Quartic Box-Spline Reconstruction on the BCC Lattice

Minho Kim

Abstract—This paper presents an alternative box-spline filter for the body-centered cubic (BCC) lattice, the 7-direction quartic box-spline M_7 that has the same approximation order as the 8-direction quintic box-spline M_8 but a lower polynomial degree, smaller support, and is computationally more efficient. When applied to reconstruction with quasi-interpolation prefilters, M_7 shows less aliasing, which is verified quantitatively by integral filter metrics and frequency error kernels. To visualize and analyze distributional aliasing characteristics, each spectrum is evaluated on the planes and lines with various orientations.

Index Terms—Volume reconstruction, BCC lattice, box-spline, quasi-interpolation.

1 Introduction

In multivariate signal processing, not only reconstruction filters, but also sampling lattices play an important role in the quality of reconstruction. While computationally more beneficial, the widely used Cartesian lattice is an inefficient sampling lattice compared to other regular sampling lattices [22].

In particular, for three-dimensional volume dataset reconstruction, the BCC lattice, the dual of the face-centered cubic (FCC) lattice, is the optimal sampling lattice; *i.e.*, it requires the fewest number of samples to recover an isotropic band-limited signal without aliasing. Among the BCC reconstruction filters, the 8-direction quintic box-spline M_8 is the most popular due to its approximation power and computational efficiency [10], [13], [14].

In this paper, we present an alternative symmetric box-spline filter M_7 on the BCC lattice that is of lower degree but has better approximation power than M_8. Moreover, its computational cost to evaluate reconstructed splines is lower. The contributions of this paper are

- in-depth investigation of M_7 including its spline structure, stencil (samples on the lattice required to evaluate a spline) and the BB (Bernstein-Bézier)-form;
- an efficient evaluation algorithm of splines; and
- detailed comparison and analysis of three reconstruction filters in view of integral filter metrics, frequency error kernels, and spectra visualization.

2 Previous Works

Petersen and Middleton [22] extended Shannon's theorem to the multi-dimensional case and showed that the optimal sampling lattice is the dual of the densest sphere packing lattice for a band-limited and isotropic input signal. In dimension three, the FCC lattice is the densest sphere packing lattice [4]; therefore its dual, the BCC lattice, is the optimal sampling lattice.

Based on this observation, many researchers have investigated volume reconstruction on the BCC lattice using various filters. One of the most efficient filters is the quintic box-spline proposed by Entezari *et al.* [10]. Not only does it show superior reconstruction quality compared to the comparable tensor-product B-spline filter on the Cartesian lattice, but its computational cost is small enough for real-time rendering on the GPU (Graphics Processing Unit) thanks to the (partially) factored power form [13]. Csébfalvi and Hadwiger [7] proposed the tri-cubic B-spline filter M_{12} for reconstruction on the BCC lattice, which has the same approximation order as M_8. While it is analytically disadvantageous due to its higher polynomial degree and larger support, it shows superb performance for real-time volume rendering on the GPU thanks to its hardware-friendly tensored structure. Generalizing the hex-splines to arbitrary lattices, Mirzargar and Entezari [19] recently proposed Voronoi splines that can be used as reconstruction filters on both the FCC and BCC lattices. While interesting, they are not practical due to the lack of an efficient evaluation algorithm. As a box-spline reconstruction filter on the FCC lattice, Kim *et al.* [15] proposed the symmetric cubic box-spline filter which has an approximation order of three but requires only 16 samples for evaluation. The 7-direction box-spline on the Cartesian lattice, which shares many properties with M_7, was first proposed by Peters [21] and later revisited by Entezari and Möller [12]. The basic properties of the quartic box-spline M_7 were investigated by Kim and Peters [17] but were limited to the approximation order, linear independence of its shifts, and the optimal quasi-interpolant.

Entezari *et al.* [11] derived a discrete quasi-interpolant for M_8 that provides the maximal approximation order [8]. Csébfalvi [5] also derived various discrete quasi-interpolants for both M_{12} and M_8 and evaluated their

• *M. Kim is with the School of Computer Science, University of Seoul, Seoul, South Korea.*
E-mail: minhokim@uos.ac.kr

reconstruction results.

Marschner and Lobb [18] proposed numerical filter metrics for several types of aliasing, methods which are useful to measure integral characteristics of filters. Blu and Unser [1] proposed a frequency error kernel that is useful to measure a sharp L_2 error according to sampling frequency. Csébfalvi and Domonkos [6] relied on isosurface and direct rendering of spectra to visualize and analyze distributional aliasing characteristics of three-dimensional filters. While their approach provides useful insight, it is still challenging to visualize the detailed distribution of the spectra since they visualize either only one level set (isosurface rendering) or integral measures (direct rendering).

Condat *et al.* [3] derived the explicit polynomial formula of the three-direction box-spline using inverse Fourier transform followed by finite differences. Entezari *et al.* [13] derived the explicit polynomial formula of M_8 in the same way. Kim and Peters [16] proposed a fast and stable algorithm to evaluate box-splines in general, leveraging pre-computed and tabulated rational BB-coefficients.

3 Background

In this section, we first characterize the BCC lattice and review box-splines, including two box-splines on the BCC lattice. Finally we introduce the frequency error kernel devised by Blu and Unser [1].

3.1 The BCC Lattice

An *n-dimensional lattice* $\mathcal{L}_n$ is the set of points generated by an $n \times n$ (invertible) square *generator matrix* as all integer linear combinations of its column vectors:

$$\mathcal{L}_n := \mathbf{G}\mathbb{Z}^n = \{\mathbf{G}\boldsymbol{j} : \mathbf{G} \in \mathbb{R}^{n\times n}, \operatorname{rank}\mathbf{G} = n, \boldsymbol{j} \in \mathbb{Z}^n\}.$$

Its *dual lattice* $\mathcal{L}_n^*$ is defined as

$$\mathcal{L}_n^* := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}\cdot\boldsymbol{u} \in \mathbb{Z}, \forall \boldsymbol{u} \in \mathcal{L}_n\},$$

and its generator matrix is $\mathbf{G}^{-t}$ [4]. In multi-dimensional signal processing, sampling a signal on a lattice $\mathcal{L}_n$ is equivalent to replicating its spectrum on the dual lattice $\mathcal{L}_n^*$ in the frequency domain [9].

The BCC lattice can be considered to be generated either [4]

(i) as integer linear combinations of any three of the four vectors from the center of a regular tetrahedron to its vertices or
(ii) by inserting additional lattice points at the center of each Cartesian grid cell.

Formally, we can define the BCC lattice with generator matrix $\mathbf{G}_{\text{bcc}}$ as

$$\mathbb{Z}_{\text{bcc}} := \mathbf{G}_{\text{bcc}}\mathbb{Z}^3, \quad \text{where } \mathbf{G}_{\text{bcc}} := \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}.$$

Note that $\det \mathbf{G}_{\text{bcc}} = 4$. The BCC lattice is the dual of the FCC lattice, the optimal 3D sphere packing lattice, and hence is the optimal 3D sampling lattice [22].

We can group the lattice neighbors of a point into *shells* [4] according to their distances from that point. Let $\mathcal{S}_k$ be the k-th shell of the origin. For $\mathbb{Z}_{\text{bcc}}$, the Cartesian coordinates of the first three shells are as follows:

$$\mathcal{S}_0 = \{(0,0,0)\},$$
$$\mathcal{S}_1 = \{(\pm1,\pm1,\pm1)\}, \text{ and} \tag{1}$$
$$\mathcal{S}_2 = \{(\pm2,0,0),(0,\pm2,0),(0,0,\pm2)\}, \tag{2}$$

with squared distances 0, 3 and 4, respectively. Considering non-parallel vectors corresponding to $\mathcal{S}_1$ and $\mathcal{S}_2$, we get two sets of vectors,

$$\boldsymbol{\Xi}_1 := \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \quad \text{and}$$
$$\boldsymbol{\Xi}_2 := \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

which are the building blocks of the three box-spline filters we consider.

3.2 Box-Splines

In this section, we review the basics of box-splines. For more details, refer to de Boor *et al.* [8]. In the following, a matrix also denotes a (multi)-set of column vectors allowing multiplicity, depending on the context.

