

# Gradient Estimation Revitalized

Usman R. Alim, Torsten Möller, *Member, IEEE* and Laurent Condat, *Member, IEEE*

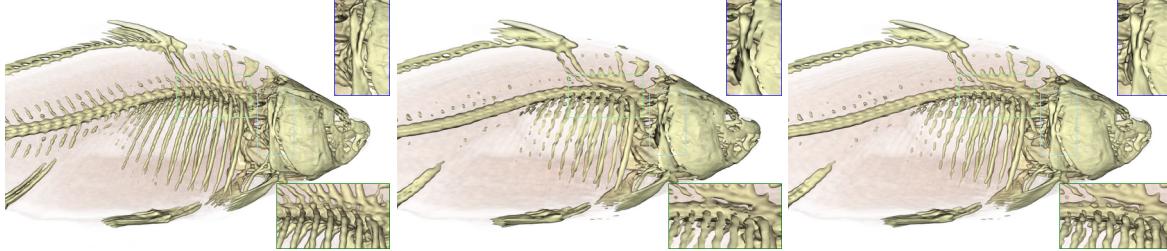


Fig. 1. DVR images of the carp dataset. Left: Original Cartesian ( $256^3$ ) dataset reconstructed using prefiltered tricubic B-splines and shaded using the analytic gradient. Middle: Downsampled Cartesian ( $128^3$ ) dataset reconstructed using prefiltered tricubic B-splines and shaded by interpolating the gradient obtained through a fourth-order central differencing scheme ( $pFIR$ ). Right: The middle dataset shaded using our proposed fourth-order shifted gradient estimation scheme ( $pFIR-s$ ). Notice how the details in the bones and skull are much better preserved as compared to the central-differencing scheme even though the underlying scalar interpolation is the same. This gain in quality comes at no additional cost.

**Abstract**— We investigate the use of a Fourier-domain derivative error kernel to quantify the error incurred while estimating the gradient of a function from scalar point samples on a regular lattice. We use the error kernel to show that gradient reconstruction quality is significantly enhanced merely by shifting the reconstruction kernel to the centers of the principal lattice directions. Additionally, we exploit the algebraic similarities between the scalar and derivative error kernels to design asymptotically optimal gradient estimation filters that can be factored into an infinite impulse response interpolation prefilter and a finite impulse response directional derivative filter. This leads to a significant performance gain both in terms of accuracy and computational efficiency. The interpolation prefilter provides an accurate scalar approximation and can be re-used to cheaply compute directional derivatives on-the-fly without the need to store gradients. We demonstrate the impact of our filters in the context of volume rendering of scalar data sampled on the Cartesian and Body-Centred Cubic lattices. Our results rival those obtained from other competitive gradient estimation methods while incurring no additional computational or storage overhead.

**Index Terms**—Derivative, Gradient, Reconstruction, Sampling, Lattice, Interpolation, Approximation, Frequency Error Kernel



## 1 INTRODUCTION

The quality of volume rendering has been of central interest to our community since the beginning. While in the early days, the trade-off between quality and rendering speed was the defining issue, today the focus is on the ability to quantify the truthfulness of our images using various methods of uncertainty visualization.

One of the defining measures of quality has been the ability to contain the error during function reconstruction. This is a broad field of interest with many different approaches. One staple of understanding of this topic has been Fourier Analysis and the reasoning about band-limited reconstruction, which is based on the insights of Whittaker-Nyquist-Kotelnikov-Shannon [29]. According to this theory, we need an infinitely supported reconstruction kernel (the sinc function to be exact) and we can measure our error relative to that particular reconstruction kernel.

However, we often deal with functions that are not band-limited, and we reconstruct them based on compact reconstruction kernels, like the B-splines. To our rescue, comes a theory developed by Blu *et al.* [4] and Unser [35] among others, that extends the reasoning of

*perfect reconstruction* to arbitrary shift-invariant spaces. These are spaces that are constructed by a single basis function and its shifts. This is exactly the scenario we face in volume rendering.

In our previous work, we have pointed out that, one of the crucial aspects of volume rendering is the accuracy of normal computation since it will mask the appearance of a bad interpolation filter [24]. Even with a good underlying scalar interpolation filter, image quality is very sensitive to the type of normal computation scheme used (Figure 1). Yet, the computation of proper normals remains a very difficult issue, as the performance of normal estimation in our recent work shows [20] where *average angular errors* of 20 to 30 degrees were not uncommon. Therefore, it is crucial to find more accurate and practical ways of estimating the normal for rendering applications.

The design of so-called *digital differentiators* [12, 19, 16] is usually motivated by the fact that the gradient only needs to be accurately determined at the sample points. In volume visualization on the other hand, to compute shadings, we need to accurately estimate derivatives everywhere and not just at the sample points. We are faced with a reconstruction problem where, for efficiency reasons, we would like to choose a compact reconstruction kernel and design a digital derivative filter that enables us to fully exploit the approximation power of the kernel. Very few works have focused on this problem, *a fortiori* on reconstructing continuous gradients on multi-dimensional lattices.

A common method consists in computing exact derivatives of the interpolated function. This approach has been contrasted in [24, 25] with a direct approach, in which a finite-difference-like digital filter is combined with a continuous reconstruction kernel. This direct methodology is preferable, since all the degrees of freedom can be exploited towards a good estimation of the *true* underlying derivative, without being constrained by the reconstruction of the function itself.

• Usman R. Alim and Torsten Möller are with the School of Computing Science, Simon Fraser University, Burnaby BC, Canada, E-mail: {ualim,torsten}@cs.sfu.ca.

• Laurent Condat is with GREYC, a joint CNRS-UCBN-ENSICAEN research unit, 6 Bd du Maréchal Juin, 14050 CAEN Cedex, France, Email: laurent.condat@greyc.ensicaen.fr.

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Hence, a reconstruction space for the derivative can be specified independent of the way the scalar function is interpolated.

Within the Shannon paradigm of bandlimited function, the ideal reconstruction function was shown to be the derivative of the sinc interpolator [2]. However, its slow decay and the ringing artifacts it may introduce, prevent its practical use. Moreover, for non-bandlimited signals, this framework is not valid any more [35]. That is why practitioners rely on convolutions with more localized kernels having compact support. To mitigate the adverse effects of truncating the derivative of the sinc, the idea of windowing was extended to derivative reconstruction [18, 32]. In [2], the tuning of the various parameters of a piecewise cubic derivative filter was discussed. In none of these works, there is an analytic comparison of different filters nor a quantitative analysis of the estimation error.

Once the reconstruction space for the derivative is chosen, e.g. a spline space [33], the scheme, which determines a particular continuous function in this space from the available point samples, has to be designed. To that end, Möller *et al.* developed tools and derived absolute error bounds for the spatial analysis of both interpolation and derivative filters of arbitrary order in [24]. More recently [20], they extended the orthogonal projection (OP) paradigm [34] to the reconstruction of derivatives in given shift-invariant spaces.

A relevant viewpoint to analyse reconstruction is offered by approximation theory. In this perspective, we focus on the approximation order of the derivative reconstruction schemes, that governs how the error behaves asymptotically. To fully exploit the approximation power of a given reconstruction space, it is required to prefilter the data [35]. The interest of prefiltering for the visualization community has been recognized [7, 17], but has not been transposed to derivative reconstruction so far. A notable exception is the recent work of Csébfalvi *et al.* [8] in which they propose FIR derivative prefilters designed to fully exploit the approximation power of the reconstruction space. Our motivation is the same as in [8]. However, we depart from existing approaches to design IIR prefilters with specific properties, either within the OP framework or by designing combinations of interpolation prefilters and finite differences. Our methodology is generic, in the sense that it can be deployed on arbitrary lattices. It is now a wide-spread result that the Body Centered Cubic (BCC) lattice outperforms the Cartesian Cubic (CC) lattice for visualization tasks [22, 14, 26, 31].

Hence, the main contribution of this work is the demonstration that, when reconstructing the gradient continuously in appropriate shift-invariant spaces with specific prefilters, we can obtain normals whose accuracy goes beyond previously known limits. This quality comes without increase in the computational burden. Even more, the efforts for obtaining a good normal estimation provide a good function reconstruction for no additional cost. Incidentally, we break the common belief that the best gradient is obtained by computing the analytic partial derivatives of the reconstructed function.

The outline of this paper is as follows. In Section 2, we recall the basics of reconstruction in shift-invariant box-spline spaces. In Section 3, we depart from previous works by shifting the reconstruction spaces between lattice sites, along the principle directions of the lattice, following an idea originally developed in [3]. We adapt the Hilbert space framework based on orthogonal projection (OP) proposed in [20] to this shifted setting. In Section 4, we propose a new methodology, which combines the optimal approximation capabilities of the OP framework with much lower computation cost and memory requirement. For this, we make use of the recently developed error kernel [5], which allows to accurately quantify the error between the reconstructed gradient and its true underlying counterpart. We detail the methodology in Section 5 with the design of practical schemes for the 3D Cartesian and BCC lattices. Finally, Section 6 is devoted to the experimental illustration of the proposed approaches.

## 2 PRELIMINARIES

Unless otherwise stated, all functions are assumed to be of the form  $f : \mathbb{R}^d \rightarrow \mathbb{C}$  that belong to the Hilbert space  $L^2(\mathbb{R}^d)$ . We denote the inner product between two such functions  $f$  and  $g$  as  $\langle f|g \rangle := \int_{\mathbb{R}^n} f(\mathbf{x})g^*(\mathbf{x})d\mathbf{x}$ , where  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$  and

$g^*(\mathbf{x})$  indicates the complex conjugate of  $g(\mathbf{x})$ . This inner product induces a norm (the  $L^2$ -norm) on any function  $f$ , which we denote as  $\|f\|_{L^2} := \sqrt{\langle f|f \rangle}$ .

