Practical Box Splines for Reconstruction on the Body Centered Cubic Lattice

Alireza Entezari, Dimitri Van De Ville, *Member*, *IEEE*, and Torsten Möller, *Member*, *IEEE*

Abstract—We introduce a family of box splines for efficient, accurate, and smooth reconstruction of volumetric data sampled on the body-centered cubic (BCC) lattice, which is the favorable volumetric sampling pattern due to its optimal spectral sphere packing property. First, we construct a box spline based on the four principal directions of the BCC lattice that allows for a linear C^0 reconstruction. Then, the design is extended for higher degrees of continuity. We derive the explicit piecewise polynomial representations of the C^0 and C^2 box splines that are useful for practical reconstruction applications. We further demonstrate that approximation in the shift-invariant space—generated by BCC-lattice shifts of these box splines—is *twice* as efficient as using the tensor-product B-spline solutions on the Cartesian lattice (with comparable smoothness and approximation order and with the same sampling density). Practical evidence is provided demonstrating that the BCC lattice not only is generally a more accurate sampling pattern, but also allows for extremely efficient reconstructions that outperform tensor-product Cartesian reconstructions.

Index Terms—BCC, box splines, discrete/continuous representations, optimal regular sampling.

---

1 INTRODUCTION

IN this paper, we focus on sampling and representation of volumetric data. Two important issues are dealt with: 1) the choice of the sampling pattern and 2) the signal model that links the discrete to the continuous domain and, thus, serves to effectively interpolate or approximate the underlying continuous phenomenon. We start by introducing both aspects.

1.1 Optimal Regular Sampling

The study of optimal regular sampling patterns is not new [11], [18], [24], [29]. For the class of signals that have an isotropic band-limited (or essentially low-pass) spectrum, the problem of optimal regular sampling can be answered using the solution to the optimal circle (2D) or sphere (3D) packing problem. This is due to the fact that the sparsest regular (lattice) distribution of samples in the spatial domain demands the tightest arrangement of the replicas of the spectrum in the frequency domain. Therefore, the optimal sampling lattice is simply the dual of the densest packing lattice. This sampling lattice constitutes the *best generic* lattice for sampling trivariate functions.

Gauß proved that the densest packing in 2D is obtained by the hexagonal lattice [6]. For the 3D sphere packing problem, which is also known as the Kepler problem dating from the early 17th century, Gauß proved that the face-centered cubic (FCC) lattice attains the highest possible density [6]. Further, the Kepler conjecture—that FCC packing is an optimal packing of spheres even when the

lattice condition is not imposed—was not proven until 1998 by a lengthy computer-aided proof [15].

For the class of isotropic band-limited signals, the hexagonal lattice can contain 14 percent more information than the Cartesian lattice without introduction of any aliasing [24]. The analogous 3D FCC replication of the spectrum allows about 30 percent more information to be captured without any aliasing in the spectrum of the function. Having the spectrum replicated on the FCC lattice in the frequency domain corresponds to sampling on the body-centered cubic (BCC) lattice in the spatial domain.

Despite their theoretical advantages, optimal sampling strategies have only had a limited impact on practical applications. On one hand, there are no 3D scanning devices yet that produce data directly sampled on the BCC lattice; on the other hand, if the scanning machines adopt the optimal sampling pattern, main signal processing tools such as reconstruction to process and analyze the data are needed.

1.2 Reconstruction Kernels

Reconstruction from sampled data refers to the procedure of interpolating or approximating the underlying continuous-domain signal. Traditionally, the design of reconstruction *filters* is a rich area in signal processing. This approach is discrete to discrete and thus provides estimates of the signals on another regular sampling lattice. Typically, constraints in the frequency domain are used to guide the filter design process (for example, [3], [12], [28]). On the other hand, the approximation theory has shown how to design reconstruction *kernels*, which are defined in the continuous domain and allow the estimation of the signal at any point in space (for example, [16], [25], [27]).

These well-known solutions are 1D, and for image processing and volume rendering, they are extended to multiple dimensions through a separable extension (often called the tensor-product approach) or through a spherical extension (for example, McClellan transformation [22]). The problem of these extensions is that they do not deal well with the multidimensional nature of the sampling lattice, in particular, for non-Cartesian lattices [19]. The separable extension clearly is only satisfactory for Cartesian lattices,

- *A. Entezari and T. Möller are with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC, V5A 1S6. E-mail: {aentezar, torsten}@cs.sfu.ca.*
- *D. Van De Ville is with the Biomedical Imaging Group, EPFL/STI/LIB, Bat. BM 4.140, Swiss Federal Institute of Technology Lausanne, Station 17, CH-1015 Lausanne, Switzerland. E-mail: dimitri.vandeville@epfl.ch.*

for which the lattice's natural Nyquist region (see Section 2) would coincide with the kernel's low-pass region in the frequency domain. The spherical extension, however, has difficulties imposing zero crossing in the frequency domain at the dual-lattice points, which is crucial to guarantee approximation order and polynomial reproduction [30].

An interesting example in 2D is "hex-splines," which form a B-spline family for hexagonally sampled data [32]. The first-order hex-spline is defined as the indicator function of the Voronoi cell (and is thus nonseparable). Higher order hex-splines are defined in terms of successive convolutions of the first-order function. Analytical formulas are available in both the frequency and the spatial domain.

Box splines offer a mathematically elegant framework for constructing a class of multidimensional elements with flexible shape and support that can be nonseparable in a natural way. The general topic of box splines is rather intricate, and a general survey of results on the topic has been gathered in [9]. In this paper, we design a four-directional box spline that is geometrically aligned with the BCC lattice and allows for piecewise polynomial approximation of the data sampled on the BCC lattice.

The most commonly used method for evaluating box splines at arbitrary points is through de Boor's recurrence relation [9]. Unfortunately, the recursive evaluation of box splines is computationally inefficient and prone to numerical instabilities [10]. Kobbelt addresses the instability issues by delaying the evaluation of the discontinuous step function until the latest stages of recursion [17]. Even though the numerical inaccuracies of the recursive algorithm can be minimized, to make box splines practical in the field of volume graphics (for example, volume rendering), the computational complexity of their evaluation needs to be significantly reduced. Although the use of box splines in surface subdivision in graphics demands evaluations of a box spline on a fixed mesh, in the volume rendering domain, one needs to evaluate a box spline at arbitrary points. For traditional B-splines, the explicit piecewise polynomial representation is commonly used for fast evaluation; therefore, we introduce a new piecewise polynomial representation for the proposed box splines in Section 5.

In [8], Dæhlen proposes an algorithm to evaluate a four-directional box spline on a fixed mesh shifted to an arbitrary position. Somewhat similar to our evaluation method, he relies on the relation of box splines to *cone splines* (truncated power functions). In Dæhlen's method, the evaluation of truncated power functions is still based on a recurrence relation and is based on the connection with simplex splines. In our case, however, we derive the explicit polynomial representation of the truncated power function in Section 5. This representation provides us with the exact evaluation of box splines that is free of numerical inaccuracies since we avoid any recurrence relations. Furthermore, similar to the piecewise polynomial evaluation methods of B-splines, our method exploits the symmetries in the support of the box spline to further reduce the computational cost (see Section 5.5.4).

In volume graphics, the optimality of BCC sampling has been explored by Theußl et al. [20]. They applied the spherical extension of reconstruction filters, which resulted in rather blurry and unsatisfying results. Different ad hoc approaches were studied for reconstruction and derivative reconstruction on the BCC lattice, with mixed results [19]. Also, isosurface extraction on the optimal sampling lattice has been studied with inconclusive results [4]. We also

exploited the BCC lattice in multiresolution analysis [14]. Recently, Csébfalvi [7] demonstrated a reconstruction using a Gaussian kernel and the principle of generalized interpolation [31]. Although this method provides an isotropic solution, it does not guarantee approximation order. It is also a numerical scheme without any closed-form solution for the interpolation kernel. Moreover, our explicit piecewise polynomial representation makes the box-spline solution several times more computationally efficient than that in [7] (see Section 6).

1.3 Scope and Organization of the Paper

In this paper, we provide accurate and efficient reconstruction methods for the BCC lattice. Such reconstructions have been sought for in the volume graphics community [4], [19], [20] to better exploit the theoretical advantages of the BCC lattice. Several contributions are proposed:

- We establish a four-directional box spline that is geometrically tailored to the BCC lattice in Section 4.[1] The first-order box spline is a 3D piecewise linear function. Higher order versions are obtained by successive convolutions. This way, we can choose the required smoothness and approximation order.
- We explicitly characterize the polynomial patches defining these box splines, which is detailed in Section 5. Our characterization method is general (for any order) and leads to polynomial expressions that can be implemented to evaluate the box spline at any point. Specifically, we derive the explicit expressions for the C^0 and C^2 members of our family of BCC box splines, since they are the most relevant for the practitioner in rendering applications.
- We demonstrate that our box splines (for C^0 and C^2 continuity) on the BCC lattice are computationally *twice* as efficient as tensor-product B-splines on the Cartesian lattice (for comparable smoothness and the same sampling density); see Table 2. Based on these results, in Section 6, we conclude that BCC lattice sampling can be more attractive not only on a theoretical level, but also in a practical setting.

2 GEOMETRY OF THE BODY-CENTERED CUBIC LATTICE

A *lattice* is a periodic pattern made up by an infinite array of points in which each point has a neighborhood identical to those of all the other points [2]. In other words, every lattice point has the same Voronoi cell, and we can refer to the Voronoi cell of the lattice without ambiguity.

Periodic sampling of a function in the spatial domain gives rise to a periodic replication of its spectrum in the frequency domain. The lattice that describes the centers of the replicas in the frequency domain is called the *dual*, *reciprocal*, or *polar* lattice. Reconstruction amounts to eliminating the replicas of the spectrum in the frequency domain and preserving the primary spectrum. Therefore, the ideal kernel for reconstruction in the space of "band-limited" functions is a function that is the inverse Fourier transform of the characteristic function of the Voronoi cell of the dual lattice.

In 3D, the BCC lattice points are located on the corners of a cube with an additional sample in the center of this cube.

---

1. A preliminary version of these box splines was presented in our conference paper [13].

Therefore, the BCC lattice can be considered as two interleaving Cartesian lattices where the vertices of the secondary Cartesian lattice are moved to the center of the primary Cartesian cells. An alternative way of describing the BCC lattice as a sublattice of the Cartesian is to start with a Cartesian lattice (that is, $\mathbb{Z}^3$) and retain only those points whose coordinates have identical parity. For an integer point in $\mathbb{Z}^3$ to belong to the BCC lattice, all three of x, y, and z-coordinates need to be odd, or all three need to be even. Therefore, the BCC lattice is a subgroup whose quotient group is of order *four*.[2] Therefore, the BCC lattice is a sublattice of $\mathbb{Z}^3$ whose density is $1/4$ in $\mathbb{Z}^3$; in other words, the volume of the Voronoi cell of each lattice point is four. This BCC lattice can be generated by integer linear combinations of the columns of the *sampling matrix*:

$$BCC = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}. \qquad (1)$$

Sampling a function with lattice points generated by the BCC sampling matrix, has the effect of replicating the spectrum of that function periodically on the dual lattice, which is an FCC-type lattice generated by [11]

$$FCC = 2\pi BCC^{-\mathrm{T}} = -\pi \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \qquad (2)$$

The FCC lattice is often referred to as the D_3 lattice [6]. In fact, D_3 belongs to a general family of lattices D_n, sometimes called checkerboard lattices. The checkerboard property implies that the sum of the coordinates of the lattice sites is always even (multiple of π with the scaling in (2)). We will use this property to demonstrate the zero crossings of the frequency response of the reconstruction filters at the FCC lattice sites.

The simplest interpolation kernel on any lattice is the indicator function of the Voronoi cell of the lattice. The corresponding interpolation scheme is the generalization of the so-called nearest neighbor interpolation. More sophisticated reconstruction kernels involve information from the neighboring points of a given lattice point. With this in mind, we focus in the next section on the geometry and the polyhedra associated with the BCC lattice.

2.1 Polyhedra Associated with the Body-Centered Cubic Lattice

Certain polyhedra arise naturally in the process of constructing interpolation filters for a lattice. The Voronoi cell of the lattice is one such example. The Voronoi cell of the Cartesian lattice is a cube, and the Voronoi cell of the BCC lattice is a truncated octahedron, as illustrated in Fig. 1a.

We are also interested in the cell formed by the immediate neighbors of a lattice point. The first (nearest) neighbors of a lattice point are defined via the Delaunay tetrahedralization of the lattice; a point q is a *first neighbor* of p if their respective Voronoi cells share a (nondegenerate) face. The *first neighbors' cell* is the polyhedron whose vertices are the first neighbors. Again, this cell is the same for all points on the lattice.