3.2.1 Definition and basic properties

A box-spline is a piecewise polynomial with finite support and certain continuity and is uniquely defined by a *direction matrix*. Given an $n\times m$ ($n \le m$) direction matrix $\boldsymbol{\Xi}$, a box-spline $M_{\boldsymbol{\Xi}}$ can be constructed by performing consecutive directional convolutions in each column direction. In other words, starting from the base case ($n = m$),

$$M_{\boldsymbol{\Xi}}(\boldsymbol{x}) := \frac{1}{|\det \boldsymbol{\Xi}|}\chi_{\boldsymbol{\Xi}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n$$

where $\boldsymbol{\Xi}$ is invertible, and $\chi_{\boldsymbol{\Xi}}(\boldsymbol{x})$ is the characteristic function on the half-open parallelepiped $\boldsymbol{\Xi}[0,1)^m$ (Figure 1(a)), namely

$$\chi_{\boldsymbol{\Xi}}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \boldsymbol{\Xi}[0,1)^m \\ 0 & \text{otherwise,} \end{cases}$$

a box-spline can be recursively defined as (Figure 1)

$$M_{\boldsymbol{\Xi}\cup\{\boldsymbol{\xi}\}}(\boldsymbol{x}) := \int_0^1 M_{\boldsymbol{\Xi}}(\boldsymbol{x} - t\boldsymbol{\xi})dt, \quad \boldsymbol{\xi} \in \mathbb{R}^n.$$

Since the direction matrix $\boldsymbol{\Xi}$ uniquely defines the box-spline $M_{\boldsymbol{\Xi}}$, all the properties of box-splines can be deduced from $\boldsymbol{\Xi}$:

(i) the (total) polynomial degree of $M_{\boldsymbol{\Xi}}$ is $m - n$,

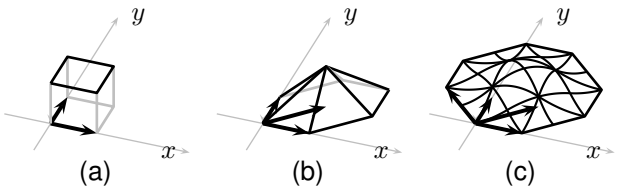


Fig. 1. Construction of the box-splines with direction matrices (a) $\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$, (b) $\left[\begin{smallmatrix}1&0&1\\0&1&1\end{smallmatrix}\right]$ and (c) $\left[\begin{smallmatrix}1&0&1&-1\\0&1&1&1\end{smallmatrix}\right]$ via consecutive directional convolutions.

(ii) the support of M_{Ξ} is defined as the Minkowski sum of the directions of Ξ (Figure 1 and Figure 3), and
(iii) the polynomial pieces in M_{Ξ} join with $C^{\rho(\Xi)-2}$ continuity, where $\rho(\Xi)$ is the minimum number of (column) directions of Ξ such that, when they are removed from Ξ, the remaining columns in Ξ do not span $\mathbb{R}^n$:

$$\rho(\Xi) := \{\min_{\mathbf{Z}\subseteq\Xi} \#\mathbf{Z} : \operatorname{rank}(\Xi\backslash\mathbf{Z}) < n\}. \quad (3)$$

In most cases, the *centered* box-spline

$$M^c_{\Xi}(\boldsymbol{x}) := M_{\Xi}(\boldsymbol{x} + \sum_{\boldsymbol{\xi}\in\Xi} \boldsymbol{\xi}/2)$$

is used for reconstruction to prevent splines from shifting.

3.2.2 Reconstruction with Box-Spline Filters

In signal processing, reconstruction is the process to recover the original signal from its discrete samples via convolution with a reconstruction filter. Given data samples

$$V : \mathbf{G}\mathbb{Z}^n \to \mathbb{R}$$

on the lattice $\mathbf{G}\mathbb{Z}^n$, we can reconstruct a continuous spline using box-spline filter M_{Ξ} as follows:

$$\sum_{\boldsymbol{j}\in\mathbf{G}\mathbb{Z}^n} V(\boldsymbol{j}) M_{\Xi}(\boldsymbol{x}-\boldsymbol{j}). \quad (4)$$

To interpolate the samples, the shifts of a reconstruction filter should be linearly independent to form a Riesz basis. This can be easily verified for box-splines because the shifts of M_{Ξ} on $\mathbf{G}\mathbb{Z}^n$,

$$(M_{\Xi}(\cdot - \boldsymbol{j}))_{\boldsymbol{j}\in\mathbf{G}\mathbb{Z}^n},$$

are linearly independent and form a Riesz basis if and only if [8]

$$|\det \mathbf{Z}| \in \{0, |\det \mathbf{G}|\}, \quad \forall \mathbf{Z} \subseteq \Xi \text{ and } \mathbf{Z} \text{ is square},$$

since this ensures that the shifts of M_{Ξ} "minimally overlap" one another.

Let $s(\boldsymbol{x})$ be an n-variate continuous signal. When $s(\boldsymbol{x})$ is sampled on the lattice $h\mathbf{G}\mathbb{Z}^n$, $h > 0$, we can reconstruct a spline $s_h(\boldsymbol{x})$ as follows:

$$s_h(\boldsymbol{x}) := \sum_{\boldsymbol{j}\in h\mathbf{G}\mathbb{Z}^n} s(\boldsymbol{j}) M_{\Xi}(\boldsymbol{x}-\boldsymbol{j}).$$

Then the *approximation order* L, the rate of decay of the error as $h \to 0$, is defined as follows:

$$\|s(\boldsymbol{x}) - s_h(\boldsymbol{x})\|_{L_2} := \int_{\mathbb{R}^n} |s(\boldsymbol{x}) - s_h(\boldsymbol{x})|^2 d\boldsymbol{x} = O(h^L).$$

While interpolation results in no error on lattice points, in general it shows inferior L_2 error. In most cases, we can obtain better reconstruction by applying a *discrete quasi-interpolation prefilter* q on the discrete dataset before convolution:

$$\sum_{\boldsymbol{j}\in\mathbf{G}\mathbb{Z}^n} (V \star q)(\boldsymbol{j}) M_{\Xi}(\boldsymbol{x}-\boldsymbol{j})$$

where $\star$ denotes the *discrete convolution*

$$(V \star q)(\boldsymbol{j}) := \sum_{\boldsymbol{k}\in\mathbf{G}\mathbb{Z}^n} V(\boldsymbol{k}) q(\boldsymbol{j}-\boldsymbol{k}).$$

Then the reconstructed spline annihilates all the lower terms of the Taylor expansion of the input signal, reducing the approximation error. The optimal approximation order of M_{Ξ} is $\rho(\Xi)$ (3), and such a prefilter can be found by following the procedures of de Boor *et al.* [8] or Blu and Unser [1]. See Table 1 for examples of discrete quasi-interpolant prefilters.

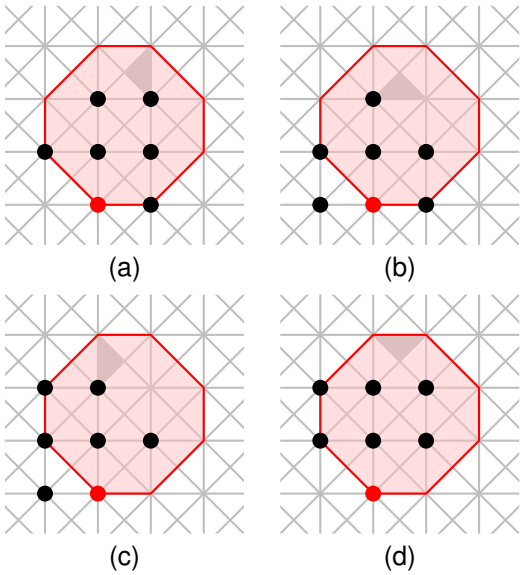


Fig. 2. Stencils (dots) of four shift-invariant polynomial pieces (gray triangles) of a spline generated by the box-spline defined by $\left[\begin{smallmatrix}1&0&1&-1\\0&1&1&1\end{smallmatrix}\right]$, with stencil size 7. Support of one shift (red dot and pink octagon) is shown as an illustration.

The polynomial pieces of a spline are delineated by the shifts of the *knot planes* spanned by the directions of Ξ (Figure 1 and Figure 2). Therefore, it is important to determine the polynomial structure in order to evaluate a spline. Since box-splines have finite support, for evaluation of a spline at $\boldsymbol{x} \in \mathbb{R}^n$, a finite number of shifts M_{Ξ} are required. The set of such shifts is called a *stencil* and is unique for each shift-invariant polynomial piece

(Figure 2). The stencil size is the same as

$$\mathrm{vol}\left(\mathrm{supp}\left(M_{\boldsymbol{\Xi}}\right)\right)/|\det \mathbf{G}|,$$

the (normalized) volume of the support of $M_{\boldsymbol{\Xi}}$. Note that a large stencil size results in slow evaluation due to required fetch operations.