The Fourier transform of a function  $f(\mathbf{x})$  is defined as  $\hat{f}(\omega) := \int_{\mathbb{R}^n} f(\mathbf{x}) \exp(-j\omega^\top \mathbf{x})d\mathbf{x}$ , where  $j = \sqrt{-1}$  and  $\omega^\top$  indicates the transpose of  $\omega$ . In  $L^2(\mathbb{R}^d)$ , a useful property is Parseval's theorem [27] which states that  $\|f\|_{L^2}^2 = \frac{1}{(2\pi)^d} \|\hat{f}\|_{L^2}^2$ .

A  $d$ -dimensional lattice  $\mathcal{L}$  generated by the  $d \times d$  matrix  $\mathbf{L} = [\mathbf{l}_1, \dots, \mathbf{l}_d]$  is the set of points given by  $\mathcal{L} := \{\mathbf{Lk} : k \in \mathbb{Z}^d\}$  [11]. We call the column vectors  $\mathbf{l}_i$ , the principal direction of  $\mathcal{L}$ . We denote by  $\mathcal{L}_h$ , a scaled version of the lattice  $\mathcal{L}$  generated by the matrix  $h\mathbf{L}$ , where  $h$  is a positive isotropic scaling parameter. We denote a discrete sequence on lattice  $\mathcal{L}_h$  as  $f[\mathbf{k}]$ , where  $\mathbf{k} \in \mathbb{Z}^d$ . Usually, the sequence will consist of point samples of a continuous function  $f(\mathbf{x})$  (i.e.  $f[\mathbf{k}] = f(h\mathbf{Lk})$ ). Such a sequence can be used as coefficients in a multi-dimensional Fourier series to yield the Fourier transform of the sequence which is given by

$$\begin{aligned} \hat{F}(\omega) &:= \sum_{\mathbf{k} \in \mathbb{Z}^d} f[\mathbf{k}] \exp(-jh\omega^\top \mathbf{Lk}) \\ &= \frac{1}{h^d |\det \mathbf{L}|} \sum_{\mathbf{r} \in \mathbb{Z}^d} \hat{f}(\omega - \frac{2\pi}{h} \mathbf{L}^{-\top} \mathbf{r}), \end{aligned} \quad (1)$$

where the latter equality represents the sampling induced aliasing of the spectrum  $\hat{f}$  on the reciprocal (or dual) lattice generated by the matrix  $(2\pi/h)\mathbf{L}^{-\top}$  [23]. We denote the reciprocal lattice as  $\mathcal{L}_h^\circ$ . To indicate continuous and discrete Fourier transform pairs, we use the notations  $\leftrightarrow$  and  $\overset{L}{\leftrightarrow}$  respectively, e.g.  $f(\mathbf{x}) \leftrightarrow \hat{f}(\omega)$  and  $f[\mathbf{k}] \overset{L}{\leftrightarrow} \hat{F}(\omega)$ .

### 2.1 Shift-Invariant Approximation Spaces

The goal of function reconstruction is to faithfully approximate a function from its discrete measurements on a lattice  $\mathcal{L}_h$ . This is usually achieved by convolving the discrete measurements with a scaled version of a continuous reconstruction kernel  $\psi(\mathbf{x})$ . This process can be written as

$$f(\mathbf{x}) \approx f_{\text{app}}(\mathbf{x}) = \sum_{\mathbf{k}} c[\mathbf{k}] \psi_{h,\mathbf{k}}(\mathbf{x}), \quad (2)$$

where  $\psi_{h,\mathbf{k}}(\mathbf{x}) := \psi(\frac{\mathbf{x}}{h} - \mathbf{Lk})$  is a scaled and shifted version of  $\psi(\mathbf{x})$  and the sequence  $c$  represents discrete linear measurements of  $f$  at the lattice sites — not necessarily the ideal point samples of  $f$ . If however,  $c$  is obtained by point-sampling  $f$  then the reconstruction kernel  $\psi$  is typically chosen such that  $f_{\text{app}}$  exactly interpolates the sample values. Furthermore, when  $f$  is bandlimited, then exact reconstruction is possible if and only if  $\psi(\mathbf{x})$  is chosen to be the *sinus cardinalis* function associated with  $\mathcal{L}_h$  [27]. This method however is not practical because of the infinite support of the sinus cardinalis function. Moreover, when  $f$  is not bandlimited or when the sampling process is non-ideal, it is more desirable to seek an approximation that attempts to minimize the  $L^2$  error  $\|f - f_{\text{app}}\|_{L^2}$ .

Towards this end, it is useful to introduce the notion of an approximation space spanned by a basis formed by the functions  $\psi_{h,\mathbf{k}}$  where  $\mathbf{k} \in \mathbb{Z}^n$ . We denote this space as

$$\mathbb{V}(\mathcal{L}_h, \psi) := \left\{ f_{\text{app}}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} c[\mathbf{k}] \psi_{h,\mathbf{k}}(\mathbf{x}) : c[\mathbf{k}] \in l_2(\mathbb{Z}^d) \right\}. \quad (3)$$

With some additional constraints on the reconstruction function  $\psi(\mathbf{x})$ , it can be shown that  $\mathbb{V}(\mathcal{L}_h, \psi) \subset L^2(\mathbb{R}^d)$  [4] and any  $f_{\text{app}} \in \mathbb{V}(\mathcal{L}_h, \psi)$  has a unique representation in terms of a finite-energy coefficient sequence  $c$ . Given a function  $f \in L^2(\mathbb{R}^d)$ , the minimum error approximation of  $f$  in  $\mathbb{V}(\mathcal{L}_h, \psi)$  in the least-squares sense is obtained by orthogonally projecting  $f$  onto  $\mathbb{V}(\mathcal{L}_h, \psi)$  [35]. This is accomplished by taking inner products of  $f$  with functions that are bi-orthogonal duals of  $\psi_{h,\mathbf{k}}$ . Let  $\hat{\psi}(\mathbf{x})$  denote the bi-orthogonal dual of  $\psi(\mathbf{x})$  that satisfies the condition  $\langle \psi(\mathbf{x} - \mathbf{Lk}) | \hat{\psi}(\mathbf{x} - \mathbf{Ll}) \rangle = \delta_{k,l}$ , where  $\delta$  is the multi-dimensional Kronecker delta function. The dual function  $\hat{\psi}$  has the following Fourier domain representation obtained

by solving the bi-orthogonality constraint in the Fourier domain [35].

$$\widehat{\psi}(\omega) = \frac{\widehat{\psi}(\omega)}{\widehat{A}_\psi(\omega)}, \text{ where } \widehat{A}_\psi(\omega) \xrightarrow{L} a_\psi[\mathbf{k}] \quad (4)$$

and the sequence  $a_\psi[\mathbf{k}]$  is obtained by sampling the auto-correlation function of  $\psi$  at the lattice sites, i.e.

$$a_\psi[\mathbf{k}] := (\psi * \bar{\psi})(\mathbf{x})|_{\mathbf{x}=\mathbf{Lk}}, \quad (5)$$

where  $\bar{\psi}(\mathbf{x}) := \psi(-\mathbf{x})$  and  $*$  indicates the continuous convolution operation. A scaled and translated version of  $\widehat{\psi}$  yields the bi-orthogonal dual of  $\psi_{h,\mathbf{k}}$ , i.e.  $\widehat{\psi}_{h,\mathbf{k}}(\mathbf{x}) := h^{-d}\widehat{\psi}(\frac{\mathbf{x}}{h} - \mathbf{Lk})$ . The minimum error orthogonal projection can now be written as

$$(\mathbf{P}_{\mathbb{V}(\mathcal{L}_h, \psi)} f)(\mathbf{x}) := \sum_{\mathbf{k}} c[\mathbf{k}] \psi_{h,\mathbf{k}}(\mathbf{x}), \text{ where } c[\mathbf{k}] = \langle f | \widehat{\psi}_{h,\mathbf{k}} \rangle. \quad (6)$$

The quality of the minimum error approximation provided by  $\mathbb{V}(\mathcal{L}_h, \psi)$  can be characterized by using the results of Strang and Fix [30]. In particular, we say that the space  $\mathbb{V}(\mathcal{L}_h, \psi)$  provides an  $n$ -th order approximation of  $f$  if  $\|f - (\mathbf{P}_{\mathbb{V}(\mathcal{L}_h, \psi)} f)\|_{L^2} = O(h^n)$  as  $h \rightarrow 0$ . This is true if and only if the reconstruction function  $\psi$  satisfies

$$\widehat{\psi}(\mathbf{0}) \neq 0 \text{ and } \widehat{\psi}(\omega - 2\pi \mathbf{L}^{-T} \mathbf{k}) = O(|\omega|^n) \text{ for } \mathbf{k} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}. \quad (7)$$

If  $\psi$  satisfies this condition, we term it an  $n$ -th order reconstruction function.

## 2.2 Box Splines

The box splines are very useful reconstruction functions that are well suited for designing approximation spaces on arbitrary sampling lattices. They satisfy the Strang-Fix relations (7) and have attractive mathematical properties similar to the B-splines. For a thorough exposition of the box splines, we refer the reader to de Boor *et al.* [9]. Associated with a box spline in  $\mathbb{R}^d$  is a  $d \times n$  ( $n \geq d$ ) matrix  $\Xi = [\xi_1, \xi_2, \dots, \xi_n]$  consisting of direction vectors  $\xi_k$ . We denote this box spline as  $M_\Xi(\mathbf{x})$ . It is obtained by successively convolving line segments along the direction vectors contained in  $\Xi$ . The simplest box spline is obtained by choosing  $n = d$  linearly independent direction vectors and is the indicator function of the parallelepiped formed by these direction vectors. Successive directional convolutions are defined inductively as

$$M_{[\Xi, \xi]}(\mathbf{x}) := \int_0^1 M_\Xi(\mathbf{x} - t\xi) dt. \quad (8)$$

The box splines have a simple Fourier domain representation given by

$$\widehat{M}_\Xi(\omega) := \prod_{\xi \in \Xi} \frac{1 - \exp(-j\xi^\top \omega)}{j\xi^\top \omega}. \quad (9)$$

The support of  $M_\Xi$  consists of all the points contained within the polytope formed by taking the Minkowski sum of the direction vectors in  $\Xi$ . This implies that  $M_\Xi$  is centered at the point  $c_\Xi := \sum_{\xi \in \Xi} \frac{1}{2}\xi$ . The smoothness of  $M_\Xi$  and the approximation order it provides are readily obtained by inspecting the columns of  $\Xi$  [9].

