Preprint
UCRL-JC-146824

Hierarchical Volume Representation with $\sqrt[3]{2}$ Subdivision and Trivariate B-spline Wavelets

L. Linsen, J. T. Gray, V. Pascucci, M. A. Duchaineau, B. Hamann, K. I. Joy

This article was submitted to Siggraph 2002, San Antonio, Texas, July 21 – 26, 2002

January 11, 2002

U.S. Department of Energy

Lawrence Livermore National Laboratory

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Hierarchical Volume Representation with $\sqrt[3]{2}$ Subdivision and Trivariate B-spline Wavelets

Lars Linsen* Jevan T. Gray* Valerio Pascucci† Mark A. Duchaineau† Bernd Hamann*
Kenneth I. Joy*

* Center for Image Processing and Integrated Computing
University of California - Davis‡

† Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

Abstract

Multiresolution methods provide a means for representing data at multiple levels of detail. They are typically based on a hierarchical data organization scheme and update rules needed for data value computation. We use a data organization that is based on what we call $\sqrt[n]{2}$ subdivision. The main advantage of $\sqrt[n]{2}$ subdivision, compared to quadtree ($n = 2$) or octree ($n = 3$) organizations, is that the number of vertices is only doubled in each subdivision step instead of multiplied by a factor of four or eight, respectively. To update data values we use n-variate B-spline wavelets, which yields better approximations for each level of detail. We develop a lifting scheme for $n = 2$ and $n = 3$ based on the $\sqrt[n]{2}$-subdivision scheme. We obtain narrow masks that could also provide a basis for view-dependent visualization and adaptive refinement.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling Project—; J [Computer Applications]: Physical Sciences and Engineering—Life and Medical Science

Keywords: hierarchical volume modeling, subdivision, B-spline wavelets, lifting, multiresolution modeling, data approximation

1 Introduction

Multiresolution schemes are used in computer graphics mainly for editing and rendering curves and surfaces at multiple levels of resolution. While most existing schemes could, in principle, be generalized for higher dimensions, only a few have been extended to data (or functions) defined over higher-dimensional domains. The combined subdivision-wavelet scheme we are describing in this paper is driven by the need to represent trivariate data (or functions) at multiple resolution levels.

Representing volume data hierarchically is especially important in the context of "volume modeling" and visualizing volume data, e.g., scalar or vector fields defined over volumetric domains. Visualizing inherently trivariate phenomena often requires one to apply rendering operations to the volumetric data - examples being volume slicing via a cutting plane, isosurface extraction through marching-cubes-like algorithms, and ray casting. The multiresolution approximation approach we develop in this paper provides an elegant means of hierarchically organizing volume data, and we can use the resulting hierarchy to apply to its various levels volume data visualization methods.

We combine $\sqrt[n]{2}$ subdivision with n-variate B-spline wavelets for n-dimensional multiresolution data representation. One- and two-dimensional multiresolution schemes have been studied extensively over the past decade. A survey of the main multiresolution approaches, considering also topological constraints, is given by Kobbelt in [Kobbelt 2002]. These approaches can, for example, be used for a multiresolution representation of isosurfaces. However, when considering (bio-)medical imaging data, we must be able to switch quickly between isosurfaces corresponding to different isovalues, and when considering, for example, numerically simulated time-dependent hydrodynamics data, we even have to deal with isosurfaces changing over time. It is practically impossible to store every single isosurface for all possibly important isovalues at different resolutions and reload them during visualization. Instead, we establish a multiresolution volume data representation. We first develop a bivariate B-spline wavelet scheme for $\sqrt{2}$ subdivision and then generalize it to a trivariate B-spline wavelet scheme for $\sqrt[3]{2}$ subdivision. We apply our techniques to bivariate as well as volumetric data.

For three-dimensional multiresolution representation, one may use an octree structure, see, for example, [Pinskiy et al. 2001], but each octree refinement step doubles the number of vertices in every dimension. This leads to a factor of eight for every refinement step. In the following section, we introduce the $\sqrt[n]{2}$-subdivision scheme. Every $\sqrt[n]{2}$-subdivision step only doubles the number of vertices, which theoretically is a factor of $\sqrt[n]{2}$ in each of the n dimensions.

When using a wavelet scheme, the data value at a vertex $\mathbf{p}$ is updated when changing the level of detail, and thus the value varies in the different levels of detail. On a coarse level, the value represents the value at $\mathbf{p}$ itself as well as an average value of a certain region around $\mathbf{p}$. This leads to better approximations on coarser levels. Wavelets based on the $\sqrt[n]{2}$-subdivision scheme unfortunately have the disadvantage of creating over- and undershoots. For example, for isosurface extraction ($n = 3$) this characteristic can cause the side effect of creating isosurfaces (or isosurface components) that vanish when increasing the resolution. Therefore, we use n-variate B-spline wavelets and adjust them to the $\sqrt[n]{2}$-subdivision scheme.

