generalizes the iteration number L (the L-depth tree) and yields varying levels of refinement resolutions, potentially with less than 2^l fine values ($l \leq L$) at each initial interval. The complementaries to full 2^l values are artificial values, as discussed in the analysis in Section 3.2.2.

Note that, also for a uniform scheme, there can generally be a different tree (structure) for each initial interval (as portrayed in Figure 3). For a non-uniform scheme, not only the tree structure but also its content – the rules – is generally different for each initial interval.

The rules tree may also be permuted to yield the 'adjoint rules tree', shifted or modified to greater division numbers. These modifications and other mathematical details on the refinement models described above can be found in [16]. The musical interpretation of this model is demonstrated in Section 4.1.

3.2. *Subdivision-based models for data analysis*

In the following, we review the analysis models proposed in [17]. These models revert the synthesis models reviewed above.

3.2.1. *The 'subdivision regression' model*

The 'subdivision regression' model reverts the 'generalized perturbed schemes' model: it is the *inverse problem* of the least-squares system (1b) or (7). It extracts the masks ('generalized perturbed schemes') from the given fine points, coarse points and a level number L describing their resolution difference. The relation describing this model, therefore, reverses[13] the traditional subdivision operation [*coarse points*, [*masks*, L]] $\rightarrow$ *fine points* by [*coarse points*, (*fine points*, L)] $\rightarrow$ [*masks, error*]. This is done by extracting the tension vector parameter $\boldsymbol{\omega}$ in the following least-squares system:

$$\mathbf{P}(\boldsymbol{\omega})\boldsymbol{f}^k \simeq \boldsymbol{f}^{k+1}, \quad \boldsymbol{f}^k \in \mathbb{R}^n, \ \boldsymbol{f}^{k+1} \in \mathbb{R}^{2n-1}, \ \mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{(2n-1)\times n}. \tag{14}$$

The least-squares form with respect to $\boldsymbol{\omega}$ is

$$\mathbf{A}\boldsymbol{\omega} \simeq \boldsymbol{f}^{k+1}, \text{ where } [\mathbf{A}]_j = \boldsymbol{\Phi}^{[j]}\boldsymbol{f}^k, \mathbf{A} \in \mathbb{R}^{(2n-1)\times m}, \tag{15}$$

where $\{\boldsymbol{\Phi}^{[j]}\}_{j=1}^m$ are 'template matrices', as defined in Section 3.1.1.

The minimum l_2 norm solution to this least-squares system is a 'tension vector parameter' $\boldsymbol{\omega}^\star_{\text{lsq}}$, according to which $\mathbf{A}\boldsymbol{\omega}$ is closest in l_2 to $\boldsymbol{f}^{k+1}$:

$$\boldsymbol{\omega}^\star_{\text{lsq}} = \mathbf{A}^\dagger \boldsymbol{f}^{k+1}. \tag{16}$$

This solution represents the scheme $\boldsymbol{p}(\boldsymbol{\omega}^\star_{\text{lsq}}) = \sum_{j=1}^m (\omega^\star_{\text{lsq}})_j \boldsymbol{\phi}^{[j]}$ that best fits the data, namely, that its operation on the coarse data fits best to the fine data. The solution can be decomposed again to $\boldsymbol{\omega_p}$ and $\boldsymbol{\omega_q}$ representing the [smoothness, noise] operators $\mathbf{P}$ and $\mathbf{Q}$, interpreted again as MRA decomposition (5) that omits the outer differences $\boldsymbol{d}^k$. A set of solutions $\boldsymbol{\omega}^\star_{\text{lsq}}$ may be extracted on windows sliding along the data, with a weight matrix localizing the solution to the middle of the window. The set of solutions would then represent a piecewise uniform scheme. This sliding resembles the moving least-squares (MLS) method, where the tension vector $\boldsymbol{\omega}$ and the 'template masks' $\{\boldsymbol{\phi}^{[j]}\}_{j=1}^m$ play the role of the coefficients and the basis functions of the MLS, respectively.

Multiple-level mapping. The $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^L$ *signature:* To find the mapping between the coarse and the fine points through several (L) levels of a non-stationary scheme, $\mathbf{P}^{(L,k)} : \boldsymbol{f}^k \rightarrow \boldsymbol{f}^{k+L}$, System (15) can be generalized to a multilinear system with $\mathbf{A}$ as a tensor. It can then be solved by an appropriate extension to Equation (16) that yields a *tension matrix parameter* solution $(\boldsymbol{\Omega}^\star_{\text{lsq}})_{(L,k)} \in \mathbb{R}^{m\times L}$, with the columns $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^L$. Alternatively, an interpolatory scheme is assumed; one then down-samples $L+1$ data levels $\{\boldsymbol{f}^{k+i}\}_{i=0}^L$ from $\boldsymbol{f}^{k+L}$ and solves a set of L consecutive linear systems (15) that map from level to level in a greedy manner. Using Equation (16), this yields $(\boldsymbol{\Omega}^\star_{\text{lsq}})_{(L,k)} = \{\boldsymbol{\omega}^{k+i}\}_{i=1}^L$, representing by Equation (6a) the L masks $\{\boldsymbol{p}_{k+i}(\boldsymbol{\omega}^{k+i})\}_{i=1}^L$. These masks can be considered a *signature* characterizing the data and specifying the relations between its different resolutions. This signature can be analysed using various criteria, such as stationarity and symmetry, reflecting specific relations within the mask coefficients in all levels $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^L$. Hed and Levin [17] included the explicit mathematical expressions of several criteria. They used these criteria to improve and constrain the solution (the coefficients) of the MLS system. They also tracked the measures of these criteria along the sliding window for analytical purposes. The musical interpretation and usage of some of the criteria, with or without a sliding window, are given in [30].

Reconstruction: Recalling the synthesis relation [*coarse points*, [*masks*, L]] $\rightarrow$ *fine points*, the coarse points $\boldsymbol{f}^k$ and the masks extracted by the analysis, $\{\boldsymbol{p}_{k+i}(\boldsymbol{\omega}^{k+i})\}_{i=1}^L$ (composed of the solution $(\boldsymbol{\Omega}^\star_{\text{lsq}})_{(L,k)}$ and the template masks) can be now considered as an approximate compressed representation for the data $\boldsymbol{f}^{k+L}$ and can, therefore, be used for the reconstruction of the fine data using Equation (14) or (15), as demonstrated in Section 4.2.3.

3.2.2. *The 'tree regression' model*

The 'tree regression' model generalizes the 'subdivision regression' model and reverts the 'varying-resolution' model. The basic idea is to extract a multiple-level mapping as before, but this time by assuming that the given fine values are coming from different (varying) resolutions and not from a constant resolution (level $k+L$).

Recall that the 'varying-resolution' model synthesizes data by letting the leaves of a pruned 'subdivision rules tree' generate the varying-resolution fine points by [*coarse points*, [*masks, tree*]]

$\rightarrow$ *fine points* or, more generally, [*coarse points*, *pruned* '*subdivision rules tree*'] $\rightarrow$ *fine points*. This relation applies a pruned tree of maximum depth L, generalizing the traditional L-iterated operation (the full L-depth tree, that is, the iteration number L). In this synthesis, the leaf rules induce the fine points, their approximation level $k + l$, where $l \leq L$, and the parameter intervals, by a 2^{-l} proportion with the initial intervals (specified in Section 3.1.2).

The analysis process conversely aims at extracting the mapping between the given coarse and fine points and their parameter intervals, assuming that the fine points and their intervals are achieved from the coarse points using the 'varying-resolution' relation given above. The corresponding analysis relation is, therefore, [*coarse points*, [*fine points, parameter intervals*]] $\rightarrow$ [*masks, tree*], *error*] or, more generally, [*coarse points*, [*fine points, parameter intervals*]] $\rightarrow$ ['*subdivision rules tree*', *tree*], *error*]. Here, turning the synthesis around, the analysis extracts the [content, structure] pair ['subdivision rules tree', tree] specifying a pruned 'subdivision rules tree' such that its leaves yield the given varying-resolution data:

(a) The *structure* (the pruning) is yielded from the x-axis values, namely, from the varying size of the free parameter intervals: interval of size $2^{-(k+l)}$ and shift r corresponds to the leaf (l, r).
(b) The *content* (leaf rules) is yielded from the y-axis values – the varying-resolution fine values. First, recall from the synthesis (Section 3.1.2) that combining the leaves of the parameter tree (x) with the fine values (y) means that $2^{-(k+l)}$ intervals (at shift r) are associated with the values of fragments $\boldsymbol{f}^{k+l,r}$. Then, recall that these fragments are attained by convolving leaf rule (l, r) with the coarse values $\boldsymbol{f}^k$ through Equation (11). Hence, $\boldsymbol{f}^{k+l,r}$ should be de-convolved with the coarse points to yield the rule in leaf (l, r). In Figure 3, for instance, the interval and shift of the first two fine points associate them with leaves $(3, 0)$ and $(3, 1)$, namely, level 3 and shifts 0 and 1.

Given the intervals tree, the analysis is, therefore, represented by the relation [*coarse points*, [*fine points*, *tree*]] $\rightarrow$ ['*subdivision rules tree*', *error*]. The pruned tree of maximum depth L generalizes the full L iterations in the basic analysis relation of the 'subdivision regression'. This means assuming that the given fine points are coming from varying resolutions dictated by the pruned tree representing their intervals, instead of coming from constant resolutions (level).