3.3 Symmetric Quintic Box-Spline on the BCC Lattice

The symmetric quintic box-spline M_8 on the BCC lattice was first proposed by Entezari *et al.* [10] as an extension of the symmetric quartic box-spline on the hexagonal lattice. M_8 is defined by (Table 1)

$$\boldsymbol{\Xi}_8 := \begin{bmatrix} \boldsymbol{\Xi}_1 & \boldsymbol{\Xi}_1 \end{bmatrix},$$

i.e., the four directions corresponding to the eight lattice points in $\mathcal{S}_1$ (1), as shown in Figure 3(a). Its support is the shape of a rhombic dodecahedron (Figure 3(b) and Figure 4(a)). The volume of the support of M_8 is 128, 24 times the volume of the tetrahedron composed of $(0,0,0)$, $(4,0,0)$, $(0,4,0)$ and $(2,2,2)$; therefore, the stencil size is $128/|\det \mathbf{G}_{\text{bcc}}| = 32$.

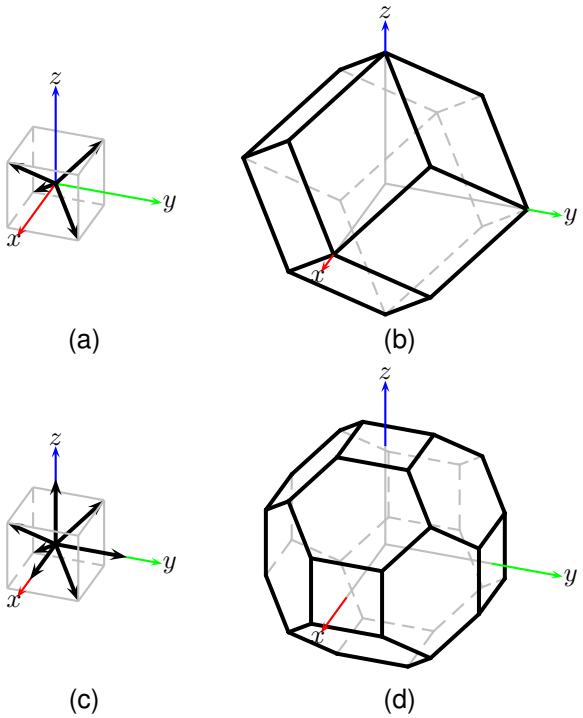


Fig. 3. The (left) directions and (right) supports of (upper) M_8 and (lower) M_7.

The (total) polynomial degree of M_8 is $8-3=5$, and hence is quintic. Since any three directions in $\boldsymbol{\Xi}_1$ span $\mathbb{R}^3$, $\rho(\boldsymbol{\Xi}_1)=2$; therefore, $\rho(\boldsymbol{\Xi}_8)=4$ is the approximation order of the quintic spline. There are six types of shift-invariant knot planes defined by $\boldsymbol{\Xi}_8$, which decompose a spline into six shift-invariant tetrahedral polynomial pieces, as in Figure 5. This decomposition coincides with the Delaunay tetrahedralization of the BCC lattice by

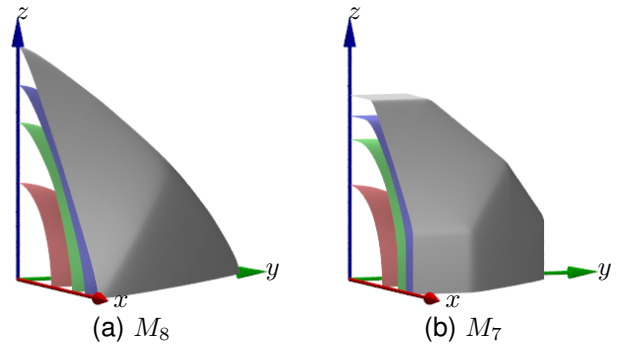


Fig. 4. Isosurfaces of two box-splines with levels 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-5} in the first octant.

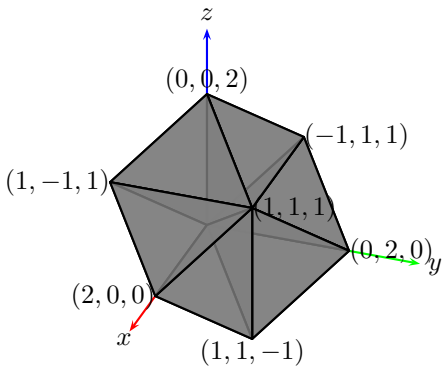


Fig. 5. Six shift-invariant (semi-regular) tetrahedra delineating a spline generated by M_8.

one type of (semi-regular) tetrahedron [24]. Therefore, only one of the six shift-invariant tetrahedral polynomial pieces needs to be considered to evaluate a spline [13], and others can be evaluated after suitable transformation. It is easy to verify that the shifts of $M_8(\boldsymbol{x})$ on $\mathbb{Z}_{\text{bcc}}$ form a Riesz basis [17]. Two quasi-interpolation prefilters are shown in Table 1, one of which, Q_8^{I}, was derived by Entezari *et al.* [11].

3.4 Tri-Cubic B-Spline on the BCC Lattice

The tri-cubic B-spline M_{12} on the BCC lattice [7] is defined by the 12 directions

$$\boldsymbol{\Xi}_{12} := \begin{bmatrix} \boldsymbol{\Xi}_2 & \boldsymbol{\Xi}_2 & -\boldsymbol{\Xi}_2 & -\boldsymbol{\Xi}_2 \end{bmatrix},$$

i.e., the three directions corresponding to the lattice points in $\mathcal{S}_2$ (2), each with multiplicity four. Its total polynomial degree is $12-3=9$, and the approximation order is four since $\rho(\boldsymbol{\Xi}_2)=1$; hence, $\rho(\boldsymbol{\Xi}_{12})=4$. In addition, $\mathrm{vol}(\mathrm{supp}\, M_{12}) = 8^3 = 512$; therefore, its stencil size is $512/|\det \mathbf{G}_{\text{bcc}}| = 128$. The shifts of M_{12} do not form a Riesz basis since $|\det \boldsymbol{\Xi}_2| = 8$. Note that tensor-product B-splines with uniform knots are special cases of box-splines.

3.5 Frequency Error Kernels

Blu and Unser [1] proposed a frequency error kernel formula that quantifies the average L_2 error of a reconstruction filter according to the sampling frequency. We review it briefly here. In the following, $\widehat{f}(\boldsymbol{\omega})$ and $f^*(\boldsymbol{x})$ denote the Fourier transform and complex conjugate of $f(\boldsymbol{x})$, respectively.

Given a shift-invariant reconstruction filter ϕ, its frequency error kernel is defined as [1]

$$E(\boldsymbol{\omega}) := \underbrace{1 - \frac{|\widehat{\phi}(\boldsymbol{\omega})|^2}{\widehat{a_\phi}(\boldsymbol{\omega})}}_{E_{\min}(\boldsymbol{\omega})} + \underbrace{\widehat{a_\phi}(\boldsymbol{\omega}) \left| Q(e^{j\boldsymbol{\omega}}) - \frac{\widehat{\phi^*}(\boldsymbol{\omega})}{\widehat{a_\phi}(\boldsymbol{\omega})} \right|^2}_{E_{\text{res}}(\boldsymbol{\omega})}, \quad (5)$$

where a_ϕ is the discrete *autocorrelation* function of ϕ defined as

$$a_\phi(\boldsymbol{k}) := \int_{\mathbb{R}^n} \phi^*(\boldsymbol{x} - \boldsymbol{k})\phi(\boldsymbol{x})d\boldsymbol{x}, \quad \boldsymbol{k} \in \mathbb{Z}^n,$$

$Q(e^{j\boldsymbol{\omega}})$ is the discrete time Fourier transform of the prefilter $q(\boldsymbol{k})$, and $E_{\min}(\boldsymbol{\omega})$ denotes the minimum error that can be obtained when the prefilter Q is the orthogonal projector. In general, the most difficult part in computing $E(\boldsymbol{\omega})$ is to compute $\widehat{a_\phi}(\boldsymbol{\omega})$. However, for real-valued symmetric box-spline filter $M^c_\Xi(\boldsymbol{x})$, $a_\phi(\boldsymbol{k})$ reduces to

$$\int_{\mathbb{R}^n} M^c_\Xi(\boldsymbol{k} - \boldsymbol{x})M^c_\Xi(\boldsymbol{x})d\boldsymbol{x} = (M^c_\Xi * M^c_\Xi)(\boldsymbol{k}) = M^c_{\Xi\cup\Xi}(\boldsymbol{k}).$$

In other words, we only need to evaluate the values of the box-spline $M^c_{\Xi\cup\Xi}(\boldsymbol{x})$ on the lattice points $\mathbf{G}_{\text{bcc}}\mathbb{Z}^3$, which can be achieved by solving an eigenvalue problem based on the refinement property of the box-spline [2].

