Box Spline Reconstruction On The Face-Centered Cubic Lattice

Minho Kim, Alireza Entezari and Jörg Peters

Abstract—We introduce and analyze an efficient reconstruction algorithm for FCC-sampled data. The reconstruction is based on the 6-direction box spline that is naturally associated with the FCC lattice and shares the continuity and approximation order of the triquadratic B-spline. We observe less aliasing for generic level sets and derive special techniques to attain the higher evaluation efficiency promised by the lower degree and smaller stencil-size of the C^1 6-direction box spline over the triquadratic B-spline.

Index Terms—Volumetric data reconstruction, box spline, Face-Centered Cubic lattice

◆

1 MOTIVATION

Sampling and reconstruction of trivariate functions is a key task for visualization of volumetric data. The Cartesian lattice is the accepted discretization pattern for sampling since its *separable* structure allows for an easy (i.e., tensor-product) extension of univariate techniques to the trivariate setting. However, the Cartesian lattice is suboptimal when sampling typical data sets. The Body Centered Cubic (BCC) lattice and the Face-Centered Cubic (FCC) lattice outperform the Cartesian lattice both in terms of reducing aliasing and by lower density of samples required for reconstruction.

Although the BCC lattice is the optimal 3D sampling lattice [26], the FCC lattice also shows nearly-optimal sampling properties. In addition the FCC lattice enjoys from a uniform notion of voxel neighborhood (i.e., discrete topology) [10]. This feature is crucial in defining morphological operators in mathematical morphology [22] and has recently proven useful for computing volumetric global illumination [28]. The FCC lattice is often regarded as the 3D counterpart of the quincunx lattice in 2D– both of which allow for subsampling of Cartesian data by a ratio of 2 [32]. This close relationship to the Cartesian lattice allows for developing granular multiresolution transforms studied in [19, 12]. In the information theory literature, the FCC lattice is proven to be the lattice of choice for sampling signals at low relative resolutions [18]. Kovačević [17] and Cooklev [4] proposed sampling video signals on the FCC lattice.

Yet, to date, there is no reconstruction algorithm available that can efficiently interpolate or approximate the underlying function based on the FCC samples.

The contribution of this paper is to fill in the nontrivial technical details that turn the theoretical advantages of the FCC lattice into practically useful algorithms; and to provide the first efficient reconstruction algorithm for data on the FCC lattice. In particular, we

- characterize the continuity and approximation properties of M_6, the C^1 6-direction box spline naturally fitting to the FCC lattice structure (Lemma 1, page 3) ;

- derive an optimal quasi-interpolant for M_6 to efficiently reconstruct a field corresponding to FCC data (Lemma 2, page 3) ;

- derive an explicit representation of M_6 in Bernstein-Bézier (BB) form (Table 1 page 4) and

- give an efficient algorithm for the evaluation of splines that are generated from shifts of M_6 (Algorithm 2, page 5).

We implemented this reconstruction algorithm and compared it to the triquadratic B-spline reconstruction of the Cartesian data. This is a fair

---

- *CISE Department, University of Florida E-mail: {mhkim,entezari,jorg}@cise.ufl.edu.*

comparison since both reconstruction schemes are of equivalent approximation order and continuity (Section 6). By contrast, the known BCC reconstruction filters do not have the same continuity and approximation order as our proposed FCC reconstruction: the well-known BCC reconstruction algorithms are C^0, and C^2; we provide C^2 reconstructions of BCC datasets for rough comparisons.

2 RELATED WORK

While there is a rich literature on univariate reconstruction schemes (e.g., [25, 9, 2, 16, 23, 24]) and tensor-product or a radial extension of univariate reconstruction schemes [21], only limited research has addressed more isotropic multi-dimensional sampling and reconstruction. Based on the arguments of Petersen and Middleton [26], Theußl et al. [30, 29] were first to consider the *BCC lattice*. However, their spherical extension of reconstruction filters does not take advantage of the underlying geometry and therefore blurs the data. Carr et al. [3] studied isosurface extraction on the BCC lattice and Csébfalvi [5] demonstrated a reconstruction using a Gaussian kernel and the principle of generalized interpolation [31]. While the latter approach is isotropic, it does not guarantee best approximation order and results in a numerical scheme without closed form reconstruction kernel. Entezari et al. [11] established the relationship between the BCC lattice and an isotropic four-direction trivariate box spline. Using this box spline, they demonstrated that the theoretical advantages of the BCC lattice can carry over to practice in the form of efficient algorithms for C^0 and C^2 reconstruction. Based on the piecewise-polynomial form of the proposed box splines the cost of reconstruction on the BCC lattice was shown to be *half* that of the corresponding tensor-product B-spline reconstruction on a comparable Cartesian lattice [13].

3 BOX SPLINE BASICS

Before we introduce the FCC-based box spline, we review the basic concepts of box splines that needed to explain the properties of the FCC-based reconstruction. We follow the notation made standard by the 'box spline book' [8] and point to this excellent source for a deeper and more detailed treatment.

Box spline The box spline $M_\Xi : \mathbb{R}^n \to \mathbb{R}$ is defined by a matrix $\Xi \in \mathbb{R}^{n \times \ell}$, $n \leq \ell$, whose columns, called directions, span $\mathbb{R}^n$. It is constructed by consecutive directional convolutions along the directions in Ξ. Alternatively, each value of M_Ξ is defined as the projected volume density of the ℓ-dimensional cube (Figure 1).

M_Ξ is zero outside $\{\Xi t : t \in [0..1]^\ell\}$, the Minkowski set of the directions in Ξ. Let $\mathbb{H}(\Xi)$ be the set of $(n - 1)$-dimensional planes spanned by the directions of Ξ. Then M_Ξ consists of polynomial pieces of (total) degree $\ell - n$ that are delineated by the *mesh* $\Gamma(\Xi)$ obtained by shifting $\mathbb{H}(\Xi)$ on the integer grid $\mathbb{Z}^n$

$$\Gamma(\Xi) := \bigcup_{H \in \mathbb{H}(\Xi)} H + \mathbb{Z}^n. \tag{1}$$
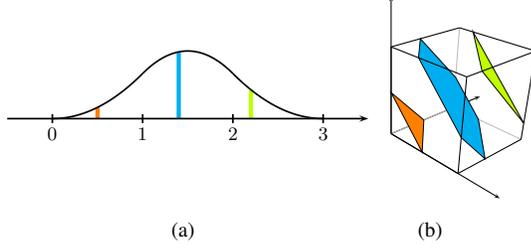
Fig. 1: Quadratic B-spline coincides with (a) box spline $M_{[1\ 1\ 1]}$ defined as a projection of (b) a three-dimensional cube.

Spline Space   The shifts of $M_{\boldsymbol{\Xi}}$ on $\mathbb{Z}^n$ form the spline space

$$S_{M_{\boldsymbol{\Xi}}} := \mathrm{span}\left(M_{\boldsymbol{\Xi}}(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbb{Z}^n}. \tag{2}$$

If $m(\boldsymbol{\Xi}) + 1$ is the minimal number of directions whose removal from $\boldsymbol{\Xi}$ makes the remaining directions do not span $\mathbb{R}^n$, then all polynomials up to degree $m(\boldsymbol{\Xi})$ are contained in $S_{M_{\boldsymbol{\Xi}}}$ [8, (II.59)] and the continuity is at least

$$s \in C^{m(\boldsymbol{\Xi})-1}(\mathbb{R}^n), \quad \forall s \in S_{M_{\boldsymbol{\Xi}}}.$$

The sequence $\left(M_{\boldsymbol{\Xi}}(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbb{Z}^n}$ is *linearly independent* if and only if $\boldsymbol{\Xi}$ is unimodular [8, (II.57)], i.e., all square sub-matrices of $\boldsymbol{\Xi}$ satisfy

$$\det \mathbf{Z} \in \{0, \pm 1\}, \quad \forall \mathbf{Z} \in \boldsymbol{\Xi}. \tag{3}$$

Quasi-interpolation   In image and signal processing literature, the data are often used as the coefficients of the basis splines (i.e., B-spline or box spline). Such reconstruction, therefore, only approximates when the basis functions are of higher degree than linear. The 'pre-filtering' technique modifies the data so that using them as coefficients yields **interpolating** or **optimally approximating** splines. Different spline bases require different pre-filtering rules.