2. Looking at the (x, y, z) coordinates mod 2, there are eight different combinations, and we are interested in only two: $(0, 0, 0)$ and $(1, 1, 1)$.
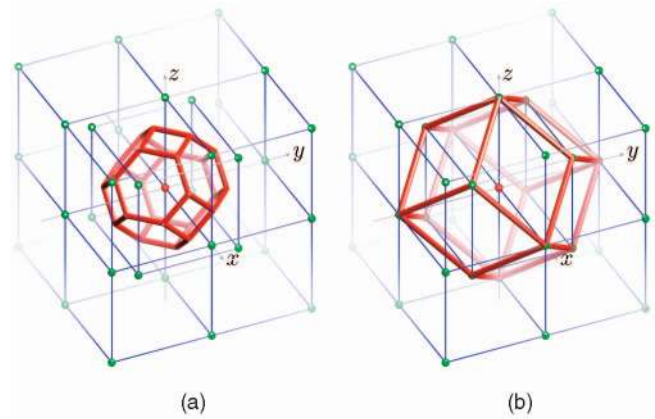

Fig. 1. (a) The Voronoi cell of the BCC lattice is the truncated octahedron. (b) The first neighbors' cell is the rhombic dodecahedron.

For example, by this definition, there are six first neighbors of a point in a Cartesian lattice; the first neighbors' cell for the Cartesian lattice is the octahedron. For the BCC lattice, there are 14 first neighbors for each lattice point. The first neighbor cell is a rhombic dodecahedron, as illustrated in Fig. 1b.

3 BOX SPLINES REVIEW

Here, we begin by briefly introducing box splines and state their properties that will be useful further on.

A box spline is characterized by a set of direction vectors that indicate its construction by successive convolution of line segments along these vectors. The linear combination of shifts of a box spline generates a spline whose smoothness and ability to approximate continuous functions also depend on these direction vectors. Notationally, the direction vectors are usually gathered in a matrix; that is, a box spline in $\mathbb{R}^s$ is specified by $n \geq s$ vectors in $\mathbb{R}^s$ that are columns of its matrix $\Xi = [\xi_1, \xi_2, \ldots, \xi_n]$. The support of the box spline is all points $x \in \mathbb{R}^s$ such that $x = \Xi t$, where $t \in \mathbb{R}^n$, and $0 \leq t_k \leq 1$ for $1 \leq k \leq n$. In other words, the support of the box spline is contained in the convex combination of these direction vectors.

The simplest box spline is constructed by $n = s$ vectors and is the (area-normalized) characteristic function of its support:

$$M_\Xi(x) = \begin{cases} \frac{1}{|\det \Xi|} & \text{where } x = \Xi t \text{ and } t \in [0,1)^n, \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

Clearly, the box spline from (3) is discontinuous at the boundary of its support. Its 1D version is the *boxcarr* function that is simply the indicator function for the interval $[0, 1)$.

For the general case $n > s$, the box splines are defined recursively:

$$M_{[\Xi, \xi]}(x) = \int_0^1 M_\Xi(x - t\xi)dt. \qquad (4)$$

This inductive definition implies that starting from the base case as in (3) the indicator function is smeared *along* the additional direction vector. Hence, the convolution of two box splines is yet another box spline:

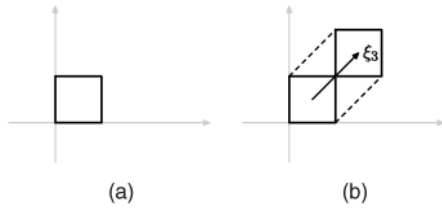$$M_{\Xi_1} * M_{\Xi_2}(x) = M_{[\Xi_1, \Xi_2]}(x). \qquad (5)$$

Fig. 2. Construction of the linear element from the simplest box spline.

A box spline is a piecewise polynomial of degree at most $n - s$. Moreover, let ρ be the minimal number of vectors such that if they were removed from Ξ, the remaining vectors would *not* span $\mathbb{R}^s$. Then, $M_\Xi \in C^{\rho - 2}$, where C^n is the space of n-time continuously differentiable functions. The Fourier transform of a box spline is

$$\hat{M}_\Xi(\omega) = \prod_{\xi \in \Xi} \frac{1 - \exp(-i\xi^T \omega)}{i\xi^T \omega}, \qquad (6)$$

where $i = \sqrt{-1}$ as usual. In 2D, the simplest box spline is specified by

$$\Xi_0 = [\,\xi_1 \quad \xi_2\,] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

which is the indicator function of the unit square $[0, 1)^2$.

Adding a direction vector of $\xi_3 = [\;1 \quad 1\;]^T$ to Ξ_0 smears the unit square across its diagonal. This is illustrated in Fig. 2. As the basic box spline is a constant function on the unit square, the result of smearing it along the diagonal produces a linear box spline that is represented by $[\Xi_0, \xi_3]$. The support of this box spline is illustrated in Fig. 2b. This box spline is a bivariate piecewise polynomial of degree one. This box spline generates a C^0 spline function space, as $\rho = 2$.

4   FOUR-DIRECTIONAL BOX SPLINE ON THE BODY-CENTERED CUBIC LATTICE

The construction of box splines dedicated to the BCC lattice is guided by the fact that the rhombic dodecahedron (the first neighbors' cell of the BCC lattice, see Fig. 1b) is the 3D shadow of a 4D hypercube (tesseract) along its antipodal axis. This construction is a generalization of the 2D linear box spline with hexagonal support, which can be obtained by projecting a 3D cube along its antipodal axis; see Fig. 3b.

4.1   Geometric Construction

Integrating a constant tesseract along its antipodal axis yields a function that has a rhombic dodecahedron support (see Fig. 1b), has its maximum value at the center, and has a *linear* falloff toward the 14 first-neighbor vertices. Since it arises from the projection of a higher dimensional box, this function serves as the linear box spline interpolation kernel on the BCC lattice.

Let $\mathcal{B}$ denote the boxcarr function. The characteristic function of the unit tesseract is given by a tensor product of four $\mathcal{B}$ functions on each axis. By projecting the unit tesseract, one obtains a rhombic dodecahedron whose geometric scale is only half of the first neighbors' cell of the BCC lattice described by (1). In this BCC lattice, with integer lattice coordinates, the first neighbors' cell is scaled such that the blue edges in Fig. 1b are of length two.
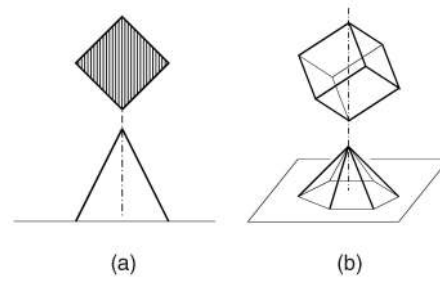

Fig. 3. (a) One-dimensional linear box spline (triangle function). (b) Two-dimensional hexagonal linear box spline.

Therefore, we scale the geometry of the unit tesseract by two and normalize by its hypervolume:

$$\mathcal{T}(x, y, z, w) := \frac{1}{16} \mathcal{B}(x/2)\mathcal{B}(y/2)\mathcal{B}(z/2)\mathcal{B}(w/2). \qquad (7)$$

Let $v = (2, 2, 2, 2) := [2, 2, 2, 2]^T$ denote a vector along the antipodal axis. In order to project along this axis, it is convenient to rotate it so that it aligns with the w-axis:

$$R = \frac{1}{2}[\rho_1 \rho_2 \rho_3 \rho_4] = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \qquad (8)$$

This rotation matrix transforms v to $(0, 0, 0, 4)$. Also, by examining (8), one can see that each vertex of the rotated tesseract, when projected along the w-axis, will coincide with the origin or one of the vertices of the rhombic dodecahedron: $(\pm 1, \pm 1, \pm 1)$, $(\pm 2, 0, 0)$, $(0, \pm 2, 0)$, $(0, 0, \pm 2)$, or $(0, 0, 0)$. Let $x = (x, y, z, w)$; now, the linear box spline is given by

$$L_{RD}(x, y, z) = \frac{1}{16} \int \mathcal{T}(R^{-1}x) \, dw.$$

Substituting in (7), we get

$$L_{RD}(x, y, z) = \frac{1}{16} \int \prod_{k=1}^{4} \mathcal{B}\left(\frac{1}{4}\rho_k^T x\right) dw. \qquad (9)$$

Note that value at the origin is $L_{RD}(0, 0, 0) = 1/4$ (see [9, II.8]). This is due to the fact that the box splines are normalized to $\int L_{RD}(x)dx = 1$, whereas the sampling density of the BCC lattice $(\frac{1}{4})$ demands a kernel whose integral is four. Therefore, in order to preserve the energy in the discrete/continuous model, we employ the box splines scaled by four on the BCC lattice. This scaling ensures that the value of the linear box spline is one at the center and zero at all other lattice sites. Hence, the linear box spline constitutes a linear interpolator on the BCC lattice.

4.2   Fourier Transform

If the direction matrix of a box spline is known, the distributional definition of box splines easily leads to their frequency-domain representation. Here, we present a geometric derivation of the Fourier transform of our box spline.

From the projection-based construction of the rhombic dodecahedron discussed earlier, we can derive the Fourier transform of the linear box spline function described by (9). From (7), it is evident that the Fourier transform of the

characteristic function of the tesseract is given by the tensor product:

$$\hat{\mathcal{T}}(\omega_x, \omega_y, \omega_z, \omega_w) = \frac{1 - \exp(-i2\omega_x)}{i2\omega_x} \frac{1 - \exp(-i2\omega_y)}{i2\omega_y}$$
$$\frac{1 - \exp(-i2\omega_z)}{i2\omega_z} \frac{1 - \exp(-i2\omega_w)}{i2\omega_w}$$

since $\frac{1}{2}\mathcal{B}(x/2) \longleftrightarrow \frac{1 - \exp(-i2\omega)}{i2\omega}$. We use $\longleftrightarrow$ to indicate a Fourier transform pair.

By the Fourier slice-projection theorem, projecting a function along a certain direction in the spatial domain amounts to slicing its Fourier transform perpendicular to the direction of projection. This slice runs through the origin. Again, we make use of the rotation (8) to align the projection axis with the w-axis. Thus, in the frequency domain, we take the slice $\omega_w = 0$.

It is convenient to introduce the 3×4 matrix

$$\mathbf{\Xi} = [\xi_1 \xi_2 \xi_3 \xi_4] = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad (10)$$

given by the first three rows of the rotation matrix R of (8). The reason for omitting the last row is that we are taking a slice $\omega_w = 0$ orthogonal to the fourth axis at the origin. The Fourier transform of the linear box spline can now be written as

$$\hat{L}_{RD}(\omega_x, \omega_y, \omega_z) = \prod_{k=1}^{4} \frac{1 - \exp(-i\xi_k^T \boldsymbol{\omega})}{i\xi_k^T \boldsymbol{\omega}},$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$. The space-domain function L_{RD} corresponds to the box spline $M_{\mathbf{\Xi}}$; we will use this box spline symbol from now on. The Fourier transform of this box spline is then

$$\hat{M}_{\mathbf{\Xi}}(\boldsymbol{\omega}) = \prod_{k=1}^{4} \frac{1 - \exp(-i\xi_k^T \boldsymbol{\omega})}{i\xi_k^T \boldsymbol{\omega}}. \quad (11)$$

Since any three of the vectors in $\mathbf{\Xi}$ span $\mathbb{R}^3$, at least two vectors need to be removed from $\mathbf{\Xi}$, so the remaining vectors do not span; therefore, $\rho = 2$. Hence, this box spline is guaranteed to produce a C^0 reconstruction. We can verify the vanishing moments (zero crossings) of the frequency response at the aliasing frequencies on the FCC lattice points. We first note that $\sum_{k=1}^{4} \xi_k = \mathbf{0}$; therefore, the center of the box spline $M_{\mathbf{\Xi}}$ is at the origin [9], and the Fourier transform can be written as

$$\hat{M}_{\mathbf{\Xi}}(\boldsymbol{\omega}) = \prod_{k=1}^{4} \text{sinc}(\xi_k^T \boldsymbol{\omega}).$$

Recall that $\text{sinc}(t) = \frac{\sin(t/2)}{t/2}$. This reformulation provides a more convenient form to verify zero crossings. Due to the checkerboard property, for every FCC lattice point, the sum of its coordinates is always even. Since the FCC lattice dual to the BCC lattice of our discussion is scaled by π (according to (1)), for $\boldsymbol{\omega}$ on the FCC lattice, $\xi_4^T \boldsymbol{\omega} = (\omega_x + \omega_y + \omega_z) = 2\pi k$ for some $k \in \mathbb{Z}$; therefore, $\text{sinc}(\xi_4^T \boldsymbol{\omega}) = 0$ on all of the aliasing frequencies. Since $\xi_4^T \boldsymbol{\omega} = -\xi_1^T \boldsymbol{\omega} - \xi_2^T \boldsymbol{\omega} - \xi_3^T \boldsymbol{\omega}$, at least one of the $\xi_m^T \boldsymbol{\omega}$ for $m = 1, 2, 3$ needs to also be an even multiple of π since the sum of three odd multiples of π cannot be an even multiple. For such k, we have $\text{sinc}(\xi_k^T \boldsymbol{\omega}) = 0$; therefore, there is a zero of order at least two at each aliasing frequency, yielding a C^0 kernel whose

approximation power is two on the BCC lattice [30]. This smoothness and approximation power parallels that of the trilinear B-spline interpolation on the Cartesian lattice.

