OPTIMAL SAMPLING LATTICES AND TRIVARIATE BOX SPLINES

by

Alireza Entezari
B.Sc., Simon Fraser University, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
2007

APPROVAL

Name: Alireza Entezari

Degree: Doctor of Philosophy

Title of thesis: Optimal Sampling Lattices and Trivariate Box Splines

Examining Committee: Dr. Arthur Kirkpatrick
Chair

Dr. Torsten Möller, Senior Supervisor
Associate Professor of Computing Science

Dr. Carl de Boor, Supervisor
Emeritus Professor of Computer Sciences and
Mathematics, University of Wisconsin-Madison

Dr. David Fracchia, Supervisor
Adjunct Professor of Computing Science

Dr. Bob Russell, Internal Examiner
Professor of Mathematics

Dr. Richard Bartels, External Examiner
Emeritus Professor of School of Computer Science,
University of Waterloo

Date Approved: _____

Abstract

The Body Centered Cubic (BCC) and Face Centered Cubic (FCC) lattices along with a set of box splines for sampling and reconstruction of trivariate functions are proposed. The BCC lattice is demonstrated to be the optimal choice of a pattern for *generic* sampling purposes. While the FCC lattice is the second best choice for this purpose, both FCC and BCC lattices significantly outperform the accuracy of the commonly-used Cartesian 3-D lattice.

A set of box splines tailored to the geometry of the BCC and FCC lattices are proposed for approximation of trivariate functions on these lattices. Furthermore, for efficient evaluation, the explicit piecewise polynomial representation of the proposed box splines on the BCC lattice are derived. This derivation can be generalized for other box splines to provide efficient evaluation of box splines at arbitrary points.

Despite the common assumption on the superior computational performance of tensor-product reconstruction, it is demonstrated that these non-separable box spline-based reconstructions on the BCC and FCC lattices outperform their tensor-product counterparts on the Cartesian lattice. In particular, the box spline-based reconstruction on the BCC lattice is shown to be twice as efficient as the corresponding tensor-product B-spline solution on the Cartesian lattice. Hence, we establish the fact that not only are these non-Cartesian lattices attractive from the sampling-theory aspects, they also allow for efficient and superior reconstruction algorithms.

Acknowledgments

It is a great privilege to attend graduate school. I have truly enjoyed my time as a graduate student, and this would not have been possible without the guidance and the support of my parents; thank you! I am thankful to my senior supervisor, Torsten Möller, for his remarkable persistence, enthusiasm, criticisms and continued support. I am grateful to Carl de Boor for stimulating and rigorous comments and his invaluable feedback. I have also largely benefited from suggestions I received from Richard Bartels for which I would like to thank him. I would like to extend my gratitude to Dimitri Van De Ville and Ramsay Dyer for many interesting and thoughtful discussions and collaborations.

I have benefited from the `m3D` MetaPost package of Anthony Pham for drawing non-trivial 3-D figures.

Contents

List of Tables

List of Figures

Chapter 1

Motivation

The omnipresence of the Cartesian lattice in a wide range of areas in computing science is certainly indisputable. The simple structure of the Cartesian lattice and its separable nature allows one to readily apply a tensor-product paradigm to many problems in a multi-dimensional setting. The power of the dimensionality reduction will remain the major reason that the Cartesian lattice will be the preferred tool in numerical algorithms. The other attraction of the Cartesian lattice is that it simply exists in *any* dimension and often tools and theory extend to problems in a higher dimensional setting in a trivial manner.

Attracted to the above mentioned luxuries, most researchers shy away from the overhead of understanding and dealing with alternate structures that compete with the Cartesian lattice. However, there is sufficient evidence showing that nature does not ignore the advantages of employing these alternative structures. The goal of this chapter is to discuss some instances where non-Cartesian lattices appear in nature and describe how these lattices help us better model natural phenomena in the computational domain.

1.1 Non-Cartesian Lattices in Nature

Perhaps the most striking non-Cartesian example observed in nature is the structure of a honeycomb [1] in Figure 1.1.[2] The center of each cell lies on a *hexagonal lattice* whose Voronoi cells (which are regular hexagons) are filled with honey and separated from each other by

[1] `http://commons.wikimedia.org/wiki/Image:Honey_comb.jpg`

[2] Permission is granted to copy, distribute and/or modify this and subsequent GNU Free Documentation License images under the terms of the GNU Free Documentation License, Version 1.2 or any later version

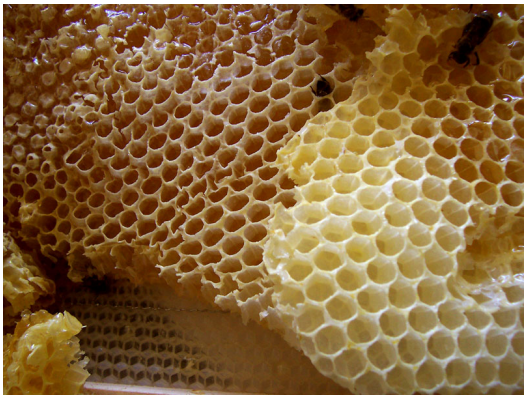
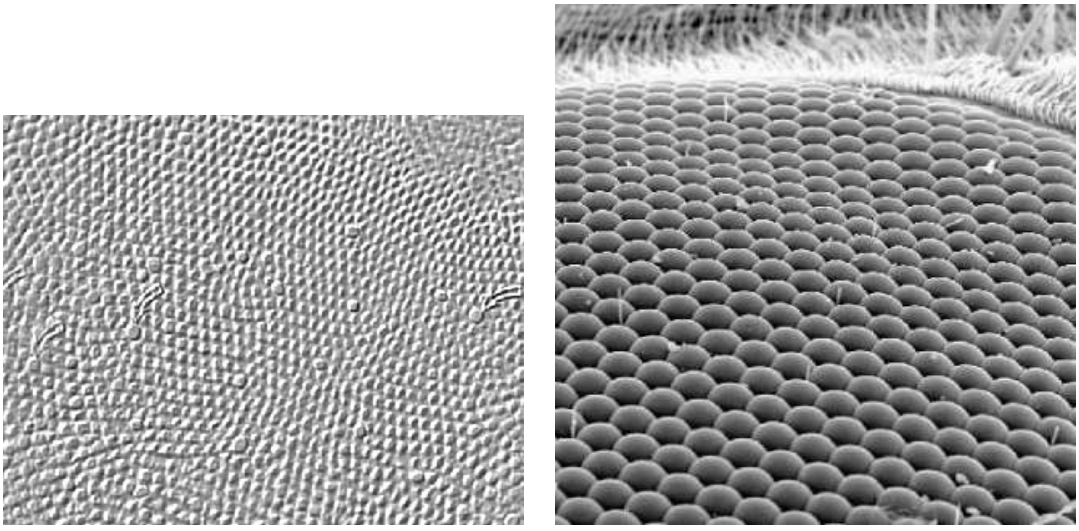


Figure 1.1: In a honeycomb each cell has an exact hexagonal shape. Image from wikipedia by permission of the GNU Free Documentation License.

wax. There are two possible explanations for the curious hexagonal shape of honeycomb. The first argument is based on the fact that, while tiling the 2-D plane, the hexagon covers the biggest possible surface area for its perimeter length [64]. A circle would maximize the containing surface area for its perimeter, but fails to fill the 2-D plane. Hence, for constructing such a honeycomb, bees use the minimal amount of wax to cover the space; therefore, they optimize the volume of the honey containers. A second plausible explanation is that the bees put together the cells individually and they push as far as they can. This argument is based on the fact that the hexagonal lattice allows for the optimal packing of disks in 2-D. An argument in favor of the latter explanation is that the queen cells which are constructed in isolation have irregular and lumpy shapes without any concern for optimality [97]. It is likely that the honey bee constructs the honeycomb based on instinct, and the prevailing theory of biology is that the appearance of such efficient shapes in nature is a result of natural selection [114].

The hexagonal lattice is also observed in the anatomical structure of photo-receptors in the central part of the fovea where the light is projected to the back of the human eyes (Figure 1.2(a)). These photo-receptors act as a scanning device that samples the intensity of the incoming light and generates an electrical signal that is sent to the brain for further visual processing [46]. It is likely that the optimal placement of these photo-receptors has

similar roots in the natural selection.



(a) (b)

Figure 1.2: (a) The photo-receptors in the fovea are optimally packed in a hexagonal fashion. (b) The facets (mini eyes) in the eye of a fly are optimally packed on a hexagonal lattice. Public domain images, courtesy of WebVision [46].

The hexagonal lattice is found in the structure of many insect eyes. Several species of insects have many (≈ 700) mini-eyes called facets inside each eye (Figure 1.2(b)). These facets are also densely packed in a hexagonal fashion [74]. One can argue that the arrangement of the facets optimizes the resolution at which the insect's eye perceives the visual information. The hexagonal lattice appears in various other physical phenomena like in the structure of superconductors and ice crystals in the snow flakes.

Non-Cartesian 3-D structures appear mostly in crystallography in the atomic structure of crystals, elemental metals and minerals. The projection of crystal lattice structures are obtained by studying the X-ray diffraction patterns through crystals [80]. The Body Centered Cubic (BCC) lattice (see Section 1.1.2) is fairly common in nature. For instance, the atomic structure of iron, chromium and tungsten are BCC lattices. Also the Face Centered Cubic (FCC) lattice (see Section 1.1.2) appears in the structure of lead, aluminum, copper and gold [113]. The Atomium monument in Brussels represents the BCC model of an iron crystal's unit cell.

As we will see in the following section, the FCC lattice has been shown to be the densest pattern for spherical packing of atoms or spherical shapes. This dense packing is

frequently observed in nature where an optimal arrangement of spherical objects is desired. For instance, the pomegranate seeds (see Figure 1.3) are packed densely in an almost optimal FCC fashion. The jelly around the seeds, on average, resembles the Voronoi cell of the FCC lattice which is a rhombic dodecahedron [80] –image courtesy of [116].

Figure 1.3: Pomegranate seeds resemble a rhombic dodecahedron [116]. Image from wikipedia by permission from the GNU Free Documentation License.

1.1.1 Discrete Geometry

The hexagonal lattice is the solution to several problems of discrete geometry in 2-D. The optimal **packing** problem in 2-D which is the problem of finding the densest arrangement of equally sized disks results in the hexagonal lattice as illustrated in Figure 1.4(b,c). When compared to the regular Cartesian packing of disks (Figure 1.4(a)), the hexagonal packing is about 14% more efficient. The hexagonal lattice also allows one to *cover* the plane with equal sized disks while maintaining the minimum area over the regions that two disks overlap (i.e., **covering** problem); this configuration is illustrated in Figure 1.4(d)[17].

The problem of packing spheres in 3-D has its roots in the 16th century when an English nobleman and seafarer, Sir Walter Raleigh, asked his mathematical assistant, Thomas Harriot, to develop a formula to count the number of cannon-balls stacked in his ships. Harriot who was an accomplished mathematician and astronomer was able to solve that problem; but he, curiously, pushed the question further to find the most efficient way to pack the cannon-balls on the ship. The objective is to find an arrangement in which the cannon-balls fill as large a portion of the space as possible. The proportion of the space filled by the spheres is called the packing **density**. After contemplating the problem for a while, Harriot decided to share this problem with one of the outstanding contemporary mathematicians, physicists and astronomers of the time, Johannes Kepler.

Kepler studied this problem and concluded that the densest packing must happen in sheets of 2-D hexagonally packed spheres laid on top of each other such that every sphere on a sheet falls inside the dimple formed between *three* spheres in the lower level sheet. This alternate sheet arrangement of spheres where each hexagonal sheet fills in the dimples of the hexagonal sheet below it is called **Hexagonal Close Packing** (HCP). In 1611 Kepler

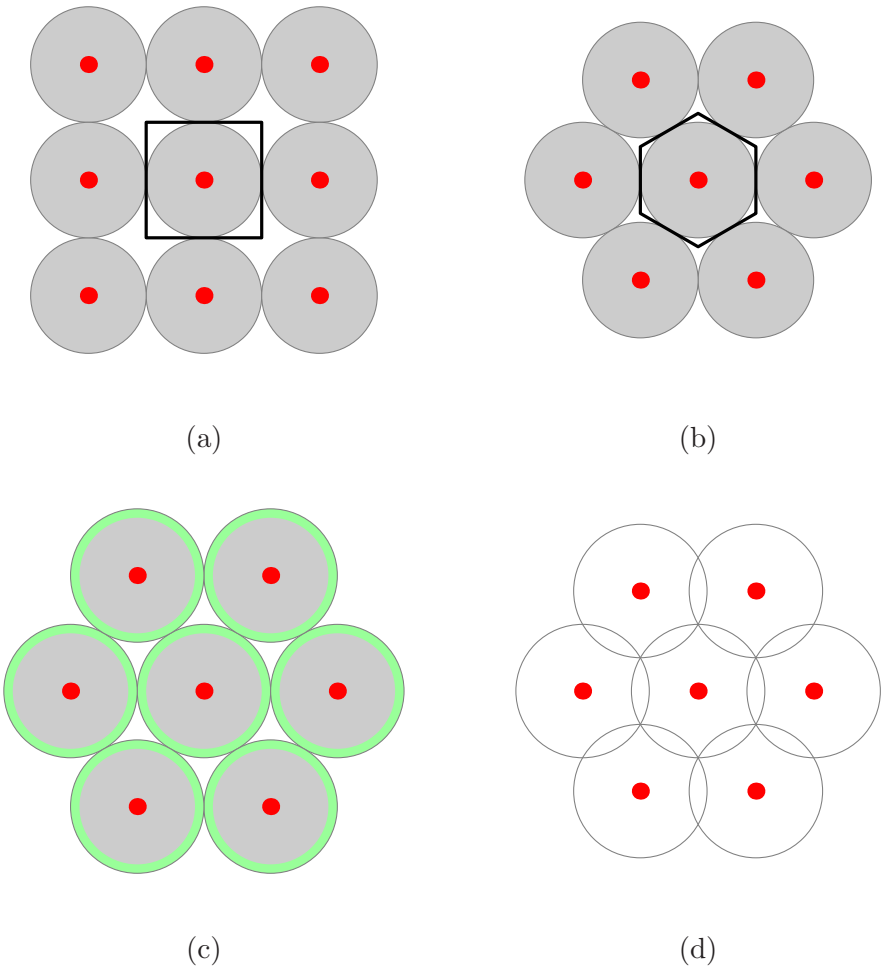


Figure 1.4: (a) Cartesian close packing. (b) Hexagonal close packing takes smaller area. The unit cells are outlined in black. (c) Hexagonal close packing of larger disks occupying the same area as covered in (a). The area of the unit cell in (b) is ≈ 0.866 of (a) and (c). (d) Covering the plane with hexagonally arranged disks minimizes the area in the overlapping regions.

conjectured that the HCP packing, whose density is about 74.05%, produces the optimal packing of 3-D spheres in a book entitled "The Six-Cornered Snowflake" [93].

In 1881, the crystallographer, William Barlow, demonstrated that the FCC packing (which is an atomic structure) achieves the *same* packing density as the HCP packing does [93]. In fact, he demonstrated that FCC is a specific type of HCP packing (when looking from a different angle) and there are infinitely many packings that match the same packing density. The FCC packing would place a sphere on the corners of a cube and additional spheres in the centers of each face. The illustration Barlow used to demonstrate this similarity is re-drawn in Figure 1.5.

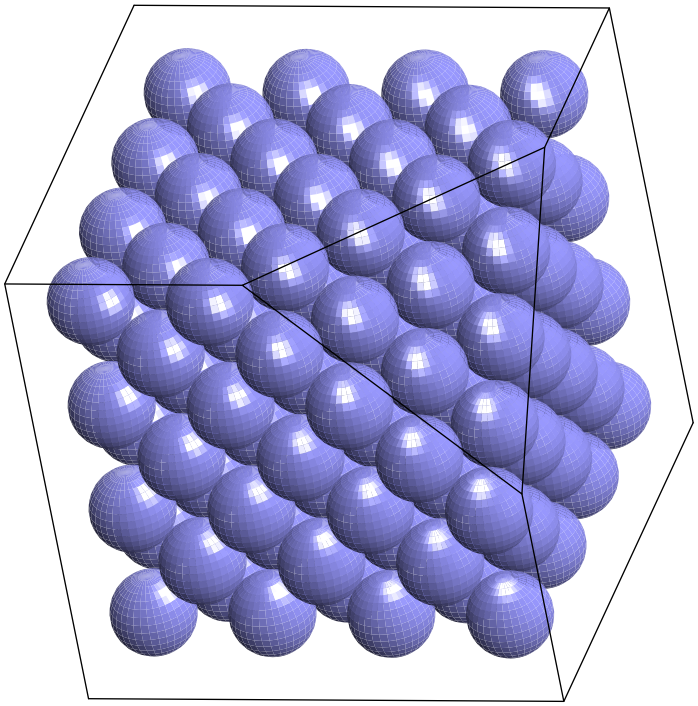
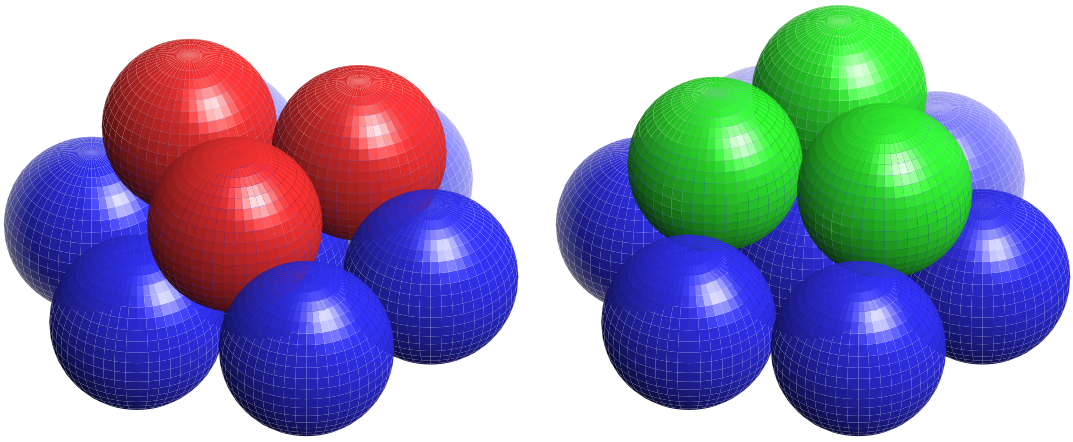


Figure 1.5: The Face Centered Cubic Packing with a cut-away view shows the hexagonal arrangement of spheres from the diagonal viewpoint of the cube.

When stacking a layer of hexagonally packed spheres on top of another layer, every sphere in the second layer fills a dimple formed by three of the spheres in the first layer. Due to the hexagonally packed arrangement of the second layer, only half of the dimples of the first layer are filled. Therefore, there are two choices for the second layer. We can call the arrangement in the first layer (A) and the two choices in the second layer (B) and (C)

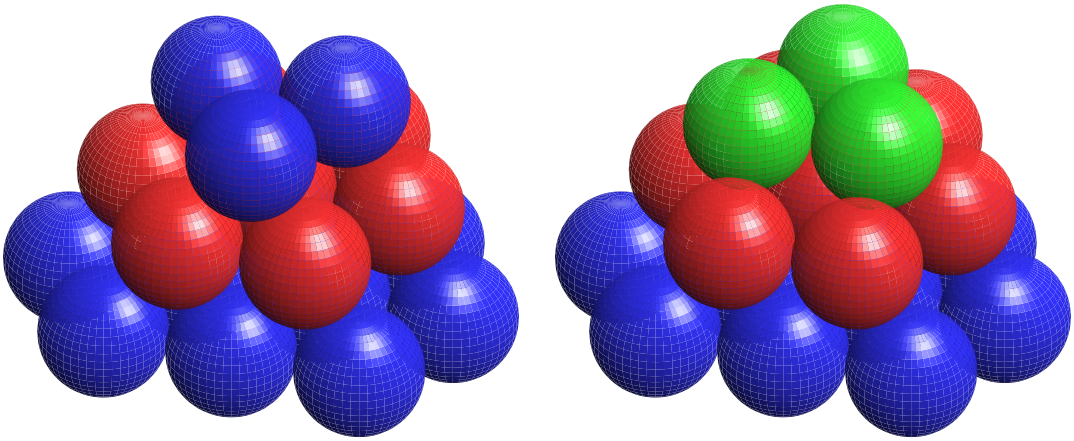
as illustrated in Figure 1.6.



(a) AB (b) AC

Figure 1.6: Two different ways for the consecutive sheets of hexagonally packed spheres to lie on top of each other.

It turns out that all possible arrangements of layers A, B and C achieves the same packing density. In particular the alternating sheets described by $ABAB\ldots$ is the HCP that Kepler studied and $ABCABCABC\ldots$ is the FCC packing mentioned above (Figure 1.7). The attractiveness of the FCC packing is that the centers of the spheres form a discrete sub-group of the Euclidean space that contains the origin, i.e., a **point lattice** (see Section 1.1.2). However, the HCP points do not form a sub-group and hence do *not* form a lattice.



(a) An example of HCP for ABA (b) The FCC packing for ABC

Figure 1.7: HCP *vs* FCC packing

The FCC packing is (perhaps not incidentally) the grocer's method for piling up oranges, apples and melons on a pyramid; also, the FCC lattice is a common arrangement for storing cannon-balls; see Figure 1.8. It is important to note that the packing density, mentioned above, is not defined over a finite domain, but is an asymptotic measure. For finite packings, depending on the geometry of the boundary of the space, there are irregular packings that attain the maximum density.

Kepler's conjecture remained open for nearly 400 years. The difficulty in proving Kepler's conjecture was that, for finite domains, irregular packings outperform the regular ones; however, extending these irregular packings significantly decreases their density. On the other hand, the truth of Kepler's conjecture suggests a special property of the Euclidean geometry that regular packings outperform irregular packings in the limit. In 1831, Gauß proved that the packing density of HCP attains the maximum possible packing density among the regular packings [17]. This meant that if Kepler's conjecture was to be disproved, a more dense packing structure had to be found with an irregular pattern. Gauß's result was perhaps a disappointing result for disproving Kepler's conjecture by counter-examples since eliminating all possible irregular packings is intractable.

After Gauß's observations, there was no progress on this problem for a while. David Hilbert included Kepler's conjecture in his list of unsolved problems in mathematics (Hilbert's 18th problem). The situation turned more hopeful in 1953 when Fejes Tóth showed that the search for disproving Kepler's conjecture among all possible irregular patterns can be reduced to a finite, but very large, number of cases. This was an important observation since it offered the hope of enumerating all possible irregular packings in finite time. Following Tóth's contribution, Thomas Hales determined that the maximum density of all possible arrangements could be determined by minimizing a function of 150 variables. In 1998 Hales announced that he has completed the (computer-aided) proof and believes the Kepler conjecture to be true [38].

Sphere packings in higher dimensions are of great interest to both geometers and engineers with various applications such as coding theory and quantization theory [17]. One curious lattice of interest in 24 dimensions is the *Leech lattice* that attains an unusually high packing density and is used to design error-correcting codes [17] (codes whose bit-representation is resistant to a few erroneous bit flips caused by the presence of noise in communications).

The covering problem has similarly found great attention in geometry and has various

Figure 1.8: Cannon-balls stacked in a face-centered-cubic lattice. (Arlington, Virginia. Taken by Matthew B. Brady about 1863. Public Domain Image. Obtained from the Library of Congress Image Collection.)

applications in Engineering. As mentioned earlier, the hexagonal lattice in 2-D is the solution to the cover problem in 2-D. However, in 3-D the BCC lattice allows the best covering of the 3-D space while allowing for the minimum possible overlap [17]. In conclusion, the FCC lattice allows for the most efficient packing, while the BCC lattice allows for the best covering of space with spheres.

1.1.2 Point Lattices

In this section we will briefly present selected topics in geometry of three-dimensional non-Cartesian lattices of interest.

In geometry, a **point lattice** is a discrete subgroup of the Euclidean space that contains the origin [80]. In other words, the set of points in a lattice is closed under addition and negation. The discrete nature of a lattice implies that every lattice point has a neighborhood in which it is the only lattice point. Such a neighborhood can be formally defined using a Delaunay diagram and the corresponding Voronoi cell. The **Voronoi cell** (Wigner-Seitz cell) of a lattice point comprises all Euclidean points for which that lattice point is a closest one. Each lattice point has surroundings identical to those of all the other lattice points hence one can refer to the Voronoi cell of a lattice without ambiguity [8].

In the context of close packings (i.e., configurations where spheres are touching each

other), cubic lattices are of interest. The term 'cubic' refers to the fact that in a close packing on a **cubic lattice**, one can find a unit cell that is a cube. A **unit cell** is a parallelepiped that contains a minimal portion of the space such that the close packing is obtained by periodic tiling with the unit cell.

There are three different types of cubic close packings: **Cartesian Close** (CC) packing, **Face Centered Cubic** (FCC) packing, and **Body Centered Cubic** (BCC) packing. The Voronoi cells of these lattices are a cube, a rhombic dodecahedron (Figure 1.9), and a truncated octahedron (Figure 1.10), respectively. The FCC and BCC lattices are the three-dimensional counterparts of the hexagonal lattice in 2-D.

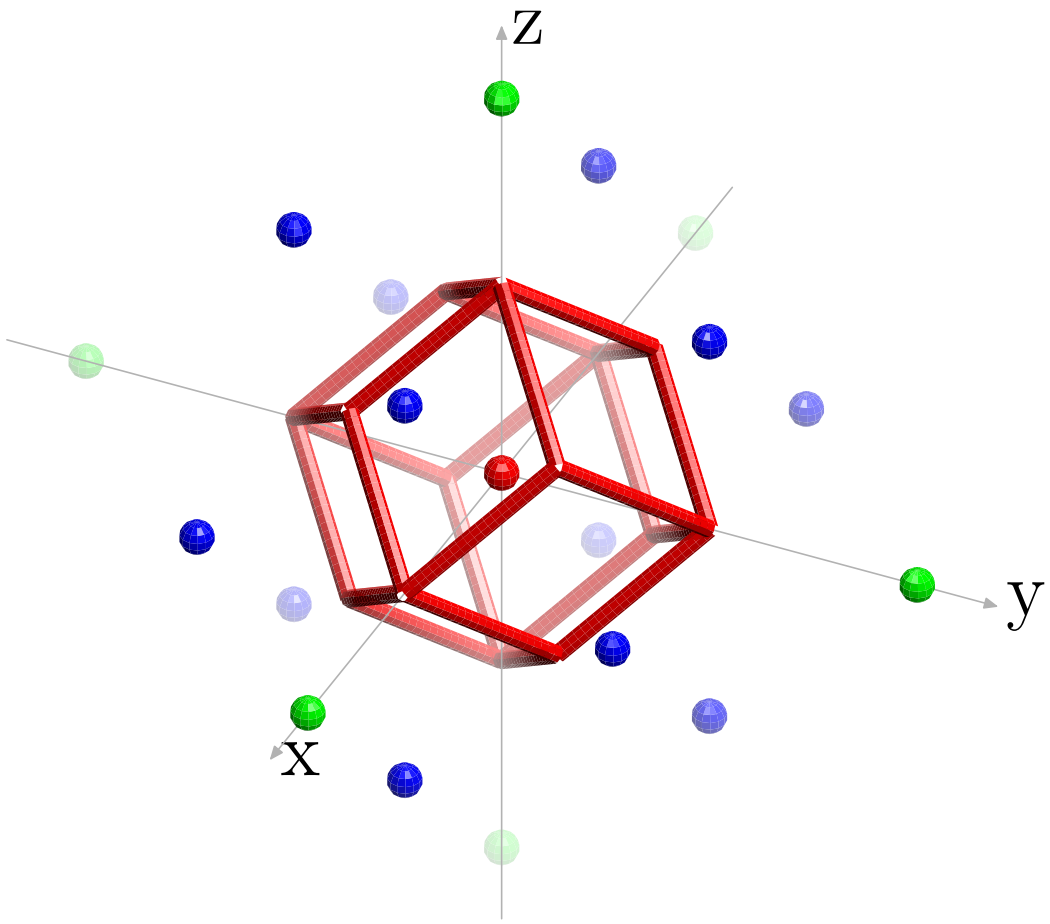


Figure 1.9: The Voronoi cell of the FCC lattice is a rhombic dodecahedron. The blue neighbors are at offsets of $(\pm 1, \pm 1, 0)$, $(\pm 1, 0, \pm 1)$ and $(0, \pm 1, \pm 1)$. The green neighbors are at offsets of $(\pm 2, 0, 0)$, $(0, \pm 2, 0)$ and $(0, 0, \pm 2)$.

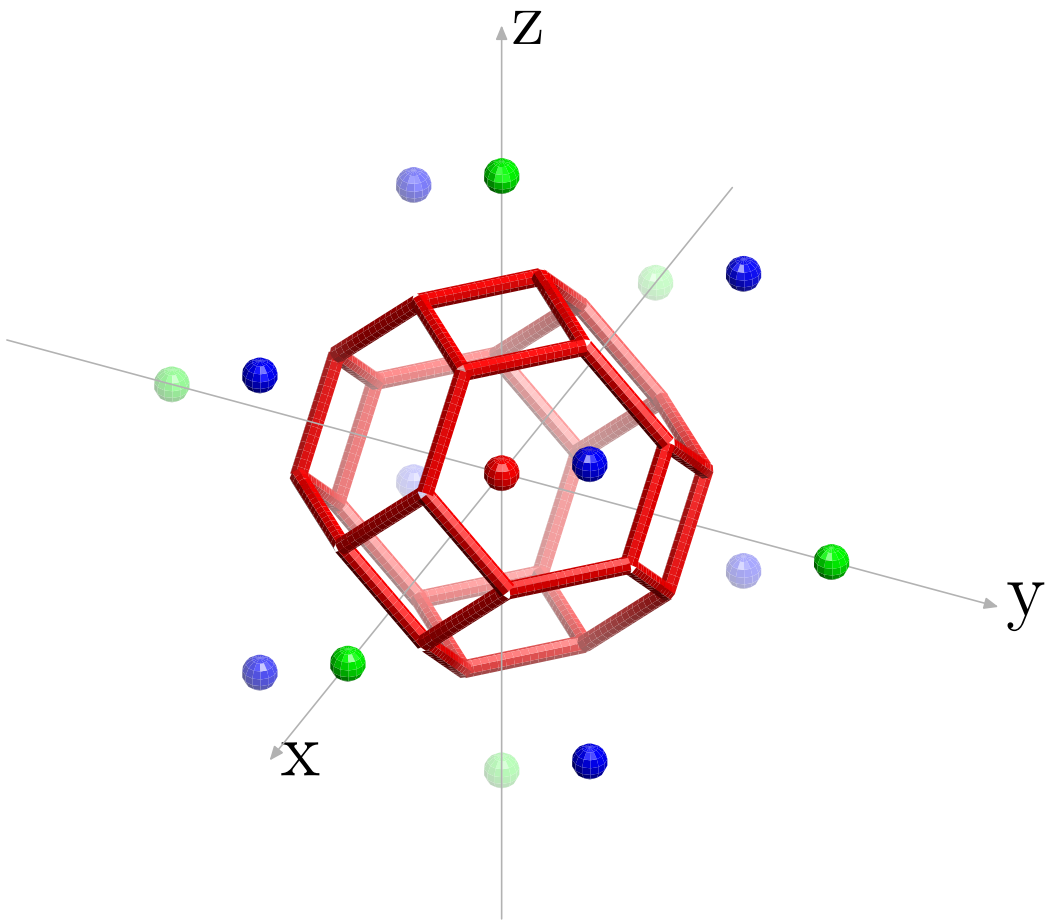


Figure 1.10: The Voronoi cell of the BCC lattice is a truncated octahedron. The blue neighbors are at offsets of $(\pm 1, \pm 1, \pm 1)$. The green neighbors are at offsets of $(\pm 2, 0, 0)$, $(0, \pm 2, 0)$ and $(0, 0, \pm 2)$.

The FCC lattice is often referred to as the $\boldsymbol{D}_3$ lattice [17]. In fact, $\boldsymbol{D}_3$ belongs to a general family of lattices, $\boldsymbol{D}_n$, sometimes called checkerboard lattices. The checkerboard property implies that the sum of the coordinates of the FCC lattice sites is always even. Therefore, one can start with $\mathbb{Z}^3$ and retain only points whose sum of coordinates are even. The resulting FCC lattice has a density of 1/2 in the original $\mathbb{Z}^3$ and therefore, its Voronoi cell has a volume of 2. Alternatively, one can describe the FCC lattice as the points at the corners of a cube in addition to the centers of the faces of the cube, hence the name Face Centered Cubic.

The BCC lattice points are located at the corners of a cube with an additional sample in the center of this cube. Therefore, the BCC lattice can be considered as two interleaving Cartesian lattices where the vertices of the secondary Cartesian lattice are moved to the center of the primary Cartesian cells. An alternative way of describing the BCC lattice as a sub-lattice of the Cartesian lattice is to start with a Cartesian lattice (i.e., $\mathbb{Z}^3$) and retain only those points whose coordinates have identical parity. For an integer point in $\mathbb{Z}^3$ to belong to the BCC lattice, all three of x, y and z coordinates need to be odd or all three need to be even. Therefore, the BCC lattice is a subgroup whose quotient group is of order *four*[3]. Therefore, the BCC lattice is a sub-lattice of $\mathbb{Z}^3$ whose density is 1/4 in $\mathbb{Z}^3$; in other words, the volume of the Voronoi cell of each lattice point is 4.

The FCC and BCC lattice points can be generated by integer linear combinations of the columns of their **generating** matrices:

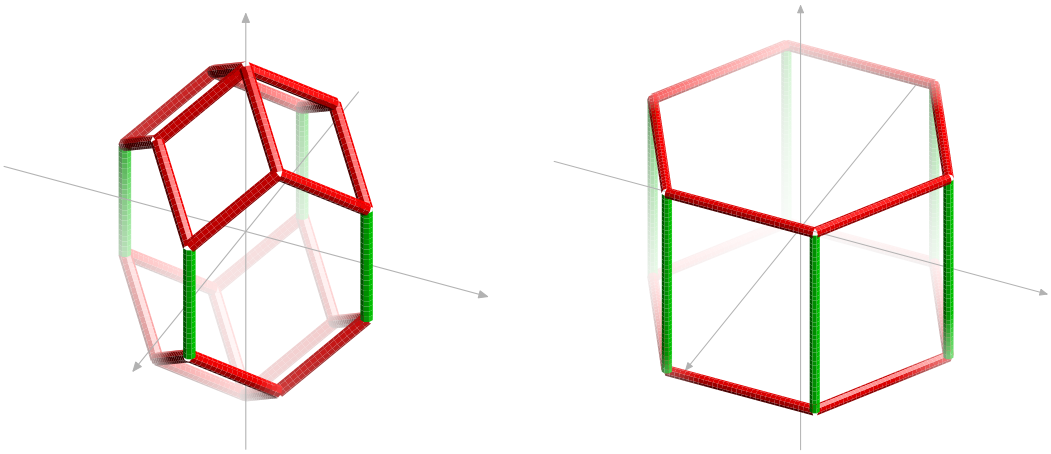
$$\boldsymbol{BCC} = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}, \boldsymbol{FCC} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{1.1}$$

It is fairly obvious to note that for a particular lattice there are many generating matrices [104]. The span of all of these different generating matrices yields the same set of points, but the integer coordinates of points are different for each choice of a generating matrix. Different choices of generating matrices yield different **fundamental regions** in the lattice– the parallelepiped formed by the basis vectors (columns) of the generating matrix [32, 105].

Since the Voronoi cell of these lattices tile the 3-D space, cube, truncated octahedron and rhombic dodecahedron are called **space filling polyhedra**. In 3-D, there are only

[3]Looking at the (x, y, z) coordinates mod 2, there are 8 different combinations and we are interested in only two: $(0, 0, 0)$ and $(1, 1, 1)$.

five (combinatorially) different types of polyhedra that have this nice property. The other two are the hexagonal prism and the elongated rhombic dodecahedron. These polyhedra are sometimes referred to as Fedorov's parallelohedra [80]. It is a curious fact that all of



(a) (b)

Figure 1.11: (a) Elongated rhombic dodecahedron is a space filling zonotope with five zones. (b) Hexagonal prism is a space filling zonotope with four zones. Edges belonging to one of the zones of these zonotopes have been colored green.

these polyhedra are (the support of) shadows (projections) of higher-dimensional hypercubes down to 3-D. Polytopes that are the support of the projection of higher dimensional hypercubes are called **zonotopes**. A zonotope is a polytope where both itself as well as its facets exhibit point symmetry with respect to their centers (e.g., a rectangle is symmetric with respect to its center, but a triangle is not). A three-dimensional zonotope is often referred to as **zonohedron**. It turns out that a polyhedron is a zonohedron if and only if its faces are (center) symmetric [112]. The set of edges of a zonohedron can be partitioned into **zones** in such a way that all edges in one zone are parallel [40]. The edges in one zone of a zonohedron construct a "belt"-like structure around the polyhedron. For instance, the elongated rhombic dodecahedron is a five zone zonohedron, one of which has been illustrated with green color in Figure 1.11(a). A zonohedron can be specified by a representative vector from each zone. These representative vectors can, in turn, be used to construct a projection matrix that yields the polyhedron from the corresponding higher dimensional hypercube. The dimension of the hypercube is simply the number of zones in a zonohedron.

With this knowledge one can easily verify that the cube is the identity projection of

a 3-D cube, rhombic dodecahedron and hexagonal prism are different projections of a 4-D tesseract, elongated rhombic dodecahedron is a projection of a 5-D hypercube and truncated octahedron is a projection of a 6-D hypercube.

1.2 Non-Cartesian Lattices in Visual Computing

Motivated by attractive features of the hexagonal lattice and its employment in biological visual systems, researchers in image processing and computer vision have tried to exploit the hexagonal lattice in acquisition, processing and visualization algorithms.