In image processing it is common to design pre-filters to optimally approximate ideal interpolation (i.e., sinc based interpolation) [31]. In approximation theory, the corresponding concept is called **quasi-interpolation**. Quasi-interpolation converts the original data to spline control points so that, if the data are sampled from low-order polynomials, the reconstruction reproduces the polynomials exactly. Reproducing as many low-order polynomials as the basis spline allows often guarantees the optimal approximation order of the reconstruction algorithm [8].

For the box spline associated with $\boldsymbol{\Xi}$, the polynomials up to degree $m(\boldsymbol{\Xi})$ are *reproduced* by the quasi-interpolant of the form

$$(Q_{M_{\boldsymbol{\Xi}}} f)(\boldsymbol{x}) := \sum_{\boldsymbol{j} \in \mathbb{Z}^n} M_{\boldsymbol{\Xi}}(\boldsymbol{x} - \boldsymbol{j}) \lambda_{M_{\boldsymbol{\Xi}}}(f(\cdot + \boldsymbol{j})) \tag{4}$$

with the functional $\lambda_{M_{\boldsymbol{\Xi}}} f$ defined in [8, (III.22)] and explicitly stated in all cases, below. While $Q_{M_{\boldsymbol{\Xi}}}$ provides the *optimal* approximation order $m(\boldsymbol{\Xi}) + 1$ from $S_{M_{\boldsymbol{\Xi}}}$, computing it is efficient since $\lambda_{M_{\boldsymbol{\Xi}}}$ averages only local data at each lattice point (see e.g. (9) and (13)).

Change of Variables   Under an invertible linear map $\boldsymbol{L}$, [8, (I.23)]

$$M_{\boldsymbol{\Xi}} = |\det \boldsymbol{L}| M_{\boldsymbol{L}\boldsymbol{\Xi}} \circ \boldsymbol{L}. \tag{5}$$

In particular, any sum of shifts, by $\boldsymbol{j} \in \boldsymbol{L}\mathbb{Z}^n$ (on the possibly non-Cartesian lattice), of the box spline $|\det \boldsymbol{L}| M_{\boldsymbol{L}\boldsymbol{\Xi}}$ weighted by coefficients $a(\boldsymbol{j}) \in \mathbb{R}$ can be expressed as a weighted sum of the shifts of $M_{\boldsymbol{\Xi}}$ on the Cartesian lattice $\mathbb{Z}^n$:

$$\sum_{\boldsymbol{j} \in \boldsymbol{L}\mathbb{Z}^n} |\det \boldsymbol{L}| M_{\boldsymbol{L}\boldsymbol{\Xi}}(\cdot - \boldsymbol{j}) a(\boldsymbol{j}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^n} M_{\boldsymbol{\Xi}}(\boldsymbol{L}^{-1} \cdot -\boldsymbol{k}) a(\boldsymbol{L}\boldsymbol{k}). \tag{6}$$

The quasi-interpolant $Q_{M_{\boldsymbol{\Xi}}}^{\boldsymbol{L}}$ for the spline space

$$S_{M_{\boldsymbol{\Xi}}}^{\boldsymbol{L}} := \mathrm{span}\left(|\det \boldsymbol{L}| M_{\boldsymbol{L}\boldsymbol{\Xi}}(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \boldsymbol{L}\mathbb{Z}^n}$$

that provides the same approximation order as does $Q_{M_{\boldsymbol{\Xi}}}$ for $S_{M_{\boldsymbol{\Xi}}}$ is defined by the functional

$$\lambda_{M_{\boldsymbol{\Xi}}}^{\boldsymbol{L}}\left(f(\cdot + \boldsymbol{j})\right) := \lambda_{M_{\boldsymbol{\Xi}}}\left((f \circ \boldsymbol{L})(\cdot + \boldsymbol{L}^{-1}\boldsymbol{j})\right). \tag{7}$$

3.1   The triquadratic B-spline on the Cartesian lattice

We will compare FCC reconstruction to tensor-product B-spline reconstruction. With $\mathbf{I}$ the 3×3 identity matrix, the triquadratic B-spline has the *centered* box spline representation

$$M_{333} := M_{\boldsymbol{\Xi}_{333}}(\cdot + (1.5, 1.5, 1.5)), \qquad \boldsymbol{\Xi}_{333} := \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix}.$$

Since up to six directions of $\boldsymbol{\Xi}_{333}$ lie in a plane, $m(\boldsymbol{\Xi}_{333}) = (9-6) - 1 = 2$. Therefore all quadratics are contained in

$$S_{333} := \mathrm{span}\left(M_{333}(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbb{Z}^3} \tag{8}$$

and all the splines in S_{333} are C^1. The total degree of the polynomial pieces is $9 - 3 = 6$. The quasi-interpolant Q_{333} with optimal approximation order for S_{333} is defined by the functional

$$\lambda_{333} f := f(\mathbf{0}) - \frac{1}{8}\left((D_1^2 + D_2^2 + D_3^2)f\right)(\mathbf{0}).$$

For a discrete input $f : \mathbb{Z}^3 \to \mathbb{R}$, approximating D^2 with central differencing yields

$$\lambda_{333} f \approx \frac{7}{4} f(\mathbf{0}) - \frac{1}{8} \sum_{k=1}^{3} (f(\mathbf{i}_k) + f(-\mathbf{i}_k)). \tag{9}$$

De Boor's algorithm [7] is the standard stable evaluation method for splines in S_{333}.

4   SPLINES ON THE FCC LATTICE

In this section, we introduce splines suitable for C^1 reconstruction on the FCC lattice. We start with a more detailed description of the geometry of the lattice to paint a broad picture.

4.1   The geometry of the FCC lattice

The FCC lattice consists of all Cartesian lattice points whose coordinates sum to an even number;

$$\mathbf{G}\mathbb{Z}^3, \qquad \mathbf{G} := \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

$\mathbf{G}$ is called *generator matrix* of the FCC lattice.

The Voronoi cell of any FCC lattice point is a rhombic dodecahedron (see Figure 2a). In order to perform nearest neighbor interpolation, one can use the structure of this polyhedron and quickly find the nearest FCC point to an arbitrary point $\boldsymbol{x} \in \mathbb{R}^3$. Algorithm 1 explains an efficient method to determine the nearest FCC lattice point. If the sum of coordinates of the nearest Cartesian lattice point $\boldsymbol{j}$, obtained by rounding $\boldsymbol{x}$, is even then $\boldsymbol{j}$ is on the FCC lattice (Figure 3a). Otherwise, we determine which of 6 Voronoi cell $\boldsymbol{j}$ belongs to (Figure 3b).
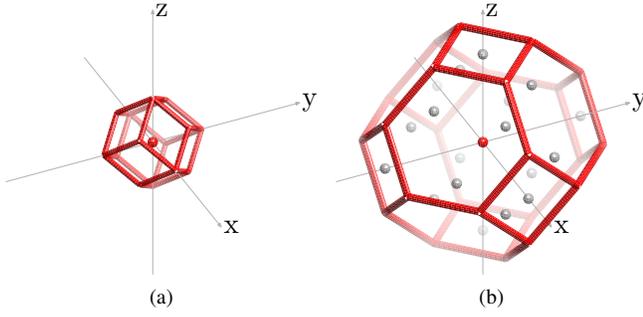
Fig. 2: (a) Rhombic dodecahedron; the Voronoi cell of the FCC lattice. (b) Truncated Octahedron; the support of M_6.