4.3 Higher Order Box Splines

The number of vanishing moments can be doubled by convolving the linear box spline with itself. Hence, the resulting reconstruction kernel will have twice the approximation power on the BCC lattice due to the Strang-Fix result [30]. As noted before, the resulting box spline can then be represented by $\mathbf{\Xi}^2 := [\mathbf{\Xi}, \mathbf{\Xi}]$, where every direction vector is duplicated.

An equivalent method of deriving this function would be to convolve the constant function on the tesseract with itself and project the resulting distribution along a diagonal axis (this commutation of convolution and projection is easy to understand in terms of the corresponding operators in the frequency domain—see Section 4.2). Convolving the constant function on the tesseract with itself results in another function supported on a tesseract that is the tensor product of four 1D triangle (linear B-spline) functions. Let Λ denote the triangle function. Then, the convolution yields

$$\mathcal{T}_c(x, y, z, w) = \frac{1}{16} \Lambda\left(\frac{1}{2}x\right) \Lambda\left(\frac{1}{2}y\right) \Lambda\left(\frac{1}{2}z\right) \Lambda\left(\frac{1}{2}w\right). \quad (12)$$

Following the same 4D rotation as in the previous section, we obtain a space-domain representation of the new box spline:

$$C_{RD}(x, y, z) = \frac{1}{16} \int \prod_{k=1}^{4} \Lambda\left(\frac{1}{4}\rho_k^T \boldsymbol{x}\right) dw. \quad (13)$$

Similar to the linear case, we use the matrix of $\mathbf{\Xi}^2$ to represent this properly scaled box spline. Since convolution in the space domain amounts to a multiplication in the frequency domain, we use (11) to derive the Fourier transform of the new box spline:

$$\hat{M}_{\mathbf{\Xi}^2}(\boldsymbol{\omega}) = \hat{M}_{\mathbf{\Xi}}^2(\boldsymbol{\omega}) = \prod_{k=1}^{4} \left(\frac{1 - \exp(-i\xi_k^T \boldsymbol{\omega})}{i\xi_k^T \boldsymbol{\omega}}\right)^2. \quad (14)$$

We can see that the number of vanishing moments of this box spline is doubled when compared to the linear kernel. This implies that this box spline has fourth-order approximation power on the BCC lattice [30]. The eight directions of this box spline $\mathbf{\Xi}^2$ are duplicates of the original four directions. Consequently, the minimum number of directions that one needs to remove from $\mathbf{\Xi}^2$ so that the remaining vectors do not span $\mathbb{R}^3$ is $\rho = 4$; hence, the weighted shifts of this box spline are guaranteed to produce C^2 continuous reconstructions with fourth-order approximation. This smoothness and approximation power parallels that of the tricubic B-spline reconstruction on the Cartesian lattice; for this reason, we have referred to $M_{\mathbf{\Xi}^2}$ as the "cubic" box spline in [13]. However, since there are eight directions, this trivariate box spline is composed of quintic polynomials. Therefore, according to de Boor's notations [9], we will call this box spline a quintic box spline.

As we noted earlier, $M_{\mathbf{\Xi}}$ is of second-order approximation power on the BCC lattice. The nth convolution of the linear kernel with itself, denoted by $M_{\mathbf{\Xi}^n}$, will have an approximation power of $2n$ on the BCC lattice. These box splines would produce C^{2n-2} reconstructions.
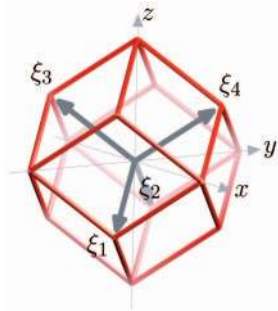
Fig. 4. The support of the box spline represented by Ξ is a rhombic dodecahedron formed by the four direction vectors in Ξ.

4.4 Support

The support of M_Ξ is a rhombic dodecahedron, as shown in Fig. 4. The support of M_{Ξ^n} is the Minkowski sum of n copies of rhombic dodecahedron. Since a rhombic dodecahedron is a convex and symmetric polyhedron (with respect to its center), its Minkowski addition with itself will have the same shape, scaled by two. In general, the support of M_{Ξ^n} would be a rhombic dodecahedron scaled by n [33].

The volume of the support of the box spline M_Ξ as depicted in Fig. 4 is 16. Therefore, for a point x in a general position, 16 points from $\mathbb{Z}^3$ intersect the support of $M_\Xi(x)$ [9, II.15]. Since only $1/4$ of these points belong to the BCC lattice, only four BCC points fall inside the support of $M_\Xi(x)$. Similarly, the support of M_{Ξ^2} is a rhombic dodecahedron whose direction vectors are scaled by two. Therefore, its volume is 128, which implies that only 32 BCC points fall inside the support of $M_{\Xi^2}(x)$.

This fact implies that a C^0 reconstruction with second-order approximation power on BCC only needs four data points,[3] whereas for this smoothness and accuracy on the Cartesian lattice, trilinear interpolation requires a neighborhood of $2 \times 2 \times 2 = 8$ data points. Similarly, a C^2 reconstruction with fourth-order approximation on BCC only needs 32 data points, whereas for this smoothness and accuracy on the Cartesian lattice, tricubic B-spline requires a neighborhood of $4 \times 4 \times 4 = 64$ data points (see Table 1 for a summary). Hence, as we will see in Section 6, the computational cost of BCC reconstruction is significantly lower than a similar reconstruction on the Cartesian lattice with an equivalent sampling density.

5 EXPLICIT PIECEWISE POLYNOMIAL REPRESENTATION

The previous section defined the four-directional box spline on the BCC lattice and showed some of its main properties derived from its Fourier transform. However, a literal implementation of (9) and (13), as we implemented in [13], turned out to be extremely inefficient (especially in the case of the quintic box spline). Hence, although theoretically exciting, these splines were not useful in a practical setting. To make them practical for computer graphics and visualization applications, we derive a piecewise polynomial representation that allows an extremely fast evaluation as desired for these applications.

3. The four points in the linear interpolant construct a barycentric interpolation on the tetrahedron they form.

TABLE 1
Reconstruction Properties of the Proposed Box Splines on the BCC Lattice and Tensor-Product B-Splines on the Cartesian Lattice

	Cartesian B-splines		BCC Box Splines	
	Trilinear	Tricubic	Linear	Quintic
Degree of continuity	C^0	C^2	C^0	C^2
Order of approximation	2	4	2	4
Degree of poly-nomial pieces	3	9	1	5
# Data points in the support	8	64	4	32

TABLE 2
Rendering Times

	Marschner-Lobb		Carp		Teapot	
	C^0	C^2	C^0	C^2	C^0	C^2
BCC	11.99	63.75	33.85	184.33	29.05	150.45
Cartesian	21.49	122.69	60.08	363.62	50.27	294.55
Speedup	1.79	1.92	1.77	1.97	1.73	1.96

C^0 and C^2 indicate the linear and quintic box splines on the BCC lattice and the trilinear and tricubic B-splines on the Cartesian lattice, respectively.

5.1 Preliminaries and Outline of Derivation

In the following discussion, the symbol ∇_ξ denotes a (directional) *finite-difference operator* and is defined by $\nabla_\xi f(x) = f(x) - f(x - \xi)$. For a matrix of directions, Ξ, the difference operator is defined as successive applications of difference operators along each direction in $\Xi : \nabla_\Xi = \prod_{\xi \in \Xi} \nabla_\xi$. The corresponding differential operator is denoted by D_Ξ. Green's function of a differential operator is a function g that satisfies $Dg = \delta$, where δ denotes Dirac's delta (generalized) function. The Fourier transform of δ in the distributional sense is the constant function one.

Box splines, similar to B-splines, are piecewise polynomial functions with bounded support. In this section, we will see that the box spline M_Ξ can be derived by applying the finite-difference operator ∇_Ξ to a single function G_Ξ, which is Green's function for the differential operator D_Ξ corresponding to ∇_Ξ.

The essential idea in our derivation is to closely analyze the numerator and denominator of the Fourier transform of box splines (as in (11)). The numerator corresponds to the box spline's difference operator in the space domain, which is defined as

$$\nabla_\Xi \longleftrightarrow \prod_{\xi \in \Xi} 1 - \exp(-i\xi^T \omega). \qquad (15)$$

In Section 5.2, we will derive the weights and their respective positions in 3D as a discrete series for the finite-difference operator of our box splines.

Using distribution theory, we can identify the remaining part of (11) as the Fourier transform of G_Ξ in space domain, since

$$D_\Xi \longleftrightarrow \prod_{\xi \in \Xi} i\xi^T \omega, \text{ and } G_\Xi \longleftrightarrow \prod_{\xi \in \Xi} \frac{1}{i\xi^T \omega}.$$

If this differential operator D_Ξ is applied to its Green's function G_Ξ, Dirac's delta is obtained. However, if we apply the corresponding finite-difference operator to the Green's function, the box splines are obtained:

$$M_\Xi(x) = \nabla_\Xi G_\Xi(x).$$

We will show in Section 5.3 that the function G_Ξ is constructed by superpositions and linear transformations of a tensor product of (two-sided) signed monomials: $x^k \mathrm{sgn}(x)$. In Section 5.4, we will see that we can also derive box splines by applying the difference operators on their *truncated powers* T_Ξ. Truncated powers are very similar to G_Ξ, but instead of the two-sided signed monomials, they are constructed from one-sided monomials:

$$(x)_+^k = \begin{cases} x^k & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{16}$$

$(x)_-^k$ is also defined as $(x)_-^k = x^k - (x)_+^k$. Since one-sided monomials are supported on half-spaces, they are more convenient than Green's functions in derivations.

Four-directional box splines of generally higher degrees are obtained by n convolutions of the linear box spline, which amounts to

$$M_{\Xi^n}(x) = \nabla_{\Xi^n} T_{\Xi^n}(x). \tag{17}$$

These box splines are represented in the frequency domain by

$$\hat{M}_{\Xi^n}(\omega) = \hat{M}_\Xi^n(\omega) = \frac{1}{i^{4n}} \prod_{k=1}^4 \frac{(1 - \exp(-i\xi_k^T \omega))^n}{(\xi_k^T \omega)^n}.$$

For notational convenience, we introduce the scalar variables:

$$\begin{aligned} z_k &:= \exp(-i\xi_k^T \omega), \\ w_k &:= \xi_k^T \omega. \end{aligned} \tag{18}$$

This notation allows us to write the Fourier transforms of higher degree box splines more compactly as

$$\hat{M}_\Xi^n(\omega) = \prod_{k=1}^4 \frac{(1 - z_k)^n}{w_k^n}.$$

Furthermore, we note that due to the structure of $\sum_{k=1}^4 \xi_k = 0$ in Ξ, $\sum_{k=1}^4 w_k = 0$, and $\prod_{k=1}^4 z_k = 1$.

