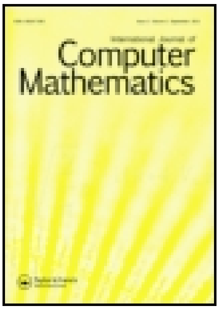

International Journal of Computer Mathematics

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

A 'subdivision regression' model for data analysis

Sigalit Hed[a] & David Levin[a]

[a] School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel

Accepted author version posted online: 12 Nov 2013.Published online: 14 Jan 2014.

To cite this article: Sigalit Hed & David Levin , International Journal of Computer Mathematics (2014): A 'subdivision regression' model for data analysis, International Journal of Computer Mathematics, DOI: 10.1080/00207160.2013.859252

To link to this article: http://dx.doi.org/10.1080/00207160.2013.859252

International Journal of Computer Mathematics, 2014
http://dx.doi.org/10.1080/00207160.2013.859252

Taylor & Francis
Taylor & Francis Group

A ‘subdivision regression’ model for data analysis

Sigalit Hed* and David Levin

School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel

(*Received 20 March 2013; revised version received 1 June 2013; second revision received 16 October 2013; accepted 18 October 2013*)

Subdivision schemes are multi-resolution methods used in computer-aided geometric design to generate smooth curves or surfaces. We propose two new models for data analysis and compression based on subdivision schemes:

(a) The ‘subdivision regression’ model, which can be viewed as a special multi-resolution decomposition.
(b) The ‘tree regression’ model, which allows the identification of certain patterns within the data.

The paper focuses on analysis and mentions compression as a byproduct. We suggest applying certain criteria on the output of these models as features for data analysis. Differently from existing multi-resolution analysis methods, these new models and criteria provide data features related to the schemes (the filters) themselves, based on a decomposition of the data into different resolution levels, and they also allow analysing data of non-smooth functions and working with varying-resolution subdivision rules. Finally, applications of these methods for music analysis and other potential usages are mentioned.

Keywords: subdivision schemes; multi-resolution analysis and approximation; trees; patterns; symmetry

2010 AMS Subject Classifications: 65D05; 65D10; 65D17; 65D18; 65D25

1. Introduction

1.1 *Background and contribution*

Subdivision schemes, and specifically binary subdivision schemes (BSS), are methods used to *generate* and represent smooth curves and surfaces by an iterative process of refinement [3,4]. This paper proposes new methods for exploiting the subdivision process for data *analysis* and compression. These methods are developed from recent *synthesis* methods, as follows.

Hed and Levin [9] laid an infrastructure for new synthesis techniques using subdivision schemes. That paper proposed two dual ‘subdivision rules trees’, the ‘primal rules tree’ and the ‘adjoint rules tree’, as arrangements of the subdivision rules that exemplify the multi-resolution nature of the scheme through decomposition into ‘resolution levels’ and ‘shifts’ and improve the convergence and smoothness analysis of subdivision schemes [8]. In [9], a ‘varying-resolution’ synthesis model is proposed based on these rules trees, a model which generalizes the standard subdivision process and allows a controlled generation of patterns. Hed and Levin [9] additionally proposed

*Corresponding author. Email: sigalit.hed.hakim@math.tau.ac.il the ‘generalized perturbed schemes’ synthesis model, which provides an alternative multiresolution analysis (MRA) representation and a flexible method for adding details to the refined data. Both models yield modified schemes that maintain structural similarity with the original schemes.

In this paper, we go a step further and suggest using the above properties of subdivision schemes for the analysis and compression of data, focusing here on the analysis aspect. Based on the synthesis models described above, we develop respective analysis models, the ‘subdivision regression’ and the ‘tree regression’, which allow the extraction of new features of the data, and that can be used for compression. These new models differ from existing MRA and subdivision methods; first, subdivision schemes are traditionally used for synthesis purposes, whereas we propose here using them for analysis. Second, as opposed to various wavelet transforms [18] and reverse subdivision [19], which do offer analysis methods, our methods extract from the given data some parameters related to the filters themselves, namely to the subdivision operator, based on a decomposition of the data into different resolution levels. The creation of these varying-resolution levels is a crucial phase of the analysis process, and it is based on anchor values selected as representatives of the given data. As such, this phase is generally dependent on the specific application involved. As will be explained, the models proposed are based on an inversion of the subdivision process that is different from the traditional reverse subdivision. In addition to the general value of the analysis models proposed in this paper, these models also have musical realizations and are used to yield methods for music analysis [8,10].[1]

The paper is structured as follows. In the rest of this section, we give the mathematical background and related work. In Sections 2 and 3, we propose the ‘subdivision regression’ and the ‘tree regression’ analysis models, respectively, while reviewing briefly the synthesis models in [9]. In Section 4, we discuss some extensions to the new analysis models. In Section 5, we propose criteria applied on these models that can serve as more compact features for analysis. In Section 6, some generalizations of the models are proposed for future development. In Section 7, we describe the applications for music analysis and suggest potential applications in other areas. Section 8 provides a summary.

1.2 *Subdivision schemes*

In the following, we briefly review rudiments of *scalar* (univariate) subdivision schemes [3,5] while emphasizing their relevance to the paper.

Mask, rules and iterations of subdivision schemes. Given a set of coarse values on the integers $\boldsymbol{f}^0 = \{f_i^0\}_{i\in\mathbb{Z}} \subset \mathbb{R}$, a BSS operator S recursively generates refined values $\boldsymbol{f}^k = \{f_i^k\}_{i\in\mathbb{Z}} \subset \mathbb{R}$ at refined resolution levels $k \geq 0$ on the grid $\{2^{-k}i\}_{i\in\mathbb{Z}}$. 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, $\boldsymbol{f}^0$, forms the ‘control points’ and recursively $\{\boldsymbol{f}^k = S^k\boldsymbol{f}^0\}_{k=1}^{\infty}$, such that iteration k of the scheme, S^k, generates, or corresponds to, the refined data forming the ‘fine points’ at resolution levels $k, \boldsymbol{f}^k$. An *interpolatory* subdivision scheme (ISS) additionally satisfies $f_i^k = f_{2i}^{k+1}$, $i \in \mathbb{Z}$, $k \in \mathbb{Z}_+$.

In case of a linear subdivision operator, the scheme hereby, denoted by $S_{\boldsymbol{p}}$, is represented by a mask $\boldsymbol{p} = \{p_j : j \in \mathbb{Z}\} \subseteq \mathbb{R}$. The refinement is defined through the mask by

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

For a general mask $\boldsymbol{p} = \{p_j\}_{j=m_1}^{m_2} \subseteq \mathbb{R}$, with size (cardinality) $|\boldsymbol{p}| = m_2 - m_1 + 1$, denote its support by $\varsigma(\boldsymbol{p}) = [m_1, m_2]$ and the size of the support by $|\varsigma(\boldsymbol{p})| = m_2 - m_1 = |\boldsymbol{p}| - 1$. We concentrate on masks with finite support $|\varsigma(\boldsymbol{p})| < \infty$.

Equation (1) has an alternative formulation as a convolution or a matrix operation:

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

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k, \tag{2b}$$

where $\mathbf{P} \equiv \mathbf{P}(\boldsymbol{p}) = \{P_{i,j}\}_{i,j\in\mathbb{Z}} \subseteq \mathbb{R}$, s.t. $P_{i,j} = p_{i-2j}$,[2] and $[\boldsymbol{v}]^{\uparrow}$ is the up-sampling-by-2 of the vector $\boldsymbol{v}$. $\boldsymbol{p} \star [\boldsymbol{f}^k]^{\uparrow}$ is the convolution of (the mirror of) $\boldsymbol{p}$ with the data $\boldsymbol{f}^k$ after up-sampling-by-2. Equation (2a) can be split into two convolution equations, by splitting $\boldsymbol{p}$ into two interlaced 'rules' (filters), $\boldsymbol{p}_{\{r\}}$:

$$\boldsymbol{f}^{k+1,r} = \boldsymbol{p}_{\{r\}} \star \boldsymbol{f}^k, \quad r = 0, 1, \tag{3a}$$

$$\boldsymbol{f}^{k+1,r} \equiv [\boldsymbol{f}^{k+1}]^{\downarrow}_r, \tag{3b}$$

$$\boldsymbol{p}_{\{r\}} \equiv [\boldsymbol{p}]^{\downarrow}_r, \tag{3c}$$

where $[\boldsymbol{v}]^{\downarrow}_r \equiv \{v_{2j+r},\ j \in Z\}$, $r = 0, 1$, is the r-shifted down-sampling-by-2 of the vector $\boldsymbol{v}$. The two rules $\boldsymbol{p}_{\{0\}}$ and $\boldsymbol{p}_{\{1\}}$ are responsible for a single refinement iteration ($L = 1$), where rule $\boldsymbol{p}_{\{0\}}$ transfers to the new values and $\boldsymbol{p}_{\{1\}}$ transfers to the in-between values on the next, doubly finer, grid. Subdivision schemes with general division number d (e.g. ternary) can be described by d (e.g. three) basic rules by appropriate extensions of Equation (3) and the grid subdivisions. For interpolatory scheme, $\boldsymbol{p}_{\{0\}} = \boldsymbol{\delta}$, the Kronecker delta $\boldsymbol{\delta} = \delta_0 = \{\delta_{i,0}\}_{i\in\mathbb{Z}}$.

A *stationary* subdivision scheme, $S_{\boldsymbol{p}}$, can be represented by a (seed) mask $\boldsymbol{p} = \{p_j : j \in \mathbb{Z}\} \subseteq \mathbb{R}$ and satisfies $\boldsymbol{f}^{k+1} = S_{\boldsymbol{p}}\boldsymbol{f}^k$. In a *non-stationary* scheme, the mask is dependent on the iteration, namely, $\mathfrak{p} = \{\boldsymbol{p}_k\}, k \in \mathbb{Z}_+$ and $\boldsymbol{f}^{k+1} = S_{\boldsymbol{p}_k}\boldsymbol{f}^k$. In a *non-uniform* scheme, the mask is additionally dependent on the location, namely, $\mathfrak{p} = \{\boldsymbol{p}_{(k,j)}\}, k \in \mathbb{Z}_+, j \in \mathbb{Z}$ and $(\boldsymbol{f}^{k+1})_j = S_{\boldsymbol{p}_{(k,j)}}\boldsymbol{f}^k$ [15]. A *symmetric* scheme $S_{\boldsymbol{p}}$ satisfies $p_j = p_{-j}, \forall j \in \mathbb{Z}$.

The L-iterated operator $S^L_{\boldsymbol{p}}$, $L \in \mathbb{Z}_+$, satisfies $\boldsymbol{f}^{k+L} = S^L_{\boldsymbol{p}}\boldsymbol{f}^k$, where $\boldsymbol{f}^k$ are called the control(/coarse) points(/data) and $\boldsymbol{f}^{k+L}$ the fine(/refined) points(/data).[3] $S^L_{\boldsymbol{p}}$ is represented by the mask $\boldsymbol{p}^{(L)} = \boldsymbol{p} \star [\boldsymbol{p}^{(L-1)}]^{\uparrow} = \{p^{(L)}_j : j \in \mathbb{Z}\} \subseteq \mathbb{R}$, where $\boldsymbol{p}^{(0)} \equiv \boldsymbol{\delta}$. The operation of $S^L_{\boldsymbol{p}}$ is defined by the mask $\boldsymbol{p}^{(L)}$, as follows:

$$f^{k+L}_i = (S^L_{\boldsymbol{p}} f^k)_i = \sum_{j\in\mathbb{Z}} p^{(L)}_{i-2^L j} f^k_j. \tag{4}$$

In a non-stationary scheme, the L-iterated operator is $S^{(L,k)}_{\mathfrak{p}}$, with the matrix $\mathbf{P}^{(L,k)}$ s.t.

$$\boldsymbol{f}^{k+L} = \mathbf{P}^{(L,k)}\boldsymbol{f}^k. \tag{5}$$

Equation (4) can be written by 2^L rules:[4]

$$f^{k+L}_{j2^L+r} = \sum_{i\in\mathbb{Z}} p^{(L)}_{i2^L+r} f^k_{j-i}, \quad 0 \le r < 2^L. \tag{6}$$

Equation (6) is the basis for the dual rules trees and the 'varying-resolution' proposed in [9], which, in turn, are the basis for the 'tree regression' model proposed here.

Limit function and smoothness. Denote by $f^k(t)$ the piecewise linear interpolating function satisfying $f^k(j2^{-k}) = f^k_j$. Define the limit function (if it exists) of the subdivision process as $f(t) = \lim_{k\to\infty} f^k(t) = (S^{\infty}_{\boldsymbol{p}}\boldsymbol{f}^0)(t)$. In an *interpolatory* scheme $f(2^{-k}j) = f^k(j2^{-k}) = f^k_j$.

The scheme $S_{\boldsymbol{p}}$ is said to be $C^d SS$ if the limit function satisfies $f \in C^d$ (has d smooth derivatives). From [3,5], the following two conditions, Equations (7a) and (7b), imply that there exists a limit function satisfying $f(t) \in C^d(\mathbb{R})$, that is, $S_{\boldsymbol{p}}$ is $C^d SS$

$$p_{(d)}(1) = 2, p_{(d)}(-1) = 0, \tag{7a}$$

$$\|S^L_{\boldsymbol{q}(d)}\|_\infty < 1 \quad \text{(contractivity)}, \tag{7b}$$

where $\boldsymbol{q}(d)$ is the mask of the differences of the d-divided differences [3,5].

Perturbed subdivision schemes. A *perturbed* scheme has a *tension* parameter controlling its smoothness [5]. For example, the four-point interpolatory scheme [4] is a symmetric perturbation to the linear interpolation scheme $\boldsymbol{p} = \{\frac{1}{2}, 1, \frac{1}{2}\}$ by a tension parameter ω

$$\boldsymbol{p} \equiv \boldsymbol{p}(\omega) = \{-\omega, 0, \tfrac{1}{2} + \omega, 1, \tfrac{1}{2} + \omega, 0, -\omega\}, \tag{8}$$

and the six-point interpolatory scheme [20] is a symmetric perturbation by θ to the four-point scheme (8) with $\omega = \frac{1}{16}$

$$\boldsymbol{p} \equiv \boldsymbol{p}(\theta) = \{\theta, 0, \tfrac{-1}{16} - 3\theta, 0, \tfrac{9}{16} + 2\theta, 1, \tfrac{9}{16} + 2\theta, 0, \tfrac{-1}{16} - 3\theta, 0, \theta\}. \tag{9}$$

Hed and Levin [9] proposed the ‘generalized perturbed schemes’ synthesis model underlying the ‘subdivision regression’ model proposed in this paper, pointed out its relation to MRA and showed how it can be used conveniently to control the details of the refined data.

Subdivision schemes and MRA. Recall that Equation (4) for $L = 1$ can be written as Equation (2). In this operation, as clarified already, the coarse points $\boldsymbol{f}^k$ and the matrix $\mathbf{P}$ are given, whereas the fine points $\boldsymbol{f}^{k+1}$ are the unknown yielded by them.

The subdivision (forth) operation may be reversed [19], such that the fine points $\boldsymbol{f}^{k+1}$ and the matrix $\mathbf{P}$ are given, whereas the coarse points $\boldsymbol{f}^k$ are the unknown yielded by them. That reverse (back) operation works by finding the solution to the least-squares system in Equation (2b)

$$\mathbf{P}\boldsymbol{f}^k \simeq \boldsymbol{f}^{k+1}, \tag{10a}$$

$$\boldsymbol{f}^k = \mathbf{P}^\dagger \boldsymbol{f}^{k+1}, \tag{10b}$$

where $\mathbf{M}^\dagger = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^\dagger \mathbf{M}^{\mathrm{T}}$ denotes the Moore–Penrose pseudo-inverse of the matrix $\mathbf{M}$.

