

Artifacts Caused By Simplicial Subdivision

Hamish Carr, Torsten Möller, and Jack Snoeyink

Abstract—We review schemes for dividing cubic cells into simplices (tetrahedra) for interpolating from sampled data to \mathbb{R}^3 , present visual and geometric artifacts generated in isosurfaces and volume renderings, and discuss how these artifacts relate to the filter kernels corresponding to the subdivision schemes.

Index Terms—G.1.1 Interpolation, G.1.2.a Approximation of surfaces and contours, I.4 Image Representation, I.4.10.e Volumetric Representation, I.6.9.g Visualization techniques and methodologies, I.6.9.h Volume visualization

I. INTRODUCTION

SCIENTIFIC applications often generate data on regular rectilinear grids in three dimensions. The sampled values are extended to the entire space by some type of interpolation, and the interpolated function f visualized, often with isosurfaces [1] or volume rendering [2]. Isosurfaces of f are geometric surfaces of the form $f^{-1}(h)$ for an *isovalue* h , while volume rendering maps f to opacity and emissive values then computes the amount of light that reaches an image plane.

Interpolation often uses a trilinear interpolant, which is relatively complex and costly [3], so many researchers use tetrahedral approximations. Each cubic cell is divided into several tetrahedra, and barycentric interpolation applied to each tetrahedron. For volume rendering, Projected Tetrahedra [2] requires input in the form of a simplicial mesh, often achieved by simplicial subdivision of cubic cells.

Tetrahedra have several advantages. They can be generated in advance or at run-time. Tetrahedral isosurfaces have 3 cases, compared with 15 for Marching Cubes [1], or 38 for trilinear interpolation [3]. Barycentric interpolation on tetrahedra is monotonic: critical points are at vertices of the mesh, making topological analysis easier. And some techniques require tetrahedra [2], [4], [5], [6], [7].

These advantages do not come without a price. Subdivision increases mesh size by a factor of 5 or more, and polygonal output by a factor of 2 or more, compared with Marching Cubes [1]. Additionally, the barycentric interpolation does not match the assumed trilinear interpolation function, causing unpleasant geometric and visual artifacts when rendering.

In this paper, we illustrate isosurface and volume rendering artifacts which should be considered when deciding when or how to apply simplicial subdivision for particular tasks.

We review previous work in Sec. II, present desiderata for simplicial subdivisions in Sec. III, and describe standard 3-D subdivisions in Sec. IV. We then consider the numerical accuracy of the subdivisions in Sec. V, show characteristic geometric, topological and visual artifacts for isosurfaces in

Sec. VI and for Projected Tetrahedra [2] in Sec. VII, and discuss the implications in terms of sampling theory in Sec. VIII. Finally, we present our conclusions in Sec. IX.

II. PREVIOUS WORK

Previous analysis of simplicial subdivision of cubic cells [8], [9], [10] focusses on the number of triangles generated [10], and on the topological consistency [8], [10] and correctness [10] of the isosurfaces.

The simplicial subdivision most often used is the minimal subdivision (Sec. IV-B) of 5 tetrahedra per cube [2], [5], [8], [9], [10], [11], [12]. Also reported [8], [10], [11], [13], [14], [15] is the Freudenthal subdivision (Sec. IV-C) of 6 tetrahedra. We have also experimented with the body-centred cubic lattice (Sec. IV-E) of 12 tetrahedra. Bloomenthal [9] use a different subdivision (Sec. IV-D) of 12 tetrahedra. Albertelli & Crawfis [8] extend this to 14, 16, 18, 20, 22 & 24 tetrahedra. Based on the desiderata reported here, we prefer the 24-fold subdivision (Sec. IV-F), as do others [8], [13], [16], [17]. A 48-fold subdivision (Sec. IV-G) is also reported in [15], [18].

III. DESIDERATA FOR SIMPLICIAL SUBDIVISION

There are several competing goals when subdividing cubic cells which mean that no subdivision is best under all circumstances, so we simply state these goals as desiderata.

We assume that the data has been sampled on a cubic grid in three dimensions, although our analysis also applies to curvilinear and irregular grids. We assume that the desired interpolation function $f(p)$ is the trilinear interpolant over the cube unless stated otherwise. We subdivide each cube, replacing the grid with a tetrahedral mesh and perform barycentric interpolation over each tetrahedron, substituting a new piecewise-linear interpolation $F(p)$ for $f(p)$.

We assess different subdivision schemes according to how well each achieves the following desiderata:

- i) Parsimonious: the subdivision should add as few data points as possible.
- ii) Contained: $F(p)$ should depend only on sample values that $f(p)$ depends on: for trilinear f , only on the sample values at the vertices of the cube containing p .
- iii) Symmetric: the subdivision should be symmetric under rotations and reflections of the cube.
- iv) Minimal: the subdivision should use as few simplices as possible, to reduce the cost of subsequent processing.
- v) Implicit: the subdivision should not require explicit creation and storage of tetrahedral elements.
- vi) Continuous: F should be continuous, to prevent cracks or holes in generated isosurfaces.
- vii) Exact: the subdivision should minimize numerical, geometric, topological, visual and sampling differences between F and f .

Hamish Carr was at the University of British Columbia, and is now with University College Dublin.

Torsten Möller is at Simon Fraser University.

Jack Snoeyink is at the University of North Carolina at Chapel Hill.

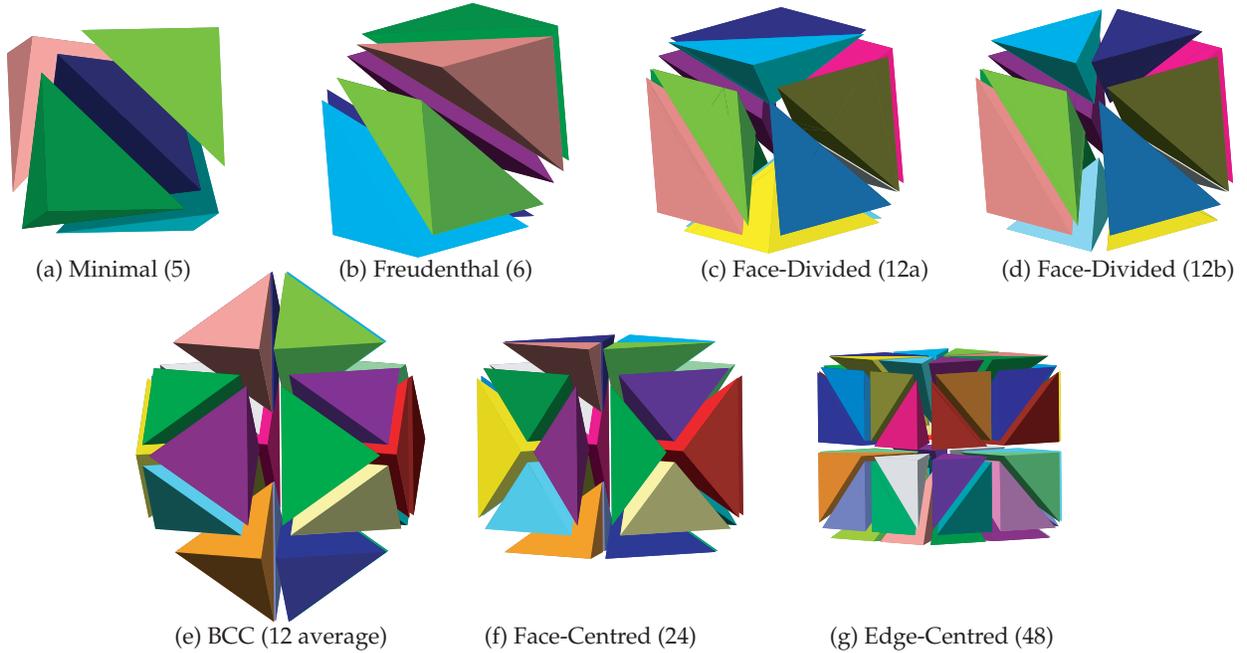


Fig. 1. The Subdivisions

Subdivision	Contained	Symmetric	Parsimonious	Minimal	Implicit	Continuous	Exact
Minimal (5)	yes	no	yes	yes	yes	possible	no
Freudenthal (6)	yes	no	yes	no	yes	possible	no
Face-Divided (12)	yes	no	no	no	possible	possible	no
BCC (12 average)	no	yes	no	no	yes	yes	no
Face-Centred (24)	yes	yes	no	no	yes	yes	no
Edge-Divided (48)	yes	yes	no	no	yes	yes	no

TABLE I

TABLE OF SUBDIVISIONS. “POSSIBLE” IMPLIES A DESIDERATUM THAT CAN BE SATISFIED BY A SUITABLE CHOICE OF ADDED VERTICES OR EDGES.

IV. 3-D SUBDIVISIONS

Having stated desiderata, we now describe each of the subdivisions in Fig. 1, give a taxonomy and test each one against the desiderata, summarizing our results in Table I.