The beginnings of digital image processing are usually traced back to the 1960s when it was motivated by the need to enhance lunar surface images transmitted by Ranger 7 [64]. Even though in the early stages the Cartesian lattice was studied heavily, McCormick considered a hexagonal lattice (in addition to the Cartesian lattice) for a thinning algorithm to process digital images and detect nuclear particles in bubble chamber experiments [60].

It was Petersen [72] who realized the attractive property of the hexagonal lattice for sampling functions whose support in the Fourier domain is bounded (i.e., band-limited). In fact, he showed that the hexagonal lattice is the optimal pattern for sampling (isotropically) band-limited functions. Given an isotropic and band-limited spectrum of the original function, one can properly sample that function on a hexagonal lattice with only 86% of the number of samples one would require for a proper sampling on the Cartesian lattice. Further details on this subject are discussed in Section 2.2.1.

1.2.1 Acquisition

Motivated by the optimal sampling property of the hexagonal sampling lattice, there has been interest in exploiting this optimality and increase the "information" captured by the samples taken on the hexagonal lattice in sensor arrays. One (perhaps misleading) approach is to sample images on the regular Cartesian lattice with generic hardware and then adopt a resampling scheme [64]. However, during the resampling process no information can be injected to the captured data and such re-sampled hexagonal data will contain no more information than was originally captured.

Staunton [85, 84] has been an advocate for building custom hardware for acquiring hexagonal images. He designed a pipeline to introduce a delay to alternate rows of video

images for hexagonal acquisition. Also, general purpose sensors have been proposed in [41]. However, the progress in this field has not been enough compared to the Cartesian pipeline.

The attractive features of 3-D optimal sampling demonstrate tremendous potential to be exploited in medical scanning devices such as Magnetic Resonance Imaging (MRI); this is discussed in the work of Ehrhardt [29]. This technique can be used to reduce scan time, data throughput, and data storage requirements while maintaining resolution or to improve resolution while maintaining scan time. It is worthwhile to mention that while it is reasonably easy to modify the k-space trajectories to collect MRI samples on the BCC or FCC lattices, adjusting the impulse response (excitation function) for proper anti-aliasing is the challenging part. Similarly, a simple phase shift in k-space to introduce a shift in the space domain does not provide an optimal BCC sampling. The optimality of the BCC lattice allows one to capture about 30% more information given that the same number of samples are taken on a Cartesian lattice. Similarly, for tomographic reconstruction, Matej and Lewitt considered the optimal sampling properties of the BCC lattice and developed a tomographic reconstruction method based on radial basis functions [54, 55]. A good exposition of the methodology for Positron Emission Tomography (PET) they developed can be found in a technical report [43].

In computed tomography, the notion of 'interlaced lattice' has been explored and proved to significantly improve the quality of reconstruction [19, 27]. Cormack [19], suggested that one can improve the quality of tomography by almost 50% if the interlaced lattice was chosen for reconstruction purposes. While the interlaced lattice bears resemblance to the hexagonal lattice, for a given identical angular and distance discretization resolution, it matches the hexagonal lattice exactly. Rattey and Lindgren pushed this idea further and explored optimally sampling the 2-D Radon Transform (RT) plane [76]. In this work, the information on the RT plane is considered as a bivariate function (as opposed to the traditional univariate parameterization) and the optimal hexagonal sampling is applied on that function. The advantages of using the interlaced lattice are explained by means of Fourier analysis in works by Faridani [33, 34, 35, 52].

Wells et al. [110] also advocate the use of the hexagonal sampling pattern in practical CT applications. They state that conventional CT scanners are not making the best use of the available data if they are not using some scheme that approximates hexagonal sampling, but are using a scheme equivalent to Cartesian sampling. "It seems that one ought to get the maximum spatial resolution from the number of samples acquired as the CT scanning

and image reconstruction are expensive enough in time and dollars without unnecessarily throwing resolution away" [110]. The mentioned framework and simulation studies and experiments for typical first generation scanners have been described in Suparta's PhD thesis [92].

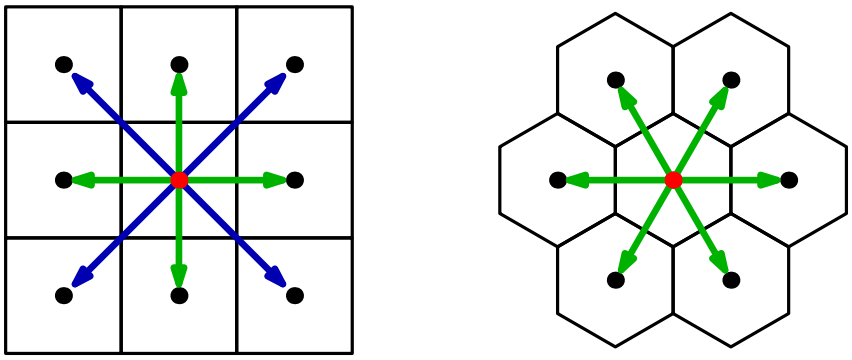
Unser et al. used the BCC lattice for measuring their proposed volumetric spectral signal to noise ratio (VSSNR) model for assessment of 3-D reconstructions of biological macromolecules by transmission electron microscopy [101]. They used the BCC lattice points to approximate the VSSNR by a radial basis function based on generalized Kaiser-Bessel window functions.

1.2.2 Other Applications

The hexagonal lattice has also found interest in the mathematical morphology domain thanks to the elegant notion of neighborhood on each lattice point. In a 2-D Cartesian lattice, there are two distinct sets of neighbors in the immediate neighborhood of each lattice point. There are four closest neighbors whose Voronoi cells share a face with the Voronoi cell of the center lattice point (face-connected: green arrows in Figure 1.12(a)). There are also four more neighbors whose Voronoi cells share a vertex with the Voronoi cell of the center point (vertex-connected: blue arrows in Figure 1.12(a)). In many computer vision applications a face-connected or a vertex-connected neighborhood (or their union) is arbitrarily chosen as the neighborhood. However, in the case of a hexagonal lattice a total of six face-connected neighbors completely separate the center point from the rest of the lattice (see Figure 1.12(b)). This distinction allows one to distinguish a foreground image from its background in a more sensible fashion.

Besides the improved notion of neighborhood, for creating isotropic structural elements in morphology, it is desirable to have the richest group of rotations possible. In 2-D, the hexagonal lattice's Voronoi cell (hexagon) provides a better element than the simple square of the Cartesian lattice. Motivated by these advantages Meyer developed the corresponding morphological operators on FCC and BCC lattices [63]. The improved notion of neighborhood is also desirable in certain finite element modeling applications [65].

The improved isotropy of neighbors in FCC and BCC lattices makes their voxel representations (i.e., their Voronoi cells) more suitable to rotations than the cubic voxels on the Cartesian lattice. This property proves useful to the discrete representation of shapes and their skeletons [87]. In a distance transform (DT), each picture element in an object is



(a) (b)

Figure 1.12: (a) A Cartesian lattice point, face+vertex connectivity. (b) The hexagonal lattice point, face connectivity.

labeled with the distance to the closest element in the background. DTs are very useful tools in many types of image analysis, from simple noise removal to advanced shape recognition. Results concerning weighted DTs and Euclidean DTs on the FCC and BCC lattices have been studied [89, 88].

Miller [65] examined the advantages of non-Cartesian lattices for estimating the perimeter (in 2-D) and surface area (in 3-D) of piecewise smooth objects sampled on Cartesian and non-Cartesian lattices. The estimation process uses local counting methods on the elements (i.e., pixels or voxels) where each element's area on the boundary is locally calculated and accumulated in the total boundary of the discretized object. In local counting methods the hexagonal lattice is demonstrated to provide a better perimeter estimation than the commonly used 2-D Cartesian lattice. In 3-D it is shown that for surfaces of random orientation with a uniform distribution, the expected error of surface area estimates is significantly smaller for the FCC and BCC lattices than for the standard Cartesian lattice. The competition between FCC and BCC is in favor of BCC for the lowest mean error in estimating the area while FCC has the much smaller maximum error. This work demonstrates that FCC and BCC lattices can be exploited in various fields like medical imaging where area estimation of tissues or tumors is a challenging and important task. Similarly, in the manufacturing world, the improved surface area estimation by non-Cartesian lattices is useful for computing the flux of physical quantities across or along a surface such as shear stress,

heat or electromagnetic fields or wind resistance.

Half-toning is a technique for producing the illusion of continuous tone in binary display devices (such as laser printers) through a random placement of pixels. Hexagonal patterns have also been proposed for half-toning [86, 100, 44]; however, the blue noise dithering method proposed by Ulichney [99] ironically showed that only on a rectangular sampling lattice it is possible to isolate pixels at all gray-levels. The issue with the hexagonal lattice pattern was that the minority pixels begin to cluster when the gray-level value (normalized to $[0, 1]$) tends to $1/2$. Because of this issue Ulichney concluded that the hexagonal lattice can not support blue-noise [99]. Only recently Lau and Ulichney resolved the paradoxical result mentioned and showed that the hexagonal lattice is actually the preferred choice for stochastic dithering [50].

Sun et al. [91] demonstrated that a finite difference approximation of gradients can be obtained with a hexagonal stencil that involves a point and six of its neighbors. This seven point stencil achieves a fourth order approximation while a Cartesian finite difference stencil of the same approximation order requires the full nine-point neighborhood. It is worthwhile to mention that even though [91] considers this approximation only in 2-D on a hexagonal lattice, they refer to the hexagonal lattice as a face centered cubic lattice.

Tian [98] showed that the hexagonal lattice can help improving the estimates of the *direction-of-arrival* (DOA) of a signal on an antenna array. The estimate they found exhibits good thresholding and is close to the Cramer-Rao bound above threshold.

As we have seen in this chapter, the non-Cartesian lattices demonstrate potential in various applications specifically in the visual computing domain. In the following chapter we study the behavior of Cartesian and non-Cartesian lattices in sampling theory. We will demonstrate the theoretical advantages of using non-Cartesian lattices for sampling and reconstruction of functions – a common task in image processing, computer graphics and visualization.

Chapter 2

Sampling and Reconstruction of Functions

In this chapter, the basics of sampling theory is presented. The sampling and reconstruction theory, in the univariate and multivariate settings, is reviewed along with its implications in signal processing. We begin with the definition of the Fourier transform, $\hat{f} : \mathbb{R} \to \mathbb{C}$ of a function $f : \mathbb{R} \to \mathbb{C}$ in L_2

$$\hat{f}(\omega) = \int f(x) \exp(-\mathrm{i}\omega x)\, dx. \tag{2.1}$$

The Fourier transform extends to the multivariate setting by tensor-product:

$$\hat{f}(\boldsymbol{\omega}) = \int f(\boldsymbol{x}) \exp(-\mathrm{i}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x})\, d\boldsymbol{x}, \tag{2.2}$$

for $\hat{f}, f : \mathbb{R}^n \to \mathbb{C}$.

2.1 Uniform Univariate Sampling and Reconstruction

In signal processing sampling a continuous-domain function can be modeled as a multiplication with a *comb* (shah) function Ш_T where the samples are $T\mathbb{Z}$ [70]. Since the comb function is a distribution, its multiplication with a regular function is defined as:

$$\int \text{Ш}_T f(x)\, dx = \sum_{k\in\mathbb{Z}} f(Tk) \tag{2.3}$$

Sampling the function f with Ш_T brings about *replication* of the spectrum on a lattice with period $1/T$ in the Fourier domain; this phenomenon can be justified using the Poisson sum formula [7]. Since the Fourier transform (in the distributional sense) of a comb function is another comb function with reciprocal scale, one can demonstrate the mentioned replication phenomena as a convolution with the dual comb function [6]:

$$\text{Ш}_T f \longleftrightarrow \frac{1}{T}\text{Ш}_{\frac{2\pi}{T}} * \hat{f} \tag{2.4}$$

Here $\leftrightarrow$ denotes a Fourier transform pair.

2.1.1 Reconstruction in the Space of Band-Limited Functions

Traditionally, the subspace D of L_2 whose elements have a Fourier transform with bounded support (i.e., band-limited functions) was considered for the purpose of sampling and reconstruction. Whittaker [111] demonstrated the countable basis for the space of band-limited functions by representing a function by its cardinal series. Although the reconstruction based on the cardinal series was known to mathematicians as early as Borel [61], establishing the separable nature of this band-limited function space was Whittaker's contribution. Shannon [81] then realized the potential of this representation for communication theory and signal processing. Assuming a space $B_{(-\pi,\pi)}$ of functions whose Fourier transform is only nonzero on the *open* interval $(-\pi, \pi)$, the classical result claims that a function $f \in B_{(-\pi,\pi)}$ is completely determined by its ordinates:

$$f(x) = \sum_{k\in\mathbb{Z}} f(k)\ \text{sinc}\left(2\pi(x-k)\right) \tag{2.5}$$

where $\text{sinc}(t) = \sin(t/2)/(t/2)$ and $f(k)$ are the samples (ordinates) of the function f on integers. The supremum frequency of π allowed in the spectrum of functions in $B_{(-\pi,\pi)}$ is called the Nyquist frequency. A function can be completely recovered by samples taken at intervals inversely proportional to the Nyquist frequency of its spectrum [70]. If a function's spectrum has wider support, the Nyquist frequency is higher and the sampling interval needed to recover the function from its samples is smaller.

While Shannon's result provides an extremely useful reconstruction formula that has a wide range of applications, it has several shortcomings:

1. Due to the Riemann-Lebesgue lemma the Fourier transform of many functions, including elements of L_1 or L_2, decay to zero as the frequency increases, hence the

traditionally popular choice of band-limited functions as a space of approximation is not a bad idea; however, the band-limited assumption in the strict sense is typically not met. There is a large class of functions for modeling real phenomena that are not band-limited; for such functions, the representation on the right hand side of Equation (2.5) could turn erroneous and invalid. In practice, a pre-filtering step is applied before sampling where the original function's spectrum is filtered so that all frequencies outside of the safe range of $(-\pi, \pi)$ are removed. This pre-filtering is ideally implemented by convolving the original function with a sinc function before sampling.

2. It is difficult to use the reconstruction in Equation (2.5) since the sinc function has infinite support and decays very slowly. For instance a pointwise evaluation is a non-local operation and the value of the function at any point x is affected by many $f(k)$'s. It is common practice [70] to choose a windowed or truncated sinc where the sinc function is dampened to concentrate its energy in a windowed support. However, these techniques ruin the good approximation capabilities of the sinc function [103].

3. From the point of view of applications, it is rarely possible to obtain exact samples $f(k)$ of a function f. Instead, a sampling device usually has a point-spread function which effectively samples a weighted local average of the function [2].

These problems in sampling and reconstruction on the space of band-limited functions demands a more general framework where one could model non-band-limited functions and reconstruct functions efficiently and accurately. The concept of sampling and reconstruction in *shift-invariant spaces* remedies these problems. A good review of shift-invariant spaces can be found in [103, 2] and the references therein. The sampling and reconstruction in shift-invariant spaces can be viewed as an abstraction of Shannon's result where the restriction of band-limited space is removed with a more general space [103].

2.1.2 Sampling and Reconstruction in Shift-Invariant Spaces

If we consider X to be a Hilbert space equipped with the norm induced from the common inner product, $\langle f, g \rangle = \int f(x)g(x)\,dx$, the pre-filtering step can be viewed as an orthogonal projection of an element in X onto the space of band-limited functions D. Therefore, Shannon's result can be viewed as the "best approximation" of an arbitrary element of X

in the space of band-limited functions. To observe this relationship, we first note that the shifts of the sinc function form an orthonormal set [39]: $\{\varphi_k = \mathrm{sinc}\,(2\pi(\cdot - k))\}$ for $k \in \mathbb{Z}$ since $\langle \varphi_m, \varphi_n \rangle = \delta_{m-n}$ where δ is the Kronecker delta function. By Whittaker's result in Equation (2.5), this set forms an orthonormal basis for D. Therefore, an element $f \in X$ can be projected onto D using the dual basis to $\{\varphi_k\}$. But since $\{\varphi_k\}$ is an orthonormal basis, the elements of the dual basis are again shifted sinc functions. Therefore the best approximation of f is achieved by the coefficients $c_k = \langle f, \varphi_k \rangle$ and $\tilde{f} = \sum_{k\in\mathbb{Z}} c_k \varphi_k$. Due to the property of the sinc function that $\mathrm{sinc}\,(2\pi k) = \delta_k$, we have $\tilde{f}(k) = c_k$; it turns out that the coefficients c_k can also be viewed through sampling the function that is the convolution of the original f with the sinc function. This convolution is the ideal pre-filtering step that band-limits the spectrum of f:

$$c_k = \langle f, \varphi_k \rangle = \int f(x)\,\mathrm{sinc}\,(2\pi(x-k))dx = (f * \mathrm{sinc})(k) = \tilde{f}(k).$$

The original Shannon's theorem follows from the case that $f \in D$, in which case we have $f = \tilde{f}$ and $c_k = f(k)$.

The theory of sampling and reconstruction in shift-invariant spaces builds on using different choices of basis functions than the sinc function. Again, usually a single function φ determines the entire procedure and is called the generating function; the weighted sum of the integer shifts $\varphi_k = \varphi(\cdot - k)$ by a sequence, $c \in \ell_2$, forms the approximation space $D(\varphi)$ [2]. The choice of φ can be guided by the fact that we desire a function with compact support. Moreover, the shifts of φ should form a linearly independent set of functions.

A function $f \in X$ is approximated by $\tilde{f}$ in the space $D(\varphi)$ which is determined by its coefficient series $c \in \ell_2$, i.e.,

$$\tilde{f} = \sum_{k\in\mathbb{Z}} c_k \varphi_k.$$

These coefficients are not necessarily the samples of f but are obtained by some series of other linear functionals acting on f. It is desirable to have a stable representation, which means that small changes in the coefficients should result in small changes in $\tilde{f}$. Also, we like to have a unique representation of f; in other words we would like to be able to define $\tilde{f}$ unambiguously. Ideally, an orthonormal basis satisfies all of these requirements; however orthonormality is sometimes a prohibitively strong restriction. The next best constraint to enforce is the Riesz criterion for the generating set [23]. For $\{\varphi_k\}$ to be a Riesz basis, there

should exist two positive and finite constants C_1 and C_2 such that for every sequence of coefficients $c \in \ell_2$:

$$C_1\|c\|^2 \leq \|\tilde{f}\|^2 \leq C_2\|c\|^2 \tag{2.6}$$

The lower bound inequality implies the linear independence of the set $\{\varphi_k\}$ since $\sum_{k\in\mathbb{Z}} c_k\varphi_k = 0$ implies $c = 0$. The linear independence, in turn, implies the uniqueness of the representation. The upper bound inequality ensures that the approximation is still square integrable and $D(\varphi)$ is a subset of L_2.

In order to find the best approximation to f from the space $D(\varphi)$ we need to find the orthogonal projector that projects f to $D(\varphi)$. Since we do not necessarily have an orthonormal basis for the approximation space anymore, we need to find the dual basis $\{\varphi'_k\}$ to build the orthogonal projector to $D(\varphi)$. The dual basis satisfies the biorthogonality property: $\langle\varphi_m, \varphi'_n\rangle = \delta_{m-n}$. The best approximation then comes from:

$$\tilde{f} = \sum_{k\in\mathbb{Z}} \langle f, \varphi'_k\rangle \, \varphi_k$$

We note that the coefficients $c_k = \langle f, \varphi'_k\rangle$ are no longer exact sample values of the original function, but are the scalars obtained from the sequence of functionals $\{\langle\cdot, \varphi'_k\rangle\}_{k\in\mathbb{Z}}$.

If f happens to be in $D(\varphi)$ it is possible to recover f exactly. However, for a general element of X the exact recovery is not possible and hence one settles on the best approximation in $D(\varphi)$. Therefore, another criterion to be considered when designing or comparing generator functions is the best approximation their corresponding shift-invariant spaces can achieve. This leads to the concept of **approximation power**.

Since a typical choice of the generating function is one with a compact support, one can try to refine the space of approximation by scaling the generating function and accordingly the shifts by T. Therefore, an element in a space generated by the dilated $\varphi^T = \varphi(\cdot/T)$ has the form:

$$\tilde{f}^T = \sum_{k\in T\mathbb{Z}} c_k\varphi_k^T.$$

We also note that the density of coefficients also changes since they are now counted by $T\mathbb{Z}$. By choosing $T < 1$ we increase the "sampling" density and have a finer approximation to the function f and the choice of $T > 1$ leads to a coarser approximation of f. For a

generator function φ to be acceptable for approximation it has to be able to approximate any element of X arbitrarily closely:

$$\lim_{T \to 0} \|f - \tilde{f}^T\| = 0.$$

2.1.3 Approximation Power and the Strang-Fix Result

When it comes to choosing a generating function over another it makes sense to compare the rate at which the best approximation in their corresponding spaces converge. One of the most common measures for comparing approximation power of generating functions is developed by Strang and Fix [90]. Their result allows to predict the rate at which dilates of the space of a generating function approximates differentiable functions. In the Strang-Fix theory a generating function φ with a compact support (or an inverse polynomial decay) has approximation power of L if [90, 102]:

1. $\{\varphi_k\}$ for $k \in \mathbb{Z}$ forms a Riesz basis.
2. $\hat{\varphi}(0) = 1$ and $\hat{\varphi}(\omega)$ behaves like $(\omega - 2\pi k)^L$ around all **aliasing frequencies** $\omega = 2\pi k$ where $k \neq 0, k \in \mathbb{Z}$,

where $\hat{\varphi}$ is the Fourier transform of φ. If f is a function whose L^{th} derivative exists and is of finite energy, then the error in the best approximation of f in a dilated space generated by φ decays as fast as T^L:

$$\inf_{\tilde{f}^T \in D(\varphi^T)} \|f - \tilde{f}^T\| \leq C_\varphi T^L \|f^{(L)}\|,$$

where the constant C_φ is independent of f. The number of zero crossings at aliasing frequencies was also studied in the wavelet theory as they predict the approximation power of a wavelet decomposition [102].

The Strang-Fix result can be considered as the computational alternative to the concept of Nyquist frequency in the traditional Shannon theory. The error, generally, is not zero but can be made arbitrary small given a fine "sampling" interval of T.

Splines are a general class of approximation spaces where the function f is approximated by a *piecewise polynomial* function [26]. Given a uniform sequence of knots or seam points, the class of basis splines (B-splines) span this space. Under this assumption B-splines act as the generators of the shift-invariant space. Different orders of B-splines can be

obtained by successive convolutions of the first order B-spline (the indicator function of the sampling interval) with itself. One can design B-splines of arbitrarily high smoothness and approximation power.

2.2 Multidimensional Regular Sampling

While, in the univariate setting, the choice of uniform sampling can only be controlled by the sampling distance T, in the multivariate setting, there is a greater freedom in choosing the sampling pattern. A great introduction to multidimensional signal processing can be found in Dudgeon and Mersereau [28] and Vaidyanathan [104].

In the multidimensional case there are more choices for uniform distribution of points and it becomes a matter of regular tiling. The obvious choices in 2-D are the Cartesian lattice and the hexagonal lattice for sampling. A general point lattice for sampling is the integer span of the columns of the **sampling matrix** $\boldsymbol{M}$. Associated with every full rank lattice is a **dual** (reciprocal [28] or polar [11]) lattice whose generator matrix, $\boldsymbol{M}^{-\mathrm{T}}$, is the inverse transpose of the original lattice matrix $\boldsymbol{M}$ (see Figure 2.1).

The very same terminology as defined in Section 2.1 can be extended into the generic multivariate setting. Here one can introduce a comb function, $\text{Ш}_{\boldsymbol{M}}$, with its spikes at the lattice points $\boldsymbol{M}\mathbb{Z}^n$ [28]. The (distributional) Fourier transform of this comb function involves the dual lattice:

$$\text{Ш}_M \longleftrightarrow \frac{1}{|\det \boldsymbol{M}|}\text{Ш}_{2\pi\boldsymbol{M}^{-\mathrm{T}}}$$

When sampling a multivariate function on a lattice generated by $\boldsymbol{M}$, the effect in the Fourier domain is the replication of the spectrum on the dual lattice [28] (scaled by 2π).

$$\text{Ш}_{\boldsymbol{M}} f \longleftrightarrow \frac{1}{|\det \boldsymbol{M}|}\text{Ш}_{2\pi\boldsymbol{M}^{-\mathrm{T}}} * \hat{f}$$

The corresponding multidimensional Poisson sum formula [7] for continuous functions in L_1 whose Fourier transform decays (with a rate faster than inverse linearly) in this case becomes:

$$\sum_{\boldsymbol{k}\in\mathbb{Z}^n} f(\boldsymbol{M}\boldsymbol{k}) = \frac{1}{|\det \boldsymbol{M}|}\sum_{\boldsymbol{k}'\in\mathbb{Z}^n} \hat{f}(2\pi\boldsymbol{M}^{-\mathrm{T}}\boldsymbol{k}').$$

Since the spectrum is replicated in the Fourier domain according to the dual lattice, the Voronoi cell of the dual lattice is where the main spectrum lies. The Voronoi cell of the dual

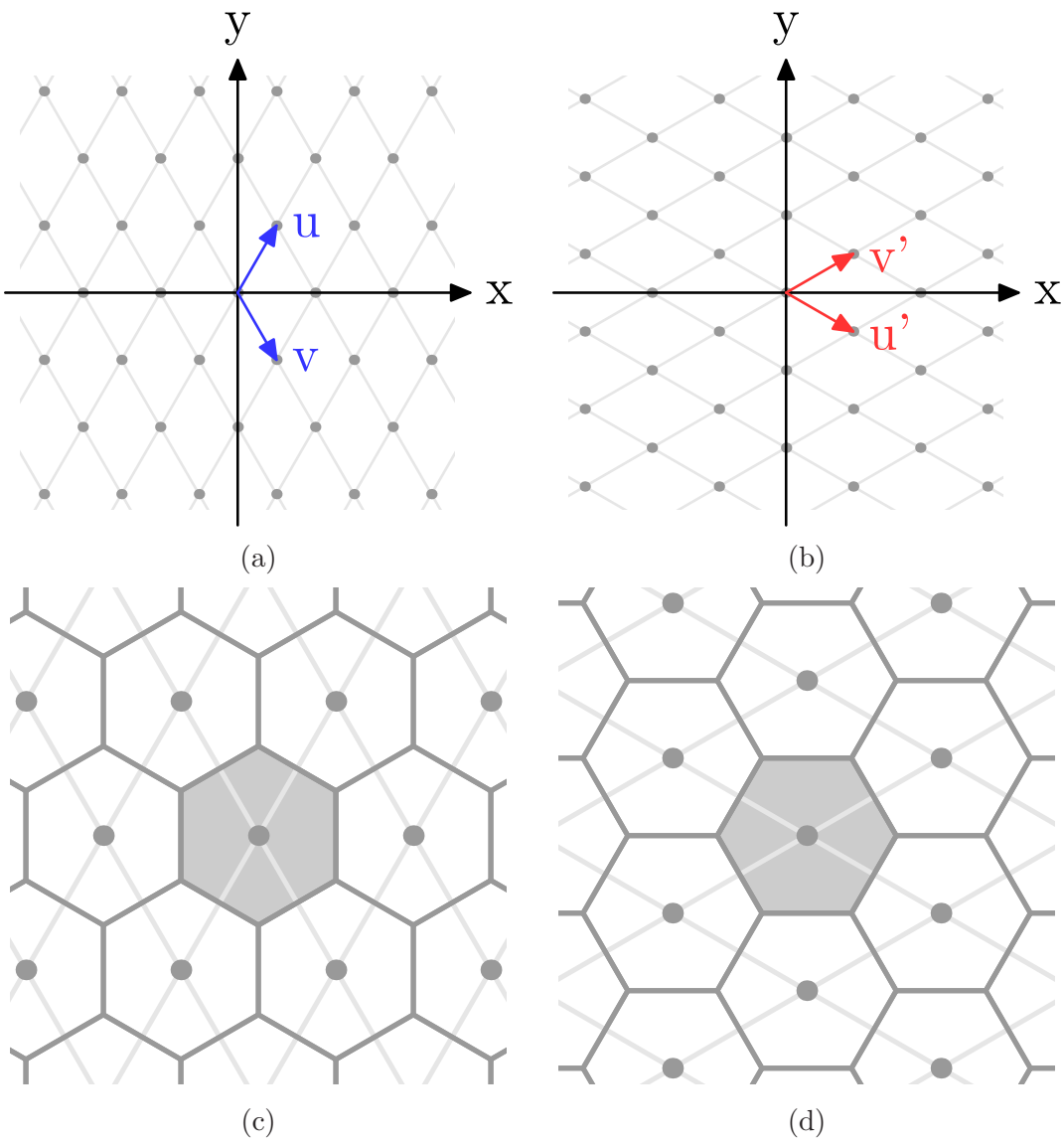


Figure 2.1: (a) A 2-D lattice with sampling matrix $\boldsymbol{M} = [\boldsymbol{u}, \boldsymbol{v}]$. (b)The dual lattice described by $\boldsymbol{M}^{-\mathrm{T}} = [\boldsymbol{u}', \boldsymbol{v}']$. (c) The Voronoi cell of the lattice. (d) The Brillouin zone: the Voronoi cell of the dual lattice.

lattice is sometimes referred to as the **Brillouin zone** [80]. The boundary of the Brillouin zone can be considered as the multidimensional version of the Nyquist frequency. Therefore, the multivariate version of Shannon's sampling theorem holds when the spectrum of the signal to be sampled on the lattice $\boldsymbol{M}$ does not contain frequencies outside the Brillouin zone (i.e., the Voronoi cell of the dual lattice).

Therefore, the (canonical) **ideal interpolation function** on the lattice $\boldsymbol{M}$ is defined as the inverse Fourier transform of the indicator function of the Brillouin zone:

$$\operatorname{sinc}_{\boldsymbol{M}} \longleftrightarrow \chi_{\boldsymbol{M}^{-\mathrm{T}}} \tag{2.7}$$

where $\chi_{\boldsymbol{M}}$ denotes the indicator function of the Voronoi cell of a lattice $\boldsymbol{M}$ at the origin.

Claim: *Lattice-shifts of* $\operatorname{sinc}_{\boldsymbol{M}}$ *on the lattice,* $\boldsymbol{M}$*, form an orthogonal system.*

Proof. We first note that a shift in the space domain amounts to a phase shift in the Fourier domain. Therefore:

$$\operatorname{sinc}_{\boldsymbol{M}}(\cdot - \boldsymbol{k}) \longleftrightarrow \exp(-\mathrm{i}\boldsymbol{k}\cdot)\chi_{\boldsymbol{M}^{-\mathrm{T}}}$$

Moreover, the multidimensional Fourier basis is an orthonormal system:

$$\langle \exp(-\mathrm{i}\boldsymbol{k}_1\cdot), \exp(-\mathrm{i}\boldsymbol{k}_2\cdot)\rangle = \delta_{\boldsymbol{k}_1-\boldsymbol{k}_2} \tag{2.8}$$

We resort to Plancherel's theorem (Fourier transform preserves angles) for lattice-shifted $\operatorname{sinc}_{\boldsymbol{M}}$ functions:

$$\begin{aligned} &\langle \operatorname{sinc}_{\boldsymbol{M}}(\cdot - \boldsymbol{M}\boldsymbol{k}_1), \operatorname{sinc}_{\boldsymbol{M}}(\cdot - \boldsymbol{M}\boldsymbol{k}_2)\rangle \\ &= \left\langle \exp(-\mathrm{i}\boldsymbol{M}\boldsymbol{k}_1\cdot)\chi_{\boldsymbol{M}^{-\mathrm{T}}}, \exp(-\mathrm{i}\boldsymbol{M}\boldsymbol{k}_2\cdot)\chi_{\boldsymbol{M}^{-\mathrm{T}}}\right\rangle \end{aligned} \tag{2.9}$$

Since $\chi_{\boldsymbol{M}^{-\mathrm{T}}}$ is the Voronoi cell of the dual lattice, its periodic replication with respect to dual lattice shifts of $2\pi\boldsymbol{M}^{-\mathrm{T}}\boldsymbol{k}$ tiles the space. Similarly the phase shifts are also periodic with the same period:

$$\begin{aligned} &\exp\left(-\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot(\boldsymbol{\omega} + 2\pi\boldsymbol{M}^{-\mathrm{T}}\boldsymbol{k}')\right) \\ &= \exp(-\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot\boldsymbol{\omega})\cdot\exp\left(-2\pi\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot(\boldsymbol{M}^{-\mathrm{T}}\boldsymbol{k}')\right) \\ &= \exp(-\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot\boldsymbol{\omega})\cdot\exp\left(-2\pi\mathrm{i}\boldsymbol{k}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}^{-\mathrm{T}}\boldsymbol{k}'\right) \\ &= \exp(-\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot\boldsymbol{\omega})\cdot\exp\left(-2\pi\mathrm{i}\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}'\right) \\ &= \exp(-\mathrm{i}(\boldsymbol{M}\boldsymbol{k})\cdot\boldsymbol{\omega}) \end{aligned}$$

since $\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}' \in \mathbb{Z}$.

Therefore, the inner product in Equation (2.8) is the sum of individual tiles in Equation (2.9) (which are identical to each other). For different lattice shifts ($\boldsymbol{M}\boldsymbol{k}_1 \neq \boldsymbol{M}\boldsymbol{k}_2$), the inner product over the whole domain (Equation (2.8)) is zero, therefore, the individual tiles in Equation (2.9) must be zero. Therefore, $\mathrm{sinc}_{\boldsymbol{M}}$ shifted to different lattice sites are orthogonal. □

When $\boldsymbol{M}\boldsymbol{k}_1 = \boldsymbol{M}\boldsymbol{k}_2$ the value of Equation (2.9) simplifies to the volume of the Brillouin zone, and it equals 1 if $|\det \boldsymbol{M}| = 1$. Then we have:

$$\langle \mathrm{sinc}_{\boldsymbol{M}}(\cdot - \boldsymbol{M}\boldsymbol{k}_1), \mathrm{sinc}_{\boldsymbol{M}}(\cdot - \boldsymbol{M}\boldsymbol{k}_2) \rangle = \delta_{\boldsymbol{M}\boldsymbol{k}_1 - \boldsymbol{M}\boldsymbol{k}_2}. \tag{2.10}$$

The orthonormality of the $\mathrm{sinc}_{\boldsymbol{M}}$ system defined readily extends the univariate sampling and ideal interpolation theory to the multivariate case.

2.2.1 Why Are Hexagonal and BCC Lattices Optimal?

The optimality of the hexagonal lattice in sampling theory can be explained by the *uncertainty principle*. As mentioned earlier, when a function is sampled in the space domain, its spectrum is replicated in the Fourier domain on a lattice with *reciprocal* density of the sampling set.

When a function is sampled coarser than the Nyquist rate, the replicates of the spectrum overlap in the Fourier domain and lead to the so called *aliasing* effect. Too dense of a sampling leads to excessive and unnecessary amounts of data; this phenomenon is attributed to a sparse replication of replicates of the spectrum. The best sampling rate is achieved when the spectrum of the function is *packed as dense as possible* (in the Fourier domain) without overlap. Figure 2.2 illustrates the uncertainty principle in the univariate case.

Petersen and Middleton [72] developed the multidimensional version of the sampling theorem of Whittaker and Shannon. They further noticed that for isotropic functions (i.e., functions with roughly low-pass isotropic spectrum) the optimal sampling rate in the multidimensional setting, can be obtained by the dense (hyper-) sphere packing arrangement which optimally packs the replicates of the spectrum in the Fourier domain. Such a dense packing of the replicates in the Fourier domain provides the notion of an optimal sampling lattice as the dual lattice to the dense sphere packing lattice.

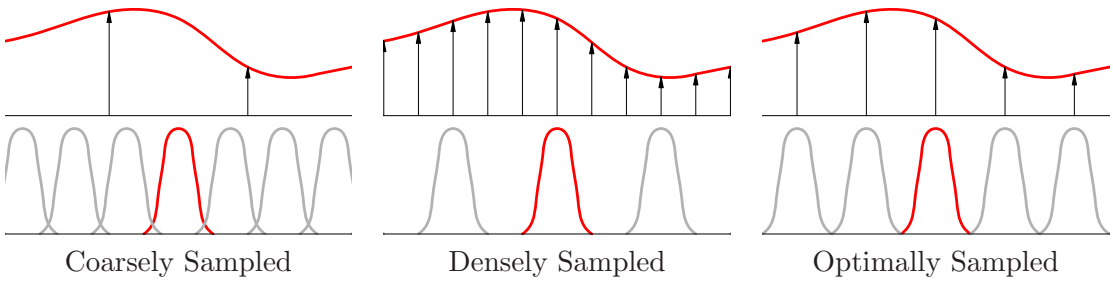


Figure 2.2: Uncertainty principle in the univariate setting. First row: sampling a function in the space domain. Second row: replication of the spectrum in the Fourier domain. The first column illustrates aliasing, the second column is dense sampling with too many unnecessary samples, the third column is the ideal sampling rate.