## 3 GRADIENT APPROXIMATION

### 3.1 A Two-Stage Orthogonal Projection Framework

In visualization and other related disciplines, the discrete measurement procedure is usually assumed to be ideal in which case the goal of gradient approximation is to reconstruct from the point samples  $f[\mathbf{k}] = f(h\mathbf{Lk})$  that lie on the lattice  $\mathcal{L}_h$ , an approximation of the gradient  $\nabla f(\mathbf{x})$ . Previously, we have developed a two-stage gradient approximation procedure that first seeks an approximation of  $f(\mathbf{x})$  in an auxiliary approximation space and then orthogonally projects each component of the gradient of the approximation onto a target approximation space [20]. A schematic illustrating this procedure is shown in Figure 2. We briefly summarize this approximation technique here

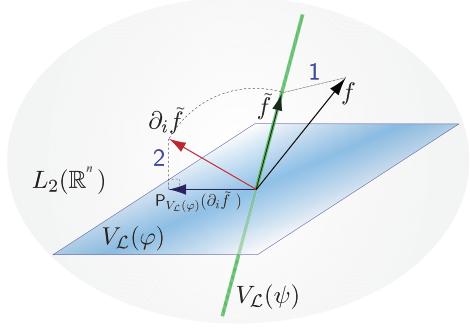


Fig. 2. An illustration of the two-stage OP framework.

and then show how to improve it by using a shifted reconstruction kernel in the target approximation space.

In the first stage, the sampled sequence  $f[\mathbf{k}]$  is used to generate an approximation  $\tilde{f}(\mathbf{x})$  that lies in the space  $\mathbb{V}(\mathcal{L}_h, \psi)$ . This is accomplished by applying a digital prefilter to  $f[\mathbf{k}]$  as follows.

$$\tilde{f}(\mathbf{x}) = \sum_{\mathbf{k}} (f * p)[\mathbf{k}] \psi_{h,\mathbf{k}}(\mathbf{x}), \quad (10)$$

where  $*$  now denotes a discrete convolution operation and the prefilter  $p$  is given in the Fourier domain by

$$p[\mathbf{k}] \xrightarrow{L} \widehat{P}(\omega) = \frac{1}{\widehat{Q}(\omega)}, \text{ where } \widehat{Q}(\omega) \xrightarrow{L} q[\mathbf{k}] = \psi(\mathbf{Lk}). \quad (11)$$

This prefilter ensures that  $\tilde{f}(\mathbf{x})$  exactly interpolates the sample values, i.e.  $\tilde{f}(h\mathbf{Lk}) = f[\mathbf{k}]$ . If the reconstruction kernel  $\psi$  is an interpolating one, then  $\widehat{P}(\omega) = 1$  and the prefiltering step is not necessary.

In the second stage  $\partial_i \tilde{f}(\mathbf{x})$ , the partial derivative of  $\tilde{f}(\mathbf{x})$  with respect to the  $i$ -th Cartesian direction ( $i \in \{1, \dots, n\}$ ), is orthogonally projected onto a target approximation space  $\mathbb{V}(\mathcal{L}_h, \varphi)$  where the reconstruction kernel is  $\varphi(\mathbf{x})$ . This is also achieved by applying a digital filter to the sequence obtained after the first stage. The resulting function  $\tilde{f}_i(\mathbf{x})$  is given by

$$\tilde{f}_i(\mathbf{x}) := (\mathbf{P}_{\mathbb{V}(\mathcal{L}_h, \varphi)} \partial_i \tilde{f})(\mathbf{x}) = \sum_{\mathbf{k}} \frac{1}{h} (f * p * \widehat{d}_i)[\mathbf{k}] \varphi_{h,\mathbf{k}}(\mathbf{x}), \quad (12)$$

where the digital derivative filter is given by

$$\widehat{d}_i[\mathbf{k}] := \langle \partial_i \psi | \varphi_{\mathbf{k}} \rangle. \quad (13)$$

For derivation details, we refer the reader to [20].

The familiar Shannon paradigm is also encapsulated in this framework. In particular, for bandlimited functions,  $\psi = \text{sinc}_L$ , the sinus cardinalis function associated with  $\mathcal{L}$ . The prefiltering step is unnecessary since  $\text{sinc}_L$  is interpolating. Exact derivative reconstruction can be achieved by choosing the target space generator  $\varphi$  to be  $\text{sinc}_L$  as well since the derivative of a bandlimited function is also bandlimited. For practicality reasons however, this method is rarely used and other choices are made for the generators  $\psi$  and  $\varphi$ . It is desirable to choose compactly supported generators that attempt to minimize the  $L^2$  error between the true gradient and its approximation; and that simplify the evaluation of the inner product (13).

### 3.2 Shifted Reconstruction Kernel

The overall quality of the gradient approximation scheme (12) is governed by the approximation properties of the spaces  $\mathbb{V}(\mathcal{L}_h, \psi)$  and  $\mathbb{V}(\mathcal{L}_h, \varphi)$ . In our previous work [20], we considered the case where all the first-stage derivatives  $\partial_i \tilde{f}$  in the axis aligned directions are orthogonally projected to the same target space  $\mathbb{V}(\mathcal{L}_h, \varphi)$  generated by a single reconstruction function  $\varphi$ . The target space is chosen so that it fulfills the regularity and accuracy demands of the application. With

the target space fixed, the first-stage auxiliary space should be chosen to have a higher approximation order so that the gradient of the first-stage approximation  $\nabla \tilde{f}(\mathbf{x})$  is close to the true gradient  $\nabla f(\mathbf{x})$ .

Using the same space to approximate the function as well as the gradient components is an attractive design choice from a computational point of view since the same scalar interpolation routines can be reused to interpolate the gradient as well. In general however, other choices are possible both in terms of the directions one chooses to compute derivatives along, as well as the target space in which each directional derivative is approximated. Towards this end, let  $\partial_{\hat{u}_i} \tilde{f}$  denote the directional derivative  $\nabla \tilde{f} \cdot \hat{u}_i$  and let  $\varphi^i(\mathbf{x})$  be the reconstruction function of the target space  $\mathbb{V}(\mathcal{L}_h, \varphi^i)$  onto which  $\partial_{\hat{u}_i} \tilde{f}$  is projected. Our goal is to choose each target space  $\mathbb{V}(\mathcal{L}_h, \varphi^i)$  such that it minimizes the orthogonal projection  $L^2$  error  $\|\partial_i \tilde{f} - (\mathbf{P}_{\mathbb{V}(\mathcal{L}_h, \varphi^i)} \partial_i \tilde{f})\|_{L^2}$ . Note that with this modification, the ideal scenario that yields zero error, i.e. when  $\varphi^i(\mathbf{x})$  is chosen such that  $\partial_i \tilde{f} \in \mathbb{V}(\mathcal{L}_h, \varphi^i)$ , can be easily incorporated in the framework, thus yielding an approximation that is the exact gradient of  $\tilde{f}$ .

The prospect of finding separate reconstruction spaces  $\mathbb{V}(\mathcal{L}_h, \varphi^i)$  is an ambitious one and may not be practically all that advantageous as it would require each component of the gradient to be reconstructed with a different reconstruction function  $\varphi^i(\mathbf{x})$ . However, if we choose reconstruction functions from the same family for both the first and second stages, we may be able to exploit the derivative relationships that exist between the two functions. In that case, the problem of finding separate functions  $\varphi^i(\mathbf{x})$  can be reduced to finding appropriate shifts of a symmetric reconstruction function  $\varphi(\mathbf{x})$ .

The above idea is best explained with an example where we use a tensor-product B-spline consisting of  $n$ -th degree 1D B-splines as the first stage reconstruction function on a  $d$ -dimensional Cartesian lattice  $\mathcal{I} = \mathbb{Z}^d$ . Let us denote this function as  $b_n(\mathbf{x}) = \prod_{i=1}^d b_n(x_i)$ , where  $\beta_n(x)$  is a 1D centered B-spline of degree  $n$ . The first stage approximation  $\tilde{f}(\mathbf{x})$  lies in the space  $\mathbb{V}(\mathcal{I}, b_n)$ . The centered B-splines exhibit the following derivative property [36].

$$\frac{d\beta_n}{dx}(x) = \beta_{n-1}(x + \frac{1}{2}) - \beta_{n-1}(x - \frac{1}{2}). \quad (14)$$

Using this property, it is straightforward to show that  $\partial_i \tilde{f} \in \mathbb{V}(\mathcal{I}, \varphi^i)$ , where  $\varphi^i(\mathbf{x})$  is the ideal second-stage reconstruction function for the  $i$ -th partial derivative and is given by

$$\varphi^i(\mathbf{x}) = \beta_{n-1}(x_i - \frac{1}{2}) \prod_{j \neq i} \beta_n(x_j). \quad (15)$$

Instead of the ideal functions  $\varphi^i(\mathbf{x})$ , let us choose the second-stage reconstruction functions to be

$$b_m^i(\mathbf{x}) := \beta_m(x_i - \frac{1}{2}) \prod_{j \neq i} \beta_m(x_j) = b_m(\mathbf{x} - \frac{1}{2} \hat{e}_i), \quad (16)$$

where  $m \leq n$  and  $\hat{e}_i$  represents the unit vector in the  $i$ -th Cartesian direction. Thus,  $b_m^i(\mathbf{x})$  is merely a shifted version of the centered function  $b_m(\mathbf{x})$  (Figure 3a). With this choice, we conjecture that we obtain a better approximation scheme as compared to our previous scheme [20] that uses the same symmetric function  $b_m(\mathbf{x})$  to approximate all the gradient components. The shifts ensure that the reconstruction functions remain close to the ideal. Additionally, they are easy to incorporate into existing interpolation routines as they are cheaply computed from the same symmetric function  $b_m(\mathbf{x})$  simply by shifting the point at which interpolation is to be performed by  $-\frac{1}{2}$  in the direction of the derivative.