4 Symmetric Quartic Box-Spline on the BCC Lattice

In this section, we define M_7 on the BCC lattice, investigate its properties, and establish an efficient spline evaluation algorithm.

4.1 Definition and Properties

The symmetric quartic box-spline M_7 on the BCC lattice $\mathbb{Z}_{\text{bcc}}$ is the 'centered' and 'scaled' version of the box-spline M_{Ξ_7}

$$M_7(\boldsymbol{x}) := |\det \mathbf{G}_{\text{bcc}}| M_{\Xi_7}(\boldsymbol{x} + \sum_{\boldsymbol{\xi}\in\Xi_7} \frac{\boldsymbol{\xi}}{2}) = 4M_{\Xi_7}\left(\boldsymbol{x} + \begin{bmatrix}1\\1\\1\end{bmatrix}\right)$$

defined by the direction matrix

$$\Xi_7 := \begin{bmatrix}\Xi_1 & \Xi_2\end{bmatrix}$$

corresponding to the $8 + 6 = 14$ lattice points in $\mathcal{S}_1 \cup \mathcal{S}_2$. The support of M_7 is the shape of a truncated rhombic dodecahedron, obtained by cutting six corners of the support of M_8 (Figure 3(d) and Figure 4(b)). The volume of the support is 120, since the cut-away volume is $2^3 = 8$; therefore, the stencil size is $120/|\det \mathbf{G}_{\text{bcc}}| = 30$, two less than that of M_8. The degree of M_7 is $7 - 3 = 4$,

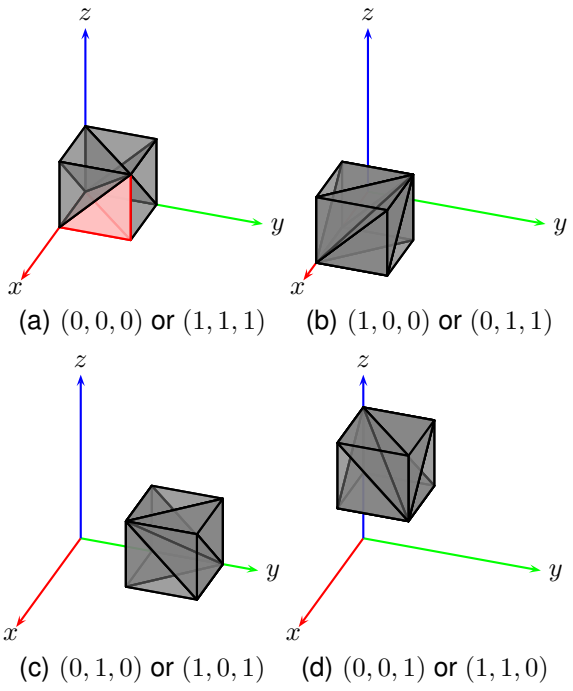


Fig. 6. The four types of partitioned cubes delineating the polynomial pieces of M_7. The coordinate triples denote the 'lower corner' of each cube (with modulo 2). Each cube is decomposed into six tetrahedra aligned along its respective diagonal, resulting in 24 shift-invariant tetrahedral polynomial pieces.

and it has the approximation order four [17]. The shifts of $M_7(\boldsymbol{x})$ on $\mathbb{Z}_{\text{bcc}}$ are linearly dependent, therefore do not form a Riesz basis since $\det \Xi_2 = 8$, while $\det \mathbf{G}_{\text{bcc}} = 4$ [17]. This would cause a problem when interpolating the samples. However, interpolation usually results in poorer reconstruction quality than quasi-interpolation [1]. Two of the optimal quasi-interpolation prefilters are shown in Table 1.

4.2 Evaluation

To evaluate a spline generated by M_7, we need to exploit the structure induced by the shifts of the knot planes of M_7 on $\mathbb{Z}_{\text{bcc}}$. First, note that Ξ_7 induces nine knot planes with normals

$$\left\{\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}\right\}.$$

The shifts on $\mathbb{Z}_{\text{bcc}}$ of the three axis-aligned knot planes first decompose a spline into unit cubes, which can be grouped in four types according to their lower corner indices (Figure 6). Each cube is further decomposed into six tetrahedra by other knot planes, as in Figure 6, resulting in 24 shift-invariant tetrahedral polynomial pieces. Due to the symmetry of the structure, all 24 tetrahedra are congruent. Hence, for evaluation, we can consider only one *reference tetrahedron*, for which we picked that

TABLE 1
Comparison of three box-splines on the BCC lattice having an approximation order of four. Note that quasi-interpolants are z-transformed.

Box-spline	M_{12}	M_8	M_7
direction matrix	$\begin{bmatrix}\boldsymbol{\Xi}_2 & \boldsymbol{\Xi}_2 & -\boldsymbol{\Xi}_2 & -\boldsymbol{\Xi}_2\end{bmatrix}$	$\begin{bmatrix}\boldsymbol{\Xi}_1 & \boldsymbol{\Xi}_1\end{bmatrix}$	$\begin{bmatrix}\boldsymbol{\Xi}_1 & \boldsymbol{\Xi}_2\end{bmatrix}$
# of directions	12	8	7
(polynomial) degree	9	5	4
approximation order	4	4	4
support	cube	rhombic dodecahedron	truncated rhombic dodecahedron
# of shift-invariant pieces	1	6	24
volume of support	512	128	120
stencil size	128	32	30
Riesz basis?	no	yes	no
optimal quasi-interpolation prefilter	$Q_{12}^{\mathrm{I}}(\boldsymbol{z}) := \frac{7}{3} - \frac{1}{6}\sum_{\boldsymbol{\xi}\in\mathcal{S}_1} \boldsymbol{z}^{-\boldsymbol{\xi}}$	$Q_7^{\mathrm{I}}(\boldsymbol{z}) = Q_8^{\mathrm{I}}(\boldsymbol{z}) := \frac{5}{3} - \frac{1}{12}\sum_{\boldsymbol{\xi}\in\mathcal{S}_1} \boldsymbol{z}^{-\boldsymbol{\xi}}$	
	$Q_{12}^{\mathrm{II}}(\boldsymbol{z}) := \frac{13}{6} - \frac{1}{12}\sum_{\boldsymbol{\xi}\in\mathcal{S}_1\cup\mathcal{S}_2} \boldsymbol{z}^{-\boldsymbol{\xi}}$	$Q_7^{\mathrm{II}}(\boldsymbol{z}) = Q_8^{\mathrm{II}}(\boldsymbol{z}) := \frac{19}{12} - \frac{1}{24}\sum_{\boldsymbol{\xi}\in\mathcal{S}_1\cup\mathcal{S}_2} \boldsymbol{z}^{-\boldsymbol{\xi}}$	

with vertices $\{\boldsymbol{v}_j\}_{j=1}^4$ as follows (Figure 6(a)):

$$\boldsymbol{v}_1 := \begin{bmatrix}0\\0\\0\end{bmatrix}, \boldsymbol{v}_2 := \begin{bmatrix}1\\1\\1\end{bmatrix}, \boldsymbol{v}_3 := \begin{bmatrix}1\\0\\0\end{bmatrix} \text{ and } \boldsymbol{v}_4 := \begin{bmatrix}1\\1\\0\end{bmatrix}.$$

Table 1 of the supplemental material shows the stencil and BB-coefficients of the reference tetrahedron.

Algorithm 1 shows a pseudocode for evaluating the spline

$$\sum_{\boldsymbol{j}\in\mathbb{Z}_{\mathrm{bcc}}} V(\boldsymbol{j}) M_7(\boldsymbol{x}-\boldsymbol{j})$$

where $V : \mathbb{Z}_{\mathrm{bcc}} \mapsto \mathbb{R}$ is the volume dataset sampled on the BCC lattice. To determine which of the 24 tetrahedra the input $\boldsymbol{x}$ belongs, we first determine the cube type (Figure 6) using the lower point ($\lfloor \boldsymbol{x} \rfloor$ modulo 2) and find the reflection matrix $\mathbf{R}$ that maps the cube to the reference cube in Figure 6(a). Then we determine which of the six tetrahedra in Figure 6(a) contains $\mathbf{R}(\boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor)$ and determine the suitable permutation matrix $\mathbf{P}$. Now $\dot{\boldsymbol{x}} := \mathbf{PR}(\boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor)$ is located in the reference tetrahedron, and we can collect the 30 data samples and construct the polynomial formula. One method is to build a BB-form and use the de Casteljau algorithm for evaluation. Table 1 of the supplemental material shows the BB-coefficients computed by following the procedure by Kim and Peters [16]. While this method is stable and fast in general, it shows poor performance in our case due to the large number of multiplications. Therefore, we derived the explicit formula following the procedure by Entezari *et al.* [13]. Figure 1 of the supplemental material shows the explicit symbolic polynomial formula after some simplification.

