Vision, Modeling, and Visualization (2010)

DC-Splines: Revisiting the Trilinear Interpolation on the Body-Centered Cubic Lattice

Balázs Domonkos[†1] and Balázs Csébfalvi[‡2]

[1]Mediso Medical Imaging Systems, Hungary
[2]Budapest University of Technology and Economics

Abstract

In this paper, we thoroughly study a trilinear interpolation scheme previously proposed for the Body-Centered Cubic (BCC) lattice. We think that, up to now, this technique has not received the attention that it deserves. By a frequency-domain analysis we show that it can isotropically suppress those aliasing spectra that contribute most to the postaliasing effect. Furthermore, we present an efficient GPU implementation, which requires only six trilinear texture fetches per sample. Overall, we demonstrate that the trilinear interpolation on the BCC lattice is competitive to the linear box-spline interpolation in terms of both efficiency and image quality. As a generalization to higher-order reconstruction, we introduce DC-splines that are constructed by convolving a Discrete filter with a Continuous filter, and easy to adapt to the Face-Centered Cubic (FCC) lattice as well.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.4.10 [Image Processing and Computer Vision]: Image Representation—Volumetric

1. Introduction

Recently, several reconstruction schemes have been proposed for the Body-Centered Cubic (BCC) [Ent07, EVDVM08, Csé08, Csé10] and Face-Centered Cubic (FCC) [QEE*05, KEP08] lattices. Both lattices show advantageous properties against the traditional Cartesian Cubic (CC) lattice. The BCC lattice, which is optimal for sampling spherically band-limited signals, requires around 30% fewer samples per unit volume than a CC lattice to represent the same amount of spatial information [TMG01, PM62]. For the suboptimal FCC lattice this ratio is 27% [KEP08]. Although the FCC lattice is less efficient for sampling volumetric signals than the BCC lattice, it is still used in specific applications, where an isotropic voxel neighborhood plays an important role [QXF*07, PQF*09, AEM09]. On the CC, BCC, and FCC lattices each voxel has 6, 8, and 12 nearest neighbors, respectively. Thus, the FCC lattice can represent the voxel neighborhood with the least direction dependence. This property is favorable in lattice-based global illumination calculation [QXF*07] or flow simulation [PQF*09, AEM09].

An interesting aspect of non-Cartesian regular sampling is how to reconstruct the original continuous signal from the discrete samples. Though the reconstruction can be implemented by a simple convolution, the choice of the filter kernel has a direct impact on both numerical accuracy and visual quality. Generally, an appropriate reconstruction filter has to fulfill the following criteria:

1. The filter should be invariant for rotations around the major axes by 90 degrees. Otherwise, the reconstruction would strongly depend on the orientation of the filter.
2. The filter should preferably be interpolating or quasi-interpolating of high order.
3. The smoothing effect should be minimal to preserve the high-frequency details.
4. The postaliasing effect should be minimal to avoid annoying artifacts.
5. The filter should be efficient to evaluate.

[†] e-mail: balazs.domonkos@mediso.hu
[‡] e-mail: cseb@iit.bme.hu

DOI: 10.2312/PE/VMV/VMV10/275-282

In this paper, we mainly focus on linear filters for two reasons. First, in time-critical applications, like direct volume rendering, real-time frame rates are hard to guarantee if computationally expensive higher-order filters are used for resampling. Second, to avoid severe artifacts, we assume that the data is relatively oversampled, which compensates the higher postaliasing effect of a linear filter. This is a reasonable compromise between image quality, storage requirements, and rendering speed.

For the BCC lattice, there are five known linear reconstruction schemes, which are sheared trilinear interpolation [TMMG02,Mat03], linear box-spline filtering [EDM04, EVDVM08], BCC-spline filtering [Csé08], trilinear B-spline filtering [CH06, Csé10], and BCC trilinear interpolation [TMMG02, Mat03]. All these filters guarantee the same polynomial approximation order of two, that is, they can perfectly reproduce linear polynomials.

The *sheared trilinear interpolation* [TMMG02, Mat03] exploits that the BCC lattice can be interpreted as a sheared CC lattice; thus, a simple trilinear interpolation can be performed inside the parallelepiped-shaped cells. This scheme is equivalent to the convolution of the BCC samples with an anisotropic sheared trilinear filter. However, this filter does not fulfill the first quality criterion; therefore, it leads to a direction-dependent reconstruction.

The *linear box spline* [EDM04,EVDVM08] is directly tailored to the geometry of the BCC lattice. Among the linear filters it has the most compact support. In fact, the linear box-spline filtering is equivalent to a linear interpolation inside the tetrahedral cells of the BCC lattice. Therefore, the filter kernel covers only four BCC samples. The shape of the filter is a rhombic dodecahedron, which has sharp peaks at the vertices. That is the reason why the linear box spline causes rivet-like artifacts on the reconstructed isosurfaces.