5.2 Difference Operator

The finite-difference operator can be represented as a filter. Its coefficients weight the Green's function that is shifted to the various lattice points, as in (17). The Z-domain representation of the difference operator allows for an easy polynomial representation of this discrete series:

$$\nabla_\Xi^n(z) = \prod_{k=1}^4 (1 - z_k)^n. \tag{19}$$

Expanding this equation for the linear box spline $n = 1$ and using the fact that $\prod_{k=1}^4 z_k = 1$, we get

$$\begin{aligned} \nabla_\Xi(z) = {}&2 - (z_1 + z_2 + z_3 + z_4) \\ &+ (z_1 z_2 + z_1 z_3 + z_1 z_4 + z_2 z_3 + z_2 z_4 + z_3 z_4) \\ &- (z_1 z_2 z_3 + z_1 z_2 z_4 + z_1 z_3 z_4 + z_2 z_3 z_4). \end{aligned} \tag{20}$$
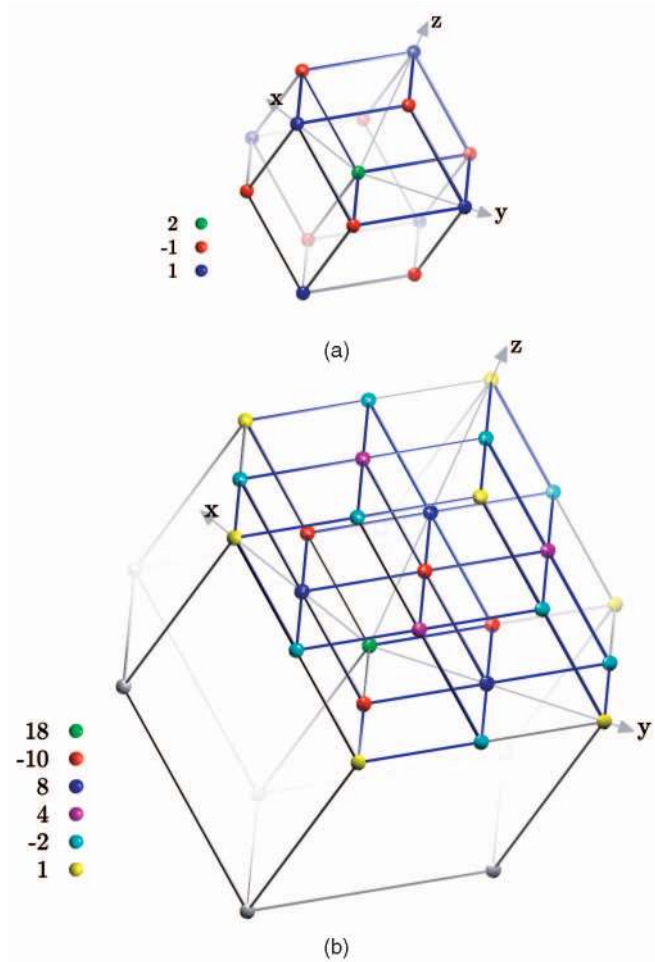

Fig. 5. (a) Weights for the difference operator of the linear box spline. (b) Weights for the difference operator of the quintic box spline. For simplicity of illustration, only one of the parallelepipeds constituting the rhombic dodecahedron has been drawn with all of its internal vertices.

For a more compact notation, we adopt a slightly different multinomial notation where the power operation on 4-tuples $Z = (z_1, z_2, z_3, z_4)$ by $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is defined as

$$Z^\alpha = \frac{1}{p(\alpha)} \sum_{(\beta_1,\dots,\beta_4) \in \mathrm{perm}(\alpha_1,\dots,\alpha_4)} z_1^{\beta_1} z_2^{\beta_2} z_3^{\beta_3} z_4^{\beta_4}, \tag{21}$$

where $\mathrm{perm}(\alpha)$ is the set of all permutations of α, and $p(\alpha)$ counts the number of permutations of repeated values in α. This is to avoid counting duplicate terms of the polynomial. For instance, if the value of α_1 is repeated in α r_1 times and the value of α_2 is repeated r_2 times, then $p(\alpha) = r_1! r_2!$. In this notation, the difference operator weights are represented as

$$\nabla_\Xi(z) = 2 - Z^{(1,0,0,0)} + Z^{(1,1,0,0)} - Z^{(1,1,1,0)}.$$

Note that since $z_1^{-1} = z_2 z_3 z_4$, both $Z^{(1,0,0,0)}$ and $Z^{(1,1,1,0)}$ denote the same set of monomials, which contain exactly one lattice vector. Geometrically, one can visualize the weights being 2 at the origin, -1 on all BCC lattice points reachable by exactly one lattice vector (positive or negative), and $+1$ on lattice points that can be reached by exactly two lattice vectors. This is illustrated in Fig. 5a. Similarly, the

difference operator weights for the quintic box spline $n = 2$ can be derived from (19) in the compact form as

$$\nabla^2_{\Xi}(z) = 18 - 10Z^{(1,0,0,0)} + 8Z^{(1,1,0,0)} + 4Z^{(2,1,1,0)}$$
$$- 2Z^{(2,1,0,0)} + Z^{(2,0,0,0)} + Z^{(2,2,0,0)}. \tag{22}$$

These weights are on BCC lattice points on a rhombic dodecahedron with a neighborhood two times larger than that of the linear-box-spline case and are displayed in Fig. 5b.

5.3 Green's Function

We now describe a procedure to derive the space-domain representation of the Green's functions of our box splines. We make use of the w_k variables introduced in (18):

$$\hat{G}_{\Xi}(\omega) = \prod_{k=1}^{4} \frac{1}{\xi_k^{\mathrm{T}}\omega} = \prod_{k=1}^{4} \frac{1}{w_k}.$$

The objective is to rewrite $\hat{G}_{\Xi}$ into a number of terms, each of which contains only three of the four w_k variables. Such a three-variable expression can then be written as a linear transformation of a trivariate function whose inverse Fourier transform can be obtained by a tensor product. The general idea is to exploit the relation $w_1 + w_2 + w_3 = -w_4$ to reduce the number of variables in the denominator and obtain a sum of terms with one less variable. This helps to eliminate any fourth variable with the help of the proper numerator, and consequently, we introduce new terms in the expression while increasing the power of w_4. This procedure is the frequency-domain reasoning of the spatial-domain recursive structure of the box splines. We can always apply this procedure since whenever the number of directions n is greater than the dimension of the space s, the additional directions of the box spline can be written as the linear combination of the s linearly independent vectors. The $\hat{G}_{\Xi}$ of the linear box spline can be rewritten as

$$\hat{G}_{\Xi}(\omega) = \frac{1}{w_1 w_2 w_3 w_4} = -\frac{\frac{w_1 + w_2 + w_3}{w_4}}{w_1 w_2 w_3 w_4}$$
$$= \frac{-1}{w_4^2 w_1 w_2 w_3} \times (w_1 + w_2 + w_3).$$

Although the general Green's function of the nth box spline is $\hat{G}_{\Xi^n} = (\hat{G}_{\Xi})^n$, for the linear box spline, we have $n = 1$:

$$\hat{G}_{\Xi}(\omega) = \frac{-1}{w_4^2} \left(\frac{1}{w_1 w_2} + \frac{1}{w_1 w_3} + \frac{1}{w_2 w_3} \right). \tag{23}$$

Similarly, the quintic box spline's Green's function is obtained by $n = 2$:

$$\hat{G}_{\Xi^2}(\omega) = \hat{G}_{\Xi}^2(\omega) =$$
$$\frac{1}{w_4^4} \left(\frac{1}{w_1^2 w_2^2} + \frac{1}{w_1^2 w_3^2} + \frac{1}{w_2^2 w_3^2} \right) +$$
$$\frac{-2}{w_4^5} \left(\frac{1}{w_1^2 w_3} + \frac{1}{w_1^2 w_2} + \frac{1}{w_2^2 w_3} + \frac{1}{w_2^2 w_1} + \frac{1}{w_3^2 w_1} + \frac{1}{w_3^2 w_2} \right) + \tag{24}$$
$$\frac{6}{w_4^6} \left(\frac{1}{w_2 w_3} + \frac{1}{w_1 w_3} + \frac{1}{w_1 w_2} \right).$$

Now, we can move back to using the frequency variables $\omega = (\omega_x, \omega_y, \omega_z)$. We first define these building-block functions:

$$\hat{\rho}_1(\omega_x, \omega_y, \omega_z) = \frac{-1}{\omega_x \omega_y \omega_z^2},$$

$$\hat{\rho}_2(\omega_x, \omega_y, \omega_z) = \frac{1}{\omega_x^2 \omega_y^2 \omega_z^4} - 2 \left(\frac{1}{\omega_x^2 \omega_y} + \frac{1}{\omega_x \omega_y^2} \right) \frac{1}{\omega_z^5} + \frac{6}{\omega_x \omega_y \omega_z^5}. \tag{25}$$

These functions are useful since the $\hat{G}_{\Xi}$ of our box splines are essentially linear transformations (for example, $(w_1, w_2, w_4) = \Xi_{\{1,2,4\}}^{\mathrm{T}} \omega$) and superpositions of these building-block functions:

$$\hat{G}_{\Xi}(\omega) = \hat{\rho}_1(\Xi_{\{1,2,4\}}^{\mathrm{T}} \omega) + \hat{\rho}_1(\Xi_{\{1,3,4\}}^{\mathrm{T}} \omega) + \hat{\rho}_1(\Xi_{\{2,3,4\}}^{\mathrm{T}} \omega),$$
$$\hat{G}_{\Xi^2}(\omega) = \hat{\rho}_2(\Xi_{\{1,2,4\}}^{\mathrm{T}} \omega) + \hat{\rho}_2(\Xi_{\{1,3,4\}}^{\mathrm{T}} \omega) + \hat{\rho}_2(\Xi_{\{2,3,4\}}^{\mathrm{T}} \omega).$$

Here, the subscript $\{i, j, k\}$ indicates the matrix formed by the ith, jth, and kth columns of Ξ.

We now derive the inverse Fourier transform of these building-block functions. First, we recognize that the Fourier inverse of $1/(i\omega)^k$ is the two-sided monomial [1]:

$$\frac{(x)_{\mathrm{sgn}}^k}{k!} := \frac{1}{2} \frac{x^k \mathrm{sgn}(x)}{k!} \longleftrightarrow \frac{1}{(i\omega)^{k+1}}. \tag{26}$$

We can derive the space-domain representation of our building-block functions as a tensor-product inverse Fourier transform of the equations in (25):

$$\rho_1(x, y, z) = - (x)_{\mathrm{sgn}}^0 (y)_{\mathrm{sgn}}^0 (z)_{\mathrm{sgn}},$$
$$\rho_2(x, y, z) = \frac{1}{3!} (x)_{\mathrm{sgn}} (y)_{\mathrm{sgn}} (z)_{\mathrm{sgn}}^3 +$$
$$\frac{-2}{4!} \left[(x)_{\mathrm{sgn}} (y)_{\mathrm{sgn}}^0 + (x)_{\mathrm{sgn}}^0 (y)_{\mathrm{sgn}} \right] (z)_{\mathrm{sgn}}^4 + \tag{27}$$
$$\frac{6}{5!} (x)_{\mathrm{sgn}}^0 (y)_{\mathrm{sgn}}^0 (z)_{\mathrm{sgn}}^5.$$

If Q is an invertible matrix, we know that $f(Qx) \longleftrightarrow \hat{f}((Q)^{-1}\omega)/|\det Q|$. Therefore, we can write the space-domain representation of the Green's function of these box splines as

$$G_{\Xi}(x) = \frac{1}{4} \left(\rho_1(\Xi_{\{1,2,4\}}^{-1} x) + \rho_1(\Xi_{\{1,3,4\}}^{-1} x) + \rho_1(\Xi_{\{2,3,4\}}^{-1} x) \right),$$
$$G_{\Xi^2}(x) = \frac{1}{4} \left(\rho_2(\Xi_{\{1,2,4\}}^{-1} x) + \rho_2(\Xi_{\{1,3,4\}}^{-1} x) + \rho_2(\Xi_{\{2,3,4\}}^{-1} x) \right), \tag{28}$$

where $x = (x, y, z)$, and

$$|\det \Xi_{\{1,2,4\}}^{-1}| = |\det \Xi_{\{1,3,4\}}^{-1}| = |\det \Xi_{\{2,3,4\}}^{-1}| = 1/4.$$

5.4 Truncated Power

Recall that the Green's functions were constructed from the signed monomials $x^k \mathrm{sgn}(x)$ in (27). The differential operator D_{Ξ}, when applied on these signed monomials, transformed in (28), results in a δ function. Consequently, this differential operator annihilates all *polynomials* of degree $\leq k$ encountered in the signed polynomials in the Green's function. Similarly, the finite-difference operator annihilates all of these polynomials [9, I.32].

Since box splines are obtained by applying the difference operator on the Green's function, we can add or subtract any polynomials up to degree k found in the

Green's function. Therefore, the box spline can also be obtained by applying the difference operator to a Green's function that is obtained from $\frac{1}{2}(x^k \mathrm{sgn}(x) - x^k) = -(x)_-^k$ or $\frac{1}{2}(x^k \mathrm{sgn}(x) + x^k) = (x)_+^k$. The contributions from adding or subtracting x^k is eliminated since the difference operator annihilates the polynomials made of x^k. The advantage of working with these one-sided monomials is that they are supported on half-spaces, whereas the support of Green's functions in (28) is the entire space.