As in the synthesis, note again that there is generally a different tree (structure) for each initial interval. In the more general case of a non-uniform scheme, not only the tree structure but also its content – the rules – is varying along the initial intervals at each level.

The $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^{L}$ *signature:* To extract the leaf rules, the de-convolution in (b) can be simplified by assuming again a *uniform* non-stationary scheme. Instead of this de-convolution, extract $S_{\mathrm{p}}^{(L,k)}$ – a set of L masks, uniform for all initial intervals: [*coarse points*, [*fine points*, *tree*]] $\rightarrow$ [*masks, error*]. Given the coarse points now, the varying-resolution fine points and the tree (structure) induced by the varying-size intervals associated with these fine points, it is left to extract the masks (the content).

Given the varying-resolution fine points attained by applying a subdivision rules tree of maximum depth L to coarse points at level k, the maximum resolution of these fine points is $k + L$ and their minimal interval is of size $2^{-(k+L)}$. Then, imagine that these varying-resolution fine values are artificially completed to full L-resolution data $\boldsymbol{f}^{k+L}$ – 2^L values at each initial interval. Assuming an interpolatory scheme, this completion would generally induce *real* and *artificial* fine values among the 2^l values at each data level $k + l$, for $l \leq L$. Bearing the synthesis model in mind, the real fragments at each level are yielded through Equation (11) by leaf rules. However, the artificial (missing) fragments at each level are yielded through Equation (11) by internal nodes (non-leaf) or nodes below the leaves. In Figure 3, for instance, in the first initial interval, at level

2, the fine points belonging to nodes (2, 2) and (2, 3) (leaves) are real and the fine points belonging to nodes (2, 0) and (2, 1) (non-leaves) are artificial.

Now, to extract $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^{L}$ representing the L masks that map between the coarse and the full L-resolution fine points, one can solve a set of L consecutive greedy systems (15) using Equation (16), as in the 'subdivision regression' model described above. Albeit, here not all the fine points are real and so the artificial fine points at each level are given zero weights. For this, weighted least-squares systems are applied on top of the basic systems (15). Since an interpolatory scheme is assumed, a real fine point at level $k+l$ (produced by some leaf rule at level l) is participating (assigned a non-zero weight) in all least-squares equations of its producer's level (l) and up (down the rules tree), that is, in equations $l, \ldots, L$.

Remark 3.3 If the number of fine points is rounded to the closest exponent 2^{L_r}, then L_r is the depth of the scheme extracted (number of equations required) by the basic 'subdivision regression' model. Note that the scheme extracted by the 'tree regression' model is with maximum depth $L \geq L_r$; the equality holds if a full-depth tree (constant resolution) is assumed and the inequality holds when assuming a *partial* tree (true varying resolutions).[14]

Under an appropriate constellation demonstrated in [30], the masks extracted using this model are signature characterizing certain time–pitch relations.

Reconstruction: Recalling the synthesis relation in this case, [*coarse points*, [*masks, tree*]] $\rightarrow$ *fine points*, the representation or the reconstruction of the fine (approximated) data also requires the tree, aside from the coarse points and the masks extracted by the analysis.

Remark 3.4 For more flexibility, template rules are used, rather than template masks (see Equation (3)). Additionally, *orthogonal* template masks/rules are used, where $\boldsymbol{\omega}$ completely represents the mask $\boldsymbol{p}(\boldsymbol{\omega})$, to make a distinction between the left side and the right side of the mask. Under these conditions, the 'masks' in the synthesis and analysis can be replaced by the tension vectors' coefficients $(\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}$. In the analysis case, then, the signature is simply $(\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}$ and the data representation/approximation is the pair $[\boldsymbol{f}^k, (\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}]$.

More options and mathematical details on the subdivision regression models described above can be found in [17].

4. Subdivision-based models for music synthesis and analysis

Supported by the rationale and based on the general principle described in Section 2, this section applies the data synthesis and analysis models described above to specific tasks in music synthesis and analysis, respectively.

4.1. *Subdivision-based models for music synthesis*

We propose procedures for the generation of monophony and polyphony, based on the musical realization of the synthesis models reviewed in Section 3.1.

4.1.1. *Music synthesis principle*

Basically, we propose to convert the synthesis models reviewed in Section 3.1 to a musical domain by taking the free parameter as the time and the value as the pitch. More generally, all musical dimensions – time, pitch, duration, loudness, etc. – can serve as the values of the subdivision operation and can be synthesized using the synthesis models, independently or rather with some dependence rules. This conversion means applying the [*masks, tree*] subdivision representation of the 'varying-resolution' synthesis model in the musical context. Accordingly, the subdivision operation in the musical domain mapping the coarse notes to the fine notes may overall be formulated by [*coarse notes*, [*masks, tree*] $\rightarrow$ *fine notes*]$_c$, where $(\cdot)_c$ indicates a single component (dimension) of the notes. Figure 3 depicts this idea when its y-axis can be any musical dimension, including pitch or time, as detailed in Procedure 8 and Figure 3. In the refinement of the time values, the 'tree' structure yields the basic rhythm tree and the masks create certain deviations from the rigid time intervals dictated by this rhythm tree.

When time is taken as the free parameter of pitch, the principle of the 'varying-resolution' model (Section 3.1.2) holds with the assignment of 'time' instead of with that of 'free parameter'. Here, therefore, the pruning of the 'subdivision rules tree' (the leaves) of the pitch dimension describes the relations between the interonset intervals, that is, *the pruned rules tree and the rhythm tree coincide*. The synthesis relation turns to [*coarse pitches*, [*masks, rhythm tree*]] $\rightarrow$ *fine pitches*, where the onset times of the fine pitches are dictated by the rhythm tree. Figure 3 depicts this case when its y-axis is pitch and its x-axis is time. This idea is explained in more detail in the procedures that follow and specifically in Procedure 3 and Figure 3.

This approach, when adapted and tuned for specific musical constraints and stylistic norms, yields an interactive tool for algorithmic music composition, which controls time and pitch simultaneously through a certain hierarchy and coincides with some perceptual findings [28], as explained later. At the same time, and importantly, this synthesis approach also constitutes a *didactic platform* critical for comprehending the *music analysis* tools proposed in Section 4.2.

4.1.2. *Music synthesis application and configurations*

The general process [*coarse notes*, [*masks, tree*] $\rightarrow$ *fine notes*]$_c$ is implemented as follows.[15] The input for the synthesis is a series of coarse notes ('controls' or 'anchors') for each part of the piece. Each input series is a vector-valued vector of several dimensions (time, pitch, loudness, duration, instrument and channel). The output is a piece, optionally of several parts, composed of the corresponding fine notes (possibly perceived as 'diminutions' or 'ornaments'). The notes, at some point, are generalized to vertical objects (e.g. chords). The masks are specified by [*Template masks, Tension vector parameter*] and the tree by [*division numbers(s), pruning, permutation, offsets*].

This application allows concurrent subdivision of several parts, each by independent [*masks, tree*] subdivisions for all musical dimensions, as demonstrated in the procedures below. Several options allow to coordinate between the masks and the trees of the different dimensions and parts. In addition, the musical scale is specified separately for each part. All the parameters are specified by the user interactively. The control (coarse) values and the trees for all parts and dimensions can either be inserted by the user or be generated randomly.

4.1.3. *Procedures*

In light of the musical conversion of the subdivision synthesis models, the synthesis procedures described below are musical realization of the respective geometric procedures proposed in [16]. The procedures are presented in increasing order of generalization[16] and are accompanied by examples.[17].

The first two procedures demonstrate the application of a standard subdivision and its perturbations, respectively. Procedures 3–6 demonstrate that synthesis results based on the trees' parallelism sound musically interesting.[18] Procedures 7–12, which have been proposed in [16] as enhancements to the geometric subdivision models, bear a special meaning in the musical space and extend the trees' parallelism, involving varying tempo, simple and compound meters, accentuation, unexpectedness and polyphony. These additional features and their relevance to our methods are explained in the procedures themselves.

For clarity, in the procedures, we consider uniform (either stationary or non-stationary) subdivision schemes, namely, the pair [*masks, tree*]. To deal with non-uniform schemes, this pair should be replaced by a general pruned rules tree. Also, please note that in the rest of the paper, we refer to the linear interpolation scheme by either 'the linear scheme', 'the linear mask' or 'the linear subdivision scheme'.

Procedure 1 (A basic pitch approximation) Bearing in mind the basic relation $\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k$ (Equation (1b)), we now deal with finite data vectors and matrices, that is, $\boldsymbol{f}^k \in \mathbb{R}^{n(k)}$, $\boldsymbol{f}^{k+1} \in \mathbb{R}^{n(k+1)}$ and $\mathbf{P} \in \mathbb{R}^{n(k+1)\times n(k)}$. We start with the application of a subdivision scheme to $n \equiv n(k)$ musical notes at level k (pitches defined by their fundamental frequency and chosen from any real or artificial scale), iteratively L times, up to a maximum approximation level $k + L$.

At this stage, we are still constrained by five precepts: (a) The subdivision is binary ($d = 2$). (b) Pitch is the time-dependent variable and time is the free parameter. (c) Subdivision applies only to pitch and time. (d) Durations are the interonset intervals at each level, with an optional articulation. (e) The subdivision scheme is uniform.