B-spline wavelets have the property that they do not only influence the neighbors of a vertex $\mathbf{p}$. Therefore, lifting schemes with narrow filters for quadtree structures were introduced, see, for example, [Bertram et al. 2001]. We review and generalize the lifting

*llinsen@ucdavis.edu, {grayj,hamann,joy}@cs.ucdavis.edu
†{pascucci1,duchaineau1}@llnl.gov
‡http://graphics.cs.ucdavis.edu

scheme from [Bertram et al. 2001] in Section 3. In Sections 4 and 6, we develop a similar lifting scheme for the $\sqrt[n]{2}$-subdivision scheme for $n = 2$ and $n = 3$, respectively. We show some results in Sections 5 and 7, respectively.

2 $\sqrt[n]{2}$ subdivision

The splitting step of the $\sqrt[n]{2}$-subdivision scheme goes back to [Maubach 1995], and was used more recently for subdivision surfaces, see, for example, [Velho and Zorin 2001]. Velho and Zorin introduced an averaging step and showed that the produced surfaces are C^4-continuous at regular and C^1-continuous at extraordinary vertices. (For an introduction to subdivision methods, we refer to [Warren and Weimer 2002].)

We first describe the case $n = 2$. The two-dimensional scheme can be defined for triangular as well as quadrilateral meshes. Without loss of generality, we describe the quadrilateral case only.

For a $\sqrt{2}$-subdivision step of a quadrilateral Q, we compute the centroid $\mathbf{c}$ of Q, and connect $\mathbf{c}$ to all four vertices of Q by new edges. The former edges of the mesh are removed (except for the edges determining the outer mesh boundary). Figure 1 illustrates four $\sqrt{2}$-subdivision steps.

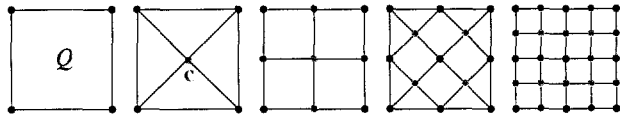


Figure 1: $\sqrt{2}$ subdivision.

The mask used for the computation of the centroid $\mathbf{c}$ is given in Figure 2(a). Figure 2(b) shows the mask of the averaging step according to [Velho and Zorin 2001]. A $\sqrt{2}$-subdivision step is executed by first applying the mask in Figure 2(a), which inserts the new vertices, and then (after the topological mesh modifications) applying the mask in Figure 2(b), which repositions the old vertices.

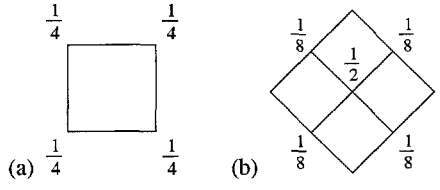


Figure 2: Masks of $\sqrt{2}$-subdivision step: (a) inserting centroid; (b) repositioning old vertices.

When splitting every quadrilateral along one of its diagonal, we obtain a subdivision scheme for triangular meshes. The splitting step for triangular meshes is based on bisection of the longest edge. By using quadrilateral meshes the subdivision scheme is analogous to the $\sqrt{3}$-subdivision scheme of [Kobbelt 2000]. Therefore, we call it $\sqrt{2}$ subdivision.

We now generalize the subdivision scheme to $\sqrt[n]{2}$ subdivision for arbitrary dimension n. The splitting step is executed by inserting the centroid and adjusting the vertex connectivity. The averaging step applies to every old vertex $\mathbf{v}$ the update rule

$$\mathbf{v} = \alpha\mathbf{v} + (1-\alpha)\mathbf{w}\,,$$

where $\mathbf{w}$ is the centroid of the adjacent new vertices.

We are especially interested in the case $n = 3$. Little research has been done to date concerning three-dimensional (volumetric) subdivision. One example is the work described in [MacCracken and Joy 1996]. The literature currently provides no analysis of averaging steps for dimensions larger than two. Thus, at present, we cannot provide a solution for the choice of α in the update rule. Some investigations about applying the update rule in arbitrary dimensions were made by Pascucci in [Pascucci 2001].

When applying the $\sqrt[3]{2}$-subdivision scheme to large volumetric data sets, we usually deal with more structured rectilinear grids, especially when considering imaging data sets, In this case, eight cuboids share a vertex, and the cuboids have the same size. For such "structured-rectilinear" grids, the update rule does not change the position of the vertices regardless of the specific α value.

In Figure 3, three $\sqrt[3]{2}$-subdivision steps are shown. In each step, the centroids of the polyhedral shapes are inserted and the connectivity is adjusted. Three kinds of polyhedral shapes arise. They are shown in Figure 4.

In the first step, each cuboid (first picture of Figure 3 or third picture of Figure 4, respectively) is subdivided by inserting the cuboid's centroid and connecting the centroid to all old vertices (second picture of Figure 3). In the second step, each octahedron (first picture of Figure 4) is subdivided by inserting the octahedron's centroid and connecting the centroid to all old vertices, while all old edges, except for the (red) edges inserted in the last subdivision step, are deleted (third picture of Figure 3). In the third step, each octahedron with split faces (second picture of Figure 4) is subdivided by inserting its centroid and connecting the centroid to all old vertices except for the (red) vertices inserted in the next-to-the-last subdivision step, while all old edges, except for the edges between the (red) vertices inserted in the next-to-the-last subdivision step and the (green) vertices inserted in the last step, are deleted (fourth picture of Figure 3).