Similarly to the linear box spline, a *linear BCC-spline* [Csé08] also represents a non-separable filter tailored to the BCC lattice. The linear BCC-spline is constructed by convolving the BCC nearest-neighbor kernel with itself. The shape of the obtained filter inherits the shape of the nearest-neighbor kernel, which is a truncated octahedron (the Voronoi cell of the BCC lattice). Although the linear BCC-spline takes the eight nearest voxels into account just like the trilinear interpolation on the CC lattice, an efficient analytic evaluation of the kernel itself is not known yet. Therefore, previously a discrete approximation calculated in the frequency domain has been applied. This solution, however, is not favorable in a GPU implementation, since the filter kernel is represented by a 3D lookup table taking space from the limited texture memory.

On the BCC lattice, a *trilinear B-spline filtering* [CH06, Csé10] can be easily performed exploiting that the BCC lattice consists of two interleaved CC lattices. A simple trilinear interpolation is evaluated separately on these CC lattices, and then the contributions are averaged in the given sample position. This evaluation is equivalent to the convolution of the BCC samples with a trilinear B-spline kernel. Unlike the other linear filtering schemes, the BCC trilinear B-spline reconstruction is not an interpolation. It cannot be used even for generalized interpolation on the BCC lattice, since a discrete prefilter that would make it interpolating does not exist [CH06]. On the other hand, after an appropriate discrete prefiltering, it can be applied for quasi-interpolation of order two [Csé10]. Nevertheless, the trilinear B-spline is not a Riesz basis on the BCC lattice [FEVDVM10]; therefore, different set of coefficients can represent the same function in the shift-invariant function space generated by the trilinear B-spline. This might be problematic in specific applications, where a unique solution is searched for during an optimization process like an iterative tomographic reconstruction.

Note that the *trilinear interpolation* [TMMG02, Mat03] proposed for the BCC lattice is not the same as the non-interpolating BCC trilinear B-spline reconstruction [CH06, Csé10]. This technique is theoretically equivalent to the discrete upsampling of the BCC-sampled volume on a higher-resolution CC lattice, where the standard trilinear interpolation is used for resampling. In practice, however, the missing CC samples are calculated on the fly and not in a preprocessing. So far, the BCC trilinear interpolation has not been sufficiently evaluated (e.g., its impulse and frequency responses have not been derived for a theoretical analysis and its practical GPU implementation has not been considered either) and compared to the other linear filtering schemes. Therefore, in our opinion, it has not received the attention that it deserves.

The contributions of this paper are the following:

- We thoroughly analyze the smoothing and postaliasing effects of the BCC trilinear interpolation by analytically deriving its frequency response. We demonstrate that this reconstruction scheme shows almost the same advantageous properties as the linear box-spline interpolation, moreover, its postaliasing effect is even lower and more isotropic.
- We propose an efficient GPU acceleration that requires only six trilinear texture fetches per sample. Our implementation is slightly faster than a recently published GPU-based linear box-spline reconstruction [FEVDVM10].
- We show that the impulse response of the trilinear interpolation on the BCC lattice is, in fact, a convolution of a discrete filter and a continuous filter. This scheme can be generalized to the FCC lattice as well. The discrete filter is used to reconstruct the missing CC samples and a simple trilinear interpolation is applied on the higher-resolution CC representation. We call the obtained Discrete/Continuous filters as DC-splines. Similarly to the B-splines and box splines, higher-order DC splines can be obtained by convolving the linear DC-splines with themselves.

2. Trilinear Interpolation on the BCC Lattice

The BCC lattice can be obtained from a CC lattice such that all CC lattice points are removed where the discrete coordinates i, j, and k are neither all even nor all odd numbers. The BCC trilinear interpolation reproduces these "missing CC samples" by interpolating between the available BCC samples. The BCC lattice can also be interpreted as two interleaved CC lattices. Figure 1 shows two cubic cells of these interleaved CC lattices that intersect each other in a smaller cubic cell depicted in green. The green cubic cell has only two corners that are BCC lattice points, the others (the green dots) need to be interpolated from two BCC samples along either red or blue edges. Afterwards, a simple trilinear interpolation is performed inside the green cell.

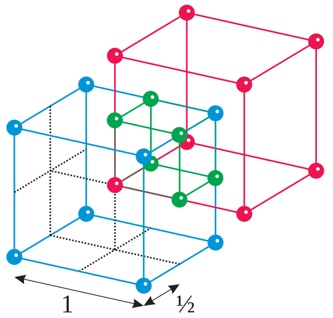


Figure 1: *Trilinear interpolation on the BCC lattice: The BCC lattice points are located on the interleaved red and blue CC lattices. The green samples are interpolated from two BCC samples of the same color. Inside the green cubic cell a standard trilinear interpolation is applied.*

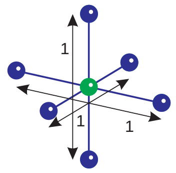
The green samples are, in fact, reconstructed using a discrete filter on the BCC lattice. This filter is shown in Figure 2. The resultant impulse response χ^1_{bcc} of the BCC trilinear interpolation is obtained by convolving this discrete filter with a scaled trilinear kernel $\beta^1(2\boldsymbol{x})$:

$$\chi^1_{\mathrm{bcc}}(\boldsymbol{x}) = \beta^1(2\boldsymbol{x}) + \frac{1}{2}\sum_{k=1}^{6}\beta^1(2(\boldsymbol{x}-\boldsymbol{v}_k)), \tag{1}$$

where

$$[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6] = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

Note that χ^1_{bcc} is isotropic, that is, it satisfies the first quality criterion. Concerning the second criterion, it is easy to see that χ^1_{bcc} is interpolating and also quasi-interpolating of order two on the BCC lattice, moreover, it satisfies the Riesz conditions [Uns00]. In this sense, the BCC trilinear interpolation is equivalent to the linear box-spline reconstruction, which provides the same polynomial approximation order. The third and fourth criteria will be studied in Section 3.1, while the fifth criterion will be evaluated in Section 2.1.