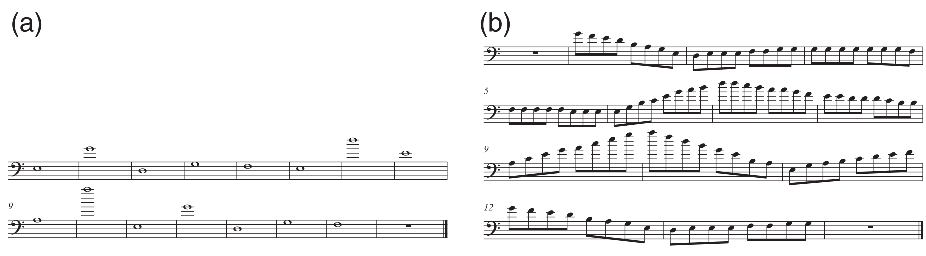
These limitations are relaxed later in Procedures 7–10.

The initial interonset intervals are not necessarily equal. The result is a sort of a melodic contour that passes through, or near, the initial notes. The initial (coarse) interonset intervals are the bars of this melody's score. The initial notes may be considered as a skeleton or a template for the refined data. The skeleton and the masks together may be considered as a representation of the generated piece.

Example 4.1 As mentioned earlier, Figure 4(a) and (b) (Audio4a and Audio4b, respectively) provides a simple example of a set of control notes subdivided so as to produce a linear approximation of resolution 3 (quavers, $\frac{1}{2^3}$). Note that the preceding rests are due to ignoring the marginal notes (see Remark 4.1).

Remark 4.1 The fine result at each level $k + l$, of size $n(k + l)$, is optionally aligned or truncated to include only the fine data in the relevant $n - m$ initial interonset intervals or in all $n - 1$ initial interonset intervals (letting notes be affected by the zeros outside the coarse notes). This alignment is more crucial in the next procedures.

The monophonic examples throughout the procedures include fine data from the relevant $n - m$ initial interonset intervals only. Their polyphonic extensions (explained later) include fine data from all $n - 1$ intervals.



(a) Control notes (Audio4a). (b) Linear approximation (mask $\{\frac{1}{2}, 1, \frac{1}{2}\}$) (Audio4b).

Figure 4. A basic pitch approximation using the linear subdivision scheme (level 3).

Procedure 2 (Pitch perturbations) Applying the 'generalized perturbed schemes', we choose a tension vector parameter outside the smoothness range of the scheme. This type of noising can add a particular character to the respective unperturbed melody while still maintaining a basic resemblance.

Example 4.2 Figure 5 (Audio5, Audio11), derived from the same control notes shown in Figure 4(a), is an example of a melody generated by the perturbed linear mask $\{\frac{6}{100}, 0, \frac{1}{2}, 1, \frac{1}{2}, 0, \frac{6}{100}\}$, selecting $\boldsymbol{\omega} = \{1, 0, \frac{6}{100}\}$ for the template masks in Equation (8). The certain audio resemblance between this perturbed melody and the unperturbed melody in Figure 4(b) (be aware of the Sol *vs*. Fa keys) is due to the relatively low degree of noise (perturbation of $\frac{6}{100}$) and due to the interpolation property, which preserves the control notes of both melodies (see Remark 3.1).

Note that here there are more preceding rests than in Figure 4(b) (Example 4.1), because the mask support is wider, creating wider margins. Thus, the low G3 on the downbeat of the fourth bar in Figure 5 corresponds to the same pitch in the fourth bar in Figure 4(b).

Procedure 3 (Rhythms and pitch-and-rhythm interrelationships) Assume a 'full approximation table', based on Equation (10), containing all the approximation levels $\boldsymbol{f}^{k+l}, l \in \mathbb{Z}_+$ (or $l = 1, \ldots, L$), with respect to $S_{\boldsymbol{p}}$, aligned according to Remark 4.1. A varying-resolution traverse within this table is induced by a set of pruned binary trees, which, along with a certain scheme and some initial values $\boldsymbol{f}^k$, induce a function, as demonstrated in Figure 3.

Here, we convert this [*masks, tree*] model to melody generation, by assigning time as the free parameter and pitch as the value. The resultant time-varying-resolution traverse induces the pitch values and the rhythm, such that the generated function is interpreted as a melody; the pitches are defined by the masks and the trees, and the rhythm is defined by the trees (this is generalized in Procedure 8). Figure 3 depicts this traverse when its y-axis is pitch and its x-axis is time.[19] At each point ('phase' or 'shift'), the duration[20] is derived by the approximation level. We consequently get a tool that simultaneously interpolates pitches and controls their rhythms. Overall, the trees represent the rhythm and, together with the approximation table, they also represent the pitch-and-rhythm interrelationships. A binary subdivision scheme generally induces a binary meter and, in general, division number d induces a meter of d-multiplicity (see Procedure 7).

A more general traverse that is not confined to trees would generalize the relations between the melodic and the interonset intervals. However, under fair interpolation conditions – for example, a small perturbation to the linear scheme – smaller melodic intervals would still correspond to smaller interonset intervals and vice versa.

This pitch-and-duration linkage in the context of music composition is partly aligned with, or may generalize, certain *perceptual* findings of van Noorden [28]. van Noorden indicated an inverse ratio between pitch intervals and perceived temporal coherence ('connectivity'). This finding, as he had emphasized, reinforces knowledge in music theory, according to which smaller melodic intervals are more common. He then showed that, in addition, there is a direct ratio between the size

Figure 5. Perturbations: approximation with a perturbed linear mask $\{\frac{6}{100}, 0, \frac{1}{2}, 1, \frac{1}{2}, 0, \frac{6}{100}\}$ applied on the control points in Figure 4(a) (Audio5, Audio11).

of the pitch intervals and the (minimal) length of the time intervals required to achieve temporal coherence – a finding that, according to him, does not have a parallel in musical knowledge. This very finding is of particular interest to us, as it gives a perceptual rationale for the direct ratio between pitch and duration pertinent to our synthesis methods, under certain conditions.

Remark 4.2 Schillinger [31] translated a given curve to a notated melody by quantizing the time axis, such that the grid spaces induced the rhythm and the pitch, forming certain pitch-and-rhythm correspondences controlled by the curve and the rhythmic pattern. Note that if a Schillinger curve would have been built hypothetically by some ISS with control points on a diadic time grid, then we would have a special case of Procedure 3.

Remark 4.3 In the musical context, a traverse in the approximation table (of a binary scheme) that is not confined to (binary) tree leaves is manifested by dots and ties (in a binary meter). If one sticks to (binary) trees, then it is necessary to add special handling for dotted notes and ties, indicating connections between terminal nodes that are not necessarily of the same level (instead of just note truncation [29]). Note that such special connections in a general traverse are necessary in all meters (scheme's subdivision numbers), but are not necessarily equivalent to dots and ties (in a ternary meter, for instance, dots and ties are manifested in a ternary tree without any special connections (see Example 4.7 in Procedure 7)).

Procedure 4 (Rhythmic patterns and musical event hierarchies) Using a single rhythm tree with maximum depth L as a template in Procedure 3 yields a repeated rhythmic pattern (Figure 6). This template tree represents the repeated rhythm for the whole piece and, together with the full approximation table, it also represents the pitch-and-rhythm interrelationships.[21] The pitches are arranged in (l, r) groups $\boldsymbol{f}^{k+l,r}$ of different levels $l \leq L$ and are generated as follows: a non-terminal node (l, r) recursively activates its two subordinates (descendants), requests their products and interlaces them to a single product (delivered in turn to its father). A terminal node ends the recursion by generating $\boldsymbol{f}^{k+l,r}$ by the appropriate down-sampling at level l of the approximation table. The final refined data are eventually attained by node (0,0).

This approach gives a different perspective on musical hierarchy, based on musical event levels ('frequencies'), rather than on their individual location in time.

Procedure 5 (Rhythmic patterns and musical event hierarchies using the rules tree) Here, we associate the template rhythm tree described above with a pruned rules tree as follows.

Assume a rules tree with maximum depth L. According to Equation (11), the level–shift hierarchical grouping of the fine data by the fragments $\boldsymbol{f}^{k+l,r}$, $l \leq L$, can be associated with the

Perturbed, Rhythmic pattern (Audio6a, Audio6ap). A rhythm/rules-tree

Figure 6. [*Masks, tree*] subdivision application: (a) is a perturbed approximation as in Figure 5 (same controls and mask) with the addition of a rhythmic pattern corresponding to the pruned rules tree in (b). (a) Perturbed, rhythmic pattern (Audio6a, Audio6ap); (b) a rhythm/rules tree.

hierarchical grouping of the creators of these fragments – rules $\boldsymbol{p}^{(l,r)}$, $l \leq L, 0 \leq r < 2^l$. Therefore, the final result achieved in Procedure 4, by a template-tree pointer to the approximation table, can equivalently be produced using a pruned rules tree. The rhythm and the pitch-and-rhythm interrelationships are, therefore, expressed here by the pruned rules tree as follows. Each rule $\boldsymbol{p}^{(l,r)}$ – a node in the rules tree – represents a composer that is responsible for certain level (l) and shift (r). A non-terminal node is a pseudo-rule that recursively operates its assistants – the two descendant composers/rules, $\boldsymbol{p}^{(l+1,2r)}$ and $\boldsymbol{p}^{(l+1,2r+1)}$ – and then requests and merges their products. A terminal node[22] ends the recursion by generating $\boldsymbol{f}^{k+l,r}$ by the convolution in Equation (11). The final refined data are eventually attained by pseudo-rule $\boldsymbol{p}^{(0,0)}$.