Now, these subdivision operation in Equation (2b) (or Equation (10b)) and its reverse operation in Equation (10b) may be extended to include differences/noise and associated with MRA reconstruction and decomposition, respectively, or with displaced subdivision [13]. These reconstruction and decomposition (under mutual orthogonality $\mathbf{P}^{\mathrm{T}}\mathbf{Q} = \mathbf{0}$) are

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k + \mathbf{Q}\boldsymbol{d}^k \text{ reconstruction (synthesis)}, \tag{11}$$

$$\left.\begin{aligned} (a)\, \boldsymbol{f}^k &= \mathbf{P}^\dagger \boldsymbol{f}^{k+1}, \\ (b)\, \boldsymbol{d}^k &= \mathbf{Q}^\dagger \boldsymbol{f}^{k+1}, \end{aligned}\right\} \text{ orthogonal decomposition (analysis)}. \tag{12}$$

$\boldsymbol{f}^k$ (low resolution data) is the minimum norm solution to the least-squares system in Equation (10a) yielded by the matrix $\mathbf{P}^\dagger$ and minimizing the error $\mathbf{Q}\boldsymbol{d}^k$ in Equation (11), and $\boldsymbol{d}^k$ (high-resolution data) is the differences representing the least-squares error vector. In [19], the traditional reverse subdivision problem is hence formulated by Equation (10a) and solved by Equation (12).

In [8,9], the primal and the dual rules (or matrices) trees are shown to extend the standard MRA synthesis (inverse wavelet transform, Equation (11)) and analysis (wavelet transform, Equation (12)) cascades, respectively. The 'generalized perturbed schemes' synthesis model [9] and the 'subdivision regression' analysis model (here, Section 2) are associated with these MRA synthesis and analysis, respectively. The 'varying-resolution' synthesis model [9] and the 'tree regression' analysis model (here, Section 3) extend these models, by allowing further the synthesis and analysis of varying-resolution data using varying-resolution subdivision rules.

In this paper, we propose an inversion of the subdivision process (the 'inverse problem') that is different from the reverse subdivision or wavelet transform: in our case, when referring to Equation (10a), the operator **P** is the unknown, rather than the coarse points $\boldsymbol{f}^k$.

1.3 *Related work*

As mentioned, we propose models for data analysis and compression based on subdivision schemes. Other MRA methods, such as wavelet transform and its inverse [18], related subdivision compression methods [11,16] and reverse subdivision [7,12,19], analyse and compress data using specific fixed filters, finally yielding coarser data and displacements. However, our methods analyse and compress the given (fine) data using some sparse (coarse) data, finally yielding the filters themselves, related in this sense to subdivision fitting.

The problem of fitting a surface to a given set of points using subdivision schemes serves in 3D graphics for editing, compression and rendering [17] and for image processing and compression [1]. The underlying assumption in these works is that the given surface is approximately an offset of a smooth surface, hence assuming specific schemes of strict properties. Our analysis methods relax this smoothness assumption, and also allow varying-resolution subdivision rules. By introducing such flexibility, a wider class of data sets, which may have fractal nature [6], might still be expressible as a displaced subdivision.

2. The 'subdivision regression' model

In the following, we present the 'subdivision regression' analysis model reverting the 'generalized perturbed scheme' synthesis model proposed in [9]. We begin with a special case (Section 2.1) and then generalize it to a model for reconstructing **P** from $\boldsymbol{f}^{k+1}$ and $\boldsymbol{f}^k$ (Sections 2.2 and 2.3). In Section 2.4, we discuss models extracting the mapping between $\boldsymbol{f}^k$ and $\boldsymbol{f}^{k+L}$ and in Section 2.5 we discuss the reconstruction of the data.

2.1 *Preliminary example: extracting the '4–6 point' scheme*

The following example relates to the 4–6 point scheme [9], and it conveys part of the incentive for the 'subdivision regression' model. Some terms used in this example (e.g. 'template masks' and 'tension vector parameter') are explained formally later.

Choose the *template masks* $\boldsymbol{\phi}^{[1]} = \{\frac{1}{2}, 1, \frac{1}{2}\}$ and $\boldsymbol{\phi}^{[2]} = \{-1, 0, 1, 0, 1, 0, -1\}$, and a *tension vector parameter* $\boldsymbol{\omega} = \{\omega, \theta\}$ (or $\boldsymbol{\omega} = \{1, \omega, \theta\}$), the *4–6 point scheme* is the two-tension parameters scheme

$$\boldsymbol{p}(\boldsymbol{\omega}) = \boldsymbol{p}(\{\omega,\theta\}) = \boldsymbol{p}(\omega) + \theta\boldsymbol{\phi}^{[3]} = \boldsymbol{\phi}^{[1]} + \omega\boldsymbol{\phi}^{[2]} + \theta\boldsymbol{\phi}^{[3]}, \tag{13a}$$

$$\mathbf{P}(\boldsymbol{\omega}) = \mathbf{P}(\{\omega,\theta\}) \equiv \mathbf{P}(\boldsymbol{p}(\{\omega,\theta\})) = \mathbf{P}(\boldsymbol{p}(\omega)) + \theta\mathbf{P}(\boldsymbol{\phi}^{[3]}), \tag{13b}$$

with subdivision matrix $\mathbf{P}(\boldsymbol{p})$, s.t. $\mathbf{P}_{i,j} = \boldsymbol{p}_{i-2j}$, satisfying Equation (2b). $\boldsymbol{p}(\{\frac{1}{16}, 0\})$, the 4-point mask at $\omega = \frac{1}{16}$, reproduces π_3 (cubics) and $\boldsymbol{p}(\{\frac{1}{16}, \frac{3}{256}\})$, the 6-point mask at $\theta = \frac{3}{256}$ reproduces π_5. These specific schemes, therefore, have geometric significance of smoothness and approximation. Both ω and θ can be extracted by solving the following least-squares system:

$$\mathbf{P}(\boldsymbol{\omega})\boldsymbol{f}^k \equiv \mathbf{P}(\{\omega, \theta\})\boldsymbol{f}^k \simeq \boldsymbol{f}^{k+1}. \tag{14}$$

Solving Equation (14) while setting $\omega = \frac{1}{16}$ yields the θ of the closest 6-point scheme, whereas solving Equation (14) while setting $\theta = 0$ yields the ω of the closest 4-point scheme. However, by solving Equation (14) with both parameters free, we essentially assume that the desired scheme is potentially a θ-perturbation of a 4-point scheme with tension parameter other than $\omega = \frac{1}{16}$. Therefore, the solution $\{\omega, \theta\}$ does not generally fulfill the smoothness requirement in Equation (7b) even if the extracted ω is in the range of smoothness of the original 4-point scheme. Still, if ω *is* in the smoothness range of the 4-point scheme, θ may be considered as a measure for the deviation in direction $\boldsymbol{\phi}^{[3]}$, from some 4-point-scheme-smoothness.

We, therefore, suggest to use the perturbations ω and θ as measures for the smoothness and the noise of the mapping $\mathbf{P} : \boldsymbol{f}^k \rightarrow \boldsymbol{f}^{k+1}$, respectively. This concept and the system in Equation (14) are generalized in Section 2.3.

2.2 *The inverse problem*

The 'generalized perturbed schemes' model. The 'generalized perturbed schemes' model, used for synthesis, represents the mask $\boldsymbol{p}$ using *template masks* $\boldsymbol{h} = \{\boldsymbol{\phi}^{[j]}\}_{j=1}^{m}$, *template matrices* $\mathcal{H} = \{\boldsymbol{\Phi}^{[j]}\}_{j=1}^{m}$ and a *tension vector parameter* $\boldsymbol{\omega}$ by

$$\boldsymbol{p}(\boldsymbol{\omega}) = \sum_{j=1}^{m} \omega_j \boldsymbol{\phi}^{[j]} = \mathbf{M}_{\boldsymbol{h}}\boldsymbol{\omega}, \tag{15a}$$

$$\mathbf{P}(\boldsymbol{\omega}) = \sum_{j=1}^{m} \omega_j \boldsymbol{\Phi}^{[j]} = \mathbf{M}_{\mathcal{H}}\boldsymbol{\omega}, \tag{15b}$$

where $\mathbf{M}_{\mathbf{h}}$ and $\mathbf{M}_{\mathcal{H}}$ are the matrices whose columns are the template masks $\mathbf{h}$ and the template matrices[5] $\mathcal{H}$, respectively. The least-squares equation (10a) presented in the following form indicates that $\boldsymbol{\omega}$ is the unknown:

$$\boldsymbol{f}^{k+1} = \boldsymbol{p}(\boldsymbol{\omega}) \star [\boldsymbol{f}^k]^{\uparrow} = \mathbf{M}_{\boldsymbol{h}}\boldsymbol{\omega} \star [\boldsymbol{f}^k]^{\uparrow}, \tag{16a}$$

$$\boldsymbol{f}^{k+1} = \mathbf{P}(\boldsymbol{\omega})\boldsymbol{f}^k = \mathbf{M}_{\mathcal{H}}\boldsymbol{\omega}\boldsymbol{f}^k, \tag{16b}$$

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

The 'subdivision regression' model. Now, in the analysis, we are interested in finding characteristic patterns in the input $\boldsymbol{f}^{k+1}$, based on $\mathbf{P}(\boldsymbol{\omega})$, and specifically, based on $\boldsymbol{\omega}$. For this purpose, we suggest to use the decomposition of $\mathbf{P}$ to these 'atomic' basis masks – the template masks. Accordingly, the 'subdivision regression' model reverts the 'generalized perturbed schemes' model for analysis purposes. It extracts the masks ('generalized perturbed schemes') from the given fine points, coarse points and a level number L describing their resolution difference. This is the inverse problem of the least-squares systems (2b) or (16b),[6] where the mapping $\mathbf{P}(\boldsymbol{\omega}) : \boldsymbol{f}^k \rightarrow \boldsymbol{f}^{k+1}$ itself is the variable data (rather than $\boldsymbol{f}^k$). $\mathbf{P}(\boldsymbol{\omega})$ becomes the underlying analysed data and $\boldsymbol{\omega}$ may be viewed as the response of $\mathbf{P}$ to the chosen template masks.

The relation describing this model, therefore, reverses in a special way (not the straightforward reverse) the traditional subdivision operation {*coarse points*, {*masks, L*}} → *fine points* by {*coarse points*, {*fine points*, *L*}} → {*masks, error*}.

2.3 *The least-squares form and the* $\boldsymbol{\omega}$ *signature*

This extraction of the scheme, when both the coarse and the fine values are given, is done by extracting $\boldsymbol{\omega}$ from the least-squares form of Equation (16), with respect to $\boldsymbol{\omega}$. We transform the inverse system (16) to a standard least-squares system with variable $\boldsymbol{\omega}$.

Definition 1 *For a given vector* $\boldsymbol{c} \in \mathbb{R}^n$ *and template matrices* $\mathcal{H} \equiv \{\boldsymbol{\Phi}^{[j]}\}_{j=1}^{m}$, *denote by* $\boldsymbol{A} \equiv \mathcal{A}(\mathcal{H}, \boldsymbol{c}) \in \mathbb{R}^{(2n-1)\times m}$ *the matrix whose columns are* $\{\boldsymbol{\Phi}^{[j]}\boldsymbol{c}\}_{j=1}^{m} \in \mathbb{R}^{2n-1}$, *namely*

$$[\boldsymbol{A}]_j = \boldsymbol{\Phi}^{[j]}\boldsymbol{c}, \quad j = 1, \dots, m, \tag{17}$$

where $[\boldsymbol{A}]_j$ *denotes the* jth *column of* $\boldsymbol{A}$.

We reformulate the subdivision regression problem given in Equation (16) based on Definition 1

$$\mathbf{A}\boldsymbol{\omega} \simeq \boldsymbol{f}^{k+1}, \tag{18a}$$

$$\mathbf{A} \equiv \mathcal{A}(\mathcal{H}, \boldsymbol{f}^{k}) \in \mathbb{R}^{(2n-1)\times m}. \tag{18b}$$

Note that $[\mathbf{A}]_j = \boldsymbol{\Phi}^{[j]}\boldsymbol{f}^k, j = 1, \dots, m$, namely

$$\mathbf{A} = \left(\begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[1]}\boldsymbol{f}^k \\ \vdots \end{bmatrix} \dots \begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[2]}\boldsymbol{f}^k \\ \vdots \end{bmatrix} \dots \begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[m]}\boldsymbol{f}^k \\ \vdots \end{bmatrix} \right). \tag{19}$$

The system (18) puts us back to a standard least-squares problem, searching for $\boldsymbol{\omega}$ according to which $\mathbf{A}\boldsymbol{\omega}$ is closest in L_2 to $\boldsymbol{f}^{k+1}$.[7]

The minimum L_2 norm solution $\boldsymbol{\omega}^{\star}_{\text{lsq}}$ to the least-squares (lsq) system (18) is

$$\boldsymbol{\omega}^{\star}_{\text{lsq}} = \mathbf{A}^{\dagger}\boldsymbol{f}^{k+1} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{\dagger}\mathbf{A}^{\mathrm{T}}\boldsymbol{f}^{k+1}, \tag{20}$$

where $\mathbf{A}^{\dagger}$ denotes, again, the Moore–Penrose pseudo-inverse of the matrix $\mathbf{A}$.[8] Equivalently

$$\boldsymbol{\omega}^{\star}_{\text{lsq}} = \operatorname*{arg\,min}_{\boldsymbol{\omega}\in\mathbb{R}^m} \sum_{i=1}^{2n-1} ([\mathbf{A}]^i\boldsymbol{\omega} - f_i^{k+1})^2, \tag{21}$$

where $[\mathbf{A}]^i$ denotes the ith row of $\mathbf{A}$.

This solution, $\boldsymbol{\omega}^{\star}_{\text{lsq}}$, represents the scheme $\boldsymbol{p}(\boldsymbol{\omega}^{\star}_{\text{lsq}}) = \sum_{j=1}^{m} (\boldsymbol{\omega}^{\star}_{\text{lsq}})_j \boldsymbol{\phi}^{[j]}$ 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 and the noise operators $\mathbf{P}$ and $\mathbf{Q}$, interpreted again as the MRA decomposition in Equation (12) avoiding outer differences $\boldsymbol{d}^k$ (see Section 6.2).

In addition, a set of solutions $\boldsymbol{\omega}^{\star}_{\text{lsq}}$ can be extracted on windows sliding along the free parameter axis, 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 (see Section 4.5).

$\mathbf{P}(\boldsymbol{\omega}^{\star}_{\mathrm{lsq}})$ characterizes the transition $\boldsymbol{f}^k \to \boldsymbol{f}^{k+1}$. Hence, given $\boldsymbol{f}^{k+1}$, under judical selection of $\boldsymbol{f}^k$ and the template masks $\boldsymbol{h}$ we may refer to $\boldsymbol{\omega}^{\star}_{\mathrm{lsq}}$ itself as to the *signature* of $\boldsymbol{f}^{k+1}$.

The template masks are not obliged to be symmetric, nor to a specific relation between their odd and the even coefficients (rules), as explained in the next section.

Remark 1 Obviously, for any $\boldsymbol{f}^{k+1} \in \mathrm{Im}(\mathbf{A})$, Equation (18) has a true equality. Note that

(a) $\boldsymbol{p}(\boldsymbol{\omega})$ is not obliged to smoothness.
(b) We can pick independent template masks.
(c) We can pick $m \geq 2n - 1$.[9] Instead of just increasing the number of template masks, this can be achieved by reducing the number of analysed points n and sliding a small-enough least-squares window, as explained in Section 4.5.

Hence, under certain conditions on $\boldsymbol{f}^k$,[10] the columns of $\mathbf{A}$ can (or almost) span $\mathbb{R}^{2n-1}$ (true equality in Equation (18)). On the other hand, we can work with $m \ll 2n - 1$ and the specific $\boldsymbol{f}^k$ and $\boldsymbol{f}^{k+1}$ might still fulfill $\boldsymbol{f}^{k+1} \in \mathrm{Im}(\mathbf{A})$. This in itself is an attribute of the data.

Figure 1 describes schematically the ‘subdivision regression’ model, while comparing it to the existing ‘reverse subdivision’ method [19]. Note that this figure relates to non-stationary masks $\mathbf{P}_k$ and to different sets of template masks at each level, $\boldsymbol{h}_k = \{\boldsymbol{\phi}^{[k,j]}\}_{j=1}^{m}$, $k \in \mathbb{Z}_+$.

Below, we discuss how to find the mapping between the coarse and the fine points through *several* (L) levels of a non-stationary scheme, apply certain criteria on it and use it for data reconstruction.