Therefore, we redefine the building-block functions by using $(x)_+^k$, $(y)_+^k$, and $-(z)_-^k$:

$$\tau_1(x, y, z) = (x)_+^0 (y)_+^0 (z)_-,$$
$$\tau_2(x, y, z) = \frac{-1}{3!}(x)_+(y)_+(z)_-^3 +$$
$$\frac{2}{4!}\left[(x)_+(y)_+^0 + (x)_+^0(y)_+\right](z)_-^4 +$$
$$\frac{-6}{5!}(x)_+^0(y)_+^0(z)_-^5. \qquad (29)$$

Note that the support of these building-block functions is on points $x \in \mathbb{R}^3$ such that $x, y > 0$ and $z < 0$. We further derive the truncated power functions to be

$$T_\Xi(x) = \frac{1}{4}\left(\tau_1(\Xi_{\{1,2,4\}}^{-1}x) + \tau_1(\Xi_{\{1,3,4\}}^{-1}x) + \tau_1(\Xi_{\{2,3,4\}}^{-1}x)\right),$$
$$T_{\Xi^2}(x) = \frac{1}{4}\left(\tau_2(\Xi_{\{1,2,4\}}^{-1}x) + \tau_2(\Xi_{\{1,3,4\}}^{-1}x) + \tau_2(\Xi_{\{2,3,4\}}^{-1}x)\right). \qquad (30)$$

The crucial point here is that the values of truncated power functions at any point x are affected only by one of the three terms on the right-hand sides of the above equations. To see this fact, recall that $\tau_1(x, y, z)$ and $\tau_2(x, y, z)$ are nonzero only when $x, y > 0$ and $z < 0$. The support of each building-block function is transformed in (30) to cones formed by columns of Ξ (for example, the support of $\tau_1(\Xi_{\{1,2,4\}}^{-1}x)$ is all points $x = \Xi_{\{1,2,4\}}(t_1, t_2, t_3)$ for $t_1, t_2 > 0$ and $t_3 < 0$). The support of the building-block functions is transformed as

$$\Xi_{\{1,2,4\}}^{-1}x = \frac{1}{2}(x - z, y - z, x + y),$$
$$\Xi_{\{1,3,4\}}^{-1}x = \frac{1}{2}(x - y, z - y, x + z), \qquad (31)$$
$$\Xi_{\{2,3,4\}}^{-1}x = \frac{1}{2}(y - x, z - x, y + z).$$

Therefore, the support of $\tau_1(\Xi_{\{1,2,4\}}^{-1}x)$ is the cone that is the intersection of the half-spaces determined by $x - z > 0$, $y - z > 0$, and $x + y < 0$ (see Fig. 6). Similarly, the support of $\tau_1(\Xi_{\{1,3,4\}}^{-1}x)$ is the cone that is the intersection of the half-spaces determined by $x - y > 0$, $z - y > 0$, and $x + z < 0$. Since $y - z < 0$ and $z - y < 0$ are disjoint, the support of $\tau_1(\Xi_{\{1,2,4\}}^{-1}x)$ and $\tau_1(\Xi_{\{1,3,4\}}^{-1}x)$ are disjoint. Therefore, the supports of each building-block function transformed in (30) are nonoverlapping.

The support of the truncated power functions of the box splines is the union of each cone formed by the matrices of the equations in (30). The support of each transformed building-block function along with the support of the truncated power of the box splines is illustrated in Fig. 6. The red arrows indicate the half-spaces in the support.
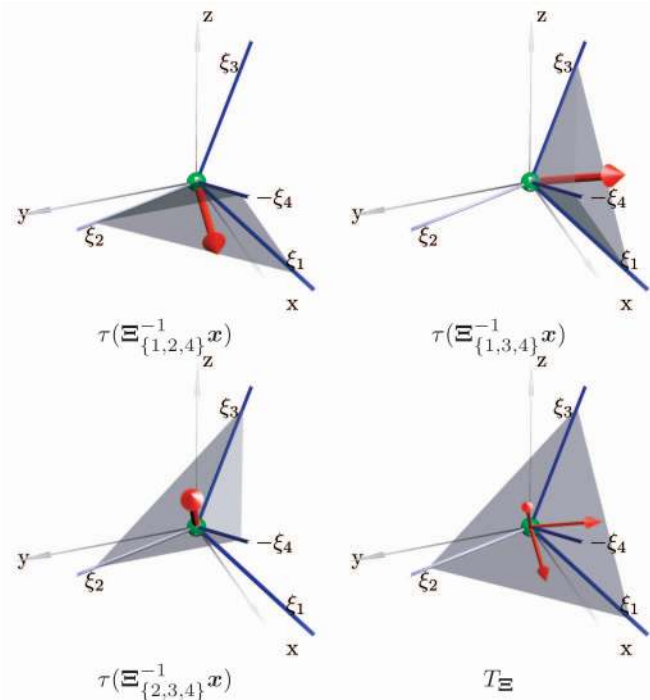

Fig. 6. The support of the truncated power function is the cone formed by the three directions in $\Xi_{\{1,2,3\}}$. This volume is a *disjoint union* of the supports of three τ functions, each transformed by $\Xi_{\{1,2,4\}}^{-1}$, $\Xi_{\{1,3,4\}}^{-1}$, and $\Xi_{\{2,3,4\}}^{-1}$.

Therefore, one can verify that the support of the truncated power functions is the *union* of the half-spaces determined by $x + y < 0$, $x + z < 0$, and $y + z < 0$.

Since the supports of the transformed τ functions are nonoverlapping, at any point x, only one of the three transformed τ's contribute to the value of the truncated power. For x to be within the support of one of the transformed τ functions, its first and second components of the transformed vector need to be positive, whereas the third component needs to be negative. Under this assumption, the last component of each vector of the right-hand side of (31) is the sum of the two largest values out of x, y, and z. Moreover, the other two components in each set is the difference of the largest and the middle value from the minimum of the three. We also notice that these basic building-block functions are symmetric with respect to the first and second components of the position vector x of their argument (see (27)). For example, when τ is transformed by $\Xi_{\{1,2,4\}}^{-1}$, its support is determined by the region specified by $x > z$, $y > z$, and $x + y < 0$. Using these observations, we can write the truncated power in terms of only one basic building-block function:

$$T_\Xi(x, y, z) = \frac{1}{4}\tau_1\left(\frac{1}{2}(\tilde{x} - \tilde{z}), \frac{1}{2}(\tilde{y} - \tilde{z}), \frac{1}{2}(\tilde{x} + \tilde{y})\right), \qquad (32)$$

$$T_{\Xi^2}(x, y, z) = \frac{1}{4}\tau_2\left(\frac{1}{2}(\tilde{x} - \tilde{z}), \frac{1}{2}(\tilde{y} - \tilde{z}), \frac{1}{2}(\tilde{x} + \tilde{y})\right), \qquad (33)$$

where

$$\tilde{x} = \max(x, y, z), \quad \tilde{y} = \mathrm{mid}(x, y, z), \text{ and } \tilde{z} = \min(x, y, z). \qquad (34)$$
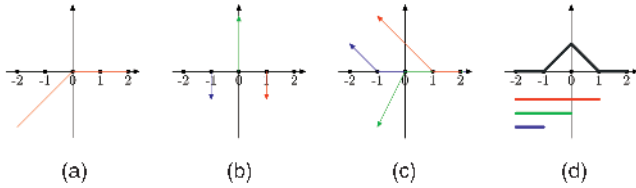
Fig. 7. Convolution of the truncated power $(x)_-$ with the difference operator, a 1D example. (a) The truncated power $(x)_-$. (b) The discrete-difference operator weights. (c) The convolution by overlaying the truncated power functions. (d) The resulting convolution yields the linear B-spline.

5.5 Efficient Evaluation

Having obtained the explicit form of the truncated power, we shall apply the difference operator derived in Section 5.2 to T_{Ξ^n} in order to obtain M_{Ξ^n}. This operation indicated as in (17) can be implemented as a convolution of the finite-difference operator sequence with the truncated power. In this section, we exploit the symmetries in the support of these box splines and find a *region* for the efficient evaluation of this convolution.

5.5.1 Region of Evaluation: 1D Example

To better understand the procedure, we use to derive the polynomial pieces of the box spline, we first illustrate this procedure in 1D for a linear B-spline ($\Xi = \begin{bmatrix} 1 & -1 \end{bmatrix}$). In this case, the Green's function is $-\frac{1}{2}x\operatorname{sgn}(x)$, and the truncated power is: $T(x) = (x)_-$. The difference operator is represented in the Z-domain by $\nabla(z) = -z^{-1} + 2 - z$. The linear B-spline is obtained by the following convolution:

$$\Lambda(x) = -T(x+1) + 2T(x) - T(x-1).$$

The process of this convolution is illustrated in Fig. 7. Fig. 7a shows the truncated power $(x)_-$, Fig. 7b shows the difference operator weights, Fig. 7c shows the result of the convolution illustrated by overlaying the truncated powers at their respective difference operator *sites*, and Fig. 7d shows the resulting B-spline. In Fig. 7d, the red band indicates the regions of the x-axis that are affected by the convolution site at 1, the green band indicates the region that is affected by the convolution site at 0, and the blue band indicates the region that is affected by the convolution site at -1. The symmetry of the support of the linear B-spline suggests an efficient evaluation in the interval of $[0, 1]$, where only one convolution site contributes to the values of the B-spline in this region.

Therefore, for an efficient evaluation of the linear B-spline, we would map any point in $[-1, 1]$, the support of the B-spline, to the interval $[0, 1]$ using the symmetry of its support. Once this mapping is performed, the B-spline can be computed by evaluating the truncated power shifted only to the sites that affect this region. In the case of the linear B-spline, there is only one site that affects this region, which is $T(x-1)$.

5.5.2 Region of Evaluation: Trivariate Case

Since our trivariate box splines are obtained through a projection along the antipodal axis of a tesseract, they exhibit the symmetries present in their polyhedral support, which is a rhombic dodecahedron. We exploit the symmetries present
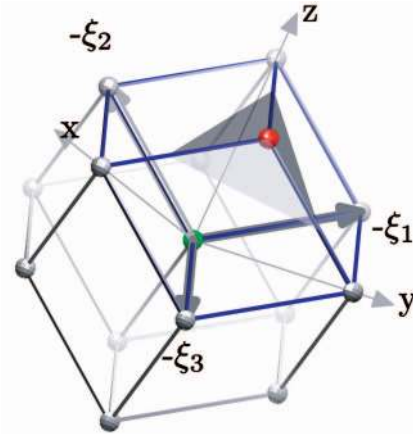

Fig. 8. When computing the operation of the difference operator on the truncated power, only one term, which shifts the truncated power to (1, 1, 1), affects the parallelepiped of interest formed by $-\Xi_{\{1,2,3\}}$. The support of the truncated powers shifted to the other sites does not intersect this parallelepiped.

within the rhombic dodecahedron to achieve an efficient evaluation method for the linear and quintic box splines.

First, we observe that a rhombic dodecahedron can be decomposed into four nonoverlapping parallelepipeds in two different ways. For a rhombic dodecahedron formed by the vectors in Ξ as in Fig. 4, one can construct four parallelepipeds, each formed by three of the four vectors from Ξ. Alternatively, one can choose the negative directions from $-\Xi$ to decompose the rhombic dodecahedron into four parallelepipeds. Therefore, we can confine the evaluation region to one of these parallelepipeds, and the evaluation at the other points can be inferred by symmetry.

The support of T_Ξ (or T_{Ξ^2}) is the positive cone of $\Xi_{\{1,2,3\}}$, as in Fig. 6. A minimal number of convolution sites contribute to the value of the box spline in the parallelepiped that is cornered at the origin and formed by $-\xi_1$, $-\xi_2$, and $-\xi_3$. This parallelepiped contains the positive octant of $\mathbb{R}^3$ and is illustrated with blue edges in Fig. 8. Similarly, for the quintic box spline, a minimal number of convolution sites contribute to the parallelepiped formed by $-2\xi_1$, $-2\xi_2$, and $-2\xi_3$, which is illustrated in Fig. 9.

5.5.3 Linear Box Spline

The operation of the difference operator on the truncated power is a sum of truncated power functions shifted and weighted according to the difference operator sites, as in Fig. 5a. As the support of the truncated power is limited to the cone of the direction vectors in $\Xi_{\{1,2,3\}}$, only one of the terms of the convolution contributes to the value of M_Ξ in the parallelepiped that we deal with; this term is the one obtained from shifting the T_Ξ to (1, 1, 1) and multiplying by -1 the difference operator weight at this point (Fig. 5a). None of the other difference operator sites affect this region of interest, as illustrated in Fig. 8.