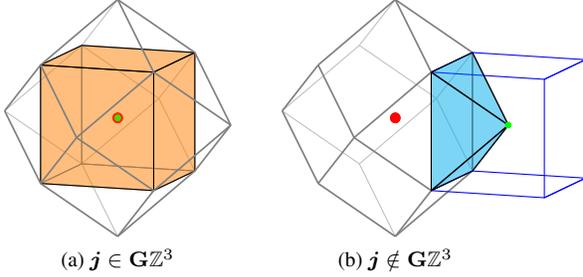(a) $\boldsymbol{j} \in \mathbf{G}\mathbb{Z}^3$ (b) $\boldsymbol{j} \notin \mathbf{G}\mathbb{Z}^3$

Fig. 3: (a) Region where mapping to the nearest Cartesian lattice point (green) yields the nearest FCC lattice point (red); (b) Region requiring further work (Algorithm 1).

As illustrated in Figure 5a, there are 12 FCC neighbors located at $(0, \pm 1, \pm 1)$, $(\pm 1, 0, \pm 1)$ and $(\pm 1, \pm 1, 0)$. This neighborhood forms a cuboctahedron (and can be used to construct a linear element [27] on the FCC lattice). To obtain a smoother construction, we expand this neighborhood to include 6 additional FCC samples at offsets $(\pm 2, 0, 0)$, $(0, \pm 2, 0)$ and $(0, 0, \pm 2)$. This yields the *truncated octahedron* (Figure 2b) associated with the 19-point neighborhood on the FCC lattice

$$\{\boldsymbol{k} \in \mathbf{G}\mathbb{Z}^3 : \|\boldsymbol{k} - \boldsymbol{j}\|_2 \leq 2\}.$$

Since all faces of the truncated octahedron exhibit point symmetry, it is a zonohedron with 6 zones. This relationship allows for a natural construction of a box spline, whose support is a truncated octahedron, by projecting a six-dimensional hypercube.

4.2   A Six-Direction C^1 Box Spline on the FCC lattice

A box spline corresponding to the projection of the six-dimensional hypercube is introduced in [10] for the FCC lattice. This box spline is represented by the direction vectors of the 6 zones:

$$\Xi_6 := [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_6] := \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix}. \quad (10)$$

Algorithm 1 Finding the nearest FCC lattice point from $\boldsymbol{x} \in \mathbb{R}^3$

$\boldsymbol{j} \leftarrow$ Nearest Cartesian lattice point from $\boldsymbol{x}$
if $\boldsymbol{j}$ is not an FCC lattice point **then**
　$\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{x} - \boldsymbol{j}$
　$k \leftarrow$ index of $\max\{|\widetilde{\boldsymbol{x}}(1)|, |\widetilde{\boldsymbol{x}}(2)|, |\widetilde{\boldsymbol{x}}(3)|\}$
　$\boldsymbol{j}(k) \leftarrow \boldsymbol{j}(k) + \text{sign}(\widetilde{\boldsymbol{x}}(k))$
end if
return $\boldsymbol{j}$

Since the theory recorded in [8] is formulated for a box spline shifted on the Cartesian lattice, we consider

where

$$\widetilde{\Xi}_6 := \left[\tilde{\boldsymbol{\xi}}_1, \ldots, \tilde{\boldsymbol{\xi}}_6\right] := \mathbf{G}^{-1}\Xi_6 = \begin{bmatrix} 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}.$$

Then, by (5),

$$M_{\Xi_6} := \frac{1}{|\det \mathbf{G}|} M_{\widetilde{\Xi}_6} \circ \mathbf{G}^{-1}$$

is a trivariate box spline on the FCC lattice. Since M_{Ξ_6} is normalized according to the Cartesian lattice rather than the FCC lattice, we renormalize it and center it to obtain the box spline

$$M_6 := |\det \mathbf{G}| M_{\Xi_6}(\cdot + (1, 1, 1))$$

to be used as the *reconstruction filter* on the FCC lattice. The spline based on the FCC shifts of M_6 can therefore (see (6)) be represented on $\mathbb{Z}^3$ in terms of $M_{\widetilde{6}} := M_{\widetilde{\Xi}_6}(\cdot + (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}))$ as

$$\sum_{\boldsymbol{j} \in \mathbf{G}\mathbb{Z}^3} M_6(\cdot - \boldsymbol{j})a(\boldsymbol{j}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^3} M_{\widetilde{6}}(\mathbf{G}^{-1} \cdot -\boldsymbol{k})a(\mathbf{G}\boldsymbol{k}).$$

Lemma 1 (spline space)**.** *The box spline M_6 is C^1 and piecewise polynomial of total degree 3. The corresponding spline space*

$$S_6 := \text{span}\left(M_6(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbf{G}\mathbb{Z}^3} \quad (11)$$

contains all quadratic polynomials and has approximation order 3. The sequence $\left(M_6(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbf{G}\mathbb{Z}^3}$ is linearly independent.

Proof. By Section 3, M_6 is piecewise polynomial of (total) degree $6 - 3 = 3$. Since at most 3 directions lie in a plane, we conclude $m(\Xi_6) = m(\widetilde{\Xi}_6) = (6 - 3) - 1 = 2$. This implies that the splines are C^1, that all quadratics lie in the space [8, (II.59)] and that the approximation order is 3 [8, (III.1)]. The structure of the matrix $\widetilde{\Xi}_6$ makes it easy to verify that all triples of directions form invertible matrices $\boldsymbol{Z} \in \mathbb{R}^{3 \times 3}$ except for

$$[\tilde{\boldsymbol{\xi}}_2 \tilde{\boldsymbol{\xi}}_4 \tilde{\boldsymbol{\xi}}_6], [\tilde{\boldsymbol{\xi}}_1 \tilde{\boldsymbol{\xi}}_4 \tilde{\boldsymbol{\xi}}_5], [\tilde{\boldsymbol{\xi}}_1 \tilde{\boldsymbol{\xi}}_3 \tilde{\boldsymbol{\xi}}_6], [\tilde{\boldsymbol{\xi}}_2 \tilde{\boldsymbol{\xi}}_3 \tilde{\boldsymbol{\xi}}_5].$$

Since the determinants of all remaining elements are ± 1, (3) implies that the sequence $\left(M_6(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbf{G}\mathbb{Z}^3}$ is linearly independent. □

Approximation order 3 can be achieved by the following quasi-interpolant.