	BCC	FCC
🟢	1	1
🔵	1/2	1/6

Figure 2: *The discrete components of the trilinear filters for the BCC and FCC lattices. These discrete filters are responsible for producing the missing CC samples in an upsampled CC representation.*

2.1. GPU Implementation

To efficiently implement the trilinear interpolation on the BCC lattice, we store the BCC-sampled volume in an interleaved manner such that a 3D texture map contains two data values per texel. The first and second data values represent the blue and red CC lattice points, respectively (see Figure 1). The key idea is to exploit the hardware-accelerated trilinear texture fetching for calculating the missing CC samples. Figure 3 illustrates our GPU implementation. Assume that the given sample to be interpolated is located inside the black square defined by corners $q_{0,0}$, $q_{1,0}$, $q_{0,1}$, and $q_{1,1}$. After having the density values at these corner points determined, a simple bilinear interpolation needs to be applied. For the sake of clarity, we introduce a linear interpolation operator $L(a,b,t) = (1-t)a + tb$. Our algorithm performs the following calculation for the corner points q:

operation	implementation
$\tilde{f}(\boldsymbol{p}_{1,0,0}) = L(f(\boldsymbol{p}_{0,0,0}), f(\boldsymbol{p}_{2,0,0}), 1/2)$	trilinear fetch
$\tilde{f}(\boldsymbol{p}_{1,1,0}) = L(f(\boldsymbol{p}_{1,1,-1}), f(\boldsymbol{p}_{1,1,1}), 1/2)$	trilinear fetch
$\tilde{f}(\boldsymbol{q}_{1,0}) = L(\tilde{f}(\boldsymbol{p}_{1,0,0}), \tilde{f}(\boldsymbol{p}_{1,1,0}), 2t)$	lerp instruction
$\tilde{f}(\boldsymbol{p}_{0,0,1}) = L(f(\boldsymbol{p}_{0,0,0}), f(\boldsymbol{p}_{0,0,2}), 1/2)$	trilinear fetch
$\tilde{f}(\boldsymbol{p}_{0,1,1}) = L(f(\boldsymbol{p}_{-1,1,1}), f(\boldsymbol{p}_{1,1,1}), 1/2)$	trilinear fetch
$\tilde{f}(\boldsymbol{q}_{0,1}) = L(\tilde{f}(\boldsymbol{p}_{0,0,1}), \tilde{f}(\boldsymbol{p}_{0,1,1}), 2t)$	lerp instruction
$\tilde{f}(\boldsymbol{q}_{0,0}) = L(f(\boldsymbol{p}_{0,0,0}), f(\boldsymbol{p}_{0,2,0}), t)$	trilinear fetch
$\tilde{f}(\boldsymbol{q}_{1,1}) = L(f(\boldsymbol{p}_{1,-1,1}), f(\boldsymbol{p}_{1,1,1}), t+1/2)$	trilinear fetch

Note that the intermediate samples at $q_{0,0}$ $q_{1,1}$ can be calculated by a single trilinear texture fetch instead of two; therefore, the total number of the trilinear texture fetches is six. The bilinear interpolation between the corner points q is implemented using *lerp* instructions. Although the linear box-spline interpolation requires only four nearest-neighbor texture fetches, its indexing overhead is much higher. Furthermore, due to the hardware-accelerated trilinear interpolation, the cost of a trilinear texture fetch is approximately the same as that of a nearest-neighbor texture fetch [SH05].

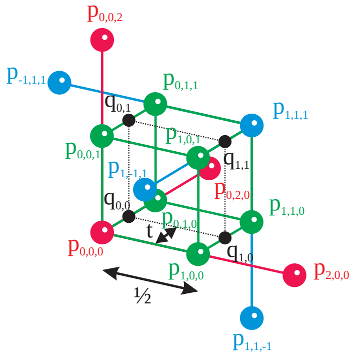


Figure 3: *Optimized trilinear interpolation on the BCC lattice. The missing CC samples (green dots) are calculated by a GPU-accelerated trilinear sampling on either the red or the blue CC lattice.*

data set	engine	brain
resolution	128 × 128 × 83 × 2	128 × 128 × 55 × 2
linear box spline	5.33 fps	3.28 fps
BCC trilinear	5.73 fps	3.37 fps

Table 1: *Frame rates of GPU-accelerated texture-based isosurface rendering of BCC-sampled volume data using linear box-spline interpolation and BCC trilinear interpolation.*

To compare the BCC trilinear interpolation to the linear box-spline interpolation in terms of efficiency, we implemented a texture-based isosurface-rendering application. The frame rates are shown in Table 1. Despite that the BCC trilinear interpolation is theoretically more expensive, its practical GPU implementation is still slightly faster than a linear box-spline filtering optimized for the GPU [FEVDVM10]. In the following, we will investigate whether the BCC trilinear interpolation is also competitive in terms of image quality.