Example 4.3 As outlined above, replacing the full rules tree in the setup of Example 4.2 (Figure 5) by the pruned rules tree in Figure 6(b) yields the rhythmic patterns in Figure 6(a) (Audio6a. Audio6ap is its extension to several voices, by Procedure 11).

Procedure 6 (Rhythm and pitch manipulations) *Pitch permutations, rhythmic patterns and pitch-and-rhythm interrelationships*. The 'adjoint scheme' yields the corresponding reordering of the data fragments $\boldsymbol{f}^{k+l,r}$ of the fine data $\boldsymbol{f}^{k+l}$, $l \leq L$. In the musical context, applying the 'adjoint scheme' yields permuted approximation notes that coincide with the unpermuted approximation notes in the first and last notes of each bar (initial interonset interval), preserving the coarse points in the case of an interpolatory scheme. The permuted and the unpermuted melodies, therefore, have structural similarity.

Example 4.4 Applying the 'adjoint scheme' (the 'adjoint rules tree' permutation) on the full rules tree of Example 4.2 (Figure 5) yields the note approximation in Figure 7(a) (Audio7a). For instance, G3-G4-E4-A4-C4-A4-E4-G4 in the fourth bar in Figure 5 are transformed to G3-C4-E4-E4-G4-A4-A4-G4 in Figure 7(a). This type of transformation seems to be musically plausible and satisfying, yet cannot be deduced easily from any readily apparent rule.

Example 4.5 Applying the 'adjoint scheme' (the 'adjoint rules tree' permutation) on the pruned rules tree of Example 4.3 (Figure 6) yields the note approximation in Figure 7(b) (Audio7b), which adds a rhythmic pattern and rearranges the pitches in Figure 6(a). Note that the half notes (minims) remain, because the first level's permutation is the identity. Example 4.3 supplied an amount of unexpectedness due to the perturbations. Here, on top of that, we have additional unexpectedness due to the permutation, which nonetheless preserves the structural resemblance.

Audio7bp extends the example to several voices by Procedure 11.

(a) (b)

Permuted tree, no rhythmic pattern (Audio7a).

Permuted tree with rhythmic pattern (Audio7b, Audio7bp).

Figure 7. Application of the pitch permutation corresponding to the 'adjoint scheme' [16] on (a) the perturbed approximation in Figure 5 and (b) the perturbed approximation in Figure 6(a). (a) Permuted tree, no rhythmic pattern (Audio7a); (b) permuted tree with rhythmic pattern (Audio7b, Audio7bp).

Pitch accentuation. In addition to the permutation and the pruning, we optionally attach an offset to the phase of each rule in the tree. It specifies the number of initial interonset intervals (those between the coarse notes $\boldsymbol{f}^k$) by which to shift the appropriate fine data fragment $\boldsymbol{f}^{k+l,r}$, $l \leq L$. When doing these disturbances for the pitch or the time dimension,[23] the notes are taken from the later or earlier bars of the respective unshifted refined piece. Since this breaks the general contour, it yields an effect of accentuation, based on the uniqueness, unexpectedness or prominence of the shifted pitches. Different offsets may be applied to each rule in the pruned tree, allowing a variety of such modifications to the melody.

Example 4.6 Figure 8(b) (Audio8b) is the result of applying a phase offset of 1 (out of 2^3) on rules $\boldsymbol{p}^{(3,1)}$ and $\boldsymbol{p}^{(3,2)}$, with the setup of Example 4.3 (Figure 6(a) is drawn again in Figure 8(a)). This shifts one bar backwards all the second and the third quavers in all bars. Audio samples Audio8bp and Audio8cp are the result of applying a phase offset of 1 on rules $\boldsymbol{p}^{(3,1)}$ and $\boldsymbol{p}^{(3,2)}$, with the setup of the polyphonic audio samples in Examples 4.3 and 4.5, respectively. In Procedure 12, we give a more complex example for phase offsets.

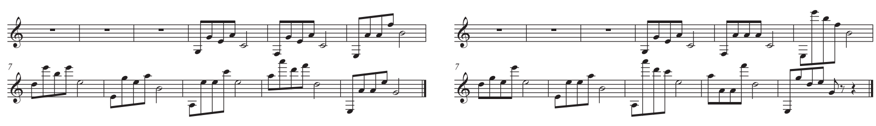
Procedure 7 (Simple and compound meters) The 'varying-resolution' methods can be extended to deal with trees that have any number of divisions at any node. Some observations can be made with regard to the musical interpretation of the division number and its extension.

Meters. The division number of the subdivision scheme, d, induces the meter of the melody, due to the following:

(a) Control notes (alternation): When the contour of the control notes is not monotonic, but rather, alternating, these notes will accentuate within the refined melody and eventually be the downbeats within d-meter bars.
(b) Mask (perturbations): The perturbations in Procedure 2. When the scheme is interpolatory, the control notes (at each level, recursively) are kept in place, whereas the in-between notes are significantly far from the control notes' average. This creates an accentuation at each d note, as can be heard in Examples 4.2 and 4.7.
(c) Tree (pruning): If the rules tree (the resulting rhythm tree) is *pruned*, then it creates grouping of the notes corresponding to d. This has been discussed and demonstrated in Procedures 3–6.
(d) Polyphonic schemata: Each d note is strengthened by other voices, as explained in Procedure 11.

Our approach, therefore, unifies the term 'subdivision' from both worlds – subdivision schemes and music – letting the division number of the former induce the time subdivision of the latter. Due to this parallelism, we occasionally use these terms interchangeably (e.g. 'ternary subdivision' refers both to the mathematical method applied and to the resulting triple meter). This is generalized further in Procedure 8.

(a) (b)



Perturbed, Rhythmic pattern (Audio8a, Audio8ap).

Phase offsets (Audio8b, Audio8bp, Audio8cp).

Figure 8. Phase offsets: (b) is the result of applying an offset of 1 on rules $\boldsymbol{p}^{(3,1)}$ and $\boldsymbol{p}^{(3,2)}$, on the setup of (a), shifting back by one bar each of second and third quavers in all bars. (a) Perturbed, rhythmic pattern (Audio8a, Audio8ap); (b) phase offsets (Audio8b, Audio8bp, Audio8cp).

Compound meters. The binary subdivision with pitch as the value and time as the free parameter correlates with rational rhythms.[24] When allowing subdivision by any division number $d \geq 2$ and enabling different division numbers in different resolution levels and shifts (tree nodes), the discussion is extended to composing pieces with compound meters, correlating with irrational rhythms.[25]

Example 4.7 Figure 9(a) (Audio9a), derived from the same control notes shown in Figure 4(a), is an example of a melody generated by three iterations of a perturbed *ternary* linear mask $\{\frac{1}{10}, \frac{1}{3}, \frac{2}{3}, 1, \frac{2}{3}, \frac{1}{3}, \frac{1}{10}\}$, selecting $\boldsymbol{\omega} = \{1, \frac{1}{10}\}$ with the template masks (9), and the pruned tree in Figure 9(b).[26] Figure 9(c) (Audio9c) – adjusted to C minor – is created in the same way, except that the mask is perturbed now by $\boldsymbol{\omega} = \{1, \frac{5}{100}\}$ and the rhythmic pattern, corresponding to the tree in Figure 9(d), includes rests (dashed branches with hollow circles). Here, all $n - 1$ (14) intervals are rendered (only 6 are shown), leaving no margins.

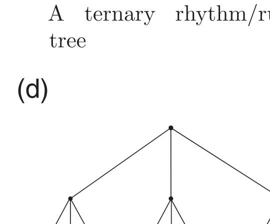
According to Remark 3.2, these masks yield non-interpolatory ternary schemes and hence the control notes in both cases are not part of the final result. In these monophonic examples, the meter is triple and is created by the alternation of the control points (in most parts), the mask perturbation and the rhythm (tree pruning).

Audio samples Audio9ap1 and Audio9ap2 are generated by the first perturbed ternary linear scheme, with modified controls. Audio9ap1 is the result of operating a *full* subdivision tree, that is, no actual rhythmic pattern is created. The triple meter in this example is hence created by the alternation of the control points, the mask perturbation and the polyphony. Audio9ap2 is the result of operating a *pruned* subdivision tree, resulting in a rhythmic pattern. The triple meter in this

(a)

Ternary subdivision with rhythm (Audio9a).