The three subdivision steps can also be described in the following way: The first step inserts the centroid of the cuboid, the second inserts the centers of the faces of the original cuboid, and the third inserts the midpoints of the edges of the original cuboid. Three $\sqrt[3]{2}$-subdivision steps produce the same result as one octree refinement step. Hence, for multiresolution purposes, we obtain a much finer granularity through $\sqrt[3]{2}$ subdivision. We can thus get much closer to a required level of mesh-element size, and therefore we most likely will have to render less data to obtain a desired image / visualization quality.

The $\sqrt[3]{2}$-subdivision step for a tetrahedral grid is again based on bisection of the longest edge. In [Zhou et al. 1997], a refinement scheme for tetrahedral grids is defined, which is analogous to the splitting step used in $\sqrt[3]{2}$ subdivision. Zhou et al. do not provide a full subdivision scheme, since the averaging step is missing. Thus, their scheme is restricted to structured-rectilinear grids.

The $\sqrt[3]{2}$-subdivision scheme also applies to structured-curvilinear grids, where, in general, eight cubes share a common vertex. The scheme could even handle extraordinary vertices, but for the lifting scheme, which we introduce in this paper, we do not consider extraordinary vertices. It is our goal to apply the scheme to large data sets. Grids of arbitrary topology might be too time-consuming to process.

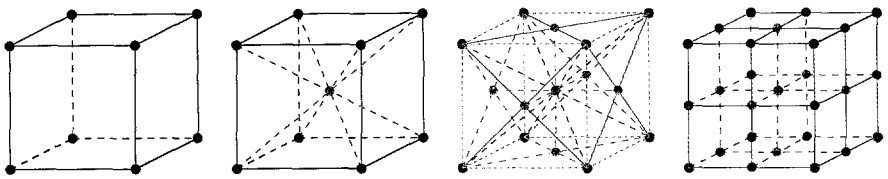


Figure 3: $\sqrt[3]{2}$ subdivision.

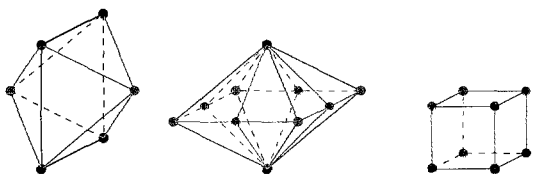


Figure 4: Polyhedral shapes created by $\sqrt[3]{2}$ subdivision: octahedron, octahedron with split faces, and cuboid.

3 The B-spline wavelet lifting scheme

The main advantage of wavelet schemes is the fact that they provide a means to generate best approximations in a multiresolution hierarchy. Stollnitz et al. in [Stollnitz et al. 1996] describe how to generate wavelets for subdivision schemes. However, $\sqrt[n]{2}$-subdivision wavelets can lead to over- and undershoots, which are especially disturbing when extracting isosurfaces from different levels of approximation. They can even cause topological changes of isosurfaces when changing the level of resolution. Therefore, we have decided to generate B-spline wavelets for the $\sqrt[n]{2}$-subdivision scheme. (For an introduction to B-spline techniques, we refer to [Prautzsch et al. 2002].)

The computation of the B-spline wavelet coefficients at a certain vertex is not limited to using only adjacent vertices. In [Bertram et al. 2000], and more recently in [Bertram et al. 2001], a lifting scheme was developed. The described lifting scheme decomposes the computation into several steps, but narrow filters are asserted, i. e., only adjacent vertices are used. In [Bertram et al. 2000], Bertram et al. define the lifting scheme for the one-dimensional and two-dimensional cases using a quadtree organization of the vertices.

We review and define masks for the one-dimensional lifting scheme of [Bertram et al. 2000] and generalize them to the two- and three-dimensional cases. In the following sections, we will adjust the two-dimensional lifting scheme to $\sqrt{2}$ subdivision and the three-dimensional lifting scheme to $\sqrt[3]{2}$ subdivision.

The one-dimensional B-spline wavelet lifting scheme makes use of two operations that are defined by the following two masks, called s-lift and w-lift:

$$\text{s-lift}(a,b): \quad \begin{pmatrix} a & b & a \end{pmatrix}, \tag{1}$$

$$\text{w-lift}(a,b): \quad \begin{pmatrix} a & b & a \end{pmatrix} \tag{2}$$

The s-lift mask is applied to the old vertices (black) and their new neighbors (blue), whereas the w-lift mask is applied to the new vertices (blue) and their neighbors (black), see Figure 5(a). For a detailed derivation of the lifting scheme that we use as a basis for this paper, we refer to [Bertram 2000].