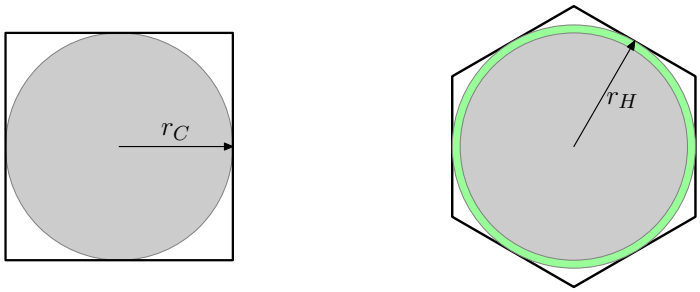
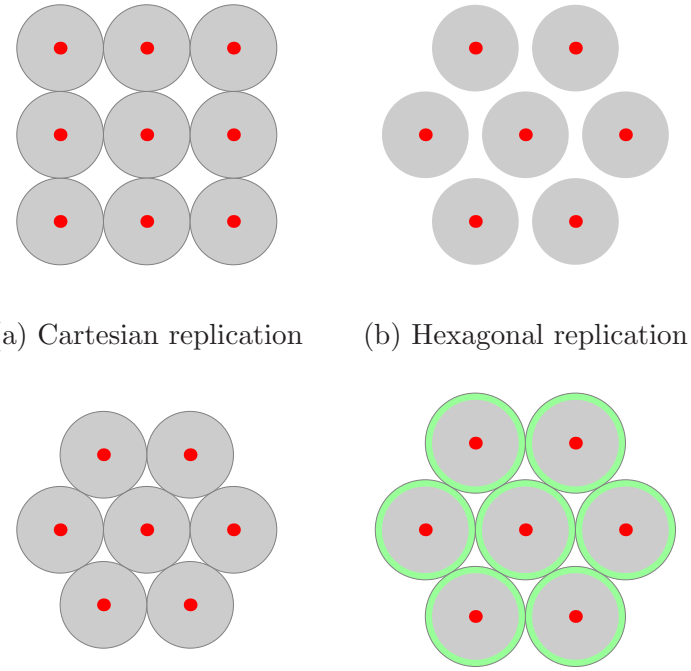


Figure 2.3: A square and a hexagon with unit area corresponding to the Brillouin zone of Cartesian and hexagonal sampling. The area of inscribing disk to a square is about 14% less than the area of the inscribing disk to the hexagon.

Since the density of hexagonally packing disks in 2-D is roughly 14% more than the regular Cartesian, one can retain roughly 14% more wave-number frequency information in the spectrum of the function without introducing aliasing [72] (see Figure 2.4). For a sampling scheme with unit sampling density (one sample per unit area), the area of the Brillouin zone is 1. For Cartesian sampling, the Brillouin zone is a square of unit area 1 and for the comparable hexagonal sampling it is a hexagon with unit area. For an isotropic sampling, one needs to band-limit the function to the radius of *inscribing* radius of the Brillouin zone. While the inscribing radius for the aforesaid square is $r_C = .5$, the inscribing radius for the corresponding hexagon is $r_H \approx .537$ (see Figure 2.3). This larger radius allows for inclusion of roughly 14% more information in the spectrum without introducing any aliasing since $\frac{r_C^2}{r_H^2} \approx .866$.

The optimal packing of spheres in 3-D is obtained by FCC packing whose packing density is 74.05%. In comparison to the packing density of the Cartesian lattice (52.3%), the FCC



(a) Cartesian replication (b) Hexagonal replication

(c) Hex: Fewer samples (d) Hex: More information

Figure 2.4: (a) Cartesian replication of the spectrum versus (b) Hexagonal replication with the same sampling density in space domain. (c) One could represent the same spectrum with tighter replication in the Fourier domain, therefore sparser sampling in the space domain. (d) One could add extra wave-number information (in the highlighted region) and still avoid aliasing.

replication of the spectrum has to lose about 30% of the information in its Voronoi cell for an alias-free sampling on the Cartesian lattice. Therefore, the dual to the FCC lattice, the BCC lattice, is considered the best generic sampling pattern in 3-D. For a unit sampling density, the volume of Voronoi cells in the Fourier domain is 1. The Brillouin zone of the BCC lattice is a rhombic dodecahedron with unit volume and the Brillouin zone of the corresponding Cartesian lattice is a cube of unit volume. As tabulated in Table A.1 in appendix A, the radius of the inscribing sphere of the rhombic dodecahedron is $r_B = .561$ while the corresponding radius of a cube is $r_C = .5$. Therefore, the ratio between the volume of the inscribing spheres is $\frac{r_C^3}{r_B^3} \approx .708$. Hence, one can capture potentially up to 30% more information in the spectrum of the BCC sampling lattice.

For the non-ideal band-limited cases, one can argue that optimal sampling still does a better job of capturing the spectrum of the function. Based on the assumption of decaying magnitude of the Fourier transform of the function one can argue that the optimal sampling strategy captures wave-modes with higher magnitudes than what Cartesian sampling captures. This is illustrated in Figure 2.5. In this figure, the darker bands indicate a higher magnitude of the Fourier transform of the function at lower wave-numbers, while the lighter bands indicate lower magnitudes at higher wave-numbers. The red band indicates the Brillouin zone of Cartesian sampling and the blue hexagon indicates the Brillouin zone of hexagonal sampling. The red band includes high wave-numbers hence low-magnitude areas at its corners. One can argue that the hexagon has covered the "more important" frequencies since it covers wave-numbers that are closer to the origin. One can quantitatively show that the integral of squared distances of points to the center is smaller for a hexagon than the corresponding square [17]. Therefore, the hexagon covers frequencies that are in general closer to the center. Based on the assumption of decay of magnitude in Fourier transform, one can conclude that the hexagonal region captures the "more important" frequencies than the Cartesian region.

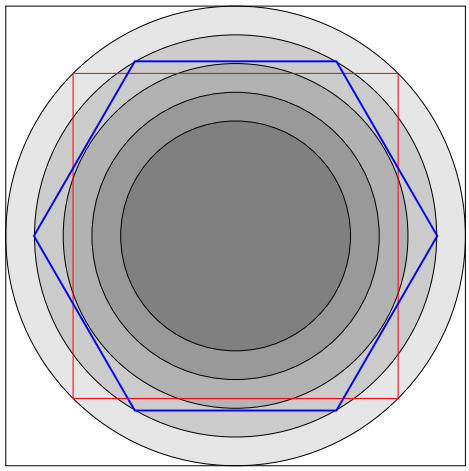


Figure 2.5: Red band indicates the Brillouin zone of a Cartesian sampling while the blue band indicates the Brillouin zone of the corresponding hexagonal sampling. The areas of the red square and the blue hexagon are the same. Since the sum (integral) of distances to the origin is smaller in the hexagonal region, "more" of the frequencies closer to the origin are covered in the hexagonal case.

2.3 Multivariate Reconstruction

Reconstruction from sampled data refers to the procedure of interpolating or approximating the underlying continuous-domain function. In this section we will see the ideal interpolation function for reconstructing in the space of band-limited functions sampled on the BCC lattice. As mentioned earlier, the ideal interpolation function has an unbounded support and is not suitable for computational applications. Therefore, we will focus on the reconstruction functions with bounded support.

2.3.1 Ideal Interpolation on the BCC Lattice

As we have illustrated before, the spectrum of a function sampled by a lattice $\boldsymbol{M}$, is contained in the Voronoi cell of the dual lattice, i.e., the Brillouin zone. The canonical ideal interpolating function we defined, $\mathrm{sinc}_{\boldsymbol{M}}$, has a Fourier transform which is the indicator function of the Brillouin zone of the lattice $\boldsymbol{M}$. In this section, we shall derive, explicitly, the ideal interpolation function on the BCC lattice, described in Equation (1.1): sincBCC.

The Brillouin zone of the BCC lattice is a rhombic dodecahedron; i.e., the Voronoi cell of the FCC lattice. In order to construct a function whose Fourier transform is the

indicator function of the rhombic dodecahedron, we note that the rhombic dodecahedron can be cloven into four 3-D parallelepipeds as illustrated in Figure 2.6. This decomposition process can be visualized by moving the four parallelepipeds along the vectors $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_4$ illustrated in the figure:

$$\boldsymbol{\xi}_1 = \begin{bmatrix} 1/2 \\ -1/2 \\ -1/2 \end{bmatrix}, \boldsymbol{\xi}_2 = \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \end{bmatrix}, \boldsymbol{\xi}_3 = \begin{bmatrix} -1/2 \\ -1/2 \\ 1/2 \end{bmatrix}, \boldsymbol{\xi}_4 = \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}.$$

Note that $\sum_{k=1}^{4} \boldsymbol{\xi}_k = \mathbf{0}$. We further note that the parallelepiped that is moving along $\boldsymbol{\xi}_1$ is formed at the origin, by vectors $-\boldsymbol{\xi}_2, -\boldsymbol{\xi}_3$ and $-\boldsymbol{\xi}_4$, similar to the other three parallelepipeds.

Therefore, one can construct sincBCC by superposition of four functions whose Fourier transforms are the indicator functions of the parallelepipeds illustrated in Figure 2.6. To derive such functions, we first note that $\frac{1-\exp(\mathrm{i}x)}{\mathrm{i}x} \longleftrightarrow \chi_{[0,1)}(\omega)$; therefore, by tensor-product, one can construct the indicator function of the unit cube in the frequency domain. Let $\chi(\boldsymbol{\omega})$ denote the indicator function of the unit cube ($[0,1)^3$) in the frequency domain, and $S(\boldsymbol{x})$ for $\boldsymbol{x} = (x, y, z)$, its Fourier pair:

$$S(\boldsymbol{x}) := \frac{1-\exp(\mathrm{i}x)}{\mathrm{i}x} \frac{1-\exp(\mathrm{i}y)}{\mathrm{i}y} \frac{1-\exp(\mathrm{i}z)}{\mathrm{i}z} \longleftrightarrow \chi(\boldsymbol{\omega}).$$

Recall that $\operatorname{sinc}(t) = \frac{\sin(t/2)}{(t/2)}$. Since $\frac{1-\exp(\mathrm{i}x)}{\mathrm{i}x} = -\exp(\mathrm{i}x/2)\operatorname{sinc}(x)$, we have

$$S(\boldsymbol{x}) = -\exp(\mathrm{i}(x+y+z)/2)\operatorname{sinc}(x)\operatorname{sinc}(y)\operatorname{sinc}(z).$$

We need to transform the unit cube to the parallelepiped formed by the above mentioned vectors. For transforming the unit cube to the parallelepiped formed by the above vectors, we introduce the matrix $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1\boldsymbol{\xi}_2\boldsymbol{\xi}_3\boldsymbol{\xi}_4]$ and the operation $\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1 = [\boldsymbol{\xi}_2\boldsymbol{\xi}_3\boldsymbol{\xi}_4]$. Then the indicator function of the parallelepiped formed by $-\boldsymbol{\xi}_2, -\boldsymbol{\xi}_3$ and $-\boldsymbol{\xi}_4$ is $\chi\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{-1}\boldsymbol{\omega}\right)$. If $\boldsymbol{Q}$ is an invertible matrix, we know that $f(\boldsymbol{Q}\boldsymbol{x}) \longleftrightarrow \hat{f}((\boldsymbol{Q}^{-1})^{\mathrm{T}}\boldsymbol{\omega})/|\det \boldsymbol{Q}|$. Hence we have:

$$S\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{\mathrm{T}}\boldsymbol{x}\right) \longleftrightarrow \chi\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{-1}\boldsymbol{\omega}\right)/(\frac{1}{2})$$

where $S\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{\mathrm{T}}\boldsymbol{x}\right) = \exp\left(-\mathrm{i}/2\left(\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{x}\right)\right)\operatorname{sinc}\left(\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{x}\right)$. Since $\sum_{k=1}^{4} \boldsymbol{\xi}_k = \mathbf{0}$, we have:

$$S\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{\mathrm{T}}\boldsymbol{x}\right) = \exp\left(\mathrm{i}/2\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{x}\right).$$

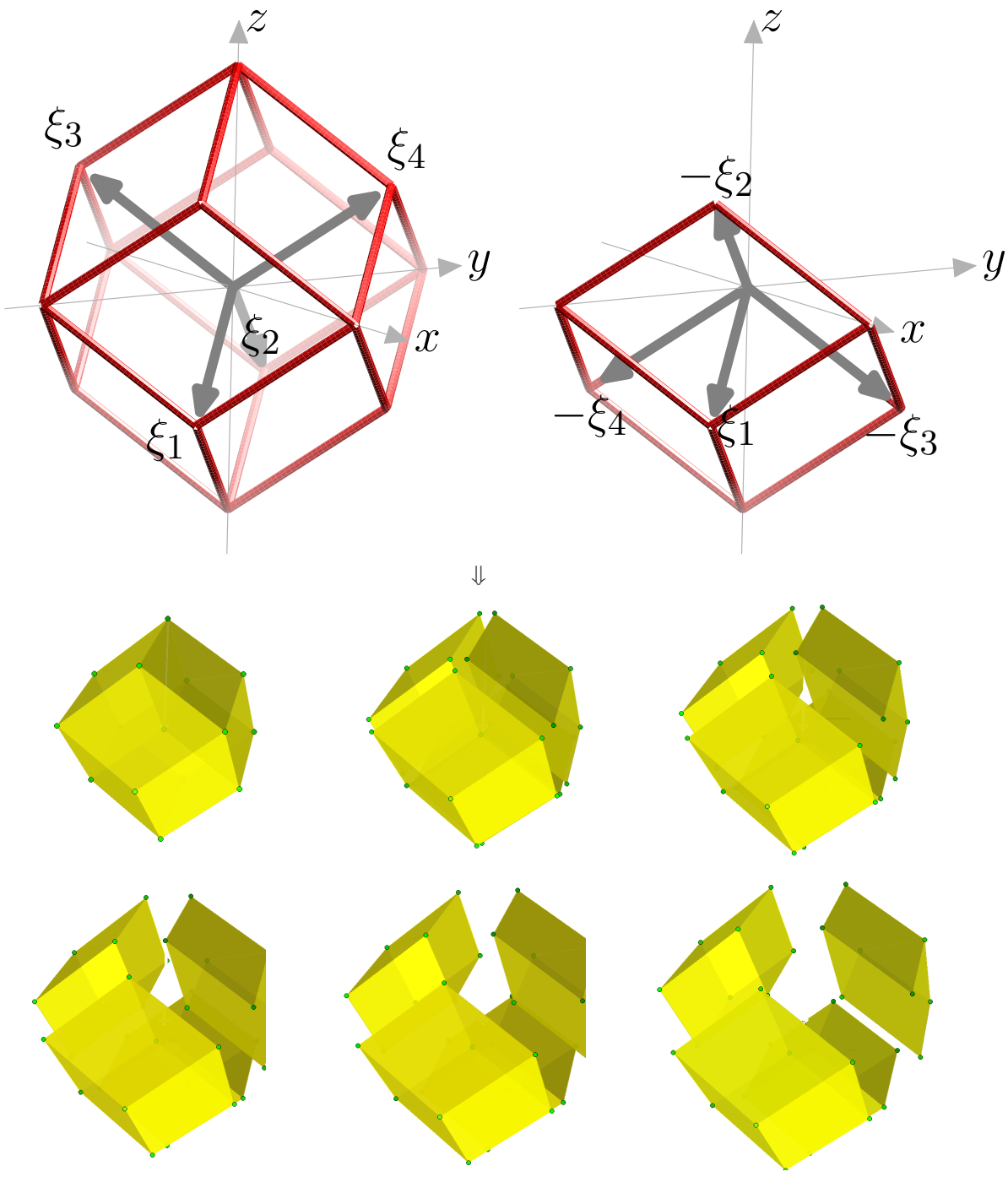


Figure 2.6: A rhombic dodecahedron can be divided into four 3-D parallelepipeds.

Now, we can write the sincBCC as the superposition of S-functions:

$$\text{sincBCC}\left(\boldsymbol{x}\right) = \frac{1}{2}\sum_{k=1}^{4} S\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_k\right)^{\mathrm{T}}\boldsymbol{x}\right). \tag{2.11}$$

Since the rhombic dodecahedron has point symmetry with respect to the origin, it makes sense for the sincBCC to be a real valued function; therefore, we consider the conjugate $\overline{S}$ of the function S and subtract it from S:

$$\begin{aligned} S\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{\mathrm{T}}\boldsymbol{x}\right) - \overline{S}\left(-\left(\boldsymbol{\Xi}\backslash\boldsymbol{\xi}_1\right)^{\mathrm{T}}\boldsymbol{x}\right) =& 2\mathrm{i}\sin\left(\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{x}/2\right)\operatorname{sinc}\left(\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{x}\right)\operatorname{sinc}\left(\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{x}\right) \\ =& \mathrm{i}\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{x}\prod_{m=1}^{4}\operatorname{sinc}\left(\boldsymbol{\xi}_m^{\mathrm{T}}\boldsymbol{x}\right). \end{aligned}$$

Adding the four terms of sincBCC, we get:

$$\begin{aligned} \text{sincBCC}\left(\boldsymbol{x}\right) - \overline{\text{sincBCC}}(\boldsymbol{x}) =& \frac{1}{2}\sum_{k=1}^{4}\left(\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{x}\prod_{m=1}^{4}\operatorname{sinc}\left(\boldsymbol{\xi}_m^{\mathrm{T}}\boldsymbol{x}\right)\right) \\ =& \frac{1}{2}\mathrm{i}\left(\prod_{m=1}^{4}\operatorname{sinc}\left(\boldsymbol{\xi}_m^{\mathrm{T}}\boldsymbol{x}\right)\right)\sum_{k=1}^{4}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{x} \\ =& 0, \end{aligned}$$

since $\sum_{k=1}^{4}\boldsymbol{\xi}_k = \mathbf{0}$. Therefore, we have the explicit real valued function sincBCC:

$$\text{sincBCC}\left(\boldsymbol{x}\right) = \frac{1}{2}\sum_{k=1}^{4}\left(\cos\left(\frac{\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{x}}{2}\right)\prod_{m\neq k}\operatorname{sinc}\left(\boldsymbol{\xi}_m^{\mathrm{T}}\boldsymbol{x}\right)\right). \tag{2.12}$$

It is interesting to note that any zonotope can be decomposed into parallelopipeds (see [24, I.53]). Therefore, similar to the above derivation, sinc functions for other lattices can be explicitly derived using such decompositions.

2.3.2 Finite Multivariate Reconstruction

Traditionally, the design of reconstruction *filters* is a rich area in signal processing. This approach is discrete-to-discrete and thus provides estimates of the function values on another regular sampling lattice. Typically, constraints in the frequency domain are used to guide the filter design process (e.g., [9, 70, 79]). On the other hand, in shift-invariant approximation spaces, one considers spaces that are generated with a finite (small) set of continuous-domain

functions (i.e., generators, or 'kernel' in the image processing domain). This model allows to recover/reconstruct a function with best error norms within a space of choice. Space domain analysis of functions such as smoothness and numerical accuracy (approximation order) guide the process of designing reconstruction schemes in graphics and image processing domains [66, 68, 77].

These well-known solutions are 1-D, and for image processing and volume rendering they are extended to multiple dimensions through a separable extension (often called tensor-product approach) or through a spherical extension (e.g., McClellan transformation [56], or Radial Functions). The problem of these extensions is that they do not deal well with the multi-dimensional nature and geometry of the sampling lattice, in particular for non-Cartesian lattices [94]. The separable extension, clearly, is only satisfactory for the Cartesian lattice, for which the lattice's Nyquist region coincides with the kernel's low-pass region in the frequency domain. The spherical extension, however, has difficulties imposing zero-crossing in the frequency domain at the dual-lattice points, which is crucial to guarantee approximation order and polynomial reproduction.

In the field of approximation theory there has been a great deal of effort on approximating multivariate functions. In particular, Birkhoff and de Boor discuss multivariate splines that are tensor-products of one dimensional splines [3, 5, 4].

In the tensor-product approach, when approximating a function over a rectangular domain in $\mathbb{R}^n$, splines are constructed as a tensor-product of splines of degree s in any of the variables and the tensor-product is of total degree ns. Zwart [117] considered a different family of multivariate splines which aren't necessarily a tensor-product, but rather they are piecewise polynomial functions where each piece is a polynomial of total degree s. The well known Zwart-Powell element was one of the example splines where the reconstruction kernel is not a tensor-product of univariate splines. Rather it has non-degenerate partitions and on each partition it is a polynomial of second degree. This element gained popularity as an example of a nontrivial bivariate box spline [24]. Powell also devised the corresponding subdivision scheme to reconstruct functions with continuous first derivatives that was widely used for contour drawing programs [73]. The significance of the ZP element is that it achieves a C^1 smooth reconstruction with only a polynomial of degree two in 2-D. A tensor-product solution can only do the job with a polynomial of degree four. Hence the ZP element stands out among the low degree splines that achieve the maximal smoothness.

Another interesting example in 2-D is the class of "hex-splines", which form a B-spline

family for hexagonally sampled data [109]. The first-order hex-spline is defined as the indicator function of the Voronoi cell and is thus non-separable. Higher-order hex-splines are defined in terms of successive convolutions of the first-order function. Analytical formulae are available both in the frequency and the spatial domains.

In volume graphics, optimality of BCC sampling has been explored by Theußl et al. [95]. They applied the spherical extension of reconstruction filters, which resulted in rather blurry and unsatisfying results. Different ad-hoc approaches were studied for reconstruction and derivative reconstruction on the BCC lattice, with mixed results [94]. Also, iso-surface extraction on the optimal sampling lattice has been studied with inconclusive results [10]. Recently, Csébfalvi [21] demonstrated a reconstruction using a Gaussian kernel and the principle of generalized interpolation of Thévenaz et al. [96]. While this method provides an isotropic solution, it does not guarantee approximation order. It is also a numerical scheme without any closed form solution for the interpolation kernel.

McCool [57, 58, 59] has explored the use of polyhedral splines in computer graphics. In his work, the flexiblity of polyhedral splines (tailored to the geometry of the domain) are exploited for accurate (anti-aliased) computation of integrals involved during the rendering process.

2.3.3 Box Splines

Box splines offer a mathematically elegant framework for constructing a class of multi-dimensional elements with flexible shape and support that can be non-separable in a natural way. The general topic of box splines is rather intricate and a general survey of results on the topic has been gathered in [24]. In the next chapters we present a set of trivariate box splines with proper geometry for reconstruction on the BCC and FCC lattices. Here, we begin by briefly introducing box splines and state their properties that will be useful in our discussions.

Definitions and Properties of Box Splines

A box spline is characterized by a set of direction vectors that indicate its construction by successive convolution of line segments along these vectors. The linear combination of shifts of a box spline generates a spline whose smoothness and ability to approximate continuous functions also depend on these direction vectors. Notationally, the direction vectors are

usually gathered in a matrix; i.e., a box spline in $\mathbb{R}^s$ is specified by $n \geq s$ vectors in $\mathbb{R}^s$ that are columns of its matrix $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n]$. The support of the box spline is all points $\boldsymbol{x} \in \mathbb{R}^s$ such that $\boldsymbol{x} = \boldsymbol{\Xi}\boldsymbol{t}$ where $\boldsymbol{t} \in \mathbb{R}^n$ and $0 \leq t_k \leq 1$ for $1 \leq k \leq n$. In other words, the support of a box spline is the image of the corresponding unit cube or 'box' under its matrix $\boldsymbol{\Xi}$.

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

$$M_{\boldsymbol{\Xi}}(\boldsymbol{x}) = \begin{cases} \frac{1}{|\det \boldsymbol{\Xi}|} & \text{where } \boldsymbol{x} = \boldsymbol{\Xi}\boldsymbol{t} \text{ and } \boldsymbol{t} \in [0,1)^n \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

Clearly, the box spline from Equation (2.13) is discontinuous at the boundary of its support. Its 1-D version is the box function that is simply the indicator function for the interval $[0, 1)$.

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

$$M_{[\boldsymbol{\Xi},\boldsymbol{\xi}]}(\boldsymbol{x}) = \int_0^1 M_{\boldsymbol{\Xi}}(\boldsymbol{x} - t\boldsymbol{\xi})dt. \tag{2.14}$$

This inductive definition implies that starting from the base case as in Equation (2.13) the indicator function is smeared *along* the additional direction vector. Hence, the convolution of two box splines is yet another box spline:

$$M_{\boldsymbol{\Xi}_1} * M_{\boldsymbol{\Xi}_2}(\boldsymbol{x}) = M_{[\boldsymbol{\Xi}_1,\boldsymbol{\Xi}_2]}(\boldsymbol{x}). \tag{2.15}$$

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

$$\hat{M}_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) = \prod_{\boldsymbol{\xi} \in \boldsymbol{\Xi}} \frac{1 - \exp(-\mathrm{i}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\omega})}{\mathrm{i}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\omega}} \tag{2.16}$$

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

$$\boldsymbol{\Xi}_0 = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

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

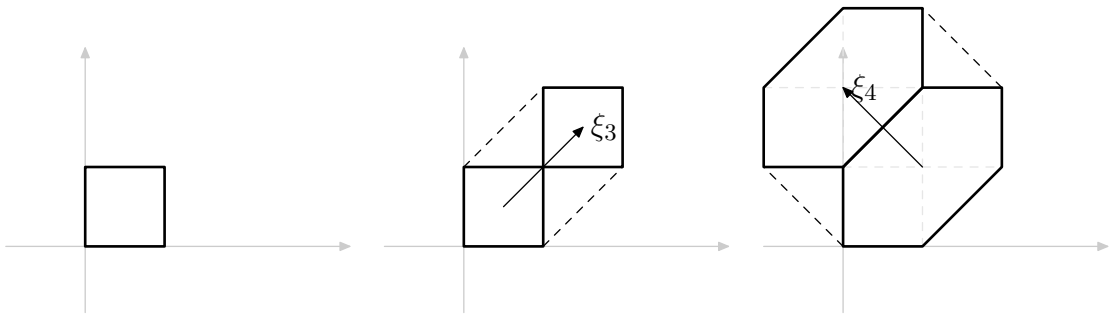


Figure 2.7: Construction of the linear element and the Zwart-Powell element from the simplest box spline.

Adding a direction vector of $\boldsymbol{\xi}_3 = [1\ 1]^{\mathrm{T}}$ to $\boldsymbol{\Xi}_0$ smears the unit square across its diagonal. This is illustrated in Figure 2.7. As the basic box spline is a constant function on the unit square, the result of smearing it along the diagonal produces a linear box spline that is represented by $[\boldsymbol{\Xi}_0, \boldsymbol{\xi}_3]$. The support of this box spline is illustrated in Figure 2.7(b). This box spline is a bivariate piecewise polynomial of degree one. This box spline generates a C^0 spline function space, as $\rho = 2$. Adding one more direction vector to the above box spline produces a quadratic box spline. The choice of $\boldsymbol{\xi}_4 = [-1\ 1]^{\mathrm{T}}$ produces a symmetric octagonal shape as the support of this quadratic box spline which is known as the Zwart-Powell (ZP) element. The process of this convolution is illustrated in Figure 2.7(c). The ZP element contains bivariate polynomials that are only of second degree, yet it achieves a C^1 reconstruction in 2-D ($\rho = 3$).

Chapter 3

Box Spline Reconstruction on the BCC Lattice

In this chapter, we present accurate and efficient reconstruction methods for the BCC lattice. Such reconstructions have been sought by the volume graphics community [95, 94, 10] to better exploit the theoretical advantages of the BCC lattice. Several contributions are proposed:

- We establish a four-direction box spline that is geometrically tailored to the BCC lattice in Section 3.1. The linear box spline is a 3-D piecewise linear function. Higher-order versions are obtained by successive convolutions. This way, we can choose the required smoothness and approximation power.

- We explicitly characterize the polynomial pieces defining these box splines, which is detailed in Section 3.2. Our characterization method is general (for any number of repetitions) and leads to polynomial expressions that can be implemented to evaluate the box spline at arbitrary points. Specifically, we derive the explicit expressions for the C^0 and C^2 members of our family of BCC-box-splines, since they are the most relevant for the practitioner in the graphics and visualization domain.

- We demonstrate that our box splines (for C^0 and C^2 continuity) on the BCC lattice are computationally *twice* as efficient as tensor-product B-splines on the Cartesian lattice (for comparable smoothness and the same sampling density); see Table 3.1. Based on these results, in Section 3.3, we conclude that BCC lattice sampling can be

more attractive not only on a theoretical level, but also in practical settings.

The simplest interpolation kernel on any lattice is the indicator function of the Voronoi cell of the lattice. The corresponding interpolation scheme is the generalization of the so-called nearest neighbor interpolation. The Voronoi cell of the Cartesian lattice is a cube and the Voronoi cell of the BCC lattice is a truncated octahedron as shown in Figure 3.1(a). Let $\chi_{\boldsymbol{BCC}}$ indicate the indicator function of the truncated octahedron, the Voronoi cell of the BCC lattice described by Equation (1.1). This interpolation kernel is normalized to:

$$\int \chi_{\boldsymbol{BCC}}(\boldsymbol{x}) d\boldsymbol{x} = |\det \boldsymbol{BCC}| = 4, \tag{3.1}$$

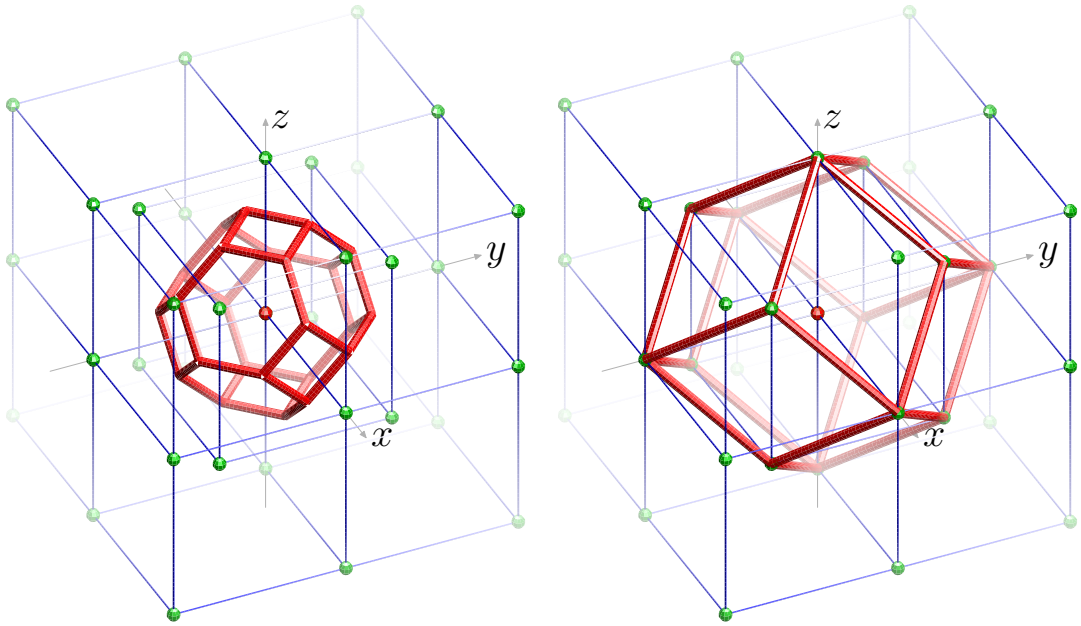
so that the energy between the discrete/continuous model is consistent. This is due to the fact that the sampling density of the BCC lattice described by Equation (1.1) is $\frac{1}{4}$ samples per unit volume.

More sophisticated reconstruction kernels involve information from the neighboring points of a given lattice point. We are also interested in the cell formed by the immediate neighbors of a lattice point. The first neighbors of a lattice point are defined via the Delaunay tetrahedralization of the lattice; a point $\boldsymbol{q}$ is a **first neighbor** of $\boldsymbol{p}$ if their respective Voronoi cells share a (non-degenerate) face. The **first neighbors cell** is the polyhedron whose vertices are the first neighbors. Again, this cell is the same for all points on the lattice.

For example, by this definition there are six first neighbors of a point in a Cartesian lattice; the first neighbors cell for the Cartesian lattice is the octahedron. For the BCC lattice there are fourteen first neighbors for each lattice point. The first neighbor cell is a rhombic dodecahedron as depicted in Figure 3.1(b).

3.1 Four-Direction Box Spline on BCC

The construction of box splines dedicated to the BCC lattice is guided by the fact that the rhombic dodecahedron (the first neighbors cell of the BCC lattice, see Figure 3.1(b)) is the 3-D shadow of a 4-D hypercube (tesseract) along its antipodal axis. This construction is a generalization of the 2-D linear box spline with hexagonal support, which can be obtained by projecting a 3-D cube along its antipodal axis; see Figure 3.2(b).



(a) (b)

Figure 3.1: (a) The Voronoi cell of the BCC lattice is the truncated octahedron. (b) The first neighbors cell is a rhombic dodecahedron.

3.1.1 A Geometric Construction

Integrating a constant tesseract along its antipodal axis yields a function that has a rhombic dodecahedron support (see Figure 3.1(b)), has its maximum value at the center and has a *linear* falloff towards the fourteen first-neighbor vertices. Since it arises from the projection of a higher dimensional box, this function serves as the linear box spline interpolation kernel on the BCC lattice.

Let $\mathcal{B}$ denote the box function. The characteristic function of the unit tesseract is given by a tensor-product of four $\mathcal{B}$ functions on each axis. By projecting the unit tesseract, one obtains a rhombic dodecahedron whose geometric scale is only half of the first-neighbors cell of the BCC lattice described by Equation (1.1). In this BCC lattice, with integer lattice coordinates, the first-neighbors cell is scaled such that the blue edges of Figure 3.1(b) are of length two. Therefore, we scale the geometry of the unit tesseract by two and normalize by its hyper-volume:

$$\mathcal{T}(x,y,z,w) := \frac{1}{16}\,\mathcal{B}(x/2)\,\mathcal{B}(y/2)\,\mathcal{B}(z/2)\,\mathcal{B}(w/2). \tag{3.2}$$

Let $\boldsymbol{v} = (2,2,2,2) := [2,2,2,2]^{\mathrm{T}}$ denote a vector along the antipodal axis. In order to project

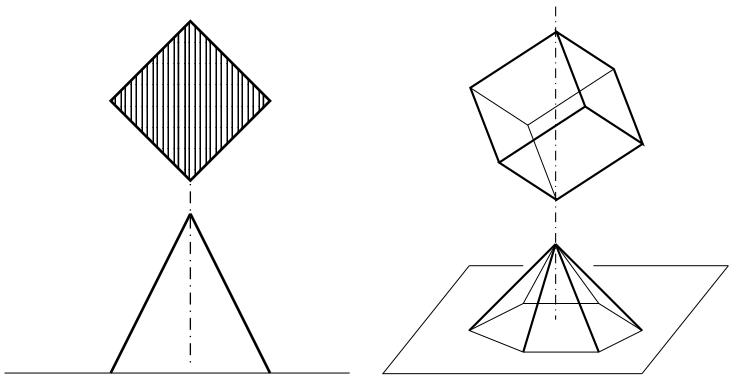


Figure 3.2: (a) 1-D linear box spline (triangle function). (b) 2-D linear hexagonal box spline.

along this axis, it is convenient to rotate it so that it aligns with the w axis:

$$\boldsymbol{R} = \frac{1}{2}\left[\boldsymbol{\rho}_1\boldsymbol{\rho}_2\boldsymbol{\rho}_3\boldsymbol{\rho}_4\right] = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.3}$$

This rotation matrix transforms $\boldsymbol{v}$ to $(0,0,0,4)$. Also by examining Equation (3.3), one can see that each vertex of the rotated tesseract, when projected along the w axis, will coincide with the origin or one of the vertices of the rhombic dodecahedron: $(\pm1,\pm1,\pm1)$, $(\pm2,0,0)$, $(0,\pm2,0)$, $(0,0,\pm2)$ or $(0,0,0)$. Let $\boldsymbol{x} = (x,y,z,w)$; now the linear box spline is given by

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

Substituting in Equation (3.2) we get

$$L_{RD}(x,y,z) = \frac{1}{16}\int \prod_{k=1}^{4} \mathcal{B}(\frac{1}{4}\boldsymbol{\rho}_k^{\mathrm{T}}\boldsymbol{x})\,dw. \tag{3.4}$$

Note that the value at the origin is $L_{RD}(0,0,0) = 1/4$ (see [24, II.8]). This is due to the fact that the box splines are normalized to $\int L_{RD}(\boldsymbol{x})d\boldsymbol{x} = 1$, whereas the sampling density of the BCC lattice ($\frac{1}{4}$) demands a kernel whose integral is 4 (see Equation (3.1)). Therefore, in order to preserve the energy in the discrete/continuous model we employ the box splines scaled by 4 on the BCC lattice. This scaling ensures that the value of the linear box spline at the center is 1 and zero at all other lattice sites. Hence the linear box spline constitutes a linear interpolator on the BCC lattice.

3.1.2 Fourier Transform

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

From the projection-based construction of the rhombic dodecahedron discussed earlier, we can derive the Fourier transform of the linear box spline function described by Equation (3.4). From Equation (3.2), it is evident that the Fourier transform of the characteristic function of the tesseract is given by the tensor-product:

$$\begin{aligned}\hat{\mathcal{T}}(\omega_x,\omega_y,\omega_z,\omega_w) =& \frac{1-\exp\left(-\mathrm{i}2\omega_x\right)}{\mathrm{i}2\omega_x}\frac{1-\exp\left(-\mathrm{i}2\omega_y\right)}{\mathrm{i}2\omega_y}\\ &\frac{1-\exp\left(-\mathrm{i}2\omega_z\right)}{\mathrm{i}2\omega_z}\frac{1-\exp\left(-\mathrm{i}2\omega_w\right)}{\mathrm{i}2\omega_w}\end{aligned}$$

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

By the Fourier slice-projection theorem, projecting a function along a certain direction in the spatial domain amounts to slicing its Fourier transform perpendicular to the direction of projection. This slice runs through the origin. Again, we make use of the rotation (3.3) to align the projection axis with the w axis. Thus in the frequency domain we take the slice $\omega_w = 0$.

It is convenient to introduce the 3×4 matrix

$$\boldsymbol{\Xi} = [\boldsymbol{\xi}_1\boldsymbol{\xi}_2\boldsymbol{\xi}_3\boldsymbol{\xi}_4] = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \tag{3.5}$$

given by the first three rows of the rotation matrix $\boldsymbol{R}$ of Equation (3.3). The reason for omitting the last row is that we are taking a slice $\omega_w = 0$ orthogonal to the fourth axis at the origin. The Fourier transform of the linear box spline can now be written as

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

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

$$\hat{M}_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) = \prod_{k=1}^{4} \frac{1-\exp\left(-\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}\right)}{\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}}. \tag{3.6}$$

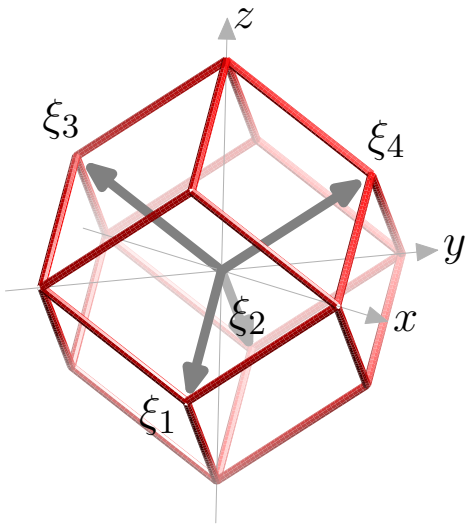


Figure 3.3: The support of the box spline represented by Ξ is a rhombic dodecahedron formed by the four direction vectors in Ξ.