2.4 *Multiple-level mapping and the $\boldsymbol{\Omega}_{(L,k)}$ signature*

Here, we propose several options to extend the regression above to extract a multiple-level mapping from level k to level $k+L$. This mapping, $\mathbf{P}^{(L,k)} : \boldsymbol{f}^k \to \boldsymbol{f}^{k+L}$, is described in Equation (5) as $\mathbf{P}^{(L,k)}\boldsymbol{f}^k = \boldsymbol{f}^{k+L}$. We, therefore, aim at finding a non-stationary L-iterated scheme $\mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)})$ represented by the *tension matrix parameter* $\boldsymbol{\Omega}_{(L,k)} = \{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$, such that $\boldsymbol{f}^{k+L} = \mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)})\boldsymbol{f}^k$.

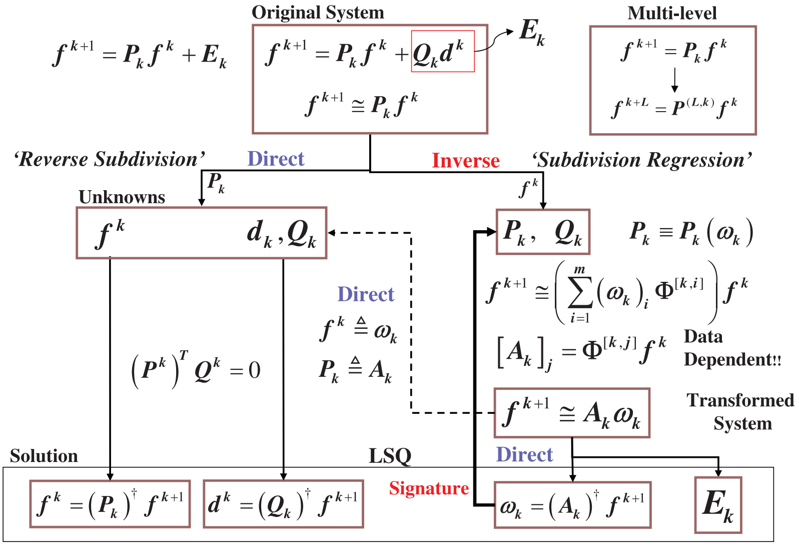
Original System
$f^{k+1} = P_k f^k + E_k$
$f^{k+1} = P_k f^k + Q_k d^k$
E_k
$f^{k+1} \cong P_k f^k$
Multi-level
$f^{k+1} = P_k f^k$
$f^{k+L} = P^{(L,k)} f^k$
‘Reverse Subdivision’
Direct
Inverse
‘Subdivision Regression’
P_k
f^k
Unknowns
f^k
d_k, Q_k
$P_k, \; Q_k$
$P_k \equiv P_k(\omega_k)$
$f^{k+1} \cong \left(\sum_{i=1}^{m} (\omega_k)_i \, \Phi^{[k,i]} \right) f^k$
Direct
$f^k \triangleq \omega_k$
$P_k \triangleq A_k$
$(P^k)^T Q^k = 0$
$[A_k]_j = \Phi^{[k,j]} f^k$
Data Dependent!!
$f^{k+1} \cong A_k \omega_k$
Transformed System
Solution
LSQ
Direct
$f^k = (P_k)^{\dagger} f^{k+1}$
$d^k = (Q_k)^{\dagger} f^{k+1}$
Signature
$\omega_k = (A_k)^{\dagger} f^{k+1}$
E_k

Figure 1. The ‘subdivision regression’ model compared with the ‘reverse subdivision’ [19].

In the greedy option of solving this problem, an interpolatory scheme is extracted one at a time, from one level to the next; in this case, one 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 (18) with the solution in Equation (20). We also propose a non-greedy option generalizing Equation (18) to a multilinear system with a *tensor* $\mathbf{A}$ yielding a *tension matrix parameter* solution $[\boldsymbol{\Omega}_{(L,k)}]^{\star}_{\mathrm{lsq}} \in \mathbb{R}^{m\times L}$, with the columns $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$.

Either way, the set of L tension vector parameters $\boldsymbol{\Omega}_{(L,k)} = \{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$ represents by Equation (15a) 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 their different resolutions, indicating the relation between $\boldsymbol{f}^{k}$ and $\boldsymbol{f}^{k+L}$ with respect to the chosen template masks. This signature can be used for reconstruction and can alternatively be analysed using various criteria, such as stationarity, variation and symmetry, reflecting specific relations within the mask coefficients in all levels $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$. The criteria, given explicitly in Section 5.1, are also used to improve and constrain the solution (the coefficients) of the least-squares system and are tracked along sliding windows for analysis purposes. The template masks may be designed to identify special features as symmetry or asymmetry of the mapping (see Section 5.2), or alternatively, as an orthogonal basis (see Remark 5).

The principles of the multiple-level mapping and the $\boldsymbol{\Omega}_{(L,k)}$ signature are the following.

Problem formulation. The non-stationary scheme to be extracted is

$$\mathbf{P}^{(L,k)} \equiv \prod_{i=0}^{L-1} \mathbf{P}_{k+i} \in \mathbb{R}^{(2^{L}n-1)\times n}, \tag{22}$$

where $\mathbf{P}_{k+i} = \mathbf{P}(\boldsymbol{p}_{k+i})$. We are interested in the inverse problem that generalizes the least-squares system (16)

$$\mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)})\boldsymbol{f}^{k} \simeq \boldsymbol{f}^{k+L}, \tag{23a}$$

$$\mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)}) \equiv \prod_{i=0}^{L-1} \mathbf{P}_{k+i}(\boldsymbol{\omega}_{k+i}), \tag{23b}$$

extracting the tension matrix parameter $\boldsymbol{\Omega}_{(L,k)} \in \mathbb{R}^{m\times L}$, whose columns are $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$.

The matrix $\mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)})$ represents a *non-stationary generalized perturbed scheme*; $\mathbf{P}^{(L,k)}$ is *not* assumed to be stationary, namely, $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$ need not be equal. However, to simplify the formulation below we assume that the template masks are stationary, namely, that the same set $\mathfrak{h}$ is used for all levels, rather than different sets of template masks at each level, namely, $\boldsymbol{h}_k = \{\boldsymbol{\phi}^{[k,j]}\}_{j=1}^{m}$, $k \in \mathbb{Z}_{+}$.

We define $\boldsymbol{f}_{\star}^{k+i}$, $i = 1, \dots, L$, as follows:

$$\mathbf{P}^{(i,k)}(\boldsymbol{\omega}_{k+i})\boldsymbol{f}^{k} = \boldsymbol{f}_{\star}^{k+i}, \quad i = 1, \dots, L, \tag{24}$$

where the last one, $\boldsymbol{f}_{\star}^{k+L}$, is the least-squares approximation for $\boldsymbol{f}^{k+L}$.

Remark 2 (Mask support) Recall that the support of a finite mask $\boldsymbol{p} = \{p_j\}_{j=m_1}^{m_2} \subseteq \mathbb{R}$ is $\varsigma(\boldsymbol{p}) = [m_1, m_2]$ with size $|\varsigma(\boldsymbol{p})| = m_2 - m_1$. The perturbed scheme with tension vector parameter, $\boldsymbol{p}(\boldsymbol{\omega})$ in Equation (15a), has the support $\varsigma(\boldsymbol{p}(\boldsymbol{\omega})) = \cup_{j=1}^{m}\varsigma(\boldsymbol{\phi}^{[j]})$, hence, if the template masks are independent, then $|\boldsymbol{p}(\boldsymbol{\omega})| \geq m$ and $|\varsigma(\boldsymbol{p}(\boldsymbol{\omega}))| \geq m-1$.

A greedy model. In the greedy model, we solve Equation (23) with input consisting of all the data levels $\{\boldsymbol{f}^{k+i}\}_{i=0}^{L}$, for which we require

$$\sum_{i=1}^{L}(\boldsymbol{f}^{k+i}-\boldsymbol{f}_{\star}^{k+i})^2 \to \min . \tag{25}$$

The following systems

$$\mathbf{P}^{(i,k)}(\boldsymbol{\omega}_i)\boldsymbol{f}^k=\boldsymbol{f}^{k+i}, \quad i=1,\ldots,L, \tag{26}$$

are not appropriate because they involve L multilinear systems, since here $\boldsymbol{\omega}_i \not\subset \boldsymbol{\omega}_{i+1}$.

A greedy approach is to solve a least-squares system (18) for each two consecutive levels

$$\mathbf{P}_{k+i}(\boldsymbol{\omega}_{k+i})\boldsymbol{f}^{k+i} \simeq \boldsymbol{f}^{k+i+1}, \quad i=0,\ldots,L-1. \tag{27}$$

These are L linear systems in the form (16), which can be solved by Equation (18). The fine points of equation i, $\boldsymbol{f}^{k+i+1}$ on the right-hand side of the equations, are the down-sampling by 2^{L-i} of $\boldsymbol{f}^{k+L}$ and are assigned as the coarse points for the next equation, $i+1$. Accordingly, the index of the points in $\boldsymbol{f}^{k+L}$ dictates their level, which is the lowest number of Equation (27) in which it will participate.

In Equation (27), we also allow using instead of $\{\boldsymbol{f}^{k+i}\}_{i=1}^{L}$ their least-squares approximation set $\{\boldsymbol{f}_{\text{approx}}^{k+i}\}_{i=1}^{L}$. It is attained recursively by solving the current level (equations appear in [8]).

A non-greedy model. To yield the least-squares solution for Equation (23), given only $\boldsymbol{f}^k$ and $\boldsymbol{f}^{k+L}$, we generalize the one-level least-squares system (18). Details on this extension are given in Section 6.3.

2.5 *Reconstruction: compression and decompression*

The analysis proposed inverts the synthesis relation $\{$*coarse points*, $\{$*masks, L*$\}\}$ → *fine points* by assuming certain coarse points $\boldsymbol{f}^k$ to extract the masks, $\{\boldsymbol{p}_{k+i}(\boldsymbol{\omega}^{k+i})\}_{i=1}^{L}$ (composed of the solution $(\boldsymbol{\Omega}_{\text{lsq}}^{\star})_{(L,k)}$ and the template masks). Now, these coarse points and masks can be considered as approximate compressed representation for the data $\boldsymbol{f}^{k+L}$ and, therefore, can be used for the reconstruction of the fine data using Equation (16) or Equation (18).

In Section 4, we extend and add constraints to the analysis and in Example 1, we demonstrate the analysis and reconstruction, with and without constraints.

3. The ‘tree regression’ model

The ‘tree regression’ model generalizes the ‘subdivision regression’ model described above and reverts the ‘varying-resolution’ model proposed in [9]. Therefore, to explain the ‘tree regression’ model properly, we summarize first the ‘varying-resolution’ model using Figures 2 and 3. The basic idea of the inversion is to extract a multiple-level mapping as before, but this time assume that the given fine values are coming from different (varying) resolutions and not from a constant resolution (level $k+L$), as explained below. The explanation is assisted by Figure 3.

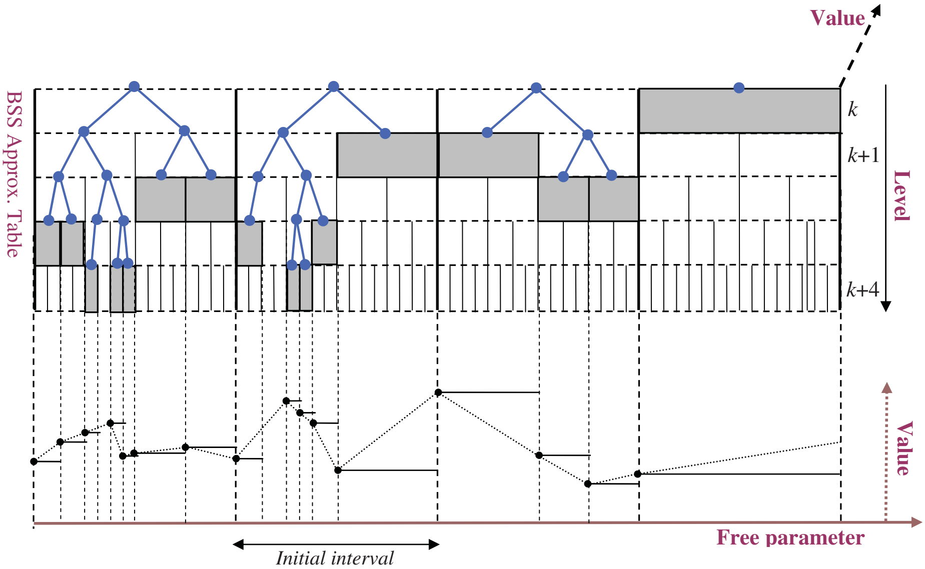

Figure 2. Varying-resolution [9]. Top: the subdivision full approximation table ($L_m = 4$) with an example for a varying-resolution traverse (gray rectangles). The coarse points $\boldsymbol{f}^k$ and the 'initial intervals' between them are represented by the top (widest) rectangles. The gray rectangles represent $\boldsymbol{f}^{k+L,r}$, namely, the resolution and shift (L, r) of the approximation values corresponding to the leaves of the chosen pruned trees (blue). Bottom: the actual refinement result $\boldsymbol{f}^{k\oplus\mathbb{T}}$ in Equation (30), depicted by the black dots, is generated by applying the $\mathbb{T}$-depth operator $S^{\mathbb{T}}$ on the equidistant coarse data $\boldsymbol{f}^k$ directly.

3.1 *The tree inverse problem*

The 'varying-resolution' model. The L-iterated scheme, $S^L_{\boldsymbol{p}}$, can be represented by the 2^L leaves of a binary rules tree of depth L. These are the 2^L rules

$$\boldsymbol{p_r} \equiv \boldsymbol{p}^{(L,r)} = [\boldsymbol{p}^{(L)}]^{\downarrow}_{(L,r)}, \quad L \in \mathbb{Z}_+,\ 0 \le r < 2^L,\ \boldsymbol{r} \in \{0,1\}^L, \tag{28}$$

which extend exponentially the two subdivision rules in Equation (3c) corresponding to $L = 1$. The standard operation of $S^L_{\boldsymbol{p}}$ is equivalent to operating singly each of these 2^L rules at level L, where each rule $\boldsymbol{p}^{(L,r)}$, $0 \le r < 2^L$, is responsible for the generation of a separate fine fragment $\boldsymbol{f}^{k+L,r}$, by the convolution

$$\boldsymbol{f}^{k\oplus\boldsymbol{r}} = \boldsymbol{f}^{k+L,r} = \boldsymbol{p_r} \star \boldsymbol{f}^k, \quad \boldsymbol{r} \in \{0,1\}^L, \tag{29}$$

generalizing Equation (3a) from $L = 1$ to any L.