Linear B-spline wavelets
Using the s-lift and w-lift masks, a linear B-spline wavelet encoding step is defined by sequentially executing the two operations

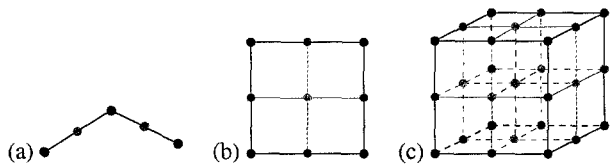


Figure 5: Refinement step for one-, two-, and three-dimensional meshes.

$$\text{w-lift}(-\tfrac{1}{2},1) \text{ and}$$
$$\text{s-lift}(\tfrac{1}{4},1)\,.$$

A linear B-spline wavelet decoding step is defined by sequentially executing the two operations

$$\text{s-lift}(-\tfrac{1}{4},1) \text{ and}$$
$$\text{w-lift}(\tfrac{1}{2},1)\,.$$

Cubic B-spline wavelets
Using the same masks, a cubic B-spline wavelet encoding step is defined by the lifting operations

$$\text{s-lift}(-\tfrac{1}{2},2)\,,$$
$$\text{w-lift}(-\tfrac{1}{2},1) \text{ and}$$
$$\text{s-lift}(\tfrac{3}{8},1)\,.$$

A cubic B-spline wavelet decoding step is defined by the lifting operations

$$\text{s-lift}(-\tfrac{3}{8},1)\,,$$
$$\text{w-lift}(\tfrac{1}{2},1) \text{ and}$$
$$\text{s-lift}(\tfrac{1}{4},\tfrac{1}{2})\,.$$

When applying two-dimensional B-spline wavelets to a quadtree-organized set of vertices, two kinds of new vertices are obtained when executing a refinement step, namely the new vertices inserted at the midpoint of an old edge (blue) and the new vertices inserted at the center of an old face (green), see Figure 5(b). Therefore, we have two different masks. We derive the needed two-dimensional masks by convolution of the one-dimensional masks in the two coordinate directions. The results of this convolution step

are the two-dimensional masks in

$$\text{s-lift}(a,b): \quad \begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}, \tag{3}$$

$$\begin{pmatrix} a & b & a \end{pmatrix} \quad \text{and} \tag{4}$$

$$\text{w-lift}(a,b): \quad \begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}, \tag{5}$$

$$\begin{pmatrix} a & b & a \end{pmatrix}. \tag{6}$$

The one-dimensional masks defined by (4) and (6) are applied in both directions. The masks (3) and (4), as well as masks (5) and (6), are applied simultaneously.

When applying three-dimensional B-spline wavelets to an octree-organized set of vertices, three kinds of new vertices are obtained when executing a refinement step, namely the new vertices inserted at the midpoint of an old edge (blue), the new vertices inserted at the center of an old face (green), and the new vertices inserted at the centroid of an old cube (red), see Figure 5(c). Therefore, we have three different masks. For three-dimensional masks, we show the structure of the mask and separately define the values for the black, blue, green, and red vertices, respectively. We derive the needed three-dimensional masks by convolution of the one-dimensional masks in all three coordinate directions. They are given by

$$\text{s-lift}(a,b): \quad \begin{matrix} b^3 \\ ab^2 \\ a^2b \\ a^3 \end{matrix}, \tag{7}$$

$$\begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}, \tag{8}$$

$$\begin{pmatrix} a & b & a \end{pmatrix} \quad \text{and} \tag{9}$$

$$\text{w-lift}(a,b): \quad \begin{matrix} a^3 \\ a^2b \\ ab^2 \\ b^3 \end{matrix}, \tag{10}$$

$$\begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}, \tag{11}$$

$$\begin{pmatrix} a & b & a \end{pmatrix}. \tag{12}$$

4 A lifting scheme for $\sqrt{2}$ subdivision

Using $\sqrt{2}$ subdivision instead of a quadtree-based scheme, we only obtain new vertices at the centers of old faces (green) when executing a subdivision step; at the midpoints of old edges (blue) no vertices are inserted, see second picture in Figure 1 and compare to Figure 5(b). Thus, no data is available at the positions of the blue vertices, and we must adjust the two-dimensional masks (3) and (5).

First, we consider linear B-spline wavelets. For encoding in the linear case, the w-lift operation is executed first. Since we have no values at the blue positions required for mask (5), we linearly interpolate the values at the black vertices. This approach changes mask (5) to

$$\text{w-lift}_{encode}(a,b): \quad \begin{pmatrix} a^2+ab & & a^2+ab \\ & b^2 & \\ a^2+ab & & a^2+ab \end{pmatrix}. \tag{13}$$

Next, the s-lift operation is executed. Again, we have entries at the blue positions in mask (3). However, the w-lift operation has (theoretically) executed mask (6), and we have assumed that the values at the blue vertices are linear interpolations of the values at the black vertices; therefore, the values at the blue vertices have vanished. Mask (3) changes to

$$\text{s-lift}_{encode}(a,b): \quad \begin{pmatrix} a^2 & & a^2 \\ & b^2 & \\ a^2 & & a^2 \end{pmatrix}. \tag{14}$$