Since any three of the vectors in Ξ span $\mathbb{R}^3$, at least two vectors need to be removed from Ξ so the remaining vectors do not span, therefore $\rho = 2$. Hence this box spline is guaranteed to produce a C^0 reconstruction. We can verify the vanishing moments (zero crossings) of the frequency response at the aliasing frequencies on the FCC lattice points. We first note that $\sum_{k=1}^{4} \boldsymbol{\xi}_k = \mathbf{0}$; therefore, the center of the box spline M_Ξ is at the origin [24] and the Fourier transform can be written as:

$$\hat{M}_\Xi(\boldsymbol{\omega}) = \prod_{k=1}^{4} \operatorname{sinc}\left(\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}\right).$$

Recall that $\operatorname{sinc}(t) = \frac{\sin(t/2)}{t/2}$. This re-formulation provides a more convenient form to verify zero crossings. Due to the checkerboard property (see Section 1.1.2), for every FCC lattice point the sum of its coordinates is always even. Since the FCC lattice dual to the BCC lattice of our discussion is scaled by π (according to Equation (1.1)), for $\boldsymbol{\omega}$ on the FCC lattice, $\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{\omega} = (\omega_x + \omega_y + \omega_z) = 2\pi k$ for some $k \in \mathbb{Z}$; therefore, $\operatorname{sinc}(\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{\omega}) = 0$ on all of the aliasing frequencies. Since $\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{\omega} = -\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{\omega} - \boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{\omega} - \boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{\omega}$, at least one of the $\boldsymbol{\xi}_m^{\mathrm{T}}\boldsymbol{\omega}$ for $m = 1, 2, 3$ needs to also be an even multiple of π since the sum of three odd multiples of π can not be an even multiple. For such k we have $\operatorname{sinc}(\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}) = 0$; therefore, there is a zero of order at least two at each aliasing frequency, yielding a C^0 kernel whose approximation power is two on the BCC lattice [90]. This smoothness and approximation power parallels that of

the trilinear B-spline interpolation on the Cartesian lattice.

3.1.3 Higher-Order Box Splines

The number of vanishing moments can be doubled by convolving the linear box spline with itself. Hence the resulting reconstruction kernel will have twice the approximation power on the BCC lattice due to the Strang-Fix result [90]. As noted before, the resulting box spline can then be represented by $\boldsymbol{\Xi}^{[2]} := [\boldsymbol{\Xi}, \boldsymbol{\Xi}]$ where every direction vector is duplicated.

An equivalent method of deriving this function would be to convolve the constant function on the tesseract with itself and project the resulting distribution along a diagonal axis (this commutation of convolution and projection is easy to understand in terms of the corresponding operators in the frequency domain – see Section 3.1.2). Convolving the constant function on the tesseract with itself results in another function supported on a tesseract which is the tensor-product of four 1-D triangle (linear B-spline) functions. Let Λ denote the triangle function. Then the convolution yields

$$\mathcal{T}_c(x, y, z, w) = \frac{1}{16}\,\Lambda(\frac{1}{2}x)\,\Lambda(\frac{1}{2}y)\,\Lambda(\frac{1}{2}z)\,\Lambda(\frac{1}{2}w). \tag{3.7}$$

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

$$C_{RD}(x, y, z) = \frac{1}{16}\int \prod_{k=1}^{4} \Lambda(\frac{1}{4}\boldsymbol{\rho}_k^{\mathrm{T}}\boldsymbol{x})\,dw. \tag{3.8}$$

Similar to the linear case, we use the matrix $\boldsymbol{\Xi}^{[2]}$ to represent this properly scaled box spline. Since convolution in the space domain amounts to a multiplication in the frequency domain, we use Equation (3.6) to derive the Fourier transform of the new box spline:

$$\hat{M}_{\boldsymbol{\Xi}^{[2]}}(\boldsymbol{\omega}) = \hat{M}^2_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) = \prod_{k=1}^{4}\left(\frac{1-\exp\left(-\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}\right)}{\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}}\right)^2. \tag{3.9}$$

We can see that the number of vanishing moments of this box spline are doubled when compared to the linear kernel. This implies that this box spline has fourth order approximation power on the BCC lattice [90]. The eight directions (counting multiplicities) of this box spline $\boldsymbol{\Xi}^{[2]}$ are duplicates of the original four directions. Consequently, the minimum number of directions that one needs to remove from $\boldsymbol{\Xi}^{[2]}$ so that the remaining vectors do not span $\mathbb{R}^3$ is $\rho = 4$; hence, weighted shifts of this box spline are guaranteed to produce C^2 continuous reconstructions with fourth order approximation. This smoothness and approximation

power parallels that of the tricubic B-spline reconstruction on the Cartesian lattice; for this reason we have referred to $M_{\boldsymbol{\Xi}^{[2]}}$, as the 'cubic' box spline in [30]. However, since there are eight directions, this trivariate box spline is composed of quintic polynomials. Therefore, in accordance with [24], we will call this box spline a quintic box spline.

As we noted earlier $M_{\boldsymbol{\Xi}}$ is of second order approximation power on the BCC lattice. The $(n-1)$-fold convolution of the linear kernel with itself, denoted by $M_{\boldsymbol{\Xi}^{[n]}}$, will have approximation power of $2n$ on the BCC lattice. These box splines would produce C^{2n-2} reconstructions.

3.1.4 Support

The support of $M_{\boldsymbol{\Xi}}$ is a rhombic dodecahedron as shown in Figure 3.3. The support of $M_{\boldsymbol{\Xi}^{[n]}}$ is the Minkowski sum of n rhombic dodecahedra. Since a rhombic dodecahedron is a convex and symmetric polyhedron (with respect to its center), its Minkowski addition with itself will have the same shape, scaled by two. In general the support of $M_{\boldsymbol{\Xi}^{[n]}}$ would be a rhombic dodecahedron scaled by n [115].

The volume of the support of the box spline $M_{\boldsymbol{\Xi}}$ as depicted in Figure 3.3 is 16. Therefore, for a point $\boldsymbol{x}$ in general position, 16 points from $\mathbb{Z}^3$ intersect the support of $M_{\boldsymbol{\Xi}}(\cdot - \boldsymbol{x})$ [24, II.15]. Since only 1/4 of these points belong to the BCC lattice, only 4 BCC points fall inside the support of $M_{\boldsymbol{\Xi}}(\cdot - \boldsymbol{x})$. Similarly, the support of $M_{\boldsymbol{\Xi}^{[2]}}$ is a rhombic dodecahedron whose direction vectors are scaled by two. Therefore, its volume is 128 which implies that only 32 BCC points fall inside the support of $M_{\boldsymbol{\Xi}^{[2]}}(\cdot - \boldsymbol{x})$.

This fact implies that a C^0 reconstruction with a second order approximation power on BCC only needs 4 data points[1], while for this smoothness and accuracy on the Cartesian lattice, trilinear interpolation requires a neighborhood of $2 \times 2 \times 2 = 8$ data points. Similarly, a C^2 reconstruction with a fourth order approximation on BCC only needs 32 data points, while for this smoothness and accuracy on the Cartesian lattice, tricubic B-spline requires a neighborhood of $4 \times 4 \times 4 = 64$ data points. Hence, as we will see in Section 3.3, the computational cost of BCC reconstruction is significantly lower than a similar reconstruction on the Cartesian lattice with an equivalent sampling density.

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

3.1.5 Quasi-interpolation

Considering the shifts of a box spline, a function is **represented** by a sequence $c_{\boldsymbol{k}} \in \ell_2$ and the reconstruction follows from:

$$\tilde{f} = \sum_{\boldsymbol{k}} c_{\boldsymbol{k}} M_{\boldsymbol{\Xi}}(\cdot - \boldsymbol{k}).$$

The number of zero crossings (at the aliasing frequencies) of the Fourier transform of a box spline establishes the approximation order of the box spline space. The approximation order guarantees that upon dilating the box spline space the approximation error converges to zero with a rate determined by the approximation order.

When it comes to the choice of a representation, $c_{\boldsymbol{k}}$, taking the sample values of the underlying function is the most convenient choice which is modeled by the point-evaluation (Dirac's delta) functional ($c_{\boldsymbol{k}} = \delta_{\boldsymbol{k}}(f) = f_{\boldsymbol{k}}$). This approach is often employed in image processing and visualization applications [66, 53] since the sample values are the input to the system. The resulting reconstructions are approximating (i.e., not interpolating) when the tricubic B-spline and the quintic box spline are used. Even though their spline spaces are capable of providing a fourth order approximation, these schemes do not necessarily achieve the best approximation order that is attainable by representations from the orthogonal projectors.

The approximation order of the box spline space can be realized by a **quasi-interpolation** scheme [24, III]. A quasi-interpolant $\boldsymbol{Q}$ is a linear map into the box spline space of the form

$$\boldsymbol{Q}_\lambda f(\boldsymbol{x}) := \sum_{\boldsymbol{k}} M_{\boldsymbol{\Xi}}(\boldsymbol{x} - \boldsymbol{k}) \lambda f(\cdot + \boldsymbol{k}),$$

where λ is a continuous linear functional with a bounded (and small) support. If $\boldsymbol{Q}_\lambda$ reproduces all polynomials of degrees less than r (i.e., $\boldsymbol{Q}_\lambda p = p$ for any $p \in \Pi_{<r}$), then the approximation order of the scheme based on $\boldsymbol{Q}_\lambda$ is shown to be r [24, Proposition III.4].

Here we follow the general recipe presented in [24, III] to build a generic functional for the quintic box spline. The objective is to find a continuous linear functional λ such that the corresponding quasi-interpolant reproduces all polynomials $\Pi_{<4}$ since the approximation order of the quintic box spline space is 4.

The functional λ is constructed from the values (at zero) of a series of polynomials $g_{a,b,c}$

that are mapped to (normalized) monomials (i.e., **Marsden identity**) by the box spline:

$$[\![\boldsymbol{x}]\!]^{(a,b,c)} = \sum_{\boldsymbol{k}} M_{\boldsymbol{\Xi}}(\boldsymbol{x}-\boldsymbol{k}) g_{a,b,c}(\boldsymbol{k})$$

with the trivariate normalized power function $[\![\boldsymbol{x}]\!]^{(a,b,c)} := \frac{x^a}{a!}\frac{y^b}{b!}\frac{z^c}{c!}$. Since the degree of the polynomials to be reproduced is one less than the approximation power, for the quintic box spline we consider only $g_{a,b,c}$ such that $a+b+c<4$.

The functional λ that attains the desired approximation order is derived by [24, III.13]:

$$\lambda := f \mapsto \sum_{(a,b,c)} g_{a,b,c}(\mathbf{0})(\boldsymbol{D}^{(a,b,c)} f)(\mathbf{0}). \tag{3.10}$$

Here $\boldsymbol{D}^{(a,b,c)}$ indicates a tensor-product differential operator. For convenience, one can work with a 'mesh function' $c(a,b,c) := g_{a,b,c}(\mathbf{0})$. Then according to [24, III.34], we have:

$$c(a,b,c) = [\![-\mathrm{i}\boldsymbol{D}]\!]^{(a,b,c)} \frac{1}{\hat{M}_{\boldsymbol{\Xi}}(\boldsymbol{\omega})}|_{\boldsymbol{\omega}=\mathbf{0}}.$$

Computing the values of $c(a,b,c)$ for $a+b+c<4$ is rather complicated since the derivatives of $1/\hat{M}_{\boldsymbol{\Xi}}$ are quite involved. One can resort to symbolic computing packages like Maple for computing the derivatives. In Maple, one can easily define $1/\hat{M}_{\boldsymbol{\Xi}}$ and take derivatives; however, when it comes to evaluating the derivative at zero, evaluation fails due to the presence of zeros in the numerator and the denominator. Therefore, taking the limit of the resulting expression when $\boldsymbol{\omega} \to \mathbf{0}$ is the alternative to produce correct values. However, when the matrix $\boldsymbol{\Xi}$ of the box spline has many columns (such as the case for the quintic box spline), this limiting procedure becomes complex and Maple fails to respond. Since the limit exists, one can construct a univariate expression out of the resulting expression (after taking the derivatives) by setting $\omega_x = \omega_y = \omega_z$. This approach simplifies the derivative expression and the limit to zero is then easily taken by Maple.

For the quintic box spline the non-zero elements of c are $c(0,0,0)=1$ and $c(2,0,0) = c(0,2,0) = c(0,0,2) = -1/3$. Therefore, the generic quasi-interpolation functional is:

$$\lambda_{\boldsymbol{\Xi}^{[2]}} := f \mapsto f(\mathbf{0}) - \frac{1}{3}\left((\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2})f\right)(\mathbf{0}). \tag{3.11}$$

Even though the mechanics of the polynomial reproduction is developed for Cartesian shifts of a box spline, the BCC shifts of a box spline can similarly be treated using the invertible linear transformation $\boldsymbol{P} := \boldsymbol{BCC}^{-1}$ applied to $\boldsymbol{\Xi}$ and the argument of the box spline (i.e.,

using $M_{\boldsymbol{\Xi}} = |\det \boldsymbol{P}| M_{\boldsymbol{P\Xi}}(\boldsymbol{P}\cdot))$. Since the stability of the scheme does not change the approximation order holds under this linear transformation.

The matrix of directions for the tricubic B-spline is $\boldsymbol{I}^{[4]}$ where $\boldsymbol{I}$ is the identity 3×3 matrix. The corresponding generic functional for the quasi-interpolation with the tricubic B-spline is:

$$\lambda_{\boldsymbol{I}^{[4]}} := f \mapsto f(\mathbf{0}) - \frac{1}{6}\left((\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})f\right)(\mathbf{0}). \tag{3.12}$$

3.2 Explicit Piecewise Polynomial Representation

The previous section defined the four-direction box spline on the BCC lattice and showed some of its main properties derived from its Fourier transform. However, a literal implementation of Equation (3.4) and Equation (3.8), as we implemented in [30], turned out to be extremely inefficient (especially in the case of the quintic box spline). Hence, although theoretically exciting, these splines were not useful in a practical setting. To make them practical for computer graphics and visualization applications, we derive a piecewise polynomial representation that allows an extremely fast evaluation as desired for these applications.

The most commonly used generic method for evaluating box splines at arbitrary points is through de Boor-Höllig recurrence relation [24]. Unfortunately, in reconstructions for rendering in the trivariate setting, the recursive evaluation of box splines is computationally inefficient and prone to numerical instabilities [25]. Kobbelt addresses the instability issues by delaying the evaluation of the discontinuous step function until the latest stages of recursion [45]. Even though the numerical inaccuracies of the recursive algorithm can be minimized, to make box splines practical in the field of volume graphics (e.g., volume rendering), the computational complexity of their evaluation needs to be significantly reduced. While the use of box splines in surface subdivision in graphics demands evaluations of a box spline on a fixed mesh, in the volume rendering domain one needs to evaluate a box spline at arbitrary points. For traditional B-splines, the explicit piecewise polynomial representation is commonly used for fast evaluation; therefore, we introduce a piecewise polynomial representation for the proposed box splines in Section 3.2.

Chui and Lai [12] proposed a method to derive the polynomial pieces of three- and four-direction bivariate box splines in Bézier form. Even though one can convert the Bézier form to the power form, the generalization of this method to the trivariate setting is not straight forward.

In [22], Dæhlen proposes an algorithm to evaluate a four-direction box spline on a fixed mesh shifted to an arbitrary position. Somewhat similar to our evaluation method, he relies on the relation of box splines to Dahmen's *cone splines* (truncated power functions). In Dæhlen's method, evaluation of truncated power functions is still based on a recurrence relation and is based on the connection with simplex splines. In our case, however, we derive the explicit polynomial representation of the truncated power function in Section 3.2. This representation provides us with the exact evaluation of box splines free of numerical inaccuracies since we avoid any recurrence relations. Furthermore, similar to the piecewise polynomial evaluation methods of B-splines, our method exploits the symmetries in the support of the box spline to further reduce computational cost (see Section 3.2.5).

3.2.1 Preliminaries and Outline of Derivation

In the following discussion, the symbol $\nabla_{\boldsymbol{\xi}}$ denotes a (directional) *backward-difference operator* and is defined by: $\nabla_{\boldsymbol{\xi}} f(\boldsymbol{x}) = f(\boldsymbol{x}) - f(\boldsymbol{x} - \boldsymbol{\xi})$. For a matrix of directions, $\boldsymbol{\Xi}$, the difference operator is defined as successive applications of difference operators along each direction in $\boldsymbol{\Xi}$: $\nabla_{\boldsymbol{\Xi}} = \prod_{\boldsymbol{\xi}\in\boldsymbol{\Xi}} \nabla_{\boldsymbol{\xi}}$. The corresponding differential operator is denoted by $\boldsymbol{D}_{\boldsymbol{\Xi}}$. A Green's function of a differential operator is a function g that satisfies $\boldsymbol{D}g = \delta$ where δ denotes Dirac's delta (generalized) function. The Fourier transform of δ in the distributional sense is the constant function 1.

Box splines, similar to B-splines, are piecewise polynomial functions with bounded support. In this section we will see that the box spline $M_{\boldsymbol{\Xi}}$ can be derived by applying the backward-difference operator, $\nabla_{\boldsymbol{\Xi}}$, to a single function $G_{\boldsymbol{\Xi}}$ which is a Green's function for the differential operator $\boldsymbol{D}_{\boldsymbol{\Xi}}$ corresponding to $\nabla_{\boldsymbol{\Xi}}$.

The essential idea in our derivation is to closely analyze the numerator and denominator of the Fourier transform of box splines (as in Equation (3.6)). The numerator corresponds to the box spline's difference operator in the space domain, which is defined as

$$\nabla_{\boldsymbol{\Xi}} \longleftrightarrow \prod_{\boldsymbol{\xi}\in\boldsymbol{\Xi}} 1 - \exp\left(-\mathrm{i}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\omega}\right). \tag{3.13}$$

In Section 3.2.2, we will derive the difference coefficients, and their respective positions in 3-D, as a discrete series for the backward-difference operator of our box splines.

Using distribution theory, we can identify the remaining part of Equation (3.6) as the

Fourier transform of $G_{\boldsymbol{\Xi}}$ in space domain, since

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

If this differential operator $\boldsymbol{D}_{\boldsymbol{\Xi}}$ is applied to its Green's function $G_{\boldsymbol{\Xi}}$, Dirac's delta is obtained. However, if we apply the corresponding backward-difference operator to the Green's function, the box splines are obtained:

$$M_{\boldsymbol{\Xi}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\Xi}} G_{\boldsymbol{\Xi}}(\boldsymbol{x}).$$

We will show in Section 3.2.3 that the function $G_{\boldsymbol{\Xi}}$ is constructed by superpositions and linear transformations of a tensor-product of (two-sided) signed monomials: $x^k \operatorname{sgn}(x)$. In Section 3.2.4 we will see that we can also derive box splines by applying the difference operators on their *truncated powers*, $T_{\boldsymbol{\Xi}}$. Truncated powers are very similar to $G_{\boldsymbol{\Xi}}$ but instead of the two-sided signed monomials, they are constructed from one-sided monomials:

$$(x)_+^k = \begin{cases} x^k & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{3.14}$$

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

Four-direction box splines of general higher degrees are obtained by $(n-1)$-fold convolutions of the linear box spline with itself, which amounts to:

$$M_{\boldsymbol{\Xi}^{[n]}}(\boldsymbol{x}) = \nabla_{\boldsymbol{\Xi}^{[n]}} T_{\boldsymbol{\Xi}^{[n]}}(\boldsymbol{x}). \tag{3.15}$$

These box splines are represented in the frequency domain by:

$$\hat{M}_{\boldsymbol{\Xi}^{[n]}}(\boldsymbol{\omega}) = \hat{M}_{\boldsymbol{\Xi}}^n(\boldsymbol{\omega}) = \frac{1}{\mathrm{i}^{4n}} \prod_{k=1}^{4} \frac{(1 - \exp(-\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}))^n}{(\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega})^n}.$$

For notational convenience we introduce the scalar variables:

$$\begin{aligned} \mathrm{z}_k &:= \exp(-\mathrm{i}\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}) \\ \mathrm{w}_k &:= \boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}. \end{aligned} \tag{3.16}$$

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

$$\hat{M}_{\boldsymbol{\Xi}}^n(\boldsymbol{\omega}) = \prod_{k=1}^{4} \frac{(1 - \mathrm{z}_k)^n}{\mathrm{w}_k^n}.$$

Furthermore, we note that due to the fact that $\sum_{k=1}^{4} \boldsymbol{\xi}_k = 0$ in $\boldsymbol{\Xi}$, $\sum_{k=1}^{4} \mathrm{w}_k = 0$ and $\prod_{k=1}^{4} \mathrm{z}_k = 1$.

3.2.2 Difference Operator

The backward-difference operator can be represented as a filter. Its coefficients weight the Green's function that is shifted to the various lattice points, as in Equation (3.15). The Z-domain representation of the difference operator allows for an easy polynomial representation of this discrete series:

$$\nabla^n_{\boldsymbol{\Xi}}(\boldsymbol{z}) = \prod_{k=1}^{4} (1 - \mathrm{z}_k)^n. \tag{3.17}$$

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

$$\begin{aligned} \nabla_{\boldsymbol{\Xi}}(\boldsymbol{z}) =& 2 - (\mathrm{z}_1 + \mathrm{z}_2 + \mathrm{z}_3 + \mathrm{z}_4) \\ &+ (\mathrm{z}_1\mathrm{z}_2 + \mathrm{z}_1\mathrm{z}_3 + \mathrm{z}_1\mathrm{z}_4 + \mathrm{z}_2\mathrm{z}_3 + \mathrm{z}_2\mathrm{z}_4 + \mathrm{z}_3\mathrm{z}_4) \\ &- (\mathrm{z}_1\mathrm{z}_2\mathrm{z}_3 + \mathrm{z}_1\mathrm{z}_2\mathrm{z}_4 + \mathrm{z}_1\mathrm{z}_3\mathrm{z}_4 + \mathrm{z}_2\mathrm{z}_3\mathrm{z}_4). \end{aligned} \tag{3.18}$$

For a more compact notation, we adopt a slightly different multinomial notation where the power operation on 4-tuples $Z = (\mathrm{z}_1, \mathrm{z}_2, \mathrm{z}_3, \mathrm{z}_4)$ by $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is defined as:

$$Z^\alpha = \frac{1}{p(\alpha)} \sum_{(\beta_1,\ldots,\beta_4) \in \mathrm{perm}(\alpha_1,\ldots,\alpha_4)} \mathrm{z}_1^{\beta_1}\mathrm{z}_2^{\beta_2}\mathrm{z}_3^{\beta_3}\mathrm{z}_4^{\beta_4} \tag{3.19}$$

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

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

Note that since $\mathrm{z}_1^{-1} = \mathrm{z}_2\mathrm{z}_3\mathrm{z}_4$, both $Z^{(1,0,0,0)}$ and $Z^{(1,1,1,0)}$ denote the same set of monomials which contain exactly one lattice vector. Geometrically, one can visualize the coefficients being 2 at the origin, -1 on all BCC lattice points reachable by exactly one lattice vector (positive or negative) and $+1$ on lattice points that can be reached by exactly two lattice

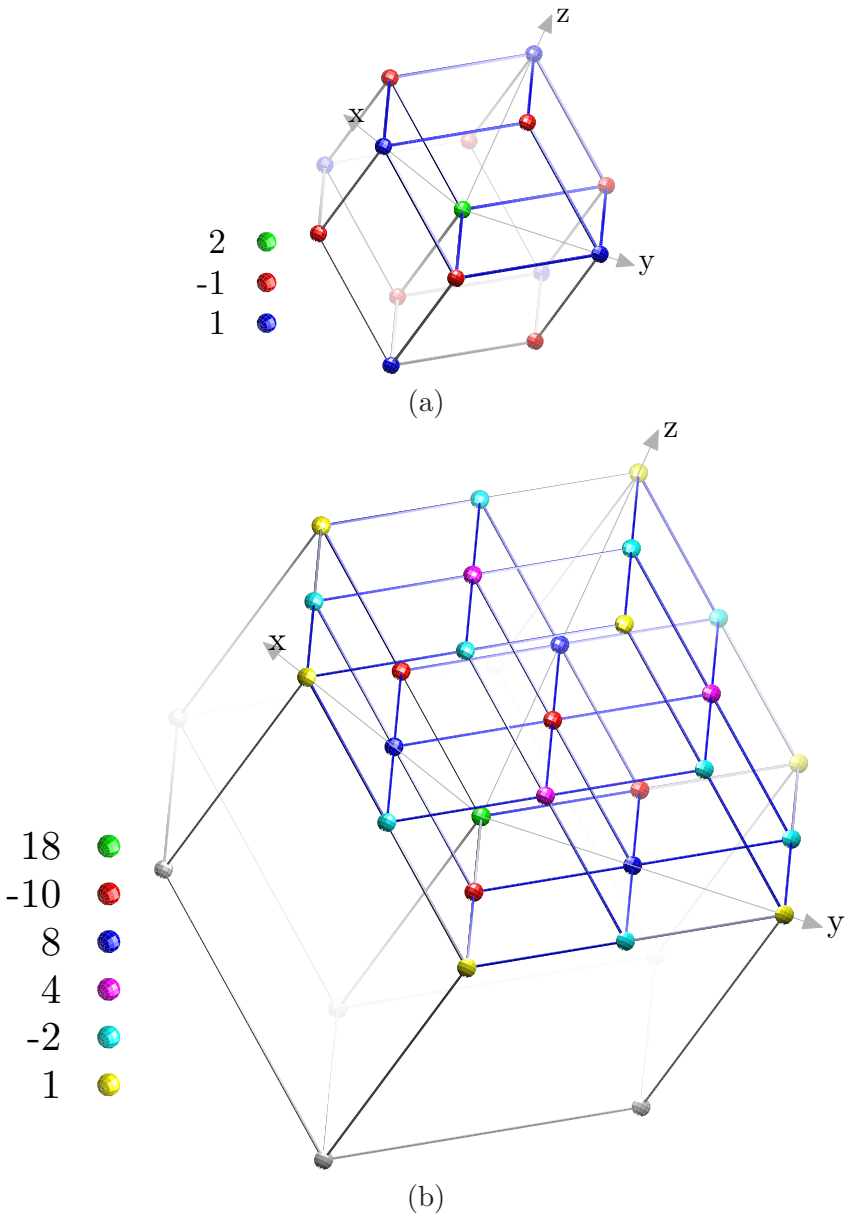


Figure 3.4: (a) Coefficients for the difference operator of the linear box spline. (b) Coefficients for the difference operator of the quintic box spline. For simplicity of illustration only one of the parallelepipeds constituting the rhombic dodecahedron has been drawn with all of its internal vertices.

vectors. This is shown in Figure 3.4(a). Similarly the difference operator coefficients for the quintic box spline $n = 2$ can be derived from Equation (3.17) in the compact form as:

$$\begin{aligned}\nabla^2_{\boldsymbol{\Xi}}(\boldsymbol{z}) =& 18 - 10Z^{(1,0,0,0)} + 8Z^{(1,1,0,0)} + 4Z^{(2,1,1,0)} \\ &- 2Z^{(2,1,0,0)} + Z^{(2,0,0,0)} + Z^{(2,2,0,0)}.\end{aligned} \tag{3.20}$$

These coefficients are on BCC lattice points on a rhombic dodecahedron with twice as large a neighborhood than that of the linear box spline case and are displayed in Figure 3.4(b).

3.2.3 Green's Function

We now describe a procedure to derive the space domain representation of the Green's functions of our box splines. We make use of the w_k variables introduced in Equation (3.16):

$$\hat{G}_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) = \prod_{k=1}^{4} \frac{1}{\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\omega}} = \prod_{k=1}^{4} \frac{1}{\mathrm{w}_k}.$$

The objective is to re-write $\hat{G}_{\boldsymbol{\Xi}}$ into a number of terms, each of which contains only three of the four w_k variables. Such a three-variable expression can then be written as a linear transformation of a trivariate function whose inverse Fourier transform can be obtained by a tensor-product. The general idea is to exploit the relation $\mathrm{w}_1 + \mathrm{w}_2 + \mathrm{w}_3 = -\mathrm{w}_4$ to reduce the number of variables in the denominator and obtain a sum of terms with one less variable. This helps to eliminate any fourth variable with the help of the proper numerator and consequently we introduce new terms in the expression while increasing the power of w_4. This procedure is the frequency domain reasoning of the spatial domain recursive structure of the box splines. We can always apply this procedure since whenever the number of directions n is greater than the dimension of the space s, the additional directions of the box spline can be written as the linear combination of the s linearly independent vectors. The $\hat{G}_{\boldsymbol{\Xi}}$ of the linear box spline can be re-written as:

$$\begin{aligned}\hat{G}_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) &= \frac{1}{\mathrm{w}_1\mathrm{w}_2\mathrm{w}_3\mathrm{w}_4} = -\frac{\frac{\mathrm{w}_1+\mathrm{w}_2+\mathrm{w}_3}{\mathrm{w}_4}}{\mathrm{w}_1\mathrm{w}_2\mathrm{w}_3\mathrm{w}_4} \\ &= \frac{-1}{\mathrm{w}_4^2\mathrm{w}_1\mathrm{w}_2\mathrm{w}_3} \times (\mathrm{w}_1 + \mathrm{w}_2 + \mathrm{w}_3).\end{aligned}$$

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

$$\hat{G}_{\boldsymbol{\Xi}}(\boldsymbol{\omega}) = \frac{-1}{\mathrm{w}_4^2}\left(\frac{1}{\mathrm{w}_1\mathrm{w}_2} + \frac{1}{\mathrm{w}_1\mathrm{w}_3} + \frac{1}{\mathrm{w}_2\mathrm{w}_3}\right). \tag{3.21}$$

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

$$\begin{aligned}
&\hat{G}_{\mathbf{\Xi}^{[2]}}(\boldsymbol{\omega}) = \hat{G}^2_{\mathbf{\Xi}}(\boldsymbol{\omega}) = \\
&\frac{1}{\mathrm{w}_4^4}\left(\frac{1}{\mathrm{w}_1^2\mathrm{w}_2^2} + \frac{1}{\mathrm{w}_1^2\mathrm{w}_3^2} + \frac{1}{\mathrm{w}_2^2\mathrm{w}_3^2}\right) + \\
&\frac{-2}{\mathrm{w}_4^5}\left(\frac{1}{\mathrm{w}_1^2\mathrm{w}_3} + \frac{1}{\mathrm{w}_1^2\mathrm{w}_2} + \frac{1}{\mathrm{w}_2^2\mathrm{w}_3} + \frac{1}{\mathrm{w}_2^2\mathrm{w}_1} + \frac{1}{\mathrm{w}_3^2\mathrm{w}_1} + \frac{1}{\mathrm{w}_3^2\mathrm{w}_2}\right) + \\
&\frac{6}{\mathrm{w}_4^6}\left(\frac{1}{\mathrm{w}_2\mathrm{w}_3} + \frac{1}{\mathrm{w}_1\mathrm{w}_3} + \frac{1}{\mathrm{w}_1\mathrm{w}_2}\right).
\end{aligned} \tag{3.22}$$

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

$$\begin{aligned}
\hat{\rho}_1(\omega_x, \omega_y, \omega_z) =& \frac{-1}{\omega_x\omega_y\omega_z^2} \\
\hat{\rho}_2(\omega_x, \omega_y, \omega_z) =& \frac{1}{\omega_x^2\omega_y^2\omega_z^4} - 2\left(\frac{1}{\omega_x^2\omega_y} + \frac{1}{\omega_x\omega_y^2}\right)\frac{1}{\omega_z^5} + \frac{6}{\omega_x\omega_y\omega_z^6}.
\end{aligned} \tag{3.23}$$

These functions are useful since the $\hat{G}_{\mathbf{\Xi}}$ of our box splines are essentially linear transformations (e.g., $(\mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_4) = \mathbf{\Xi}^{\mathrm{T}}_{\{1,2,4\}}\boldsymbol{\omega}$) and superpositions of these building-block functions:

$$\begin{aligned}
\hat{G}_{\mathbf{\Xi}}(\boldsymbol{\omega}) =& \hat{\rho}_1(\mathbf{\Xi}^{\mathrm{T}}_{\{1,2,4\}}\boldsymbol{\omega}) + \hat{\rho}_1(\mathbf{\Xi}^{\mathrm{T}}_{\{1,3,4\}}\boldsymbol{\omega}) + \hat{\rho}_1(\mathbf{\Xi}^{\mathrm{T}}_{\{2,3,4\}}\boldsymbol{\omega}) \\
\hat{G}_{\mathbf{\Xi}^{[2]}}(\boldsymbol{\omega}) =& \hat{\rho}_2(\mathbf{\Xi}^{\mathrm{T}}_{\{1,2,4\}}\boldsymbol{\omega}) + \hat{\rho}_2(\mathbf{\Xi}^{\mathrm{T}}_{\{1,3,4\}}\boldsymbol{\omega}) + \hat{\rho}_2(\mathbf{\Xi}^{\mathrm{T}}_{\{2,3,4\}}\boldsymbol{\omega}).
\end{aligned}$$

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

We now derive the inverse Fourier transform of these building-block functions. First, we recognize that the Fourier inverse of $1/(\mathrm{i}\omega)^k$ is the two-sided monomial [6]:

$$\frac{(x)^k_{\mathrm{sgn}}}{k!} := \frac{1}{2}\frac{x^k\,\mathrm{sgn}(x)}{k!} \longleftrightarrow \frac{1}{(\mathrm{i}\omega)^{k+1}}. \tag{3.24}$$

We can derive the space domain representation of our building-block functions as a tensor-product inverse Fourier transform of equations in (3.23):

$$\begin{aligned}
\rho_1(x, y, z) =& -(x)^0_{\mathrm{sgn}}(y)^0_{\mathrm{sgn}}(z)_{\mathrm{sgn}} \\
\rho_2(x, y, z) =& \frac{1}{3!}(x)_{\mathrm{sgn}}(y)_{\mathrm{sgn}}(z)^3_{\mathrm{sgn}} + \\
&\frac{-2}{4!}\left[(x)_{\mathrm{sgn}}(y)^0_{\mathrm{sgn}} + (x)^0_{\mathrm{sgn}}(y)_{\mathrm{sgn}}\right](z)^4_{\mathrm{sgn}} + \\
&\frac{6}{5!}(x)^0_{\mathrm{sgn}}(y)^0_{\mathrm{sgn}}(z)^5_{\mathrm{sgn}}.
\end{aligned} \tag{3.25}$$

If $\boldsymbol{Q}$ is an invertible matrix, we know that $f(\boldsymbol{Q}\boldsymbol{x}) \longleftrightarrow \hat{f}((\boldsymbol{Q}^{-1})^{\mathrm{T}}\boldsymbol{\omega})/|\det \boldsymbol{Q}|$. Therefore, we can write the space domain representation of the Green's function of these box splines as:

$$\begin{aligned} G_{\boldsymbol{\Xi}}(\boldsymbol{x}) =& \frac{1}{4}\left(\rho_1(\boldsymbol{\Xi}_{\{1,2,4\}}^{-1}\boldsymbol{x}) + \rho_1(\boldsymbol{\Xi}_{\{1,3,4\}}^{-1}\boldsymbol{x}) + \rho_1(\boldsymbol{\Xi}_{\{2,3,4\}}^{-1}\boldsymbol{x})\right) \\ G_{\boldsymbol{\Xi}^{[2]}}(\boldsymbol{x}) =& \frac{1}{4}\left(\rho_2(\boldsymbol{\Xi}_{\{1,2,4\}}^{-1}\boldsymbol{x}) + \rho_2(\boldsymbol{\Xi}_{\{1,3,4\}}^{-1}\boldsymbol{x}) + \rho_2(\boldsymbol{\Xi}_{\{2,3,4\}}^{-1}\boldsymbol{x})\right) \end{aligned} \tag{3.26}$$

where $\boldsymbol{x} = (x, y, z)$ and $|\det \boldsymbol{\Xi}_{\{1,2,4\}}^{-1}| = |\det \boldsymbol{\Xi}_{\{1,3,4\}}^{-1}| = |\det \boldsymbol{\Xi}_{\{2,3,4\}}^{-1}| = 1/4$.

3.2.4 Truncated Power

Recall that the Green's functions were constructed from the two-sided signed monomials (i.e., of the form $x^k \operatorname{sgn}(x)$) in equations (3.25). The differential operator $D_{\boldsymbol{\Xi}}$, when applied on these signed monomials, transformed in equations in (3.26), results in a δ function. Consequently, this differential operator annihilates all *polynomials* of degree $\leq k$ encountered in the signed polynomials in the Green's function. Similarly the backward-difference operator annihilates all of these polynomials [24, I.32].

Since box splines are obtained by applying the difference operator on the Green's function, we can add or subtract any polynomials up to degree k found in the Green's function. Therefore, the box spline can also be obtained by applying the difference operator to a Green's function that is obtained from $\frac{1}{2}(x^k \operatorname{sgn}(x) - x^k) = -(x)_-^k$ or $\frac{1}{2}(x^k \operatorname{sgn}(x) + x^k) = (x)_+^k$. The contributions from adding or subtracting x^k are eliminated since the difference operator annihilates the polynomials made of x^k. The advantage of working with these one-sided monomials is that they are supported on half-spaces; whereas the support of Green's functions in Equation (3.26) is the entire space.