The box spline  $M_{\Xi}(\mathbf{x})$  also exhibits an analogous derivative relationship [9]. If  $\xi \in \Xi$ , then the directional derivative  $\partial_{\xi} M_{\Xi}$  is given by

$$\partial_{\xi} M_{\Xi}(\mathbf{x}) = M_{\Xi \setminus \xi}(\mathbf{x}) - M_{\Xi \setminus \xi}(\mathbf{x} - \xi), \quad (17)$$

where  $\Xi \setminus \xi$  is the matrix obtained by removing one occurrence of  $\xi$  from  $\Xi$ . The directional derivative is therefore obtained by the backward difference of two lower-order box splines. If the box spline

is symmetric (i.e.  $\sum_{\eta \in \Xi} \eta = 0$ ), then the lower-order box spline thus obtained is shifted in the direction  $\xi$  as illustrated in Figure 3b for the hexagonal lattice in 2D. Thus, when working with symmetric box splines, we argue that we can obtain a better approximation scheme if instead of approximating a partial derivative in an axis-aligned direction using symmetric box splines, we approximate directional derivatives using the same box spline shifted along the direction of the derivative.

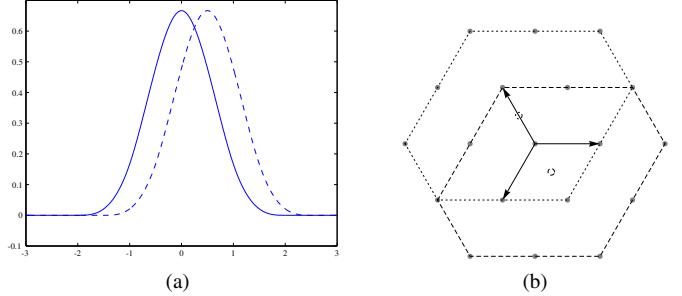


Fig. 3. (a) 1D illustration of the shifted derivative estimation scheme. Instead of using a centered kernel, e.g. a cubic B-spline  $\beta_3(x)$  (solid), to reconstruct the derivative, we propose to use the shifted version  $\beta_3(x - \frac{1}{2})$  (dashed). (b) Box splines on the hexagonal lattice are generated by three direction vectors (indicated as arrows). The support of a fourth-order box spline is a hexagon formed by the second-nearest neighbors. The directional derivative of this box spline along a convolution direction is a linear combination of two lower-order box splines that are shifted along the convolution direction as illustrated.

We formalize this notion in the following section by quantitatively analyzing the  $L^2$  error incurred as a result of choosing a shifted second-stage reconstruction function. Hereinafter, we collectively refer to both of these gradient estimation strategies as the orthogonal projection (OP) framework.

## 4 FOURIER-DOMAIN ERROR QUANTIFICATION

### 4.1 Scalar and Derivative Error Kernels

In order to quantitatively assess the error behavior of the OP derivative approximation framework, we propose to use the result of Blu *et al.* [4] who have devised a way to quantify the  $L^2$  error of a shift-invariant approximation technique in terms of a frequency domain error kernel. If a  $d$ -dimensional function  $f(\mathbf{x})$  is approximated in the space  $\mathbb{V}(\mathcal{L}_h, \varphi)$  by first applying a discrete prefilter  $p[\cdot]$  to the point samples and then reconstructing with the function  $\varphi(\mathbf{x})$  to yield the approximation  $f_{\text{app}}$ , then an estimate of the  $L^2$  error  $\|f - f_{\text{app}}\|_{L^2}$  is given by

$$\epsilon_f(h) := \sqrt{\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\widehat{f}(\omega)|^2 E(h\omega) d\omega}, \quad (18)$$

where  $E(\omega)$  is a Fourier-domain error kernel defined as

$$E(\omega) := 1 - \underbrace{\frac{|\widehat{\varphi}(\omega)|^2}{\widehat{A}_{\varphi}(\omega)}}_{E_{\min}(\omega)} + \underbrace{\widehat{A}_{\varphi}(\omega) \left| \widehat{P}(\omega) - \widehat{\varphi}^*(\omega) \right|^2}_{E_{\text{res}}(\omega)}. \quad (19)$$

The term  $E_{\min}(\omega)$  measures the minimum error incurred as a result of the orthogonal projection (6) and  $E_{\text{res}}(\omega)$  measures any additional error incurred as a result of deviating from the orthogonal projection. In particular,  $E_{\text{res}}(\omega)$  characterizes the error behaviour of various interpolative and quasi-interpolative approximation schemes by comparing the frequency response of the prefilter  $p$  to that of the dual  $\widehat{\varphi}$ .

Condat *et al.* [5] have recently extended this result to the Fourier-domain error quantification of one-dimensional derivatives of any order. Their result can be easily applied to arbitrary sampling lattices in higher dimensions to yield the following Fourier-domain derivative

error kernel,

$$E^l(\omega) := E_{\min}(\omega) + \underbrace{\widehat{A}_\varphi(\omega) \left| \frac{\widehat{D}(\omega)}{jl^\top \omega} - \widehat{\varphi}^*(\omega) \right|^2}_{E_{\text{res}}^l(\omega)}. \quad (20)$$

Here,  $\mathbf{l}$  is a principle direction of the lattice  $\mathcal{L}$  and  $\widehat{D} \xrightarrow{\mathcal{L}} d$  is a discrete filter that is to be applied to the samples of  $f$  to yield the directional derivative approximation,

$$\partial_l f(\mathbf{x}) \approx f_{\text{app}}^l(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} \frac{1}{h} (f * d)[\mathbf{k}] \varphi_{h,k}(\mathbf{x}). \quad (21)$$

The overall  $L^2$  error  $\|\partial_l f - f_{\text{app}}^l\|_{L^2}$  can be predicted according to

$$\epsilon_{fl}(h) := \sqrt{\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left| \widehat{f}(\omega) jl^\top \omega \right|^2 E^l(h\omega) d\omega}. \quad (22)$$

The derivative error kernel (20) has an algebraic form that is very similar to the scalar error kernel (19). It is also bounded below by  $E_{\min}(\omega)$  which suggests that the derivative approximation error can never be lower than the minimum scalar orthogonal projection error. The term  $E_{\text{res}}^l(\omega)$  can be interpreted as first undoing the directional derivative operation performed by the filter  $d$  effectively yielding an approximation of the original function  $f$ , and then measuring the deviation from the orthogonal projection.

Since we are dealing with point samples, the minimum error approximation scenario is not realizable. However, for functions that have most of their spectral power contained in the vicinity of  $\omega = 0$ , we can still achieve a similar asymptotic error behavior if the filter  $d$  is chosen appropriately. If  $\varphi$  is an  $n$ -th order reconstruction function, then the minimum error kernel satisfies  $E_{\min}(\omega) = O(|\omega|^{2n})$ . Thus, in order to ensure that  $f_{\text{app}}^l$  provides an  $n$ -th order approximation of  $\partial_l f$ , we require that  $E_{\text{res}}^l(\omega) = O(|\omega|^{2n})$  [4, 5]. In other words, the derivative filter  $d$  should be chosen so that the approximation scheme matches the orthogonal projection as closely as possible. This boils down to requiring that

$$\begin{aligned} \frac{\widehat{D}(\omega)}{jl^\top \omega} &= \widehat{\varphi}^*(\omega) + O(|\omega|^n), \quad \text{or equivalently,} \\ \widehat{D}(\omega) &= jl^\top \omega \widehat{\varphi}^*(\omega) + O(|\omega|^{n+1}). \end{aligned} \quad (23)$$

## 4.2 Assessment of the two-stage OP framework

We restrict attention to the case where derivatives are taken along the principle lattice directions only. For a symmetric first-stage reconstruction function  $\psi$  of the B-spline or box spline type, the directional derivative  $\partial_{l_i} \psi$  in the lattice direction  $\mathbf{l}_i$  ( $i \in \{1, 2, \dots, d\}$ ) is given by the backward difference of two lower-order spline functions that are centered about the points  $\frac{-l_i}{2}$  and  $\frac{l_i}{2}$  respectively. We therefore choose the second-stage reconstruction function to be  $\varphi^i(\mathbf{x}) := \varphi(\mathbf{x} - \frac{l_i}{2})$  where  $\varphi$  is an  $n$ -th order function of the spline variety that is symmetric about the origin. We then orthogonally project the directional derivative of the first-stage approximation onto the space  $\mathbb{V}(\mathcal{L}_h, \varphi^i)$ .