However, a generalized, 'varying-resolution', subdivision operation uses a *pruned* rules tree (not necessarily pruned at a constant level L) as depicted on the $x - z$ plane in Figure 3. That figure shows how the pruned rules trees, along with the masks(s), are used to generate, through their leaf rules, *varying-resolution fine data* with varying sizes of intervals, as portrayed by the gray curve. These 'varying-resolution' fine data are composed of various fine fragments $\boldsymbol{f}^{k+l,r}$ from *different resolutions* $l < L$, where the values at each such fragment, $\boldsymbol{f}^{k+l,r}$, are associated with parameter intervals of size $2^{-(k+l)}$. Formally, the stationary $\mathbb{T}$-depth operator $S^{\mathbb{T}}_{\boldsymbol{p}}$ defined in [9] generates the *non-uniform varying-resolution fine data* $\boldsymbol{f}^{k\oplus\mathbb{T}}$ by

$$\boldsymbol{f}^{k\oplus\mathbb{T}} = S^{\mathbb{T}}_{\boldsymbol{p}} \boldsymbol{f}^k = \{S^{\boldsymbol{T}^j}_{\boldsymbol{p}} \boldsymbol{f}^k|_{I_j}\}_{j\in\mathbb{Z}}. \tag{30}$$

Altogether, 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*,

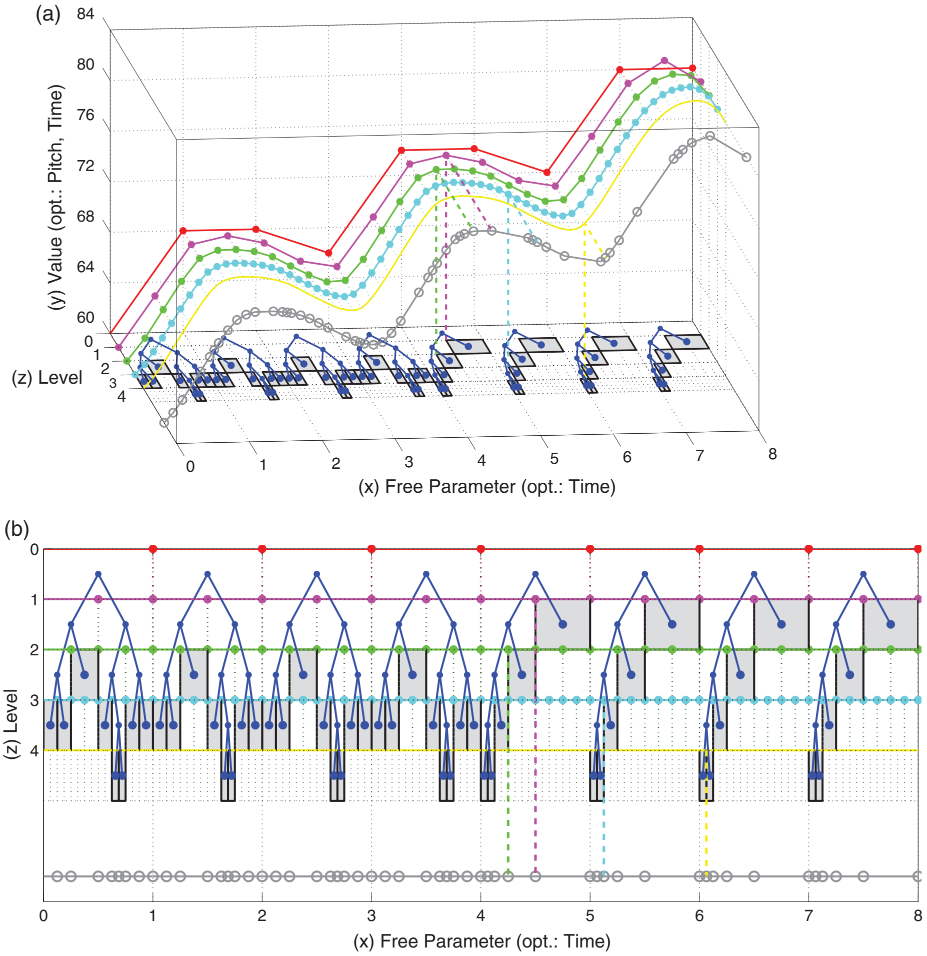

Figure 3. The 'tree regression' model, example: five approximation levels at $z = 0, \ldots, 4$ (red, pink, green, cyan and yellow, respectively) achieved by applying four iterations of the subdivision mask repeatedly on the coarse values (red at $z = 0$). The yellow curve at $z = 4$ also illustrates the smooth limit function. In the generalized operations, pruned binary trees (blue) are combined, such that their leaves (blue points) select approximation entries (gray rectangles) that their values are gathered into the varying-resolution fine data (gray). The vertical and the horizontal dashed lines demonstrate these selection and gathering, respectively. (b) The $x - z$ projection of (a). (a) The 'tree regression' model; example; (b) the 'tree regression' model; example; $x - z$ projection.

tree}} → *VR fine points* or, more generally, {*coarse points, pruned 'subdivision rules tree'*} → *VR fine points*. This relation applies a pruned tree of maximum depth L, generalizing the traditional L-iterated operation (the full L-depth tree, namely, 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. These values are equivalently achieved by a direct access to the different approximation levels, by the leaves of that tree, as depicted by the gray selections in the $x - z$ plane of Figure 3.

The 'tree regression' model. 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 above.

Accordingly, the 'tree regression' model reverts the 'varying-resolution' model: given coarse and fine values, it extracts a multiple-level mapping as described above for the 'subdivision regression' model, but it further assumes that the fine values are originated by rules from different resolutions. These resolutions are dictated by the intervals sizes, namely by the leaves of a pruned binary tree describing the intervals.[11] In referring to Figure 3 again, this means that the 'varying-resolution' fine values, depicted by the gray curve, are given along with the trees and the coarse data, and the mask is to be found. Assuming that the fine values are generally not equidistant, intervals of size $2^{-(k+l)}$ at shift r (within the initial intervals of size 2^{-k}) are associated with the fine values fragment at level $k+l$, $\boldsymbol{f}^{k+l,r}$ (created by leaves). Then, at each level, there will be missing values (created by non-leaves), or *virtual values*, to which a zero weight is assigned while solving the relevant least-squares equation (18) (extended to a weighted least-squares system (40) thereafter).

Altogether, the corresponding analysis relation is, therefore, {*coarse points*, {*VR fine points*, *parameter intervals*}} → {{*masks, tree*}, *error*} (more generally, {{*'subdivision rules tree', tree*}, *error*}). Here, inverting the synthesis, the analysis extracts a {*content, structure*} pair of the form {*'subdivision rules tree', tree*} specifying a pruned 'subdivision rules tree', such that its leaves would yield the given varying-resolution fine data:

(a) The *structure* (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, namely, the varying-resolution fine values. First, recall from the synthesis [9] 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 fragment $\boldsymbol{f}^{k+l,r}$. Then, recall that these fragments are attained by convolving leaf rule (l, r) with the coarse values $\boldsymbol{f}^k$ through Equation (29). Hence, $\boldsymbol{f}^{k+l,r}$ should be de-convolved with the coarse points to yield the rule in leaf (l, r).

    In Figure 2, 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*}} → {*'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 resolution (level).

As in the synthesis [9], note 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.

3.2 *Tree multiple-level mapping and the $\boldsymbol{\Omega}_{(L,k)}$ signature*

To extract the leaf rules, the de-convolution in (b) above can be simplified by assuming again a *uniform* non-stationary scheme and extracting $S_{\mathrm{p}}^{(L,k)}$, a set of L masks, uniform for all initial intervals: {*coarse points*, {*fine points, tree*}} → {*masks, error*}. Here, given the coarse points, the varying-resolution fine points and the tree (structure) induced by the varying-size intervals associated with these fine points, it is now left to extract the masks (content).

Given varying-resolution fine points attained by applying a 'subdivision rules tree' of maximum depth L on 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 a full L-resolution data $\boldsymbol{f}^{k+L}$, with 2^L values at each initial interval of size 2^{-k}. 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 (29) by leaf rules. However, the artificial (missing) fragments at each level are yielded through Equation (29) by internal nodes (non-leaf) or nodes below leaves. In Figure 2, 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 (27) (based on Equation (18)) using Equation (20), 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 (Equation (40) thereafter) are applied on top of the basic systems (18). 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), namely, in Equations (27), $l, \ldots, L$.

Remark 3 If the number of fine points is rounded to the closest exponent $2^{L_{\text{real}}}$, then L_{real} 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 $L \geq L_{\text{real}}$; the equality holds if a full-depth tree (constant resolution) is assumed and the inequality holds when assuming a *partial* tree (true varying resolutions).[12]

To summarize, we basically work with the greedy model, with these exceptions:

(a) Assume that for each level $i = 1, \ldots, L$, the fine points $\boldsymbol{f}^{k+i}$ consist of both the given points and virtual (complementary/missing) points to yield equidistant point set.

(b) Solve a 'greedy' system, but use the weighted least-squares system (Equation (40) afterwards) at each level in order to ignore the values in the virtual locations and get the least-squares solution.

Remark 4 As explained, at each level, there will generally be missing fine points belonging to the data fragments $\boldsymbol{f}^{k+L,r}$ generated theoretically by the non-terminal nodes (non-leaves), or non-existing nodes, at that level (by convolution (29)). Therefore, the level of each fine point in the full set of fine points, $\boldsymbol{f}^{k+L}$ (both real and complementary), is again dictated by its index in $\boldsymbol{f}^{k+L}$. Accordingly, the level of each point in the subset of *real* fine point, $\boldsymbol{f}_{\text{real}}^{k+L}$, is dictated by both its interval and index in $\boldsymbol{f}_{\text{real}}^{k+L}$.

To sum up, we solve Equation (27) with incomplete vectors $\{\boldsymbol{f}^{k+i}\}_{i=0}^{L-1}$. The missing data on the right-hand side of the equations – the fine points $\boldsymbol{f}^{k+i+1}$ – are handled by the weighted least-squares method, putting small or zero weights on the non-existing components.

In Equation (27), we also allow using instead of $\{\boldsymbol{f}^{k+i}\}_{i=1}^{L}$ or $\{\boldsymbol{f}_{\text{approx}}^{k+i}\}_{i=1}^{L}$ the theoretical points set $\{\boldsymbol{f}_{\text{Th}}^{k+i}\}_{i=1}^{L}$. It is composed recursively by inserting into the 'missing' components of the original sets the result of the application of the current level mask on the next (theoretical) coarse level data (equations appear in [8]).

3.3 *Reconstruction: compression and decompression*

The basic reconstruction principles are presented in Section 2.5. Now, the synthesis relation in the 'varying-resolution' case is generalized to the form $\{\textit{coarse points}, \{\textit{masks, tree}\}\} \rightarrow$

VR fine points. Therefore, in this case, the representation or the reconstruction of the fine (approximated) data require the tree as well, aside from the coarse points and the masks extracted by the analysis.

Remark 5 To gain more flexibility, we use template rules, rather than template masks, by allowing separate weights for odd and even rules (see Equation (28)). Additionally, we use *orthogonal* template masks/rules, $\mathfrak{h} = \{\boldsymbol{\phi}^{[j]} = \delta_{j-\lceil m/2 \rceil}\}_{j=1}^{m}$, where $\boldsymbol{\omega}$ completely represents the mask $\boldsymbol{p}(\boldsymbol{\omega})$ itself[13] and releases it from any symmetry constraints[14]. Under these conditions, the 'masks' can be replaced by the coefficients of the tension vectors (matrix) $(\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}$. In the analysis case, then, the signature is simply $(\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}$ and the data representation/approximation is the pair $\{\boldsymbol{f}^{k}, (\boldsymbol{\Omega}^{\star}_{\mathrm{lsq}})_{(L,k)}\}$.

Also, note that margins have to be excluded, such that the corresponding size of $\mathbf{A}$ in Equation (18) is in practice $(2(n-M)+1) \times m$, where $M = |\boldsymbol{p}_{\{1\}}|$, the size of the odd rule (generalized for any $L > 0$ in the multiple-level mapping in Section 6.3).

3.4 *General configuration*

The data structure {*Input, Translation, Output*} in Structure 3.1 specifies the degrees of freedom of the analysis infrastructure. It is based on the inversion of the synthesis data structure described in [9] (indicated by the square brackets [s: …]). The 'Output' structure specifies the schemes extracted by the analysis process, schemes that transfer Input(1) to Input(2). The mathematical models are implemented using this structure.

(A) **Input(1):** Coarse vectors ('skeleton' or 'control') – vector-valued vector.
(B) **Input(2):** Fine vectors [s: Output].
(C) **Output:** Array of {*Threads*} [s: Translation].

- **'Thread':** Array of {*Levels*}.
  - **'Level (iteration)':** Array of {*Dimensions*}.
    - **'Dimension':** Each 'Dimension' is characterized by:
      (a) **'Input(1)':** Scalar-valued vector.
      (b) **'Subdivision parameters':** {*mask, tree, error*}.
        (i) **'Mask':** {*Template masks, Tension vector parameter*} [s: 'Dimension' for non-uniformity].
        (ii) **'Tree':** {*Pruning, [s: Division numbers(s), Permutation, Shifts]*}.
        (iii) **'Error':** The residual $Res(\boldsymbol{\Omega})$ ((46) in Section 5.1) of the least-squares system, or its norm.

Structure 3.1: Analysis data structure.

4. Practical extensions, special cases and constraints

In this section, we elaborate on the implementation of the 'subdivision regression' model, add some extensions and constraints (Sections 4.1–4.3) and demonstrate them in Example 1. We also extend the model to handle piecewise uniform subdivision schemes (Sections 4.4 and 4.5).

4.1 *Shift invariance*

The subdivision solutions $\mathbf{P}(\boldsymbol{\omega})$ can be constrained to be invariant to shifting $\boldsymbol{f}^{k+1}$ and $\boldsymbol{f}^{k}$ by some constant $d \in \mathbb{R}$. This can identify similar patterns that differ only by their height.

Denoting $\boldsymbol{a}_i \equiv \{a, \dots, a\}^{\mathrm{T}} \in \mathbb{R}^i$, we, therefore, may require that

$$\mathbf{P}(\boldsymbol{\omega})(\boldsymbol{f}^k + \boldsymbol{d}_n) \simeq \boldsymbol{f}^{k+1} + \boldsymbol{d}_{2n-1}, \quad \boldsymbol{d}_i \in \mathbb{R}^i, \tag{31}$$

with the same least-squares error (residual), $\boldsymbol{e} \in \mathbb{R}^{2n-1}$, of system (16) or (18), meaning that if

$$\mathbf{P}(\boldsymbol{\omega})\boldsymbol{f}^k - \boldsymbol{f}^{k+1} = \boldsymbol{e}, \tag{32}$$

then

$$\mathbf{P}(\boldsymbol{\omega})(\boldsymbol{f}^k + \boldsymbol{d}_n) - (\boldsymbol{f}^{k+1} + \boldsymbol{d}_{2n-1}) = \boldsymbol{e}, \quad \boldsymbol{d}_i \in \mathbb{R}^i. \tag{33}$$

Equations (32) and (33) imply

$$\mathbf{P}(\boldsymbol{\omega})\mathbf{1}_n = \mathbf{1}_{2n-1}. \tag{34}$$

Since we restricted $\mathbf{P}(\boldsymbol{\omega})$ to the truncated size $(2(n-m)+1) \times n$, each $\mathbf{P}(\boldsymbol{\omega})$ row has all the components of either $\mathbf{p}(\boldsymbol{\omega})_{\{0\}}$ or $\boldsymbol{p}(\boldsymbol{\omega})_{\{1\}}$ and hence Equation (34) is equivalent to Equation (7a) with $d = 0$, namely π_0 reproduction by $S_{\boldsymbol{p}}$ (smoothness necessary condition for interpolatory schemes). Imposing the sufficient condition described in Equation (7b) is discussed in Section 6.1.

In a similar manner to the transition from Equations (16) to (18), Equation (34) can be rewritten as

$$\mathbf{B}\boldsymbol{\omega} = \mathbf{1}_{2n-1}, \tag{35a}$$

$$\mathbf{B} \equiv \mathcal{A}(\mathcal{H}, \mathbf{1}_n) \in \mathbb{R}^{(2n-1)\times m}, \tag{35b}$$

where $\mathbf{B}_j = \boldsymbol{\Phi}^{[j]}\mathbf{1}_n$

$$\mathbf{B} = \left(\begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[1]}\mathbf{1}_n \\ \vdots \end{bmatrix} \dots \begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[j]}\mathbf{1}_n \\ \vdots \end{bmatrix} \dots \begin{bmatrix} \vdots \\ \boldsymbol{\Phi}^{[m]}\mathbf{1}_n \\ \vdots \end{bmatrix} \right). \tag{36}$$

Here, however, Equation (35) are exact equalities rather than the least-squares ones in Equation (18).

We restrict $\mathbf{B}$ to the truncated size of $\mathbf{A}$, $(2(n-m)+1) \times m$. If the scheme is uniform, then two rows of $\mathbf{B}$ (one of each rule) will suffice (piecewise uniform in Section 4.4).

Now, we can solve the original least-squares system (18) under the constraints system (35) to yield a shift-invariant solution $\boldsymbol{\omega}_{\text{inv}}$. Generally, $\boldsymbol{\omega}_{\text{inv}} \neq \boldsymbol{\omega}^{\star}_{\text{lsq}}$.

Remark 6 If $\boldsymbol{\phi}^{[j]}(z = \pm 1) = 0$, then the jth column can be excluded from $\mathbf{B}$.

The shift invariance constraints system (35) can be generalized to fit the general non-greedy system, as explained in Section 6.3.