For decoding, we first execute the s-lift operation. Prior to executing the s-lift operation of the encoding, the values at the blue vertices have vanished, but the s-lift operation (theoretically) executed mask (4). Hence, the values at the blue vertices are now given by linear interpolation of the values at the green neighbor vertices multiplied by the factor $2a$ of mask (4). We rename the factor a to $\bar{a}$ and derive from mask (3) the new mask

$$\text{s-lift}_{decode}(a,b): \quad \begin{pmatrix} a^2+2\bar{a}ab & & a^2+2\bar{a}ab \\ & b^2 & \\ a^2+2\bar{a}ab & & a^2+2\bar{a}ab \end{pmatrix}. \tag{15}$$

Finally, the w-lift operation is executed again. The s-lift decoding operation has (theoretically) applied mask (4). Since mask (4) applied by the s-lift decoding operation is the inverse of mask (4) applied by the s-lift encoding operation, the values at the blue vertices are the same as before the execution of these two s-lift operations, i.e., they vanish. These considerations define a new mask derived from mask (5), given by

$$\text{w-lift}_{decode}(a,b): \quad \begin{pmatrix} a^2 & & a^2 \\ & b^2 & \\ a^2 & & a^2 \end{pmatrix}. \tag{16}$$

The lifting operations for encoding and decoding do not have the same structure any longer, but, by substituting the a's and b's by new coefficients, the lifting operations could be transformed into a unique structure again. However, the masks are as narrow as they can be.

We now consider cubic B-spline wavelets. Analogously to the linear case, we derive masks for the cubic case. The encoding starts with an s-lift operation, i.e., we use a modified form of mask (3). Assuming that the values at the blue vertices are given by linear interpolation of the ones at the black neighbor vertices, we obtain the mask

$$\text{s-lift}_{encode1}(a,b): \quad \begin{pmatrix} & \frac{ab}{2} & \\ a^2 & & a^2 \\ \frac{ab}{2} & b^2+2ab & \frac{ab}{2} \\ a^2 & & a^2 \\ & \frac{ab}{2} & \end{pmatrix}.$$

Next, the w-lift operation is executed using mask (5). Since both the blue and black vertices have been similarly updated by the first s-lift operation, the values at the blue vertices can still be viewed as linearly interpolated values of the ones at the black neighbor vertices. This view however, is only an "approximation." With

this approximation, we are left with the same situation as the one before the start of the encoding in the linear case, which leads to the masks w-lift$_{encode}(a,b)$: mask (13) and s-lift$_{encode2}(a,b)$: mask (14).

Using the ideas of the linear case, we obtain for decoding the masks s-lift$_{decode1}(a,b)$: mask (15) and w-lift$_{decode}(a,b)$: mask (16). Before executing the w-lift decoding operation, the values at the blue vertices have vanished (as in the linear case). During the w-lift decoding operation, mask (6) is applied (theoretically); thus, the values at the blue vertices are now given by linear interpolation of the black neighbor vertices multiplied by the factor $2a$ of mask (6). We rename the factor a to $\bar{a}$ and derive from mask (3) the new mask

$$\text{s-lift}_{decode2}(a,b): \quad \begin{pmatrix} & & \bar{a}ab & & \\ & a^2 & & a^2 & \\ \bar{a}ab & & b^2+4\bar{a}ab & & \bar{a}ab \\ & a^2 & & a^2 & \\ & & \bar{a}ab & & \end{pmatrix}.$$

The lifting operations have lost even more of their unique structure. Some of the masks have also grown in size.

5 2D Results

In Figure 6, we provide an example for $\sqrt{2}$ subdivision and two-dimensional wavelets. The original surface shown in 6(a) and 6(e) results from sampling a twodimensional Gaussian function at 64^2 vertices. The surface is encoded and decoded again. In 6(b) and 6(f), we show two different levels of detail obtained by $\sqrt{2}$-subdivision wavelets. In 6(c) and 6(g), and in 6(d) and 6(h), respectively, we show the same levels of detail obtained when combining B-spline wavelets and $\sqrt{2}$ subdivision in the way described in the previous section. In 6(c) and 6(g), we have used bilinear B-spline wavelets, whereas in 6(d) and 6(h) we have used bicubic B-spline wavelets.

In 6(b), and especially in 6(f), the over- and undershoots caused by the $\sqrt{2}$-subdivision wavelets can be recognized. In 6(d) and 6(h), less pronounced over- and undershoots result, when combining $\sqrt{2}$ subdivision with cubic B-spline wavelets. No over- and undershoots emerge when combining $\sqrt{2}$ subdivision with linear B-spline wavelets, see 6(c) and 6(g).

6 A lifting scheme for $\sqrt[3]{2}$ subdivision

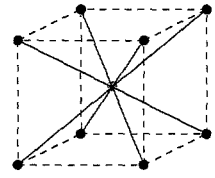
In this section, we generalize the ideas of Section 4 to the three-dimensional case. Since the masks for the cubic lifting scheme are not as narrow as the masks for the linear lifting scheme (see Section 4), and since the linear B-spline wavelets are more suitable for high approximation quality on every level of a multiresolution scheme (see Section 5), we only consider the linear case. However, the ideas can be extended to the cubic case easily.