A. Taxonomy of Subdivisions

Tetrahedral subdivisions are constructed using only the vertices of the mesh, or by adding additional vertices. Additional vertices can belong to an edge, a face or the body of the cell. In practice, body vertices are added first, followed by face and edge vertices, since this disturbs symmetry the least and minimizes the resulting number of simplices. This results in the following taxonomy:

- 1) Subdivisions with no additional vertices:
 - a) Minimal (5-fold) - Sec. IV-B
 - b) Freudenthal (6-fold) - Sec. IV-C
- 2) Subdivisions with body vertex added:
 - a) Face-Divided (12-fold) - Sec. IV-D
 - b) Body Centered Cubic (BCC) - Sec. IV-E
- 3) Subdivisions with body and face vertices added:
 - a) Face-Centred (24-fold) - Sec. IV-F
 - b) 14-, 16-, 18-, 20- and 22-fold: see Sec. IV-F
- 4) Subdivisions with body, face and edge vertices added:
 - a) Edge-Divided (48-fold) - Sec. IV-G

B. The 5-fold Minimal Subdivision

The most popular simplicial subdivision [2], [5], [8], [9], [10], [11], [12] is the minimal subdivision shown in Fig. 1(a). To obtain it, four vertices of the same parity are cut from the cube along the plane defined by their neighbouring vertices. Each cut reduces the number of vertices remaining by 1. After four such cuts, a regular tetrahedron is left over.

Minimality is assured because each tetrahedron shares at least one triangular face with another tetrahedron. Thus, each cut removes at most one vertex from the cube and 4 cuts are required to reduce the original 8 vertices to 4 vertices. When the first vertex is removed, its neighbours become degree four and cannot themselves be “cut off”. As a result, all vertices removed must have the same parity, and only two ways exist to divide the cube into five tetrahedra.

This subdivision is contained, parsimonious, minimal, and implicit, but not symmetric, because the cut corners and their neighbours are treated differently, and because the central tetrahedron is a different shape from the others.

For continuity, note that opposite faces are divided by opposing diagonals. As shown in [2], [10], [11], [16], the two possible 5-fold subdivisions must alternate if F is to be continuous. This is possible in regular meshes but not for curvilinear meshes such as a torus of odd circumference.

C. The Freudenthal (6-fold) Subdivision

This subdivision is older than the minimal subdivision, and is used for numerical computation [13], [14], [15] and rendering [8], [10], [11]. The vertices on a major diagonal of the cube are each connected to all vertices of the cube. This gives 6 tetrahedra packed around the major diagonal as shown in Fig. 1(b). Three of these tetrahedra are mirror images of the other three, and are not isomorphic without reflection.

This subdivision is contained, parsimonious, implicit, and only slightly less minimal than the minimal 5-fold subdivision. Provided that face diagonals are consistent between adjacent cubes, this subdivision will be continuous. This is achieved most simply by using the same major diagonal in each cube, or by alternating diagonals in adjacent cubes, as in Fig. 1(g). This choice of one of four major diagonals prevents symmetry and imposes a strong directional bias on this subdivision.

Other subdivisions into 6 tetrahedra exist [8], but these are even less symmetric, use non-isomorphic tetrahedra, and do not guarantee continuity. Since these subdivisions are rarely used, we will not consider them further.

D. Face-Divided 12-fold Subdivision

This subdivision is constructed from pyramids whose bases are the faces of the cube [8], [9]. Each face center is joined to the body center, giving 6 square pyramids. Each of these is divided into two tetrahedra by an arbitrary face diagonal (see Figs. 1(c) and 1(d)). Not all vertices are treated equally, so this subdivision is not symmetrical. Moreover, there are two possible diagonals per face, giving 64 possible configurations.

In practice, continuity is assured by using the same subdivision in each cube, in which case opposing faces must have matching diagonals, and only two unique cases remain. In the first case (Fig. 1(c)), a main diagonal is picked and its vertices connected to the body center and to diagonally adjacent vertices. It is easy to see that this is merely a further subdivision of the Freudenthal subdivision.

It is also possible to have two vertices on a main diagonal which belong to no face diagonals (Fig. 1(d)). Because no such vertices exist in the Freudenthal subdivision, this version of the face-divided subdivision cannot be obtained by further dividing the Freudenthal subdivision.

Neither of these two cases is symmetric, parsimonious, minimal, or exact, and care is required to ensure that the subdivision is implicit and continuous.

E. The Body Centered Cubic Subdivision

This subdivision, suggested to us by Herbert Edelsbrunner, is based on a body-centered cubic (BCC) lattice. A vertex is added to the center of each cell, connected to the vertices of the cell and to the center of each adjacent cell. This generates 24 tetrahedra each shared between two cells. On average, 12 tetrahedra per cube are generated.

We show this subdivision in Fig. 1(e): unlike other subdivisions, it is not possible to show a single cube divided into tetrahedra. Instead, we show the 24 tetrahedra that intersect a given cube. Although the BCC subdivision is strongly symmetric, implicit and continuous, it is not parsimonious (it adds extra vertices), minimal, contained, or exact.

(i)	(x_i, y_i, z_i)	(σ_i)
1	(3, 3, 3)	0.6
2	(4, 4, 3)	0.6
3	(5, 3, 3)	0.6
4	(6, 4, 3)	0.6
5	(7, 3, 3)	0.6
6	(8, 4, 3)	0.6
7	(9, 3, 3)	0.6
8	(8, 8, 5)	2.4
9	(9.5, 3, 9)	1.5

$$f(x, y, z) = \sum_{i=1}^9 e^{-\frac{d_i^2}{2\sigma_i^2}}$$

$$d_i^2 = (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2$$

Fig. 2. A Small Analytical Dataset Defined As A Sum Of Gaussians

F. Face-Centered 24-fold Subdivision

The face-divided subdivision (see Fig. 1(f)) can be divided by adding some or all of the face centers [8], [13], [16], [17], [18], giving subdivisions with 14, 16, 18, 20, 22 or 24 tetrahedra, respectively. The 24 tetrahedra subdivision can also be obtained by dividing each of the 24 simplices in the BCC subdivision (Fig. 1(e)) along the plane of the cube's face.

This subdivision is contained, symmetric, implicit, and continuous. It is not parsimonious, because it requires an average of 4 interpolated data points per voxel. Nor is it minimal, because it has nearly five times as many simplices as the minimal subdivision.

We had previously claimed [19] that this subdivision could be made exact by selecting the face and body saddles as the additional points that define the tetrahedra. This claim was based on work by Natarajan [20], which assumed a maximum of one body saddle. As Nielson [3] shows, however, two body saddles are possible in a single cube. Thus, the 24-fold subdivision cannot be made topologically correct, as it has at most one body point added.

G. Edge-Divided 48-fold Subdivision

The 24-fold subdivision can be further subdivided by adding edge vertices to obtain a 48-fold subdivision [15], [18], as in Fig. 1(g), or by dividing the original cube into 8 sub-cubes, then applying Freudenthal subdivision to the sub-cubes, choosing the main diagonals that coincide with the main diagonals of the original cube. Since barycentric interpolation is linear, it is not difficult to prove that this subdivision generates the same F as the Face-Centred (24-fold) subdivision.

This subdivision is contained, symmetric, implicit, and continuous, but neither parsimonious nor minimal, as it requires 8 times the original number of vertices, and nearly 10 times the minimal number of simplices.

V. NUMERICAL RESULTS

To compare the different subdivision schemes quantitatively, we performed several computational experiments on the small test function shown in Fig. 2. This function is a sum of Gaussian distributions, with most of the peaks aligned in a zigzag along the sampling grid. Although at or beyond the Nyquist limit for resolution, this dataset neatly illustrates the worst-case behaviour of simplicial subdivisions.

The atom9 dataset fits into a cube of size $13 \times 13 \times 13$. We took samples at spacings of $1 = 2^0$ down to 2^{-5} , then evaluated the maximum absolute error and maximum relative