Lemma 2 (quasi-interpolant)**.** *A quasi-interpolant Q_6 for S_6 reproducing quadratic polynomials is defined by the functional*

$$\lambda_6 f := f(\mathbf{0}) - \frac{1}{24} \sum_{\boldsymbol{\xi} \in \Xi_6} (D_{\boldsymbol{\xi}}^2 f)(\mathbf{0}). \quad (12)$$

Proof. Following the construction suggested in [8], the quasi-interpolant $Q_{\widetilde{6}}$ for the spline space

$$S_{\widetilde{6}} := \text{span}\left(M_{\widetilde{6}}(\cdot - \boldsymbol{j})\right)_{\boldsymbol{j} \in \mathbb{Z}^3}$$

is defined by the functional

$$\begin{aligned} \lambda_{\widetilde{6}} f =& f(\mathbf{0}) - \frac{1}{8}((D_1^2 + D_2^2 + D_3^2)f)(\mathbf{0}) \\ & + \frac{1}{12}((D_1 D_2 + D_2 D_3 + D_3 D_1)f)(\mathbf{0}) \\ =& f(\mathbf{0}) - \frac{1}{24}((D_1^2 + D_2^2 + D_3^2 \\ & + (D_1 - D_2)^2 + (D_2 - D_3)^2 + (D_3 - D_1)^2)f)(\mathbf{0}) \\ =& f(\mathbf{0}) - \frac{1}{24} \sum_{\boldsymbol{\xi} \in \widetilde{\Xi}_6} (D_{\boldsymbol{\xi}}^2 f)(\mathbf{0}). \end{aligned}$$

By (7), the quasi-interpolant Q_6 for S_6 can be obtained from $\lambda_{\widetilde{6}}$ by replacing $D_{\boldsymbol{\xi}}$ with $D_{\mathbf{G}\boldsymbol{\xi}}$ and this yields the claim. $\qquad\square$

For a discrete input $f : \mathbf{G}\mathbb{Z}^3 \to \mathbb{R}$, approximating D^2 with central differencing yields

$$\lambda_6 f \approx \frac{3}{2} f(\mathbf{0}) - \frac{1}{24} \sum_{\boldsymbol{\xi} \in \boldsymbol{\Xi}_6} (f(\boldsymbol{\xi}) + f(-\boldsymbol{\xi})), \tag{13}$$

where the right hand side averages the 12 first-ring neighbors around a FCC lattice point.

4.3 The polynomial pieces and evaluation of the FCC spline

Since M_6 is centered at the origin, the mesh Γ_6 generated by M_6 is the same as $\Gamma(\boldsymbol{\Xi}_6)$ but shifted by $(1,1,1)$. The non-axis-aligned knot planes of M_6 decompose $\mathbb{R}^3$ into *octet-truss* [15] structures (Figure 4) and the axis-aligned knot planes decompose each octahedron further into eight congruent tetrahedra such as the one shown in Figure 5a. Note that, the equilateral tetrahedron with blue edges in Figure 5b is decomposed into four tetrahedra (cyan tetrahedron in this figure) by the Voronoi cell of the FCC lattice. However, this equilateral tetrahedron is not split by knot planes and hence is the domain of a single polynomial piece, not of four. Shifting the truncated octahedron (Figure 2b) that represents the support of M_6, we find that 16 different shifts of M_6 overlap at any generic location. In other words, the size of the stencil of our reconstruction (set of coefficients to evaluate a given point) is 16.

Lemma 3 (reconstruction cost). *Reconstruction at an arbitrary point depends on at most 16 FCC points.*

Proof. The set of all the square invertible sub-matrices of $\widetilde{\boldsymbol{\Xi}}_6$ identified in Lemma 1 contains $\binom{3+3}{3} - 4 = 16$ elements. The claim follows by [8, (II.57)] and [8, (II.32)]. $\qquad\square$

Fig. 4: The octet-truss structure: layers of octahedra in an egg carton pattern with two types of tetrahedra filling in the voids.

To evaluate a spline $s \in S_6$ at $\boldsymbol{x} \in \mathbb{R}^3$ via the polynomial form, let $\dot{\boldsymbol{x}}$ be the nearest FCC lattice point from $\boldsymbol{x}$. Since the setup is symmetric with respect to the three axes, we need only consider the $(+,+,+)$ octant. There are two types of tetrahedra, σ_- and σ_+,

	σ_-	σ_+
v_0	0,0,0	1,1,1
v_1	1,0,0	0,1,0
v_2	0,1,0	1,0,0
v_3	0,0,1	0,0,1

$$\tag{14}$$

Table 1: BB-coefficients (scaled up by 24) of the polynomial pieces of type J_- and J_+. The columns index the BB-coefficients by their barycentric coordinates (scaled up by 3) with respect to the vertices in (14) and the row indices are the shifts of the box spline M_6 to the positions marked in Figure 5.

in the Voronoi cell of $\dot{\boldsymbol{x}}$ (Figure 5). Correspondingly, for $\widetilde{\boldsymbol{x}} := \boldsymbol{x} - \dot{\boldsymbol{x}}$ in the $(+,+,+)$ octant,

$$s(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbf{G}\mathbb{Z}^3} a(\boldsymbol{k}) M_6(\boldsymbol{x} - \boldsymbol{k}) \tag{15}$$

$$= \begin{cases} \sum_{\boldsymbol{j} \in J_-} a(\dot{\boldsymbol{x}} + \boldsymbol{j}) B_-^{\boldsymbol{j}}(\beta_-(\widetilde{\boldsymbol{x}})) & \text{if } \|\widetilde{\boldsymbol{x}}\|_1 \le 1 \quad (\text{type}-) \\ \sum_{\boldsymbol{j} \in J_+} a(\dot{\boldsymbol{x}} + \boldsymbol{j}) B_+^{\boldsymbol{j}}(\beta_+(\widetilde{\boldsymbol{x}})) & \text{if } \|\widetilde{\boldsymbol{x}}\|_1 > 1 \quad (\text{type}+). \end{cases}$$

where (for $* \in \{-,+\}$),

- $J_* := \{\boldsymbol{j} \in \mathbb{Z}^3 : M_6(\widetilde{\boldsymbol{x}} - \boldsymbol{j}) \ne 0, \widetilde{\boldsymbol{x}} \text{ interior of } \sigma_*\}$ denotes the stencil, the set of 16 FCC shifts of the box spline (Figure 5 and row indices in Table 1);

- $B_*^{\boldsymbol{j}}(\mathbf{u}) := \sum_{|\alpha|=3} c_*^{\boldsymbol{j}}(\alpha) b_\alpha(\mathbf{u})$ is the polynomial in Bernstein-Bézier (BB) form associated with the shift $\boldsymbol{j}$ of type $*$ via its BB-coefficients $\{c_*^{\boldsymbol{j}}(\alpha)\}$ listed in Table 1. Here $\{b_\alpha(\mathbf{u}) := \binom{3}{\alpha} \mathbf{u}^\alpha : \sum_{i=0}^3 |\alpha(i)| = 3, \alpha \in \mathbb{Z}_+^4\}$ are the well-known BB basis polynomials [6, 14];

- $\beta_*(\widetilde{\boldsymbol{x}})$ the barycentric coordinate of $\widetilde{\boldsymbol{x}}$ with respect to the domain tetrahedron σ_* (Figure 5 and (14)).

The BB-coefficients in Table 1 have been computed with a symbolic tool.

Algorithm 2 implements (15) taking symmetry into account, i.e. shows the pseudocode for evaluating a spline in S_6 at $\boldsymbol{x}$. (We write $\boldsymbol{\sigma}\mathbf{z}$ to indicate $(\boldsymbol{\sigma}\mathbf{z})(j) := \sigma(j)\mathbf{z}(j)$ for $j = 1, 2, 3$. Also the assignment $\boldsymbol{\sigma} \leftarrow \text{sign}(\widetilde{\boldsymbol{x}})$ is coordinate-wise.) We first transform $\boldsymbol{x}$ to the local coordinate $\widetilde{\boldsymbol{x}}$ depending on the type, $-$ or $+$, and leveraging symmetry, we sum the 16 polynomials $\{B_*^{\boldsymbol{j}} : \boldsymbol{j} \in J_*\}$ defined in Table 1 weighted by their coefficients $a(\boldsymbol{j})$. (C++ sample codes can be found at http://www.cise.ufl.edu/research/Surflab/08vis.)