Therefore, a point in the parallelepiped of focus is characterized by $(x, y, z) = -\Xi_{\{1,2,3\}}(t_1, t_2, t_3)$, where $0 \le t_1, t_2, t_3 < 1$. Using the min/mid/max variables introduced in (34), we have
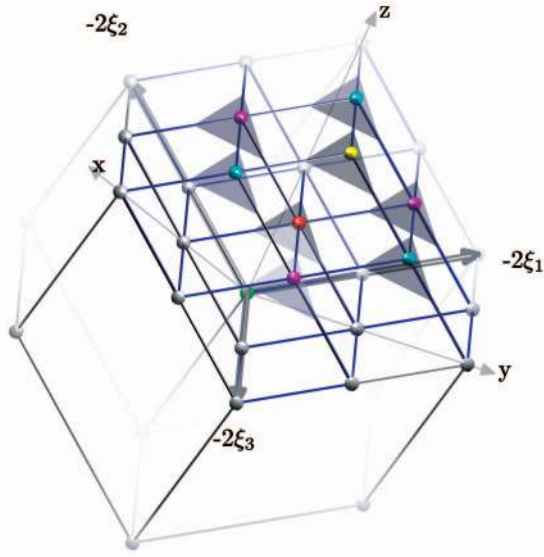
Fig. 9. When computing the convolution of the difference operator and the truncated power, only eight terms of the convolution affect the parallelepiped of interest formed by $-2\Xi_{\{1,2,3\}}$. The support of the truncated power shifted to each site is the cone indicated by the gray region. The support of the truncated power shifted to the other convolution sites does not intersect this parallelepiped.

$$M_\Xi(x,y,z) = (-1)T_\Xi(x-1,y-1,z-1)$$
$$= -\frac{1}{4}\tau_1 \begin{pmatrix} 1/2(\tilde{x}-1-(\tilde{z}-1)), \\ 1/2(\tilde{y}-1-(\tilde{z}-1)), \\ 1/2(\tilde{x}-1+\tilde{y}-1) \end{pmatrix} \qquad (35)$$
$$= -\frac{1}{8}(\tilde{x}+\tilde{y}-2)$$
$$= \frac{1}{4}\left(1 - \frac{1}{2}\left(\max(x,y,z) + \mathrm{mid}(x,y,z)\right)\right),$$

which agrees with the geometric simplification we derived in [13], normalized according to the sampling density of the BCC lattice described by (1).

5.5.4 Quintic Box Spline

The same procedure as for the linear box spline can be used. The difference operator for the quintic box spline is shown in Fig. 5b. The support of the truncated power is limited to the cone of the direction vectors in $\Xi_{\{1,2,3\}}$ such that only eight of the terms of the convolution contribute to the value of M_{Ξ^2} in the parallelepiped that we focus on; see Fig. 9. As we saw in Section 4.4, the size of the support of the quintic box spline is doubled from that of the linear box spline; therefore, the parallelepiped of focus is now eight times the size of the corresponding one in the linear-box-spline case; therefore, for a point in the parallelepiped of focus characterized by $(x,y,z) = -\Xi_{\{1,2,3\}}(t_1,t_2,t_3)$, where $0 \le t_1,t_2,t_3 < 2$, we have

$$M_{\Xi^2}(x,y,z) = T_{\Xi^2}(x-2,y-2,z-2)$$
$$- 10T_{\Xi^2}(x-1,y-1,z-1) - 2T_{\Xi^2}(x-3,y-1,z-1)$$
$$- 2T_{\Xi^2}(x-1,y-3,z-1) - 2T_{\Xi^2}(x-1,y-1,z-3)$$
$$+ 4T_{\Xi^2}(x-2,y-2,z) + 4T_{\Xi^2}(x-2,y,z-2)$$
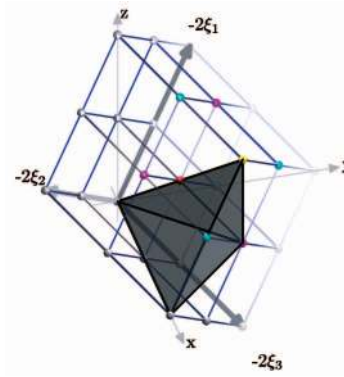$$+ 4T_{\Xi^2}(x,y-2,z-2),$$

Fig. 10. The region specified by $x+y < 4$ and $x > y > z > 0$ is illustrated with the dark tetrahedron. This tetrahedron is formed by connecting the origin to the face, which is the triangle that is one quarter of the rhombic face of the original rhombic dodecahedron.

where T_{Ξ^2} is defined as in (33). The shifts in the above equation are shifts to the difference operator sites, which are the colored nodes in Fig. 9.

Using the symmetries of the rhombic dodecahedron, we can confine the evaluation region to a tetrahedron that has a vertex at the origin, and its apex (which is on the plane $x+y=4$) is a quarter of one face of the original parallelepiped since these faces are rhomboids and have fourfold symmetry. The rhomboid face of the parallelepiped of our focus lies in the plane specified by $x+y=4$ in Fig. 10. Out of four possible choices, we pick this tetrahedron so that it contains the positive octant completely. This region is specified by its four bounding planes: $x+y < 4$, $x > y$, $y > z$, and $z > 0$. This tetrahedron is illustrated by the dark tetrahedron in Fig. 10. It is partitioned into four regions formed by the intersections with four of the eight subparallelepipeds that constitute the original parallelepiped of focus. These subparallelepipeds are highlighted in green in Fig. 11. These four regions are identified by

$$R_1 : x+y < 2; \qquad R_2 : x+y > 2, x+z < 2;$$
$$R_3 : x+z > 2, y+z < 2; \qquad R_4 : y+z > 2.$$

These regions are determined by posing the restriction of being in the dark tetrahedron of focus, which is specified by $x > y > z > 0$ and each of the four subparallelepipeds that intersect this tetrahedron. These subparallelepipeds are specified by $-\Xi_{\{1,2,3\}}t$, where the subparallelepiped of region R_1 is specified by $0 \le t_1,t_2,t_3 < 1$, the subparallelepiped of region R_2 is specified by $0 \le t_1,t_2 < 1$ and $1 \le t_3 < 2$, the subparallelepiped of region R_3 is specified by $0 \le t_1 < 1$ and $1 \le t_2,t_3 < 2$, and the subparallelepiped of region R_4 is specified by $1 \le t_1,t_2,t_3 < 2$.

In each of these regions, illustrated in Fig. 11, the box spline will be represented as a separate polynomial.

Region R_1 is affected by all of the eight difference operator sites in the parallelepiped of focus. Region R_2 is affected only by four sites at $(3,1,1)$, $(2,2,2)$, $(1,3,1)$, and $(2,2,0)$. Region R_3 is affected by two sites at $(3,1,1)$ and $(2,2,2)$. Finally, region R_4 is affected only by $(2,2,2)$. Therefore, we simplify the polynomials in each region separately. Using constants $\alpha := 1/3840$, $\beta := 1/1920$, and $\gamma := 1/960$, we have
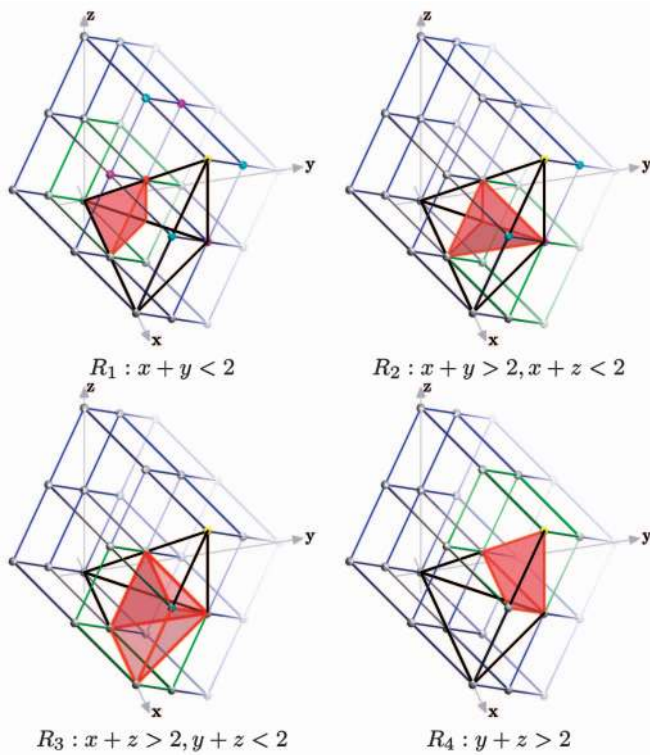
Fig. 11. Different regions of the evaluation domain are the intersection of subparallelepipeds (highlighted in green) and the tetrahedron in focus (black edges). The box spline is a homogeneous polynomial in each region.

Region R_1, $M_{\Xi^2}(x, y, z) =$

$$\alpha(x + y - 4)^3(-3xy - 5z^2 + 2x + 2y + 20z + x^2 + y^2 - 24)$$
$$+ \beta(x + z - 2)^3(x^2 - 9x - 3xz + 10y - 5y^2 + 14 + 11z + z^2)$$
$$+ \beta(y + z - 2)^3(46 - 30x - z - y + 3zy + 5x^2 - y^2 - z^2)$$
$$- \gamma(x + y - 2)^3(x^2 + x - 3xy - 5z^2 + y^2 + y - 6).$$
(36)

Region R_2, $M_{\Xi^2}(x, y, z) =$

$$\alpha(x + y - 4)^3(-3xy - 5z^2 + 2x + 2y + 20z + x^2 + y^2 - 24)$$
$$- \beta(x + z - 2)^3(-z^2 - 11z + 3xz - 14 + 5y^2 + 9x - 10y - x^2)$$
$$- \beta(y + z - 2)^3(-46 + z + 30x + y - 3zy - 5x^2 + y^2 + z^2).$$
(37)

Region R_3, as illustrated in Fig. 11, is not a simple tetrahedron with homogeneous regions with respect to the site located at (3, 1, 1). When the truncated power is centered at this site, two of the three components constituting the truncated power (see Fig. 6) intersect this region. Therefore, there are two subcases for the contribution of the truncated power centered at (3, 1, 1). However, the contribution from (2, 2, 2) remains homogeneous as only one of the three components of the truncated power (see Fig. 6) contributes to this region:

Region $R_{3A}\{x - z > 2\}$, $M_{\Xi^2}(x, y, z) =$

$$\alpha(x + y - 4)^3(-x^2 + 8x + 3xy - y^2 + 5z^2 - 16 - 12y). \quad (38)$$

Region $R_{3B}\{x - z < 2\}$, $M_{\Xi^2}(x, y, z) =$

$$\alpha(x + y - 4)^3(-3xy - 5z^2 + 2x + 2y + 20z + x^2 + y^2 - 24)$$
$$- \beta(y + z - 2)^3(30x + z - 46 - 3yz + y - 5x^2 + y^2 + z^2).$$
(39)

Region R_4, $M_{\Xi^2}(x, y, z) =$

$$\alpha(x + y - 4)^3(-3xy - 5z^2 + 2x + 2y + 20z + x^2 + y^2 - 24).$$
(40)

Therefore, a fast evaluation of the box spline can be obtained for any (x, y, z) by first transforming it to the tetrahedron of focus (the dark tetrahedron in Fig. 10) by taking $(x, y, z) = (|x|, |y|, |z|)$ and then sorting $(x, y, z) = (\max(x, y, z), \mathrm{mid}(x, y, z), \min(x, y, z))$. Once these transforms are performed, we can test for the appropriateness of the above five regions, and the polynomial form can be evaluated. We also note that due to the explicit piecewise polynomial form above and the relatively small support of these polynomial regions, our evaluation is numerically stable.

As a summary, we include the pseudocode for the evaluation of the quintic box spline referring to the regions above.

function fourdirection_boxspline(x, y, z)
% Transform the point to the tetrahedron of focus in Fig. 10
x = **abs**(x); y = **abs**(y); z = **abs**(z);
sort x, y, z in decreasing order
% For the linear box spline, the apex of the
     focus-tetrahedron is
% on the plane (x + y) = 2: the boundary
if ((x + y) > 2) **return** 0;
return $(2-(x + y))/8$; % Linear case; see (35)

% For the quintic box spline the plane (x + y) = 4 is the
     boundary:
if ((x + y) > 4) **return** 0;
if ((x + y) < 2) % Region R_1
          return $M_{\Xi_{R_1}}(x, y, z)$ as in (36);
elseif ((x + z) < 2) % Region R_2
          return $M_{\Xi_{R_2}}(x, y, z)$ as in (37);
elseif ((y + z) < 2) % Region R_3
          if ((x-z) > 2) % Region R_{3A}
                return $M_{\Xi_{R_{3A}}}(x, y, z)$ as in (38);
          else % Region R_{3B}
                return $M_{\Xi_{R_{3B}}}(x, y, z)$ as in (39);
else % Region R_4
          return $M_{\Xi_{R_4}}(x, y, z)$ as in (40);
end

6 RESULTS

In this section, we describe our experimental results and compare the BCC sampling scheme to the traditional Cartesian sampling on volumetric data sets. In order to examine the reconstruction schemes discussed in this paper, we have implemented a ray caster to render images from the Cartesian and the BCC sampled volumetric data sets. We have chosen the synthetic data set (Fig. 12) first proposed in [21] as a benchmark for our comparisons.
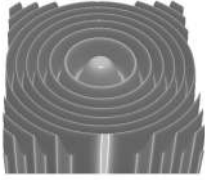
Fig. 12. The explicit function introduced by Marschner and Lobb.