It is straightforward to verify that the shift carries over to the duals, i.e.

$$\varphi^i(\mathbf{x}) = \widehat{\varphi}(\mathbf{x} - \frac{l_i}{2}) \leftrightarrow \widehat{\varphi}^i(\omega) := \frac{\widehat{\varphi}(\omega)}{\widehat{A}_\varphi(\omega)} \exp(-\frac{j}{2} l_i^\top \omega). \quad (24)$$

Consequently, the digital derivative filter (13), now takes the form

$$\begin{aligned} \dot{d}_i[\mathbf{k}] &= \langle \partial_{l_i} \psi | \dot{\varphi}^i \rangle = (\partial_{l_i} (\psi * \widehat{\varphi}^i))(\mathbf{x})|_{\mathbf{x}=\mathbf{Lk}} \\ &= (\partial_{l_i} (\psi * \dot{\varphi}))(\mathbf{x})|_{\mathbf{x}=\mathbf{Lk}+l_i/2}. \end{aligned} \quad (25)$$

By expressing the dual  $\dot{\varphi}(\mathbf{x})$  in terms of a linear combination of the primal functions  $\varphi_{1,k}(\mathbf{x})$ , we can write this as

$$\begin{aligned} \dot{d}_i[\mathbf{k}] &= (\delta_i * a_\varphi^{-1})[\mathbf{k}], \\ \text{where, } \delta_i[\mathbf{k}] &= (\partial_{l_i} (\psi * \varphi))(\mathbf{x})|_{\mathbf{x}=\mathbf{Lk}+l_i/2}, \\ \text{and, } a_\varphi^{-1}[\mathbf{k}] &\xrightarrow{\mathcal{L}} 1/\widehat{A}_\varphi(\omega). \end{aligned} \quad (26)$$

The combined OP directional derivative filter  $D_i$  that is to be applied to the point samples of  $f$  (cf. (12)) is then given by

$$\begin{aligned} D_i[\mathbf{k}] &:= (p * \dot{d}_i)[\mathbf{k}] \xrightarrow{\mathcal{L}} \widehat{D}_i(\omega) = \frac{\sum_k \dot{d}_i[\mathbf{k}] \exp(-j\omega^\top \mathbf{Lk})}{\sum_k \psi(\mathbf{Lk}) \exp(-j\omega^\top \mathbf{Lk})} \\ &= \frac{\sum_{r \in \mathcal{L}^\circ} \left( \widehat{\partial_{l_i} \psi}(\omega - \mathbf{r}) \widehat{\varphi}(\omega - \mathbf{r}) \exp(\frac{j}{2} \mathbf{l}_i^\top (\omega - \mathbf{r})) \right)}{\widehat{A}_\varphi(\omega) \sum_{r \in \mathcal{L}^\circ} \widehat{\psi}(\omega - \mathbf{r})}. \end{aligned} \quad (27)$$

From this, it can be deduced that, *if the approximation order of  $\psi$  is at least  $n$ , then the resultant filter  $D_i$  satisfies (23) and provides an  $n$ -th order approximation of  $\partial_{l_i} f$  when used in conjunction with  $\varphi^i$* . Furthermore, this result holds true irrespective of the second-stage shift.

Thus, in order to guarantee an  $n$ -th order approximation of the directional derivative, we demand that  $\psi$  and  $\varphi$  be the same  $n$ -th order reconstruction functions. In this case, the first-stage prefilter  $p$  also serves as an interpolation prefilter for a scalar approximation that lies in  $\mathbb{V}(\mathcal{L}, \varphi)$ . Since all the directional derivative filters  $d_i$  have  $p$  in common, it only has to be applied once and the resulting data can be used for both scalar interpolation as well as gradient estimation as illustrated in Figure 4. Higher quality schemes can be obtained by choosing  $\psi$  such that it has an approximation order strictly greater than  $n$ . However, we don't discuss such schemes in this paper.

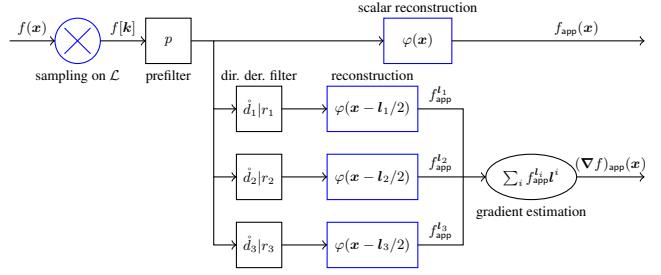


Fig. 4. Overview of the gradient estimation pipeline in  $\mathbb{R}^3$ . The sampled data is prefILTERed once and can be used for both scalar interpolation as well as gradient estimation. We use the derivative filters  $\dot{d}_1, \dot{d}_2$  and  $\dot{d}_3$  in the OP framework while  $r_1, r_2$  and  $r_3$  are FIR filters used for on-the-fly derivative estimation.

## 4.3 A Strategy for Designing Practical Filters

Even though the derivative filters that the OP framework yields are asymptotically optimal, they are not advantageous from a practical point of view since they have an infinite impulse response (IIR) and need to be applied in a preprocessing step resulting in significant storage overhead. The problem of designing derivative filters can be analyzed entirely using the derivative error kernel without resorting to a first-stage auxiliary approximation space. Here, we explore such a strategy that exploits the similarities between the scalar error kernel (19) and the derivative error kernel (20) to produce separable filters that are practically more advantageous.

In order for a directional derivative filter  $D_i$  to be asymptotically optimal, it must satisfy the optimality criterion (23). This is tantamount to requiring that the Taylor-series expansion of the frequency response  $\widehat{D}_i(\omega)$  match that of the ideal analysis function  $jl_i^\top \omega \widehat{\varphi}^*(\omega)$  upto order  $n+1$  where  $n$  is the desired approximation order. Additionally, it is practically desirable that  $D_i$  be factorable according to  $d_i[\mathbf{k}] = (p * r_i)[\mathbf{k}] \xrightarrow{\mathcal{L}} \widehat{D}_i(\omega) = \widehat{P}(\omega) \widehat{R}_i(\omega)$ , where  $r_i$  depends on the direction of the derivative while  $p$  has no such dependence and can be applied once in a preprocessing stage for all the directions in a manner akin to the OP framework (Figure 4). With these design criteria, equation (23) can be written as

$$\begin{aligned} \widehat{D}_i(\omega) &= \widehat{P}(\omega) \widehat{R}_i(\omega) = \widehat{\varphi}^*(\omega) jl_i^\top \omega + O(|\omega|^{n+1}) \\ &= (\widehat{\varphi}(\omega)) (jl_i^\top \omega \exp(\frac{j}{2} \mathbf{l}_i^\top \omega)) + O(|\omega|^{n+1}). \end{aligned} \quad (28)$$

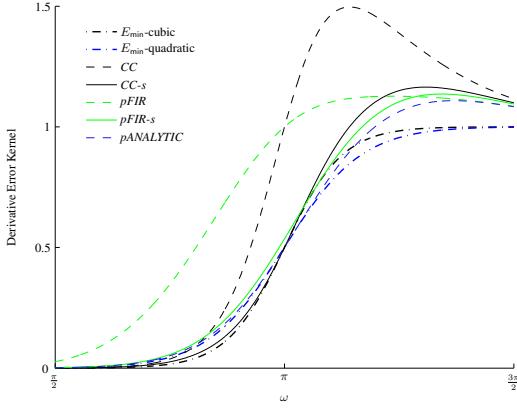


Fig. 5. The derivative error kernel for various derivative reconstruction schemes designed for the cubic B-spline.

Now, it is obvious that if  $p$  satisfies  $\hat{P}(\omega) = \hat{\varphi}(\omega) + O(|\omega|^n)$  and  $r_i$  satisfies  $\hat{R}_i(\omega) = j l_i^\top \omega \exp(j \frac{1}{2} l_i^\top \omega) + O(|\omega|^{n+1})$ , then the combined filter  $d_i$  satisfies (28) as well as the optimality criterion (23). The directional dependence due to the derivative and the shift are completely reflected in the response of the derivative filter  $r_i$  making the filter  $p$  directionally independent.

An inspection of the scalar residue term  $E_{\text{res}}(\omega)$  in (19) reveals that if we use the symmetric function  $\varphi$  to reconstruct  $f_{\text{app}}$ , then  $p$  also provides an asymptotically optimal  $n$ -th order approximation of  $f$ , i.e.  $\hat{P}(\omega) = \hat{\varphi}(\omega) + O(|\omega|^n)$  or equivalently,  $E_{\text{res}}(\omega) = O(|\omega|^{2n})$ . An interpolation prefilter that attempts to exactly interpolate the sample values (given by (11) with  $\psi = \varphi$ ) satisfies this condition [4]. Such a prefilter is usually employed anyway to approximate the scalar function. Combining it with a derivative filter  $r_i$  that satisfies  $\hat{R}_i(\omega) = j l_i^\top \omega \exp(j \frac{1}{2} l_i^\top \omega) + O(|\omega|^{n+1})$  will therefore guarantee an  $n$ -th order approximation. Higher quality quasi-interpolation prefilters are also possible [6] and are a topic of future research.

As for the directional component  $r_i$ , observe that the substitution  $\omega' = l_i^\top \omega$  converts the multi-dimensional Taylor expansion of the term  $(j l_i^\top \omega \exp(j \frac{1}{2} l_i^\top \omega))$  into a one-dimensional expansion of  $(j \omega' \exp(j \frac{1}{2} \omega'))$  in the variable  $\omega'$ . Therefore, it suffices to design derivative filters in 1D and then extend them to higher dimensions by simply applying the filter along the lattice direction  $l_i$ . This is an attractive solution for our design goals as we are interested in keeping the impulse response of  $r_i$  as short as possible so that it can be employed on the fly. The resulting overall filtering pipeline is the same as that obtained through the OP framework as shown in Figure 4.

## 4.4 Discussion

### 4.4.1 Error Behavior in 1D

We illustrate the error behavior of the two scenarios considered above with a 1D example where the centered reconstruction function is chosen to be a 4-th order cubic B-spline  $\beta_3(x)$ . For the OP scenario, the first-stage is also taken to be the cubic B-spline ( $\psi(x) = \beta_3(x)$ ) and the derivative is then projected to a second-stage centered cubic B-spline ( $\varphi(x) = \beta_3(x)$ ) and a shifted cubic B-spline ( $\varphi(x) = \beta_3(x - \frac{1}{2})$ ) yielding the filters  $CC$  and  $CC-s$  respectively.