5 Results and Discussion

In this section, we compare three reconstuction schemes with various quasi-interpolation prefilters. For rendering of the reconstructed volume, the POV-Ray ray-tracing package [20] is used. Note that POV-Ray uses the finite difference method internally to compute normal vectors for shading. In the following sections, the number of samples is denoted as $N^3 \times 2$ in the sense that the BCC lattice can be built by inserting additional points at the center of Cartesian grid cell. In addition, Q_*^{II} denotes all

Algorithm 1 Evaluation of $\sum_{\boldsymbol{j}\in\mathbb{Z}_{\text{bcc}}} V(\boldsymbol{j})M_7(\boldsymbol{x}-\boldsymbol{j})$

function EVALUATEQUARTIC(V, $\boldsymbol{x}$)

Find the reflection matrix $\mathbf{R}$ that maps the cube of type $\boldsymbol{i}_\square := (\lfloor \boldsymbol{x} \rfloor \text{ modulo } 2)$ (Figure 6) to that of Figure 6(a).

Find the tetrahedron containing $\mathbf{R}(\boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor)$ by testing against three knot planes in Figure 6(a).

Find the permutation matrix $\mathbf{P}$ that maps the tetrahedron to the reference tetrahedron.

$\dot{\boldsymbol{x}} \leftarrow \mathbf{PR}(\boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor)$

for $i = 1$ to 30 **do**

$\boldsymbol{j} \leftarrow \mathcal{J}_7(i)$

$c_i \leftarrow V(\boldsymbol{i}_\square + (\mathbf{PR})^{-1}\boldsymbol{j})$

end for

Evaluate using either the de Casteljau algorithm with the BB-form from Table 1 or the formula in Figure 1 of the supplemental material.

end function

three discrete quasi-interpolation prefilters Q_{12}^{II}, Q_8^{II}, and Q_7^{II} (Table 1).

5.1 Evaluation Time

TABLE 2
Evaluation time (in seconds) of 10^7 points randomly generated inside each volume. (System specifications: Ubuntu 11.04/ quad-core Intel® Xeon® CPU X5550 @2.67GHz with L2 Cache 8MB/ 6GB main memory)

dataset	M_{12}	M_8 (t_8)	M_7 (t_7)	t_7/t_8 (%)
$21^3 \times 2$	3.83445	2.09095	1.55458	74.3
$27^3 \times 2$	4.24062	2.24015	1.69082	75.5
$32^3 \times 2$	4.31606	2.29917	1.75987	76.5
$37^3 \times 2$	4.43997	2.35084	1.79927	76.5
$45^3 \times 2$	4.41845	2.35842	1.84051	78.0
$57^3 \times 2$	4.58235	2.42169	1.88100	77.7
$77^3 \times 2$	6.46921	3.24693	2.66483	82.1
$93^3 \times 2$	7.26688	3.61189	2.98389	82.6
$117^3 \times 2$	7.82863	3.91083	3.18585	81.5

Table 2 compares the evaluation times of the three box-spline filters. While splines of M_{12} can be evaluated efficiently using the de Boor algorithm, it takes much more time than the others due to the at least four times greater number of fetch operations. Evaluating splines of M_7 is faster than those of M_8 thanks to the concise and partially factored spline form. Also, the smaller stencil size induced by the smaller support contributes to the performance gain. Note that, while M_{12} shows the worst performance on the CPU, it shows the best performance on the GPU due to its tensored structure [14] since convolutions with eight data samples can be done using one tri-linear texture fetch performed by the hardware, reducing the number of fetch operations from 128 to 16 and inducing lower computational overhead. Also note that, while the evaluation time is independent of dataset size, the fetch operation overhead increases as the dataset gets larger due to cache misses.

5.2 Reconstruction Quality

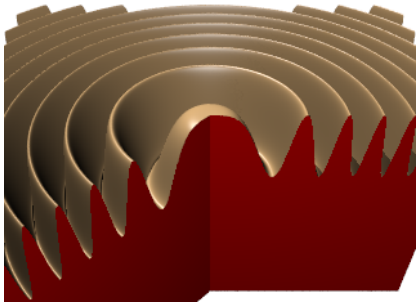


Fig. 7. Marschner-Lobb test function [18].

Figure 8 compares the reconstruction of the test function (Figure 7) using the three box-spline filters with quasi-interpolation prefilters. Note that the Nyquist frequency of the test function is just below 0.05^{-3}, corresponding to 41^3 samples on the Cartesian lattice, and $3/8$ of each volume is clipped to clearly show the cross-section.

When the sampling density is high, M_{12} shows the best quality due to its low post-aliasing property. However, as the density decreases, M_{12} quickly loses a large amount of detail due to its high smoothing aliasing property, which can be seen more clearly in the cross-section view. Comparing M_8 and M_7, while visually similar, more ripples appear in the M_8 reconstruction due to the post-aliasing property. This aliasing is evident in the close-up views in Figure 9. Intuitively, more convolution directions in Ξ_7 induce fewer preferred directions in reconstruction, resulting in reduced aliasing.

Figure 10 shows reconstruction of the `Carp` dataset. The reconstruction by M_{12} loses many details, but the other two images are hardly distinguishable. The subtle improvement by M_7 will be verified in Section 5.5.

TABLE 3
Filter metrics of three filters. [18].

filter	smoothing	post-aliasing
M_{12}	0.94495	0.00004
M_8	0.85287	0.00399
M_7	0.85488	0.00355

5.3 Integral Filter Metrics

Table 3 shows two filter metrics proposed by Marschner and Lobb [18]. Given a filter ϕ, the smoothing metric is