3. Comparison to the Linear Box Spline

The BCC trilinear kernel χ^1_{bcc} is mostly comparable to the linear box spline as both of them are interpolating, form a Riesz basis on the BCC lattice, and provide approximately the same efficiency in a GPU implementation. Concerning the image quality, it is worthwhile to analyze their frequency-domain characteristics and evaluate their performance on volumetric test data.

3.1. Frequency-Domain Analysis

Since χ^1_{bcc} is constructed as a convolution of a discrete filter and a continuous filter, its Fourier transform $\hat{\chi}^1_{\mathrm{bcc}}$ is easy to derive by multiplying the Fourier transforms of these two components:

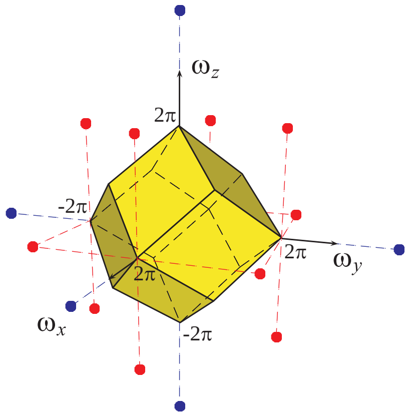


Figure 4: *The pass band corresponding to BCC sampling is the Voronoi cell of the dual FCC lattice, which is a rhombic dodecahedron. The red dots depict the twelve first nearest neighbors, whereas the blue dots depict the six second nearest neighbors.*

$$\hat{\chi}^1_{\mathrm{bcc}}(\boldsymbol{\omega}) = \left[1 + cos\left(\frac{\omega_x}{2}\right) + cos\left(\frac{\omega_y}{2}\right) + cos\left(\frac{\omega_z}{2}\right)\right]\hat{\beta}^1\left(\frac{\boldsymbol{\omega}}{2}\right), \qquad (2)$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$.

According to the well-known Strang-Fix conditions [SF71], the approximation power of a filter is N if its frequency response guarantees zero-crossings of at least order N at the points of the dual lattice except the origin. Due to the chessboard property, the dual FCC lattice points can be defined as $\boldsymbol{\omega} = 2\pi[u,v,w]^T$, where $u+v+w$ is even. Note that $\hat{\chi}^1_{\mathrm{bcc}}(\boldsymbol{\omega})$ has zero-crossings of at least order two at these points; therefore, the approximation power of χ^1_{bcc} is indeed equivalent to that of the linear box spline. Since χ^1_{bcc} is interpolating, it fully exploits its approximation power even without prefiltering [CBU05, CVDV07]. Thus, the polynomial approximation order of the BCC trilinear interpolation is two, that is, it can perfectly reproduce linear polynomials.

If the frequency response of a filter is significantly nonzero at the dual lattice points, which represent the "DC" components of the aliasing spectra, a *sample-frequency ripple* might occur [ML94]. This is not the case for $\hat{\chi}^1_{\mathrm{bcc}}$ as it satisfies the Strang-Fix conditions. However, to avoid even a *near-sample-frequency ripple* [ML94], the frequency response should preferably be closed to zero even in the vicinity of the dual lattice points except the origin. Note that $\hat{\chi}^1_{\mathrm{bcc}}$ fulfills also this requirement as it guarantees zero-crossings of at least order two at these points of the frequency domain. Furthermore, based on the theory developed in [EM06], $\hat{\chi}^1_{\mathrm{bcc}}$ isotropically suppresses the nearest aliasing spectra that contribute most to the postaliasing effect, since the order of its zero-crossings (the number of its *vanishing* *moments* [EVDVM08]) is two for both the nearest and the second nearest FCC lattice points. As shown in Figure 4, the nearest FCC lattice points (the red dots) are located along the diagonal directions at $[\pm 2\pi, \pm 2\pi, 0]$, $[\pm 2\pi, 0, \pm 2\pi]$, and $[0, \pm 2\pi, \pm 2\pi]$, whereas the second nearest FCC lattice points (the blue dots) are located along the major axes at $[\pm 4\pi, 0, 0]$, $[0, \pm 4\pi, 0]$, and $[0, 0, \pm 4\pi]$. The antialiasing effect of the linear box spline [EVDVM08] is not so isotropic as that of the BCC trilinear interpolation, since its frequency response has vanishing moments of two and four at the nearest and the second nearest FCC lattice points [CD10]. Consequently, the linear box spline suppresses the postaliasing effect stronger along the major axis than along the diagonal directions. This is confirmed by Figure 5 as well, which shows cross-sectional slices of the frequency responses.