Recalling the steps of a $\sqrt[3]{2}$-subdivision scheme depicted in Figure 3, after the execution of the different steps different kinds of polyhedral shapes arise, see Figure 4. Therefore, we have to distinguish between the different steps. The following description starts with the situation shown in the second picture of Figure 3 (*volume case*), proceeds with the situation shown in the third picture (*face case*), and finally treats the situation shown in the fourth picture (*edge case*), which is topologically equivalent to the situation shown in the first picture.

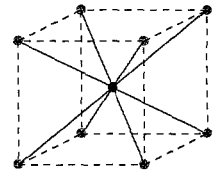
The volume case

To perform linear B-spline wavelet encoding in the situation shown in the second picture of Figure 3, we first execute a w-lift operation. Therefore, we apply three masks being similar to masks (10), (11), and (12), subject to the constraint that no values are available at the blue and green vertices.

Regarding the structures of masks (10), we assume that the value at a blue vertex is defined by linear interpolation of the values at the two black vertices (with which the blue vertex shares an edge), and that the value at a green vertex is defined by bilinear interpolation of the values at the four black vertices (with which the green vertex shares a face). One obtains the mask

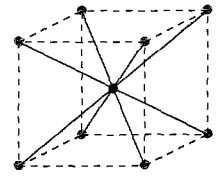
w-lift$_{encode}(a,b)$:  $a^3+\frac{3}{2}a^2b+\frac{3}{4}ab^2$, b^3.

The masks being analogous to masks (11) and (12) are only "applied theoretically." However, since the values at the blue vertices are assumed to be linear interpolations of the values at the black vertices, and since the values at the green vertices are assumed to be bilinear interpolations of the values at the black vertices, the values at both the blue and green vertices vanish. Therefore, the mask for the next s-lift operation, which is an analogue of mask (7), reduces to

s-lift$_{encode}(a,b)$:  b^3, a^3.

Again, the analogous versions of masks (8) and (9) are only applied theoretically.

For the decoding step, we start with the s-lift operation, i. e., we adjust mask (7). Having (theoretically) applied masks (8) and (9) with vanishing values at the blue and green vertices, the values at the green vertices are linear interpolations of the values at the red neighbor vertices, multiplied by the factor $2a$, and the values at the blue vertices are bilinear interpolations of the values at the red neighbor vertices, multiplied by the factor $4a^2$. By renaming the factor a to $\bar{a}$, we obtain the following mask

s-lift$_{decode}(a,b)$:  b^3, $a^3+3\bar{a}a^2b+3\bar{a}^2ab^2$.

Again, the analogous versions of masks (8) and (9) are only applied theoretically. Since masks (8) and (9) of this s-lift operation are the inverse masks of masks (8) and (9) of the encoding s-lift operation, the blue and green vertices have their former values assigned again, i. e., the values vanish. Hence, the mask for the final w-lift operation, which is the mask being analogous to mask (10), reduces to