5 RECONSTRUCTION

There are three approaches to reconstructing a function in terms of M_6 from data on the FCC lattice:

- *approximation*: interpret data as control points of the shifts of M_6,

Table 1

Header (α, scaled up by 3):

	α:	3 0 0 0	2 1 0 0	1 2 0 0	0 3 0 0	2 0 1 0	1 1 1 0	0 2 1 0	1 0 2 0	0 1 2 0	0 0 3 0	2 0 0 1	1 1 0 1	0 2 0 1	1 0 1 1	0 1 1 1	0 0 2 1	1 0 0 2	0 1 0 2	0 0 1 2	0 0 0 3
J_-	0, 0, 0	12	12	8	4	12	12	8	8	8	4	12	12	8	12	12	8	8	8	8	4
	0, 1, 1	1	1			2	2		4	4	4	2	2		4	8	4		4	8	4
	1, 0, 1	1	2	4	4	1	2	4		2		2	4	8	2	4		4			
	1, 1, 0	1	2	4	4	2	4	8	4	8	4	1	2	4	2	4	4				
	-1, 0, 1	1				1						2						4		4	4
	-1, 1, 0	1				2			4		4	1									
	0, -1, 1	1	1			2	2					2	2					4	4		4
	0, 1, -1	1	1			2	2		4	4	4										
	1, -1, 0	1	2	4	4	1	2	4				1	2	4							
	1, 0, -1	1	2	4	4	1	2	4													
	0, -1, -1	1	1																		
	-1, 0, -1	1				1															
	-1, -1, 0	1										1									
	2, 0, 0				4																
	0, 2, 0									4											
	0, 0, 2																				4
J_+	0, 0, 0			4	4			4	8	4	8	4		4	8	4	12	8	4	8	8	4
	0, 1, 1	4	8	4	4	4	4			8	12	8	4		8	8	4	8	4	4	
	1, 0, 1	4	4			8	4		8	4	4	8		4	12		4	8	4	8	4
	1, 1, 0	4	8	4	4	8	12	8	8	8	4	4	4	4	4	4	4				
	1, 1, 2	4																			
	1, 2, 1	4	4																		
	2, 1, 1	4					4														
	0, 2, 0			4	4																
	-1, 1, 0				4									4							
	0, 1, -1				4				4												
	2, 0, 0								4		4										
	1, -1, 0										4										
	1, 0, -1										4						4				
	0, 0, 2																	4			4
	-1, 0, 1																		4		4
	0, -1, 1																			4	4

(a) σ_-: $\|\widetilde{\boldsymbol{x}}\|_1 \leq 1$ (b) σ_+: $\|\widetilde{\boldsymbol{x}}\|_1 > 1$
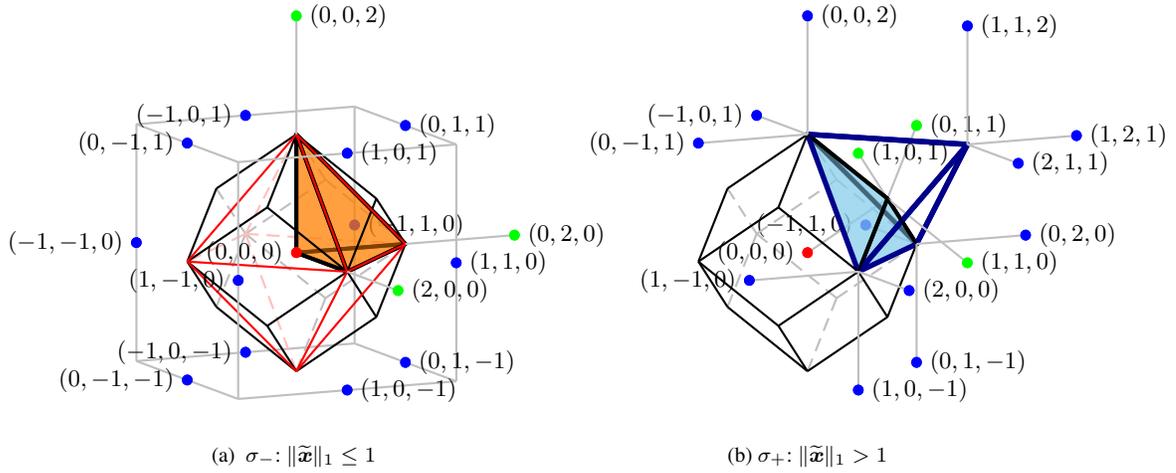
Fig. 5: Two congruence classes of tetrahedra: (a) σ_- and (b) σ_+. 16 lattice points, of which 13 are shared between the two types, determine the polynomial piece on σ_*, $* \in \{-, +\}$. All other polynomial pieces can be obtained from σ_- and σ_+ by symmetry. The black lines of the rhombic dodecahedron delineate the Voronoi cell of the FCC lattice. The 16 FCC points marked in each figure, are used as row indices to select BB-form coefficients of the resulting spline in Table 1.

Algorithm 2 Evaluation of the FCC spline at $\boldsymbol{x} \in \mathbb{R}^3$

$\dot{\boldsymbol{x}} \leftarrow$ nearest FCC lattice point from $\boldsymbol{x}$
$\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{x} - \dot{\boldsymbol{x}}$
$\boldsymbol{\sigma} \leftarrow \text{sign}(\widetilde{\boldsymbol{x}})$
if $\|\widetilde{\boldsymbol{x}}\|_1 \leq 1$ **then**
    $y \leftarrow \sum_{\boldsymbol{j} \in J_-} a(\dot{\boldsymbol{x}} + \boldsymbol{\sigma}\boldsymbol{j}) B_-^{\boldsymbol{j}}(\beta_-(\boldsymbol{\sigma}\widetilde{\boldsymbol{x}}))$
else
    $y \leftarrow \sum_{\boldsymbol{j} \in J_+} a(\dot{\boldsymbol{x}} + \boldsymbol{\sigma}\boldsymbol{j}) B_+^{\boldsymbol{j}}(\beta_+(\boldsymbol{\sigma}\widetilde{\boldsymbol{x}}))$
end if
return y

- *quasi-interpolation*: pre-process the data according to (13), or

- *interpolation*: compute the coefficients by solving the linear system.

Let $f : \mathbf{G}\mathbb{Z}^3 \to \mathbb{R}$ be the discrete FCC dataset and a spline is represented as (15). Interpreting data at $f(\boldsymbol{j})$ as control points, i.e. setting $a(\boldsymbol{j}) := f(\boldsymbol{j})$, is clearly the most convenient option but results in additional smoothing and shifts of the level sets that are some times undesirable. Interpolation on the other extreme, while always possible because the linear systems for interpolation are invertible (Lemma 1), has a prohibitive cost for large data sets. Moreover, while the data are now exactly interpolated, the rationale for doing so is weak since the data are typically the result of filtering and discretization; so pointwise exact replication by itself says little about the fidelity of the reconstructed function. In fact, spline interpolation is well-known to suffer from a ringing phenomenon [20] where gradients change rapidly. Our method of choice is therefore quasi-interpolation since it balances cost and approximation quality: the approximation power matches that of interpolation, but with a small, local footprint.