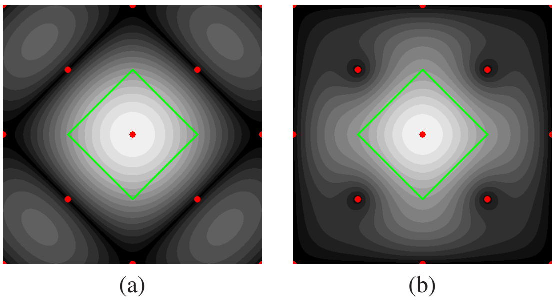


Figure 5: *Frequency responses of the linear box spline (a) and the BCC trilinear interpolation (b). The images represent the central slice* ($\omega_z = 0$) *of the frequency response in the domain* $[-4\pi, 4\pi]^2$*, where the green rhombus depicts the border of the pass band and the red dots depict the points of the dual FCC lattice.*

In order to analyze the smoothing and postaliasing effects, we visualized the frequency responses also in 3D by using direct volume rendering as it has been proposed in [CD10]. Figure 6 shows the frequency responses separately in the pass band and in the stop band. Note that the linear box-spline interpolation performs better in the pass band, while the BCC trilinear interpolation is better in the stop band, as its aliasing frequencies are mainly concentrated inside a sphere of a smaller radius. Thus, the BCC trilinear interpolation introduces less high-frequency postaliasing, but its smoothing effect is slightly higher.

3.2. Empirical Analysis

To visually compare the BCC trilinear interpolation to the linear box-spline interpolation, we rendered synthetic and measured test data. Figure 7 shows the isosurfaces of the Marschner-Lobb signal reconstructed from $64 \times 64 \times 64 \times 2$ BCC samples. These results are consistent with our frequency-domain analysis. The linear box spline introduces oscillation especially along the diagonal directions, where the postaliasing effect is suppressed less than along the major axes. In contrast, the BCC trilinear interpolation suppresses the nearest aliasing spectra more isotropically; therefore, it can better reconstruct the circular shape of the rings. Between the angular gradient errors, however, there is no significant difference.

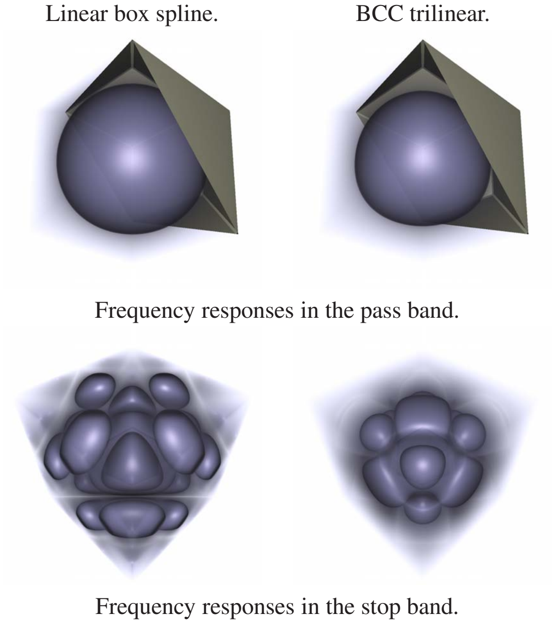
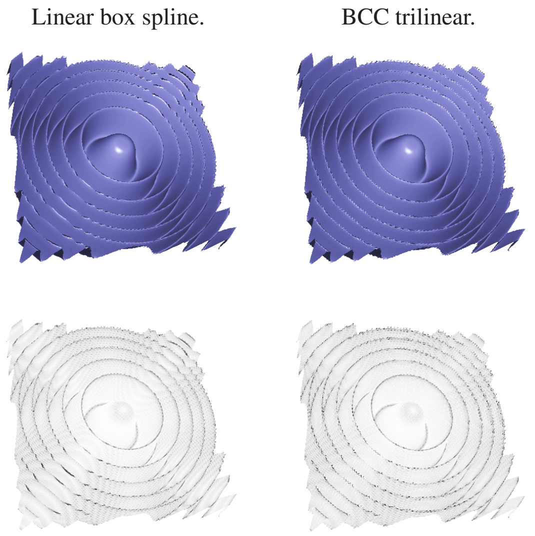

Linear box spline.
BCC trilinear.
Frequency responses in the pass band.
Frequency responses in the stop band.

Figure 6: *Frequency responses of the linear box-spline interpolation and the BCC trilinear interpolation. The boundary of the pass band, which is a rhombic dodecahedron, is cut by a half plane to show the frequency responses inside.*

We also rendered real-world measured data sets downsampled to BCC lattices (see Figure 8). Although the BCC trilinear interpolation reduces the postaliasing artifacts slightly better, the nature of the artifacts is rather similar as in case of a linear box-spline filtering. This is not surprising as the impulse response of the BCC trilinear interpolation approximates the rhombic dodecahedral shape of the linear box spline.

4. DC-Splines

Similarly to the B-spline, BCC-spline, and box-spline families of filters, the impulse response χ^1_{bcc} of the BCC trilinear interpolation can also be used for generating higher-order filters by consecutively convolving χ^1_{bcc} with itself. As χ^1_{bcc} can be factorized to a discrete filter and a continuous filter, the obtained higher-order filters can also be factorized in this way due to the associative property of the convolution operator. Therefore, we call the filter family generated by χ^1_{bcc} as DC-splines (DC is the abbreviation of Discrete/Continuous). A BCC DC-spline of odd order n is defined by a recursive


Linear box spline.

BCC trilinear.