(b)

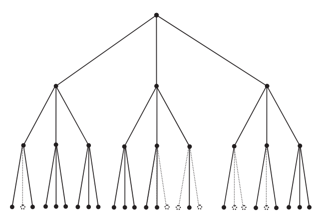


A ternary rhythm/rules-tree

(c)

Ternary subdivision with rhythm and rests (Audio9c)

(d)



A ternary rhythm/rules-tree with rests

Figure 9. [*Masks, tree*] ternary subdivision scheme and triple meter: (a) is the $\frac{27}{8}$ time result (drawn here without the preceding rests) of applying a perturbed ternary linear mask $\{\frac{1}{10}, \frac{1}{3}, \frac{2}{3}, 1, \frac{2}{3}, \frac{1}{3}, \frac{1}{10}\}$, on the controls in Figure 4(a), with the rhythmic pattern corresponding to the pruned rules tree in (b). (c) is a portion of the result (C minor) of applying a perturbed ternary linear mask $\{\frac{5}{100}, \frac{1}{3}, \frac{2}{3}, 1, \frac{2}{3}, \frac{1}{3}, \frac{5}{100}\}$, on the controls in Figure 4(a), with the rhythmic pattern corresponding to the pruned rules tree in (d), which includes rests, represented by dashed branches with hollow circles.

example is hence created again by the alternation of the control notes, by the mask perturbation and the polyphony, but also by the rhythm (tree pruning).

Procedure 8 (Generalized metric subdivision and varying tempo) It is useful to first note that subdividing pitch by given [*masks, tree*] with time as the free parameter, as above, is equivalent to subdividing time by the linear subdivision scheme with the same given tree. In light of this, we can now extend the subdivision operation by applying a general (not necessarily linear) scheme on the time dimension. Overall, we, therefore, operate independent masks, permutations, offsets and trees separately on the pitch and the *time*. All the effects achieved by the previous procedures on the pitch dimension – perturbations, permutations, shifts and accentuation – are now also applied on the time dimension. The pitch-and-rhythm interrelationships and the meter subdivisions described by the previous procedures are generalized accordingly. These extend the degrees of freedom of the musical result and produce interesting effects on the perceived melody.

Figure 3 depicts this idea when its y-axis can be any musical dimension, including pitch or time. The intervals of the free parameter (x-axis) at any level are no longer the interonset intervals. The latter are now found on the y-axis of the time subdivision. The rendered melody (pitches and rhythms) is now the combination of the separate time and pitch subdivisions: the pitches are defined by [*masks, tree*] applied on the pitch dimension, and the rhythm is defined by [*masks, tree*] applied on the time dimension, where the tree induces the general rhythm – the rough onset times, and the masks define the actual deviations from these rough onset times. In the general case, therefore, the ‘rough’ rhythm (rhythm tree) is induced by the *structure* (pruning) of the rules tree of the time dimension, and yet this ‘rough’ rhythm is being refined by the *content* (the actual rules derived from the masks) of the same rules tree.

The flexibility of stretching certain initial interonset intervals relative to others gives control over the tempo, as the beats speed up (stretched) and down (shrink) accordingly, either abruptly or smoothly. When the initial interonset intervals are equal and the scheme applied on the time dimension is shift invariant (not necessarily linear), the refined interonset intervals are also equal. If the initial interonset intervals are unequal (e.g. alternating), then applying the linear subdivision on the time dimension creates unequal bars with different fixed tempo, yielding a piece with piecewise fixed tempo, whereas a nonlinear subdivision would yield a gradual change in tempo. Bearing in mind the observations in Procedure 7, this can be viewed as an extension to the notion of metric subdivision: now the subdivided notes are not necessarily attained by integer number division but rather by some subdivision scheme, not necessarily linear, applied on the time dimension and by the proportion between the initial interonset intervals.[27]

Example 4.8 In the setup of Example 4.2 (Figure 5), we now alternate the sizes of the initial interonset intervals. Keeping the linear scheme applied on the time dimension yields Figure 10(a) (Audio10a). Replacing the time scheme by the nonlinear scheme having $\boldsymbol{\omega} = \{1, \frac{9}{100}, 0\}$, with the template masks (8), yields Figure 10(b) (Audio10b).

Procedure 9 (Multi-musical dimensions) The above can be extended to a *vector*-valued subdivision that applies a separate [*masks, tree*] setup on each dimension, with the same free parameter. The number of terminal nodes of the different trees should be coordinated. In the musical space, each note can be represented by a multi-dimensional point [time, pitch, loudness, duration, instrument, channel][28] that can be subdivided independently according to different pruned trees, mask parameters (not necessarily linear), tree permutations and phase offsets. Figure 3 is, therefore, valid for each dimension independently, retaining the free parameter and replacing the value by [time, pitch, loudness, duration, instrument, channel]. The parameter intervals and interonset intervals are as explained in Procedure 8. The rendered melody is now the combination of the separate subdivisions of all musical dimensions. The durations now are not necessarily derived

Linear time approximation (Audio10a, Audio10cp adds rhythm and voices).

Smooth (non-linear) time approximation (Audio10b, Audio10dp adds rhythm and voices).

Figure 10. Generalized time subdivision. A perturbed approximation as in Figure 5, with alternating sizes of initial interonset intervals: (a) with the non-smooth (linear mask) time approximation, yielding abrupt tempo (or meter) changes, and (b) with the smooth (four-point with $\omega = 0.09$) time approximation, yielding smooth tempo (or meter) changes. (a) Linear time approximation (Audio10a, Audio10cp adds rhythm and voices); (b) smooth (nonlinear) time approximation (Audio10b, Audio10dp adds rhythm and voices).

from (some articulation of) the interonset intervals in the various approximation levels, but rather are optionally rendered by their own [*masks, tree*] applied on some initial 'control durations' (some articulation of the initial interonset intervals).

Procedure 10 (Non-uniform schemes) We apply a *non-uniform* scheme such that the tension vector parameter is varied within each level and between levels. The implementation of the varying tension approach on the musical dimensions adds more effects and an additional amount of unexpectedness. Specifically, the results are interesting when the non-uniform schemes are applied on the time dimension.

Procedure 11 (Polyphony and homophony) In the geometrical procedure, concurrent ('vertical') subdivision processes – 'threads' – are operated independently on different sets of coarse points and are developed by different [*masks, tree*] setups, according to the 'horizontal' procedures. We convert this concurrency to polyphonic music generation, by applying separate schemes [*masks, tree*] on several concurrent sets of initial notes, where each initial set of notes, and its approximation result, is associated with one part of the polyphony. Each part is generated by the 'horizontal' subdivision procedures described above and musical constraints synchronize them to create harmony. The overall tree depth of each part induces its maximum resolution. Optionally, the coarse notes of the p different parts are induced by coarse p-note chords.

Example 4.9 Figure 11 (Audio11) is an extension of Example 4.2 (Figure 5) for two simultaneous parts (concurrent subdivisions), with approximation resolution levels 2 (crotchets, $\frac{1}{2^2}$) and 3 (quavers, $\frac{1}{2^3}$), respectively. Some other examples have already been described in the previous ('horizontal') procedures, as extensions to the monophonic setup.

The analogy observed below can serve to build counterpoint-like patterns while keeping the intervallic/harmonic relationships of the parts.

PRINCIPLE 4.1 (The rhythmic structure of counterpoint *vs*. concurrent subdivisions) *The rhythmic relations of the nth specie – 2^{n-1} notes against note – can be formed by subdividing concurrently two sets of control notes, by L ('cantus firmus') and $L + n - 1$ iterations, respectively: (a) First to third species: achieved by full rules trees. (b) Fourth and fifth species: These*

Figure 11. Concurrent subdivisions, where the upper part (higher notes) is the perturbed approximation in Figure 5 (Audio11).

have additional properties, such as suspensions, ornaments and special combinations, that can be implemented by specific selections of pruned trees for each part. (c) Generalized forms and unconventional tempo and metric relations can be achieved by more than two threads/parts ('complex counterpoint') with a different pruned tree for each part and dimension ('free counterpoint').

Procedure 12 (Combinations) We can achieve more flexibility by combinations of the various features of all procedures: perturbations, permutations, pitch-and-rhythm correlations and polyphony, alongside varying tempo, compound meters, accentuation and non-uniform schemes. For example, the perturbations in Procedure 2 can be done with either a full or a pruned subdivision tree, as shown in some examples throughout this section. Also, the 'vertical' Procedure 11 can be combined with 'horizontal' procedures. This was done in the multi-part examples of the 'horizontal' procedures, which were generated by applying variants of Principle 4.1-(3) on top of the 'horizontal' setup. Another possible combination involves phase offsets (Procedure 6) on all possible dimensions (Procedure 9), including the time dimension.

Example 4.10 Audio11 is the result of applying a phase offset of 1 on rules $\boldsymbol{p}^{(3,1)}, \boldsymbol{p}^{(3,2)}$ and $\boldsymbol{p}^{(3,3)}$, with the gradual tempo change in Figure 10(b) and with the addition of voices.

4.2. *Subdivision-based models for music analysis*

We propose procedures for music analysis based on the musical realization of the analysis models reviewed in Section 3.2.

4.2.1. *Music analysis principle*

Recall that the analysis models reviewed in Section 3.2 are the inversion of the data synthesis tools harnessed above for music synthesis. As such, we propose that these analysis models would be used analogously to achieve new music analysis tools. Technically speaking, music analysis, analogously to the music synthesis case, is based on transferring the analysis models reviewed in Section 3.2 to a musical domain, by taking any musical dimension (time, pitch, etc.) as the value of the subdivision operation and analysing it using the analysis models. This analysis is done independently between the musical dimensions or rather with some dependence rules. As a special case, time is taken as the free parameter, as explained hereafter.