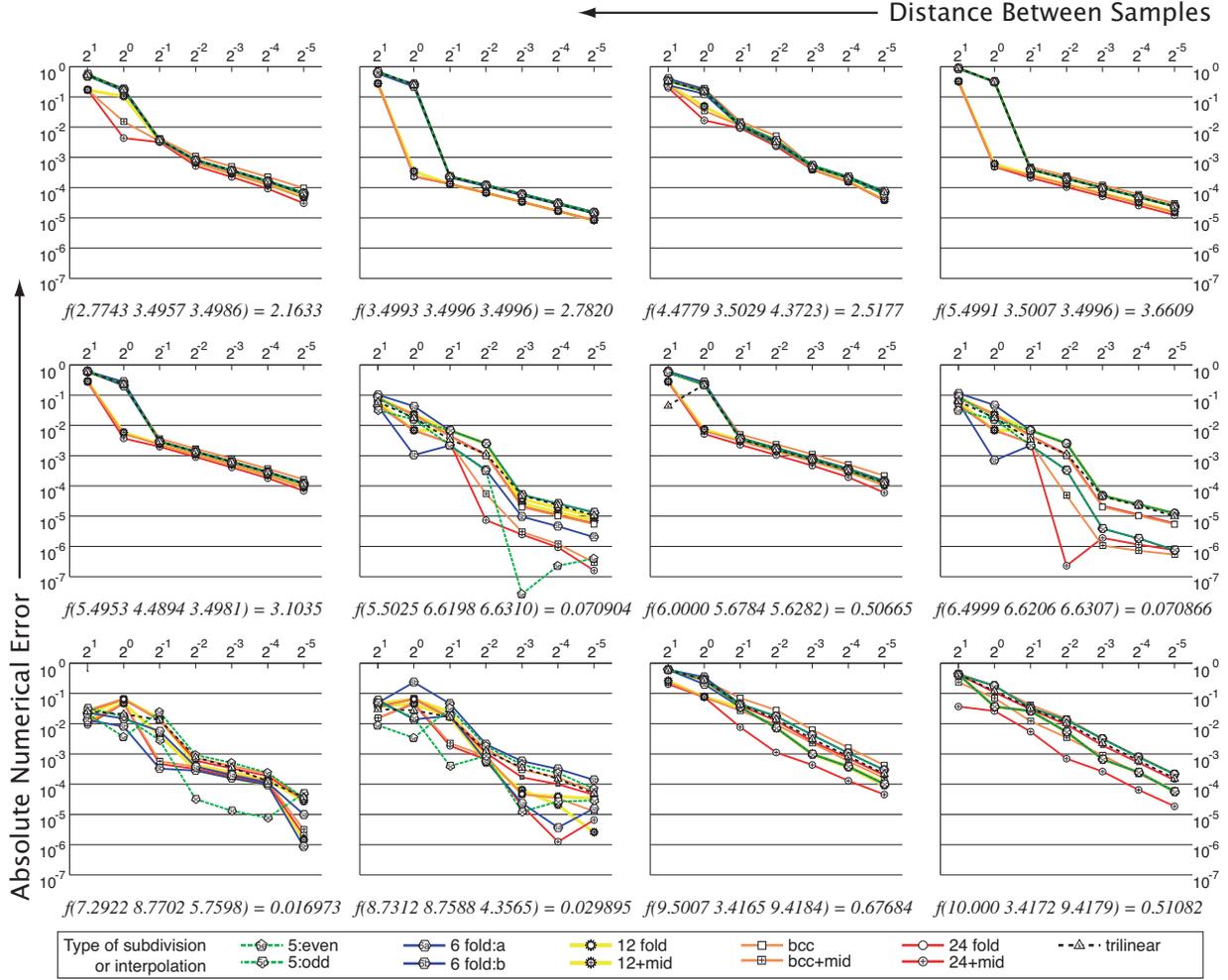


Fig. 3. Numerical Errors From Simplicial Subdivision. These graphs show the numerical error between the analytical form of the atom9 dataset and the interpolated form induced by the various subdivisions for selected cubic cells in the data. Subdivisions with vertices added were tested twice: once by interpolation from the previously sampled values, once (marked mid) by sampling the analytical form at the added vertices.

error by comparing the analytically correct function with the interpolated function induced by each subdivision using MATLAB’s constrained minimization function `fmincon`.

We did this for each type of cubic subdivision, plus trilinear interpolation. For the 5-fold and 6-fold, which are non-symmetric, we used subdivisions in each of two orientations. For the 12-fold, BCC, and 24-fold, which need additional data points and are thus not parsimonious, we tested subdivisions that interpolated this new data from the lattice vertices, and those that obtained the true data by function evaluation.

After studying these results, we chose to focus on absolute error, since large absolute errors produced more noticeable artifacts. We therefore selected a number of cubes with absolute errors greater than 10^{-2} , and ran experiments to compute maxima for different subdivisions and different scales.

Finally, from these experiments, we chose 40 points that had realized the maximum errors in their respective cubes. We tracked these points using different subdivisions and different lattice cube sizes. We plotted the absolute error of the results on logarithmic scales, using logarithms base 2.

We show some sample graphs in Fig. 3: we note that these are not exhaustive, but are best viewed as representative of the

numerical behaviour of the induced interpolant. In these there are some common features:

- 1) A smaller lattice reduced the error roughly in proportion to the length of the cube side. (Unfortunately, it blows up the size proportionally to $1/\text{side}^3$.)
- 2) For test points near a maximum, we observed a large increase in accuracy with the first steps of refinement, with accuracy tapering off with further refinement.
- 3) The maximum error for trilinear interpolation is similar to the simpler, simplex-based interpolation schemes.
- 4) As can be seen from the clumping of the error plots, the numerical error under the various subdivisions does not vary significantly. Moreover, it was not uncommon for the asymmetric subdivisions (5-fold, and 6-fold) to have both the best and the worst errors. It follows that geometric (numerical) error is a poor test of the topological artifacts which we will see in Sec. VI and Sec. VII. This is particularly worrisome, as many hierarchical decomposition schemes, such as [18] rely on measuring numerical (geometric) error to determine which simplices to remove.
- 5) Using the real midpoints sampled from the function

generally reduces the error of the 12-fold, 24-fold, and BCC subdivisions. The reduction is comparable to that achieved by halving the cube size, when the midpoints are interpolated.

VI. ISOSURFACE ARTIFACTS

In addition to numerical tests, we illustrate characteristic visual artifacts from simplicial subdivisions using atom9 test dataset and the UNC head. Fig. 4 shows isosurface artifacts for the various subdivisions compared to a ray-tracing of the ideal isosurface (Fig. 4(a)), a ray-tracing of an isosurface using the trilinear interpolant (Fig. 4(b)), and the surfaces generated by Marching Cubes (Fig. 4(n)).

As predicted, if we use the minimal subdivision without a parity rule, we get visible cracks in the surfaces in Fig. 4(c). But if we apply a parity rule, we get either Fig. 4(d) or Fig. 4(e). Note how different the topologies of the surfaces in these two images are. This means that the topology of the surfaces generated is effectively dependent on a coin-toss.

If we move to the Freudenthal subdivisions, Fig. 4(f) and Fig. 4(g) show that the topology is dependent on the choice of major diagonal and that pronounced directional biases are visible, aligned with the major diagonal chosen.

For all its symmetry and theoretical advantages, the body-centred cubic subdivision in Fig. 4(j) produces even worse artifacts, seen in close-up in Fig. 4(k). We call these artifacts “girders”: spurious connections between surfaces that form through the faces of cubes. To explain how this happens, we show in Fig. 5(b) a similar 2-D simplicial subdivision of a square cell. This subdivision is obtained by adding a vertex at the center of each cell. The central vertex is then connected to all four vertices of that square cell, and all 4 neighbouring central vertices. The edges between vertices of each square cell are suppressed. As with the 3-D BCC subdivision, girders form when we use this subdivision to interpolate function values.

To show how the girders are formed, consider the zero-valued vertices in the second and third rows of Fig. 5(a). Intuitively, we expect these vertices to be connected: that is, we expect that no contour exists that separates them. The bilinear interpolation function satisfies this intuitive expectation: a sample contour at the isovalue 0.20 is shown.

For the pseudo-BCC subdivision, the contour at isovalue 0.20 is shown in Fig. 5(c). Note that the lower row of cell centers all have the value 0.25, so they separate the zeros in the second and third rows of vertices: a sample contour at 0.20 is again shown. This contour “encloses” the edge between the cell centers. It is not difficult to see that the two parallel girders shown in Fig. 4(k) are caused by the contour enclosing the edge between cell centers in the same way.

This girder effect is not restricted to interpolation by simplicial subdivision, it has also been observed in data sampled on a BCC grid [21], and is a major disadvantage of using barycentric interpolation over BCC grids.

Since the face-divided subdivision schemes depend on a major diagonal of the cube, we expect to see directional biases similar to those for the Freudenthal subdivision. These are apparent in Fig. 4(h) and Fig. 4(i). Although the surfaces generated are more rounded, they are not much smoother.

For the 24- and 48- fold subdivisions in Fig. 4(l) and Fig. 4(m), we see identical isosurfaces, as predicted. Although not entirely smooth, these surfaces do not have visible directional biases. For applications where directional biases should be avoided, the 24- fold subdivision is clearly to be preferred.

Since the features in this dataset are at or beyond the Nyquist limit, we might expect the artifacts to be purely local in character and of no particular significance in larger data sets. Unfortunately, this is not true: these artifacts manifest themselves on the larger scale as *texture*: that is to say, a consistent pattern across a large area of the surface.

To illustrate this, we show some isosurfaces from the UNC head data set in Fig. 6 in medium zoom, where the artifacts are most apparent. Using the minimal subdivision without a parity rule results in small cracks below the eye and along the bridge of the nose in Fig. 6(c). As before, the parity rules in Fig. 6(d) and Fig. 6(e) prevent cracks, but give different textures along the bridge of the nose and different topologies in the eye socket. With the Freudenthal subdivision, we see strong directional biases in Fig. 6(f) and Fig. 6(g). As expected, the face-divided subdivisions show diagonal bias in Fig. 6(h) and Fig. 6(i), although less strongly than the Freudenthal subdivision. For this particular image, the BCC, 24- and 48-fold subdivisions all perform acceptably.

These artifacts emphasize high frequencies inherent to empirical data, and it is worth asking whether they also occur in simulation data. Since this data is often smoother, we expect to see fewer artifacts, and Fig. 7 demonstrates that this can be the case. In this case, we rendered isosurfaces for the fuel dataset: since the dataset is highly symmetric, we would expect the impact of asymmetric subdivisions to be reduced. Yet, as we see from this figure, the directional biases result once more in oriented topological connections. Even though the isosurfaces are relatively smooth, oriented differences are still visible, particularly for the Freudenthal subdivision in Fig. 7(e) and Fig. 7(f). And, although girders are not apparent for the BCC subdivision in Fig. 7(i), a related artifact appears in the form of pyramidal pits whose boundaries are defined by the girders: these pits can also appear in the form of pyramidal peaks.