The function was sampled at a resolution of $41 \times 41 \times 41$ on the Cartesian lattice and at an almost[4] equivalent sampling on the BCC lattice of $32 \times 32 \times 64$. The images in Fig. 13 are rendered using the quintic box spline on the BCC sampled data sets and the tricubic B-spline on the Cartesian sampled data sets. The analytical function was rendered by evaluating the actual function proposed in [21]. The images in the second row in Fig. 13 document the corresponding error images that are obtained from the angular error that occurred in estimating the normal by central differencing on the reconstructed function. Although a direct reconstruction of the normal is possible by using the analytical gradient of the reconstruction kernel, we chose central differencing with a relatively small step on the reconstructed function to approximate the true gradient. Central differencing is commonly the method of choice in the visualization domain, and there is no reason to believe that it performs any better or worse than taking the analytical derivative of the reconstruction kernel [26]. The gray value of 255 (white) denotes an angular error of 0.3 radian between the computed normal and the exact normal. The superiority of BCC sampling is apparent by comparing the images in Figs. 13a and 13b, as these are obtained from an equivalent sampling density over the volume. Although the lobes are mainly preserved in the BCC case, they are more smoothed out in the case of Cartesian sampling. This is also confirmed by their corresponding error images in the second row in Fig. 13.

Typical volumetric data are scanned on and reconstructed from the Cartesian lattice. A suitable (antialiasing) prefiltering step is applied to limit the spectrum of the sampled data within the Nyquist region, which is the Voronoi cell of the reciprocal lattice. For the BCC lattice, this cell is clearly different from the one of the Cartesian lattice. Therefore, the ultimate test of the BCC reconstruction on real-life data sets cannot be performed until there are true BCC sampling scanners available.

Nevertheless, we constructed comparable BCC and Cartesian data sets by merely subsampling a fairly densely sampled Cartesian data set. Cartesian sampled data can be downsampled onto a BCC lattice by retaining Cartesian points whose x, y, and z-coordinates are all odd or all even. Such a BCC lattice has a quarter of the sampling density of the original data set. To obtain an almost equivalent subsampling ratio into a lower resolution Cartesian data set, we choose a rational subsampling scheme where each

4. Since a finite sampling of a volume cannot produce the exact same number of samples for the BCC and Cartesian sampling patterns, for our discrete resolutions, we chose the resolutions conservatively in favor of the Cartesian sampling. Therefore, the actual sampling density in the Cartesian sampled data sets is slightly higher than the BCC sampling density.
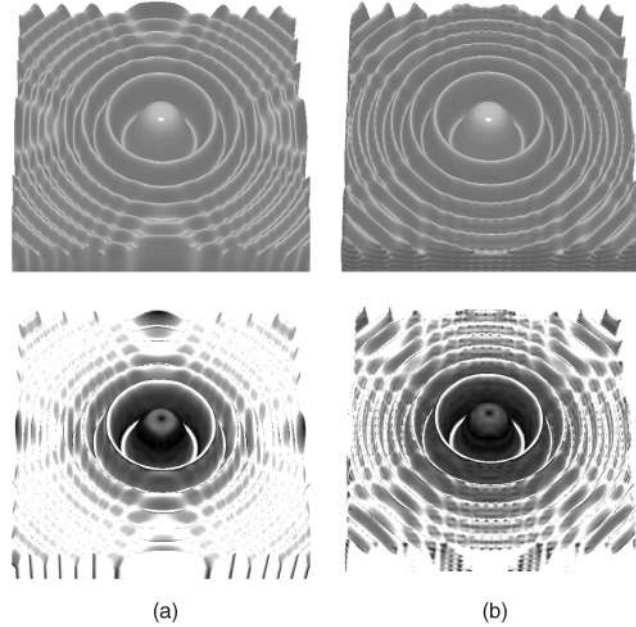

(a)   (b)

Fig. 13. The Marschner-Lobb data set. (a) Sampled on the Cartesian lattice at a resolution of $41 \times 41 \times 41$. (b) Sampled on the BCC lattice at a resolution of $32 \times 32 \times 64$. The first row illustrates the volume rendering of the sampled data using the tricubic B-spline on the Cartesian and our quintic box spline on the BCC data set. The second row illustrates the corresponding angular errors in estimating the gradient on the isosurface from the reconstruction process. An angular error of 0.3 radian is mapped to white. The darker error image of the BCC data confirms smaller errors and a more accurate reconstruction.

dimension of the original Cartesian data set is subsampled by 63/100 since $(63/100)^3 = 0.250047 \approx 1/4$. To achieve this subsampling, we first upsampled by zero-padding in the frequency domain by a factor of 63. Then a subsampling of the rate 1/100 yields the properly subsampled Cartesian volume.

As a first practical case, we chose the Boston Teapot data set. The original data set has a resolution of $162 \times 162 \times 113$. The subsampled Cartesian volume has a resolution of $103 \times 103 \times 72$, and the subsampled BCC volume has a resolution of $81 \times 81 \times 113$. These volumes were rendered using the tricubic B-spline on the Cartesian lattice and the quintic box spline on the BCC lattice in Fig. 14. These images demonstrate the superiority of the BCC sampling scheme since the Cartesian undersampled data set developed cracks on the surface of the teapot lid, whereas the BCC undersampled data set maintains the original content with higher fidelity. We also examined the carp fish data set. The original data set has a resolution of $256 \times 256 \times 256$. The subsampled Cartesian volume has a resolution of $140 \times 140 \times 140$, and the subsampled BCC volume has a resolution of $111 \times 111 \times 222$. These volumes were also rendered with the tricubic B-spline and the quintic box spline, respectively; see Fig. 15. Again, these results show the superiority of the BCC sampling scheme since the Cartesian undersampled data set misses the fish tail and most of the bones.

In [13], we have discussed the issues pertaining to linear order interpolation. Although in the Cartesian volumes they demonstrate grid-aligned artifacts, in BCC, they display girdering artifacts [4]. The result of linear box spline on the
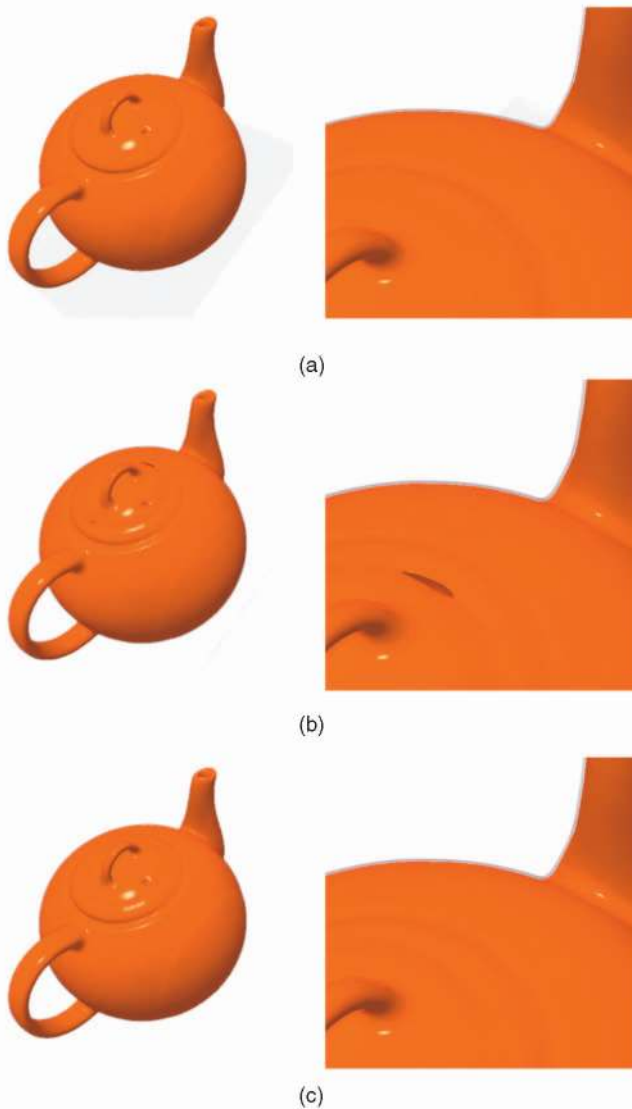
Fig. 14. The Boston teapot data set. (a) The original Cartesian sampled data set with 2,965,000 samples reconstructed with the tricubic B-spline. (b) Undersampled on the Cartesian lattice with 763,848 samples reconstructed with the tricubic B-spline. The surface shows an artifact. (c) Undersampled on the BCC lattice with 741,393 samples reconstructed with the quintic box spline.

BCC and trilinear B-spline interpolation on the Cartesian lattice are demonstrated in Fig. 16.

We have also approximated the mean square error existing in the volumes subsampled on BCC and Cartesian lattices. The error calculation was carried out by a random sampling of the error and summing over these random points to approximate the L_2 error integral. These experiments also confirmed that BCC subsampling is more accurate than the comparable Cartesian subsampling since the error of the Cartesian subsampled volume matched that of the BCC volume with only about 70 percent of the number of samples. Further, we have examined the visual quality of the rendered images and found empirical evidence that a BCC sampled volume with roughly about 70 percent of the number of samples of a Cartesian volume produces equivalent visual quality [23].

Computational cost. The computational cost of the reconstruction is mainly due to computing the convolution of the data values and the continuous-domain box-spline kernel. For trilinear and tricubic B-spline reconstructions on the Cartesian lattice, a neighborhood of $2 \times 2 \times 2 = 8$ and $4 \times 4 \times 4 = 64$ points fall inside the support of the kernels, respectively. Therefore, eight terms of the convolution in the case of the trilinear and 64 terms in the case of the tricubic B-spline need to be computed. Computing the convolution weights involves evaluating a *third*-degree trivariate polynomial for the trilinear, whereas a *ninth*-degree trivariate polynomial needs to be evaluated for the tricubic B-spline. However, due to the tensor-product structure of these kernels, the third-degree polynomial, in the case of the trilinear interpolation, factors into a product of three first-degree univariate polynomials. Similarly, the ninth degree trivariate polynomial of the cubic B-spline factors into a product of three third-degree univariate polynomials.

For linear and quintic box-spline reconstructions on the BCC lattice, a neighborhood of 4 and 32 points fall inside the support of the kernels, respectively. Therefore, only four terms of the convolution in the case of linear and 32 terms in the case of the quintic box spline need to be computed. Computing the convolution weights involves evaluating a *first*-degree trivariate polynomial for the linear box spline, whereas a *fifth*-degree trivariate polynomial needs to be evaluated for the quintic box spline. However, due to the structure of the quintic kernel, the fifth-degree polynomial is factored into the product of a third-degree polynomial and a second-degree polynomial, as (27) can be factored in terms of the z variable. All of the polynomial pieces of the quintic box spline listed in Section 5.5.4 are in the form of this building-block function.

Our experiments also support these predictions as the Cartesian data set in Fig. 13a was rendered in 122.69 seconds, whereas the BCC data set in Fig. 13b was rendered in 63.75 seconds. These images were computed at a resolution of 500×500 on a dual processor (Dual-Core AMD Opteron 280) machine running Linux with a GCC compiler (4.0.2). A similar rendition using trilinear interpolation on the Cartesian image took 21.49 seconds, whereas the linear box spline on the BCC took 11.99 seconds. Similar timing differences were observed on the real-life data sets; the timings for these reconstructions are summarized in Table 2. We note that for C^0 reconstructions, the speedups are less than a factor of two. Since linear interpolation is relatively light, a smaller portion of the rendering time is consumed by the reconstruction step; hence, twice a speedup in reconstruction plays a slightly less significant role in the rendering time.