More specifically, recall the subdivision mapping $\mathbf{P}^{(L,k)} : \boldsymbol{f}^k \to \boldsymbol{f}^{k+L}$. A melody, for example, can be represented by an array of points having the components (time, pitch, duration and loudness). We choose fine points $\boldsymbol{f}^{k+L}$ that describe this contour at some resolution. We assume that we are also given a set of coarse points $\boldsymbol{f}^k$ based on musical models that separate music into different levels of significance, as discussed later. We then use the 'subdivision regression' model to find a subdivision scheme mapping from the coarse values to the fine values. This scheme best characterizes the musical data, specifies the relations between their different resolutions and may be considered their *signature*.

In this way, we may analyse each component (e.g. time and pitch) of $\boldsymbol{f}^{k+L}$ independently by applying the 'subdivision regression' model in each dimension: [[*coarse notes*, [*fine notes, L*]] $\rightarrow$ [*masks, error*]]$_c$, where $(\cdot)_c$ indicates again a single component (dimension) of the notes. For convenience, rename L as L_r, to indicate 2^{L_r} real fine values at each initial interval. Now, alternatively to applying the plain 'subdivision regression' model, we can assume that these fine values are coming from different resolutions and apply the 'tree regression' model in each dimension. The 'tree regression' model is applied with a pruned tree of maximum depth L, where $L \geq L_r$ by Remark 3.3. The appropriate relation is then [[*coarse notes*, [*fine notes, tree*]] $\rightarrow$ [*masks, error*]]$_c$. Figure 3 depicts this idea when its y-axis can be any musical dimension, including pitch or time. In either cases, dependencies between the dimensions may be imposed.

A special case of this is created assuming that time is the free parameter of the pitch, in which case the tree is the rhythm tree induced by the interonset intervals. In this case, the analysis process becomes [*coarse pitches*, [*fine pitches, rhythm tree*]] $\rightarrow$ [*masks, error*]. Figure 3 depicts this case when its y-axis is pitch and its x-axis is time. We name this special case accordingly the 'rhythm tree regression' model.

The principle of the 'tree regression' model (Section 3.2.2) holds here with the assignment of 'time' instead of with that of 'free parameter': the pitches of the melody in this case are assumed to be with varying resolutions where the different levels (and shifts) are induced by the interonset intervals, that is, by the rhythm tree. Therefore, at each level, there would be real and artificial fine pitches according to the ratios of the interonset intervals.

Here, conversely to the music synthesis with the same assumption (time is the free parameter of the pitch), the interonset intervals induce the rhythm tree. We look for a 'subdivision rules tree' such that its structure (pruning) is described by the relations between the interonset intervals, that is, we look again for a *pruned rules tree coinciding with the rhythm tree*. Altogether, this analysis assumes that the rhythm tree connects between the time and the pitch through the subdivision scheme (tension parameters) extracted.

In accordance with the 'tree regression' model, the rhythm tree describes the real (leaf) and the artificial (non-leaf) pitches at each level (e.g. in Figure 3, the notes corresponding to nodes (2, 0) and (2, 1) are artificial). The rhythm tree also derives the approximation level of each (real/leaf) pitch: rule level l and data level $k + l$ are assigned to a pitch with an interonset interval of size $2^{-(k+l)}$. Now, L masks or tension parameters $\{\boldsymbol{\omega}^{k+i}\}_{i=1}^{L}$ are extracted by solving L weighted least-squares systems (15), with zero weights assigned to the artificial notes at each level. A real pitch with level $k + l$ participates with non-zero weight in all least-squares equations (15) of its rule level and up, that is, $l, \dots, L$.

The coarse notes may be taken as the downbeats (with a sampling factor, see Section 4.2.3). In this case, the fine values $\boldsymbol{f}^{k+L}$ are $1/(k + L)$ notes (e.g. $L = 3$ and $k = 0$ (initial interval of size 1) correspond to the eighth notes (quavers)). This analysis model can be used in the musical context to extract certain musical features and further to synthesize (reconstruct) given musical patterns using the synthesis relation [*coarse pitches*, [*masks, tree*]] $\rightarrow$ *fine pitches* again.

The general procedure of these music analysis methods is as follows:

(1) Choose certain coarse and fine notes within the given musical pieces. The coarse notes are selected based on cognitive models, musical schemata or known musical filters. The fine notes are sampled from the piece according to various options.
(2) Analysis (decomposition) and feature extraction: By the 'subdivision regression' or the 'tree regression' models and appropriate hierarchical reductions of the musical piece, extract non-stationary subdivision schemes (masks at different levels).

(3) Use the extracted features/masks in two directions:
   (a) Music synthesis by analysis: Reconstruct or represent the piece using the coarse notes, the scheme (masks) and the tree(s). This direction is discussed and demonstrated in Section 4.2.3.
   (b) Music analysis (both theoretical and automatic) and classification: The masks serve as typical signatures of the analysed pieces. Apply various criteria on the different levels of the extracted scheme and characterize the piece using these criteria. This direction has been studied and demonstrated in [30], which has proposed methods to characterize a musical piece through certain symmetry and pitch-and-rhythm interrelationships patterns.

4.2.2. *Music analysis application and configurations*

The process [[*coarse notes*, [*fine notes, tree*]] $\rightarrow$ [*masks, error*]]$_c$ is supported[29] by a general musical configuration allowing the extraction of the masks of either the time or the pitch dimension, by either the 'subdivision regression' model or the 'tree regression' model. Various options allow defining the coarse and the fine points and constrain the masks to be extracted in each dimension (time/pitch). The time can be left unanalysed (considered the pitch parameter), assuming that it is achieved by the linear subdivision scheme and the rhythm tree, or it can be analysed as well to extract its own mask (see synthesis Procedure 8).

The *input pieces* are any midi or **kern files (e.g. Humdrum, CCARH; [32,33]). The parameters are managed by scripts enabling automatic recording and playing of complex scenarios. The *coarse points* are based on musical knowledge and musical cognition principles, such as melodic peaks[30] [34], peaks, lows,[31] downbeats, known musical filters [35–38][32] and various down-samplings. The fine points are sampled from the piece to yield 2^L points between two adjacent coarse points. The sampling is based on the subdivision regression model (regression, tree regression and rhythm tree regression), the number of maximum levels, various weights imposed (musical filters, artificial value weights and others from [17]) and several options for interpolation in the x-axis (time) and the y-axis (pitch/time). For the *mask* and its smoothness-and-noise decomposition, the user specifies for each level the template masks (rules), their parity and tension value. The tension value is either known (inserted) or unknown (to be extracted). Weights and bounds are put on the coefficients of the tension vector to control the support of the mask and impose different patterns of symmetry [17,30]. The *time and the pitch dimensions* can be treated independently, each having its own configuration, acting above its own free parameter and yielding its own mask and approximation (reconstruction) results.[33] They can be made dependent by a matrix subdivision scheme or time may be considered as the free parameter of pitch, where points are not necessarily equidistant, as in the models proposed in [30]. The piece, the approximated piece and the fine points are optionally transposed to a chromatic or a diatonic scale. The masks extracted under these configurations are used to reconstruct the piece as explained below, and the quality measures (criteria) of the masks are used for the analysis or the classification of the piece, as explained in [30].

4.2.3. *Musical patterns synthesis (reconstruction) using the analysis models*

Using the 'subdivision regression' model and the 'tree regression' model, we extract masks from selected musical pieces and then reconstruct or approximate them.

In this example, we first extract the masks by the analysis relation of the 'rhythm tree regression' model [*coarse pitches*, [*fine pitches, rhythm tree*)] $\rightarrow$ [*masks, error*], where the onset times induce the rhythm tree, and then approximate the given piece by the synthesis relation [*coarse pitches*, [*masks, rhythm tree*]] $\rightarrow$ *fine pitches*, with the onset times induced by the rhythm tree. The triplet resulting from the analysis relation (coarse notes, masks and tree) represents the piece in a compressed manner by replacing the full varying-resolution data with several small masks and sparse coarse notes. The triplet's decompression by the synthesis relation restores the full varying-resolution data by applying the masks and the tree on the coarse notes.

The piece selected to demonstrate this idea is Chopin's Op. 10, No.12 ('Revolution'). In some respect, the bass part in this piece bears its melody. In the two examples given below, we show how to represent and reconstruct the low pitches of the bass part in the time segment [10, 60] with masks of support (width) 7.

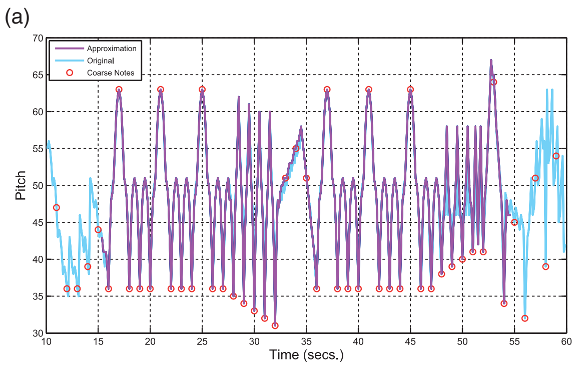
Example 4.11 In this example, we took the coarse notes as the downbeats with factor 0.5, namely, the downbeats and the half downbeats (1:4 note compression). Note that these coarse notes happen to be all the lows (local minima), occupied by the downbeats, and many of the high points (local maxima), occupied by the half downbeats. The fine notes are taken simply as the piece notes. Since most of the notes in the segment in this example are equidistant in time (semi-quavers), no special interpolation is needed in this case for the sampling of the fine pitches.