These images support the observation that cheap subdivisions come at the expense of topological and geometric artifacts that are not well-predicted numerically, while higher-quality subdivisions require increasingly large numbers of tetrahedra and therefore of rendered triangles.

In Table II, we consider the output (rendering) cost in terms of numbers of triangles generated for a sample isosurface. Note that even the minimal scheme generates more than twice as many triangles as Marching Cubes. And using the face-centred subdivision to avoid visual artifacts comes at the cost of seven times as many triangles.

Of course, some of this cost can be reduced by using a modified version of Marching Cubes that corresponds to the topology of the particular subdivision: the set of cases for the Freudenthal subdivision, for example, can be found in [21]. However, for subdivisions with additional vertices, such as the face-centred subdivision, these cases become quite complex.

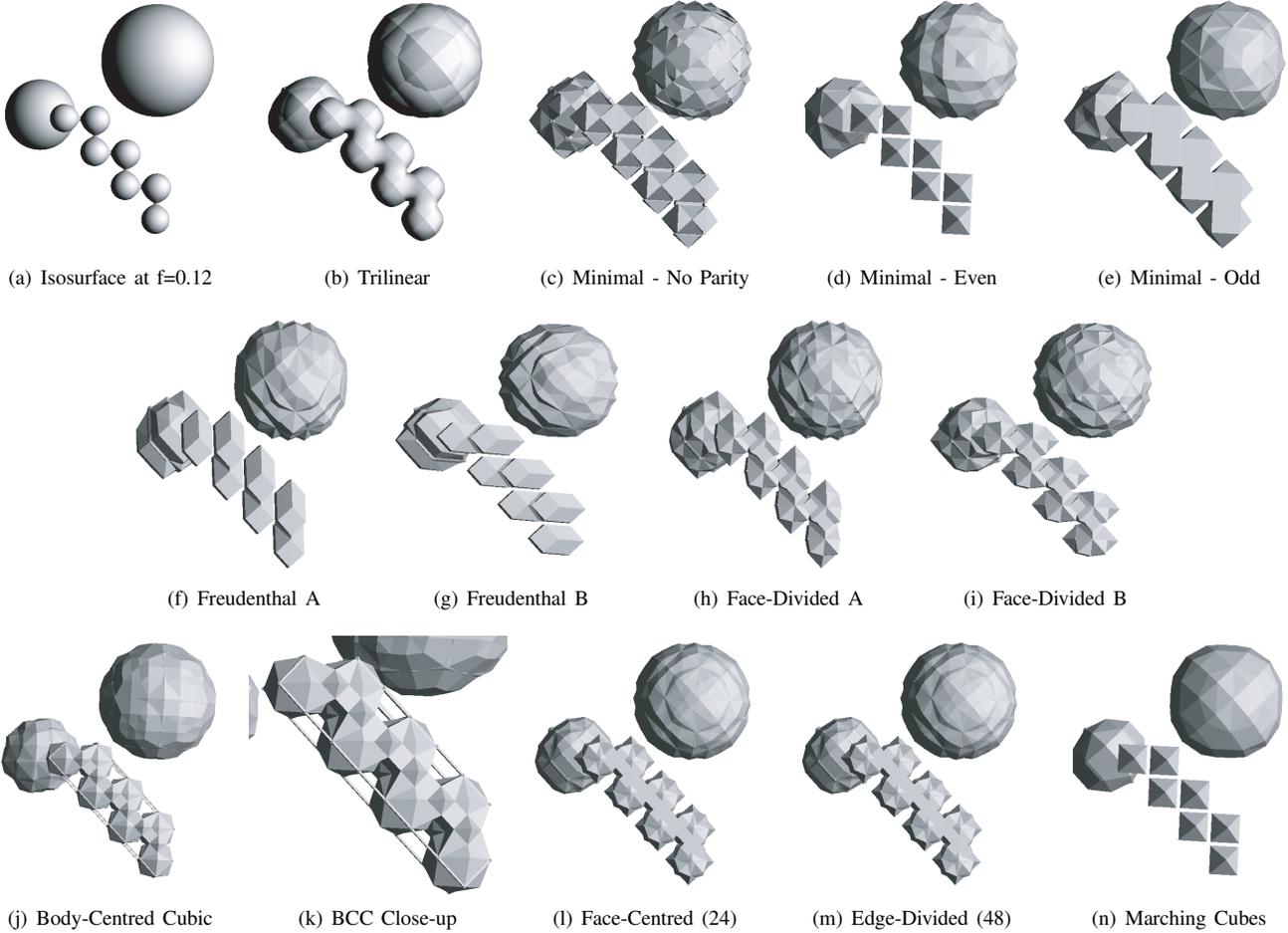


Fig. 4. Artifacts In The Atom9 Dataset At $f = 0.12$

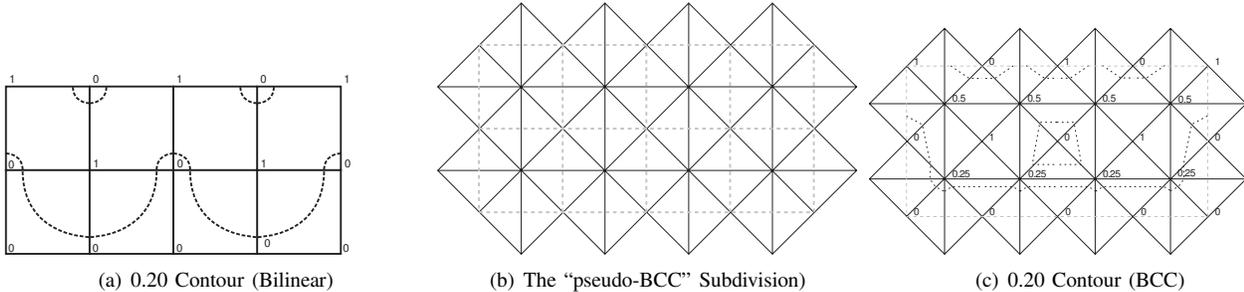


Fig. 5. “Pseudo-BCC” Subdivision In 2-D, Illustrating “Girder” Formation

VII. DIRECT VOLUME RENDERING ARTIFACTS

Visual artifacts are also visible in volume renderings generated using Projected Tetrahedra [2]. Fig. 8 shows images of the atom9 dataset using a simple step transfer function. We used 1-D textures for accurate calculation of the exponential term in the rendering integral [6], but did not optimize for speed [22], [23], [24] because we were solely concerned with image quality. We also did not apply perspective correction [25], because we used orthogonal projection. Since we were working with regular cubic meshes, we did not explicitly construct the tetrahedral mesh, instead rendering the cells in back-to-front order, one cube at a time. Since the number of tetrahedra used

in each subdivision is a constant multiple of the number of cubes, we did not measure the memory footprint.

It is not a surprise that the artifacts from Fig. 4 are still visible, complete with parity differences in the minimal subdivision (Fig. 8(b) and Fig. 8(c)). Moreover, the discontinuity of the function when using 5 simplices without a parity rule causes the unpleasant artifacts visible in Fig. 8(a). The expected axial biases can be seen in Fig. 8(d) through Fig. 8(g), while girders are visible for the BCC subdivision in Fig. 8(h). For clarity, this last image is also shown in Fig. 8(i) with an increased extinction coefficient to make the girders more apparent. Lastly, both the 24 and 48 simplex versions in Fig. 8(j) and Fig. 8(k) give reasonable results.