7 CONCLUSION

In this paper, for the first time, we have demonstrated the practicality of box splines, as well as BCC lattice sampling. This is a significant contribution in two areas. First, it takes the mathematically elegant construct of box splines and shows that they are practical and can be efficiently implemented. Although the derivation in this paper is for a specific class of box splines, we believe that its principle can be extended to general box splines. Second, although BCC lattice sampling has been known to be theoretically superior over (standard) Cartesian sampling, it has not received the attention it deserves from practitioners— mainly due to the absence of computational tools. This
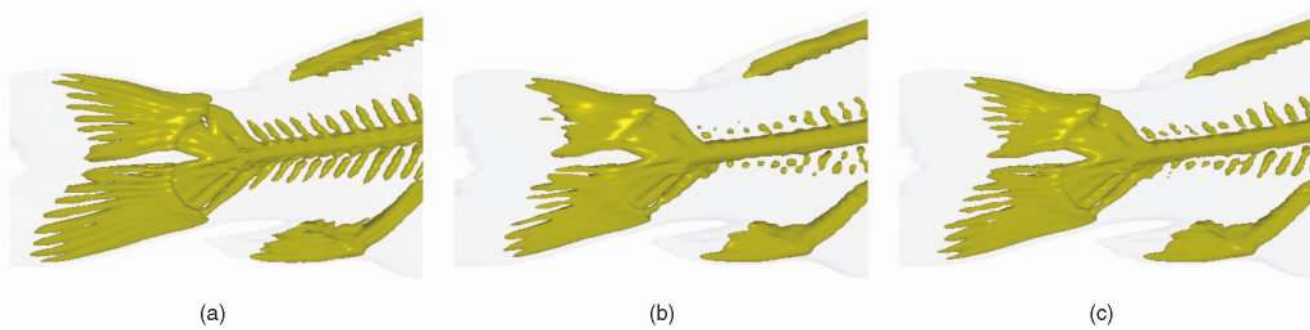
Fig. 15. The carp fish data set. (a) The original Cartesian sampled data set with 16,777,216 samples reconstructed with the tricubic B-spline. (b) Undersampled on the Cartesian lattice with 2,744,000 samples reconstructed with the tricubic B-spline. (c) Undersampled on the BCC lattice with 2,735,262 samples reconstructed with the quintic box spline.

paper makes a major step in bringing the theoretical advantages of the BCC lattice to a solid and efficient practical foundation and should pave the way to a mainstream adoption of alternative sampling structures. We plan to investigate higher degree box splines for efficient prefiltering (interpolation and quasi-interpolation); see [5] for some recent advances on the hexagonal lattice.

ACKNOWLEDGMENTS

Fig. 16. Trilinear B-spline versus linear box-spline reconstructions. (a) and (b) The C^0 reconstruction of the volumes in Fig. 15. (c) and (d) The C^0 reconstruction of the volumes in Fig. 14. (a) Cartesian. (b) BCC. (c) Cartesian. (d) BCC.

REFERENCES

[1] R.N. Bracewell, *The Fourier Transform and Its Applications,* McGraw-Hill Series in Electrical Eng. Circuits and Systems, third ed. McGraw-Hill, 1986.
[2] G. Burns, *Solid State Physics.* Academic Press, 1985.
[3] I. Carlbom, "Optimal Filter Design for Volume Reconstruction and Visualization," *Proc. IEEE Conf. Visualization (VIS '93),* pp. 54-61, Oct. 1993.
[4] H. Carr, T. Möller, and J. Snoeyink, "Artifacts Caused by Simplicial Subdivision," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 2, pp. 231-242, Mar./Apr. 2006.
[5] L. Condat and D. Van De Ville, "Quasi-Interpolating Spline Models for Hexagonally-Sampled Data," *IEEE Trans. Image Processing,* vol. 16, no. 5, pp. 1195-1206, May 2007.
[6] J.H Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups,* third ed. Springer, 1999.
[7] B. Csébfalvi, "Prefiltered Gaussian Reconstruction for High-Quality Rendering of Volumetric Data Sampled on a Body-Centered Cubic Grid," *Proc. IEEE Visualization Conf. (VIS '05),* pp. 311-318, 2005.
[8] M. Dæhlen, "On the Evaluation of Box Splines," *Mathematical Methods in Computer Aided Geometric Design,* pp. 167-179, 1989.
[9] C. de Boor, K. Höllig, and S. Riemenschneider, "Box Splines," *Applied Mathematical Sciences,* vol. 98, Springer, 1993.
[10] C. de Boor, "On the Evaluation of Box Splines," *Numerical Algorithms,* vol. 5, nos. 1-4, pp. 5-23, 1993.
[11] D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing,* first ed. Prentice Hall, 1984.
[12] S.C. Dutta Roy and B. Kumar, "Digital Differentiators," *Handbook of Statistics,* vol. 10, Elsevier, pp. 159-205, 1993.
[13] A. Entezari, R. Dyer, and T. Möller, "Linear and Cubic Box Splines for the Body Centered Cubic Lattice," *Proc. IEEE Visualization Conf. (VIS '04),* pp. 11-18, Oct. 2004.
[14] A. Entezari, T. Meng, S. Bergner, and T. Möller, "A Granular Three Dimensional Multiresolution Transform," *Proc. Eurographics/IEEE-VGTC Symp. Visualization (EuroVis '06),* pp. 267-274, May 2006.
[15] T.C. Hales, "Cannonballs and Honeycombs," *Notices of the AMS,* vol. 47, no. 4, pp. 440-449, Apr. 2000.
[16] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 29, no. 6, pp. 1153-1160, Dec. 1981.
[17] L. Kobbelt, "Stable Evaluation of Box Splines," *Numerical Algorithms,* vol. 14, no. 4, pp. 377-382, 1997.
[18] H.R. Künsch, E. Agrell, and F.A. Hamprecht, "Optimal Lattices for Sampling," *IEEE Trans. Information Theory,* vol. 51, no. 2, pp. 634-647, Feb. 2005.
[19] T. Theußl, O. Mattausch, T. Möller, and E. Gröller, "Reconstruction Schemes for High Quality Raycasting of the Body-Centered Cubic Grid," Technical Report TR-186-2-02-11, Inst. Computer Graphics and Algorithms, Vienna Univ. of Technology, Dec. 2002.
[20] T. Theußl, T. Möller, and E. Gröller, "Optimal Regular Volume Sampling," *Proc. IEEE Visualization Conf. (VIS '01),* pp. 91-98, Oct. 2001.
[21] S.R. Marschner and R.J. Lobb, "An Evaluation of Reconstruction Filters for Volume Rendering," *Proc. IEEE Conf. Visualization (VIS '94),* pp. 100-107, 1994.
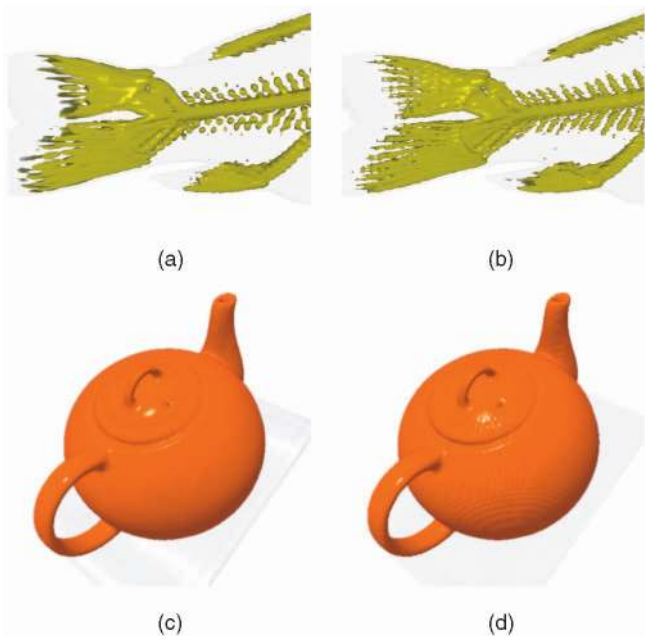
[22] J.H. McClellan, "The Design of Two-Dimensional Digital Filters by Transformations," *Proc. Seventh Ann. Princeton Conf. Information Sciences and Systems,* pp. 247-251, 1973.

[23] 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," *Proc. 33rd Graphics Interface Conf. (GI '07),* pp. 265-272, May 2007.

[24] R.M. Mersereau, "The Processing of Hexagonally Sampled Two-Dimensional Signals," *Proc. IEEE,* vol. 67, no. 6, pp. 930-949, June 1979.

[25] D.P. Mitchell and A.N. Netravali, "Reconstruction Filters in Computer Graphics," *Computer Graphics (Proc. ACM SIGGRAPH '88),* vol. 22, pp. 221-228, Aug. 1988.

[26] T. Möller, R. Machiraju, K. Mueller, and R. Yagel, "A Comparison of Normal Estimation Schemes," *Proc. IEEE Visualization Conf. (VIS '97),* pp. 19-26, Oct. 1997.

[27] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel, "Design of Accurate and Smooth Filters for Function and Derivative Reconstruction," *Proc. IEEE Symp. Volume Visualization (VVS '98),* pp. 143-151, Oct. 1998.

[28] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[29] D.P. Petersen and D. Middleton, "Sampling and Reconstruction of Wave-Number-Limited Functions in N-Dimensional Euclidean Spaces," *Information and Control,* vol. 5, no. 4, pp. 279-323, Dec. 1962.

[30] G. Strang and G.J. Fix, "A Fourier Analysis of the Finite Element Variational Method," *Constructive Aspects of Functional Analysis,* pp. 796-830, 1971.

[31] P. Thévenaz, T. Blu, and M. Unser, "Interpolation Revisited," *IEEE Trans. Medical Imaging,* vol. 19, no. 7, pp. 739-758, July 2000.

[32] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle, "Hex-Splines: A Novel Spline Family for Hexagonal Lattices," *IEEE Trans. Image Processing,* vol. 13, no. 6, pp. 758-772, June 2004.

[33] Wikipedia, *Minkowski Addition,* http://en.wikipedia.org/wiki/Minkowski_addition, 2007, June 2007.

Alireza Entezari received the BSc degree from Simon Fraser University in 2001. He started his undergraduate studies at Sharif University of Technology, Tehran, Iran, but in 1998, he transferred to Simon Fraser University. He is a PhD candidate in the School of Computing Science, Simon Fraser University. His research focus is currently on spline interpolation and approximation of trivariate functions on regular sampling lattices.

Dimitri Van De Ville (M'02) received the MS degree in engineering and computer sciences and the PhD degree from Ghent University, Ghent, Belgium, in 1998 and 2002, respectively. He obtained a grant as a Research Assistant with the Fund for Scientific Research Flanders Belgium (FWO). In 2002, he joined Professor M. Unser's Biomedical Imaging Group at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. In December 2005, he became responsible for the Signal Processing Antenna at the University Hospital of Geneva, Geneva, as part of the Centre d'Imagerie BioMédicale (CIBM). His research interests include wavelets, statistical analysis, multidimensional splines, and applications in biomedical imaging, such as functional magnetic resonance imaging, spectroscopy, electro-encephalography, and microscopy. He has been an associate editor of the *IEEE Transactions on Image Processing* since February 2006 and was previously an associate editor of *IEEE Signal Processing Letters* from 2004 to 2006. Since 2003, he has also been an editor and webmaster of *The Wavelet Digest*. He was a cochair of the Wavelets XII International Conference in August 2007, together with V. Goyal and M. Papadakis. He is a member of the IEEE.

Torsten Möller received the Vordiplom (BSc) degree in mathematical computer science from Humboldt University of Berlin, Germany, and the PhD degree in computer and information science from Ohio State University in 1999. He is an associate professor in the School of Computing Science, Simon Fraser University. He is a codirector of the Graphics, Usability, and Visualization Laboratory and serves on the board of advisors of the Centre for Scientific Computing, Simon Fraser University. His research interests include the fields of visualization and computer graphics, especially the mathematical foundations thereof. His main focus is currently on tools for optimal sampling lattices (acquisition, interpolation, and multiresolution), as well as the exploration of high-dimensional continuous data. The main application areas are functional medical imaging, as well as simulation tools for computational science. He has been appointed as the vice chair for publications of the IEEE Technical Committee of Visualization and Graphics. He is currently on sabbatical at the Ecole Polytechnique Fédérale de Lausanne. He is a member of the IEEE, ACM, Eurographics, and CIPS. He has served on a number of program committees (including Eurographics and IEEE Visualization Conferences) and has been a paper cochair of the 2003 Graphics Interface (GI) Conference, as well as for the 2006 Workshop on Volume Graphics (VG '06). He is currently serving as the general chair for the 2007 Symposium on Volume Graphics (VG '07), as well as the paper cochair for the 2007 and 2008 EuroVis Conferences.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.