Choosing quasi-interpolation, the coefficients in (15) are

$$a(\boldsymbol{j}) := \lambda_6(f(\cdot + \boldsymbol{j})) = \frac{3}{2}f(\boldsymbol{j}) - \frac{1}{24}\sum_{\boldsymbol{\xi} \in \boldsymbol{\Xi}_6}(f(\boldsymbol{j} + \boldsymbol{\xi}) + f(\boldsymbol{j} - \boldsymbol{\xi})).$$

Equation (15) and Table 1 then define the polynomial pieces of the reconstructed function. The function can then be evaluated, for example, using de Casteljau's algorithm [6, 14] and *exact* level sets can be rendered by ray-tracing (see Section 6). All images in this work are rendered from reconstruction of data after quasi-interpolation (i.e., pre-filtering).

6 COMPARISON OF C^1 CARTESIAN VS. FCC SPLINE RECONSTRUCTION

(a)

(b)
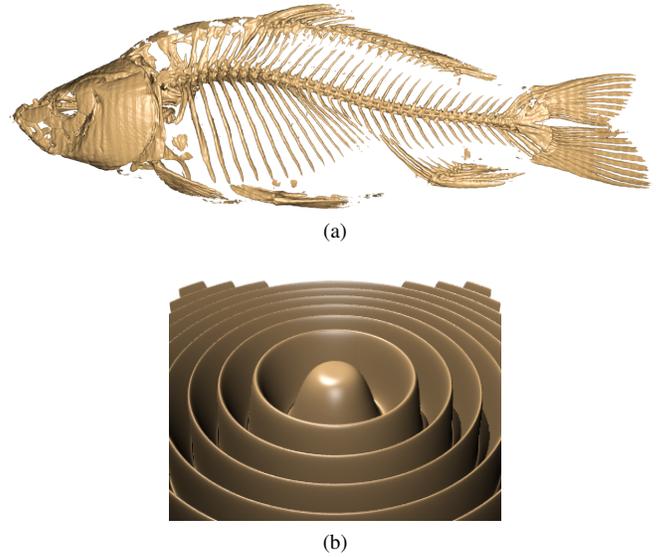
Fig. 6: Benchmark datasets: (a) Carp dataset at the resolution of $256 \times 256 \times 512$ (density 1) reconstructed by quasi-interpolating triquadratic B-spline (Courtesy of http://www9.informatik.uni-erlangen.de/External/vollib). This relatively high-resolution dataset is used as the standard 'truth' function for our experiments at low resolutions. (b) Marschner-Lobb test function.

Volumetric data are traditionally sampled and reconstructed on the Cartesian lattice. A suitable anti-aliasing filter is applied to band-limit the spectrum of the sampled data within the Nyquist region, which is the Voronoi cell of the dual lattice. The FCC lattice cell is clearly different from the Cartesian and therefore testing FCC sampling and reconstruction on real-life datasets is only possible once true FCC sampling devices are available. We constructed comparable FCC and Cartesian datasets by subsampling a high-density Cartesian dataset into coarser FCC and Cartesian datasets with comparable resolutions.

We experimented with the Marschner-Lobb benchmark dataset (Fig-

ure 6b) for evaluating the accuracy of reconstruction algorithms [20]. Figure 8 compares three reconstruction methods on different lattices and filters described in Table 3. It demonstrates reduced aliasing in the FCC method compared to the commonly-used Cartesian method. In particular, the circular rims are almost perfectly reconstructed in the FCC method while the Cartesian method shows severe oscillations. Compared with the BCC method, the FCC method captures the 'valley' of the original function better than the BCC method where the valleys are smoothed out due to the higher degree filter. This can also be justified by the higher smoothing metric of the reconstruction filter of the BCC method (Table 3). But the FCC method shows high-frequency artifacts on the isosurface which do not appear in the BCC method. In any case, it is obvious that both reconstruction methods show superior results than the Cartesian method. Figure 7 shows another reconstruction of ML dataset and their corresponding error images. The angular error, capped at 0.2 radians, is mapped to white and 0 error to black. Increased blackness in the FCC method implies increased accuracy.

(a)   (b)

(c)   (d)
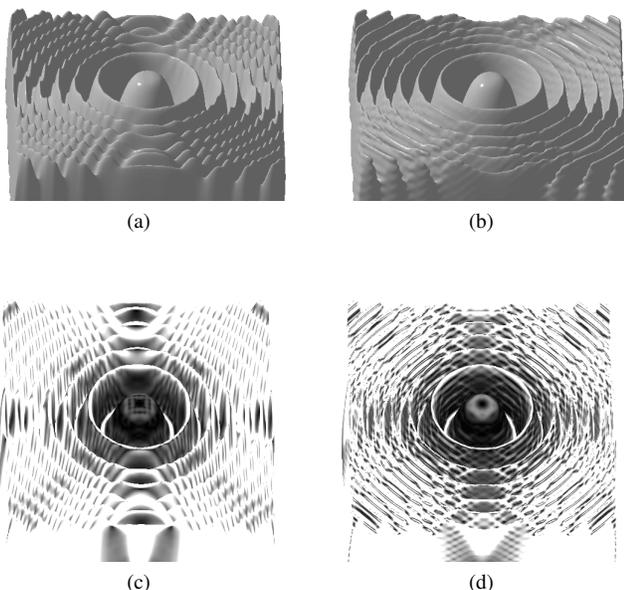
Fig. 7: Marschner-Lobb test images (see page 5). (a) (c) C^1 reconstruction on Cartesian with $41 \times 41 \times 41 = 68,921$ samples and (b) (d) C^1 reconstruction on FCC with $25 \times 25 \times 25 \times 4 = 62,500$ samples. (Right) the angular error, capped at 0.2 radians, is mapped to white and 0 error to black. Increased blackness from left to right implies increased accuracy.

For our experiments, we used the Carp fish dataset of original resolution of $256 \times 256 \times 512$ (density 1, see Figure 6a). This dataset was subsampled to $100 \times 100 \times 200$ resolution using a rational subsampling scheme. For this subsampling, a proper upsampling operation was applied by zero-padding in the frequency domain so that a downsampling operation produces the target resolution of 100. We performed the equivalent rational subsampling to a Cartesian volume of resolution $126 \times 126 \times 126$ to obtain the matching resolution on the FCC lattice. Then this Cartesian volume is simply downsampled onto a FCC lattice by retaining Cartesian points whose sum of coordinates are even, halving the number of samples. Hence the resulting FCC lattice is of resolution $63 \times 63 \times 126 \times 4$ which is very close to the Cartesian volume The images are rendered with (customized) POV-Ray (http://www.povray.org).

Figure 9 compares C^1 reconstruction on the FCC and Cartesian lattices (see Table 3). As can be seen, the FCC method captures the detailed features of the dataset better than the Cartesian method. Most

of the ribs in the Cartesian data set are disconnected while on the FCC data set are mostly connected. Also staircase artifacts on the Cartesian method are more severe (see, e.g., the ribs area of Figure 9). It is clear that the Cartesian method has difficulty to capture the features that are not axis-aligned.

6.1 Computational Cost

Another advantage of the FCC method is the efficient evaluation. Since the number of neighbor samples (stencil size) required to evaluate a point is small (16 compared to 27 for the Cartesian method), it takes less time to evaluate the reconstructed field if the memory access introduces relatively big overhead. This overhead becomes dominant, due to cache miss, as the dataset size grows. For example, while it takes 125 seconds to generate Figure 7a, it takes only 98 seconds (72%) to generate Figure 7b. These images were rendered at the resolution of 500×500 pixels. Table 2 compares the time for raycasting the images in this paper. These experiments were performed on a dual core Intel 2.13 GHz machine with 2 GB memory.

Dataset	Cartesian	FCC	Ratio
Marschner-Lobb	135	98	72%
Carp-Fish	515	358	69%

Table 2: Rendering time (in seconds) to generate ray-casted images in Figure 7.