4.2 *Translation relaxation*

We can relax the requirements by searching the subdivision solution $\boldsymbol{\Phi}(\boldsymbol{\omega})$ that best maps $\boldsymbol{f}^k$ to some shifted version of $\boldsymbol{f}^{k+1}$. As opposed to the invariance constraint in Section 4.1, here we look for a specific $c \in \mathbb{R}$ that will yield that best mapping, and it becomes one of the unknowns

$$\boldsymbol{\Phi}(\boldsymbol{\omega})\boldsymbol{f}^k + \boldsymbol{c}_{2n-1} \simeq \boldsymbol{f}^{k+1}. \tag{37}$$

We define $\mathbf{A}' = [\mathbf{A}|\mathbf{1}_{2n-1}]$ and $\mathbf{B}' = [\mathbf{B}|\mathbf{0}_{2n-1}]$. Then, we can solve the least-squares system (18) under the constraints system (35), with the variables substitution

$$\boldsymbol{\omega} \leftarrow \{\omega_1, \dots, \omega_m, c\}, \mathbf{A} \leftarrow \mathbf{A}', \mathbf{B} \leftarrow \mathbf{B}'. \tag{38}$$

4.3 *Known and unknown tension parameters*

For applicative purposes, we allow the flexibility of fixing some of the tension parameters values. Denote $\mathcal{I} = \{i_l\}_{l=1}^{m'}$, $m' \leq m$, such that $\boldsymbol{\omega}' = \{\omega_j, j \in \mathcal{I}\}$ are the unknowns and $\boldsymbol{\omega}'' = \{\omega_j, j \notin \mathcal{I}\}$ are the known (fixed) ones. To solve the least-squares systems, we just substitute $\{j\}_{j=1}^{m'} \leftarrow \mathcal{I}$, $m \leftarrow m'$ and $\mathcal{H} \leftarrow \{\boldsymbol{\Phi}^{[j]}\}_{j\in\mathcal{I}}$ ($\mathfrak{h} \leftarrow \{\boldsymbol{\phi}^{[j]}\}_{j\in\mathcal{I}}$), in Sections 2.3 and 4.1.

More explicitly, we define $\mathcal{H}' \equiv \{\boldsymbol{\Phi}^{[j]}\}_{j\in\mathcal{I}}$; $\mathcal{H}'' \equiv \{\boldsymbol{\Phi}^{[j]}\}_{j\notin\mathcal{I}}$; $\mathbf{A}' \equiv \mathcal{A}(\mathcal{H}', \boldsymbol{f}^k)$; $\mathbf{A}'' \equiv \mathcal{A}(\mathcal{H}'', \boldsymbol{f}^k)$; $\mathbf{B}' \equiv \mathcal{A}(\mathcal{H}', \mathbf{1}_n)$; $\mathbf{B}'' \equiv \mathcal{A}(\mathcal{H}'', \mathbf{1}_n)$. Then, system (18) is solved under the constraints system (35), with the variables substitution

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}', \mathbf{A} \leftarrow \mathbf{A}', \mathbf{B} \leftarrow \mathbf{B}', \boldsymbol{f}^{k+1} \leftarrow [\boldsymbol{f}^{k+1} - \mathbf{A}''\boldsymbol{\omega}''], \mathbf{1}_{2n-1} \leftarrow [\mathbf{1}_{2n-1} - \mathbf{B}''\boldsymbol{\omega}'']. \tag{39}$$

Example 1 A simple test, shown in Figure 4, exemplifies some of the principles above. We reconstructed a given 4-point mask, which does not satisfy the necessary (for ISS) condition for smoothness described in Equation (7a). We ran the greedy method with described in Equation (18) either with or without the invariance constraint in Equation (35).

Input. Apply three subdivision iterations, $\boldsymbol{f}^{k+1} = \boldsymbol{p} \star [\boldsymbol{f}^k]_2^{\uparrow}, k = 0, 1, 2,$ with $\boldsymbol{f}^0 = \{0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\}$, and $\boldsymbol{p} = \{-0.08\ 0\ 0.54\ 1\ 0.54\ 0\ -0.08\}$, such that $\boldsymbol{f}^3$ is the third level refinement of the base function of $\boldsymbol{p}$. Take $\boldsymbol{f}^3$ as the input.

Coarse points. The down-sampling-by-8 of $\boldsymbol{f}^3$.

Fine points. All the input points.

Template rules. We choose the template masks as template rules, according to Remark 5, where $\boldsymbol{\phi}^{[4]}$ spans the even rule and the rest span the odd rule:

- $\boldsymbol{\phi}^{[4]} = \{1\}$, ω_4 unknown.
- $\boldsymbol{\phi}^{[3]} = \{1, 0\}$, $\boldsymbol{\phi}^{[5]} = \{0, 1\}$, $\omega_1 = \omega_2 = 0.5$.
- $\boldsymbol{\phi}^{[2]} = \{1, 0\}$, $\boldsymbol{\phi}^{[6]} = \{0, 1\}$, ω_2, ω_6 unknown.
- $\boldsymbol{\phi}^{[1]} = \{-1, 0, 0, 0\}$, $\boldsymbol{\phi}^{[7]} = \{0, 0, 0, -1\}$, ω_1, ω_7 unknown.

Note that these template rules are dependent and are not the orthogonal set mentioned in Remark 5, and therefore, the solution $\boldsymbol{\Omega}_{(3,0)}$ does not coincide with the shape of the scheme (those template rules corresponding to the unknown components *are* orthogonal). Also, the known tension components, ω_1 and ω_2, are handled according to Equation (39).

The non-invariant solution completely reconstructs the original input, as shown in Figure 4(a). It is identical for all three iterations, namely, $\boldsymbol{\omega}_i = \{0.08\ \ 0.04\ \ (0.5)\ \ 1\ \ (0.5)\ \ 0.04\ \ 0.08\}$, for

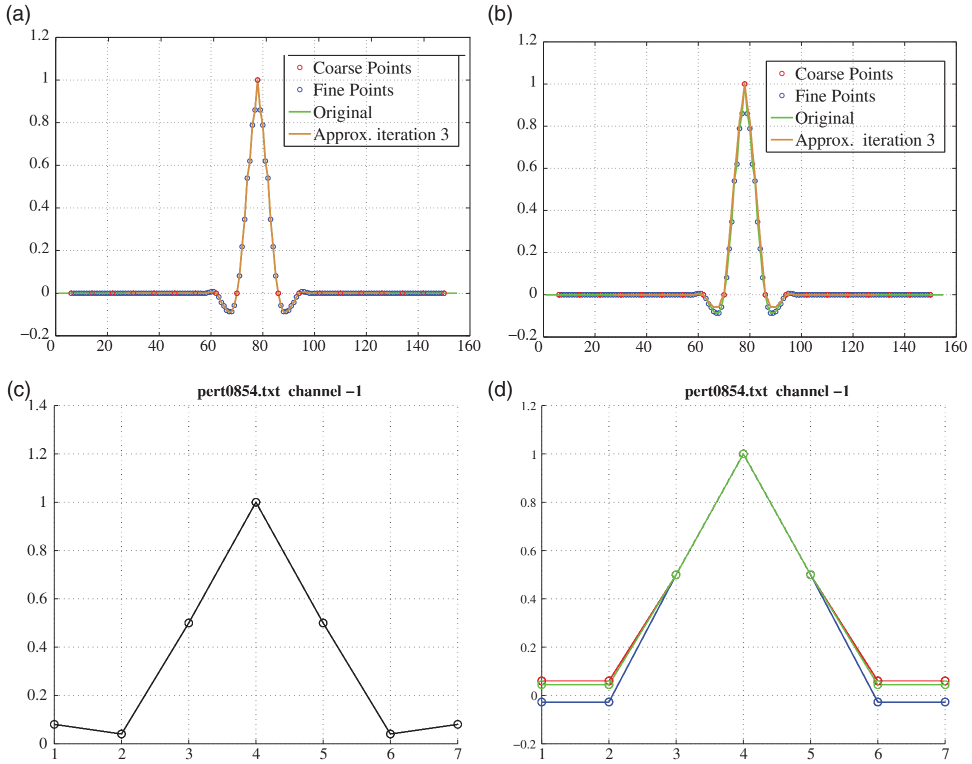

Figure 4. A 4-point reconstruction: the mask and the approximation corresponding to the non-invariant (perfect) and the invariant (imperfect) solutions. (a) Perfect reconstruction. (b) Perfect reconstruction solution. The tension vector parameter in all levels coincide. (c) Invariant reconstruction. (d) Invariant reconstruction solution. Different tension vector parameter for each level.

$i = 1, \ldots, 3$, as shown in Figure 4(b). This solution represents the original 4-point mask $\boldsymbol{p} = \{-0.08\ 0\ 0.54\ 1\ 0.54\ 0\ -0.08\}$.

The invariant solution, however, does not reconstruct the original input accurately, as shown in Figure 4(c). It is different at each level, as shown in Figure 4(d). The last iteration is $\boldsymbol{\omega}_3 = \{0.06\ \ 0.06\ \ (0.5)\ \ 1\ \ (0.5)\ \ 0.06\ \ 0.06\}$, which represents the smooth 4-point mask $\boldsymbol{p} = \{-0.06\ 0\ 0.56\ 1\ 0.56\ 0\ -0.06\}$.

4.4 *Weighted least-squares*

We incorporate a weight matrix to the least-squares systems above. Following the weighted least-squares principles, we replace $\mathbf{A} \leftarrow \mathbf{W}\mathbf{A}, \boldsymbol{f}^{k+1} \leftarrow \mathbf{W}\boldsymbol{f}^{k+1}$ in Equation (18) and solve the weighted normal equations

$$\mathbf{W}\mathbf{A}\boldsymbol{\omega} \simeq \mathbf{W}\boldsymbol{f}^{k+1}, \tag{40a}$$

$$\mathbf{A} = \mathcal{A}(\mathcal{H}, \boldsymbol{f}^{k}). \tag{40b}$$

Note that $\mathbf{B}$ in the constraint system (35) is unchanged.

We allow several options for the weights:

(a) Application-related filters (e.g. music, Section 7).
(b) Special symmetry considerations (Section 5.2).

(c) MLS, where the window is weighted by a Gaussian (Section 4.5).
(d) Window-proportional weights (larger weights for larger intervals).
(e) Multiple-level mapping weight-options, in conjunction with the 'tree regression' model (Section 3).

4.5 *MLS connections*

The discussion so far assumes a uniform scheme (see definition in Section 1.2). This means that when solving the least-squares system (18), we assume that for each level $k \geqslant 0$ the subdivision scheme is the same for all locations of $\boldsymbol{f}^k$, namely for $\{\boldsymbol{f}_j^k \in \mathbb{R}, j \in \mathbb{Z}\}$.

However, we suggest to reconstruct a *piecewise uniform* scheme by sliding a window along the free parameter axis and solve the uniform least-squares system in each window separately. Each location yields a separate tension vector solution $\boldsymbol{\omega}$ and we can track its evolution over time (or parameter, in general). The collection of all vectors $\boldsymbol{\omega}$ represents the reconstructed piecewise uniform scheme.

For each window, we solve the system (40) with a weight matrix, $\mathbf{W}$, that localizes the solution to the middle of the window, x_m, as the diagonal matrix with $\mathrm{w}_{ii} = \mathrm{e}^{-\|x_i - x_m\|^2/2}$.

This approach, of sliding a weighted least-squares window, resembles the MLS system [14] except that:

(a) It reconstructs the subdivision tension vector instead of the polynomial coefficients.
(b) We use it to track the signature $\boldsymbol{\omega}$ along x, in addition to approximation.

5. Criteria and solution-improvements

We apply various criteria on the different levels of the masks for general purposes (e.g. music, Section 7). Here, we provide several such criteria for the extracted mask (15a), $\boldsymbol{p}(\boldsymbol{\omega}) = \sum_{j=1}^{m} \omega_j \boldsymbol{\phi}^{[j]}$. We consider the multiple-level setup in Equation (27), with $L+1$ data levels $\{\boldsymbol{f}^{k+i}\}_{i=0}^{L}$ and L masks $\{\boldsymbol{p}(\boldsymbol{\omega})_{k+i}\}_{i=1}^{L}$. For simplification, we refer to the orthogonal template mask setup described in Remark 5, where the coefficient vector $\boldsymbol{\omega}$, derived by Equation (20), completely represents $\boldsymbol{p}(\boldsymbol{\omega})$. This allows applying the criteria, which should generally be applied to $\{\boldsymbol{p}(\boldsymbol{\omega})_{k+i}\}_{i=1}^{L}$, to the coefficient vectors $\boldsymbol{\Omega}_{(L,k)} = \{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$, instead.

We track these criteria along the free parameter (optionally time) axis by sliding a window, as explained in Section 4.5, and also use them to improve and constrain the solution.

5.1 *The criteria*

We propose extracting several criteria, such as stationarity, variation and symmetry, from the non-stationary scheme derived by the regression model.[15] These criteria values are tracked along a sliding window on the free parameter (or time) axis. The values are 2D plotted against the free parameter, when using sliding window, or against the distance from the original piece, if perturbed. When both slides perturbed, a 3D plot shows the criterion value (Z) against both the free parameter (X) and distance (Y).

Most of the criteria reflect specific relations between the masks at different levels $\boldsymbol{\Omega}_{(L,k)} \equiv \{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$, where a lower value of the criterion corresponds to a higher match, or 'quality'. The stationarity criterion measures the distances between the masks at different levels. The symmetry criterion measures the (signed or unsigned) distance between reflected (left and right) components of the mask (at each level). It indicates the distribution of the tension vector parameter or the dominant side of the mask and tells *whether points are affected more by their preceding or succeeding values*. The minimal (best) stationarity and symmetry values equal 0, where the mask is perfectly stationary and symmetric. The approximation residual criterion measures the least-squares error norm of Equation (18). The following are the explicit mathematical expressions and more details of these criteria. We also suggest a few more criteria (e.g. invariancy and variation) and show how to improve the solution of the least-squares system (18) by using these criteria.

(a) *Stationarity*: The distance between the levels of the masks, measured by standard deviation

$$M(\boldsymbol{\Omega}) = \frac{1}{m}\sum_{j=1}^{m}(\sigma(\{(\boldsymbol{\omega}_i)_j\}_{i=1}^{L})), \tag{41}$$

or the non-negative measure, which is suffice in this case

$$M(\boldsymbol{\Omega}) = \|\{\sigma(\{(\boldsymbol{\omega}_i)_j\}_{i=1}^{L})\}_{j=1}^{m}\|_2. \tag{42}$$

(b) *Symmetry*: The *signed* distance between reflected (left and right) components of the masks

$$M(\boldsymbol{\Omega}) = \frac{1}{L}\sum_{i=1}^{L}\left(\sum_{j=1}^{\lfloor m/2\rfloor}(\boldsymbol{\omega}_i)_{\lfloor m/2\rfloor+j} - (\boldsymbol{\omega}_i)_{\lfloor m/2\rfloor-j}\right) = \frac{1}{L}\sum_{i=1}^{L}\mathbf{D}\boldsymbol{\omega}_i, \tag{43}$$

where $\mathbf{D}$ is a square, even dimensions, matrix with 1's and -1's on its primary and secondary diagonals, respectively. Here, the orthogonal template masks in Remark 5 can be relaxed to a symmetric set, where the template masks j and $m-j+1$ are reflected.

(c) *Smoothness*: The variations of the mask vector at each level.

$$M(\boldsymbol{\Omega}) = \frac{1}{L}\{\|\nabla(\mathrm{sign}(\nabla(\{(\boldsymbol{\omega}_i)_j\}_{j=1}^{m})))\|_1\}_{i=1}^{L}. \tag{44}$$

Here, the orthogonal template masks in Remark 5 can be relaxed to a non-negative set.