Output figures: Figure 12 depicts the results of the compression and decompression of the underlying segment of the piece. Figure 12(a) illustrates the coarse notes, the original piece (lowest bass notes) and the approximated piece reconstructed from the coarse notes, the scheme extracted and the rhythm tree, using the 'rhythm tree regression' model. The reconstructed contour was slightly adjusted, automatically, by

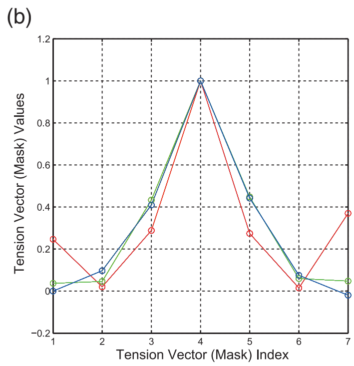
(a) straightening small angles that appear before or after large ones and
(b) rounding non-integers to integers in the scale, if possible, and maintaining the chromatic scale, otherwise.

Figure 12(b) shows the three levels of the scheme extracted. Figure 13 includes the score of the original bass part, the coarse notes and the reconstructed segment (Audio13_1, Audio13_2, Audio13_3).[34]

Results: From Figure 12(b), we learn that the three-level scheme extracted is almost stationary and closely symmetric: levels 2 and 3 coincide and are symmetric and level 1 is symmetric but deviates from levels 2 and 3. The approximate symmetry and stationarity of the scheme provide



Piece reconstruction.



Extracted masks.

Figure 12. Compression and decompression (reconstruction) of Chopin, Op. 10, No. 12, 'Revolution'. (a) shows the coarse notes (red circles), the piece (cyan) and the approximated piece (magenta) as reconstructed by the 'rhythm tree regression' model, from the coarse notes, the extracted masks and the rhythm tree. The masks at levels 1, 2, and 3 are shown in (b) plotted in red, green and blue, respectively. The rhythm tree in this case is pruned nearly at a constant level 4 (Audio13_1, Audio13_2, Audio13_3). (a) Piece reconstruction; (b) extracted masks. (colour online only)

Figure 13. Chopin, Op. 10, No. 12, ‘Revolution’, bass part. Middle staves: the original piece; bottom staves: the coarse control notes; upper staves: the reconstructed (approximated) piece. The reconstruction is done by applying the masks extracted from the original on the coarse notes (Audio13_1, Audio13_2, Audio13_3).

a slight additional compression, because it allows maintaining only one half mask (with small shifts to the first) instead of three full masks. Figure 12(a) shows that the reconstructed contour is very truthful to the original, along the whole segment analysed. The main deviations, as shown in Figure 13, appear in these isolated locations: (1) Parts of the last three quarters of bar 17 and the first quarter of bar 18, in which the down-up-down-up rising contour turns into a plain rise. (2) Parts of bars 25 and 26, in which the up-down contours surrounding the three peaks turn into plain rises and falls. The reason for these deviations is that the frequency of the changes in these locations is faster relative to the average frequency. A possible solution can be working with denser coarse points and extracting separate masks in sub-segments with a high frequency. More generally, the piece can be split into disjoint segments according to size limits and variation frequency, where each is represented or reconstructed by a separate set of coarse notes, scheme and tree(s).[35]

Conclusion: The high accuracy of the approximation and the stationarity and symmetry of the scheme suggest that the reconstructed contour or the compressed representation can serve in music comparison and in identification and retrieval.

Example 4.12 In this example, we compressed and decompressed the piece taking as coarse notes the downbeats only (factor $= 1$, namely, without the halves, 1:8 note compression). Since the half downbeats are now excluded, these new coarse notes are all the lows (local minima) and just the lows of the selected segment. The fine notes are the piece notes, as before.

In this case, we need to extract a four-level scheme to map between the coarse notes and the piece ($L = 4$). An additional mask is required to map between the downbeats and the downbeats plus half downbeats (the previous coarse notes) and the other three masks are the same as those extracted in the first (downbeats plus half downbeats) case. The quality of the reconstruction is very similar to that of the three-level scheme. However, the additional first-level mask turns to be neither symmetric nor coinciding with the three others. Therefore, the compressed representation in this case includes a full first-level mask and a single half mask representing the other three levels. Nevertheless, maintaining this full first-level mask replaces and hence saves keeping half of the initial coarse notes (the half downbeats/the high points).

Remark 4.4 (Time and rhythm considerations) Since most of the notes in the segment in this example are semi-quavers (equidistant in time), there is little significance to the rhythm, namely, the rhythm trees are nearly pruned at equal depth (level 3 in Example 4.11 or level 4 in Example 4.12). Hence, in this case, the 'rhythm tree regression' model almost coincides with the 'subdivision regression' model applied to the pitch. In [30], we analysed different cases where the rhythm *is* significant.

5. Future directions

Music synthesis: The synthesis infrastructure can be developed using artificial intelligence methods, such as genetic algorithms and Markov chains, to enhance the musical rules. This may allow the incorporation of rhythm (tree) and pitch variations along time, under melodic, harmonic, rhythmic and polyrhythmic constraints. Advanced counterpoint forms can be generated based on the degrees of freedom enabled by concurrent subdivision schemes (see Principle 4.1). In addition, the single-note interpolation can be extended to interpolation between segments of notes, chords

Table 1. Subdivision schemes – music: analogies created by assigning the time dimension to the subdivision's free parameter (x-axis) and time[a]/pitch/loudness/duration to the value (y-axis).

Subdivision synthesis model	Music	Procedures[b]
Coarse points	Contour, 'skeleton'	1
Intervals	Interonset intervals, durations	1
Initial intervals	Bars	1
Fine points	Melody, monophony	1
Linear or smooth pitch mask	Note interpolation	1
Perturbed mask (non-smooth)	Unexpectedness, metric accentuation	2
Pruned trees	Rhythmic patterns	3
Template pruned tree	Repeated rhythmic patterns, isorhythm	4, 5
Tree permutation	New musical permutations, unexpectedness	6
Phase offsets	Accentuation, unexpectedness	6
Rules tree	*Rhythm tree*	3–6
[*Masks, tree*] specification	Pitch-and-rhythm interrelationships	3–7
Division number(s)	Metric subdivision	7
General division number	Meter (binary, ternary, simple, compound)	7
Linear time mask	Standard (int.) time/metric subdivision	1–7
	Piecewise fixed tempo	1–7
Intervals, initial intervals	Grid for all selected musical dimensions	8, 9
Initial intervals for time values	Bars	8, 9
Smooth monotonic time mask	Generalized metric subdivision	8
	Varying tempo	8
Independent dimensions	Musical dimension control	9
Non-uniformity	Unexpectedness	10
Thread	Part/voice (in polyphony)	11
Concurrent subdivision threads	Polyphonic schemata, polyrhythm, counterpoint	11

[a]The time dimension may play a role in both axes.
[b]The numbers indicate the first procedures relevant to the feature.

(started in Procedure 11) or rhythms. The interactivity of the tool can be extended to collaborate several users, each associated with a separate part (voice).

Music analysis: The analysis framework can be extended to handle polyphony and characterize relations between masks of different voices. Advanced hierarchical models and musical schemata [e.g. [3,39]] can also be analysed when considering the basic notes as control notes and the ornaments or the variations as the fine points. The system can also be extended to handle non-binary and compound meters.

6. Conclusion

In this paper, we have given musical interpretation to the subdivision synthesis models proposed in [16] to achieve interesting analogies and new techniques for music generation, as described in Table 1. The synthesis infrastructure employs 'horizontal' and 'vertical' (concurrent) subdivisions for the generation of musical segments while controlling the musical scale of the monophonic pieces and the harmony of the polyphonic pieces.

The degrees of freedom of the musical parameters generate pieces that evolve from a certain skeleton but at the same time include the following: (a) unexpectedness due to resolution variations, generalized time subdivisions, perturbations, permutations and related properties, (b) pitch-and-rhythm linkages derived from the scheme's tree structure and rhythmic patterns and (c) extensions of polyphonic schemata. Additionally, the combination of the skeleton and the refinement parameters may be considered as a representation of the final result.

The new music synthesis methods are also used to derive and convey a new approach to music analysis. These music analysis methods are based on transferring the subdivision analysis models proposed in [17] to a musical domain. We have shown that the masks extracted by these analysis methods constitute signatures or features of selected musical patterns and allow their reconstruction (synthesis) and representation. We have also discussed further development directions, which may enhance monophony and polyphony generation and analysis.