6.2 Reconstruction Properties

Our proposed box spline reconstruction on the FCC lattice has the same approximation order and continuity as that of the triquadratic B-spline on the Cartesian lattice. A similar reconstruction on the BCC lattice is not presently known: smooth reconstruction is only available as a 8-direction box spline[13]. Table 3 summarizes the reconstruction properties of these schemes.

Smoothing and post-aliasing metrics have been proposed to compare reconstruction schemes [20]. In table 3, we show these metrics, numerically evaluated. While both smoothing and post-aliasing measures are relatively close, the FCC box spline filter has less smoothing artifacts but more post-aliasing artifacts compared to the triquadratic B-spline.

	C^1 on FCC	C^1 on $\mathbb{Z}^3$	C^2 on BCC
filter	M_6	M_{333}	8-dir. box spline
degree	3	6	5
approx. order	3	3	4
stencil size	16	27	32
smoothing	0.804381	0.841865	0.852867
post-aliasing	0.0114894	0.00825702	0.00401669

Table 3: Reconstruction properties of FCC box spline, Cartesian triquadratic B-spline, and the BCC box spline.

7 DISCUSSION AND EXTENSIONS

Despite their theoretical advantage, non-Cartesian lattices have only had a limited impact on practical applications. On one hand, there are no acquisition strategies that take advantage of the superior sampling on optimal non-Cartesian lattices; on the other hand, major signal processing tools, such as reconstruction, to process and analyze the resulting data have been missing. This paper addresses the second issue, and thereby motivates engineering research into the first, by providing an efficient reconstruction algorithm for data on the FCC lattice. To this end, we introduced the 6-direction box spline M_6, derived its simple
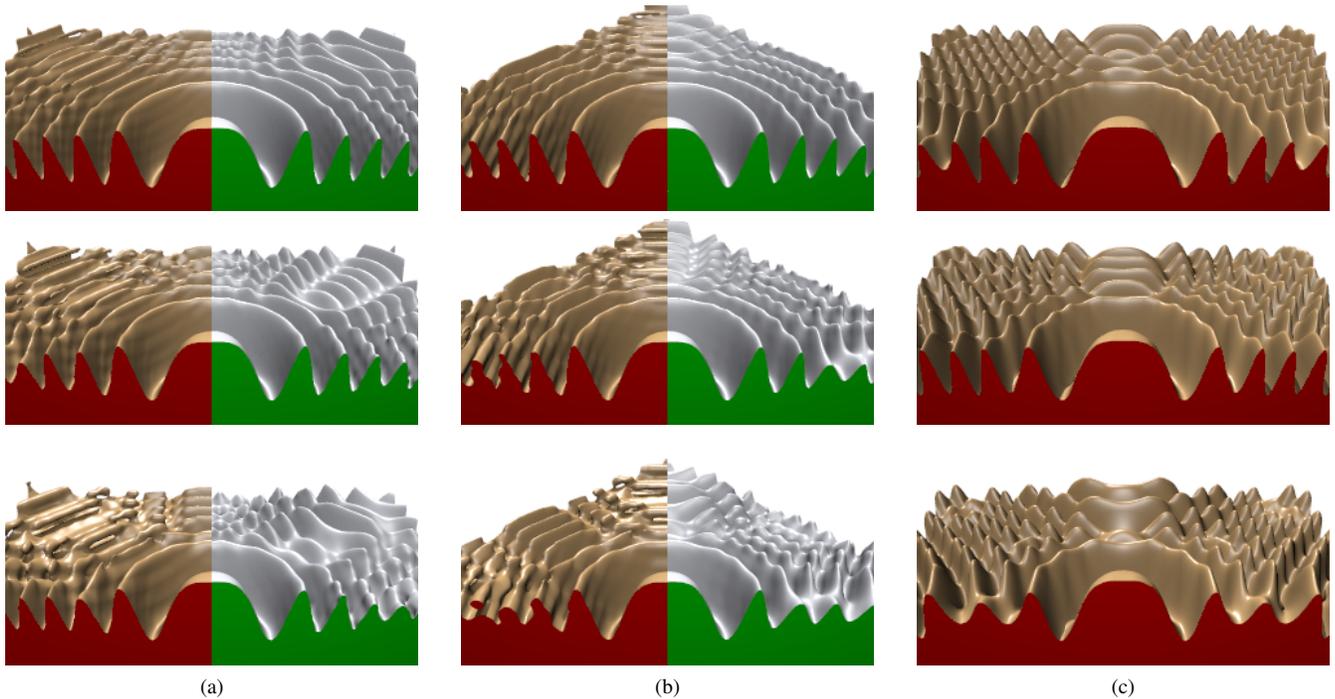
Fig. 8: Comparison of three reconstruction methods with sampling densities: top row 0.05^{-3}, middle row 0.06^{-3}, and bottom row 0.07^{-3}. (a) C^1 on FCC (copper with red cross section) and C^2 on BCC (silver with green cross section) juxtaposed. (b) the view point and slicing plan rotated by 45 degrees. (c) shows the corresponding C^1 on the Cartesian lattice.

quasi-interpolant formula and showed fast and exact evaluation via in piecewise polynomial form.

In this work, we have not yet taken full advantage of the low algebraic degree of the level sets of M_6. In principle, we can parametrize the isosurfaces of the trivariate cubic polynomial pieces of the spline based on M_6 [1] and then explicitly write out all ray intersections. (By comparison, the ray-isosurface intersection for M_{333} can generically only be computed numerically.) Our current implementation (i.e., with POV-Ray) accepts only the evaluator of an implicit function and not the exact location of the ray-level intersection, we did not exploit this low polynomial-degree property to expedite the ray-intersection operations.

REFERENCES

[1] C. L. Bajaj, R. L. Holt, and A. N. Netravali. Rational parametrizations of nonsingular real cubic surfaces. *ACM Trans. Graph.*, 17(1):1–31, 1998.

[2] I. Carlbom. Optimal Filter Design for Volume Reconstruction and Visualization. In *Proceedings of the IEEE Conference on Visualization*, pages 54–61, Oct. 1993.

[3] H. Carr, T. Möller, and J. Snoeyink. Artifacts Caused by Simplicial Subdivision. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):231–242, 2006.

[4] T. Cooklev. *Regular perfect-reconstruction filter banks and wavelet bases*. Ph.D. dissertation, Tokyo University of Technology, Tokyo, Japan, 1995.

[5] B. Csébfalvi. Prefiltered gaussian reconstruction for high-quality rendering of volumetric data sampled on a body-centered cubic grid. In *IEEE Visualization*, pages 311–318, 2005.

[6] C. de Boor. B-form basics. In *Geometric modeling*, pages 131–148. SIAM, Philadelphia, PA, 1987.

[7] C. de Boor. *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences*. Springer-Verlag, New York, revised edition, 2001.

[8] C. de Boor, K. Höllig, and S. Riemenschneider. *Box splines*, volume 98 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1993.

[9] S. C. Dutta Roy and B. Kumar. *Handbook of Statistics*, volume 10, chapter Digital Differentiators, pages 159–205. Elsevier Science Publishers B. V., North Holland, 1993.

[10] A. Entezari. *Optimal Sampling Lattices and Trivariate Box Splines*. PhD thesis, Simon Fraser University, Vancouver, Canada, July 2007.

[11] A. Entezari, R. Dyer, and T. Möller. Linear and Cubic Box Splines for the Body Centered Cubic Lattice. In *Proceedings of the IEEE Conference on Visualization*, pages 11–18, Oct. 2004.

[12] A. Entezari, T. Meng, S. Bergner, and T. Möller. A Granular Three Dimensional Multiresolution Transform. In *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization*, pages 267–274, May 2006.