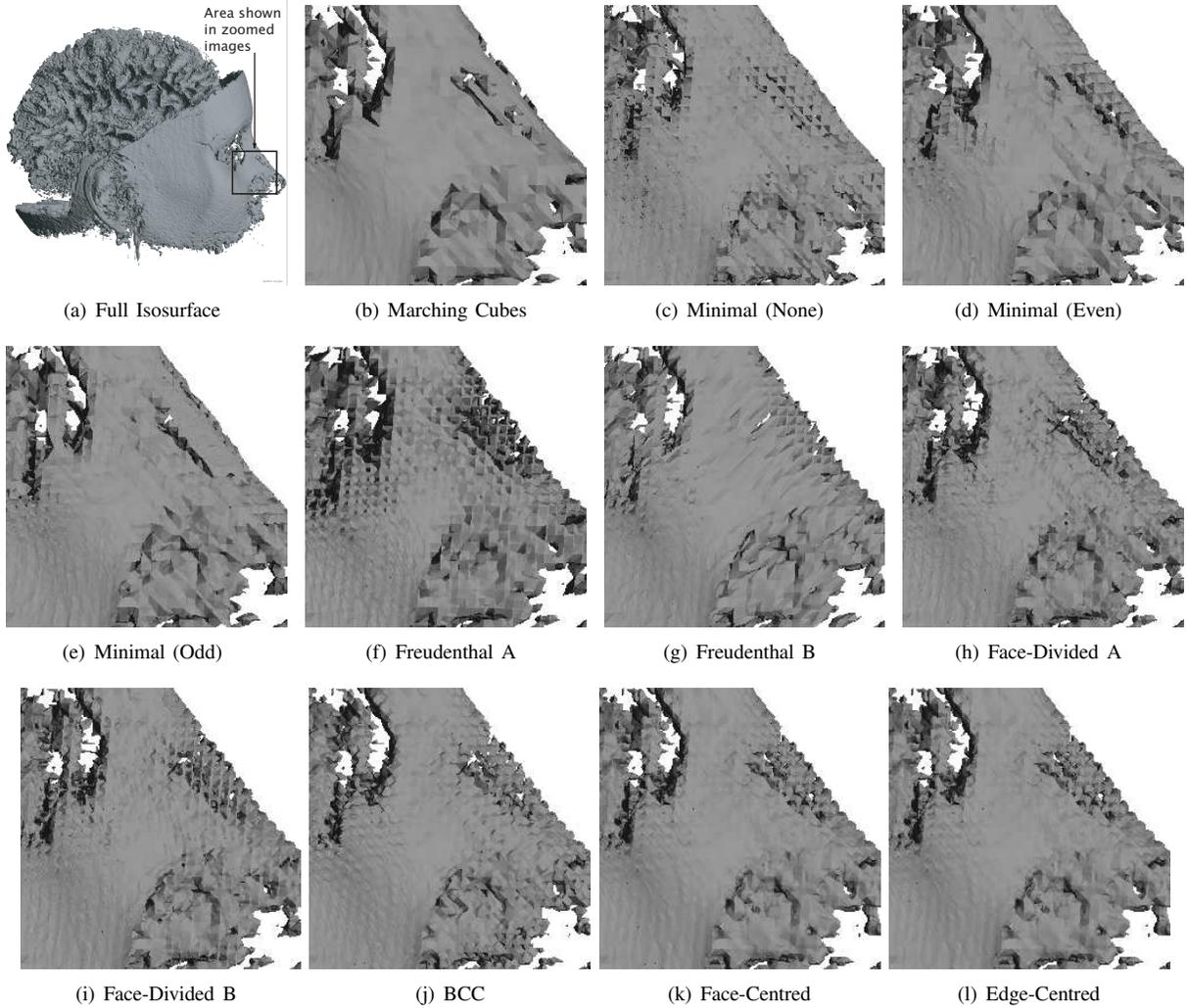


Fig. 6. Closeup of Nose of UNC Head. Visual artifacts of simplicial subdivision are visible here in the form of surface texture.

Scheme	Triangles	Ratio
Marching Cubes	1,029,936	1.0
Minimal (5), No Parity	2,452,378	2.381
Minimal (5), Even Parity	2,453,046	2.382
Minimal (5), Odd Parity	2,452,370	2.381
Freudenthal (6), Axis 000 - 111	3,011,206	2.924
Freudenthal (6), Axis 001 - 110	3,003,346	2,916
Face-Divided (12) A	4,830,508	4.690
Face-Divided (12) B	4,832,860	4.692
Body Centred Cubic (BCC) (12)	5,980,476	5.807
Face-Centred (24)	7,250,428	7.040
Edge-Divided (48)	11,855,532	11.511

TABLE II

NUMBER OF TRIANGLES GENERATED FOR ISOSURFACES OF THE UNC HEAD USING EACH SUBDIVISION.

For larger datasets, volume-rendered artifacts are less prominent than for isosurfaces. In Fig. 9, we show volume-rendered images of the fuel data set. The no-parity minimal subdivision shows diagonal discontinuities in Fig. 9(a). These discontinuities do not appear in the parity-based minimal subdivisions, but the small peak at lower left has a protusion that points to

the upper left in Fig. 9(b) but to the lower right in Fig. 9(c). In Fig. 9(d) and Fig. 9(e), the expected diagonal bias is seen in the boundary of the central (larger) peak, in a faintly visible diagonal fuzziness, and in diagonal protrusions in the smaller peaks. Fig. 9(f) and Fig. 9(g) also show boundary problems, diagonal smearing and apparent directional biases in the smaller peaks. While the BCC subdivision shows no immediate artifacts in Fig. 9(h), girder artifacts show up as a pale grid elsewhere in the data with the same transfer function (Fig. 9(i)). And finally, as expected, face-centred and edge-centred subdivisions show few and manageable artifacts.

Interestingly, the isosurface artifacts are worse for the empirical UNC head data set, but the volume rendering artifacts are worse for the simulated fuel data set. For example, the artifacts other than topological variations in Fig. 7 are less obvious than those in Fig. 6. And the artifacts in Fig. 9 are more obvious than those visible in a volume rendering of the UNC head dataset (omitted due to space constraints). Since empirical data tends to have a lot of high-frequency information and noise, techniques such as isosurfaces which emphasize high-frequency information can reasonably be expected to show worse high-frequency artifacts. However, the noise applies

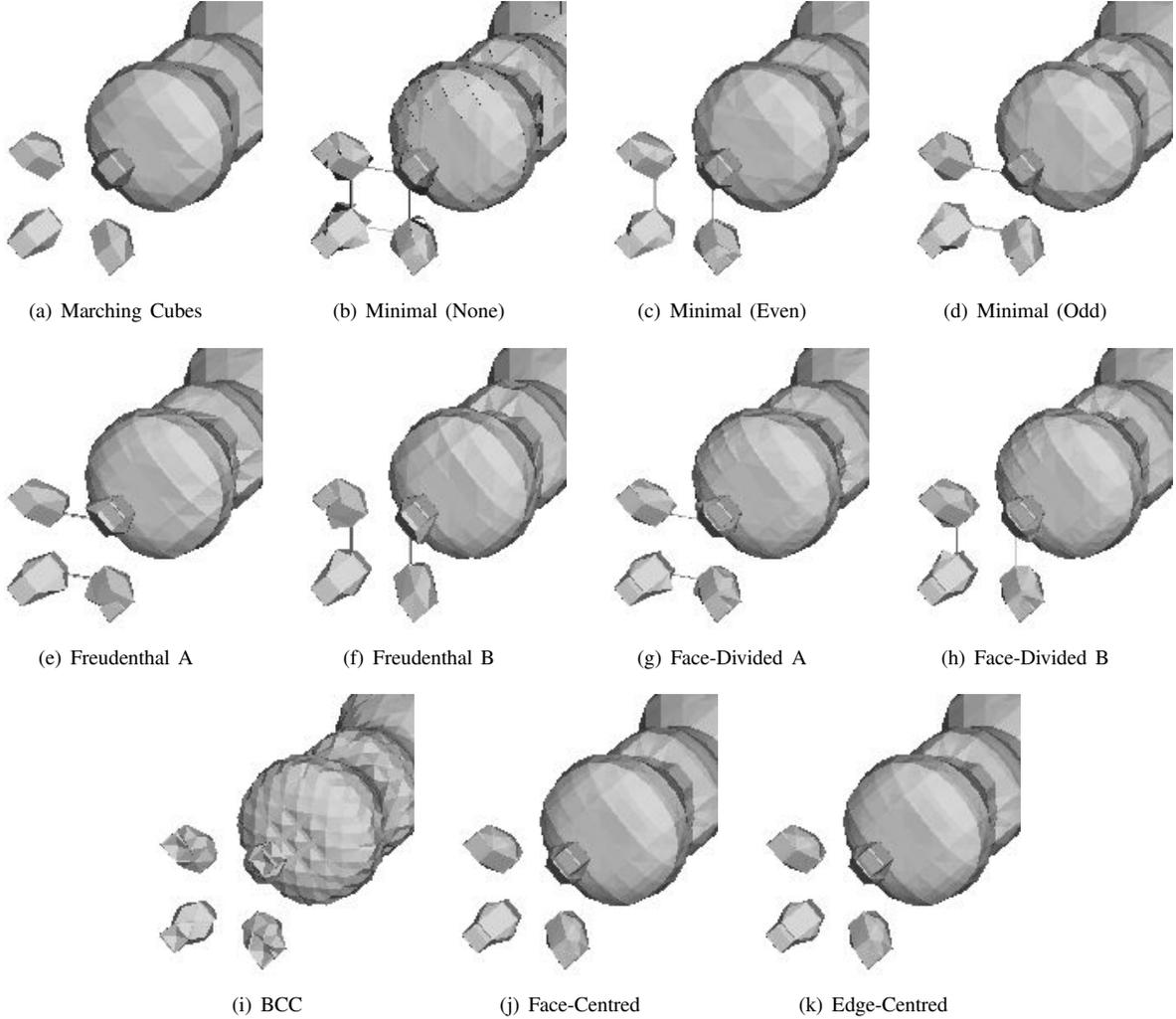


Fig. 7. Isosurface Artifacts in Fuel Dataset. Subdivision artifacts include surface texture, directional bias, sharp features and topological variations.

an implicit smoothing pass to volume rendering. In contrast, simulation data tends to have less high-frequency information and is less vulnerable to isosurface artifacts, but volume rendering artifacts are not smoothed by the effects of noise.

VIII. SAMPLING THEORY AND SUBDIVISIONS

In addition to considering numerical, topological and geometric artifacts, we also compared the interpolating kernels induced by the subdivisions with the interpolating kernels of the trilinear interpolant and the ideal (sinc) filter. We start by briefly reviewing the standard sampling theory assumptions.