Figure 7: *Reconstruction of the Marschner-Lobb signal form* 64 × 64 × 64 × 2 *BCC samples using linear box-spline interpolation and BCC trilinear interpolation. In the error images angular gradient error of 30 degrees is mapped to black, whereas angular error of zero degree is mapped to white.*

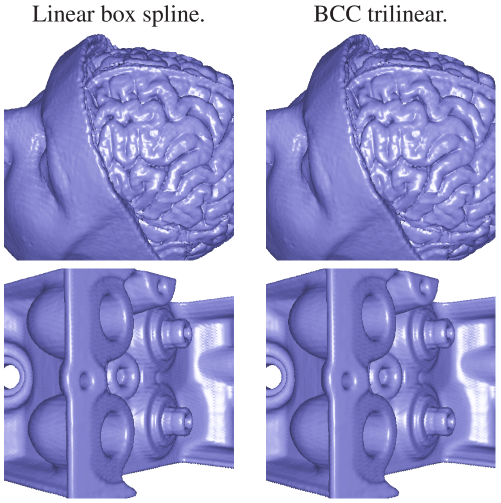
formula:

$$\chi^{n}_{\text{bcc}} = \chi^{n-2}_{\text{bcc}} * \chi^{1}_{\text{bcc}}. \tag{3}$$

From an approximation theory point of view, the cubic DC-spline χ^3_{bcc} is equivalent to the quintic box spline as it ensures the same approximation power. Both filters are non-interpolating; therefore, a discrete prefiltering is necessary to use them for interpolation. More concretely, applying the principle of *generalized interpolation* [BTU99, CD08], the coefficients of the shifted kernels are determined such that the interpolation condition is satisfied. Additionally, the prefiltering fully exploits the approximation power of the reconstruction filters, so the prefiltered cubic DC-spline and quintic box-spline reconstructions lead to quasi-interpolation [CBU05, CVDV07, CD08, Csé10] of order four, that is, they can perfectly reproduce polynomials of at most third degree.

Figure 9 shows the comparison of the cubic DC-spline to the quintic box spline. Note that, without prefiltering, both filters blur the high-frequency details. Although the cubic DC-spline can reproduce the rings more isotropically, its smoothing effect is slightly stronger. However, in case of prefiltering, the results are rather similar.

In Figure 10, the frequency responses are compared. The prefiltering significantly improves the pass-band behavior of both filters. The quintic box spline can better approximate the ideal low-pass filtering inside the pass band, but the cubic DC-spline guarantees a more isotropic suppression of the aliasing frequencies around the nearest and the second nearest FCC lattice points. The quintic box spline clearly suppresses the aliasing frequencies stronger along the major axis than along the diagonal directions.


Linear box spline.

BCC trilinear.

Figure 8: *Visualization of BCC-sampled measured data sets using the linear box-spline interpolation and the BCC trilinear interpolation.*

5. Generalization to the FCC Lattice

The DC-splines can be easily adapted to the FCC lattice as well. The discrete component of the FCC linear DC-spline shown in Figure 2 is responsible for reproducing the missing CC samples. The resultant impulse response of the FCC linear DC-spline is

$$\chi^{1}_{\text{fcc}}(\boldsymbol{x}) = \beta^{1}(2\boldsymbol{x}) + \frac{1}{6}\sum_{k=1}^{6}\beta^{1}(2(\boldsymbol{x}-\boldsymbol{v}_k)). \tag{4}$$

The higher-order DC-splines for the FCC lattice are generated similarly as for the BCC lattice.

6. Conclusion and Future Work

In this paper, we have demonstrated that the BCC trilinear interpolation is competitive to the linear box-spline interpolation from all the important practical and theoretical aspects. We proposed an efficient GPU implementation that is slightly faster than that of the linear box-spline interpolation. According to our thorough frequency-domain analysis, the BCC trilinear interpolation performs better in the stop band, but worse in the pass band. As a generalization to higher-order reconstruction, we introduced DC-splines for both BCC and FCC lattices. Overall, compared to the box splines, the DC-splines represent a simpler and more robust reconstruction scheme with comparable capabilities.