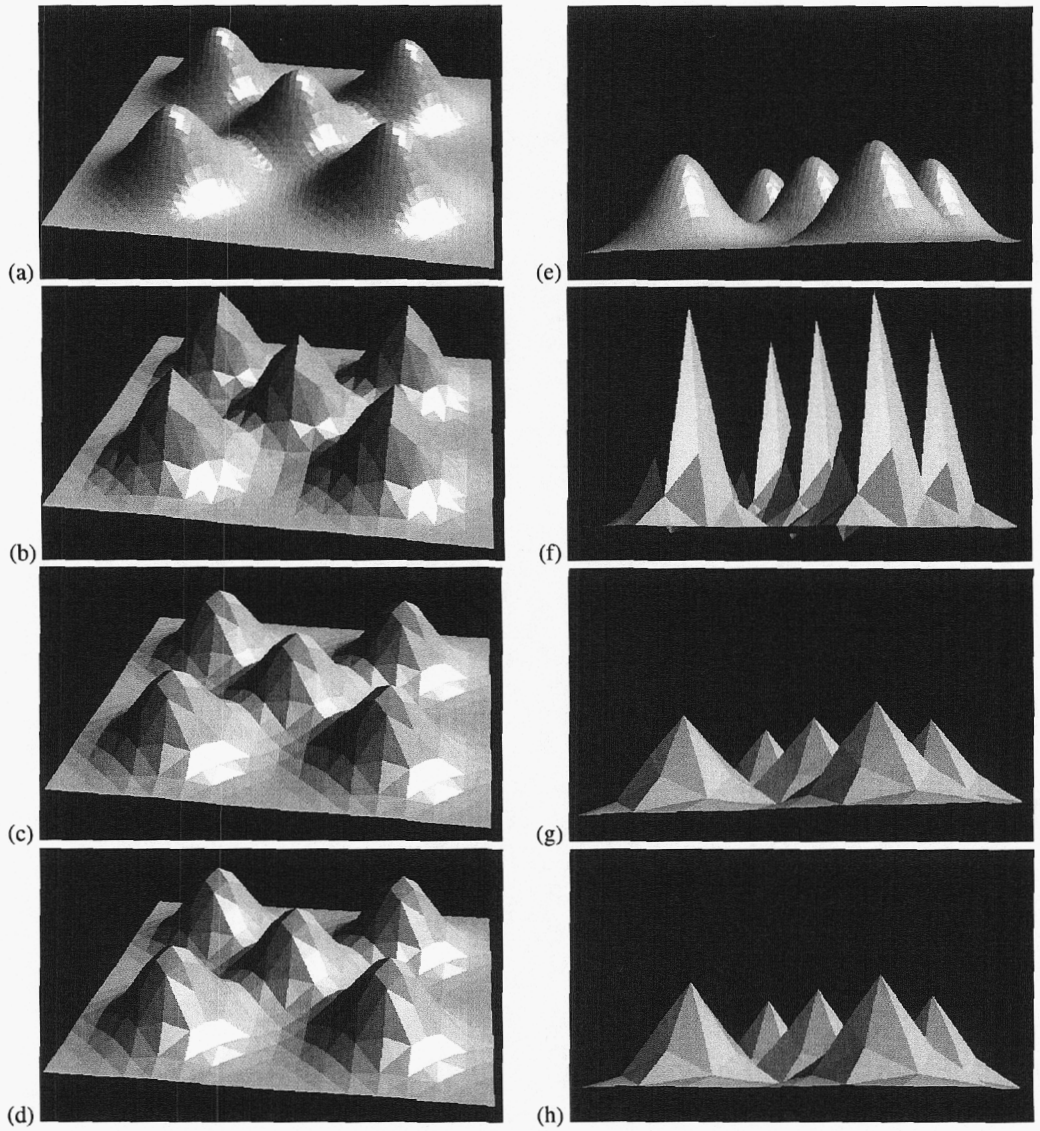
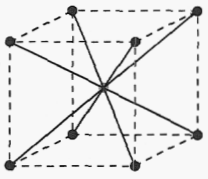


Figure 6: $\sqrt{2}$-subdivision surfaces, (a) and (e), encoded and decoded by $\sqrt{2}$-subdivision wavelets, (b) and (f), bilinear B-spline wavelets, (c) and (g), and bicubic B-spline wavelets, (d) and (h).

w-lift$_{encode}(a,b)$:  a^3 b^3 .

In the three-dimensional case, the masks are also as narrow as they can be.

The face case
When applying linear B-spline wavelet encoding to the situation depicted in the third picture of Figure 3, we have to make sure that we do not violate the assumptions made for the volume case. We assume that the values at the green vertices are bilinear interpolations of the values at the black neighbor vertices. Thus, when the values at the green vertices are available, their values should be computed only from the values at the black vertices. This insight leaves us with the two-dimensional case, and we can apply masks (13) – (16) of Section 4.

The edge case
When applying linear B-spline wavelet encoding to the situation illustrated in the fourth picture of Figure 3, we must not violate the assumption that the values at the blue vertices are linear interpolations of the values at the black neighbor vertices. When the values at the blue vertices are available, their values should be computed only from the values at the black vertices. This insight leaves us with the one-dimensional case, and we can apply masks (1) and (2) of Section 3.

It is a significant advantage of our scheme that the face and edge cases cover naturally boundary faces and boundary edges,

respectively.

7 3D Results

In Figure 7, we compare the results obtained by application of a $\sqrt[3]{2}$-subdivision multiresolution scheme, with and without trilinear B-spline wavelet encoding. The data set is a 256^3 uniform rectilinear grid, and at every vertex one scalar value between 0 and 255 is given. The data set represents a "bonsai tree solid." It was obtained by computer tomography. For the visualization of the bonsai tree, we extract and render the isosurface corresponding to the value 80, which is generated by the marching-tetrahedra algorithm described in [Guéziec and Hummel 1995].

Figures 7 (a) and (b) show the isosurface extracted from two different levels of detail of a $\sqrt[3]{2}$-subdivision hierarchy without using wavelets. Figures 7 (c) and (d) show the same isosurface extracted from the same levels of detail, where a $\sqrt[3]{2}$-subdivision hierarchy is combined with the trilinear B-spline wavelet scheme described in the previous section. Without the averaging steps of a wavelet encoding on the coarse level shown in 7(a), we seem to have three unconnected isosurface components for the chosen isovalue. When using the wavelets, we already have the correct topological information, which is shown in 7(c). Considering the finer levels shown in 7 (b) and (d), the resolution is not yet high enough to represent the finest details, like branches and twigs, but it is evident that the wavelet approach leads to better approximations.