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

$$\begin{aligned} \tau_1(x, y, z) =& (x)_+^0 (y)_+^0 (z)_- \\ \tau_2(x, y, z) =& \frac{-1}{3!}(x)_+(y)_+(z)_-^3 + \\ & \frac{2}{4!}\left[(x)_+(y)_+^0 + (x)_+^0(y)_+\right](z)_-^4 + \\ & \frac{-6}{5!}(x)_+^0(y)_+^0(z)_-^5. \end{aligned} \tag{3.27}$$

Note that the support of these building block functions is on points $\boldsymbol{x} \in \mathbb{R}^3$ such that

$x, y > 0$ and $z < 0$. We further derive the truncated power functions:

$$\begin{aligned} T_{\boldsymbol{\Xi}}(\boldsymbol{x}) =& \frac{1}{4}\left(\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x}) + \tau_1(\boldsymbol{\Xi}^{-1}_{\{1,3,4\}}\boldsymbol{x}) + \tau_1(\boldsymbol{\Xi}^{-1}_{\{2,3,4\}}\boldsymbol{x})\right) \\ T_{\boldsymbol{\Xi}^{[2]}}(\boldsymbol{x}) =& \frac{1}{4}\left(\tau_2(\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x}) + \tau_2(\boldsymbol{\Xi}^{-1}_{\{1,3,4\}}\boldsymbol{x}) + \tau_2(\boldsymbol{\Xi}^{-1}_{\{2,3,4\}}\boldsymbol{x})\right). \end{aligned} \tag{3.28}$$

The crucial point here is that the values of truncated power functions at any point $\boldsymbol{x}$, are affected only by one of the three terms on the right hand sides of the above equations. To see this fact recall that $\tau_1(x, y, z)$ and $\tau_2(x, y, z)$ are non-zero only when $x, y > 0$ and $z < 0$. The support of each building-block function is transformed in Equation (3.28) to cones formed by columns of $\boldsymbol{\Xi}$ (e.g., the support of $\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x})$ is all points $\boldsymbol{x} = \boldsymbol{\Xi}_{\{1,2,4\}}(t_1, t_2, t_3)$ for $t_1, t_2 > 0$ and $t_3 < 0$). The support of the building-block functions is transformed as:

$$\begin{aligned} \boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x} &= \frac{1}{2}(x - z, y - z, x + y) \\ \boldsymbol{\Xi}^{-1}_{\{1,3,4\}}\boldsymbol{x} &= \frac{1}{2}(x - y, z - y, x + z) \\ \boldsymbol{\Xi}^{-1}_{\{2,3,4\}}\boldsymbol{x} &= \frac{1}{2}(y - x, z - x, y + z). \end{aligned} \tag{3.29}$$

Therefore, the support of $\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x})$ is the cone that is the intersection of the half-spaces determined by $x - z > 0$, $y - z > 0$ and $x + y < 0$. Similarly, the support of $\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,3,4\}}\boldsymbol{x})$ is the cone that is the intersection of the half-spaces determined by $x - y > 0$, $z - y > 0$ and $x + z < 0$. Since $y - z < 0$ and $z - y < 0$ are disjoint, the support of $\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}\boldsymbol{x})$ and $\tau_1(\boldsymbol{\Xi}^{-1}_{\{1,3,4\}}\boldsymbol{x})$ are disjoint. Therefore, the supports of each building-block function transformed in Equations (3.28) are non-overlapping.

The support of the truncated power functions of the box splines is the union of each cone formed by the matrices of the equations in (3.28). The support of each transformed building-block function along with the support of the truncated power of the box splines is shown in Figure 3.5. The red arrows indicate the half-spaces in the support. Therefore, one can verify that the support of the truncated power functions is the *union* of half-spaces determined by $x + y < 0$, $x + z < 0$ and $y + z < 0$.

Since the supports of the transformed τ functions are non-overlapping, at any point $\boldsymbol{x}$ only one of the three transformed τ's contribute to the value of the truncated power. For $\boldsymbol{x}$ to be within the support of one of the transformed τ functions its first and second components of the transformed vector need to be positive while the third component needs to be negative. Under this assumption the last component of each vector of the right hand side of Equation (3.29) is the sum of the two largest values out of x, y and z. And the

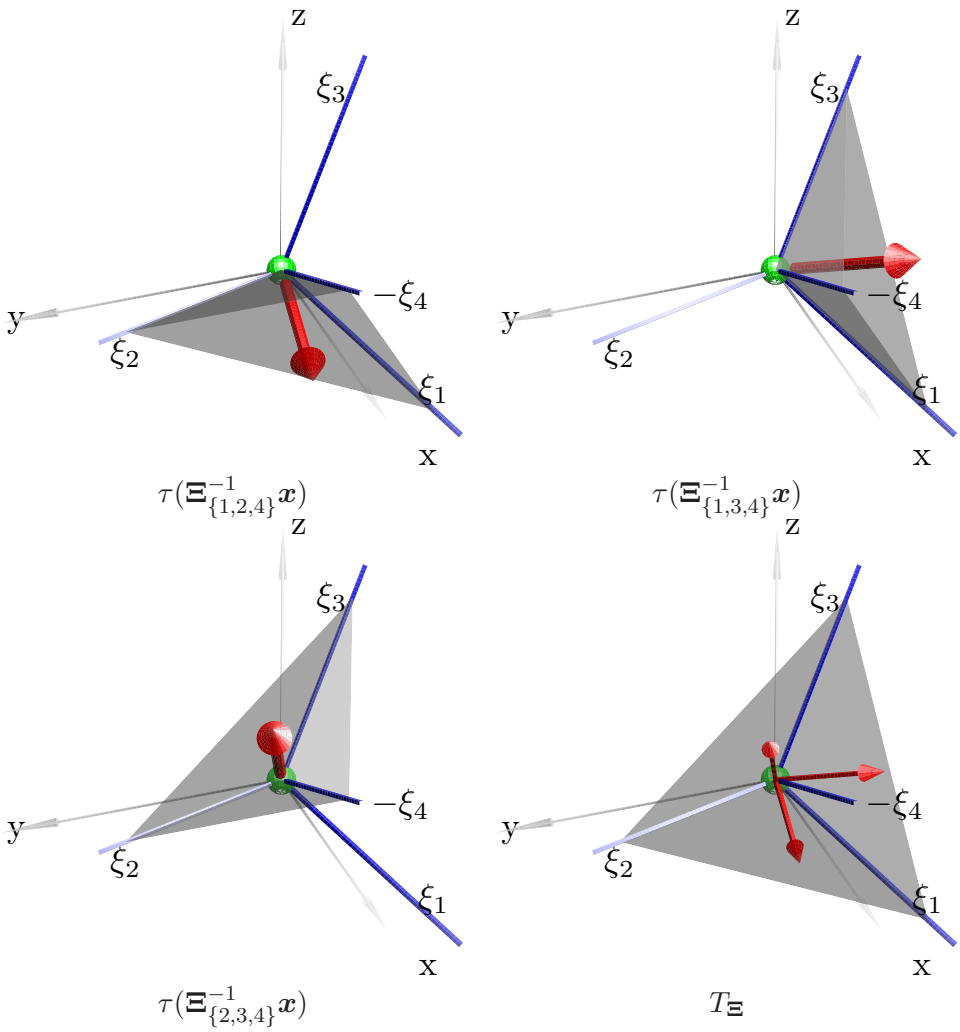


Figure 3.5: The support of the truncated power function is the cone formed by the three directions in $\Xi_{\{1,2,3\}}$. This volume is a *disjoint union* of support of three τ functions each transformed by $\Xi^{-1}_{\{1,2,4\}}$, $\Xi^{-1}_{\{1,3,4\}}$ and $\Xi^{-1}_{\{2,3,4\}}$. Note the support of the τ functions contains the negative octant of space.

other two components in each set is the difference of the largest and the middle value from the minimum of the three. We also notice that these basic building-block functions are symmetric with respect to the first and second components of the position vector $\boldsymbol{x}$ of their argument (see Equation (3.25)). For example when τ is transformed by $\boldsymbol{\Xi}^{-1}_{\{1,2,4\}}$, its support is determined by the region specified by $x > z$, $y > z$ and $x+y < 0$. Using these observations we can write the truncated power in terms of only one basic building-block function:

$$T_{\boldsymbol{\Xi}}(x,y,z) = \frac{1}{4}\tau_1(\frac{1}{2}(\tilde{x}-\tilde{z}), \frac{1}{2}(\tilde{y}-\tilde{z}), \frac{1}{2}(\tilde{x}+\tilde{y})) \tag{3.30}$$

$$T_{\boldsymbol{\Xi}^{[2]}}(x,y,z) = \frac{1}{4}\tau_2(\frac{1}{2}(\tilde{x}-\tilde{z}), \frac{1}{2}(\tilde{y}-\tilde{z}), \frac{1}{2}(\tilde{x}+\tilde{y})) \tag{3.31}$$

where

$$\tilde{x} = \max(x,y,z)\ ,\ \tilde{y} = \mathrm{mid}(x,y,z) \text{ and } \tilde{z} = \min(x,y,z). \tag{3.32}$$

3.2.5 Efficient Evaluation

Having obtained the explicit form of the truncated power, we shall apply the difference operator derived in Section 3.2.2 to $T_{\boldsymbol{\Xi}^{[n]}}$ in order to obtain $M_{\boldsymbol{\Xi}^{[n]}}$. This operation indicated as in Equation (3.15) can be implemented as a convolution of the backward-difference operator sequence with the truncated power. In this section we exploit the symmetries in the support of these box splines and find a *region* for efficient evaluation of this convolution.

Region of evaluation: 1-D example

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

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

The process of this convolution is illustrated in Figure 3.6. In this figure (a) shows the truncated power $(x)_-$, (b) shows the difference operator coefficients, (c) the result of the convolution shown by overlaying the truncated powers at their respective difference operator *sites* and (d) shows the resulting B-spline. In Figure 3.6(d) the red band indicates the regions

of the x axis that are affected by the convolution site at 1, the green band indicates the region that is affected by the convolution site at 0 and the blue band indicates the region that is affected by the convolution site at -1. The symmetry of the support of the linear B-spline suggests an efficient evaluation in the interval of $[0, 1]$ where only one convolution site contributes to the values of the B-spline in this region.

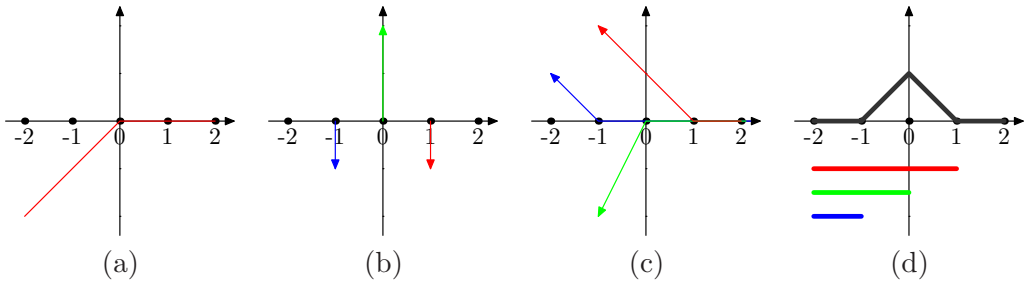


Figure 3.6: Convolution of the truncated power $(x)_-$ with the difference operator, a 1-D example. (a) the truncated power $(x)_-$. (b) the discrete difference operator coefficients. (c) the convolution by overlaying the truncated power functions. (d) the resulting convolution yields the linear B-spline. The red band indicates the region that is affected by the convolution site at $+1$, the green band indicates the region that is affected by the convolution site at 0 and the blue band indicates the region that is affected by the convolution site at -1.

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

Region of evaluation: trivariate case

Since our trivariate box splines are obtained through a projection along the antipodal axis of a tesseract they exhibit the symmetries present in their polyhedral support which is a rhombic dodecahedron. We exploit the symmetries present within the rhombic dodecahedron to achieve an efficient evaluation method for the linear and quintic box splines.

First, we observe that a rhombic dodecahedron can be decomposed into four non-overlapping parallelepipeds in two different ways. For a rhombic dodecahedron formed by the vectors in Ξ as in Figure 3.3, one can construct four parallelepipeds each formed by three of the four vectors from Ξ. Alternatively, one can choose the negative directions from $-\Xi$ to

decompose the rhombic dodecahedron into four parallelepipeds (see Figure 2.6). Therefore, we can confine the evaluation region to one of these parallelepipeds and the evaluation at the other points can be inferred by symmetry.

The support of $T_{\boldsymbol{\Xi}}$ (or $T_{\boldsymbol{\Xi}^{[2]}}$) is the positive cone of $\boldsymbol{\Xi}_{\{1,2,3\}}$ as in Figure 3.5. A minimal number of convolution sites contribute to the value of the box spline in the parallelepiped which is cornered at the origin and formed by $-\boldsymbol{\xi}_1, -\boldsymbol{\xi}_2$ and $-\boldsymbol{\xi}_3$. This parallelepiped contains the positive octant of $\mathbb{R}^3$ and is shown by blue edges in Figure 3.7. Similarly for the quintic box spline, a minimal number of convolution sites contribute to the parallelepiped formed by $-2\boldsymbol{\xi}_1, -2\boldsymbol{\xi}_2$ and $-2\boldsymbol{\xi}_3$ which is shown in Figure 3.8.

Linear Box Spline

The operation of the difference operator on the truncated power is a sum of truncated power functions shifted and weighted according to the difference operator sites as in Figure 3.4(a). As the support of the truncated power is limited to the cone of the direction vectors in $\boldsymbol{\Xi}_{\{1,2,3\}}$, only one of the terms of the convolution contributes to the value of $M_{\boldsymbol{\Xi}}$ in the parallelepiped that we deal with; this term is the one obtained from shifting the $T_{\boldsymbol{\Xi}}$ to $(1,1,1)$ and multiplying by -1, the difference operator coefficient at this point (Figure 3.4(a)). None of the other difference operator sites affect this region of interest, as shown in Figure 3.7. Therefore, a point in the parallelepiped of focus is characterized by $(x,y,z) = -\boldsymbol{\Xi}_{\{1,2,3\}}(t_1,t_2,t_3)$ where $0 \le t_1,t_2,t_3 < 1$. Using the min/mid/max variables introduced in Equation (3.32) we have:

$$\begin{aligned} M_{\boldsymbol{\Xi}}(x,y,z) =& (-1)T_{\boldsymbol{\Xi}}(x-1,y-1,z-1) \\ =& -\frac{1}{4}\tau_1\left(\frac{1}{2}(\tilde{x}-1-(\tilde{z}-1)),\frac{1}{2}(\tilde{y}-1-(\tilde{z}-1)),\frac{1}{2}(\tilde{x}-1+\tilde{y}-1)\right) \\ =& -\frac{1}{8}(\tilde{x}+\tilde{y}-2) \\ =& \frac{1}{4}\left(1-\frac{1}{2}(\max(x,y,z)+\operatorname{mid}(x,y,z))\right), \end{aligned} \tag{3.33}$$

which agrees with the geometric simplification we derived in [30] normalized according to the sampling density of the BCC lattice described by Equation (1.1).

Quintic Box Spline

The same procedure as for the linear box spline can be used. The difference operator for the quintic box spline is shown in Figure 3.4(b). The support of the truncated power is

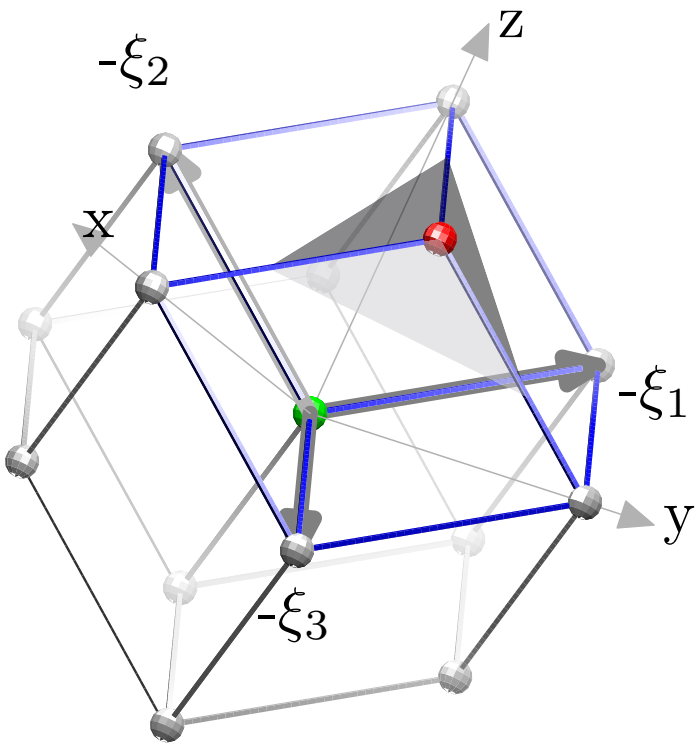


Figure 3.7: When computing the operation of the difference operator on the truncated power, only one term which shifts the truncated power to $(1, 1, 1)$ affects the parallelepiped of interest formed by $-\boldsymbol{\Xi}_{\{1,2,3\}}$. The support of the truncated powers shifted to the other sites, do not intersect this parallelepiped.

limited to the cone of the direction vectors in $\boldsymbol{\Xi}_{\{1,2,3\}}$ such that only eight of the terms of the convolution contribute to the value of $M_{\boldsymbol{\Xi}^{[2]}}$ in the parallelepiped that we focus on (see Figure 3.8). As we saw in Section 3.1.4 the size of the support of the quintic box spline is doubled from that of the linear box spline; therefore, the parallelepiped of focus is now eight times the size of the corresponding one in the linear box spline case. Therefore for a point in the parallelepiped of focus characterized by $(x, y, z) = -\boldsymbol{\Xi}_{\{1,2,3\}}(t_1, t_2, t_3)$ where $0 \leq t_1, t_2, t_3 < 2$, we have:

$$\begin{aligned} M_{\boldsymbol{\Xi}^{[2]}}(x, y, z) = {} & T_{\boldsymbol{\Xi}^{[2]}}(x-2, y-2, z-2) \\ & - 10T_{\boldsymbol{\Xi}^{[2]}}(x-1, y-1, z-1) - 2T_{\boldsymbol{\Xi}^{[2]}}(x-3, y-1, z-1) \\ & - 2T_{\boldsymbol{\Xi}^{[2]}}(x-1, y-3, z-1) - 2T_{\boldsymbol{\Xi}^{[2]}}(x-1, y-1, z-3) \\ & + 4T_{\boldsymbol{\Xi}^{[2]}}(x-2, y-2, z) + 4T_{\boldsymbol{\Xi}^{[2]}}(x-2, y, z-2) \\ & + 4T_{\boldsymbol{\Xi}^{[2]}}(x, y-2, z-2) \end{aligned}$$

where $T_{\boldsymbol{\Xi}^{[2]}}$ is defined as in Equation (3.31). The shifts in the above equation are shifts to the difference operator sites which are the colored nodes in Figure 3.8.

Using the symmetries of the rhombic dodecahedron we can confine the evaluation region to a tetrahedron that has a vertex at the origin and its apex is a quarter of one face of the original parallelepiped since these faces are rhomboids and have four-fold symmetry. The rhomboid face of the parallelepiped of our focus lies in the plane specified by $x + y = 4$ in Figure 3.9. Out of four possible choices, we pick this tetrahedron so that it completely lies in the positive octant. This region is specified by its four bounding planes: $x + y < 4$, $x > y$, $y > z$ and $z > 0$. This tetrahedron is indicated by the dark tetrahedron in Figure 3.9. It is partitioned into four regions formed by the intersections with four of the eight sub-parallelepipeds that constitute the original parallelepiped of focus. These sub-parallelepipeds are highlighted in green in Figure 3.10. These four regions are identified by:

$$\begin{aligned} R_1 &: x + y < 2; & R_2 &: x + y > 2, x + z < 2; \\ R_3 &: x + z > 2, y + z < 2; & R_4 &: y + z > 2. \end{aligned}$$

These regions are determined by posing the restriction of being in the dark tetrahedron of focus which is specified by $x > y > z > 0$ and each of the four sub-parallelepipeds that intersect this tetrahedron. These sub-parallelepipeds are specified by $-\boldsymbol{\Xi}_{\{1,2,3\}}\boldsymbol{t}$ where the

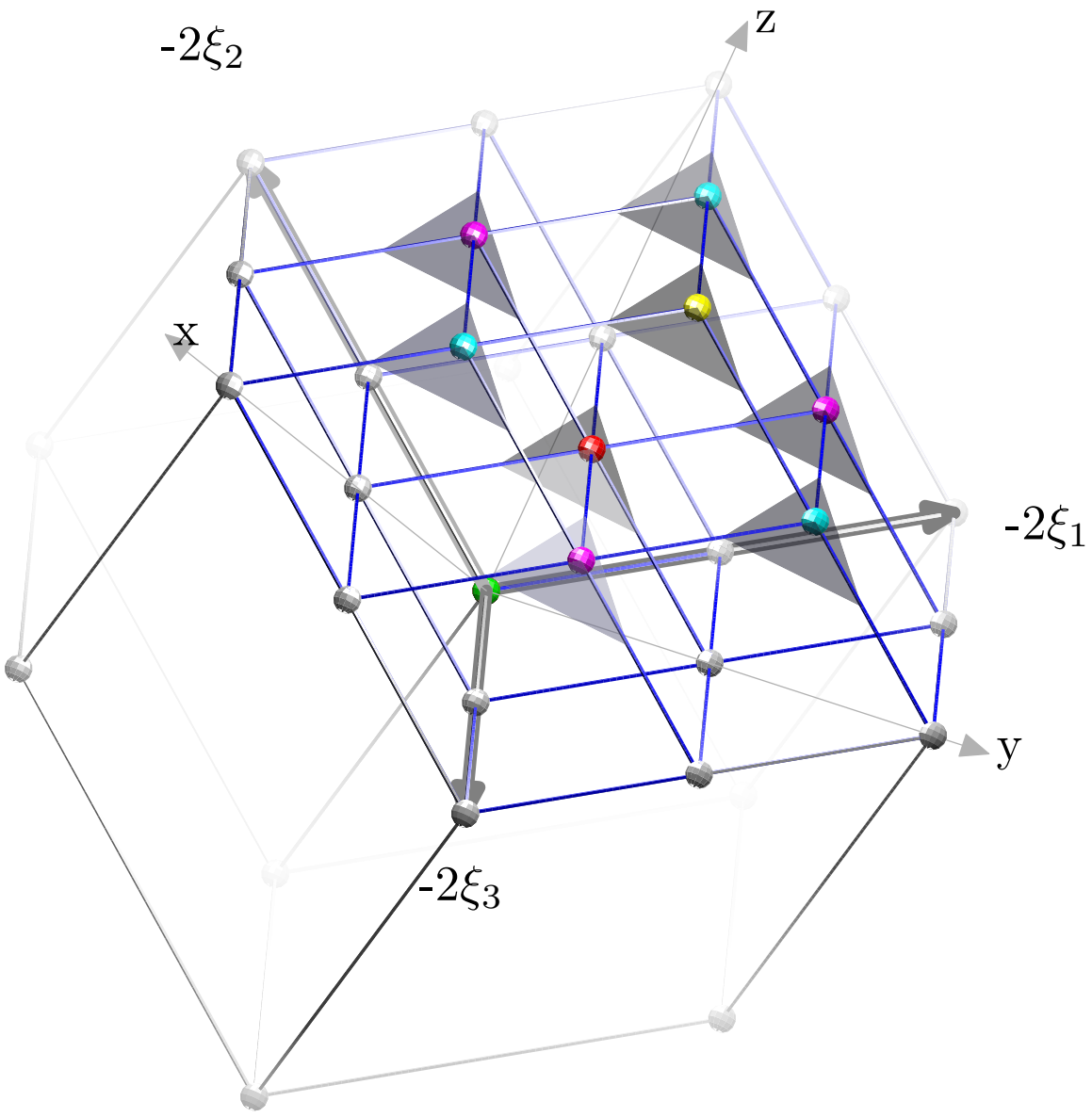


Figure 3.8: When computing the convolution of the difference operator and the truncated power, only eight terms of the convolution affect the parallelepiped of interest formed by $-2\boldsymbol{\Xi}_{\{1,2,3\}}$. The support of the truncated power shifted to each site is the cone indicated by the gray region. The support of the truncated power shifted to the other convolution sites, does not intersect this parallelepiped.

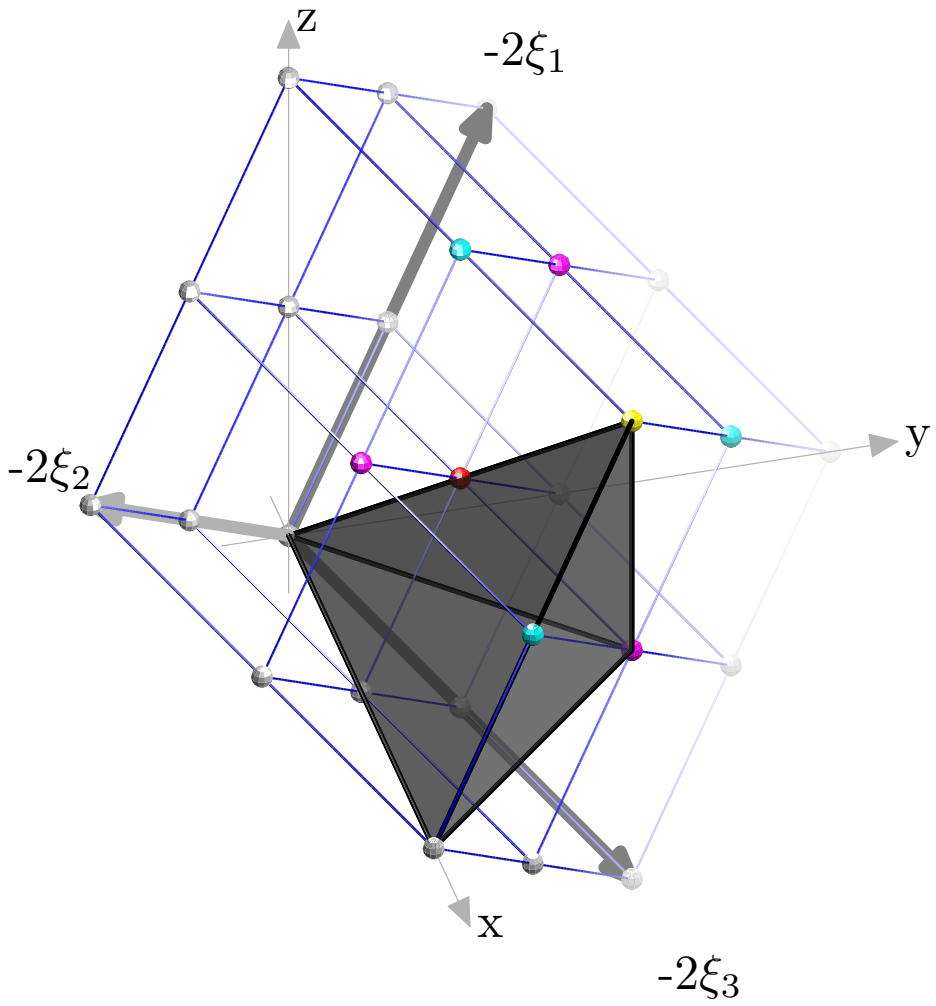


Figure 3.9: The region specified by $x + y < 4$ and $x > y > z > 0$ is indicated with the dark tetrahedron. This tetrahedron is formed by connecting the origin to the face which is the triangle that is one quarter of the rhombic face of the original rhombic dodecahedron.

sub-parallelepiped of region R_1 is specified by $0 \leq t_1, t_2, t_3 < 1$; the sub-parallelepiped of region R_2 is specified by $0 \leq t_1, t_2 < 1$ and $1 \leq t_3 < 2$; the sub-parallelepiped of region R_3 is specified by $0 \leq t_1 < 1$ and $1 \leq t_2, t_3 < 2$ and the sub-parallelepiped of region R_4 is specified by $1 \leq t_1, t_2, t_3 < 2$.

In each of these regions, shown in Figure 3.10, the box spline will be represented as a separate polynomial. The region R_1 is affected by all of the eight difference operator sites in the parallelepiped of focus. Region R_2 is affected only by four sites at $(3,1,1)$, $(2,2,2)$, $(1,3,1)$ and $(2,2,0)$. Region R_3 is affected by two sites at $(3,1,1)$ and $(2,2,2)$. Finally, region R_4 is affected only by $(2,2,2)$. Therefore, we simplify the polynomials in each region separately. Using constants $\alpha := 1/3840$, $\beta := 1/1920$ and $\gamma := 1/960$ we have:
Region R_1, $M_{\mathbf{\Xi}^{[2]}}(x,y,z) =$

$$\begin{aligned}
&\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24)\\
&+\beta(x+z-2)^3(x^2-9x-3xz+10y-5y^2+14+11z+z^2)\\
&+\beta(y+z-2)^3(46-30x-z-y+3zy+5x^2-y^2-z^2)\\
&-\gamma(x+y-2)^3(x^2+x-3xy-5z^2+y^2+y-6).
\end{aligned} \tag{3.34}$$

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

$$\begin{aligned}
&\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24)\\
&-\beta(x+z-2)^3(-z^2-11z+3xz-14+5y^2+9x-10y-x^2)\\
&-\beta(y+z-2)^3(-46+z+30x+y-3zy-5x^2+y^2+z^2).
\end{aligned} \tag{3.35}$$

Region R_3: This region as shown in Figure 3.10 is not a simple tetrahedron with homogeneous regions with respect to the site located at $(3,1,1)$. When the truncated power is centered at this site, two of the three components constituting the truncated power (see Figure 3.5) intersect this region. Therefore, there are two sub-cases for the contribution of the truncated power centered at $(3,1,1)$. But the contribution from $(2,2,2)$ remains homogeneous as only one of the three components of the truncated power (see Figure 3.5) contributes to this region:
Region $R_{3A}\{x-z>2\}$, $M_{\mathbf{\Xi}^{[2]}}(x,y,z) =$

$$\alpha(x+y-4)^3(-x^2+8x+3xy-y^2+5z^2-16-12y). \tag{3.36}$$

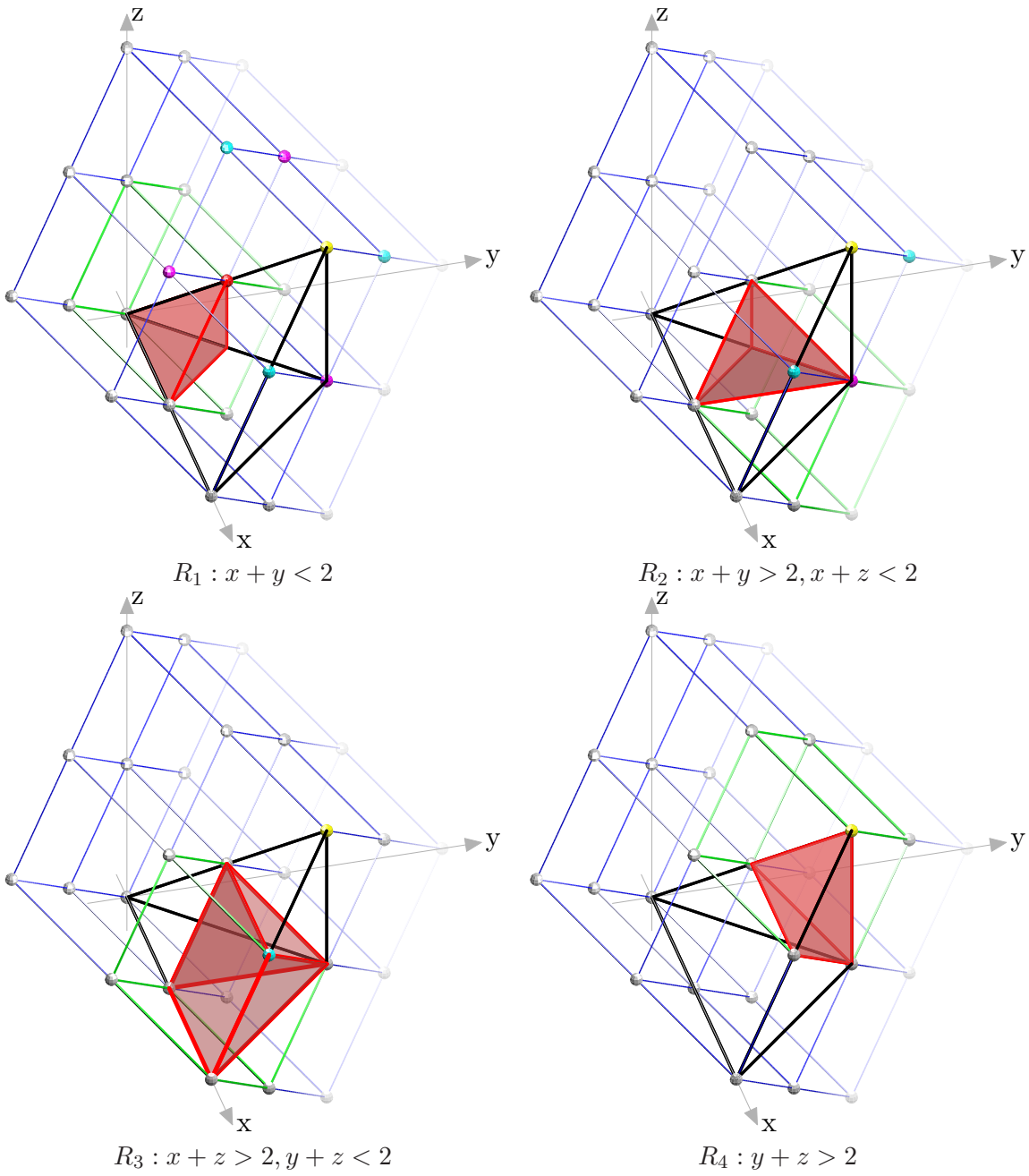


Figure 3.10: Different regions of the evaluation domain are the intersection of sub-parallelepipeds (highlighted in green) and the tetrahedron in focus (black edges). The box spline is a homogeneous polynomial in each region.

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

$$\begin{aligned}&\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24)\\&-\beta(y+z-2)^3(30x+z-46-3yz+y-5x^2+y^2+z^2).\end{aligned} \tag{3.37}$$

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

$$\alpha(x+y-4)^3(-3xy-5z^2+2x+2y+20z+x^2+y^2-24). \tag{3.38}$$

Therefore a fast evaluation of the box spline can be obtained for any (x,y,z) by first transforming it to the tetrahedron of focus (the dark tetrahedron in Figure 3.9) by taking $(x,y,z)=(|x|,|y|,|z|)$ and then sorting $(x,y,z)=(\max(x,y,z),\mathrm{mid}(x,y,z),\min(x,y,z))$. Once these transforms are performed, we can test for the appropriate of the above five regions and the polynomial form can be evaluated. We also note that due to the explicit piecewise polynomial form above, and the relatively small support of these polynomial regions, our evaluation is numerically stable.

As a summary we include the pseudo-code for the evaluation of the quintic box spline referring to the regions above.

```
function quintic_boxspline(x,y,z)
% The following two lines transform the point x,y,z to the
% corresponding point in the focus region in Figure 3.9
x = abs(x); y = abs(y); z = abs(z);
sort x,y,z in decreasing order

% If this point is outside the rhombic dodecahedron's face
% which is on the plane (x+y)=4, we are outside of the support
if((x+y)>4) return 0;
if((x+y)<2) % Region R_1
    return M_{Ξ^[2]_{R_1}}(x,y,z) as in Equation (3.34);
else if((x+z)<2) % Region R_2
    return M_{Ξ^[2]_{R_2}}(x,y,z) as in Equation (3.35);
else if((y+z)<2) % Region R_3
    if((x−z)>2) % Region R_3A
        return M_{Ξ^[2]_{R_3A}}(x,y,z) as in Equation (3.36);
```

```
    else % Region R_3B
        return M_{Ξ^[2]_{R_3B}}(x, y, z) as in Equation (3.37);
else % Region R_4
    return M_{Ξ^[2]_{R_4}}(x, y, z) as in Equation (3.38);
end
```

3.3 Reconstruction Results

In this section we describe our experimental results and compare the BCC sampling scheme to the traditional Cartesian sampling on volumetric datasets.

In order to examine the reconstruction schemes discussed in this paper, we have implemented a ray-caster to render images from the Cartesian and the BCC sampled volumetric datasets.

We have chosen the synthetic dataset (Figure 3.11) first proposed in [53] as a benchmark for our comparisons. The function was sampled at the resolution of $41 \times 41 \times 41$ on the Cartesian lattice and at an almost[2] equivalent sampling on the BCC lattice of $32 \times 32 \times 64$. The images in Figure 3.12 are rendered using the quintic box spline on the BCC sampled datasets and the tricubic B-spline on the Cartesian sampled datasets. The analytical function was rendered by evaluating the actual function proposed in [53]. The images in the second row in Figure 3.12 document the corresponding error images that are obtained of the angular error that occurred in estimating the normal by central differencing on the reconstructed function. Although direct reconstruction of the normal is possible by using the analytical gradient of the reconstruction kernel, we chose central differencing with a relatively small step (.1 of the sampling step) on the reconstructed function to approximate the true gradient. Central differencing is commonly the method of choice in the visualization domain and there is no reason to believe that it performs any better or worse than taking the analytical derivative of the reconstruction kernel [67]. The gray value of 255 (white) denotes an angular error of .3 radians between the computed normal and the exact normal.

[2]Since a finite sampling of a volume can not produce the exact same number of samples for the BCC and Cartesian sampling patterns, for our discrete resolutions, we chose the resolutions conservatively in favor of the Cartesian sampling. Therefore, the actual sampling density in the Cartesian sampled datasets is slightly higher than the BCC sampling density.