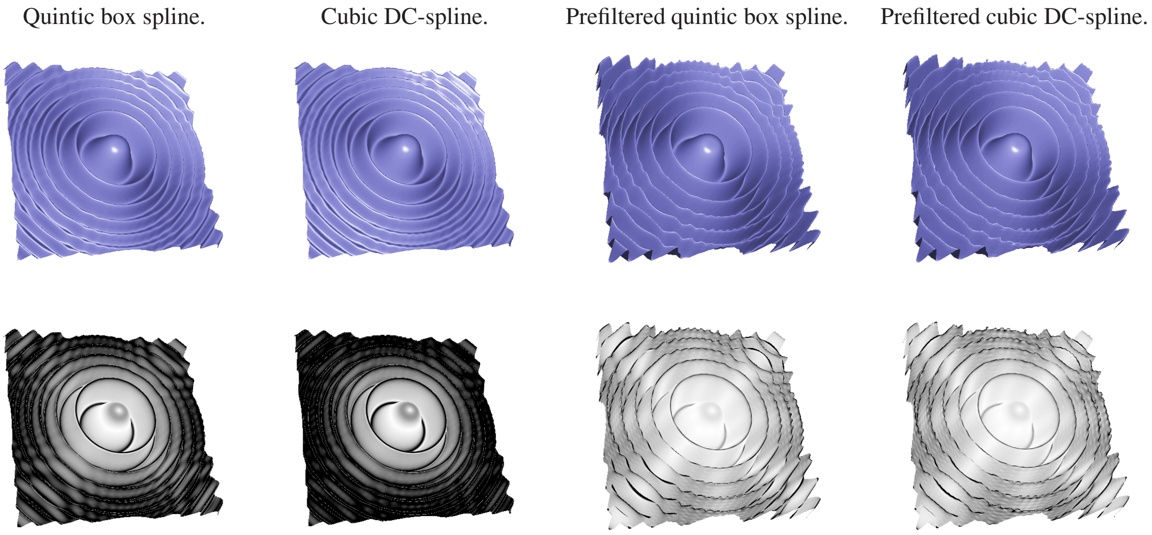


Figure 9: *Reconstruction of the Marschner-Lobb signal form* $32 \times 32 \times 32 \times 2$ *BCC samples using the quintic box spline and the cubic DC-spline. In case of prefiltering the reconstruction is interpolating. In the error images angular gradient error of 30 degrees is mapped to black, whereas angular error of zero degree is mapped to white.*

In our future work, we plan to efficiently implement the cubic DC-spline reconstruction on the GPU similarly as it has been proposed for the quintic box-spline reconstruction.

Acknowledgements

This work was supported by Mediso Medical Imaging Systems, the Hungarian National Office for Research and Technology (TECH 08/A2), the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and OTKA (F68945). The second author of this paper is a grantee of the János Bolyai Scholarship.

References

[AEM09] Alim U. R., Entezari A., Möller T.: The lattice-boltzmann method on optimal sampling lattices. *IEEE Transactions on Visualization and Computer Graphics 15*, 4 (2009), 630–641.

[BTU99] Blu T., Thévenaz P., Unser M.: Generalized interpolation: Higher quality at no additional cost. In *Proceedings of IEEE International Conference on Image Processing* (1999), pp. 667–671.

[CBU05] Condat L., Blu T., Unser M.: Beyond interpolation: Optimal reconstruction by quasi-interpolation. In *Proceedings of the IEEE International Conference on Image Processing* (2005), pp. 33–36.

[CD08] Csébfalvi B., Domonkos B.: Pass-band optimal reconstruction on the body-centered cubic lattice. In *Proceedings of Vision, Modeling, and Visualization* (2008), pp. 71–80.

[CD10] Csébfalvi B., Domonkos B.: 3D frequency-domain analysis of non-separable reconstruction schemes by using direct volume rendering. In *Proceedings of Spring Conference on Computer Graphics (to appear)* (2010).

[CH06] Csébfalvi B., Hadwiger M.: Prefiltered B-spline reconstruction for hardware-accelerated rendering of optimally sampled volumetric data. In *Proceedings of Vision, Modeling, and Visualization* (2006), pp. 325–332.

[Csé08] Csébfalvi B.: BCC-splines: Generalization of B-splines for the body-centered cubic lattice. *Journal of WSCG 16*, 1–3 (2008), 81–88.

[Csé10] Csébfalvi B.: 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 (2010), 499–512.

[CVDV07] Condat L., Van De Ville D.: Quasi-interpolating spline models for hexagonally-sampled data. *IEEE Transactions on Image Processing 16*, 5 (2007), 1195–1206.