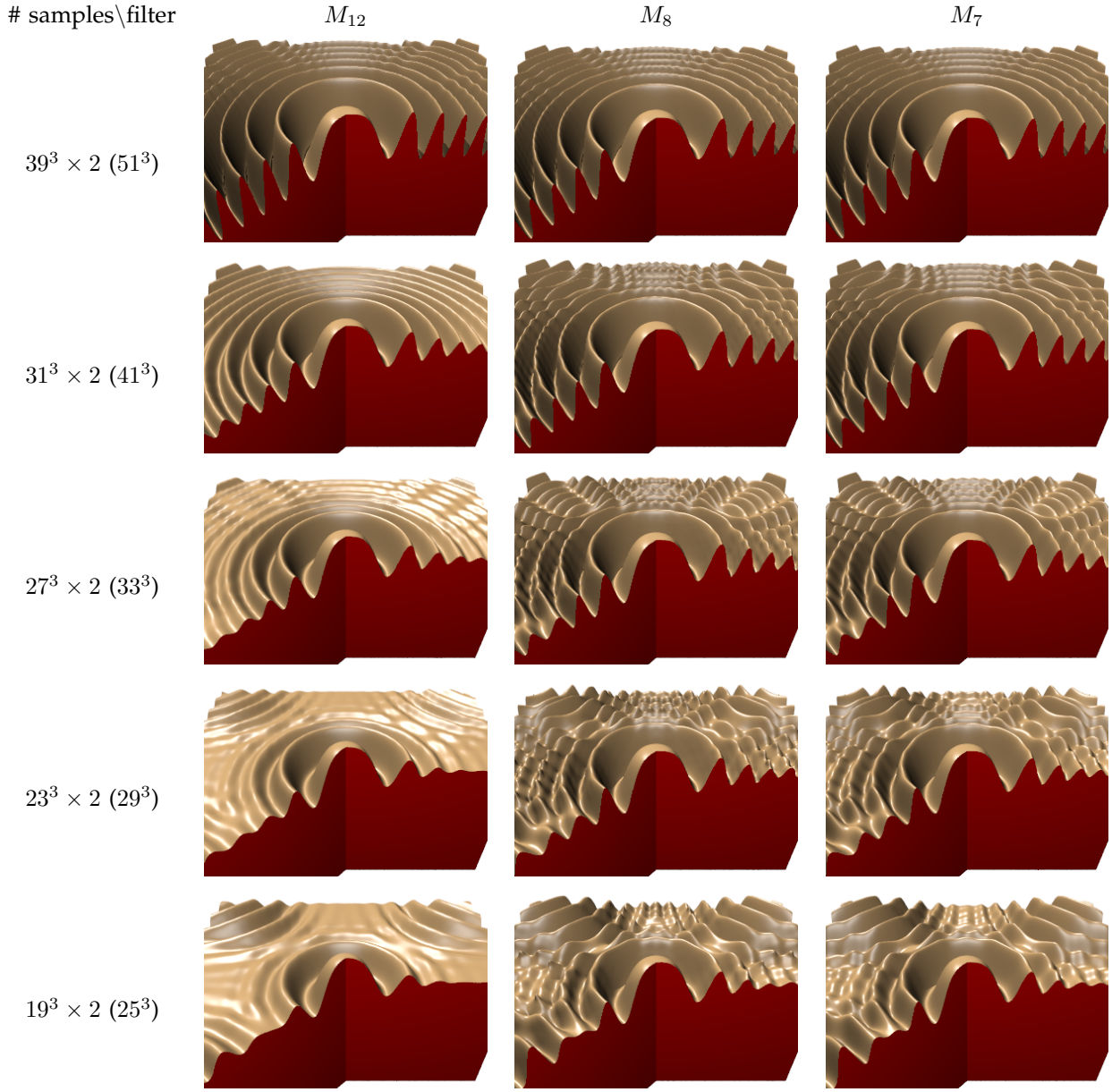


Fig. 8. Reconstruction of the ML (Marschner-Lobb) dataset [18] sampled on the BCC lattice with quasi-interpolation prefilters Q_*^{II}. Numbers in parantheses denote the corresponding number of samples on the Cartesian lattice with the same density. For any density, reconstruction with M_{12} shows the greatest smoothing aliasing. Although reconstructions by M_8 and M_7 look similar, M_7 induces less ripples (Figure 9).

defined as

$$S(\phi) := 1 - \frac{1}{|N_n|} \int_{N_n} |\widehat{\phi}|^2 dV,$$

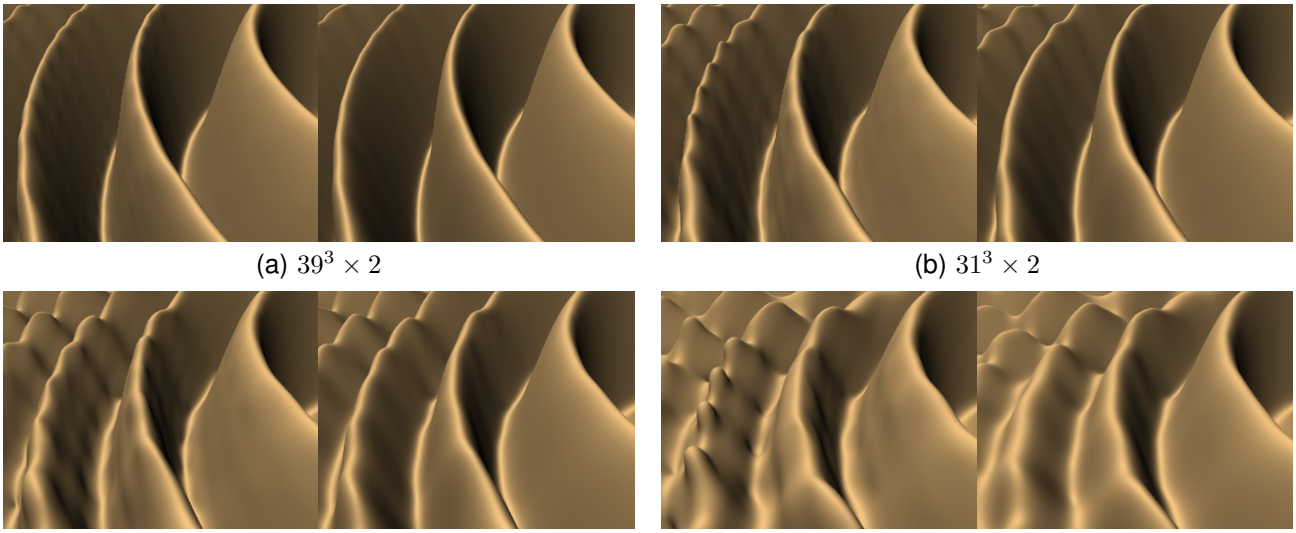
and the post-aliasing metric is defined as

$$P(\phi) := \frac{1}{|N_n|} \int_{\overline{N_n}} |\widehat{\phi}|^2 dV,$$

where N_n is the Nyquist region, and $\overline{N_n}$ is its complement. All the metrics are computed by Matlab's `triplequad` function. Comparing M_8 and M_7, while M_7 shows higher smoothing and lower post-aliasing, the differences are small, but the improvement of the post-aliasing of M_7 is greater than that of smoothing of M_8. On the other hand, M_{12} shows much lower post-aliasing and stronger smoothing properties, as verified in Section 5.2.

5.4 Spectrum Analysis

While the filter metrics are useful to show the aliasing characteristics of filters, they only convey the integral measure and are therefore not suitable to analyze the distributional aliasing characteristics. For this purpose, we evaluated and visualized the spectra of filters on the planes and lines with various orientations. Figure 11 shows the spectra of the three filters evaluated on the planes with various orientations. In any case, $|\widehat{M_{12}}|$



(a) $39^3 \times 2$ (b) $31^3 \times 2$ (c) $27^3 \times 2$ (d) $23^3 \times 2$

Fig. 9. Close-up views of the first four rims of the ML dataset reconstructed by (left) M_8 and (right) M_7. In each case, reconstruction by M_8 shows more ripples on the surface due to its stronger post-aliasing. Note that the surfaces look dull since shadowless lights are used for rendering.

shows very small distribution outside the Nyquist region, which explains the low post-aliasing of M_{12}. Also, the distribution of M_{12} inside the Nyquist region is much lower than others, corresponding to high smoothing aliasing. Comparing $|\widehat{M_8}|$ and $|\widehat{M_7}|$, their overall shapes are very similar. However, $|\widehat{M_8}|$ has a more concentrated distribution outside the Nyquist region, which is noticeable in Figure 11(d). Summing up, we can see that, while the smoothing and post-aliasing of M_8 and M_7 are similar in integral measure, the post-aliasing artifacts of M_8 are more concentrated along specific directions and thus induce more visual aliasing artifacts.

To see the directional aliasing characteristics in more detail, we plotted the spectra of the filters along three directions in Figure 12. As can be seen, M_{12} shows the strongest smoothing aliasing along all directions. While M_{12} shows almost no post-aliasing along the directions $(1,1,0)$ and $(1,1,1)$ (Figure 12(d) and 12(f)), its post-aliasing along $(1,0,0)$ is noticeable since the data are closer to the centers of the replica (Figure 12(b)). Along the direction $(1,0,0)$, M_7 shows higher smoothing but lower post-aliasing than M_8 (Figure 12(b)). They exactly match each other along $(1,1,0)$ (Figure 12(d)). Along the direction $(1,1,1)$, M_8 shows noticeably higher post-aliasing than M_7, but it is still relatively far away from the centers of its closest replica (Figure 12(f)).

5.5 Frequency Error Kernels

While integral filter metrics and distributional analysis of the spectra are helpful to compare filter performance, they do not provide a quantitative error measure of the reconstructed signal accompanied with various prefilters. To this end, we computed the frequency error kernels proposed by Blu and Unser [1].

Figure 13 shows the frequency error kernels of various configurations in the low frequency band. While the error kernels are tri-variate, we only need to consider the case $\boldsymbol{\omega} = (\omega, \omega, \omega)$ since only uniform scaling is acceptable to keep the lattice equivalent. As can be seen, E_{min} of M_{12} is the lowest followed by those of M_8 and M_7. When applied to reconstruction with or without quasi-interpolation prefilters, M_{12} shows the worst approximation behavior. Comparing M_7 and M_8, while the difference is small, M_7 always outperforms M_8.

6 Conclusion and Future Work

We present an alternative symmetric box-spline reconstruction filter on the BCC lattice which has better approximation power than previous methods while also having lower complexity. Moreover, due to the simplified explicit polynomial formula, the computational cost of our method is lower than those of the previous methods. To show the improved reconstruction quality, we computed and analyzed integral filter metrics and frequency error kernels. In addition, each spectrum is evaluated on the planes and lines with various orientations in order to visualize and analyze the distributional aliasing characteristics of the filters in more detail.

We are currently working on implementing a real-time ray-casting module on the GPU to leverage an efficient evaluation. As our future work, we plan to analyze the gradient error of the reconstructed volume and to investigate improved quasi-interpolation prefilters.