For the practical scenario, the FIR derivative filter is obtained by equating Taylor coefficients upto and including terms of order 4 as explained in Section 5.1.2. The case without the shift is termed  $pFIR$  while the one with the shift is termed  $pFIR-s$ , where  $p$  refers to the scalar prefilter.

As shown in Figure 5, using a shifted reconstruction function leads to better error behavior across the board. The error kernel for the OP filter  $CC-s$  closely follows the minimum error kernel for the cubic B-spline while  $CC$  departs significantly around  $\omega = \pi$  suggesting that the use of this filter would lead to corruption of high frequency con-

tent. Using a shifted reconstruction function has a more dramatic impact on the FIR filters as can be clearly seen from the corresponding error kernels. In comparison to  $pFIR$ ,  $pFIR-s$  vastly improves the error response making it comparable to the OP filter  $CC-s$ .

Finally, we show that simply computing the analytic derivative of the scalar approximation is not the best possible choice. The error kernel for this scheme departs from the minimum sooner as compared to the OP schemes. This should not come as a surprise since the reconstruction quality is constrained by the approximation order of the quadratic B-spline which is one order lower as suggested by the corresponding minimum error kernel in Figure 5.

### 4.4.2 Gradient Reconstruction

So far, we have only discussed how to accurately reconstruct directional derivatives. The problem of combining the different directional derivatives to estimate the function gradient deserves some attention. The column vectors of the generating matrix  $L$  of lattice  $\mathcal{L}$  define a basis for  $\mathbb{R}^d$  that is not necessarily orthogonal. The gradient of a function is coordinate-system independent and can be conveniently expressed in a dual (contravariant) basis according to

$$\nabla f(\mathbf{x}) = \sum_{i=1}^d (\partial_{l_i} f)(\mathbf{x}) l^i, \quad (29)$$

where the dual vectors  $l^i$  are column vectors of the matrix  $L^{-T}$  [37]. Thus, if the directional derivatives in the principal lattice directions are approximately known, they can be easily combined to yield an approximation of the function gradient.

## 5 EXPERIMENTAL VALIDATION

In order to validate our proposed shifted schemes, we consider various 4-th order gradient estimation filters to be used in conjunction with the tricubic B-spline on the Cartesian Cubic (CC) lattice and the quintic box spline [15] on the Body-Centered Cubic (BCC) lattice. Both of these reconstruction functions are known to have an approximation order of 4 [15].

### 5.1 Tricubic B-Spline on CC

The CC lattice  $\mathcal{C} = \mathbb{Z}^3$  is generated by the matrix  $\text{diag}(1, 1, 1)$ . Due to its inherent separability, it is customary to design continuous reconstruction functions and discrete filters in 1D and then extend them to higher dimensions via a simple tensor product. Consequently, the filters can be applied in a separable way.

#### 5.1.1 OP Derivative Filters

We consider a 3D extension of the 1D case presented in Section 4.4.1 and choose the first-stage function to be the centered tricubic B-spline  $\psi(\mathbf{x}) = \beta_3(\mathbf{x}) = \beta_3(x)\beta_3(y)\beta_3(z)$ .

**CC:** For the unshifted case, the second stage functions  $\varphi^i(\mathbf{x})$  are all taken to be  $\beta_3(\mathbf{x})$  and the components of the gradient in the three principal directions of  $\mathcal{C}$  (i.e. the canonical basis) are orthogonally projected to  $\mathbb{V}(\mathcal{C}_h, \beta_3)$ . This case has already been considered in our previous work [20]. The resulting filters are completely separable and can be obtained by a tensor product of 1D filters. The first-stage prefilter is given by the samples of  $\beta_3(x)$ , the auto-correlation sequence is obtained by sampling  $\beta_7(x)$  while the derivative filter  $\delta_i$  is given by the samples of  $\frac{d\beta_7}{dx}(x)$  in the direction of the derivative and by the samples of  $\beta_7(x)$  in the other directions.

**CC-s:** We introduce a shift in the second stage and choose the reconstruction functions to be  $\varphi^i(\mathbf{x}) = \beta_3^i(\mathbf{x})$  (cf. (16)). The gradient component in the direction  $\hat{e}_i$  is then orthogonally projected to  $\mathbb{V}(\mathcal{C}_h, \beta_3^i)$ . The first-stage prefilter and the auto-correlation sequence are the same as the unshifted case **CC**. Using (26), we see that the derivative filter  $\delta_i$  is separable and is given by the samples of  $\frac{d\beta_7}{dx}(x + \frac{1}{2})$  in the direction  $\hat{e}_i$  and by the samples of  $\beta_7(x)$  in the other directions.

Since these are IIR filters, they are efficiently applied to the sampled data in the Fourier domain via a tensor product extension of the Fast

Fourier Transform (FFT). The result is stored in a gradient component volume which is later used during rendering for the purpose of gradient reconstruction.

### 5.1.2 FIR Derivative Filters

For these schemes, the sampled data is first prefiltered using an interpolation prefilter. This is also done in a preprocessing step using the FFT and the resulting filtered data is used for all subsequent operations. Coefficients needed for reconstructing a partial derivative are computed on the fly using a 1D FIR filter that is aligned in the direction of the derivative.

**pFIR:** For this case, the centered function  $b_3(\mathbf{x})$  is used to reconstruct gradient components. We choose an antisymmetric 1D derivative filter with weights  $[b, a, 0, -a, -b]$ . The criterion to satisfy is

$$2j(a \sin(\omega) + b \sin(2\omega)) = j\omega + O(\omega^5). \quad (30)$$

Expanding both sides and equating coefficients, the solution is found to be  $a = 2/3$  and  $b = -1/12$ . This is the same as our 4-*cd* filter developed using a spatial domain Taylor-series framework [20].

**pFIR-s:** This is analogous to the case *CC-s* in the sense that the shifted function  $b_3^i(\mathbf{x})$  is used to reconstruct the partial derivative in the direction  $\hat{e}_i$ . Let us take the unknown filter weights to be  $[l_2, l_1, c, r_1, r_2]$ . The criterion to satisfy is

$$c + e^{-2j\omega}(r_2 + r_1 e^{j\omega} + l_1 e^{3j\omega} + l_2 e^{4j\omega}) = j\omega e^{j\frac{\omega}{2}} + O(\omega^5). \quad (31)$$

Equating Taylor coefficients on both sides leads to the solution  $[-\frac{1}{24}, \frac{9}{8}, -\frac{9}{8}, \frac{1}{24}, 0]$ .

For the shifted schemes *pFIR-s* and *CC-s*, we need to reconstruct the  $i$ -th partial derivative with the shifted function  $b_3^i(\mathbf{x})$ . Reconstructing at the point  $\mathbf{x}$  in the shifted basis is equivalent to reconstructing at the point  $\mathbf{y} = \mathbf{x} - \hat{e}_i/2$  in the centered basis. Therefore, we simply translate the point of interest by  $-\hat{e}_i/2$  and use the code that implements the centered interpolation scheme. Alternatively, this can also be regarded as a shift of the underlying grid by  $\hat{e}_i/2$ .

## 5.2 Quintic Box Spline on BCC

The BCC lattice  $\mathcal{B}$  is generated by the symmetric matrix

$$\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \mathbf{b}_3] = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}. \quad (32)$$

It consists of those points in  $\mathbb{Z}^3$  whose coordinates have the same parity. The BCC box splines introduced by Entezari *et al.* are a family of even order symmetric reconstruction functions [15]. The simplest BCC box spline is the 2-nd order linear box spline  $M_{\Xi}(\mathbf{x})$  generated by the matrix

$$\Xi = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}. \quad (33)$$

Convolving this box spline with itself effectively doubles each direction in  $\Xi$ , yielding the 4-th order quintic box spline  $M_{\Xi^2}(\mathbf{x}) := M_{[\Xi, \Xi]}(\mathbf{x})$ . Using (33) and the box spline derivative relationship (17), it is easy to check that the directional derivative of  $M_{\Xi^2}(\mathbf{x})$  in the principal direction  $\mathbf{b}_i$  is given by a linear combination of two lower-order box splines that are centered at  $\pm \frac{1}{2}\mathbf{b}_i$ .

### 5.2.1 OP Derivative Filters

Like the CC lattice, we consider two different cases. For both cases, the first-stage function is the quintic box spline, i.e.  $\psi(\mathbf{x}) = M_{\Xi^2}(\mathbf{x})$ .

**QQ:** The partial derivatives of  $M_{\Xi^2}(\mathbf{x})$  in the canonical directions  $\hat{e}_i$  are orthogonally projected to the same target space  $\mathbb{V}(\mathcal{B}_h, M_{\Xi^2})$ . This case has also been explored in our previous work [20]. The constituent filters are non-separable. In particular, the first-stage prefilter is given by the samples of  $M_{\Xi^2}(\mathbf{x})$  at the lattice sites of  $\mathcal{B}$  while the auto-correlation sequence is obtained by sampling the box-spline  $M_{\Xi^4}(\mathbf{x})$ . Finally, the derivative filter  $\delta_i$  is obtained by

sampling the partial derivative  $\partial_i M_{\Xi^4}(\mathbf{x})$  at the lattice sites. We refer the reader to [20] for details.

**QQ-s:** For the shifted case, we take the second stage function to be  $\varphi^i(\mathbf{x}) = M_{\Xi^2}^i(\mathbf{x}) := M_{\Xi^2}(\mathbf{x} - \frac{1}{2}\mathbf{b}_i)$  and orthogonally project the first stage derivative in the direction  $\mathbf{b}_i$  to the target space  $\mathbb{V}(\mathcal{B}_h, M_{\Xi^2}^i)$ . The first-stage prefilter and the auto-correlation sequence are unaffected by the shift and are the same as *QQ*. The weights of the directional derivative filter  $\delta_i[k]$  are obtained by evaluating  $\partial_{\mathbf{b}_i} M_{\Xi^4}(\mathbf{x})$  at the sites  $\mathbf{x} = \mathbf{Bk} + \frac{1}{2}\mathbf{b}_i$  (evaluation code is provided as supplementary material). The three approximated directional derivatives are combined according to (29) to yield an estimate of the gradient.