Usually, no *a priori* information is present for the (continuous) underlying function that is sampled. It is instead assumed that the function is band-limited. Since the function is given on a regular rectangular grid, the aliased spectra in the frequency domain are also replicated on a regular grid. The frequency support of the function is then $[-\pi/T_x, \pi/T_x] \times [-\pi/T_y, \pi/T_y] \times [-\pi/T_z, \pi/T_z]$, where T_x , T_y , and T_z are the sampling distance in x , y , and z direction [26]. This frequency π/T is known as the *Nyquist* limit.

In order to reconstruct a continuous function from these sampled values, we would have to multiply with a box function

in the frequency domain. This is equivalent to convolution with the Sinc function ($\text{sinc}(x) = (\sin \pi x)/\pi x$) in the spatial domain. Hence the 3D Sinc function shown in Fig. 10(k) is generally considered to be the ideal reconstruction kernel. [27]

However, Sinc interpolation is expensive to compute, since it is an IIR filter (infinite impulse response), and requires the entire sampled dataset to be processed to compute a single interpolated value. As a result, many approximations to the Sinc interpolation have been suggested [27], [28].

Applying an $m \times m \times m$ filter at a point costs $O(m^3)$. For trilinear interpolation, $m = 2$ and the cost is acceptable. For higher-order (i.e. larger) filters, m increases and the cost is unacceptable. Trilinear filtering is also attractive topologically as it satisfies the intuitive expectation of connectedness in Sec. IV-E, guarantees that maxima and minima occur at grid vertices, and has a closed form for saddle points [3].

Figs. 10 and 11 show the interpolating kernels for each subdivision in both spatial and frequency domains. These kernels were computed by setting the central vertex in a $3 \times 3 \times 3$ array of samples to 1.0 and all other vertices to 0, applying the simplicial subdivision to the neighbouring cubes, and sampling the interpolated value at $256 \times 256 \times 256$. For

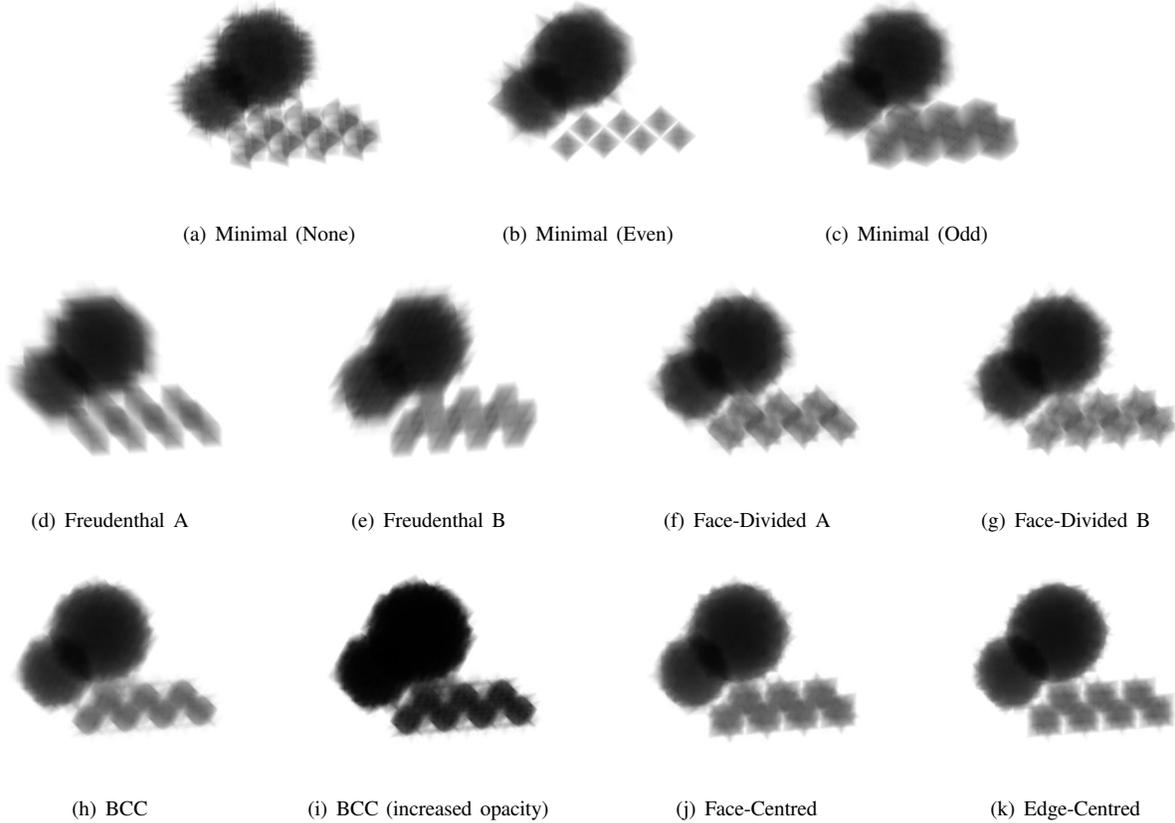


Fig. 8. Direct Volume Renderings of Atom9 Dataset Using Projected Tetrahedra

comparison, we also show the trilinear filter (j), the ideal (sinc) filter (k) and the nearest neighbour filter (l).

In Figs. 10(a) and 10(b), we show the minimal subdivision in both versions: normal and reverse parity, corresponding to the two different no-parity rules possible. It is not possible to compute interpolating kernels for the even and odd parity subdivisions, since a interpolating kernel must be uniformly applied throughout the space and these subdivisions vary throughout the space. We can see, however, that the two no-parity rules apply directional biases to the data and, in Figs. 11(a) and 11(b), significant side-lobes exist where we would prefer a compact kernel.

In Figs. 10(c) and 10(d), we again show the Freudenthal subdivision with two different dominant axes. In this subdivision the kernel aligns with the major axis in the spatial domain, and is flattened in the frequency domain along the corresponding major axis, shown in Figs. 11(c) and 11(d).

Similarly, the first face-divided subdivision shown in Figs. 10(e) and 11(e) displays the same axial bias to a lesser extent. This bias is unsurprising because this subdivision is itself a subdivision of the Freudenthal subdivision. Since accurately interpolated vertices are added and more simplices used, the error is predictably less.

Fig. 10(f) and Fig. 11(f) show the interpolating kernel for the alternate face-divided subdivision from Sec. IV-D. Here, the major diagonal is no longer explicitly visible since it is perpendicular to the simplices used. Accordingly, the kernel is flattened with respect to the major diagonal, but the bias

still exists. In the frequency domain, this causes elongation along the axis shown in Fig. 11(f).

For the BCC subdivision, we see in Fig. 10(g) that the girders in the rendered images match the similar thin structures in the interpolating kernel. Moreover, Fig. 11(g) shows us that the kernel in the frequency domain is far from ideal.

Finally, in Figures 10(h), 10(i), 11(h) and 11(i), we see that the face- and edge- centred subdivisions have kernels that are reasonably compact and well-distributed but not ideal.

We also note that all of the subdivisions show visible fingers stretching out in the frequency domain along the orthogonal and diagonal axes. One would expect these to result in high-frequency noise or artifacts in the images. If we consider Fig. 4, we see that the surfaces have sharp protruding corners along the orthogonal and diagonal axes, as expected. These protrusions are precisely what cause the texturing effect visible in Fig. 6 and Fig. 7, and we believe they represent the fingers in the interpolating kernel. Again, these sharp protrusions are not limited to isosurfacing, but are an intrinsic part of the interpolating kernel and can be seen, albeit faintly, in Fig. 8.

IX. CONCLUSION

We have reviewed the various simplicial subdivisions proposed for 3-D applications, and examined the geometric artifacts that result from each. We have also considered the sampling artifacts in the Fourier domain for each of these subdivisions. None of the subdivisions is entirely satisfactory: some care is called for when selecting one for use.