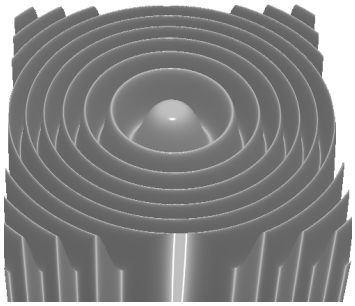


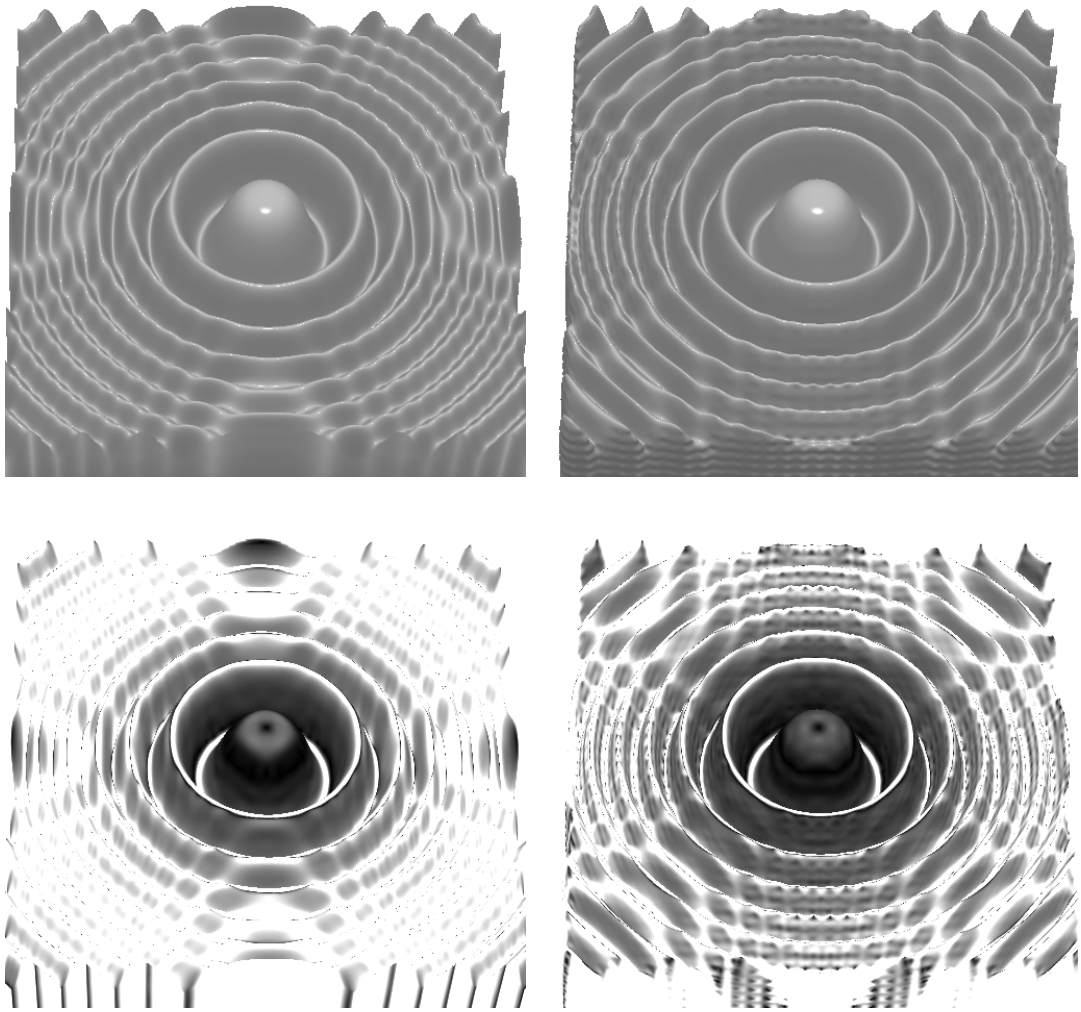
Figure 3.11: The ML dataset: An image of the explicit function introduced by Marschner [53].

The superiority of the BCC sampling is apparent by comparing the images in Figure 3.12(a) and Figure 3.12(b) as these are obtained from an equivalent sampling density over the volume. While the lobes are mainly preserved in the BCC case, they are more smoothed out in the case of Cartesian sampling. This is also confirmed by their corresponding error images in the second row of Figure 3.12.

We also examined the quasi-interpolation scheme developed in Section 3.1.5 with the ML dataset. In particular, we employed the linear functional developed in Equation (3.11) for the quintic box spline and the functional in Equation (3.12) for the tricubic B-spline. The resulting reconstructions shown in Figure 3.13 still favor the BCC sampling and reconstruction scheme as the error image is significantly darker for the BCC pipeline.

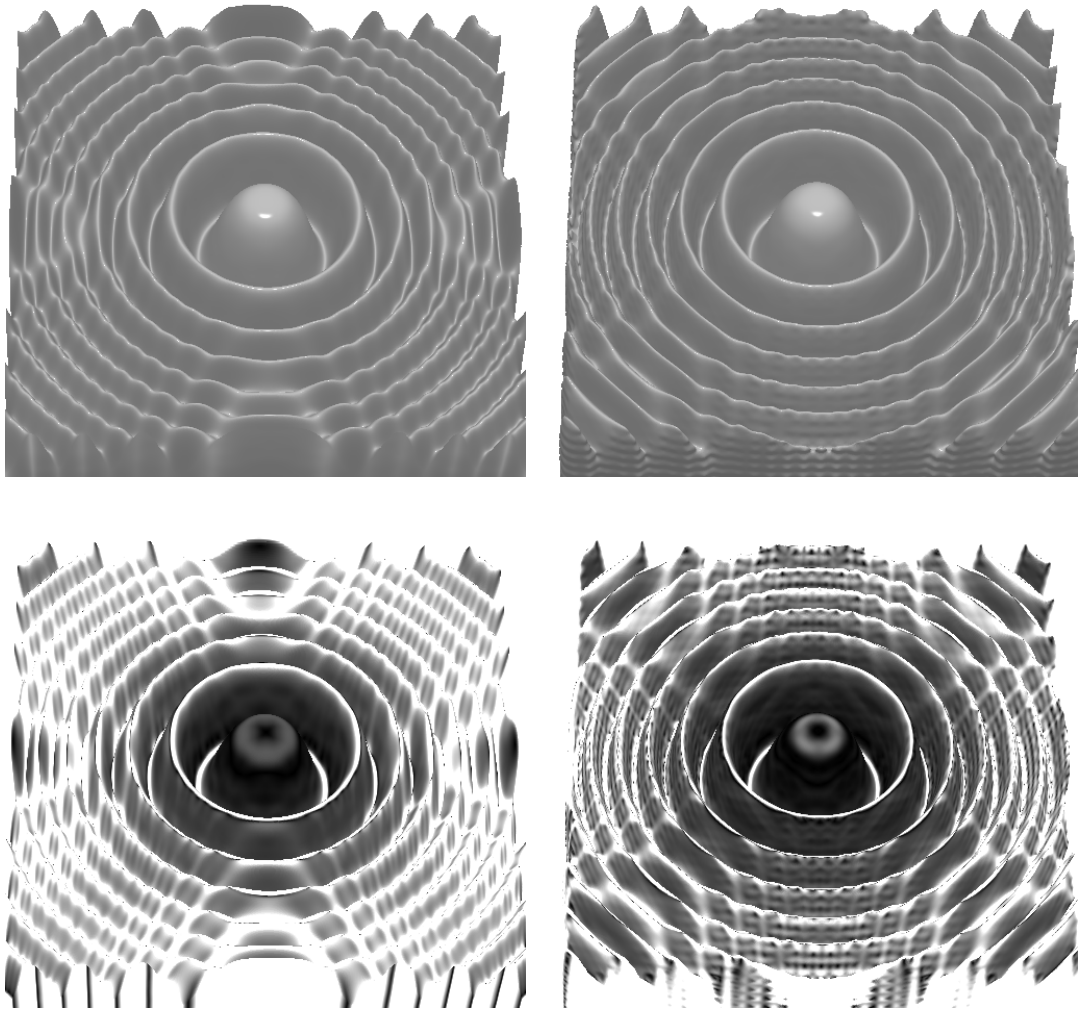
Physical objects or body tissues are scanned and reconstructed on the Cartesian lattice. There are filtering steps involved in scanning and reconstruction that tune the data according to the Cartesian sampling so the spectrum of the captured data is anti-aliased with respect to the geometry of the Cartesian lattice. To perform anti-aliasing before sampling on the BCC lattice, one needs to consider the Nyquist region of BCC sampling which is the Voronoi cell of its reciprocal lattice. Therefore, the ultimate test of the BCC reconstruction on real-life datasets can not be performed until there are true BCC sampling scanners available.

Nevertheless, we constructed comparable BCC and Cartesian datasets by merely subsampling a fairly densely sampled Cartesian dataset. Cartesian sampled data can be downsampled onto a BCC lattice by retaining Cartesian points whose x, y, z coordinates are all odd or all even. Such a BCC lattice has a quarter of the sampling density of the original



(a) Cartesian (b) BCC

Figure 3.12: The Marschner-Lobb dataset. (a) Sampled on the Cartesian lattice at the resolution of $41 \times 41 \times 41$. (b) Sampled on the BCC lattice at the resolution of $32 \times 32 \times 64$. The first row illustrates the volume rendering of the sampled data using the tricubic B-spline on the Cartesian and our quintic box spline on the BCC dataset. The second row illustrates the corresponding angular errors in estimating the gradient on the iso-surface from the reconstruction process. An angular error of .3 radians is mapped to white. The darker error image of the BCC data confirms smaller errors and more accurate reconstruction in the BCC pipeline.



(a) Cartesian (b) BCC

Figure 3.13: Quasi-interpolation on the ML dataset. (a) Sampled using the functional in Equation (3.12) on the Cartesian lattice at the resolution of $41 \times 41 \times 41$. (b) Sampled using the functional in Equation (3.11) on the BCC lattice at the resolution of $32 \times 32 \times 64$. The first row illustrates the volume rendering of the sampled data using the tricubic B-spline on the Cartesian and our quintic box spline on the BCC dataset. The second row illustrates the corresponding angular errors in estimating the gradient on the iso-surface from the reconstruction process. An angular error of .3 radians is mapped to white. The darker error image of the BCC data confirms smaller errors and more accurate reconstruction.

dataset. For obtaining an almost equivalent subsampling ratio into a lower resolution Cartesian dataset, we choose a rational subsampling scheme where each dimension of the original Cartesian dataset is subsampled by 63/100 since $(63/100)^3 = 0.250047 \approx 1/4$. To achieve this subsampling, we first upsampled by zero-padding in the frequency domain by a factor of 63. Then a subsampling of the rate 1/100 yields the properly subsampled Cartesian volume.

As a first practical dataset, we took the Boston Teapot dataset. The original dataset has a resolution of $162 \times 162 \times 113$. The subsampled Cartesian volume has a resolution of $103 \times 103 \times 72$ and the subsampled BCC volume has a resolution of $81 \times 81 \times 113$. These volumes were rendered using the tricubic B-spline on the Cartesian lattice and the quintic box spline on the BCC lattice in Figure 3.14. These images demonstrate the superiority of the BCC sampling scheme since the Cartesian undersampled dataset developed cracks on the surface of the teapot lid, while the BCC undersampled dataset maintains the original content much better. We also examined the above pipeline on the carp fish dataset. The original dataset has a resolution of $256 \times 256 \times 256$. The subsampled Cartesian volume has a resolution of $140 \times 140 \times 140$ and the subsampled BCC volume has a resolution of $111 \times 111 \times 222$. These volumes were rendered using the tricubic B-spline on the Cartesian lattice and the quintic box spline on the BCC lattice in Figure 3.15. Again, these results show the superiority of the BCC sampling scheme since the Cartesian undersampled dataset misses the fish tail and most of the bones.

In [30] we have discussed the issues pertaining to linear type interpolation. While in the Cartesian volumes they demonstrate grid-aligned artifacts, in BCC they display girdering artifacts [10]. The result of linear box spline on the BCC and trilinear B-spline interpolation on the Cartesian lattice are demonstrated in Figure 3.16. In this figure, the girdering artifacts are prominent in the teapot surface, and the trilinear reconstruction provides a more satisfying reconstruction. However, in the carp dataset, it is apparent that the linear box spline has preserved the fine details of the fish bone more accurately than the trilinear interpolant. Similarly, the ML dataset was tested with the trilinear B-spline on the Cartesian lattice versus, the linear box spline on the BCC lattice in Figure 3.17. Here we note that the error image of the BCC lattice is darker and hence the gradient estimation is more accurate in this case. The presence of the girdering effect [10] in the linear box spline solution creates artifacts on the surface of the ML dataset; these artifacts are less noticeable in the Cartesian reconstruction with trilinear B-spline.

We have also approximated the mean square error existing in the volumes subsampled

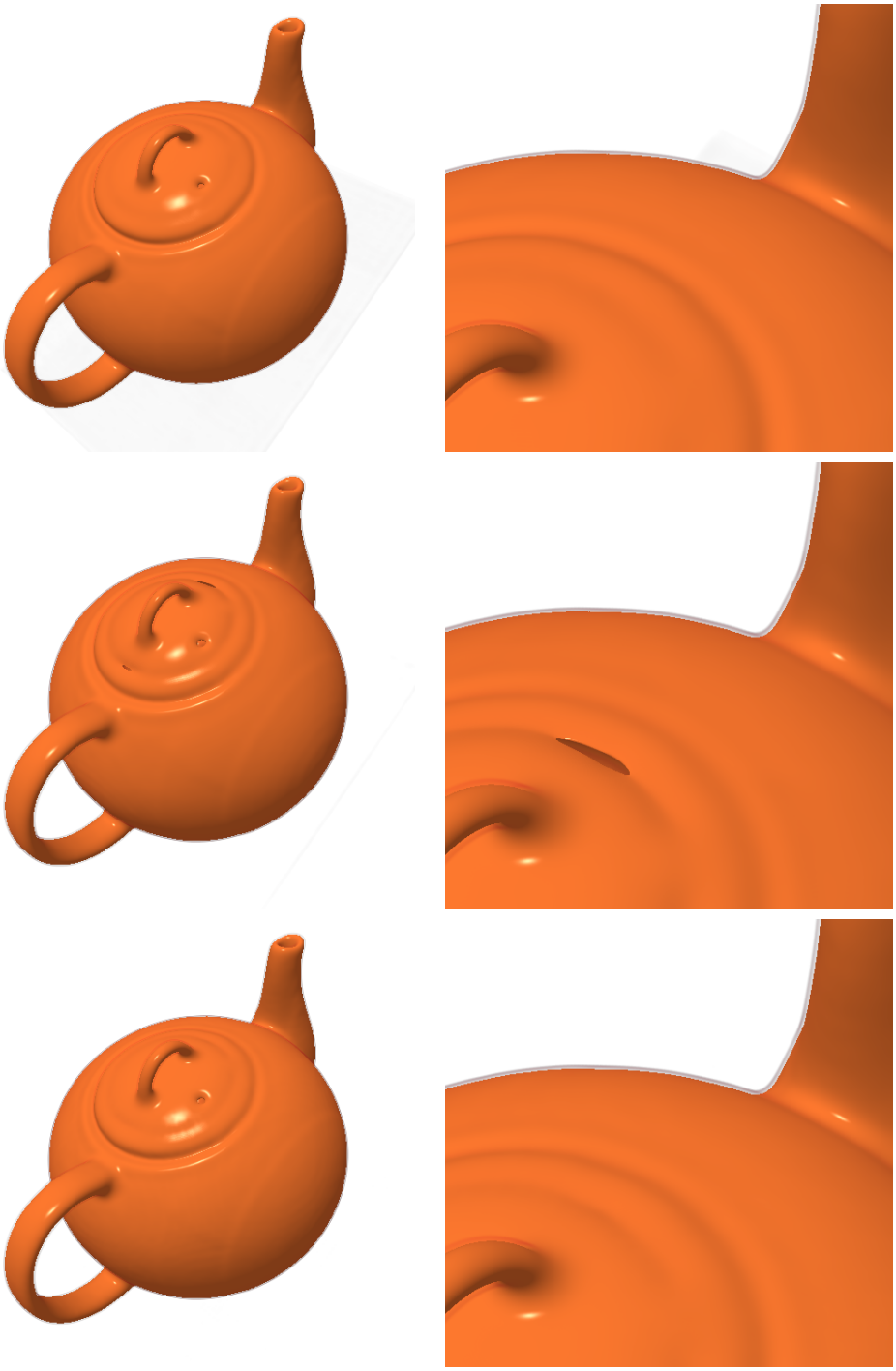
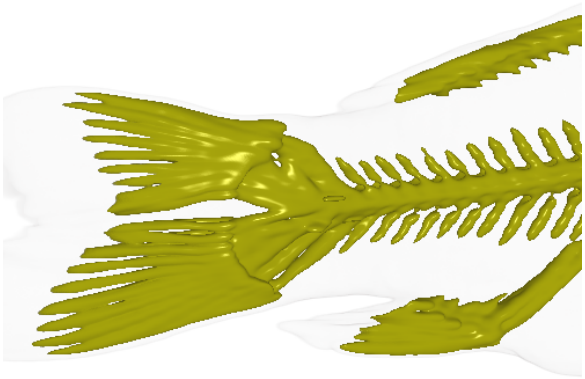
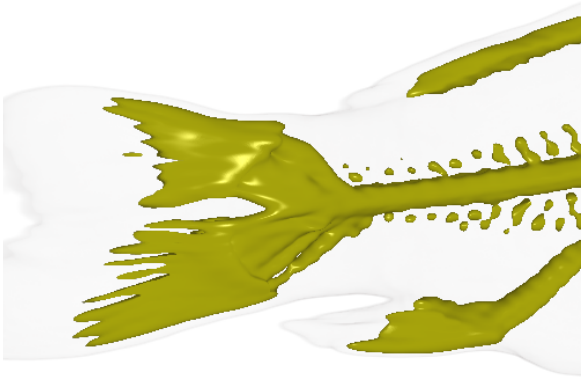


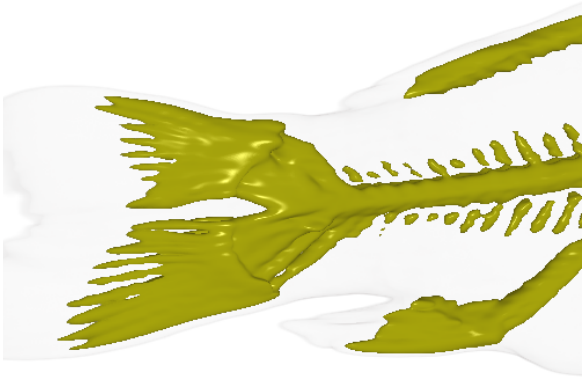
Figure 3.14: The Boston teapot dataset. First row: The original Cartesian sampled dataset with $2,965$K samples reconstructed with the tricubic B-spline. Second Row: Undersampled on the Cartesian lattice with 763K samples reconstructed with the tricubic B-spline. Third Row: Undersampled on the BCC lattice with 741K samples reconstructed with the quintic box spline.



(a)

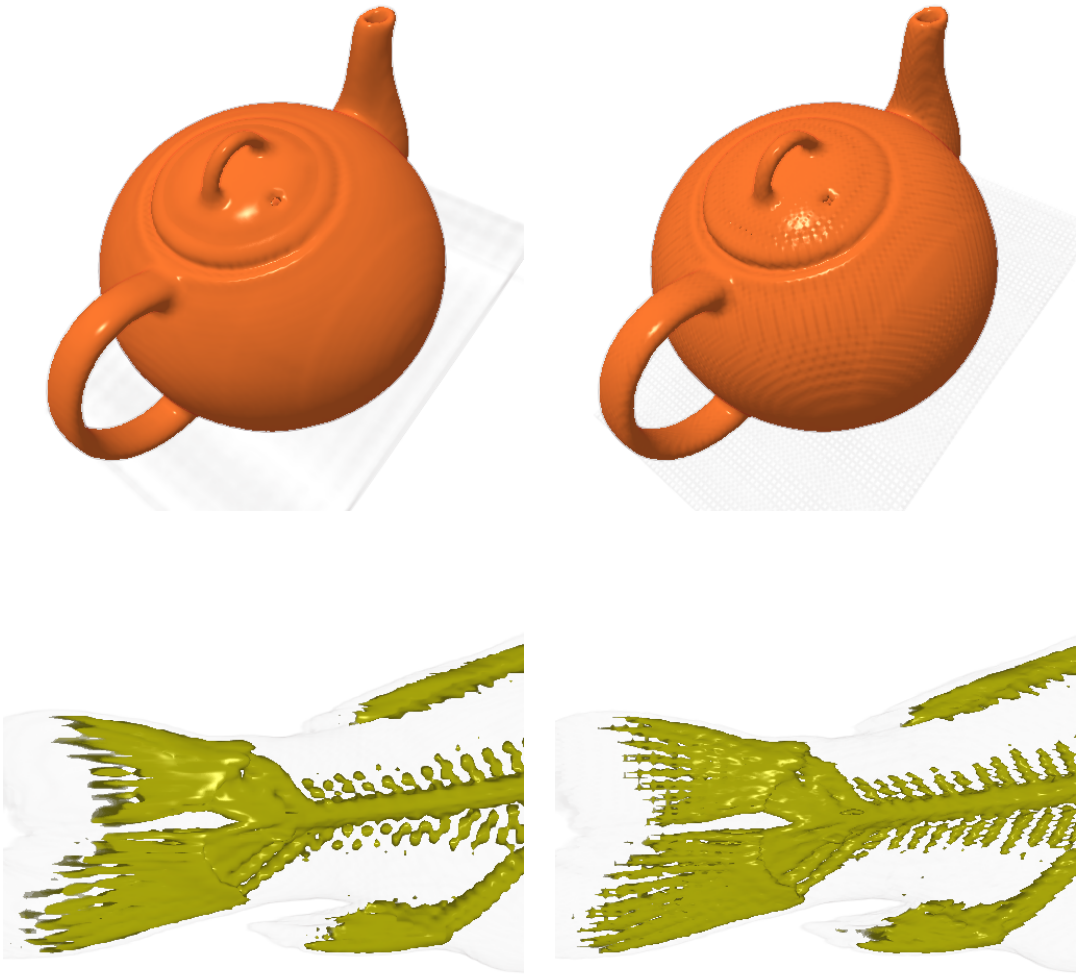


(b)



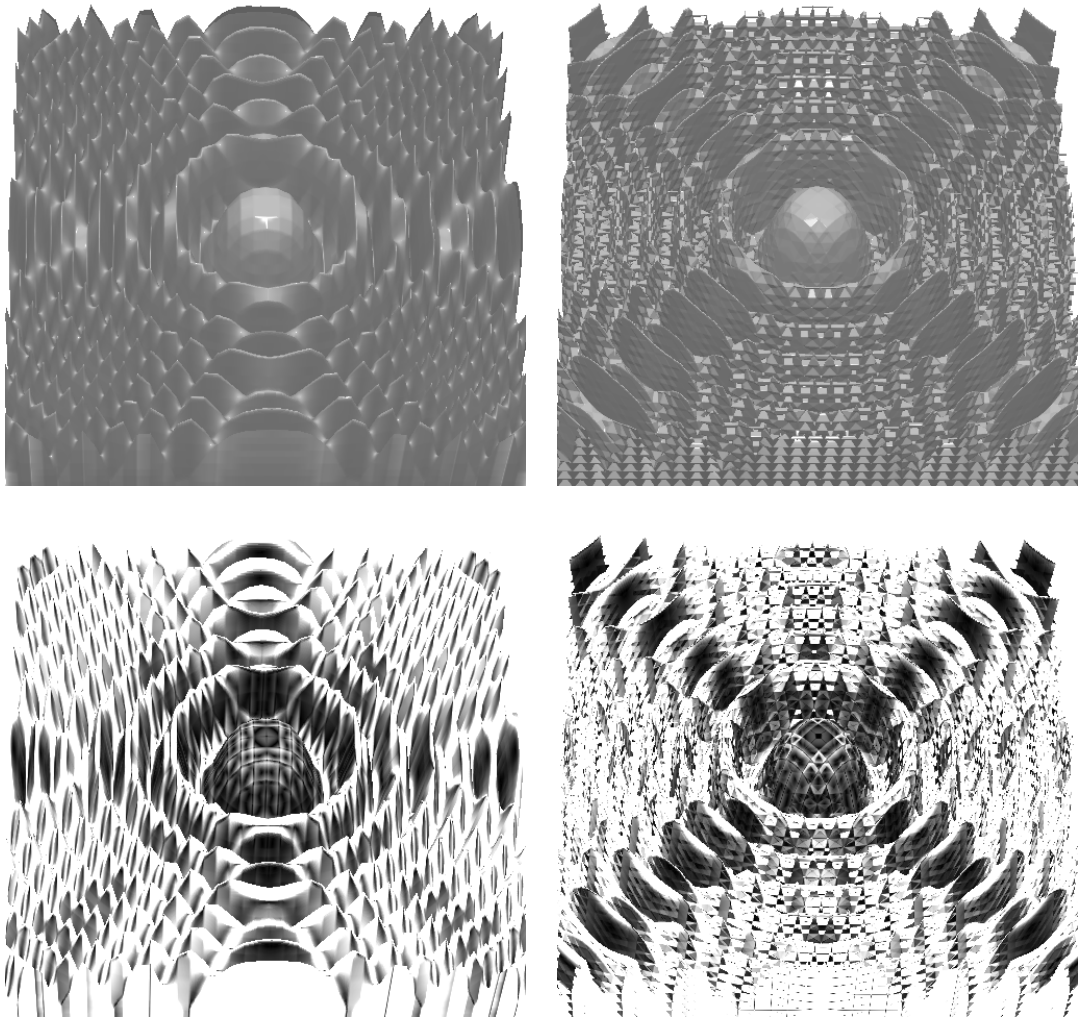
(c)

Figure 3.15: The carp fish dataset. (a) The original Cartesian sampled dataset with $16,777$K samples reconstructed with the tricubic B-spline. (b) Undersampled on the Cartesian lattice with $2,744$K samples reconstructed with the tricubic B-spline. (c) Undersampled on the BCC lattice with $2,735$K samples reconstructed with the quintic box spline.



Cartesian BCC

Figure 3.16: Trilinear B-spline versus linear box spline reconstructions. The linear type reconstruction of volumes in Figure 3.14 and Figure 3.15.



(a) Cartesian (b) BCC

Figure 3.17: The Marschner-Lobb dataset. (a) Sampled on the Cartesian lattice at the resolution of $41 \times 41 \times 41$. (b) Sampled on the BCC lattice at the resolution of $32 \times 32 \times 64$. The first row illustrates the volume rendering of the sampled data using the trilinear B-spline on the Cartesian and our linear box spline on the BCC dataset. The second row illustrates the corresponding angular errors in estimating the gradient on the iso-surface from the reconstruction process. An angular error of .3 radians is mapped to white.

on BCC and Cartesian lattices. The error calculation was carried out by a random sampling of the error and summing over these random points to approximate the error integral. These experiments also confirmed that the BCC subsampling is more accurate than the comparable Cartesian subsampling since the error of the Cartesian subsampled volume matched that of the BCC volume with only about 70% of the number of samples. Further, we have examined the visual quality of the rendered images and found empirical evidence that a BCC sampled volume with roughly about 70% of the number of samples of a Cartesian volume produces equivalent visual quality [62].

Computational Cost The computational cost of the reconstruction is mainly due to computing the convolution of the data values and the continuous-domain box spline kernel.

For trilinear and tricubic B-spline reconstructions on the Cartesian lattice, a neighborhood of $2 \times 2 \times 2 = 8$ and $4 \times 4 \times 4 = 64$ points fall inside the support of the kernels, respectively. Therefore, eight terms of the convolution in the case of the trilinear and sixty-four terms in the case of the tricubic B-spline need to be computed. Computing the convolution weights involves evaluating a *third* degree trivariate polynomial for trilinear while a *ninth* degree trivariate polynomial needs to be evaluated for the tricubic B-spline. However, due to the tensor-product structure of these kernels, the third degree polynomial, in the case of the trilinear interpolation, factors into a product of three first degree univariate polynomials. Similarly, the ninth degree trivariate polynomial of the cubic B-spline factors into a product of three third degree univariate polynomials.

For linear and quintic box spline reconstructions on the BCC lattice, a neighborhood of 4 and 32 points falls inside the support of the kernels, respectively. Therefore, only four terms of the convolution in the case of linear and thirty two terms in the case of the quintic box spline need to be computed. Computing the convolution weights involves evaluating a *first* degree trivariate polynomial for the linear box spline while a *fifth* degree trivariate polynomial needs to be evaluated for the quintic box spline. However, due to the structure of the quintic box spline, the fifth degree polynomial is factored into the product of a third degree polynomial and a second degree polynomial as Equation (3.25) can be factored in terms of the z variable. All of the polynomial pieces of the quintic box spline listed in Section 3.2.5 are in the form of this building block function.

Our experiments also support these predictions as the Cartesian dataset in Figure 3.12(a) was rendered in 122.69 seconds while the BCC dataset in Figure 3.12(b) was rendered in

Table 3.1: Rendering Times. C^0 and C^2 indicate the linear and quintic box splines on the BCC lattice and the trilinear and tricubic B-splines on the Cartesian lattice respectively.

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

63.75 seconds. These images were computed at a resolution of 500×500 on a dual processor (Dual Core AMD Opteron 280) machine running Linux with a GCC compiler (4.0.2). A similar rendition using trilinear interpolation on the Cartesian image took 21.49 seconds while the linear box spline on the BCC took 11.99 seconds. Similar timing differences were observed on the real-life datasets; the timings for these reconstructions are summarized in Table 3.1. We note that for C^0 reconstructions, the speedups are less than a factor of two. Since the computational cost of the linear interpolation is relatively light, a smaller portion of the rendering time is consumed by the reconstruction step; hence, twice a speedup in reconstruction plays a slightly less significant role in the rendering time.

Chapter 4

Other Trivariate Box Splines

In this chapter, we will see a set of box splines that are suitable for reconstruction on the FCC and Cartesian lattices. We will state their properties when used in reconstruction and motivate their applications and advantages in a practical setting.

The box splines presented in this chapter were evaluated on a discretized domain and stored on an array in an offline stage. The explicit piecewise polynomial representation of these box splines is subject of future research. We believe the main principle of the method employed in Section 3.2 can be applied for efficient evaluation of these box splines. The discretized approximation to the continuous-domain box splines are used as a lookup table for reconstruction in the rendering pipeline. The use of pre-computed discretized kernels is not uncommon in the field of volume rendering (e.g., most hardware-based rendering algorithms [69, 37]). McCool [57] has also used discretized box splines for efficient splatting during the volume rendering process. In our experiments, a resolution of 50 points per unit interval ($[0, 1]$) of the support of the box spline (on each axis) was sufficient for discretizing the box spline's domain. Higher resolutions did not improve the visual quality of the rendered images.

4.1 Box Spline Reconstruction on the FCC Lattice

While the BCC lattice is the lattice of choice for generic sampling pipelines (see Section 2.2.1), among the Cartesian, FCC and BCC lattices, the FCC lattice is the second best choice for this purpose. This is due to its second best ranking when it comes to the volume of the inscribed sphere to its Brillouin zone (see Appendix A). Moreover, when it

comes to generic sampling pipelines in the presence of aliasing, the FCC sampling allows for the least aliasing in the spectrum of the sampled function. This is due to the fact that its dual BCC lattice covers the 3-D space with the minimum amount of overlap between the spectra. This can also be observed from the best approximation of the circumscribing sphere that the Brillouin zone of the FCC lattice achieves (see Appendix A). The FCC lattice has also been proposed for sampling video signals by Kovačević [48] and Cooklev [18]. Therefore, in this section we will consider the reconstruction problem on the FCC lattice.

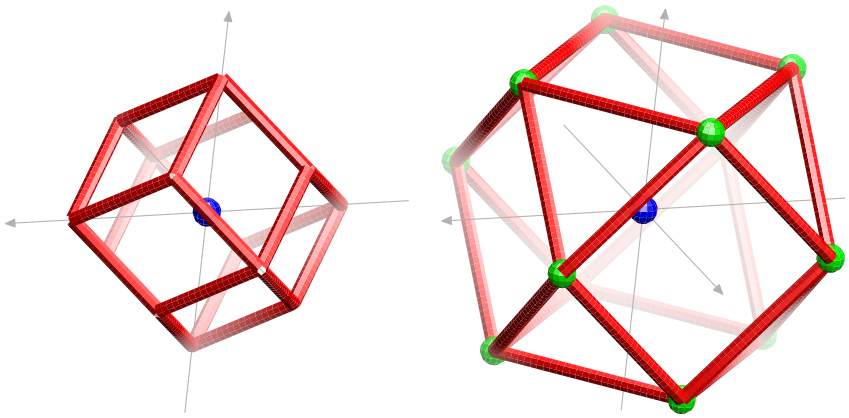
We studied four-direction box splines in Section 3 for their guaranteed smoothness and approximation power on the BCC lattice. These box splines were discovered by realizing that the *first neighbors cell* of a lattice point on the BCC lattice forms a rhombic dodecahedron. In other words, all lattice points whose Voronoi cells share a non-degenerate face with the Voronoi cell of the center point lie on the vertices of a rhombic dodecahedron. The rhombic dodecahedron is a zonohedron with four zones. This allowed the construction of a four-direction box spline with a rhombic dodecahedral support.

4.1.1 Linear Element Polyhedral Spline

The 12 vertices of the first ring of neighbors in the FCC lattice form a cuboctahedron as shown in Figure 4.1. The Voronoi cells of these 12 vertices (green) share a face with the Voronoi cell of the center (blue). If we consider the center lattice point to be at (x, y, z), then these 12 neighbors can be visited by shifts of $(\pm 1, \pm 1, 0)$, $(\pm 1, 0, \pm 1)$, $(0, \pm 1, \pm 1)$ or in short form, all permutations of $(\pm 1, \pm 1, 0)$. The 12-point neighborhood of an FCC lattice point (x, y, z) is formed by all lattice points (x', y', z') that satisfy:

$$(x - x')^2 + (y - y')^2 + (z - z')^2 = 2$$

The cuboctahedron is not a zonohedron since it has triangular faces; hence, one can not construct a linear box spline element with a cuboctahedron support. However, one can construct a linear spline element by a three-dimensional projection of a 24-cell [20]. This polyhedral spline was the basis for the reconstruction algorithm we proposed in [75]. When shifted to a general position, out of the 12 boundary points, only three of them accompany the center point for this linear reconstruction. Hence a total of $1 + 3 = 4$ points fall inside the support of this kernel. Due to its non-trivial polyhedral spline structure, studying its smoothness and approximation power is cumbersome. This difficulty is mainly due to the



(a) (b)

Figure 4.1: (a) Rhombic Dodecahedron: The Voronoi cell of the FCC lattice. (b) Cuboctahedron: The first neighbor cell of an FCC lattice point (blue) consists of all lattice points (green) whose Voronoi cell shares a face with the center point. A corresponding polyhedral spline can be constructed by projecting a 4-D 24-cell to 3-D. (i.e., a linear element with a cuboctahedral support.)

non-trivial Fourier transform of the indicator function of a 24-cell, whereas, in the case of box splines the Fourier transform of a hypercube is trivially known in any dimension.

4.1.2 The Six-Direction C^1 Box Spline

By expanding the neighborhood to include 6 additional lattice points that share a vertex with the Voronoi cell of the center, we find a truncated octahedron. These 6 neighbors are at offsets that are all permutations of $(\pm 2, 0, 0)$. Together with the 12-point neighborhood, they form an 18-point neighborhood of points (x', y', z') that satisfy:

$$(x - x')^2 + (y - y')^2 + (z - z')^2 \leq 4$$

Since all of the faces of the truncated octahedron exhibit point symmetry, it is a zonohedron with 6 zones. Therefore it can be constructed by projecting a six dimensional hypercube. The corresponding box spline can then be represented by the direction vectors representing these 6 zones:

$$\Xi_6 = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_6] = \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix} \tag{4.1}$$

The truncated octahedral support of this box spline is shown in Figure 4.2.

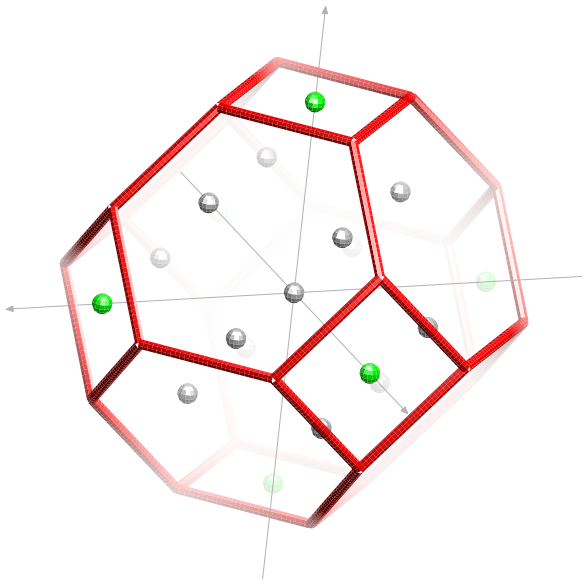


Figure 4.2: Truncated Octahedron: The support of the C^1 box spline whose shifts on the FCC lattice achieve a third order approximation.

Support

When shifted to a general position, in addition to the center and the 12 first neighbors, 3 of the 6 boundary FCC lattice points[1] on the surface of the truncated octahedron fall inside the support of this kernel. Therefore, a total of $1+12+3=16$ FCC lattice points fall inside the support of this box spline kernel.

Smoothness and Approximation Power

By inspecting these direction vectors, we realize that the vectors $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_6$ are co-planar. Therefore, the minimum number of directions to remove from $\boldsymbol{\Xi}_6$ so that the remaining directions would not span $\mathbb{R}^3$ is $\rho=3$ (for instance by removing $\boldsymbol{\xi}_2$,$\boldsymbol{\xi}_4$ and $\boldsymbol{\xi}_5$). This means that the box spline $M_{\boldsymbol{\Xi}_6}$ leads to a C^1 reconstruction.