(d) *Invariancy*: A signed difference between each rule components sum and '1'. In accordance with Equation (61), (Equation (34) or (35) in $L=1$), measure

$$M(\boldsymbol{\Omega}) = \mathbf{P}^{(L,k)}(\boldsymbol{\Omega})\mathbf{1}_n - \mathbf{1}_{2^L n} = \mathbf{B}\boldsymbol{\Omega} - \mathbf{1}_{2^L n}. \tag{45}$$

(e) *Approximation residual* (*least-squares error*) : In accordance with Equation (57) (Equation (16) or (18) in $L=1$), measure

$$M(\boldsymbol{\Omega}) = Res(\boldsymbol{\Omega}) = \boldsymbol{f}^{k+L} - \mathbf{P}^{(L,k)}(\boldsymbol{\Omega})\boldsymbol{f}^{k} = \boldsymbol{f}^{k+L} - \mathbf{A}^{(L,k)}\boldsymbol{\Omega}. \tag{46}$$

Given a measure $M(\boldsymbol{\Omega})$, we also define the corresponding non-negative measure $F(\boldsymbol{\Omega})$

$$F(M(\boldsymbol{\Omega})) \equiv F(\boldsymbol{\Omega}) \equiv \|M(\boldsymbol{\Omega})\|_2. \tag{47}$$

Other criteria. Other measures related to the relations between the minimum L_2 norm solution (20) and the shift-invariant solution in Equation (35), and the relations between the symmetry (43) and the invariancy (45) criteria, can give interesting indications of the data [8].

Solution-improvements. In case of redundancy in system (18), we suggest to pick, out of all possible least-squares solutions, the best solution in terms of the criteria (42), (43) or (44). To achieve that best solution, $\boldsymbol{\Omega}_{\text{lsq}}^{F}$, we begin with the minimum L_2 norm solution $\boldsymbol{\Omega}_{\text{lsq}}^{\star}$ in Equation (20), and then solve the following constrained minimization problem using $F(\boldsymbol{\Omega})$ given in Equation (47)

$$\boldsymbol{\Omega}_{\text{lsq}}^{M} = \min_{\mathbf{A}\boldsymbol{\Omega}=\mathbf{A}\boldsymbol{\Omega}_{\text{lsq}}^{\star}}\{F(\boldsymbol{\Omega})\}. \tag{48}$$

5.2 *Symmetry*

Aside from the symmetry criterion given above, several methods are proposed herein to check and impose the symmetry of the underlying masks, while solving either Equation (18) or (40) (with or without the extensions and constraints proposed throughout Section 4):

(a) Incorporate weights on the vector variable $\boldsymbol{\omega}$, such that each component is assigned with a different weight (as opposed to the weighted least-squares, which weights the equations).
(b) To force symmetry on $\boldsymbol{\omega}$, impose the constraint system $\mathbf{D}\boldsymbol{\omega} = \boldsymbol{\varepsilon}$, or $\mathbf{D}\boldsymbol{\omega} \leq \boldsymbol{\varepsilon}$, with $\mathbf{D}$ defined as in Equation (43) and $\boldsymbol{\varepsilon}$ is a vector of small values.
(c) Turn things around: Find a minimum to the function $\mathbf{D}\boldsymbol{\omega}$, under the constraint posed by the original system (18) or (40), assuming that the latter has redundancy (see Equation (48)).
    Both in (b) and (c) above, the invariance constraint (35) and the symmetry related constraints can be imposed simultaneously, by concatenating the equality constraints systems involved.
(d) Create two masks, with left and right asymmetric support, by building asymmetric sets of template masks $\{\boldsymbol{\phi}^{[j]}\}_{j=1}^{m}$. Then, examine which one approximates better.
(e) Track a symmetry measure (43) (Section 5.1). In addition, the norm (the unsigned symmetry measure) is applied to improve the symmetry of the solution using Equation (48).

6. Theoretical extensions and generalized views

This section is intended to give an additional, theoretical, angle on the 'subdivision regression' model and can serve for future development. We provide setups yielding an MRA-related view on the model (Sections 6.1 and 6.2) and specify explicitly its L-levels extension (Section 6.3).

6.1 *Deviated vector-perturbed scheme*

Using the 'subdivision regression' model, the following setup is relevant when there is a prior knowledge of the basic smoothness of the given data, in terms of some *vector*-perturbed smooth subdivision $\boldsymbol{p}(\boldsymbol{v})$ (knowing the template masks without necessarily knowing the perturbation vector parameter values). In the following, we use the definitions given in [9] (synthesis) for the analysis purposes of this paper.

Let $\mathfrak{h}$ be a set of m template masks. Let $\boldsymbol{p}(\boldsymbol{x})$ be the corresponding perturbed subdivision scheme as in Equation (15a), with tension vector parameter $\boldsymbol{x} \in \mathbb{R}^m$. Suppose that $\boldsymbol{p}(\boldsymbol{x})$ is a $C^d SS$, for $d > 0$ and $\boldsymbol{x} \in (\boldsymbol{x}_a, \boldsymbol{x}_b)$, $\mathfrak{h}$ are the *smoothness template masks*. Denote by $\mathfrak{g}$ an additional set of l template masks, corresponding to $\boldsymbol{q}(\boldsymbol{y})$, $\boldsymbol{y} \in \mathbb{R}^l$. $\mathfrak{g}$ are the *noise template masks*. Let $\tilde{\mathfrak{h}} \equiv \mathfrak{h} \cup \mathfrak{g}$, $\boldsymbol{\omega} \equiv \boldsymbol{x} \cup \boldsymbol{y}$, $\tilde{m} = m + l$ and $M_{\tilde{\mathfrak{h}}} = [M_{\mathfrak{h}}|M_{\mathfrak{g}}]$. $\tilde{\mathfrak{h}}$ are the *smoothness-and-noise template masks* and $\boldsymbol{\omega}$ the *smoothness-and-noise tension vector parameter*. We look for $\tilde{\boldsymbol{p}}(\boldsymbol{\omega})$

$$\tilde{\boldsymbol{p}}(\boldsymbol{\omega}) = \boldsymbol{p}(\boldsymbol{x}) + \boldsymbol{q}(\boldsymbol{y}), \tag{49}$$

where $\boldsymbol{p} \equiv \boldsymbol{p}(\boldsymbol{x})$ may be considered as the mask of the *smoothness scheme* $S_{\boldsymbol{p}}$, $\boldsymbol{q} \equiv \boldsymbol{q}(\boldsymbol{y})$ may be considered as the mask of the *noise (deviation) scheme* $S_{\boldsymbol{q}}$ and $\tilde{\boldsymbol{p}} \equiv \tilde{\boldsymbol{p}}(\boldsymbol{\omega})$ may be considered as the mask of the *smoothness-and-noise scheme* $S_{\tilde{\boldsymbol{p}}}$ (see Section 6.2).

Then, the solution to the least-squares system (18) (with or without constraints) $\mathbb{R}^{\tilde{m}} \ni \boldsymbol{\omega}_1 \equiv \boldsymbol{x}_1 \cup \boldsymbol{y}_1$ gives a certain $\boldsymbol{p}(\boldsymbol{x}_1)$ and a deviation from it, $\boldsymbol{q}(\boldsymbol{y}_1)$ ($\boldsymbol{y}_1$ in the direction of $\mathfrak{g}$). If $\boldsymbol{x}_1 \in (\boldsymbol{x}_a, \boldsymbol{x}_b)$ and the approximation error is small enough, then we might consider $\boldsymbol{x}_1$ as smoothness coefficients and $\boldsymbol{y}_1$ as noise coefficients – the deviation from $\boldsymbol{p}(\boldsymbol{x})$-smoothness ($\boldsymbol{p}(\boldsymbol{x}_1)$ in practice). We shall call $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ *the smoothness and the noise tension vector parameters (or measures)*, respectively. We refer to $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ as our data signatures.

Let $\mathcal{H}$, $\mathcal{G}$ and $\tilde{\mathcal{H}} = \mathcal{H} \cup \mathcal{G}$ be the corresponding sets of convolution matrices for $\mathfrak{h}$, $\mathfrak{g}$ and $\tilde{\mathfrak{h}}$, namely, the *smoothness template matrices*, *noise template matrices* and the *smoothness-and-noise template matrices*, respectively. Let $M_{\tilde{\mathcal{H}}} = [M_{\mathcal{H}}|M_{\mathcal{G}}]$. Define

$$\mathbf{A}_p \equiv \mathcal{A}(\mathcal{H},\boldsymbol{f}^k), \quad \mathbf{A}_q \equiv \mathcal{A}(\mathcal{G},\boldsymbol{f}^k), \quad \mathbf{A} \equiv \mathcal{A}(\tilde{\mathcal{H}},\boldsymbol{f}^k) = [\mathbf{A}_p|\mathbf{A}_q], \tag{50a}$$

$$\mathbf{B}_p \equiv \mathcal{A}(\mathcal{H},\mathbf{1}_n), \quad \mathbf{B}_q \equiv \mathcal{A}(\mathcal{G},\mathbf{1}_n), \quad \mathbf{B} \equiv \mathcal{A}(\tilde{\mathcal{H}},\mathbf{1}_n) = [\mathbf{B}_p|\mathbf{B}_q]. \tag{50b}$$

While solving the least-squares system (18) with the invariance constraint (35) assigned with $\mathbf{A}$ and $\mathbf{B}$ from Equation (50), we can impose a smoothness constraint on $\boldsymbol{p}(\boldsymbol{x})$, such that $\boldsymbol{y}$ would be considered a smoothness deviation. Note that the overall $\tilde{\boldsymbol{p}}(\boldsymbol{\omega})$ could still be smooth but $\boldsymbol{y}$ expresses the deviation from smoothness with respect to $\boldsymbol{p}(\boldsymbol{x})$.

The options to force the smoothness on $\boldsymbol{p}(\boldsymbol{x})$, in order to measure the deviation $\boldsymbol{q}(\boldsymbol{y})$, are

(a) Set a specific $\boldsymbol{p}(\boldsymbol{x}_1)$ using the setup in Section 4.3 (fixing $\boldsymbol{x}_1$).
(b) Require that $\boldsymbol{x} \in (\boldsymbol{x}_a, \boldsymbol{x}_b)$, as a constraint in solving the least-squares system.
(c) Impose the smoothness conditions (10) on $\boldsymbol{x}$. Equation (7a) implies that $\mathbf{B}_p\boldsymbol{x} = \mathbf{1}$. Hence, if we require the invariance condition (35) on the whole solution ($\mathbf{B}\boldsymbol{w} = \mathbf{1}$) then it implies that $\mathbf{B}_q\boldsymbol{y} = \mathbf{0}$ (if the rules of all template masks in $\mathfrak{g}$ sum to zero it is fulfilled automatically).

We may also pose a generalized problem by setting a general template *dictionary* $\tilde{\mathfrak{h}}$ and finding subsets $\mathfrak{h}$ and $\mathfrak{g}$ that yield the best compromise between the smoothness of $\boldsymbol{p}(\boldsymbol{x})$ (as in (b) or (c) above) and the overall approximation error. In this case, $\mathfrak{h}$ and $\mathfrak{g}$ constitute also part of the signature, alongside $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$.

Remark 7 (Non-smooth perturbation values) As a special case, we can choose $\mathfrak{g} \subseteq \mathfrak{h}$, such that $\tilde{\mathfrak{h}}$ is a dependent set of template masks. If no constraints are imposed, then all the information is in $\boldsymbol{x}$ ($\boldsymbol{\omega} = \boldsymbol{x}, \boldsymbol{y} = \mathbf{0}$), which is not necessarily in the smoothness range of $\boldsymbol{p}$. However, if imposing (b) above, then, according to Equation (49), the extracted mapping $\tilde{\boldsymbol{p}}(\boldsymbol{\omega})$ is

$$\tilde{\boldsymbol{p}}(\boldsymbol{\omega}) = \boldsymbol{p}(\boldsymbol{x}) + \boldsymbol{q}(\boldsymbol{y}) = \boldsymbol{p}(\boldsymbol{x}) + \boldsymbol{p}(\boldsymbol{y}), \tag{51}$$

($= \boldsymbol{p}(\boldsymbol{x}+\boldsymbol{y})$ if $\mathfrak{g} = \mathfrak{h}$) and $\boldsymbol{y}$ is the deviation from the smoothness range of $\boldsymbol{p}$.[16]

This special case may be extended by taking $\mathfrak{h}, \mathfrak{g}$ such that $\mathfrak{h} \cap \mathfrak{g} \neq \emptyset$.

6.2 *Smoothness-and-noise decomposition: MRA view*

We present the ‘subdivision regression’ model as an MRA decomposition. Recall the MRA reconstruction

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k + \mathbf{Q}\boldsymbol{d}^k, \tag{52}$$

in which the data space is enlarged from level to level, by the addition of linear combinations $\boldsymbol{d}^k$ of the wavelets basis (the columns of $\mathbf{Q}$). We compare this to the approach presented in Section 6.1. Referring to the specific setup and definitions in Section 6.1, we can write the subdivision regression system (18), assigned with the least-squares solutions, as

$$\boldsymbol{f}^{k+1} \simeq \mathbf{A}\boldsymbol{\omega}_1 = \mathbf{A}_p\boldsymbol{x}_1 + \mathbf{A}_q\boldsymbol{y}_1, \tag{53}$$

where $\boldsymbol{\omega}_1 = \{\boldsymbol{x}_1, \boldsymbol{y}_1\} \equiv \{\boldsymbol{\omega}_p, \boldsymbol{\omega}_q\}$.

If $\boldsymbol{f}^{k+1} \in \mathrm{Im}(\mathbf{A})$, then this least-squares equality is obviously a true equality. According to Remark 11, $\mathbf{A}_q$ can be designed to complement $\mathbf{A}_p$ to $\mathbb{R}^{2n-1}$, such that always $\boldsymbol{f}^{k+1} \in \mathrm{Im}(\mathbf{A})$, and Equation (53) can be rewritten as Equation (18) with an equality.

Here, $\boldsymbol{x}_1$ are the smoothness coefficients, selecting from $\mathbf{A}_p$, whose columns are the smoothness template masks matrices $\mathcal{H}$ operated on $\boldsymbol{f}^k$. $\boldsymbol{y}_1$ are the differences coefficients, selecting from $\mathbf{A}_q$, whose columns are the deviation template masks matrices $\mathcal{G}$ operated on $\boldsymbol{f}^k$. We refer to $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ as measures for the subdivision smoothness and noise, respectively.

As in MRA, here also the high-resolution data ($\boldsymbol{f}^{k+1}$) and the basis functions (the columns of $\mathbf{A}_p$ and $\mathbf{A}_q$) are known, whereas the smoothness and details coefficients are to be found. As opposed to traditional MRA, however, here $\mathbf{A}_p$ and $\mathbf{A}_q$ depend on the (given) data $\boldsymbol{f}^k$.

Based on Equation (49) (or by $\mathbf{A}_p\boldsymbol{x}_1 = \mathbf{P}(\boldsymbol{p}(\boldsymbol{x}_1))\boldsymbol{f}^k$, $\mathbf{A}_q\boldsymbol{y}_1 = \mathbf{P}(\boldsymbol{q}(\boldsymbol{y}_1))\boldsymbol{f}^k$ in Equation (53)), the least-squares primal form (16) turns to

$$\boldsymbol{f}^{k+1} \simeq \mathbf{P}(\tilde{\boldsymbol{p}}(\boldsymbol{\omega}_1))\boldsymbol{f}^k = \mathbf{P}(\boldsymbol{p}(\boldsymbol{x}_1) + \boldsymbol{q}(\boldsymbol{y}_1))\boldsymbol{f}^k = \mathbf{P}(\boldsymbol{p}(\boldsymbol{x}_1))\boldsymbol{f}^k + \mathbf{P}(\boldsymbol{q}(\boldsymbol{y}_1))\boldsymbol{f}^k, \tag{54}$$

or, with $\mathbf{P} \equiv \mathbf{P}(\boldsymbol{p}(\boldsymbol{x}_1))$ and $Q \equiv \mathbf{P}(\boldsymbol{q}(\boldsymbol{y}_1))$, and the assumption $\boldsymbol{f}^{k+1} \in \mathrm{Im}(\mathbf{A})$

$$\boldsymbol{f}^{k+1} = \mathbf{P}\boldsymbol{f}^k + \mathbf{Q}\boldsymbol{f}^k (= (\mathbf{P} + \mathbf{Q})\boldsymbol{f}^k \equiv \tilde{\mathbf{P}}\boldsymbol{f}^k), \tag{55}$$

in accordance with the presentation of the 'generalized perturbed scheme' as an MRA composition [9]. Compared with Equation (52), our case, therefore, uses $\boldsymbol{d}^k = \boldsymbol{f}^k$, and hence, in MRA terms, we work with only $\mathbf{P}$, $\mathbf{Q}$ and $\boldsymbol{f}^k$, where in contrast with MRA, $\mathbf{P}$ and $\mathbf{Q}$ are data dependent. The basic decomposition and composition are described schematically in Figure 5.