Like the CC case, these IIR filters are also applied to the sampled data in a preprocessing step using the Fast BCC Discrete Fourier Transform (BCC-FFT) [1].

### 5.2.2 FIR Derivative Filters

This pipeline proceeds in a manner akin to the CC case above. The sampled data is first prefiltered for use with the quintic box spline. This is implemented in a preprocessing step using the BCC-FFT. The prefiltered data is then used to evaluate derivatives on the fly. We distinguish between three filter types.

**P-OPT26:** We use our previously derived error optimal 26 weight filter [20] to compute derivatives in the canonical directions. The components are all reconstructed with the centered quintic box spline.

**P-FIR:** We apply the 4 weight central differencing filter (cf. (30)) to the prefiltered data along the principal direction  $\mathbf{b}_i$  and use the centered quintic box spline to reconstruct the directional derivative. The different directions are combined using (29) to recover the gradient.

**P-FIR-s:** We apply the 4 weight shifted FIR filter derived above (cf. (31)) to the prefiltered data along the principal directions  $\mathbf{b}_i$  and use the corresponding shifted quintic box spline  $M_{\Xi^2}^i(\mathbf{x})$  to reconstruct the directional derivative. Analogous to the CC shifted reconstruction schemes, instead of reconstructing the directional derivative at  $\mathbf{x}$ , we reconstruct it at the translated point  $\mathbf{y} = \mathbf{x} - \mathbf{b}_i/2$  using the reconstruction code for the centered quintic box-spline.

## 6 RESULTS AND DISCUSSION

In order to assess the impact of our filters on volume visualization, we rendered isosurface images of the synthetic Marschner-Lobb (ML) test function first introduced in [21]. This function has a simple analytic formulation which can be used for the purpose of quantitative comparisons. We used the parameters given in [21] and sampled the function on CC and BCC grids of equivalent resolutions. To effectively discern the effect of a gradient estimation scheme, we used the analytic form to reconstruct the isosurface but used the sampled data to reconstruct the gradients according to the different schemes presented in Section 5.

Figure 6 shows the renditions obtained using the various schemes while Figure 7 illustrates the distributions of the gradient errors on the visible isosurface. *CC vs BCC*. The BCC lattice incurs less errors as compared to CC. It is known to be an optimal sampling lattice and produces better scalar reconstructions [31, 15]. It is therefore not surprising that this benefit carries over to gradient reconstruction as well. Furthermore, BCC also seems to be more sensitive to filter combinations as can be seen by the wider separation between the angular distribution curves.

**OP vs Practical.** The OP schemes perform better than the FIR schemes both in terms of angles and magnitudes. This further corroborates our analysis in Section 4.4.1 (see also Figure 5) where we have shown that the OP filters yield lower error kernels.

**Shift vs. Centered.** The shifted OP filters incur much less magnitude errors as predicted by the derivative error kernel. However, surprisingly, this trend seems to be reversed when we consider the angular error distributions. On the other hand, for the FIR filters, the shifts

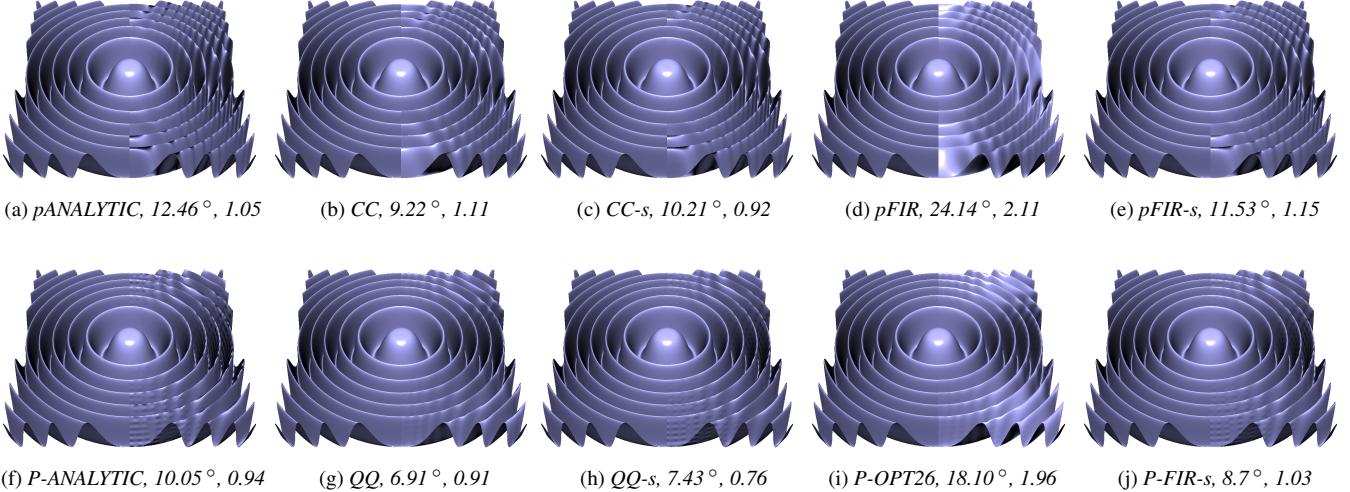


Fig. 6. The 0.5 isosurface of the ML function shaded using different gradient estimation schemes; top row, CC, and bottom row, BCC. The analytic function was used to reconstruct the isosurface while sampled data (CC:  $41 \times 41 \times 41$ , BCC:  $32 \times 32 \times 64$ ) were used for gradient estimation. For comparison, the left half of each image shows the truth. The mean angular error and the mean length of the error vector are indicated. The terms *pANALYTIC* and *P-ANALYTIC* refer to computing the analytic gradient of the scalar approximation.

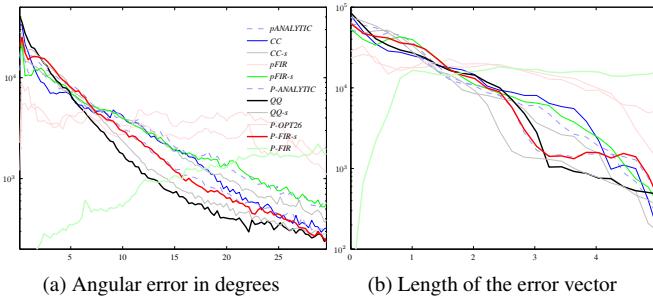


Fig. 7. Distribution of the errors for the images shown in Figure 6.

have a clear advantage. The filters *pFIR-s* and *P-FIR-s* lead to lower angular and magnitude errors as compared to their centered counterparts *pFIR* and *P-OPT26*.

*Orthogonal vs. Oblique.* We observed strong directional artifacts with the directional filter *P-FIR* which suggests that for an oblique central differencing scheme, the non-orthogonality of the basis has a major impact. However, the introduction of an oblique shift (*P-FIR-s*), not only cures these artifacts, but also leads to a more accurate scheme without any additional computational burden. In comparison, the orthogonal scheme *P-OPT26* is more expensive and less accurate.

*Analytic vs. FIR.* For the most part, the analytic derivative performs well, specially in comparison to the centered, orthogonal FIR schemes *pFIR* and *P-OPT26*. Undoubtedly, the IIR OP schemes are better, and even more so when the crucial aspect is the orientation of the gradient. With the introduction of a shift, the FIR schemes become almost as good as the analytical gradient in terms of the gradient magnitude. They seem to have an advantage in terms of the orientation of the gradient. However, this is rather fortuitous since the criterion optimized by the error kernel is the magnitude and not the orientation.

In order to investigate the effect of the shifted schemes, we experimented with an aneurysm dataset obtained through an angiography scan. Isosurface renditions of the original high resolution CC dataset are shown in Figure 8. Even at this resolution, the differences between the centered and shifted FIR schemes are remarkable, *pFIR-s* clearly reveals details that are smoothed out by the centered scheme *pFIR*.

We downsampled this dataset on equivalent CC and BCC grids

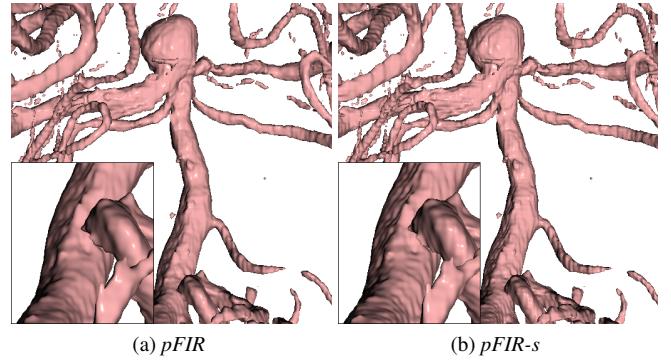


Fig. 8. An isosurface (isovalue = 1000) of the high resolution ( $512^3$ ) aneurysm CC dataset reconstructed using prefiltered tricubic B-spline interpolation.

and reconstructed the same isosurface using the gradient estimation schemes outlined in Section 5. In order to ensure that we remain in the low-pass regime, appropriate anti-aliasing filters were applied in the Fourier domain (using the FFT) before downsampling. To create a BCC volume downsampled by a factor of 4, we filtered the CC dataset by zeroing out the spectrum outside a rhombic dodecahedron that is the Voronoi cell of the dual FCC lattice. The resulting CC volume was then simply subsampled on a BCC lattice. An equivalent CC volume was created by discarding the spectrum outside the rectangular region corresponding to the Voronoi cell of the downsampled CC volume. The resulting images are shown in Figure 9. It should be stressed that the underlying isosurface for each lattice type is the same, since the same prefiltered reconstruction scheme is used to find the isosurface.