The center of this box spline is $\frac{1}{2}\sum_{\boldsymbol{\xi}\in\boldsymbol{\Xi}_6}\boldsymbol{\xi}$. To center this box spline at the origin, we can introduce a shift in the space domain (which amounts to a phase shift in the frequency

[1]These 6 boundary points are the FCC points whose Voronoi cell shares a vertex with the vertex of the center point; hence they are called vertex-connected neighbors.

domain). Then the Fourier transform of this centered box spline is:

$$\hat{M}^{c}_{\Xi_6}(\boldsymbol{\omega}) = \prod_{\boldsymbol{\xi}\in\Xi_6} \operatorname{sinc}\left(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\omega}\right) \tag{4.2}$$

Recall that $\operatorname{sinc}(t) = \sin(t/2)/(t/2)$. In order to avoid any signal shifts, we use this centered version of the box spline during reconstruction.

When a function is represented by its samples on the FCC lattice, described by Equation (1.1), the replicates of the primary spectrum lie on the dual BCC lattice points which are scaled by π; this means that the aliasing frequencies are frequencies whose three coordinates are all even or all odd multiples of π.

To establish the approximation power of this box spline on the FCC lattice, we need to examine the order of zeros of its Fourier transform at the aliasing frequencies. By examining the axis-aligned neighboring aliasing frequencies of $(\pm 2\pi, 0, 0)$, one can show that $\operatorname{sinc}(\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{\omega})$, $\operatorname{sinc}(\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{\omega})$, $\operatorname{sinc}(\boldsymbol{\xi}_3^{\mathrm{T}}\boldsymbol{\omega})$ and $\operatorname{sinc}(\boldsymbol{\xi}_4^{\mathrm{T}}\boldsymbol{\omega})$ vanish at these aliasing frequencies. Similarly we obtain four vanishing moments from $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, $\boldsymbol{\xi}_5$ and $\boldsymbol{\xi}_6$ for the aliasing frequencies at $(0, \pm 2\pi, 0)$ and four vanishing moments at the aliasing frequencies at $(0, 0, \pm 2\pi)$. However, at the aliasing frequencies whose coordinates are all odd multiples of π we only get three vanishing moments: for instance, at (π, π, π), only the sinc functions corresponding to $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_5$ vanish. Therefore, at every aliasing frequency we have a minimum of *three* vanishing moments; hence the shifts of this box spline on the FCC lattice provides a space whose approximation order is three.

The smoothness and approximation power of the six-direction box spline matches that of the triquadratic B-spline on the Cartesian lattice. However, the support of the triquadratic B-spline on the Cartesian lattice covers a total of $3\times3\times3 = 27$ points. This neighborhood is significantly reduced to 16 in case of the box spline on the FCC lattice, while maintaining the same smoothness and approximation power. This fact reflects the striking computational advantage of the FCC lattice (similar to the BCC lattice) over the commonly used Cartesian lattice.

4.1.3 The Nine-Direction C^3 Box Spline

By adding the three unit directions to the direction set of the six-direction box spline we increase the polynomial degree of the box spline while expanding the neighborhood of

influence. The resulting nine-direction box spline is represented by

$$\begin{aligned}\boldsymbol{\Xi}_9 &= [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_9] \\ &= \begin{bmatrix} 1 & -1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{4.3}$$

Since $\boldsymbol{\Xi}_9 = [\boldsymbol{\Xi}_6, \boldsymbol{I}]$, we have:

$$M_{\boldsymbol{\Xi}_9}(\boldsymbol{x}) = (M_{\boldsymbol{\Xi}_6} * M_{\boldsymbol{I}})(\boldsymbol{x})$$

where $\boldsymbol{I}$ indicates the 3×3 identity matrix and $M_{\boldsymbol{I}}$ is simply the indicator function of the unit cube in $\mathbb{R}^3$. Therefore, the support of this box spline is the Minkowski sum of the truncated octahedron (Figure 4.2) and a unit cube which is a *truncated cuboctahedron*. This polyhedron is shown in Figure 4.3. One can demonstrate the fact that the truncated cuboctahedron is a sum of a cube and a truncated octahedron by deriving the zones of the truncated cuboctahedron.

Support

When centered at an FCC lattice point, its 12 first neighbors and 6 vertex-connected neighbors fall in the interior of the support of this box spline. Moreover, 24 FCC lattice points fall *inside* the kernel's support; these points are such that their Voronoi cells are away from the Voronoi cell of the center by the distance of one edge of the Voronoi cell. These neighbors are at offsets that are permutations of $(\pm 2, \pm 1, \pm 1)$. Three of these points are shown as blue points in Figure 4.3. The 24-point neighborhood are all points (x', y', z') such that:

$$(x - x')^2 + (y - y')^2 + (z - z')^2 = 6$$

Moreover 12 FCC lattice points fall on the boundary of the support; these lattice points are at offsets generated by permutations of $(\pm 2, \pm 2, 0)$. This secondary 12-point neighborhood (three of which are displayed in green in Figure 4.3) are all FCC lattice points (x', y', z') such that:

$$(x - x')^2 + (y - y')^2 + (z - z')^2 = 8$$

Only 3 points from the secondary 12-point neighborhood fall inside the support when the kernel is moved to a general position. Therefore, a total of $1 + 12 + 24 + 3 = 40$ points fall inside this kernel.

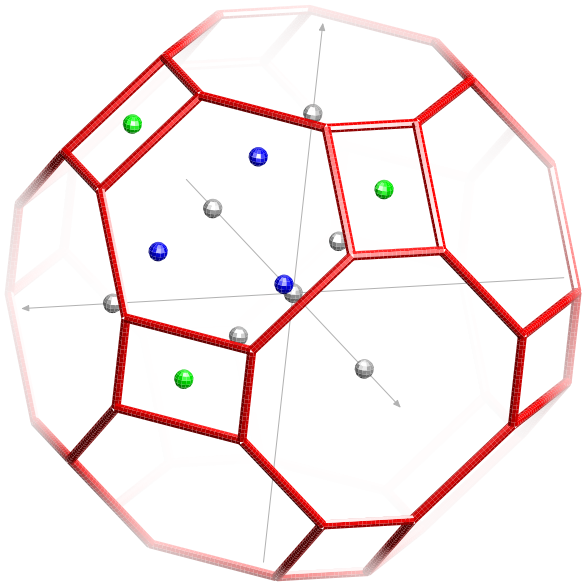


Figure 4.3: Truncated Cuboctahedron: The support of the C^3 box spline whose shifts on the FCC lattice achieves a third order approximation. To avoid cluttering, only participating neighbors in one octant are drawn.

Smoothness and Approximation Power

By inspecting these direction vectors, we realize that the minimum number of directions to remove from Ξ_9 so that the remaining directions would not span $\mathbb{R}^3$ is $\rho = 5$ (for instance, $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, $\boldsymbol{\xi}_7$ and $\boldsymbol{\xi}_8$ are co-planar). This means that the box spline M_{Ξ_9} leads to a C^3 reconstruction. We are inheriting three vanishing moments from the original six directions; the new three directions do not increase the vanishing moments at aliasing frequencies which are odd multiples of π. For instance for the aliasing frequency at (π, π, π), we still have three vanishing moments generated by $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_5$. Therefore, the FCC shifts of this box spline still generate a third-order approximation space.

This combination of smoothness and approximation power of the nine-direction box spline on the FCC lattice is not common for the tensor-product B-spline solutions.

4.1.4 Reconstruction Results

In this section, we examine the C^1 and C^3 box spline reconstructions on the benchmark dataset of Marschner-Lobb on the FCC lattice. Furthermore, we compare and contrast with similar sampling density data on the Cartesian lattice. Since, the smoothness and

approximation power of the nine-direction box spline on the FCC lattice is not matched by the common tensor-product B-spline solutions, we will compare against the commonly used triquadratic and tricubic B-spline reconstructions.

Figure 4.4 demonstrates the sampling of the ML dataset at a resolution of $41 \times 41 \times 41$ on the Cartesian lattice rendered with the triquadratic B-spline on the first row. The second row shows the sampled volume of $25 \times 25 \times 100$ samples on the FCC lattice rendered using the six-direction box spline. The error images are also depicting the angular errors that occurred when estimating the gradient of the trivariate function at the surface. A maximum angular error of .3 radians is mapped to white. The darker error image from the FCC pipeline shows a more accurate reconstruction. For the sake of comparison with the commonly-used tricubic B-spline reconstruction, Figure 4.5 shows the nine-direction box spline reconstruction on the FCC lattice versus the tricubic B-spline on the Cartesian lattice. The box spline reconstruction clearly shows darker error images hinting at a more accurate reconstruction. Note that the tricubic B-spline reconstruction involves a neighborhood of $4 \times 4 \times 4$ Cartesian points while the nine-direction box spline neighborhood contains 40 FCC lattice points.

4.2 Box Spline Reconstruction on the Cartesian Lattice

4.2.1 The Seven-Direction C^2 Box Spline

In this section, we shall present a seven-direction box spline for reconstruction on the Cartesian lattice. The construction of the seven-direction box spline is motivated by the construction of the Zwart-Powell element (see Section 2.3.3) which is formed by direction vectors of the unit square and its diagonals. In 3-D, the construction is also a box spline whose direction vectors form a cube (that is the Voronoi cell of the 3-D Cartesian lattice) and its four diagonals, see Figure 4.6. While one could extend the 2-D ZP element by adding other directions in 3-D, we have chosen the diagonal directions specifically in order to get a good approximation to a spherically symmetric reconstruction kernel. This choice of box spline directions is represented by the matrix:

$$\Xi_7 = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 \end{bmatrix}. \tag{4.4}$$

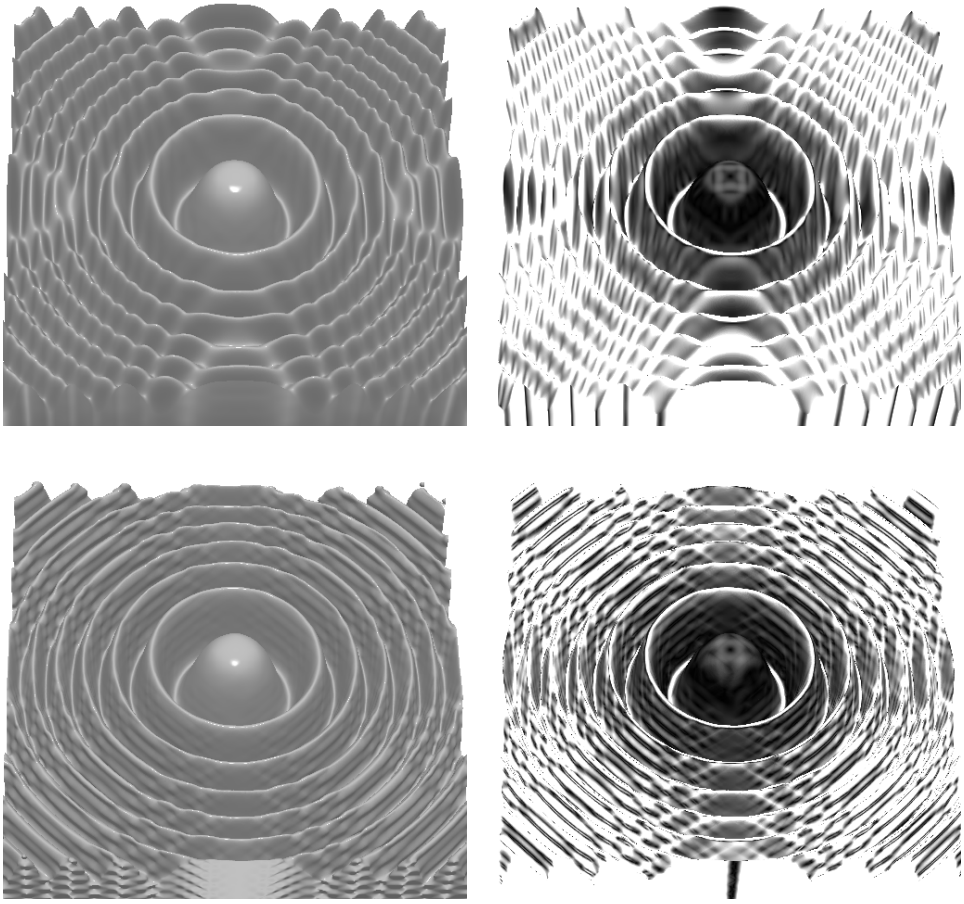


Figure 4.4: C^1 reconstructions. The first row shows the ML dataset at a Cartesian resolution of $41 \times 41 \times 41 = 68.9K$ (left) and its error image (right) rendered with the triquadratic B-spline. The second row shows the ML data sampled on the FCC lattice with a resolution of $25 \times 25 \times 100 = 62.5K$ rendered with the six-direction box spline. The max angular error of .3 radians is mapped to white.

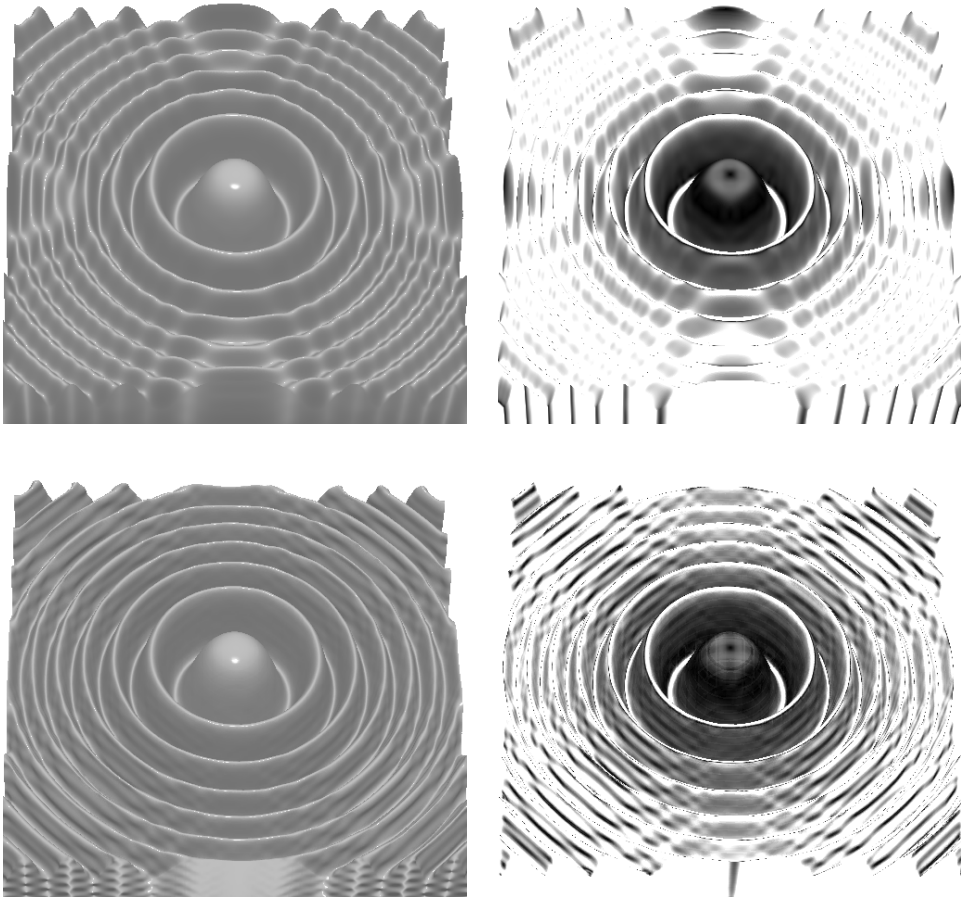


Figure 4.5: C^2 reconstruction on the Cartesian lattice versus a C^3 reconstruction on the FCC lattice. The first row shows the ML dataset at a Cartesian resolution of $41 \times 41 \times 41 = 68.9K$ (left) and its error image (right) rendered with the tricubic B-spline. The second row shows the ML data sampled on the FCC lattice with a resolution of $25 \times 25 \times 100 = 62.5K$ rendered with the nine-direction box spline. The max angular error of .3 radians is mapped to white.

Peters exploits the advantage of this box spline for constructing a curvature-continuous surface by tracing its zero sets [71]. He also describes a subdivision scheme that can be used to approximate this box spline at a particular point.

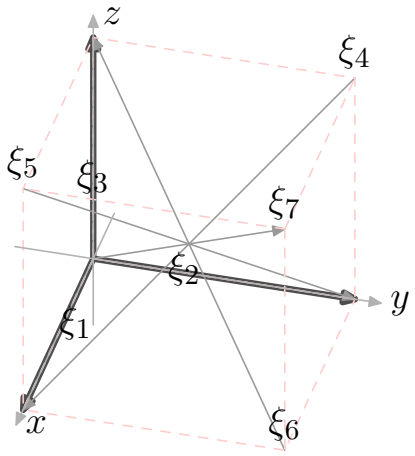


Figure 4.6: The seven directions of the box spline. The first three directions ($\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$) are axis aligned and form a cube. The last four are the antipodal diagonal directions.

Support

We note that the first three direction vectors of $\boldsymbol{\Xi}_7$ are along the coordinate axes, and the last four direction vectors are the ones on the diagonals of a cube. We can decompose this box spline matrix into two sub-matrices where the axis-aligned vectors are separated from the diagonal ones:

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{\Xi}_4 = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

Due to the convolution property of box splines as in Equation (2.15), the seven-direction box spline can be written as the convolution of the box splines corresponding to these matrices:

$$M_{\boldsymbol{\Xi}_7}(\boldsymbol{x}) = \left(M_{\boldsymbol{I}} * M_{\boldsymbol{\Xi}_4}\right)(\boldsymbol{x}) \tag{4.5}$$

We can recognize $M_{\boldsymbol{I}}$ as the trivariate box function; i.e., the indicator function of the interval $[0,1)^3$. Moreover $M_{\boldsymbol{\Xi}_4}$ is the linear box spline that was introduced for reconstruction on the BCC lattice in Section 3. The support of this box spline (when centered at the origin)

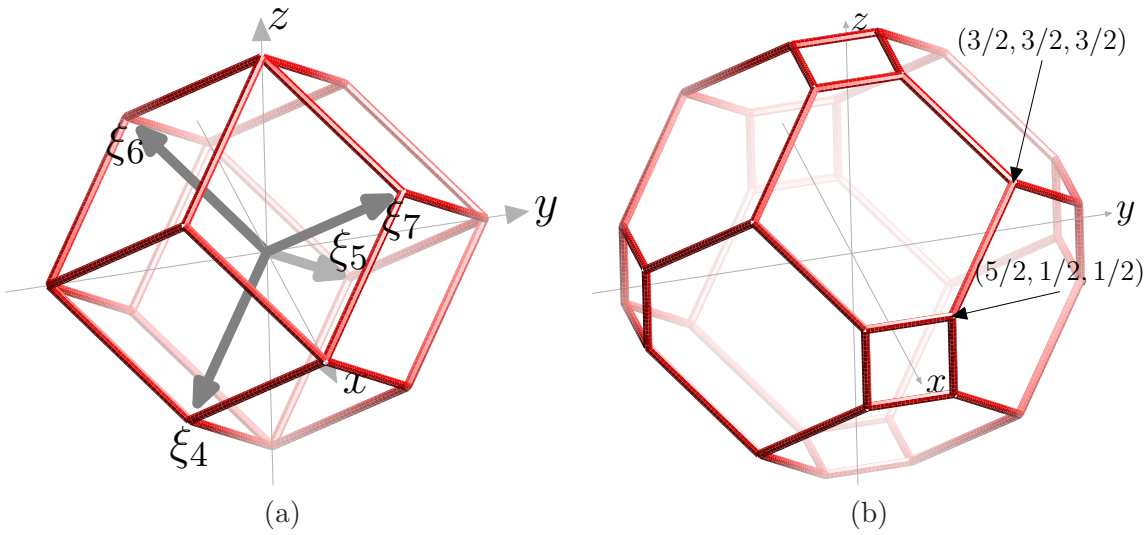


Figure 4.7: (a) The four diagonal directions give rise to a box spline whose support is a rhombic dodecahedron. (b) The support of the seven-direction box spline is a truncated rhombic dodecahedron which is the Minkowski sum of a cube and rhombic dodecahedron.

is a rhombic dodecahedron that is contained in the interval $[-2,2)^3$; see Figure 4.7(a). Since M_{Ξ_7} is the convolution of $M_{\boldsymbol{I}}$ with M_{Ξ_4}, its support is the sum of their supports. In particular, the support of M_{Ξ_7} is contained in $[-5/2,5/2)^3$.

In [71], the support of the seven-direction box spline is considered an octahedron. However, since the support of M_{Ξ_4} is a rhombic dodecahedron as displayed in Figure 4.7(a) and the support of $M_{\boldsymbol{I}}$ is a cube, the support of M_{Ξ_7} is the Minkowski sum of a rhombic dodecahedron and a cube which is a *truncated rhombic dodecahedron* as displayed in Figure 4.7(b).

When this box spline is shifted to a general position, a total of 53 Cartesian points fall inside of its support. This is the direct result from [24, II.15] since the volume of the truncated rhombic dodecahedron is 53. The truncated rhombic dodecahedron is obtained from a rhombic dodecahedron whose bounding cube is $[-3,3]^3$. The volume of the rhombic dodecahedron is a quarter of this cube (see Appendix A) which is 54. The volume of each of the six pyramids cut from the rhombic dodecahedron is 1/6 which adds up to 1; therefore, the volume of the truncated rhombic dodecahedron is 53.

Smoothness and Approximation Power

The seven-direction box spline offers a C^2 reconstruction since $\boldsymbol{\xi}_1, \boldsymbol{\xi}_4$ and $\boldsymbol{\xi}_7$ are co-planar, hence $\rho = 7 - 3 = 4$. This smoothness parallels that of the tricubic B-spline. However

the seven-direction box spline only consists of polynomials of degree four, but the tricubic B-spline consists of degree nine polynomials (i.e., product of three univariate cubic polynomials).

The center of this box spline is $\frac{1}{2}\sum_{\boldsymbol{\xi}\in\Xi_7}\boldsymbol{\xi}$. To center this box spline at the origin, we can introduce a shift in the space domain (which amounts to a phase shift in the Fourier domain). Then the Fourier transform of this centered box spline is:

$$\hat{M}^c_{\Xi_7}(\boldsymbol{\omega}) = \prod_{\boldsymbol{\xi}\in\Xi_7} \operatorname{sinc}\left(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\omega}\right). \tag{4.6}$$

Recall that $\operatorname{sinc}(t) = \sin(t/2)/(t/2)$. In order to avoid any signal shifts, we use this centered version of the box spline during reconstruction.

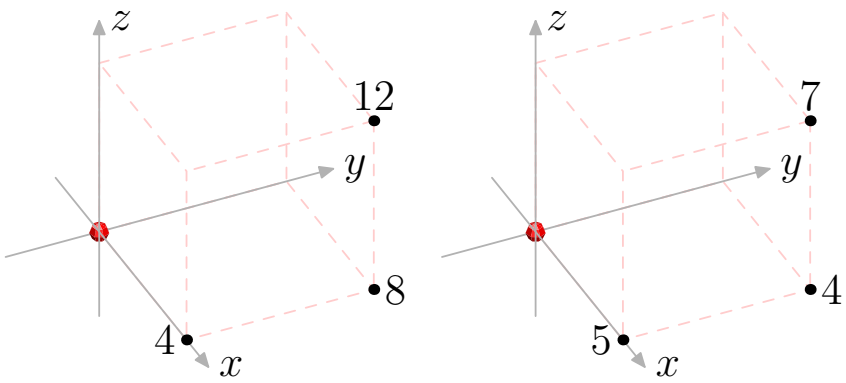
Using Equation (4.6), we can demonstrate the number of vanishing moments generated by the Cartesian shifts of this box spline and compare it with the tricubic B-spline. Recall that the Fourier transform of the tricubic B-spline is $\hat{\beta}_3(\omega_x,\omega_y,\omega_z) = \operatorname{sinc}^4(\omega_x)\operatorname{sinc}^4(\omega_y)\operatorname{sinc}^4(\omega_z)$. From careful examination of the Fourier transform of this box spline and the tricubic B-spline, we realize that they both attain a minimum of *four* vanishing moments at the center of every aliasing frequency; hence both of them have a fourth order approximation power.

Figure 4.8 demonstrates the vanishing moments in the space of the seven-direction box spline and the tricubic B-spline. From considering the aliasing frequencies along each coordinate axis at $(2\pi,0,0)$, $(0,2\pi,0)$ and $(0,0,2\pi)$ we note that the seven-direction box spline has *five* vanishing moments and the tricubic B-spline has *four*. For the aliasing frequencies on the sub-diagonals at $(2\pi,2\pi,0)$, $(2\pi,0,2\pi)$ and $(0,2\pi,2\pi)$ the seven-direction box spline has *four* vanishing moments while the tricubic B-spline has *eight* vanishing moments. Furthermore, on the diagonal direction $(2\pi,2\pi,2\pi)$ the seven-direction box spline introduces *seven* vanishing moments while the tricubic B-spline has *twelve*.

Oversmoothing and Post-Aliasing

Since we are considering the Cartesian shifts of the seven-direction box spline, the geometry of sampling is the same with a tricubic B-spline reconstruction. Therefore, for the reconstruction using this box spline, we can measure its over-smoothing and post-aliasing errors and compare with the corresponding errors for the tricubic B-spline.

The over-smoothing and post-aliasing errors were introduced by Marschner and Lobb in [53] for evaluating reconstruction schemes. The idea is to compare the Fourier transform



tricubic B-spline seven-direction box spline

Figure 4.8: A minimum of four vanishing moments at every aliasing frequency.

Table 4.1: Approximation to the mean squared differences of the Fourier transforms of seven-direction box spline and tricubic B-spline from the indicator function of the unit cube centered at the origin.

MS Error	Over-smoothing	Post-aliasing
Seven-direction box spline	7.203×10^{-1}	1.622×10^{-6}
Tricubic B-spline	6.007×10^{-1}	8.817×10^{-6}

of the reconstruction kernel against the ideal reconstruction whose Fourier transform is the indicator function of the unit cube centered at the origin in the frequency domain. Since we know the Fourier transform of the seven-direction box spline and the tricubic B-spline, we can evaluate them at any point in the frequency domain.

A numerical experiment to approximate the over-smoothing and post-aliasing errors of the seven-direction box spline and the tricubic B-spline was carried out by sampling the frequency domain inside the region of $[0, 6\pi]^3$ by a $200 \times 200 \times 200$ volume. The mean squared errors from the ideal reconstruction were estimated numerically and the results are gathered in Table 4.1. It is evident that the seven-direction box spline has lower post-aliasing errors at the cost of over-smoothing.

Variants

If we consider the decomposition of the direction vectors as above, we realize that reconstruction by Ξ_7 is a 3-D nearest neighbor $\boldsymbol{I}$ convolution followed by a convolution with the linear order Ξ_4. Reconstruction with the box spline of $\boldsymbol{I}$ is the nearest neighbor interpolation; this kernel provides a discontinuous reconstruction with first order approximation

power. The box spline of Ξ_4 by itself is a linear order reconstruction on the rhombic dodecahedral support of this box spline. When centered on a Cartesian lattice point, the support of this rhombic dodecahedron extends to the two neighbors on the left and the right on each axis. On the diagonal directions its support is only to the nearest neighbors, hence we concluded earlier that the support is contained in $[-2, 2)^3$. For this box spline $\rho = 2$; hence, it provides a C^0 reconstruction. Examining vanishing moments, one can verify that it is only of second order in terms of approximation power.

One can shrink the support of this box spline by considering the box spline induced by $\Xi'_4 = \frac{1}{2}\Xi_4$. The support of this box spline is a rhombic dodecahedron whose extent goes to the closest neighbors on each axis and half way to the nearest neighbor on the diagonal direction. Therefore, the support of this box spline is contained in $[-1, 1)^3$. Putting things back together, the box spline induced from $[\boldsymbol{I}, \Xi'_4]$ offers the same C^2 smoothness in reconstruction since the directions of the direction vectors of the box spline didn't change; hence, ρ is the same as for Ξ_7. The support of this box spline is contained in $[-3/2, 3/2)^3$ and examining its vanishing moments one can show that its approximation power is decreased to only *one*.

Another possibility is to use the box spline associated with trilinear interpolation instead of $\boldsymbol{I}$. The trilinear interpolant is induced from $[\boldsymbol{I}, \boldsymbol{I}]$. The box spline that is induced from $[\boldsymbol{I}, \boldsymbol{I}, \Xi'_4]$ is also of interest. The support of the trilinear box spline is $[-1, 1)^3$; hence, the support of this box spline is contained in $[-2, 2)^3$. For this box spline, $\rho = 6$. Therefore, it provides a C^4 reconstruction while its vanishing moments reveal that it is of *second* order approximation power. This box spline obviously favors the axis directions more strongly due to its trilinear component.

4.2.2 Reconstruction Results

The support of the seven-direction box spline, M_{Ξ_7}, is more isotropic than the support of a tensor-product reconstruction (i.e., a cube). Therefore, a reconstruction using this box spline outperforms the corresponding tricubic B-spline reconstruction when it comes to isotropy of reconstruction.

Our first case study is on reconstructing sharp planar surfaces with varying inclination. This experiment is documented in Figure 4.9. We designed a set of voxelized planar surface datasets that are essentially half spaces angled from zero to forty five degrees at increments of five degrees sampled at $21 \times 21 \times 21$ resolution on the Cartesian lattice. The zero degree

surface represents a plane that is aligned with a coordinate axis of the Cartesian lattice, while the forty five degrees planar surface makes a forty five degrees angle with the Z-axis in the XZ plane. While the box spline reconstruction performed equally well when compared with tricubic B-spline for planes between $0-30$ degrees, sharper angled planes showed a big difference in the two reconstructions. This is as expected, since for small angles the main features of the planar surface are indeed aligned with the coordinate axes.

The isotropy of the box spline reconstruction allowed the extremely smooth reconstruction of the planar surface, while the tricubic B-spline solution exhibited grid-aligned stripes on the surface known as stair-casing artifacts. The wider stripes are due to the aliasing *existing* in the volume dataset. The stair-casing artifacts appearing as the thin stripes which are dominant in the left and the middle column are due to aliasing in the reconstruction. In our experiments, none of the tensor-product reconstructions (even with wider support) in [67, 68] could decrease these stair-casing artifacts. We have also experimented with many different inclination angles, and the behavior of the seven-direction box spline reconstruction was consistent with what we reported above. This is due to the shape of the support and continuity of the box spline which is guaranteed to produce a C^2 reconstruction.

We have also experimented with the Marschner-Lobb dataset to compare the tricubic B-spline and the seven-direction box spline reconstructions. The seven-direction box spline reconstruction demonstrated its increased presence of over-smoothing error as predicted while the aliasing artifacts are better removed as in Figure 4.10.

The stronger post-aliasing cancellation in the seven-direction box splines is useful for removing grid-aligned artifacts as demonstrated in the planar surfaces dataset in Figure 4.9. This feature can also benefit the reconstruction of real datasets where the stair-casing is present as axis-aligned artifacts (see Figure 4.11). Similarly, the seven-direction box spline reconstruction is superior for the reconstruction of voxelized surfaces (Figure 4.12).

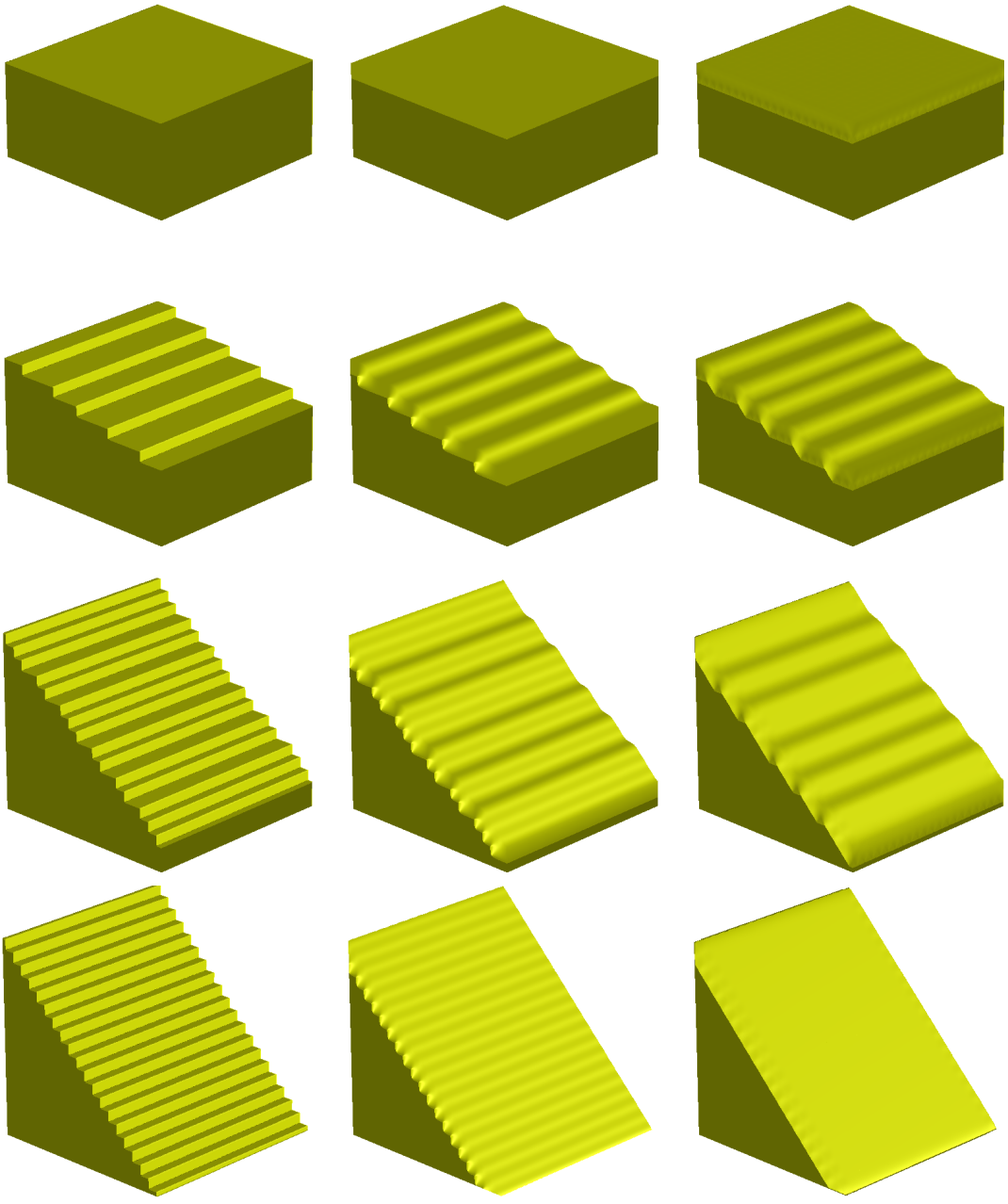


Figure 4.9: Planar surfaces with varying inclination sampled at the resolution of $21 \times 21 \times 21$. The first row shows the plane at 0 degrees, the second row is at 15 degrees, the third row is at 35 degrees and the last row at 45 degrees.

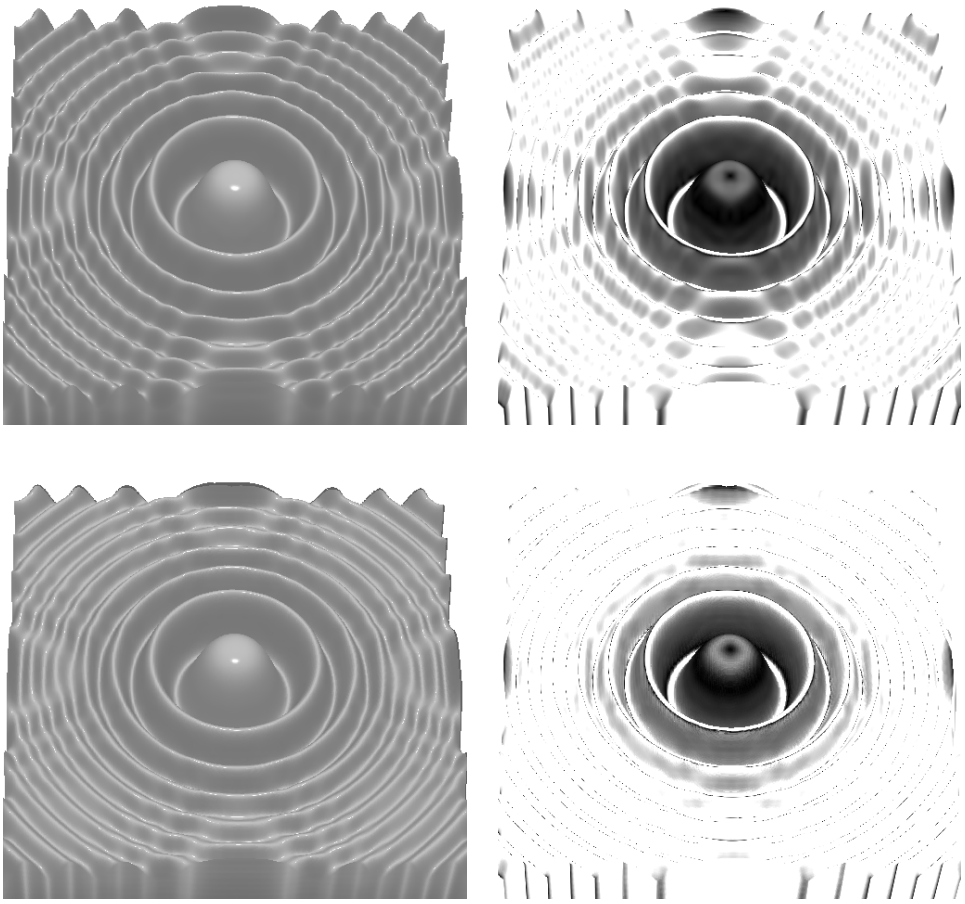


Figure 4.10: C^2 reconstructions of the ML data at a Cartesian resolution of $41 \times 41 \times 41 = 68.9K$. The first row shows the tricubic B-spline reconstruction. The second row shows the seven-direction box spline reconstruction. The max angular error of .3 radians is mapped to white.