Acknowledgments

The author appreciates Dr. Jörg Peters and Dr. Alireza Entezari for their valuable comments. The author also would like to thank all the anonymous reviewers who

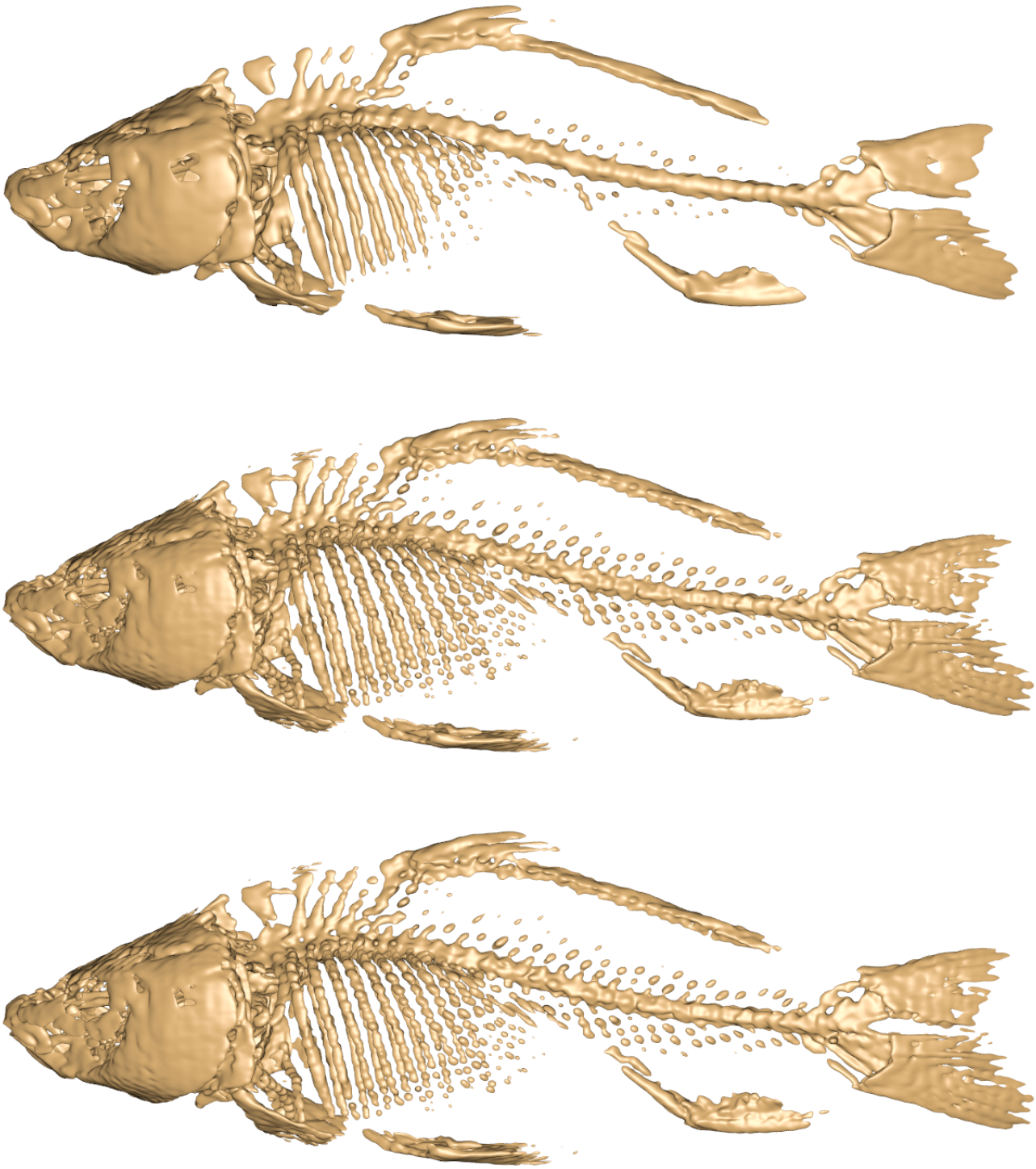


Fig. 10. Reconstruction of `Carp` dataset [23] with (top) M_{12}, (middle) M_8 and (bottom) M_7 and prefilters Q_*^{II}. Only $\approx 3\%$ of the original dataset is used.

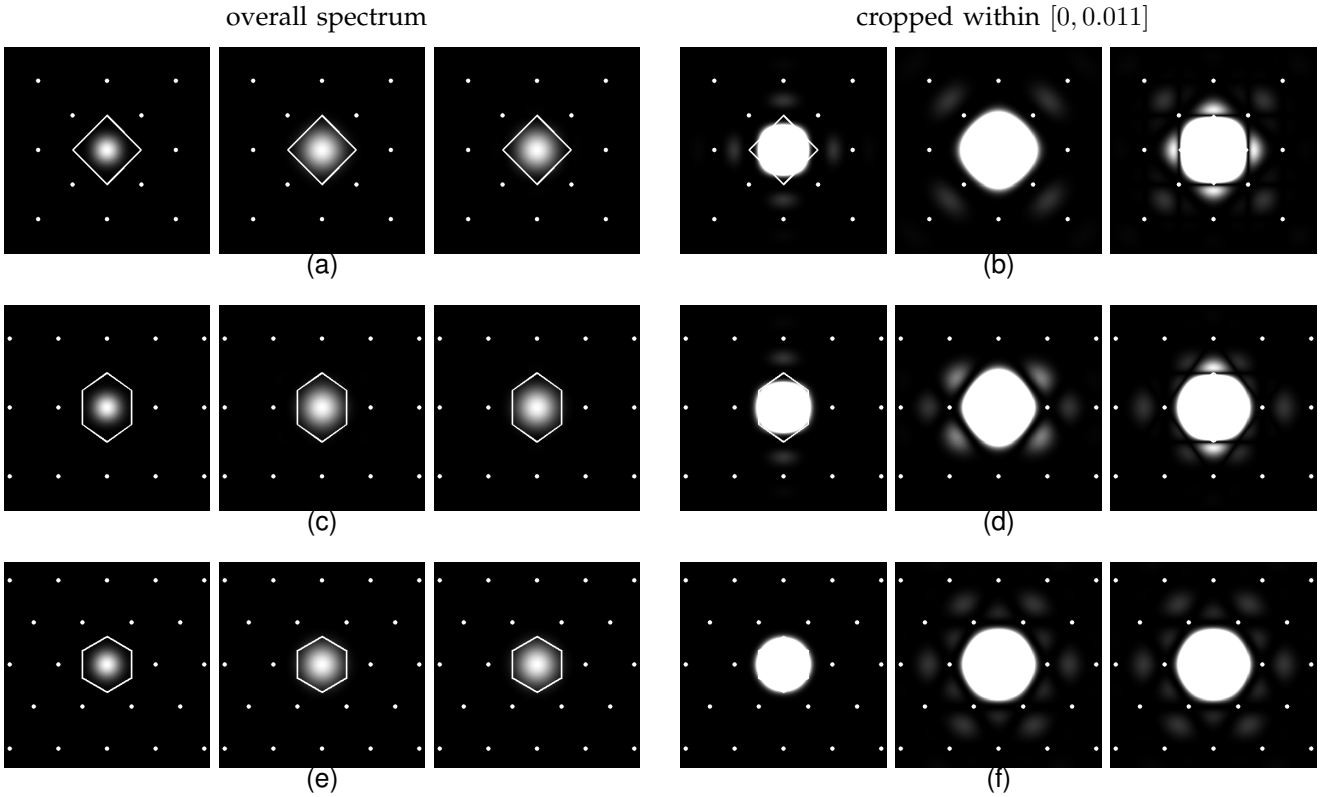


Fig. 11. Spectra of the three filters (left) $|\widehat{M_{12}}|$, (middle) $|\widehat{M_8}|$, and (right) $|\widehat{M_7}|$ evaluated on the three planes with normals (top) $(1,0,0)$ (middle) $(1,1,0)$ (bottom) $(1,1,1)$. The white polygons denote the boundaries of the Voronoi region of the dual lattice, and the white dots denote the dual lattice points.

helped to improve this article. This work was supported by the University of Seoul 2009 Research Fund.

REFERENCES

[1] 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, Oct. 1999.

[2] A. Cavaretta, C. Micchelli, and W. Dahmen. *Stationary Subdivision*. American Mathematical Society, Boston, MA, USA, 1991.

[3] L. Condat and D. Van De Ville. Three-directional box-splines: characterization and efficient evaluation. *IEEE Signal Processing Letters*, 13(7):417–420, July 2006.

[4] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag New York, Inc., New York, NY, USA, 3rd edition, 1998.

[5] B. Csébfalvi. An evaluation of prefiltered B-spline reconstruction for quasi-interpolation on the body-centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):499–512, May–June 2010.

[6] B. Csébfalvi and B. Domonkos. 3D frequency-domain analysis of non-separable reconstruction schemes by using direct volume rendering. In *Proceedings of the 26th Spring Conference on Computer Graphics*, SCCG '10, pages 51–60, New York, NY, USA, 2010. ACM.