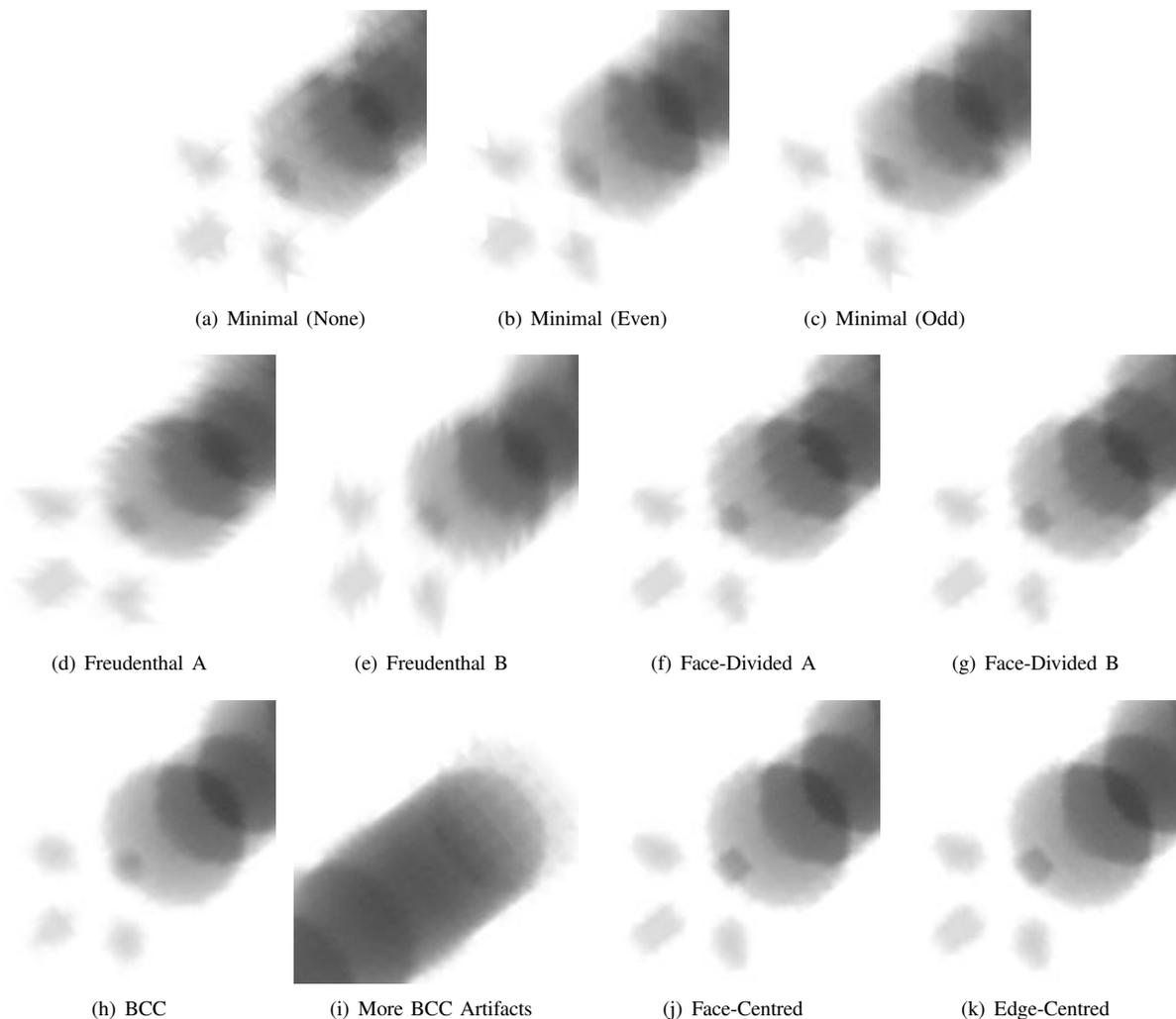


Fig. 9. Closeups of Projected Tetrahedra Volume Rendered Images for the Fuel Dataset.

Unsurprisingly, there appears to be a trade-off between size (as measured by parsimony and minimality) and accuracy (as measured by symmetry, consistence and exactness). For applications where visual artifacts are not a concern, either the minimal (5-fold) or Freudenthal (6-fold) subdivisions can be used. For isosurface rendering and segmentation, either simplicial subdivision should be avoided entirely, or cubes should be subdivided into at least 24 tetrahedra. For the Projected Tetrahedra method of volume rendering, large-scale texture is small enough that any subdivision may be used, but directional biases may still appear.

We know of no reason to use a 48-fold subdivision except for hierarchical applications [18]. Instead of interpolating additional vertices for this subdivision, additional samples should be taken (which is, in effect, what Zhou, Chen & Kaufman do). But if we increase the input size by a factor of 5 or more, how much do we gain from the simplicity of the tetrahedron's interpolation function?

We would argue that simplicial subdivisions should only be used where they are necessary for the correct operation of an algorithm, such as [2], [4], [5], [6], [12], [16]. In almost all other cases, Nielson's topologically correct trilinear cases [3]

will generate fewer triangles, require fewer interpolated points, and give a topologically correct surface.

Our analysis of tetrahedral subdivisions took place in the context of regularly sampled grids. Although most of the analysis also applies to curvilinear grids, or irregular grids using hexahedra, additional work is possible in these realms: for example, although simpler, it would be useful to extend the analysis to prisms in irregular grids. It might also be profitable to construct a more rigorous taxonomy of all possible tetrahedral subdivisions, similar to that in [8].

ACKNOWLEDGMENTS

This work has been supported by NSERC through a post-graduate fellowship and a research grant, by IRIS through a research grant, and by NSF grants 9988742 and 0076984, and by the University of British Columbia, where the first author was a Ph.D. student. Thanks are also due to the University of North Carolina at Chapel Hill for the UNC head data set.

REFERENCES

- [1] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

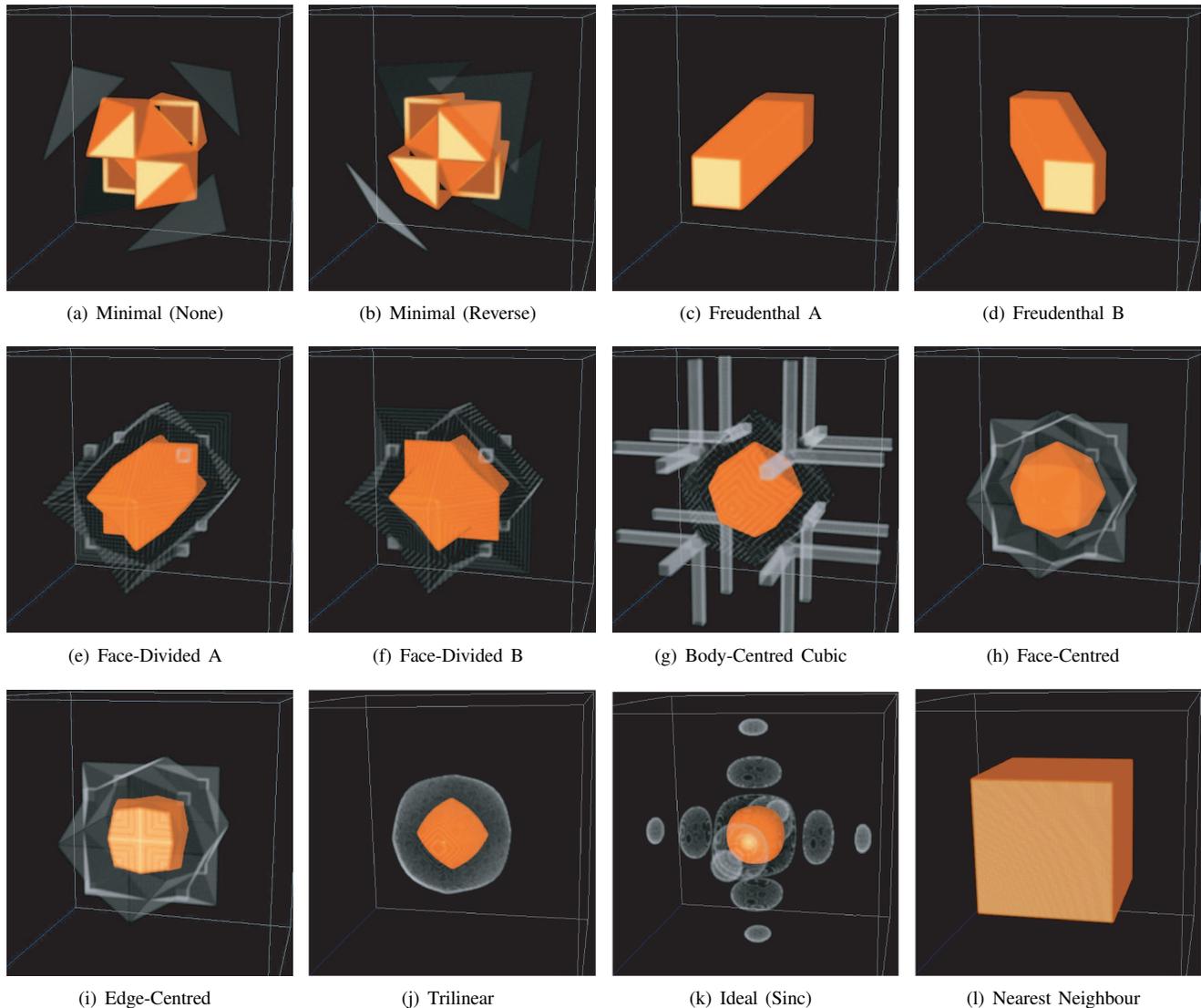


Fig. 10. Spatial Interpolating Kernels for the Subdivisions. These images were volume-rendered to emphasize a medium and a low isovalue, which illustrate the directional biases. Note that the parity-rule versions of the 5-fold minimal subdivision do not generate interpolating kernels that can be uniformly applied throughout space and are not shown. Instead, (a) and (b) show the kernels for the two possible minimal subdivisions.