The visual differences between the various CC renditions are subtle. Nevertheless, one can observe that the OP scheme *CC* does a better job at preserving the high frequency details as compared to the FIR scheme *pFIR* which has the greatest smoothing effect. With an introduction of a shift, the FIR scheme *pFIR-s* recovers the lost details and is visually comparable to the shifted OP scheme *CC-s*. In contrast to the CC lattice, the BCC lattice provides a better scalar reconstruction and is much more sensitive to the various derivative filter combinations. As before, the centered FIR scheme *P-OPT26* has a strong



Fig. 9. Downsampled isosurface renditions of the aneurysm dataset, top row CC ( $323^3$ ) and bottom row, BCC ( $256 \times 256 \times 512$ ). The shading differences are solely due to the different gradient estimation schemes.

smoothing effect. In comparison, the centered OP scheme  $QQ$  fares a lot better as shown by the zoomed in regions of the corresponding images. The greatest improvement is shown by the shifted schemes  $P\text{-}FIR\text{-}s$  and  $QQ\text{-}s$ . They dramatically improve visual quality by revealing high frequency details and enhancing contrast. We have also compared these renditions to those obtained by computing the analytic gradient. The visual differences between the shifted schemes and the analytic gradient are hard to discern, although we did notice that the shifted OP schemes reproduce edges better and are more accurate in preserving the gradient magnitude in high frequency regions. Some additional examples are also provided in the supplementary material.

The benefits of the shifted schemes also extend to Direct Volume Rendering (DVR) as shown in Figure 1 for the case of the CC carp dataset. The images obtained by rendering the downsampled dataset clearly demonstrate the dramatic impact a mere shift can have on visual quality. Even though several color values are composited to produce a DVR rendering, the effect of a poor normal estimation scheme persists specially in areas of high variability.

In summary, the shifted OP schemes yield the best results. However, they achieve the superior visual quality at the expense of an added storage overhead. On the other hand, the shifted FIR schemes not only yield results that rival those obtained through the shifted OP schemes, they are also cheap to compute and do not require any changes to the underlying interpolation kernel as is the case with the analytic gradient. This makes them ideally suited for practical applications where both efficiency and accuracy are crucial. It should be emphasized that the scenario we have considered is the bare minimum to guarantee fourth-order convergence. Both the frameworks considered in Section 4 can be easily extended to obtain higher quality filters which can be combined with the tricubic B-spline (CC) or the quintic box spline (BCC) to further reduce the error. In contrast, there is little room for improvement for the analytic gradient (cf. Figure 5) because of the lower approximation order.

## 7 CONCLUSION

We have presented a new methodology to design derivative reconstruction schemes on arbitrary lattices, to extend the state of the art. The contribution is twofold; first, the components of the gradient are computed as partial derivatives along principal lattice directions and each derivative is reconstructed in a shift-invariant box spline space whose supporting grid is shifted between two lattice points in the direction of the derivative. Second, the digital prefilter which yields the box-spline coefficients from the available point samples is split into an interpolation prefilter and a FIR finite-difference filter, designed to fully exploit the approximation order of the reconstruction space. The interpolation prefilter is applied only once and provides the coefficients for the reconstruction of both the function and its derivative. This framework globally matches the state-of-the-art quality of our previously proposed Hilbert space framework [20], while being free of the storage overhead.

In future, we shall further investigate the use of the derivative error kernel [5] to design quasi-projection prefilters, by relaxing the interpolation constraint to improve the global behavior of the reconstruction. Furthermore, we shall study the extension of the formalism to the case where the measurements are corrupted by noise [10, 13, 28], to design schemes that apodize noise while retaining maximal approximation order.

## ACKNOWLEDGMENTS

The aneurysm dataset was obtained from [www.volvis.org](http://www.volvis.org) while the carp dataset is courtesy of the volume library [www9.informatik.uni-erlangen.de/External/vollib/](http://www9.informatik.uni-erlangen.de/External/vollib/). We would also like to thank Zahid Hossain for providing his BCC volume renderer. This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC).

## REFERENCES

- [1] U. R. Alim and T. Möller. A fast Fourier transform with rectangular output on the BCC and FCC lattices. In *Proceedings of the Eighth International Conference on Sampling Theory and Applications (SampTA'09)*, Marseille, France, May 18-22, 2009.
- [2] M. J. Bentum, T. Malzbender, and B. B. Lichtenbelt. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, Sept. 1996.
- [3] T. Blu, P. Thévenaz, and M. Unser. Linear interpolation revitalized. *IEEE Transactions on Image Processing*, 13(5):710–719, May 2004.
- [4] T. Blu and M. Unser. Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors. *IEEE Transactions on Signal Processing*, 47(10):2783–2795, October 1999.
- [5] L. Condat and T. Möller. Quantitative error analysis for the reconstruction of derivatives. Research Report hal-00462203, GREYC lab., Caen, France, Sept. 2009.
- [6] L. Condat and D. Van De Ville. Quasi-interpolating spline models for hexagonally-sampled data. *IEEE Transactions on Image Processing*, 16(5):1195–1206, may 2007.
- [7] B. Csébfalvi. An evaluation of prefiltered reconstruction schemes for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):289–301, 2008.
- [8] B. Csébfalvi and B. Domokos. Prefiltered gradient reconstruction for volume rendering. *Journal of WSCG*, 17(1-3):49–56, 2009.
- [9] C. de Boor, K. Höllig, and S. Riemenschneider. *Box Splines*. Springer Verlag, 1993.
- [10] D. den Hertog, R. Brekelmans, L. Driessen, and H. Hamers. Gradient estimation schemes for noisy functions. Discussion Paper 12, Tilburg University, Center for Economic Research, Feb. 2003.
- [11] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice-Hall, Inc., Englewood-Cliffs, NJ, 1st edition, 1984.
- [12] S. C. Dutta Roy and B. Kumar. *Handbook of Statistics*, volume 10, chapter Digital Differentiators, pages 159–205. Elsevier Science Publishers B. V., North Holland, 1993.
- [13] Y. C. Eldar and M. Unser. Nonideal sampling and interpolation from noisy observations in shift-invariant spaces. *IEEE Trans. Image Proc.*, 15(7):2636–2651, July 2006.
- [14] A. Entezari, R. Dyer, and T. Möller. Linear and Cubic Box Splines for the Body Centered Cubic Lattice. In *Proceedings of the IEEE Conference on Visualization*, pages 11–18, Oct. 2004.
- [15] A. Entezari, D. Van De Ville, and T. Möller. Practical box splines for reconstruction on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):313 – 328, 2008.
- [16] H. Farid and E. Simoncelli. Differentiation of discrete multi-dimensional signals. *IEEE Transactions on Image Processing*, 13(4):496–508, 2004.
- [17] B. Finkbeiner, U. R. Alim, D. V. D. Ville, and T. Möller. High-quality volumetric reconstruction on optimal lattices for computed tomography. *Computer Graphics Forum (Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization 2009 (EuroVis 2009))*, 28(3):1023–1030, 2009.
- [18] M. E. Goss. An adjustable gradient filter for volume visualization image enhancement. In *Graphics Interface*, pages 67–74, 1994.
- [19] H. Hamers, R. Brekelmans, L. Driessen, and D. Hertog. Gradient estimation using Lagrange interpolation polynomials. *Discussion Paper*, 2003.
- [20] Z. Hossain, U. R. Alim, and T. Möller. Towards high quality gradient estimation on regular lattices. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2010.
- [21] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In R. D. Bergeron and A. E. Kaufman, editors, *Proceedings of the IEEE Conference on Visualization*, pages 100–107. IEEE Computer Society Press, Oct. 1994.
- [22] T. Meng, B. Smith, A. Entezari, A. E. Kirkpatrick, D. Weiskopf, L. Kalantari, and T. Möller. On visual quality of optimal 3D sampling and reconstruction. In *Graphics Interface 2007*, pages 265 – 272, May 2007.
- [23] R. Mersereau and T. Speake. The processing of periodically sampled multidimensional signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(1):188–194, 1983.
- [24] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A comparison of normal estimation schemes. *Proceedings of the IEEE Conference on Visualization*, pages 19–26, Oct. 1997.
- [25] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of Accurate and Smooth Filters for Function and Derivative Reconstruction. *Proceedings of the Symposium on Volume Visualization*, pages 143–151, Oct 1998.
- [26] N. Neophytou and K. Mueller. Space-time points: 4d splatting on efficient grids. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 97–106, Piscataway, NJ, USA, 2002.
- [27] D. P. Petersen and D. Middleton. Sampling and Reconstruction of Wave-Number-Limited Functions in  $N$ -Dimensional Euclidean Spaces. *Information and Control*, 5(4):279–323, Dec. 1962.
- [28] S. Riachy, S. Bachalany, M. Mboup, and J.-P. Richard. An algebraic method for multi-dimensional derivative estimation. In *Proc. of 16th Med. Conf. on Control and Automation*, pages 356–361, June 2008.
- [29] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, 1949.
- [30] G. Strang and G. J. Fix. A Fourier Analysis of the Finite Element Variational Method. *Constructive Aspects of Functional Analysis*, pages 796–830, 1971.
- [31] T. Theußl, T. Möller, and E. Gröller. Optimal Regular Volume Sampling. In *Proceedings of the IEEE Conference on Visualization 2001*, pages 91–98, Oct 2001.
- [32] T. Theußl, H. Helwig, and E. Gröller. Mastering windows: improving reconstruction. In *Proc. of the IEEE Symposium on Volume Visualization (VVS)*, pages 101–108, 2000.
- [33] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, July 2000.
- [34] M. Unser. A general Hilbert space framework for the discretization of continuous signal processing operators. In *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 51–61, San Diego CA, USA, July 9–14, 1995. Part I.
- [35] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.
- [36] M. Unser, A. Aldroubi, and M. Eden. B-Spline signal processing: Part I—Theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, February 1993.
- [37] E. Young. *Vector and tensor analysis*. Marcel Dekker Inc, 1992.