[7] B. Csébfalvi and M. Hadwiger. Prefiltered B-spline reconstruction for hardware-accelerated rendering of optimally sampled volumetric data. In *Vision, Modeling, and Visualization*, pages 325–332, 2006.

[8] C. de Boor, K. Höllig, and S. Riemenschneider. *Box splines*. Springer-Verlag New York, Inc., 1993.

[9] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

[10] 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. IEEE Computer Society, 2004.

[11] A. Entezari, M. Mirzargar, and L. Kalantari. Quasi-interpolation on the body centered cubic lattice. *Computer Graphics Forum*, 28(3):1015–1022, June 2009.

[12] A. Entezari and T. Möller. Extensions of the Zwart-Powell box spline for volumetric data reconstruction on the Cartesian lattice. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1337–1344, 2006.

[13] 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, Mar. 2008.

[14] B. Finkbeiner, A. Entezari, D. Van De Ville, and T. Möller. Efficient volume rendering on the body centered cubic lattice using box splines. *Computers & Graphics*, 34(4):409–423, Aug. 2010.

[15] M. Kim, A. Entezari, and J. Peters. Box spline reconstruction on the face-centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1523–1530, Nov.-Dec. 2008.

[16] M. Kim and J. Peters. Fast and stable evaluation of box-splines via the Bernstein-Bézier form. *Numerical Algorithms*, 50(4):381–399, April 2009.

[17] M. Kim and J. Peters. Symmetric box-splines on root lattices. *Journal of Computational and Applied Mathematics*, 235(14):3972–3989, May 2011.

[18] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of the IEEE Conference on Visualization*, pages 100–107, Oct. 1994.

[19] M. Mirzargar and A. Entezari. Voronoi splines. *IEEE Transactions on Signal Processing*, 58(9):4572–4582, Sept. 2010.

[20] Persistence of Vision Pty. Ltd. Persistence of Vision ™ Raytracer, 2004. http://www.povray.org.

[21] J. Peters. C^2 surfaces built from zero sets of the 7-direction box spline. In *IMA Conference on the Mathematics of Surfaces*, pages 463–474, 1994.

[22] 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, 1962.

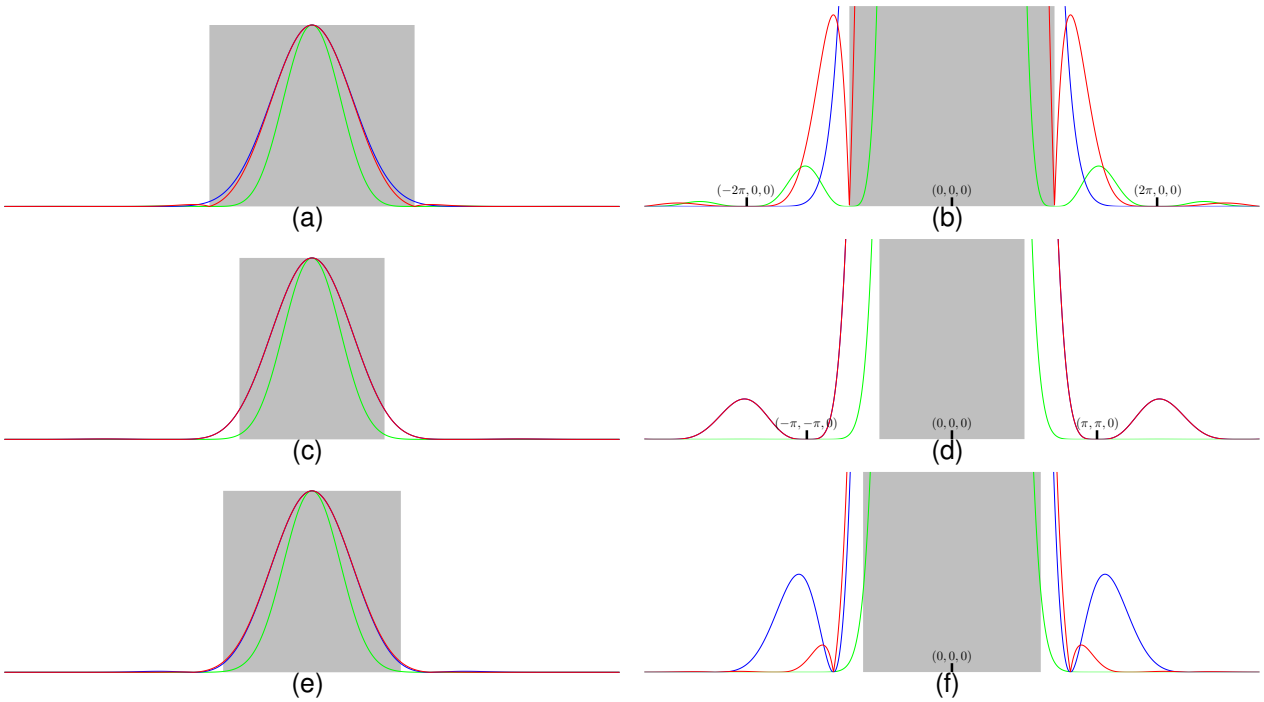


Fig. 12. Three spectra (red) $|\widehat{M_7}|$, (blue) $|\widehat{M_8}|$, and (green) $|\widehat{M_{12}}|$ evaluated along the directions (a) (b) $(1, 0, 0)$, (c) (d) $(1, 1, 0)$, and (e) (f) $(1, 1, 1)$. Figures on the righthand side are magnified 100 times vertically. Shaded areas denote the Nyquist regions along corresponding directions.

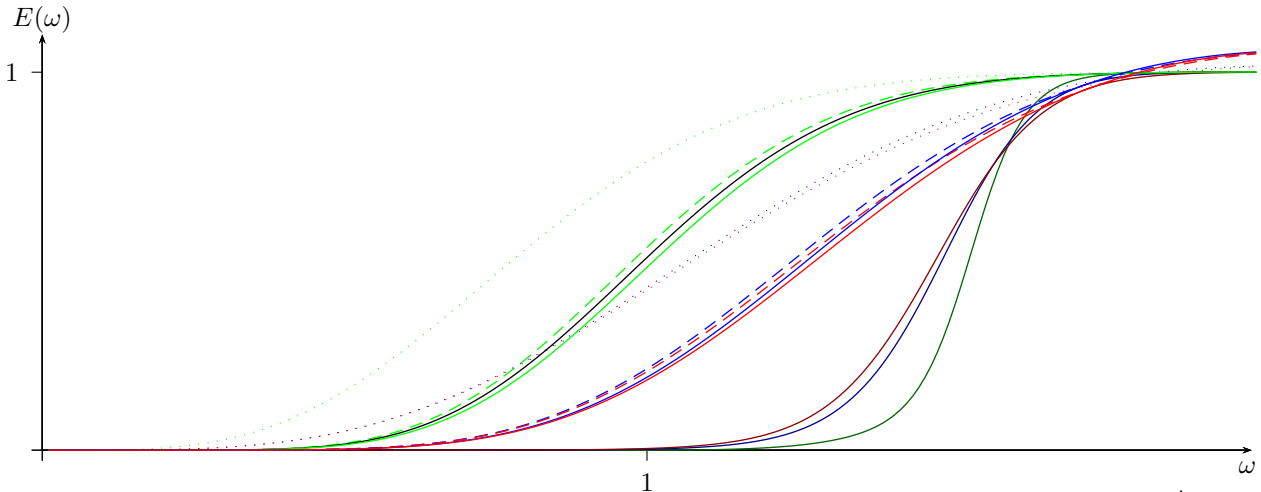


Fig. 13. Error kernels of (red) M_7, (blue) M_8 and (green) M_{12} with (dotted) no prefiltering, (dashed) Q_*^{I}, and (solid) Q_*^{II}. Dark red/blue/green curves denote the optimal error kernels $E_{\mathsf{min}}(\boldsymbol{\omega})$ of the corresponding filters. The black curve denotes the error kernel of M_{12} with the quasi-interpolation prefilter computed by Csébfalvi [5].

[23] S. Roettger. The volume library (online), Jan. 2012. http://www9.informatik.uni-erlangen.de/External/vollib.
[24] D. M. Y. Sommerville. Space-filling tetrahedra in Euclidean space. In *Proceedings of the Edinburgh Mathematical Society*, volume 41, pages 49–57. Cambridge University Press, Feb. 1922.

Minho Kim Minho Kim received a BS degree in Electrical Engineering from Seoul National University, and MS and Ph.D degrees in CISE (Computer and Information Science and Engineering) from the University of Florida in 2008. He is an assistant professor at the School of Computer Science at University of Seoul. His research interests include multi-variate splines on root lattices, volume rendering, and GPU computing.