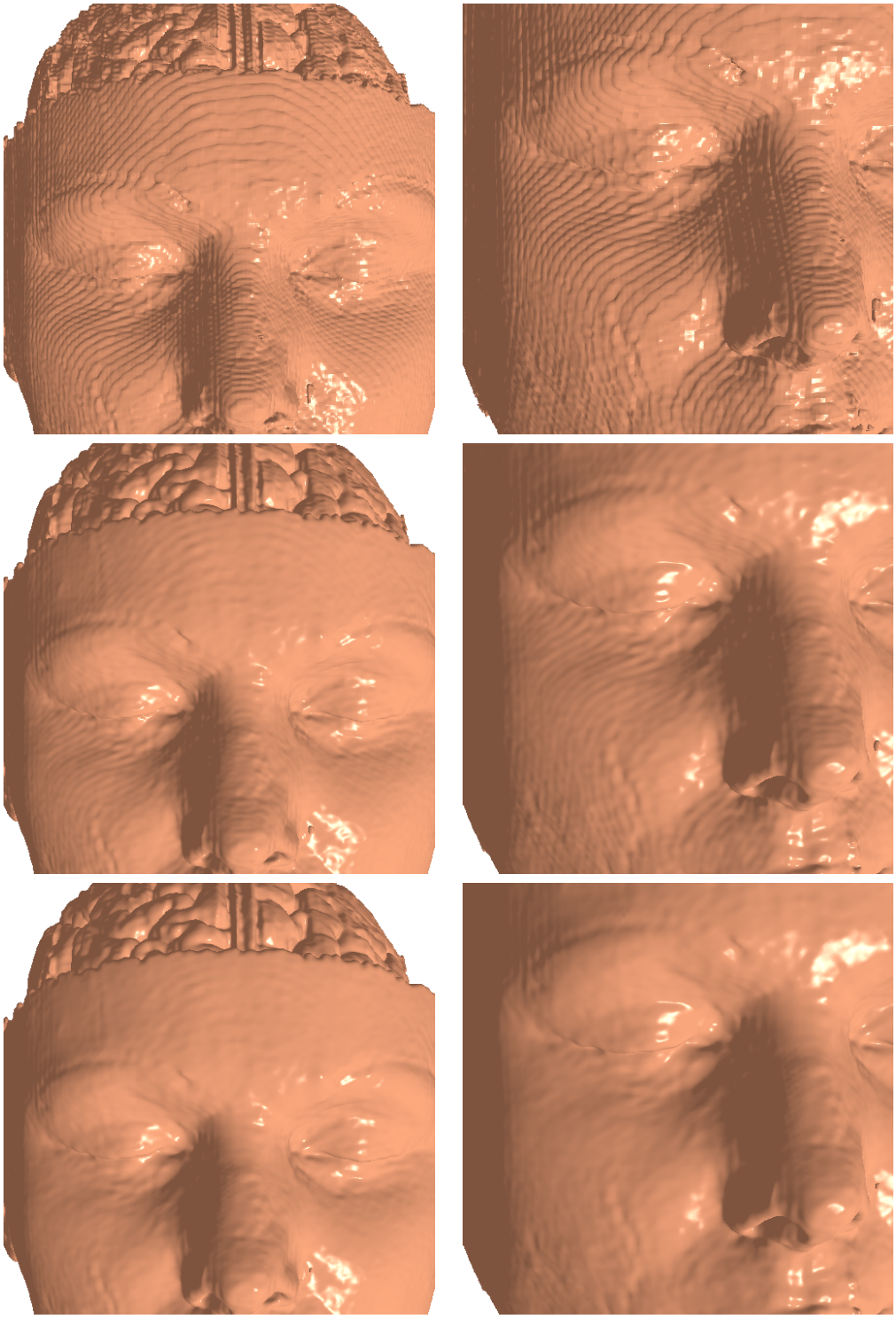


Figure 4.11: Reconstruction of the eye area in the brain dataset (the volumetric dataset is from University of North Carolina). The grid aligned artifacts are significantly reduced in the proposed reconstruction. Top row: trilinear B-spline, middle row: tricubic B-spline, and the bottom row: seven-direction box spline.

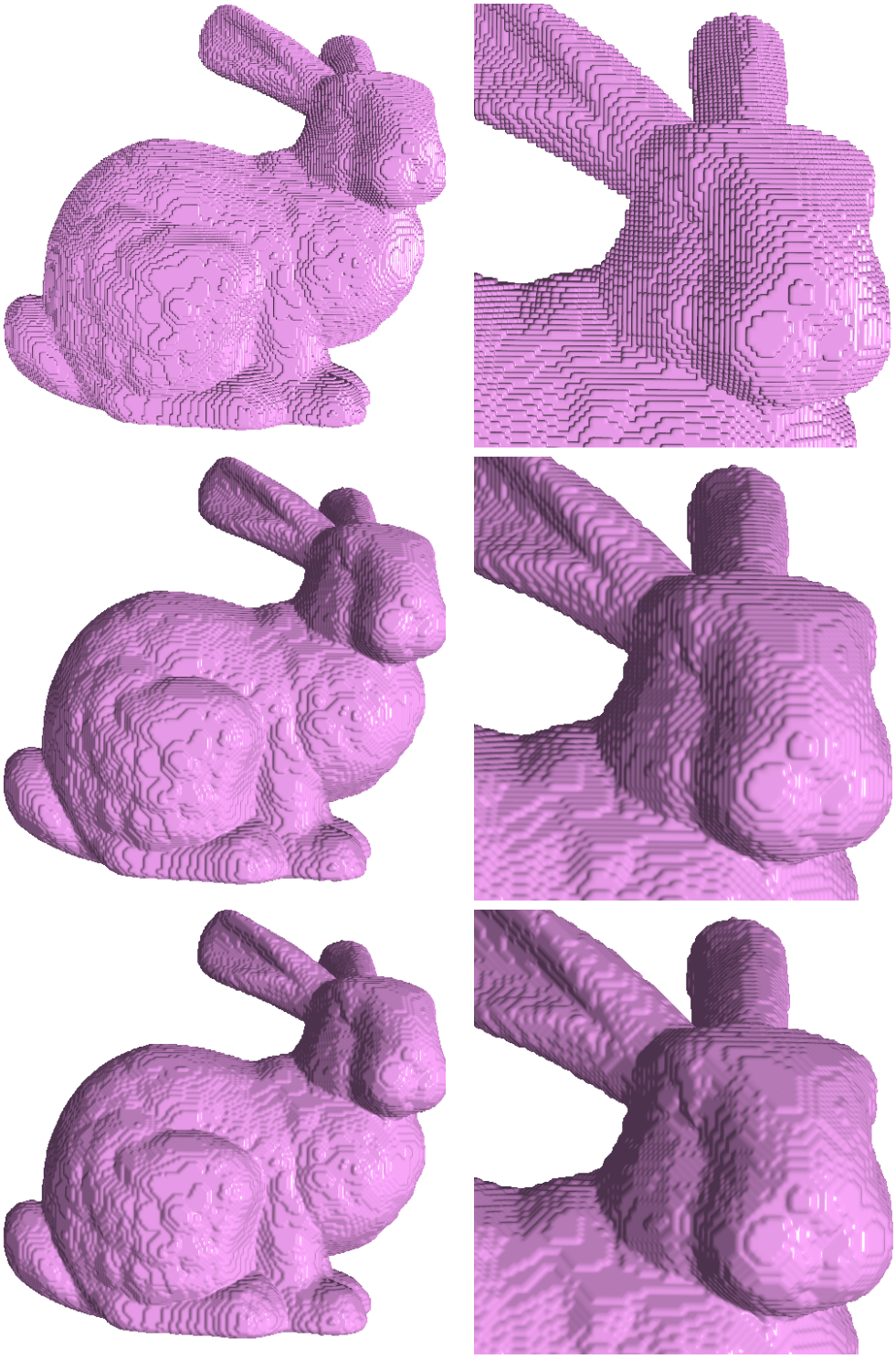


Figure 4.12: The Stanford Bunny dataset voxelized with an 18 connectivity neighborhood at the sampling resolution of $182 \times 182 \times 182$. Top row: trilinear B-spline, middle row: tricubic B-spline, and the bottom row: seven-direction box spline.

Chapter 5

Summary and Topics of Further Interest

In chapter 3, we introduced and discussed a class of four-direction box splines that allow for smooth and accurate reconstructions on the BCC lattice. These box spline-based reconstructions allowed us to show that the theoretical advantages of the BCC sampling lattices can be exploited in practical applications, particularly for volume visualization applications. For efficient evaluation of these box splines, we derived their explicit piecewise polynomial representations. Not only do these box splines allow for reconstructions with desired smoothness and numerical accuracy, also in terms of computational efficiency they outperform the corresponding tensor-product B-spline reconstructions on the Cartesian lattice.

In chapter 4, we presented a set of six-direction and nine-direction box splines suitable for reconstruction on the FCC lattice. These box splines allow for a smooth and accurate reconstruction on the FCC lattice. We also exploited the seven-direction box spline on the Cartesian lattice for its advantages for a more isotropic reconstruction than the tensor-product reconstructions. This box spline allows for significantly reducing the stair-casing artifacts in rendering applications.

The FCC and BCC lattices along with their box spline reconstructions demonstrate attractive properties for the sampling and reconstruction of trivariate functions. In the following sections, we consider other avenues of interest for exploiting the advantages of these optimal lattices in the computational domain.

Multi-resolution

Wavelet theory plays a central role in many biomedical, image processing and visualization applications [1]. Key algorithms such as compression [78] and de-noising [82] heavily rely on wavelet transforms. By sub-sampling a (sampled) function into coarser scales of resolution, the wavelet transform produces a multi-resolution representation of the function. The quincunx lattice in 2-D and the FCC and BCC lattices in 3-D produce striking sub-sampling patterns. Their curious structure allows for partitioning the spectrum of the function into a multi-resolution representation in a non-separable fashion.

Multi-resolution schemes are important tools for dealing with large data. Different levels of detail of the data can be pre-computed and an appropriate level can be picked according to the available bandwidth of the display device or the transition channel (e.g., for online music, video, or graphics applications).

Ideally, one would like to find the appropriate level-of-detail, which preserves exactly as much data as can be handled by the underlying hardware or software constraints. In other words, we would like to have a continuous level-of-detail (LOD) slider, creating the LOD that is needed. However, it is only feasible to pre-compute finitely many LOD's. The choice of LOD's is often constrained by the underlying data structures and algorithms available to process these data structures.

In image and volume processing, it has been customary to work with image and volume pyramids that treat each dimension separately which is a Cartesian tensor-product paradigm of multi-resolution. In such a scenario the smallest possible granularity of sub-sampling would be to halve the resolution of each dimension. This leads to the well-known quad-trees (in 2-D) and octrees (in 3-D). While we are only halving the resolution per coordinate axis during each iteration, we effectively reduce the overall data by a factor of four in 2-D and eight in 3-D.

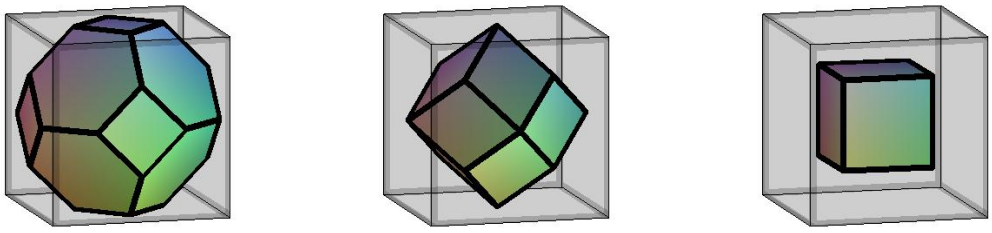
While this is a convenient and widely used pipeline, the granularity may be too coarse and alternatives are of interest. Hence, the quincunx structure has been studied in the image processing community [104, 28] with the benefit of allowing a finer granularity for 2-D image pyramids. Quincunx allows an overall data reduction by a factor of two in each iteration.

Van De Ville et al. [105] have shown that a simple extension of the quincunx scheme to 3-D is not possible, and iterating through the commonly assumed FCC subsampling cannot

provide a suitable isotropic representation of the signal at various resolution levels. We have also derived a family of BCC sub-sampling matrices suitable for dilation that allow for an isotropic reduction of resolution when iterating through the sub-sampling pipeline [32].

On the other hand, Linsen et al. [51] have studied a multi-resolution pipeline which has a change of resolution by a factor close to two in each step. In the first step of down-sampling, this method loses 3/8 of the data; in the second step it discards 3/5 of the data; and in the final step it loses 1/2 of the data to obtain a representation at 1/8 of the original data rate. Therefore, each resolution of the data is not exactly halving the information. Another drawback is that this pipeline is made up of grids for which no rendering algorithms with proper reconstruction kernels exist. Moreover, it is not possible to analyze the representation of the data at various resolution levels in terms of the spectrum of the underlying signal. In other words, there are no signal processing tools available to analyze or predict the quality of the data representation and reconstruction on such grids.

In [31] we introduced a multi-resolution pipeline in 3-D which represents a volumetric dataset at half the resolution on a FCC lattice and at one quarter the resolution on a BCC lattice. The resolution of one eighths is again represented on a Cartesian lattice. Since proper reconstruction schemes for these lattices exist we are able to render and visualize data in the decomposition domain on these 3-D lattices efficiently. The spectrum of the data at these sub-sampling steps has been illustrated in Figure 5.1.



(a) FCC (b) BCC (c) Cartesian

Figure 5.1: Brillouin zones for various subsampling steps. The gray cube indicates the support of the spectrum of the original Cartesian-sampled trivariate function.

Designing wavelets suitable for these FCC and BCC sub-sampling steps is a challenging task since they are non-separable (not a trivial tensor-product of one dimensional)

wavelets [83]. The non-separability leads to their improved isotropy and slow progression through the scales of resolution.

Non-separable multivariate regular perfect reconstruction filter banks have been studied by Cooklev [18] and Kovačević [42, 47, 48, 49] and Vetterli [106, 107, 108] that extend the techniques of filter design to two-channel multidimensional transforms. However, multi-channel non-separable wavelets are significantly harder to come by. A major difficulty is that the spectral factorization results do not easily extend to a multivariate setting [36]. Cohen and Daubechies showed that coming up with scaling functions and wavelets for arbitrary dilation matrices is difficult [14]. However, Cohen presented a class of wavelet bases with hexagonal symmetry [13].

Acquisition

We have demonstrated the attractive properties of the BCC and FCC lattices for representing trivariate functions. The application of such sampling lattices in scanning and acquisition devices is a challenging task with promising perspectives. Medical scanning modalities like CT, PET and MRI machines are to benefit from optimal sampling lattices.

One can exploit the BCC lattice to increase the quality of imaging (theoretically by 30%) in CT without increasing patient exposure time. One of the challenging tasks here is to incorporate a mathematical model (point-spread function) of the imaging system to design the pre-filters that compensate for the non-ideal physical characteristics of the imaging system. The framework of optimal quasi-interpolation [15, 16] can be used here to design fast and local pre-filters that are practical.

Optimal Sampling in Higher Dimensions

The advantage of the hexagonal sampling over the Cartesian sampling in 2-D is about 14%. In 3-D, the number goes up to about 30%; the efficiency of the best sampling pattern in 4-D ($\boldsymbol{D}_4$ lattice) is about 50% over the 4-D Cartesian lattice [17]. The optimal sampling lattices in higher dimensions demonstrate more attractive advantages, and hence studying the problem of sampling lattices and reconstruction in higher dimensions is of interest. The class of box splines (along with other polyhedral splines) allows for construction of suitable approximation kernels in arbitrary dimensions.

Piecewise Polynomial Representation of Box Splines

While the piecewise polynomial derivation of the box splines in Section 3.2 is for a specific class of box splines, we believe that it can be extended to general box splines. This piecewise polynomial representation can be further used for obtaining efficient computation of weighted sum of shifts of box splines using their barycentric forms. The box splines discussed in chapter 4 are of interest here. An explicit piecewise polynomial derivation of box splines would benefit reconstructions in chapter 4 so that we could implement them efficiently, similar to the implementation discussed in Section 3.2.

Appendix A

Geometric Facts

The cube, the rhombic dodecahedron and the truncated octahedron are the tiling polyhedra that are the Voronoi cells of the Cartesian, FCC and BCC lattices. For a sampling density of one sample per unit volume, the volume of the Brillouin zone of the sampling lattice is 1. In Table A.1, we have computed the radius of inscribed and circumscribed spheres to these polyhedra. According to these measurements, the Brillouin zone of the BCC lattice contains the largest inscribed sphere. The Brillouin zone of the FCC lattice has the smallest circumscribed sphere, while its inscribed sphere is not too much smaller than that of the BCC lattice.

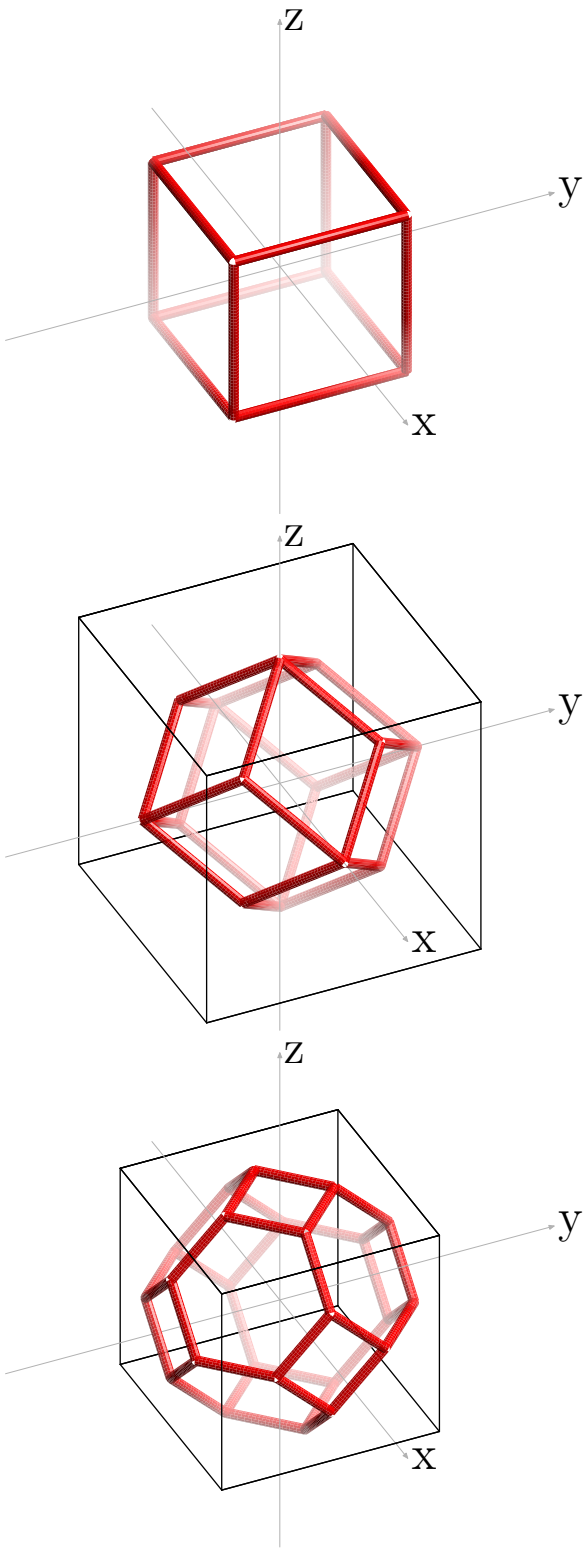


Figure A.1: Lattice Tiling Polyhedra: Cube (top), Rhombic Dodecahedron (middle) and Truncated Octahedron (bottom).

	Cube	Rhombic Dodecahedron	Truncated Octahedron
Volume	1	1	1
Bounding Cube	1	$\sqrt[3]{4} \approx 1.587$	$\sqrt[3]{2} \approx 1.260$
Insc Radius	$1/2 = .5$	$\frac{\sqrt[3]{4}}{2\sqrt{2}} \approx .561$	$\frac{\sqrt{3}}{4}\sqrt[3]{2} \approx .546$
Circ Radius	$\sqrt{3}/2 \approx .866$	$\sqrt[3]{4}/2 \approx .794$	$\frac{\sqrt{5}}{4}\sqrt[3]{2} \approx .704$
Circ / Insc	$\sqrt{3} \approx 1.732$	$\sqrt{2} \approx 1.414$	$\sqrt{\frac{5}{3}} \approx 1.291$
Vol - Vol Insc	.476	.260	.320
Vol Circ - Vol	1.721	1.094	.464

Table A.1: Geometric facts about the inscribed and circumscribed spheres to the tiling polyhedra of interest. Cube is the Voronoi cell of the Cartesian lattice. Rhombic Dodecahedron is the Voronoi cell of the FCC lattice, and Truncated Octahedron is the Voronoi cell of the BCC lattice. 'Insc' and 'Circ' stand for inscribed and circumscribed spheres respectively. 'Vol' stands for the volume of the solid.

Bibliography

[1] A. Aldroubi and M.A. Unser, editors. *Wavelets in Medicine and Biology*. CRC Press, Boca Raton FL, USA, 1996. 616 p.

[2] Akram Aldroubi and Karlheinz Gröchenig. Nonuniform sampling and reconstruction in shift-invariant spaces. *SIAM Rev.*, 43(4):585–620 (electronic), 2001.

[3] G. Birkhoff and H. L. Garabedian. Smooth Surface Interpolation. *J. Math. and Phys.*, 39:258–268, 1960.

[4] Garrett Birkhoff. Piecewise bicubic interpolation and approximation in polygons. In *Approximations with Special Emphasis on Spline Functions (Proc. Sympos. Univ. of Wisconsin, Madison, Wis., 1969)*, pages 185–221. Academic Press, New York, 1969.

[5] Garrett Birkhoff and Carl R. de Boor. Piecewise polynomial interpolation and approximation. In *Approximation of Functions (Proc. Sympos. General Motors Res. Lab., 1964)*, pages 164–190. Elsevier Publ. Co., Amsterdam, 1965.

[6] Ronald N. Bracewell. *The Fourier transform and its applications*. McGraw-Hill Series in Electrical Engineering. Circuits and Systems. McGraw-Hill Book Co., New York, third edition, 1986.

[7] Pierre Brémaud. *Mathematical principles of signal processing*. Springer-Verlag, New York, 2002. Fourier and wavelet analysis.

[8] Gerald Burns and A. M. Glazer. *Space groups for solid state scientists*. Academic Press Inc., Boston, MA, second edition, 1990.

[9] Ingrid Carlbom. Optimal filter design for volume reconstruction and visualization. In *Proceedings of the IEEE Conference on Visualization 1993*, pages 54–61, October 1993.

[10] Hamish Carr, Torsten Möller, and Jack Snoeyink. Artifacts Caused by Simplicial Subdivision. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):231–242, 2006.

[11] J. W. S. Cassels. *An introduction to the geometry of numbers*. Springer-Verlag, Berlin, 1971. Second printing, corrected, Die Grundlehren der mathematischen Wissenschaften, Band 99.

[12] Charles K. Chui and Ming-Jun Lai. Algorithms for generating B-nets and graphically displaying spline surfaces on three- and four-directional meshes. *Comput. Aided Geom. Des.*, 8(6):479–493, 1991.

[13] A. Cohen and J.-M. Schlenker. Compactly supported bidimensional wavelet bases with hexagonal symmetry. *Constr. Approx.*, 9(2-3):209–236, 1993.

[14] Albert Cohen and Ingrid Daubechies. Nonseparable bidimensional wavelet bases. *Rev. Mat. Iberoamericana*, 9(1):51–137, 1993.

[15] L. Condat, T. Blu, and M. Unser. Beyond interpolation: Optimal reconstruction by quasi-interpolation. In *Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP'05)*, volume I, pages 33–36, Genova, Italy, September 11-14, 2005.

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

[17] J.H Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 3rd edition, 1999.

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

[19] A M Cormack. Sampling the radon transform with beams of finite width. *Physics in Medicine and Biology*, 23(6):1141–1148, 1978.

[20] H. S. M. Coxeter. *Regular polytopes*. Dover Publications Inc., New York, third edition, 1973.

[21] Balázs 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.

[22] Morten Dæhlen. On the evaluation of box splines. *Mathematical methods in computer aided geometric design*, pages 167–179, 1989.

[23] Ingrid Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

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

[25] Carl de Boor. On the Evaluation of Box Splines. *Numerical Algorithms*, 5(1–4):5–23, 1993.

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

[27] L. Desbat. Efficient sampling on coarse grids in tomography. *Inverse Problems*, 9:251–269, 1993.

[28] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice-Hall, Inc., Englewood-Cliffs, NJ, 1st edition, 1984.

[29] James C. Ehrhardt. MR Data Acquisition and Reconstruction Using Efficient Sampling Schemes. *IEEE Transactions on Medical Imaging*, 9(3):305–309, September 1990.

[30] Alireza Entezari, Ramsay Dyer, and Torsten Möller. Linear and cubic box splines for the body centered cubic lattice. In *Proceedings of the IEEE Conference on Visualization*, pages 11–18, October 2004.

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

[32] Alireza Entezari, Torsten Möller, and Jacques Vaisey. Subsampling matrices for wavelet decompositions on body centered cubic lattices. *IEEE Signal Processing Letters*, 11(9):733–735, September 2004.

[33] Adel Faridani. Results, old and new, in computed tomography. In *Inverse problems in wave propagation (Minneapolis, MN, 1995)*, volume 90 of *IMA Vol. Math. Appl.*, pages 167–193. Springer, New York, 1997.

[34] Adel Faridani. Sampling theory and parallel-beam tomography. In *Sampling, wavelets, and tomography*, Appl. Numer. Harmon. Anal., pages 225–254. Birkhäuser Boston, Boston, MA, 2004.

[35] Adel Faridani. Fan-beam tomography and sampling theory. In *The Radon transform, inverse problems, and tomography*, volume 63 of *Proc. Sympos. Appl. Math.*, pages 43–66. Amer. Math. Soc., Providence, RI, 2006.

[36] Karlheinz Gröchenig and Amos Ron. Tight compactly supported wavelet frames of arbitrarily high regularity. *Proc. Amer. Math. Soc.*, (126):1101–1107, 1998.

[37] Markus Hadwiger, Thomas Theußl, Helwig Hauser, and Eduard Gröller. Hardware-Accelerated High-Quality Filtering on PC Hardware. In *Workshop on Vision, Modeling and Visualization*, pages 105–112, 2001.

[38] T.C. Hales. Cannonballs and honeycombs. *Notices of the AMS*, 47(4):440–449, April 2000.

[39] G.H. Hardy. Notes of special systems of orthogonal functions IV: The orthogonal functions of Whittakers series. *Proc. Camb. Phil. Soc.*, 37:331–348, 1941.

[40] George Hart. Zonohedrification, `http://www.georgehart.com/zonohedra/zonohedrification.html`. [Online; accessed 14-Nov-2006].

[41] R Hauschild, B Hosticka, and S Müller. A CMOS Optical Sensor System Performing Image Sampling on a Hexagonal Grid. In *Proc of European Solid-State Circuits Conference*, 1996.

[42] J. Kovačević and Martin Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathscr{R}^n$. *IEEE Transactions on Information Theory*, 17(10):939–954, Mar. 1992.

[43] F. Jacobs, S. Matej, and R Lewitt. Image reconstruction techniques for PET. University of Pennsylvania, Department of Radiology, Medical Image Processing Group (MIPG), MIPG Technical Report: MIPG245. University of Ghent, Department of Electronics and Information Systems (ELIS), Medical Imaging and Signal Processing Research Group, ELIS Technical Report: R9810.

[44] Pierre-Marc Jodoin and Victor Ostromoukhov. Halftoning over a hexagonal grid. In *Proc of SPIE*, volume 5008, pages 443–454, 2003.

[45] Leif Kobbelt. Stable Evaluation of Box Splines. *Numerical Algorithms*, 14(4):377–382, 1997.

[46] Helga Kolb, Eduardo Fernandez, and Ralph Nelson. The organization of retina and visual system, `http://webvision.med.utah.edu`. [Online; accessed 10-Feb-2007].

[47] J. Kovačević and M. Vetterli. Perfect reconstruction filter banks with rational sampling rate changes. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pages 1785–1788, Toronto, Canada, May 1991.

[48] 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.

[49] J. Kovačević, M. Vetterli, and G. Karlsson. Design of multidimensional filter banks for nonseparable sampling. In *Proc. IEEE Int. Symp. Circ. and Syst.*, pages 2004–2008, New Orleans, LA, May 1990.

[50] D. L. Lau and R. Ulichney. Blue-noise halftoning for hexagonal grids. *IEEE Transactions on Image Processing*, 15(5):1270–1284, May 2006.

[51] Lars Linsen, Valerio Pascucci, Mark A. Duchaineau, Bernd Hamann, and Kenneth I. Joy. Wavelet-based multiresolution with nth-root-of-2 subdivision. *Journal on Computing*, special edition, 2004.

[52] A. K. Louis and F. Natterer. Mathematical problems in computerized tomography. In *Proceedings IEEE*, volume 71, pages 379–389, 1983.

[53] Steve R. Marschner and Richard J. Lobb. An evaluation of reconstruction filters for volume rendering. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the IEEE Conference on Visualization 1994*, pages 100–107, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press.

[54] S. Matej and R. Lewitt. Efficient 3D grids for image reconstruction using spherically-symmetric volume elements. *IEEE Trans on Nucl Sci*, 42:1361–1370, 1995.

[55] S. Matej and R. Lewitt. Practical considerations for 3-D image reconstruction using spherically symmetric volume elements. *IEEE Trans on Med Imaging*, 15:68–78, 1996.

[56] 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.

[57] Michael McCool. *Analytic Signal Processing for Computer Graphics using Multivariate Polyhedral Splines*. PhD thesis, University of Toronto, 1995.

[58] Michael D. McCool. Analytic Antialiasing With Prism Splines. *SIGGRAPH Conference Proceedings*, pages 429–436, 1995.

[59] Michael D. McCool. Accelerated Evaluation of Box Splines via a Parallel Inverse FFT. *Computer Graphics Forum*, 15(1):35–45, 1996.

[60] B. H. McCormick. The Illinois pattern recognition computer ILLIAC III. *IEEE Trans. Computer*, 12:791–813, December 1963.

[61] E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002.

[62] Tai Meng, Benjamin Smith, Alireza Entezari, Arthur E. Kirkpatrick, Daniel Weiskopf, Leila Kalantari, and Torsten Möller. On visual quality of optimal 3D sampling and reconstruction. In *Graphics Interface 2007*, May 2007.

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

[64] Lee Middleton and Jayanthi Sivaswamy. *Hexagonal Image Processing: A Practical Approach*. Springer, 2005.

[65] Erik G. Miller. Alternative tilings for improved surface area estimates by local counting algorithms. *Computer Vision and Image Understanding: CVIU*, 74(3):193–211, 1999.

[66] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer graphics. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22, pages 221–228, August 1988.

[67] Torsten Möller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. A comparison of normal estimation schemes. In *Proceedings of the IEEE Conference on Visualization 1997*, pages 19–26, October 1997.

[68] Torsten Möller, Klaus Mueller, Yair Kurzion, Raghu Machiraju, and Roni Yagel. Design of Accurate and Smooth Filters for Function and Derivative Reconstruction. *Proceedings of the 1998 Symposium on Volume Visualization*, pages 143–151, October 1998.

[69] Klaus Mueller, Torsten Möller, J. Edward Swan II, Roger Crawfis, Naeem Shareef, and Roni Yagel. Splatting Errors and Antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):178–191, 1998.

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

[71] J. Peters. C^2 Surfaces Built from Zero Sets of the 7-direction Box Spline. In *The Mathematics of Surfaces VI*, pages 463–474, 1996.

[72] 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, December 1962.

[73] M. J. D. Powell and M. A. Sabin. Piecewise Quadratic Approximations on Triangles. *ACM Trans. Math. Softw.*, 3(4):316–325, 1977.

[74] David Pratt. Patterns in nature, `http://ourworld.compuserve.com/homepages/dp5/pattern1.htm`. [Online; accessed 10-Feb-2007].

[75] W. Qiao, D.S. 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, October 2005.

[76] P. Rattey and A. Lindgren. Sampling the 2-D Radon transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(5):994–1002, 1981.

[77] R.G.Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, December 1981.

[78] Jon Rogers and Pamela Cosman. Wavelet zerotree image compression with packetization. *IEEE Signal Processing Letters*, 17(10):939–954, Mar. 2002.

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

[80] M. Senechal. *Quasicrystals and Geometry*. Cambridge University Press, 1995.

[81] C.E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948.

[82] Eero P. Simoncelli. Noise Reduction Via Bayesian Wavelet Coring. *IEEE Transactions on Information Theory*, 17(10):939–954, Mar. 1992.

[83] E.P. Simoncelli and E.H. Adelson. Nonseparable extensions of quadrature mirror filters to multiple dimensions. *Proceedings of IEEE, Special Issue on Multi-dimensional Signal Processing*, 78(4):652–664, April 1990.

[84] R. Staunton. The processing of hexagonally sampled images. *Advances in imaging and electron physics*, 119:191–265, 2001.

[85] R. Staunton and N. Storey. A comparison between square and hexagonal sampling methods for pipeline image processing. In *Proc. SPIE*, volume 1994, pages 142–151, 1989.

[86] R. Stevenson and G. Arce. Binary display of hexagonally sampled continuous-tone images. *J. Opt. Soc. Am. A*, 2(7):1009–1013, 1985.

[87] Robin Strand. Surface skeletons in grids with non-cubic voxels. *ICPR*, 01:548–551, 2004.

[88] Robin Strand. Using the Hexagonal Grid for Three-Dimensional Images: Direct Fourier Method Reconstruction and Weighted Distance Transform. *ICPR*, 2:1169–1172, 2006.

[89] Robin Strand and Gunilla Borgefors. Distance transforms for three-dimensional grids with non-cubic voxels. *Computer Vision and Image Understanding*, 100(3):294–311, 2005.

[90] Gilbert Strang and George J. Fix. A Fourier analysis of the finite element variational method. *Construct. Aspects of Funct. Anal.*, pages 796–830, 1971.

[91] Haiwei Sun, Ning Kang, Jun Zhang, and Eric S. Carlson. A fourth-order compact difference scheme on face centered cubic grids with multigrid method for solving 2d convection diffusion equation. *Math. Comput. Simul.*, 63(6):651–661, 2003.

[92] G. Suparta. *Focussing Computed Tomography Scanner*. PhD thesis, Monash University, 1999.

[93] George G. Szpiro. *Kepler's Conjecture: How Some of the Greatest Minds in History Helped Solve One of the Oldest Math Problems in the World*. John Wiley, 2003.

[94] 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.

[95] Thomas Theußl, Torsten Möller, and Eduard Gröller. Optimal regular volume sampling. In *Proceedings of the IEEE Conference on Visualization 2001*, pages 91–98, Oct. 2001.

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

[97] D'Arcy Wentworth Thompson. *On Growth and Form*. Dover, 1992.

[98] Zhi Tian and Harry L. van Trees. DOA Estimation with Hexagonal Arrays. In *Proc of Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages IV:2053–2056, 1998.

[99] R. Ulichney. Dithering with blue noise. *Proceedings of IEEE*, 76(1):56–79, Jan 1988.

[100] Robert Ulichney. *Digital halftoning*. MIT Press, Cambridge, MA, USA, 1987.

[101] M. Unser, C.Ó. Sánchez Sorzano, P. Thévenaz, S. Jonić, C. El-Bez, S. De Carlo, J.F. Conway, and B.L. Trus. Spectral signal-to-noise ratio and resolution assessment of 3d reconstructions. *Journal of Structural Biology*, 149(3):243–255, March 2005.

[102] Michael Unser. Vanishing moments and the approximation power of wavelet expansions. In *Proceedings of the 1996 IEEE International Conference on Image Processing (ICIP'96)*, volume I, pages 629–632, Lausanne VD, Switzerland, September 16-19, 1996.

[103] Michael Unser. Sampling—50 Years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, April 2000.

[104] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Signal Processing. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1993.

[105] 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.

[106] M. Vetterli, J. Kovačević, and D. J. LeGall. Perfect reconstruction filter banks for HDTV representation and coding. *Image Communication, special issue on HDTV*, 2(3):349–364, October 1990. Invited paper.

[107] M. Vetterli, J. Kovačević, and D. J. LeGall. *Signal Processing of HDTV, II*, chapter Perfect Reconstruction Filter Banks for HDTV Representation and Coding. Elsevier, 1990.

[108] Martin Vetterli and Jelena Kovačević. *Wavelets and subband coding*. Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1995.

[109] Van De Ville, D., Blu, T., M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle. Hex-splines: A novel spline family for hexagonal lattices. *IEEE Transactions on Image Processing*, 13(6):758–772, June 2004.

[110] P. Wells, R. Smith, and G. B. Suparta. Sampling the sinogram in computed tomography. *A Back to Basics article in Materials Evaluation*, 55(7):772–776, July 1997.

[111] J. M. Whittaker. *Interpolatory Function Theory*. Cambridge University Press, London, 1935.

[112] Wikipedia. Zonohedron, `http://en.wikipedia.org/wiki/Zonohedron`, 2006. [Online; accessed 14-Nov-2006].

[113] Wikipedia. Cubic crystal system, `http://en.wikipedia.org/wiki/Cubic_crystal_system`, 2007. [Online; accessed 17-Apr-2007].

[114] Wikipedia. Honeycomb, `http://en.wikipedia.org/wiki/Honeycomb`, 2007. [Online; accessed 09-Feb-2007].

[115] Wikipedia. Minkowski addition, `http://en.wikipedia.org/wiki/Minkowski_addition`, 2007. [Online; accessed 19-Jun-2007].

[116] Wikipedia. Pomegranate seeds, `http://en.wikipedia.org/wiki/Image:Pomseeds2.jpg`, 2007. [Online; accessed 17-Apr-2007].

[117] Philip B. Zwart. Multivariate Splines with Nondegenerate Partitions. *SIAM Journal on Numerical Analysis*, 10(4):665–673, September 1973.