Subdivision regression cascades. We can describe the synthesis and analysis cascades corresponding to the 'generalized perturbed schemes' and the 'subdivision regression' models in analogy to Equations (11) and (12), respectively. To avoid confusion, we hereby rename the displacement operator $\mathbf{Q}$ in Equation (11) to $\mathbf{R}$. In the traditional reverse subdivision model, the analysis process is carried out L times by operating a series of operators $\{\mathbf{P}_{L+1-i}\}_{i=1}^{L}$ on $\boldsymbol{f}^{k+L}$. The results are then the complementary basis series $\{\mathbf{R}_{L+1-i}\}_{i=1}^{L}$, the differences series $\{\boldsymbol{d}^{k+L-i}\}_{i=1}^{L}$ and the coarser data series $\{\boldsymbol{f}^{k+L-i}\}_{i=1}^{L}$, from which only $\boldsymbol{f}^k$ is required for reconstruction.

The analogue in our case is operating the analysis process in Section 6.2 L times, yielding $\{\boldsymbol{\omega}_{k+L-i}\}_{i=1}^{L} = \{\boldsymbol{x}_{k+L-i} \,|\, \boldsymbol{y}_{k+L-i}\}_{i=1}^{L}$, by Equation (53). By assigning $\mathbf{P} \leftarrow [\tilde{\mathbf{P}} = \mathbf{P} + \mathbf{Q}]$ in Equation (27), as in Equation (55), this output becomes $\{\mathbf{P}_{L+1-i}, \mathbf{Q}_{L+1-i}\}_{i=1}^{L}$. If there are significant least-squares errors, then they still need to be modelled by some displacement system $\{\mathbf{R}_{L+1-i}\}_{i=1}^{L}$ and $\{\boldsymbol{d}^{k+L-i}\}_{i=1}^{L}$, in addition to the $\{\mathbf{P}_{L+1-i}, \mathbf{Q}_{L+1-i}\}_{i=1}^{L}$.

The synthesis and analysis cascades corresponding to the 'generalized perturbed schemes' and the 'subdivision regression' models are presented in Figure 6(a) and 6(b), respectively. As a reference, Figure 6(c) and 6(d) shows the standard MRA filter banks [18, pp. 255–257], which we present here as tree paths with non-stationary masks.

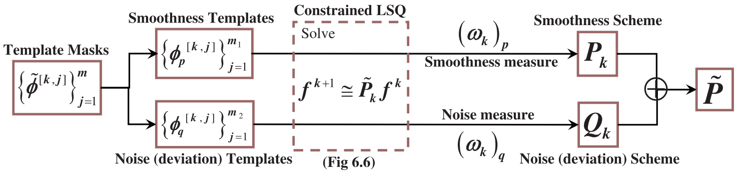
Template Masks
$\left\{\tilde{\phi}^{[k,j]}\right\}_{j=1}^{m}$
Smoothness Templates
$\left\{\phi_p^{[k,j]}\right\}_{j=1}^{m_1}$
Noise (deviation) Templates
$\left\{\phi_q^{[k,j]}\right\}_{j=1}^{m_2}$
Constrained LSQ
Solve
$f^{k+1} \cong \tilde{P}_k f^k$
(Fig 6.6)
$(\omega_k)_p$
Smoothness measure
Noise measure
$(\omega_k)_q$
Smoothness Scheme
P_k
Noise (deviation) Scheme
Q_k
$\tilde{P}$

Figure 5. 'Subdivision regression': smoothness-and-noise decomposition and composition.

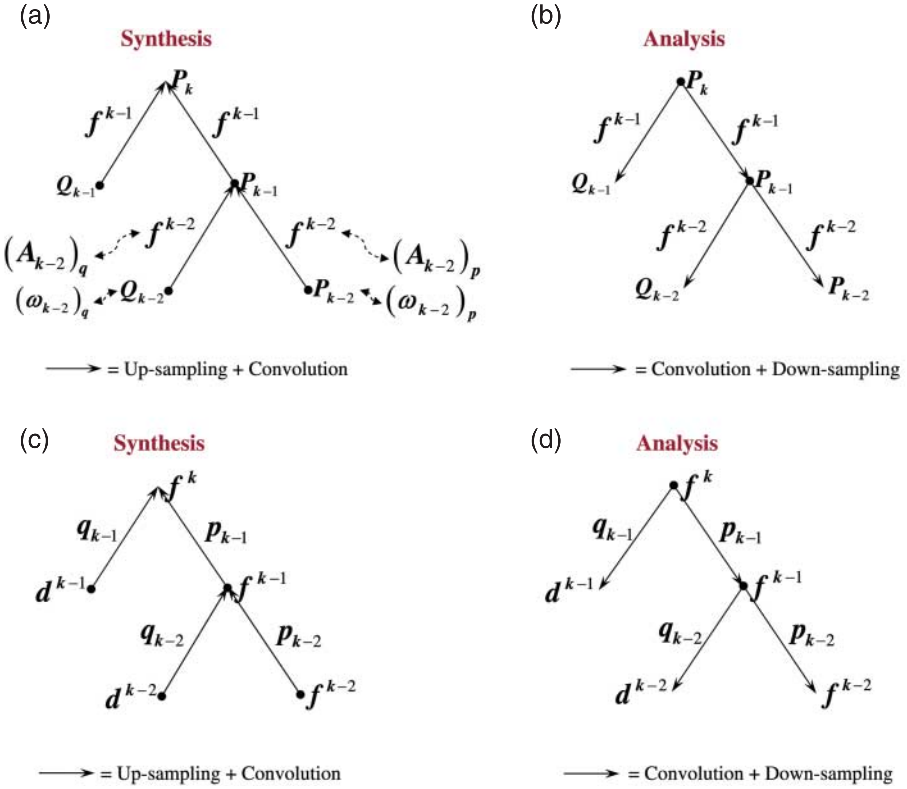

Figure 6. (a and b) The ‘generalized perturbed schemes’ (synthesis) and ‘subdivision regression’ (analysis) cascades. (c and d) The analogous MRA filter banks [18] corresponding to Equations (11) and (12), respectively. (a) New MRA cascades: synthesis. (b) New MRA cascades: analysis. (c) Traditional MRA cascades: synthesis. (d) Traditional MRA cascades: analysis.

6.3 *A generalized (non-greedy) model*

As mentioned, to yield the least-squares solution for Equation (23), given only $\boldsymbol{f}^k$ and $\boldsymbol{f}^{k+L}$, the one-level least-squares system (18) can be generalized. Recall that the solution is the tension matrix parameter $\boldsymbol{\Omega}_{(L,k)} \in \mathbb{R}^{m\times L}$, whose columns are the L tension vector parameters $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$.

We generalize Definition 1, as follows:

Definition 2 *Let $\boldsymbol{J} \equiv (J_1, J_2, \ldots, J_L) \in \{1, \ldots, m\}^L$. For a given vector $\boldsymbol{c} \in \mathbb{R}^n$ and template matrices $\mathcal{H} \equiv \{\boldsymbol{\Phi}^{[j]}\}_{j=1}^{m}$, denote by $\boldsymbol{A} \equiv \mathcal{A}(L, \mathcal{H}, \boldsymbol{c}) \in \mathbb{R}^{2^L n\times m^L}$ the tensor whose columns are $\{\prod_{i=1}^{L} \boldsymbol{\Phi}^{[J_i]}\boldsymbol{c}\}_{\boldsymbol{J}\in\{1,\ldots,m\}^L} \in \mathbb{R}^{2^L n}$, namely, the $\boldsymbol{J}^{th}$ column of the tensor $\boldsymbol{A}$ is*

$$[\boldsymbol{A}]_{\boldsymbol{J}} = \left(\prod_{i=1}^{L} \boldsymbol{\Phi}^{[J_i]}\right)\boldsymbol{c}, \quad \boldsymbol{J} \in \{1, \ldots, m\}^L. \tag{56}$$

($\mathcal{A}(1, \mathcal{H}, \boldsymbol{c}) = \mathcal{A}(\mathcal{H}, \boldsymbol{c})$ for $\mathcal{A}$ in Definition 1). Note that if $\boldsymbol{f}^k \in \mathbb{R}^n$ then $\boldsymbol{\Phi}^{[J_i]} \in \mathbb{R}^{(2^i n-1)\times(2^{(i-1)}n)}$. If there is a different set of template masks for each level, then $\boldsymbol{\Phi}^{[J_i]}$ should be replaced by $\boldsymbol{\Phi}^{[i,J_i]}$.

The linear system (18) then becomes a multilinear system

$$\left(\mathbf{P}^{(L,k)}(\boldsymbol{\Omega}_{(L,k)})\boldsymbol{f}^k \simeq \boldsymbol{f}^{k+L} \Leftrightarrow\right) \mathbf{A}\boldsymbol{\Omega}_{(L,k)} \simeq \boldsymbol{f}^{k+L}\ ,\ \boldsymbol{f}^{k+L} \in \mathbb{R}^{2^L n}, \boldsymbol{\Omega}_{(L,k)} \in \mathbb{R}^{m\times L}, \tag{57a}$$

$$\mathbf{A} \equiv \mathbf{A}^{(L,k)} \equiv \mathcal{A}(L, \mathcal{H}, \boldsymbol{f}^k) \in \mathbb{R}^{2^L n \times m^L}. \tag{57b}$$

($\mathcal{A}(1, \mathcal{H}, \boldsymbol{f}^k) = \mathcal{A}(\mathcal{H}, \boldsymbol{f}^k)$ for $\mathcal{A}$ in Equation (19)).
Note that the $\boldsymbol{J}$th column of $\mathbf{A}$ is of size $2^L n$ in the form

$$[\mathbf{A}]_{\boldsymbol{J}} = \left(\prod_{i=1}^{L} \boldsymbol{\Phi}^{[J_i]}\right)\boldsymbol{f}^k, \quad \boldsymbol{J} \in \{1, \dots, m\}^L. \tag{58}$$

$\mathbf{A}$ is a tensor of size $2^L n \times m^L$ whose ith 'row' $[\mathbf{A}]^i$ is a tensor of size m^L. Equation (57) contains $2^L n$ L-linear (scalar) equations of size m^L, one per 'row' $[\mathbf{A}]^i$

$$[\mathbf{A}]^i \boldsymbol{\Omega}_{(L,k)} \simeq f_i^{k+L}, \quad [\mathbf{A}]^i \in \mathbb{R}^{m^L}, \boldsymbol{\Omega}_{(L,k)} \in \mathbb{R}^{m\times L}, \quad i = 1, \dots, 2^L n. \tag{59}$$

Each column of $\boldsymbol{\Omega}_{(L,k)}$, $\boldsymbol{\omega}_j \in \mathbb{R}^m$, $j \in 1, \dots, L$, corresponds to the appropriate dimension of the tensor 'row' $[\mathbf{A}]^i$. The tensor $\mathbf{A}$ and relations (57) and (58) are depicted in Figure 7.

Example 2 We give simple examples for Equations (57) and (59):

For $L = 1$, Equation (57) is reduced to the special case (18) of $2n$ linear equations, where $[\mathbf{A}]^i$, $i = 1, \dots, 2n$, is a row of the matrix $\mathbf{A} \in \mathbb{R}^{2n\times m}$ (19). Equation (59) in this case is $[\mathbf{A}]^i \boldsymbol{\omega} \simeq \boldsymbol{f}_i^{k+L}$.

For $L = 2$, there are $4n$ bilinear equations, where each 'row' of the tensor $\mathbf{A} \in \mathbb{R}^{(4n)\times m\times m}$, $[\mathbf{A}]^i$, $i = 1, \dots, 4n$, is in itself an $m \times m$ matrix. Equation (59) in this case is $\boldsymbol{\omega}_2{}^T [\mathbf{A}]^i \boldsymbol{\omega}_1 \simeq \boldsymbol{f}_i^{k+L}$. Methods to solve such bilinear systems are discussed in [2].

Equation (59) implies the generalization of Equation (21) from $L = 1$ to any L:

$$[\boldsymbol{\Omega}_{(L,k)}]^\star_{\text{lsq}} = \underset{\boldsymbol{\Omega}_{(L,k)} \in \mathbb{R}^{m\times L}}{\arg\min} \sum_{i=1}^{2^L n} ([\mathbf{A}]^i \boldsymbol{\Omega}_{(L,k)} - \boldsymbol{f}_i^{k+1})^2. \tag{60}$$

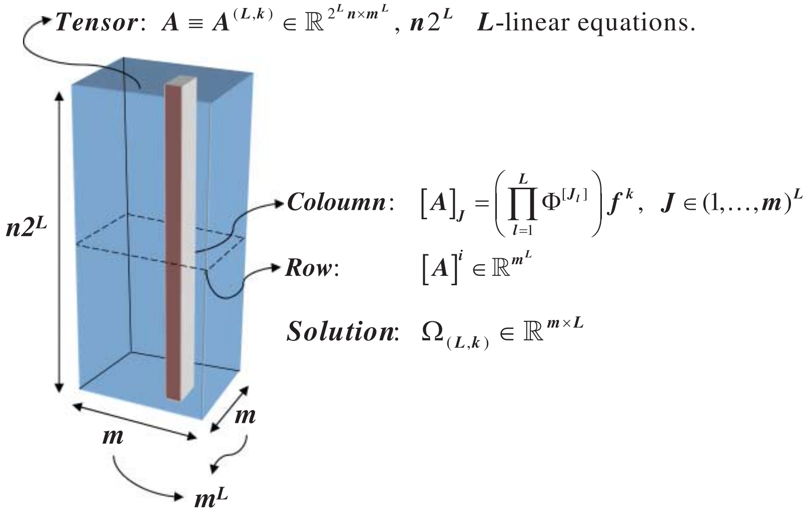
Tensor: $A \equiv A^{(L,k)} \in \mathbb{R}^{2^L n\times m^L}$, $n2^L$ L-linear equations.
$n2^L$
Coloumn: $[A]_J = \left(\prod_{l=1}^{L} \Phi^{[J_l]}\right) f^k$, $J \in (1,\dots,m)^L$
Row: $[A]^i \in \mathbb{R}^{m^L}$
Solution: $\Omega_{(L,k)} \in \mathbb{R}^{m\times L}$
m
m
m^L

Figure 7. The tensor $\mathbf{A}$ in Equations (57) and (58).

The general solution is not stationary, namely, $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$ are generally different. If we add a constraint imposing stationarity, then the multilinear equation (57) becomes a generalized quadratic equation.

As mentioned in Section 4.1, the shift invariance constraints system (35) can be generalized to fit the general non-greedy system (57), as follows:

$$\left(\mathbf{P}^{(L,k)}(\boldsymbol{\Omega})\mathbf{1}_n = \mathbf{1}_{2^L n} \Leftrightarrow\right) \mathbf{B}\boldsymbol{\Omega} = \mathbf{1}_{2^L n}, \tag{61a}$$

$$\mathbb{R}^{2^L n\times m^L} \ni \mathbf{B} \equiv \mathcal{A}(L,\mathcal{H},\mathbf{1}_n). \tag{61b}$$

The $\boldsymbol{J}$th column of $\mathbf{B}$ is of size $2^L n$ in the form

$$\mathbb{R}^{2^L n} \ni [\mathbf{B}]_{\boldsymbol{J}} = \left(\prod_{i=1}^{L} \boldsymbol{\Phi}^{[J_i]}\right)\mathbf{1}_n, \quad \boldsymbol{J} \in \{1,\dots,m\}^L. \tag{62}$$

Now, we can solve the original generalized (tensor) least-squares system (57) under the constraints system (61) to yield a shift-invariant solution $\boldsymbol{\Omega}_{\text{inv}}$.

7. Applications

7.1 *Potential directions*

Traditional models used in time series analysis assume forward dependence and try to fit a model by which data can be retrieved by previous values. This assumption does not hold for some applications in which ‘future’ events influence the design of current (‘present’) events.

One example is in music, where the present harmony is equally influenced by previous and by subsequent notes. Another example is in impedance cardiography signals, where each pulse is also preparing the heart for the next pulse. Such interrelations can be analysed using Fourier transform, which is very useful for analysing global data frequencies in time series/signals.