[13] A. Entezari, D. Van De Ville, and T. Moller. Practical box splines for reconstruction on the body centered cubic lattice. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2):313–328, March-April 2008.

[14] G. Farin. *Curves and surfaces for computer-aided geometric design*. Computer Science and Scientific Computing. Academic Press Inc., San Diego, CA, fourth edition, 1997.

[15] R. B. Fuller. United States Patent Office # 2,986,241 OCTET TRUSS, 1961.

[16] R. G. Keys. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Tran. Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, Dec. 1981.

[17] J. Kovačević and M. Vetterli. FCO sampling of digital video using perfect reconstruction filter banks. *IEEE Trans. Image Proc.*, 2(1):118–122, January 1993.

[18] H. R. Künsch, E. Agrell, and F. A. Hamprecht. Optimal lattices for sampling. *IEEE Transactions on Information Theory*, 51(2):634–647, Feb. 2005.

[19] L. Linsen, V. Pascucci, M. A. Duchaineau, B. Hamann, and K. I. Joy. Hierarchical representation of time-varying volume data with "4th-root-of-2" subdivision and quadrilinear b-spline wavelets. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applica-*
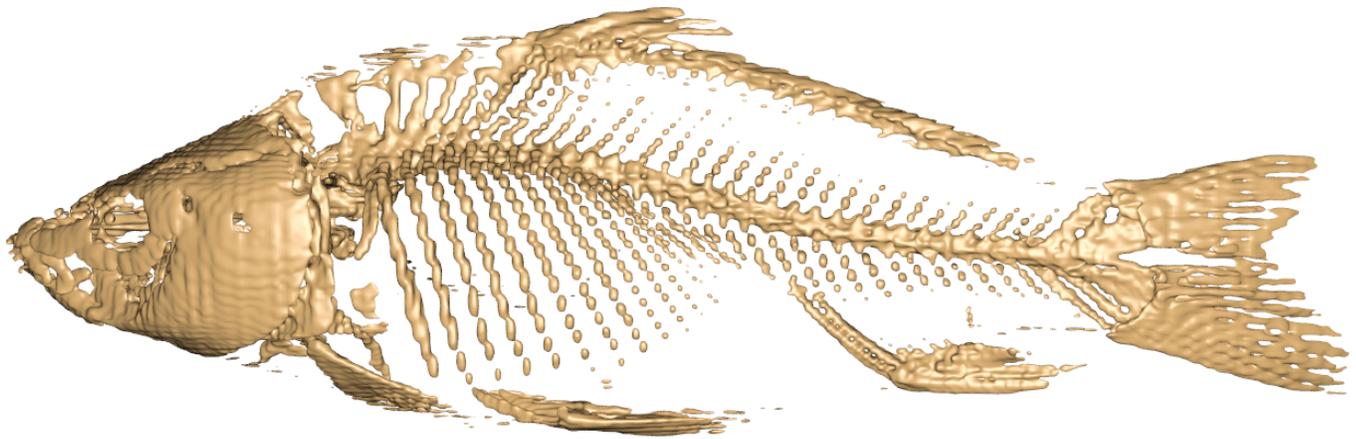
(a)
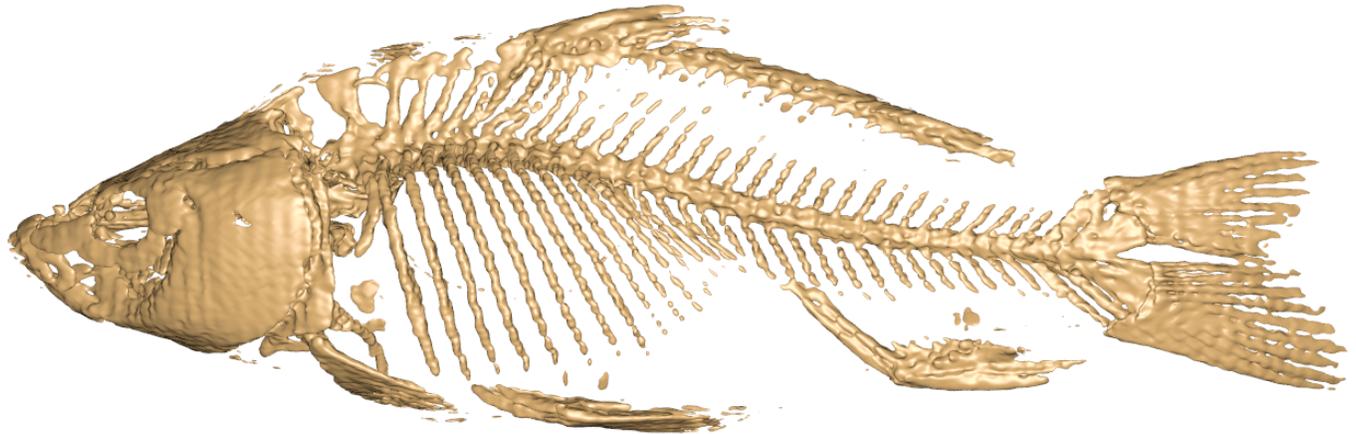
(b)

Fig. 9: Carp dataset at roughly 6% of original resolution. Reconstructed by (a) C^1 on Cartesian ($100 \times 100 \times 200 = 2,000,000$ resolution) and (b) C^1 on FCC ($63 \times 63 \times 126 \times 4 = 2,000,376$ resolution).

tions, page 346, Washington, DC, USA, 2002. IEEE Computer Society.

[20] 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, 1994.

[21] J. H. McClellan. The design of two-dimensional digital filters by transformations. In *Proc. 7th Annu. Princeton Conf. Information Sciences and Systems*, pages 247–251, Princeton, NJ, 1973.

[22] F. Meyer. Mathematical morphology: from two dimensions to three dimension s. *Journal of Microscopy*, 165(1):5–28, 1992.

[23] D. P. Mitchell and A. N. Netravali. Reconstruction Filters in Computer Graphics. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22, pages 221–228, Aug. 1988.

[24] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of Accurate and Smooth Filters for Function and Derivative Reconstruction. *Proceedings of the Symposium on Volume Visualization*, pages 143–151, Oct 1998.

[25] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Prentice Hall Inc., Englewoods Cliffs, NJ, 1989.

[26] D. P. Petersen and D. Middleton. Sampling and Reconstruction of Wave-Number-Limited Functions in N-Dimensional Euclidean Spaces. *Information and Control*, 5(4):279–323, Dec. 1962.

[27] W. Qiao, D. Ebert, A. Entezari, M. Korkusinski, and G. Klimeck. VolQD: Direct Volume Rendering of Multi-million Atom Quantum Dot Simulations. In *Proceedings of the IEEE Conference on Visualization*, pages 319–326, Oct. 2005.

[28] F. Qiu, F. Xu, Z. Fan, N. Neophytos, A. K. fman, and K. Mueller. Lattice-based volumetric global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1576–1583, 2007.

[29] T. Theußl, O. Mattausch, T. Möller, and E. Gröller. Reconstruction schemes for high quality raycasting of the body-centered cubic grid. *TR-186-2-02-11, Institute of Computer Graphics and Algorithms, Vienna University of Technology*, December 2002.

[30] T. Theußl, T. Möller, and E. Gröller. Optimal Regular Volume Sampling. In *Proceedings of the IEEE Conference on Visualization 2001*, pages 91–98, Oct 2001.

[31] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, July 2000.

[32] D. Van De Ville, T. Blu, and M. Unser. On the Multidimensional Extension of the Quincunx Subsampling Matrix. *IEEE Signal Processing Letters*, 12(2):112–115, February 2005.