- [2] P. Shirley and A. Tuchman, "A Polygonal Approximation to Direct Scalar Volume Rendering," *Computer Graphics*, vol. 24, no. 5, pp. 63–70, 1990.
- [3] G. M. Nielson, "On Marching Cubes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 283–297, 2003.
- [4] H. Carr, J. Snoeyink, and U. Axen, "Computing Contour Trees in All Dimensions," *Computational Geometry: Theory and Applications*, vol. 24, no. 2, pp. 75–94, 2003.
- [5] N. L. Max, P. Hanrahan, and R. A. Crawfis, "Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions," *Computer Graphics*, vol. 24, no. 5, pp. 27–33, 1990.
- [6] C. Stein, B. Becker, and N. L. Max, "Sorting and Hardware Assisted Rendering for Volume Visualization," in *Proceedings of Visualization 1994*, pp. 83–89.
- [7] M. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. R. Schikore, "Contour Trees and Small Seed Sets for Isosurface Traversal," in *Proceedings of the 13th ACM Symposium on Computational Geometry*, 1997, pp. 212–220.
- [8] G. Albertelli and R. A. Crawfis, "Efficient Subdivision of Finite-Element Datasets into Consistent Tetrahedra," in *Proceedings of Visualization 1997*, pp. 213–219.
- [9] J. Bloomenthal, "Polygonization of implicit surfaces," *Computer Aided Geometric Design*, pp. 341–355, 1988.
- [10] P. Ning and J. Bloomenthal, "An Evaluation of Implicit Surface Tilers," *IEEE Computer Graphics and Applications*, vol. 13, pp. 33–41, 1993.
- [11] G. M. Nielson and R. Franke, "Computing the Separating Surface for Segmented Data," in *Proceedings of Visualization 1997*, pp. 229–233.
- [12] G. M. Nielson and J. Sung, "Interval Volume Tetrahedrization," in *Proceedings of Visualization 1997*, pp. 221–228.
- [13] E. L. Allgower and S. Gnutzmann, "Simplicial Pivoting for Mesh Generation of Implicitly Defined Surfaces," *Computer Aided Geometric Design*, vol. 8, pp. 305–325, 1991.
- [14] H. Freudenthal, "Simplizialzerlegungen von Beschränkter Flachheit," *Annals of Mathematics*, vol. 43, no. 3, pp. 580–582, 1942.
- [15] S. Lefschetz, *Transformations of Complexes (ch. IV)*. Princeton, NJ: Princeton University Press, 1949, pp. 110–141.
- [16] M. P. Garrity, "Raytracing Irregular Volume Data," *Computer Graphics*, vol. 24, no. 5, pp. 35–40, 1990.
- [17] C. Weigle and D. C. Banks, "Complex-Valued Contour Meshing," in *Proceedings of Visualization 1996*, pp. 173–181.
- [18] Y. Zhou, B. Chen, and A. E. Kaufman, "Multiresolution Tetrahedral Framework for Visualizing Regular Volume Data," in *Proceedings of Visualization 1997*, pp. 135–142.
- [19] H. Carr, T. Möller, and J. Snoeyink, "Simplicial Subdivisions and Sampling Artifacts," in *Proceedings of Visualization 2001*, pp. 99–106.

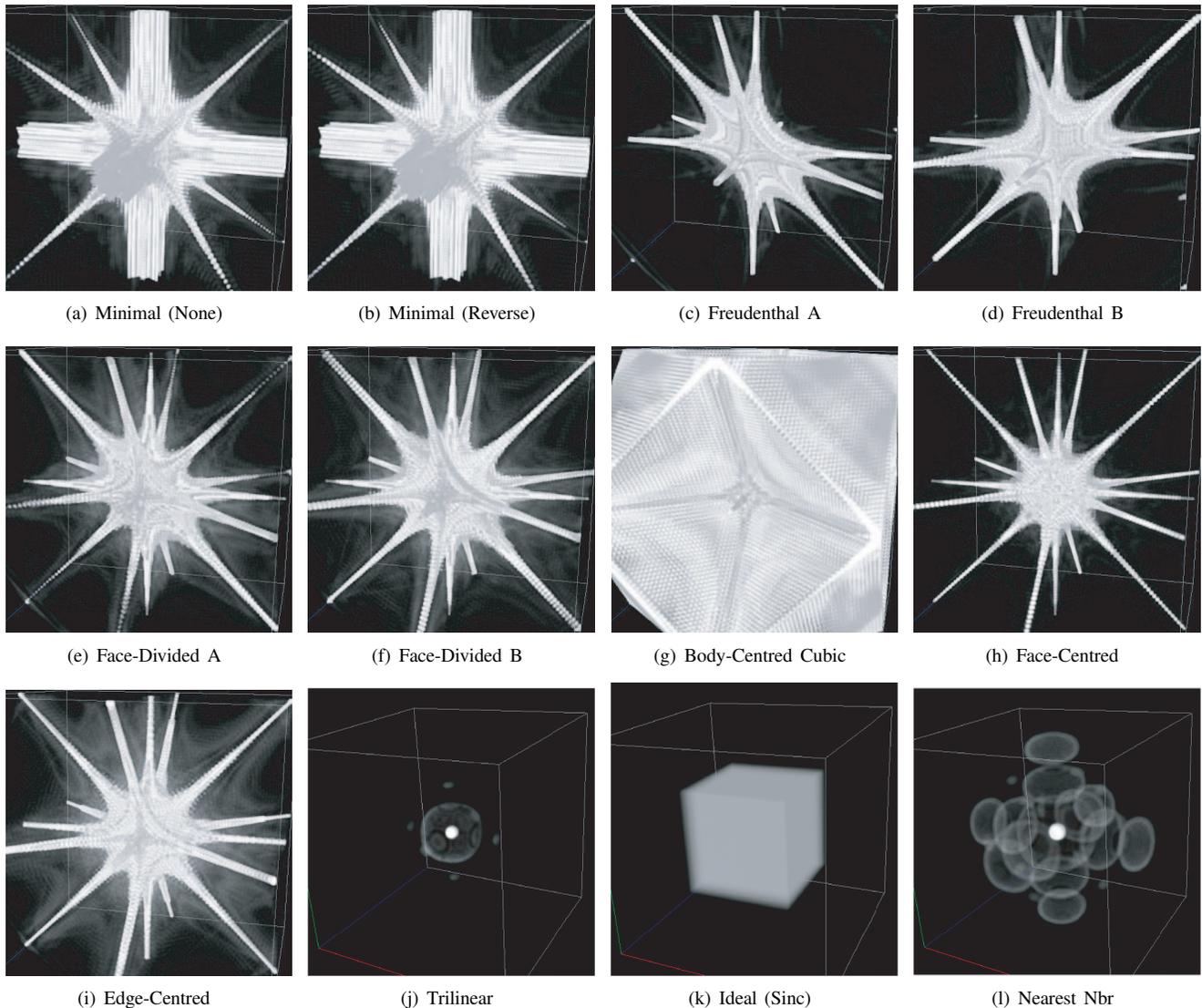


Fig. 11. Interpolating Kernels in the Frequency Domain. These images were volume-rendered to emphasize a single isovalue that illustrates the directional aliasing. Again, the parity-rule versions of the 5-fold minimal subdivision do not generate interpolating kernels that can be uniformly applied throughout space and are not shown. Instead, (a) and (b) show the kernels for the two possible minimal subdivisions.

- [20] B. Natarajan, "On generating topologically consistent isosurfaces from uniform samples," *Visual Computer*, vol. 11, pp. 52–62, 1994.
- [21] H. Carr, T. Theußl, and T. Möller, "Isosurfaces on Optimal Regular Samples," in *Proceedings of Eurographics Visualization Symposium 2003*, pp. 39–48, 284.
- [22] S. Röttger, M. Kraus, and T. Ertl, "Hardware-Accelerated Volume And Isosurface Rendering Based On Cell-Projection," in *Proceedings of Visualization 2000*, pp. 109–116.
- [23] S. Röttger and T. Ertl, "A two-step approach for interactive pre-integrated volume and isosurface rendering of unstructured grids," in *Proceedings of Volume Visualization 2002*, pp. 23–28.
- [24] B. Wylie, K. Moreland, L. A. Fisk, and P. Crossno, "Tetrahedral projection using vertex shaders," in *Proceedings of Volume Visualization 2002*, pp. 7–12.
- [25] M. Kraus, W. Qiao, and D. S. Ebert, "Projecting Tetrahedra without Rendering Artifacts," in *Proceedings of Visualization 2004*, pp. 27–34.
- [26] D. E. Dudgeon and R. Mersereau, *Multidimensional Digital Signal Processing*. Englewood-Cliffs, NJ: Prentice-Hall Inc., 1984.
- [27] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*. Englewood-Cliffs, NJ: Prentice-Hall Inc., 1989.
- [28] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel, "Design of accurate and smooth filters for function and derivative reconstruction," in *Proceedings of Volume Visualization 1998*, pp. 143–151.

Hamish Carr Hamish Carr is a Lecturer in the School of Computer Science at University College Dublin. His research interests include computational geometry, computer graphics and scientific and medical visualization.

Torsten Möller Torsten Möller is an Associate Professor at the School of Computing Science at Simon Fraser University. His research interests include the fields of scientific visualization and computer graphics, especially the mathematical foundations of visualization and graphics.

Jack Snoeyink Jack Snoeyink is a Professor in the Department of Computer Science at the University of North Carolina at Chapel Hill. His main research is in computational geometry – algorithms and data structures for geometric problems that arise in computer graphics, GIS, and molecular biology.