Notes

1. Other topologies are possible.
2. A matrix scheme is represented by a matrix of masks or a matrix of matrices.
3. Up-sampling by n means inserting $n-1$ zeros (padding) between each adjacent element of the data. For signals in time, up-sampling corresponds to increasing the sampling rate.
4. Down-sampling by n means taking every nth element of the data. For signals in time, down-sampling corresponds to decreasing the sampling rate.
5. 'Time' may play a role in both axes (not simultaneously), either as the parameter or as the value, as explained in Procedure 8.
6. In an 'admissible' binary tree, each node has either 0 or 2 descendants. In our methods, we use a non-admissible tree to represent rhythms. In this case, rests are the missing siblings of a single subordinate. Accordingly, Figure 2(a) uses non-admissible trees and represents the rests by hollow circles in the missing siblings. However, if the tree *is* admissible, then rests should be identified with a silent note (as in [29]).
7. Where $[\boldsymbol{v}]^{\downarrow}_{(1,r)} = [\boldsymbol{v}]^{\downarrow}_{r}$, $[\boldsymbol{v}]^{\uparrow}_{1} = [\boldsymbol{v}]^{\uparrow}$ from Section 1.2.
8. Or it relates to multi-dimensional coarse and fine values.
9. The order of rules $\boldsymbol{p}^{(L,r)}, 0 \leq r < 2^L$, within $\boldsymbol{p}^{(L)}$ (the location of $\boldsymbol{p}^{(L,0)}$) requires a level-dependent alignment (detailed in [16]).
10. Contained in $\boldsymbol{f}^{k+L}$ in the case of an interpolatory scheme.
11. Namely, distant by $2^{-(k+l)}$ from the values created by next node at level l, $(l, r+1)$.
12. Distant by $2^{-(k+l)}$ from the next fragment of this resolution, $\boldsymbol{f}^{k+l,r+1}$.
13. The reversed subdivision reported in [27] recovers the input $\boldsymbol{f}^k$, whereas here the mask $\boldsymbol{\omega}$ (the operator $\mathbf{P}(\boldsymbol{\omega})$) is recovered.
14. Yielding the 2^{L_r} real fine values assuming a partial tree takes more than L_r iterations.
15. Implemented in C++, with a simple GUI (Graphical User Interface), utilizing DirectX and similar packages for music handling and sequencing.

16. Except for Procedure 3, which should have followed Procedure 5, but described before, for clarity.
17. The examples include score figures, and their corresponding midi audio files are given in the online supplementary of this paper
18. Due to this tree parallelism, we occasionally refer to the underlying tree as a 'rhythm/rules tree'.
19. In Figure 3, we omit the hollow terminal nodes representing the rests.
20. More generally, the interonset intervals.
21. The rhythmic repetition can be associated with the notion of isorhythm in music.
22. The rules $\boldsymbol{p}^{(l,r)}$ in the terminal nodes are prepared by gathering the appropriate data from the approximation table of $\boldsymbol{p}$ itself, with an appropriate alignment [16].
23. See Procedure 8 for time handling.
24. Rhythms built on duple meters with only simple subdivisions. Then, we can refer to the bar-start as to the time origin and express any time point or duration as a series of powers of $\frac{1}{2}$.
25. Non-duple or compound meters, where the signature's denominator is not a power of two, for example, $\frac{6}{8}$ (not to be confused with irrational numbers).
26. Margins are omitted from the score (Figure 9(a)).
27. Note that any subdivision scheme applied on the *time* dimension should preserve monotonicity.
28. Including time (!), due to Procedure 8, which refers to time as yet another component of the vector.
29. Implemented under Matlab® and [40].
30. High points of musical phrase.
31. Local maxima or minima calculated on a given neighbourhood.
32. Avoiding recursion, if the filter's least-squares solution avoids taking boundaries by our methods (least-squares), but only by LBDM (Local Boundary Detection Model) [36].
33. The time subdivision has to preserve a monotonic function.
34. The score here includes bars 12–26. The full analysed segment includes bars 8–28 and is given in the online supplementary of this paper.
35. A sliding window (overlapping segments) is not appropriate for compression, but it *is* relevant to other analysis methods [30].

References

[1] H. Schenker, *Fünf Urlinie-Tafeln/Five Analyses in Sketchform*, Dover Publications Inc., New York, 1969.
[2] J.J. Fux, *Gradus ad parnassum*. Translated by Alfred Mann, *Steps to Parnassus: The Study of Counterpoint*, W.W. Norton & Co., New York, 1725 (1965).
[3] R.O. Gjerdingen, *Partimento, Que Me Veux-Tu?* J. Music Theory 51 (2009), pp. 85–135.
[4] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music (GTTM)*, The MIT Press, Boston, MA, 1983.
[5] H.C. Longuet-Higgins and C.S. Lee, *The perception of musical rhythms*, Perception 11 (1982), pp. 115–128.
[6] G.K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, New York, 1949.
[7] B.B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman, New York, 1977.
[8] P. Prusinkiewicz and A. Lindenmeyer, *The Algorithmic Beauty of Plants (The Virtual Laboratory)*, Springer-Verlag, New York, 1990.
[9] P.S. Langston, *Six Techniques for Automatic Music Composition*, International Computer Music Association, San Francisco, CA, 1994, pp. 164–167.
[10] S. Manousakis, *Musical L-systems*, Master thesis, Conservatoire Royal de La Haye, The Netherlands, 2006.
[11] P. Thompson, *Fractal Music*; available at http://www.fractalmusician.com/.
[12] R. Voss and J. Clarke, *1/f Noise in music and speech*, Nature 258 (1975), pp. 317–318.
[13] C. Madden, *Fractals in Music: Introductory Mathematics for Musical Analysis*, 2nd ed., High Art Press, Salt Lake City, UT, 2007 (1999).
[14] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, *Synthesizing sound textures through wavelet tree learning*, IEEE J. Comput. Graph. Appl. 22(4) (2002), pp. 38–48.
[15] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. Davis, *Zipf's Law, music classification and aesthetics*, Comput. Music J. 29(1) (2005), pp. 55–69.
[16] S. Hed and D. Levin, *Subdivision models for varying-resolution and generalized perturbations*, Int. J. Comput. Math. 88(17) (2011), pp. 3709–3749.
[17] S. Hed and D. Levin, *A 'subdivision regression' model for data analysis*, 2012, in preparation.
[18] I. Karydis, *Symbolic music genre classification based on note pitch and duration*, ADBIS, Thessaloniki, Greece, 2006, pp. 329–338.
[19] P. Rolland, *Discovering patterns in musical sequences*, J. New Music Res. 28(4) (1999), pp. 334–350.
[20] N. Dyn and D. Levin, *Subdivision schemes in geometric modeling*, Acta Numer. 11 (2002), pp. 73–144.
[21] N. Dyn, J. Gregory, and D. Levin, *A four-point interpolatory subdivision scheme for curve design*, Comput. Aided Geom. Design 4 (1987), pp. 257–268.
[22] B. Jawerth and W. Sweldens, *An overview of wavelet based multi-resolution analyses*, SIAM Rev., 36(3) (1994), pp. 377–412.
[23] S. Mallat, *A Wavelet Tour of Signal Processing (Wavelet Analysis & Its Applications)*, Vol. 2, Academic Press, San Diego, CA, 1999.

[24] N. Dyn, D. Levin, and C. Micchelli, *Using parameters to increase smoothness of curves and surfaces generated by subdivision*, Comput. Aided Geom. Design 7 (1990), pp. 129–40.

[25] S. Schaefer, D. Levin, and R. Goldman, *Subdivision schemes and attractors*, Proceedings of the Third Eurographics Symposium on Geometry Processing, July 4–6, Vienna, 2005.

[26] A. Weissman, *A 6-point interpolatory subdivision scheme for curve design*, Master thesis, Tel Aviv University, Tel Aviv, 1990.

[27] F.F. Samavati and R.H. Bartels, *Multiresolution curve and surface representation: Reversing subdivision rules by least-squares data fitting*, Comput. Graph. Forum 18 (1999), pp. 97–119.

[28] L. van Noorden, *Temporal coherence in the perception of tones sequences*, Ph.D. diss., Technische Universiteit Eindhoven, The Netherlands, 1975.

[29] D. Rizo and J. Inẽsta-Quereda, *Tree-structured representation of melodies for comparison and retrieval*, Proceedings of the 2nd International Conference on Pattern Recognition in Information Systems, PRIS 2002, Alicante, 2002, pp. 140–155.

[30] S. Hed, R.O. Gjerdingen, and D. Levin, *Pitch-and-rhythm interrelationships and musical patterns: Analysis and modeling by subdivision schemes*, 2012, in preparation.

[31] J. Schillinger, *The Mathematical Basis of the Arts*, Da Capo Press, New York, 1948 (1976).

[32] D. Huron, *The Humdrum Toolkit*; available at http://www.music-cog.ohio-state.edu/Humdrum/.

[33] D. Huron, *A Library of Virtual Musical Scores in the Humdrum Kern Data Format*; available at http://kern.ccarh.org/.

[34] Z. Eitan, *Highpoints: A Study of Melodic Peaks*, University of Pennsylvania Press, Philadelphia, PA, 1997.

[35] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-realization Model*, University of Chicago Press, Chicago, IL, 1990.

[36] E. Cambouropoulos, *The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing*, Proceedings of the International Computer Music Conference (ICMC'2001), La Habana, Cuba, 2001.

[37] J. Thomassen, *Melodic accent: Experiments and a tentative model*, J. Acoust. Soc. Am. 71(6) (1982), pp. 1598–1605.

[38] R. Parncutt, *A perceptual model of pulse salience and metrical accent in musical rhythms*, Music Percept. 11(4) (1994), pp. 409–464.

[39] R.O. Gjerdingen, *Music in the galant style*, Oxford University Press Inc., New York, 2007.

[40] P. Toiviainen and T. Eerola, *The MIDI Toolbox*; available at http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox/.