The new approach suggested in this paper targets these ‘present–future’ relations differently, as it aims at discovering local internal structure of the signal, at different resolution levels. The present work demonstrates (in Section 7.2) the application to music analysis. In a future work, we plan to use the tools developed here for the analysis of cardiography signals and stock graphs (see Section 7.3), and to extend the method to 2D in order to analyse textures. We believe that some classes of textures can be defined, and thus can be analysed by the multilevel approach proposed here.

Compression and coarse points' selection. The compression is a useful byproduct of the analysis methods suggested. It is not the focus of this paper, which rather concentrates on the general approach and on data analysis. Yet, the compression was experimented in the musical context, as explained in Section 7.2. Some general options for reconstruction were tested, based on general coarse values (not necessarily specific to music), such as extremum points. These tests indicated that under some regularity conditions, the proposed compression/decompression approach may apply to general signals, and hence be useful for other applications, as those mentioned above.

The compression is definitely dependent of the coarse points selection, and the coarse points, in turn, depend on the specific application involved. This dependency evolves from the fact that in the methods proposed, the coarse points reflect prominent values within the domain of the specific application and from the fact that the selection of the coarse points decomposes the data into different resolution levels based on intrinsic properties of the data and the application (see Section 7.2). The coarse points are generally values that have the highest significance in their neighbourhood and hence represent it. This significance can be measured only within the context of the application and hence different coarse points will be selected for different domains and applications.

It should also be emphasized that the coarse points' selection depends on whether the application is intended for analysis or for compression/decompression: if it aims at analysis, then the exact values of the selected coarse points are more significant, or more application-dependent, than if they are intended for decompression (restoration). The linkage of coarse points with the application, based on the tailored decomposition into varying-resolution levels, is the very factor allowing the retrieval of masks that carry special meaning in the application's domain. Therefore, the application-dependency enables the retrieval of masks that will serve as features for analysis in the specific domain involved. For instance, Hed [8] uses the analysis models to model rhythm and pitch interrelationships under specific selection of coarse points (see Section 7.2). On the other hand, for compression/decompression, several different options for the coarse points may still have good results for restoration (approximation).

In the context of the importance of the coarse points selection for analysis, it should also be noted that the coarse notes may have influence on the analysis if it aims at tracking the behaviour across levels. To figure this out, recall that the technique in this paper decomposes the given data into varying-resolution levels that constitute the coarse values of one level and the fine values of the next, and then it yields the different operators (mappings/masks) between these data levels. The analysis can then point out relations between these operators, as done in the musical context (see Section 7.2).

In the following, we present the above principles from the perspective of the musical application.

7.2 *Music analysis*

This paper is the mathematical infrastructure for further research conducted in music analysis. For practical evaluations, we refer the reader to separate publications, as detailed below.

General method. The analysis and reconstruction methods proposed in this paper are used in [8,10] to yield methods for music analysis. In these works, the infrastructure methods proposed in here were evaluated separately in the musical context. Hed *et al.* [10] explains the added value for music analysis and includes examples for using the compression/reconstruction methods proposed here in the context of music analysis and compression (Section 4.2.3 in [10]. See below). Hed [8] evaluates further both the analysis and the compression in the musical context (Chapters 9 and 10).

In general terms, Hed *et al.* [8,10] suggest making a parallelism between the rhythm tree and the 'subdivision rules tree' or an analogy between musical beat subdivision and subdivision schemes. That approach has led to music synthesis and analysis tools bearing insights regarding pitch-and-rhythm interrelationships, musical symmetry and musical hierarchy. The pitch-and-rhythm interrelationships findings and case studies presented in [8] point out that the relations between adjacent pitches may be correlated with their timing through simple subdivision masks, coinciding with certain cognitive foundations.

Technically speaking, in those works, the 'subdivision regression' and the 'tree regression' models are respective analogues to the 'pitch independency model' (PIM) and the 'rhythm–pitch dependency model' (RPDM) configurations. Using the symmetry, stationarity and approximation residual criteria, the latter configuration is assessed by a comparison to the former to reveal those characteristics in the piece related to pitch-and-rhythm interrelationships and symmetry patterns. Applying the 'tree regression' model to music revealed that musical pieces can be arranged in increasing resolution levels induced by the interonset intervals and correlated by means of nearly stationary subdivision schemes. As mentioned, the relations found this way between the different levels of the musical data embody influences of ‘future’ notes on current ones.

More specifically, the musical configurations are special instances of the general configuration described in Section 3.4. The musical space is either pitch or time vs. an independent parameter, or pitch vs. time. In the RPDM musical configuration, the *partial* sets of real values within the $L+1$ data levels $\{\boldsymbol{f}^{k+i}\}_{i=0}^{L}$ in Equation (27), as described in Section 3.2, are dictated by the time intervals of the musical notes (‘tree regression’ model). In the PIM musical configuration, the $L+1$ sets data levels $\{\boldsymbol{f}^{k+i}\}_{i=0}^{L}$ in Equation (27) are dictated solely by the order of the notes, without taking their timing and durations into consideration (‘subdivision regression’ model). In both cases, these sets of notes decompose the given musical piece into different resolution levels that constitute either the coarse or the fine notes of the process, recursively. The window sliding (Sections 2.3 and 4.5) is done either by sliding a points’ window or a time window. The reconstructed subdivision tension vector and the corresponding approximation value and error are attributed to the middle index point or to the middle time point. The data weights applied on the least-squares systems are derived from various musical filters (such as boundaries and durational/melodic accentuations). Alternatively, based on (c) and (d) in Section 4.4, the weights are induced by the shape of a Gaussian, which can be interpolated proportional to the size of the interonset intervals (or durations). Other time-proportional weights are also applied.

These methods are also used in [8] to classify pieces generated by the Mozart’s dice game. This categorization separates mathematically, using the analysis models in this paper, between “valid” and “invalid” contradances generated automatically by the dice game. A correlation is found between the quality of the extracted schemes and the musical correctness of the pieces, where the quality is defined using the criteria proposed in this paper.

Compression and coarse points’ selection. As a byproduct of the analysis methods, we refer the reader to a useful example on compression from [10] (Example 4.11). In this example, the coarsest notes (at the first level, k) are selected as the downbeats and the half downbeats (1:4 note compression), the finest notes are simply the piece’s notes, whereas the coarse/fine notes at the next resolution levels are induced by the rhythm tree. As a result, from the analysis aspect, the three-level scheme extracted by the analysis is almost stationary and closely symmetric. Also, from the compression/decompression aspect, the reconstruction (approximation) based on this compression is highly accurate.

Hed [8] presents methods for selecting the coarse points in musical pieces. As being generally values of high significance, examples for such significant values (notes) in the musical domain are melodic and temporal accentuation, peaks, lows and downbeats (as in the compression example mentioned above) and other known musical filters (boundary, etc.).

As mentioned, the significance of the coarse points for analysis is higher than for compression. Indeed, in [8], which uses the analysis models and criteria to point out special relations between rhythm (time) and pitch, the selection of the coarsest points (at the first level, k) as downbeats is a crucial factor in that specific analysis. The notes opening the musical measures have special meaning and prominency. Hed [8] shows that when these notes are selected as the coarsest points and the coarse notes at the next levels are induced by the time intervals (rhythm tree) then the masks extracted at different levels are more correlated. The analysis in that case, therefore, is used as a tool supporting and generalizing musicological and cognitive findings.

7.3 *Concluding remarks: time and non-time signals and applications*

The approach undertaken for music analysis is based on the decomposition of the musical piece into different resolution levels induced by an hierarchy dictated either by the time intervals of the events (‘time hierarchy’ based on the ‘tree regression’ model), or by their plain order in the piece (‘plain hierarchy’ based on the ‘subdivision regression’ model). The analysis yields the corresponding masks mapping between these levels and using several criteria it points out the decomposition yielding the best masks. Such an approach can be applied to time series from other applications. For example, stock graphs can be analysed this way, where the time intervals of an event can be defined as the time period in which the price did not change. The coarsest points (coarse points at the base level) should reflect significant events in the context of trading, and their selection should be part of the research. It would be interesting to see whether known anchor points would yield superior masks, and whether the time hierarchy would be superior over the plain hierarchy, as has been indicated in several cases in music.

As has been shown, the methods can be applied also to functions above free parameters other than time (and moreover, the time itself can be considered a function above a free parameter) [8,10]. For example, curves, surfaces and textures can be analysed this way, based on the examples given in this paper. The ‘tree regression’ and the ‘subdivision regression’ models can be applied by regarding or disregarding the parameter-distances between the values, in analogy to the time intervals consideration in the ‘time hierarchy’ vs. the ‘plain hierarchy’, respectively. Again, the coarsest points and the further decomposition into data levels are yet to be defined and researched, according to the specific application involved.

8. Summary

In this paper, we proposed two subdivision analysis models. The ‘tree regression’ model enables the analysis of patterns and is based on the subdivision rules tree and the {*mask, tree*} (pruned subdivision rules tree) representation of the ‘varying-resolution’ model. The ‘subdivision regression’ model inverses the ‘generalized perturbed scheme’ model and extracts perturbations and displacements depending merely on the input data. We also suggested applying certain general criteria on the parameters extracted by these models that have value in identifying relations between different resolutions of the data. These criteria are proposed for the purpose of data analysis. In specific, these criteria yield features applicable in music analysis and potentially in other areas, such as graphics, texture, cardiography and stocks. Aside from data analysis, we also proposed compression and decompression as byproducts of the two analysis models.

Notes

1. More details and examples related to this paper and to [9,10] appear in [8].
2. The infinite matrix $\mathbf{P}$ can be replaced by a finite bisection if $|\varsigma(\boldsymbol{p})| < \infty$.
3. The definition of the fine data is extended in [9] (reviewed in Section 3.1).
4. In general, a d-subdivision scheme (e.g. ternary) can be described by d^L (e.g. 3^L) rules.
5. Referring to the matrices $\{\boldsymbol{\Phi}^{[j]}\}_{j=1}^{m}$ as to vectors.
6. It is the inverse problem of the reverse subdivision reported in [19]; in their case the scheme and the fine values are given and the coarse notes are extracted by the pseudo-inverse of $\mathbf{P}$, whereas here the mask $\boldsymbol{\omega}$ (the operator $\mathbf{P}(\boldsymbol{\omega})$) is recovered.
7. We allow separating $\boldsymbol{f}^{k+L}$ into different vectors in which case the matrix $\mathbf{A}$ in Equation (18) turns to be a matrix of matrices [8].
8. (a) By Equation (20), the solution to Equation (18) is $\boldsymbol{\omega}^{\star}_{\text{lsq}} = \mathbf{C}^{\dagger}\boldsymbol{u}$, where $\mathbf{C}_{ji} = \mathbf{C}_{ij} = [\mathbf{A}^{\mathrm{T}}\mathbf{A}]_{ij} = (\boldsymbol{f}^k)^{\mathrm{T}}(\boldsymbol{\Phi}^{[i]})^{\mathrm{T}}\boldsymbol{\Phi}^{[j]}\boldsymbol{f}^k$ and $\boldsymbol{u}_i = (\boldsymbol{f}^k)^{\mathrm{T}}(\boldsymbol{\Phi}^{[i]})^{\mathrm{T}}\boldsymbol{f}^{k+1}$. (b) if $\{\boldsymbol{\Phi}^{[j]}\boldsymbol{f}^k\}_{j=1}^{m}$ are independent then $\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$, and the approximation is the projection $\boldsymbol{f}^{k+1}_{\text{approx}} = \mathbf{A}\boldsymbol{\omega}^{\star}_{\text{lsq}} = \mathbf{A}\mathbf{A}^{\dagger}\boldsymbol{f}^{k+1} = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\boldsymbol{f}^{k+1}$.
9. Hence, for independent template masks, $|\boldsymbol{p}(\boldsymbol{\omega})| \geq m \geq 2n - 1$ (see Remark 2).
10. Because the columns of $\mathbf{A}$ depend also on the data. See Equation (19). For instance, the data $\boldsymbol{f}^k$ need not be constant in order for $[\mathbf{A}]_j$ to be independent.
11. The ‘subdivision regression’ model is a special case, where the pruning is at a constant level.

12. Yielding the $2^{L_{\text{real}}}$ real fine values assuming a partial tree takes more than L_{real} iterations.
13. In this case, the coefficients of the solution are the coefficients of the mask.
14. Since different template masks/rules represent (are 'responsible' for) the left and the right side of the mask.
15. When orthogonal template masks or rules are taken, the $\boldsymbol{\omega}$ extracted by Equation (20) then completely represents $\boldsymbol{p}(\boldsymbol{\omega})$. Therefore, the criteria can be applied directly on the vectors $\{\boldsymbol{\omega}_{k+i}\}_{i=1}^{L}$, instead of on $\{\boldsymbol{p}_{k+i}(\boldsymbol{\omega}_{k+i})\}_{i=1}^{L}$.
16. Technically, extend $\boldsymbol{y}$ to include zeros in the entries corresponding to $\mathfrak{h} \setminus \mathfrak{g}$.

References

[1] A. Cohen and B. Matei, *Nonlinear subdivision schemes: Applications to image processing*, in *Tutorials on Multiresolution in Geometric Modelling*, Mathematics and Visualization, A. Iske, E. Quak, and M.S. Floater, eds., Springer-Verlag, Berlin, Heidelberg, 2002, pp. 93–97.

[2] S. Cohen and C. Tomasi, *Systems of bilinear equations*, Tech. Rep. STAN-CS-TR-97-1588, Computer Science Department, Stanford University, CA, April 1997.

[3] N. Dyn and D. Levin, *Subdivision schemes in geometric modeling*, Acta Numer. 11 (2002), pp. 73–144.

[4] N. Dyn, J. Gregory, and D. Levin, *A four-point interpolatory subdivision scheme for curve design*, Comput. Aided Geometr. Des. 4 (1987), pp. 257–268.

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

[6] R. Goldman, *The Fractal Nature of Bezier Curves*, GMP '04: Proceedings of the Geometric Modeling and Processing 2004, IEEE Computer Society, Washington, DC, 2004, p. 3.

[7] M.F. Hassan and N.A. Dodgson, *Reverse subdivision*, in: *Advances in Multiresolution for Geometric Modelling*, N.A. Dodgson, M.S. Floater, and M.A. Sabin, eds., Springer, Berlin, Heidelberg, 2005, pp. 271–283.

[8] S. Hed, *Multiresolution modeling for automated music synthesis and analysis*, Ph.D. thesis, Tel-Aviv University, Israel, 2012.

[9] S. Hed and D. Levin, *Subdivision models for varying-resolution and generalized perturbations*, Int. J. Comput. Math. 88 (2011), pp. 3709–3749.

[10] S. Hed, R.O. Gjerdingen, and D. Levin, *Subdivision schemes and multi-resolution modelling for automated music synthesis and analysis*, J. Math. Music: Math. Comput. Approaches Music Theory Anal. Compos. Perform. 6 (2012), pp. 17–47.

[11] A. Khodakovsky, P. Schröder, and W. Sweldens, *Progressive Geometry Compression*, Proceedings of SIGGRAPH 2000, New Orleans, LA, 2000, pp. 271–278.

[12] S. Lanquetin and M. Neveu, *Reverse Catmull–Clark Subdivision*, The 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2006 (WSCG), Plzen-Bory, 2006, pp. 319–326.

[13] A. Lee, H. Moreton, and H. Hoppe, *Displaced Subdivision Surfaces*, Proceedings of SIGGRAPH 2000, New Orleans, LA, July 2000, pp. 85–94.

[14] D. Levin, *The approximation power of moving least-squares*, Math. Comput. 67(224) (1998), pp. 1517–1531.

[15] D. Levin, *Using laurent polynomial representation for the analysis of non-uniform binary subdivision schemes*, Adv. Comput. Math. Ingenta 11(1) (1999), pp. 41–54.

[16] X.Z. Liang, Y.H. Xue, and Q. Li, *Some applications of loop-subdivision wavelet tight frames to the processing of 3D graphics*, Visual Comput. 27(1) (2010), pp. 35–43.

[17] N. Litke, A. Levin, and P. Schröder, *Fitting Subdivision Surfaces*, Proceedings of the conference on Visualization '01, San Diego, CA, October 2001, pp. 21–26.

[18] S. Mallat, *A Wavelet Tour of Signal Processing (Wavelet Analysis & Its Applications)*, 2nd ed., Academic Press, San Diego, CA, 1999.

[19] 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.

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