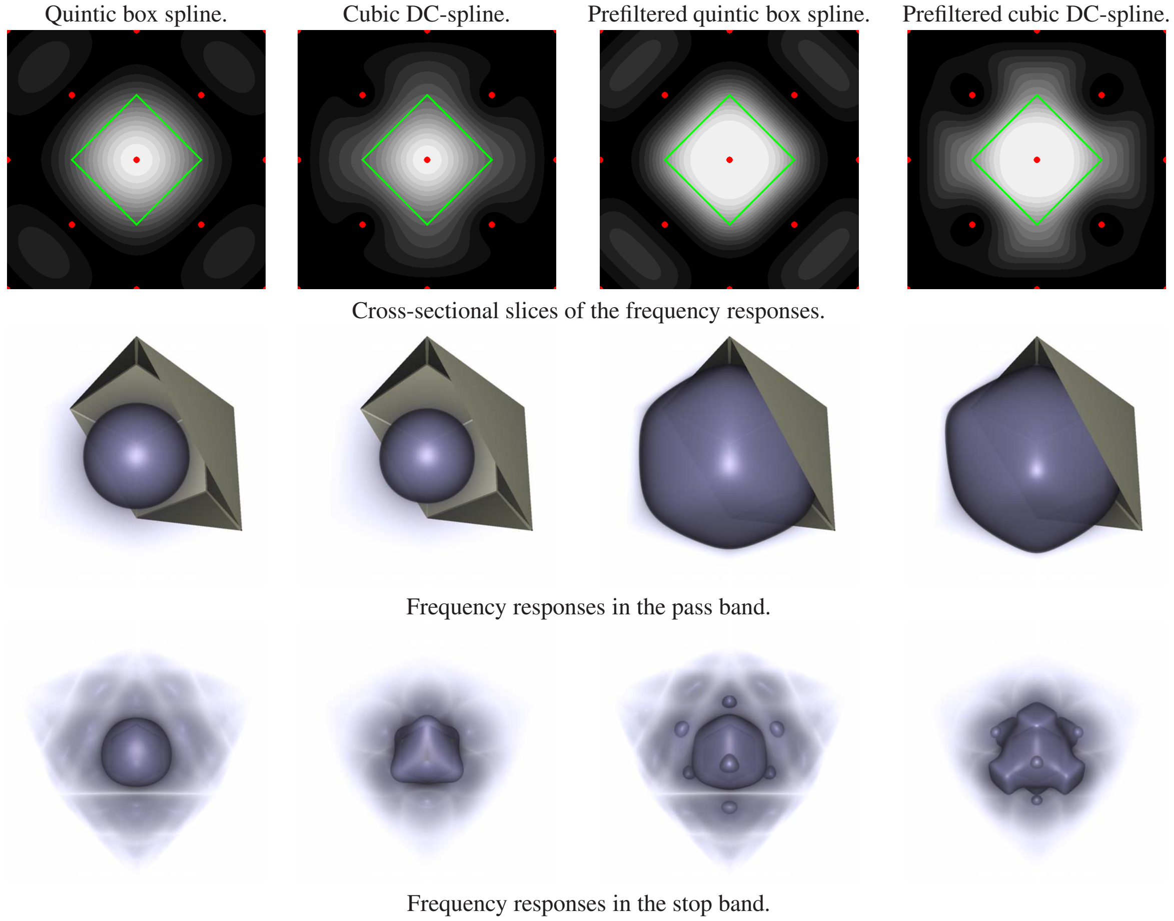
[EDM04] Entezari A., Dyer R., Möller T.: Linear and cubic box splines for the body centered cubic lattice. In *Proceedings of IEEE Visualization* (2004), pp. 11–18.

[EM06] Entezari A., Möller T.: Extensions of the Zwart-Powell box spline for volumetric data reconstruction on the Cartesian lattice. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization) 12*, 5 (2006), 1337–1344.

[Ent07] Entezari A.: Optimal sampling lattices and trivariate box splines. *PhD thesis, Simon Fraser University, Vancouver, Canada* (2007).

[EVDVM08] Entezari A., Van De Ville D., Möller T.: Practical box splines for reconstruction on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 313–328.

[FEVDVM10] Finkbeiner B., Entezari A., Van De Ville D., Möller T.: Efficient volume rendering on the body centered cubic lattice using box splines. *to appear in Computers & Graphics* (2010).



Quintic box spline. Cubic DC-spline. Prefiltered quintic box spline. Prefiltered cubic DC-spline.

Cross-sectional slices of the frequency responses.

Frequency responses in the pass band.

Frequency responses in the stop band.

Figure 10: *Frequency responses of the quintic box spline and the cubic DC-spline with and without prefiltering for generalized interpolation.*

[KEP08] Kim M., Entezari A., Peters J.: Box spline reconstruction on the face centered cubic lattice. In *Proceedings of IEEE Visualization* (2008), pp. 1523–1530.

[Mat03] Mattausch O.: Practical reconstruction schemes and hardware-accelerated direct volume rendering on body-centered cubic grids. *Master's Thesis, Vienna University of Technology* (2003).

[ML94] Marschner S., Lobb R.: An evaluation of reconstruction filters for volume rendering. In *Proceedings of IEEE Visualization* (1994), pp. 100–107.

[PM62] Petersen D. P., Middleton D.: Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces. *Information and Control 5*, 4 (1962), 279–323.

[PQF*09] Petkov K., Qiu F., Fan Z., Kaufman A., Mueller K.: Efficient LBM visual simulation on face-centered cubic lattices. *IEEE Transactions on Visualization and Computer Graphics 15*, 5 (2009), 802–814.

[QEE*05] Quiao W., Ebert D. S., Entezari A., Korkusinski M., Klimeck G.: VolQD: Direct volume rendering of multi-million atom quantum dot simulations. In *Proceedings of IEEE Visualization* (2005), pp. 319–326.

[QXF*07] Qiu F., Xu F., Fan Z., Neophytou N., Kaufman A., Mueller K.: Lattice-based volumetric global illumination. In *Proceedings of IEEE Visualization* (2007), pp. 1576–1583.

[SF71] Strang G., Fix G.: A Fourier analysis of the finite element variational method. In *Constructive Aspects of Functional Analysis* (1971), pp. 796–830.

[SH05] Sigg C., Hadwiger M.: Fast third-order texture filtering. In *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation* (2005), Matt Pharr (ed.), Addison-Wesley, pp. 313–329.

[TMG01] Theussl T., Möller T., Gröller M. E.: Optimal regular volume sampling. In *Proceedings of IEEE Visualization* (2001), pp. 91–98.

[TMMG02] Theussl T., Mattausch O., Möller T., Gröller M. E.: 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* (2002).

[Uns00] Unser M.: Sampling - 50 years after Shannon. *Proceedings of the IEEE 88*, 4 (2000), 569–587.