To quantify the improvement in approximation quality, we compute an approximation error for each coarser level of approximation by comparing it to the original, highest resolution level. Given the original function F discretely by sample values at locations $\mathbf{x_i}$, $\mathbf{i} \in [1,n_x][1,n_y][1,n_z]$, we define the root-mean-square error as

$$\mathcal{E}_{RMS} = \sqrt{\frac{1}{n_x n_y n_z} \sum_{\mathbf{i}} \left(F(\mathbf{x_i}) - f(\mathbf{x_i})\right)^2},$$

where $f(\mathbf{x_i})$ is the approximated function value by trilinear interpolation applied to a "cell" in the coarser level of resolution: If f is defined at "corner" locations $\mathbf{y_j} = (y_{\mathbf{j},x}, y_{\mathbf{j},y}, y_{\mathbf{j},z})$, and if $\mathbf{x_i}$ is inside the interval $[y_{\mathbf{j},x}, y_{\mathbf{j}+\mathbf{e}_1,x})[y_{\mathbf{j},y}, y_{\mathbf{j}+\mathbf{e}_2,y})[y_{\mathbf{j},z}, y_{\mathbf{j}+\mathbf{e}_3,z})$, the approximated function value $f(\mathbf{x_i})$ results from trilinear interpolation of the eight corner values $f(\mathbf{y_j}), \ldots, f(\mathbf{y_{j+1}})$.

In Table 1, we list the root-mean-square errors of the shown examples at various levels of resolution. We scale the root-mean-square error to the interval $[0,1]$. We define the "downsampling ratio" as the original number of vertices divided by the number of vertices at the used coarser resolution. For all examples and all resolutions, we obtain smaller root-mean-square errors when using trilinear B-spline wavelets.

In Figure 8, we show a biomedical example. The data set represents a human brain. It is stored in 753 slices, and each slice has a resolution of 1050×970 points, where 24-bit RGB-color information is stored. The original data set was preprocessed with a segmentation algorithm described in [Takanashi et al. 2002] to eliminate noise and irrelevant data. We apply the wavelet scheme to each color channel independently and, after conversion, use the value V of the HSV color model for isosurface extraction.

Figure 8 shows an isosurface for the value 78 extracted from the level of detail with downsampling ratio 2^9. For Figure 8(a), we use a $\sqrt[3]{2}$-subdivision hierarchy without using wavelets, and, for Figure 8(b), we combine the $\sqrt[3]{2}$-subdivision hierarchy with trilinear B-spline wavelets. Figure 8(b) contains much more detail information than Figure 8(a).

In Figure 9, we apply our techniques to numerically simulated hydrodynamics data. The data set results from a threedimensional simulation of the Richtmyer-Meshkov instability and turbulent mixing in a shock tube experiment, see [?]. For each vertex of a 1024^3 structured-rectilinear grid (one time step considered only), a value between 0 and 255 for the entropy is stored. We show the isosurface corresponding to the value 225 extracted from three different levels of resolution of one time step. Again, we compare the results of the $\sqrt[3]{2}$-subdivision hierarchy without (left column) and with (right column) trilinear B-spline wavelets.

Considering the example shown in Figure 9, when using the wavelet approach low-resolution visualizations suffice to understand where the turbulent mixing takes place. For example, Figure 9(d) shows clearly the big "bubble" rising in the middle of data set. The bubble can hardly be seen in Figure 9(a).

8 Conclusions and future work

We have introduced $\sqrt[n]{2}$ subdivision combined with n-variate B-spline wavelets for n-dimensional multiresolution data representation. Since visualization of, for example, biomedical imaging data and numerically simulated hydrodynamics data require efficient extraction of many isosurfaces, a three-dimensional multiresolution framework is desirable. We first have established a bivariate B-spline wavelet scheme for $\sqrt{2}$ subdivision and then have generalized it to a trivariate B-spline wavelet scheme for $\sqrt[3]{2}$ subdivision. We have provided examples documenting the value of our approach for surface and volume modeling and visualization.

By using $\sqrt[n]{2}$ subdivision, instead of using quad- or octrees, a multiresolution hierarchy can be generated that provides much more levels of detail, since, in each subdivision step, the number of vertices is only doubled instead of multiplied by a factor of four or eight, respectively. In the context of view-dependent and adaptive refinement and visualization, this characteristic supports a higher level of adaptivity. Furthermore, $\sqrt[n]{2}$ subdivision does not only work for structured-rectilinear grids, but also for more general structured-curvilinear grids, and even for arbitrary grids, i. e., grids with extraordinary vertices.

By integrating a wavelet scheme into the subdivision approach, we obtain, in general, much better approximations on each level of detail. We have chosen n-variate B-spline wavelets and have developed lifting schemes for $n = 2$ and $n = 3$, which use narrow masks. These narrow masks allow us to apply the wavelet scheme for view-dependent, adaptive multiresolution visualization.

It is a well-known fact that wavelet encoding only "reorganizes" data and does not require additional memory. The $\sqrt[n]{2}$-subdivision scheme also does not require us to store additional connectivity information. Thus, our approach, as a whole, requires no additional storage.

Since the masks of our lifting scheme are of constant size and the number of iterations for our lifting scheme is constant, our algorithms run in linear time with respect to the number of original data. Since the masks are narrow and only two iterations are needed (in the linear case), the run-time constants are small.

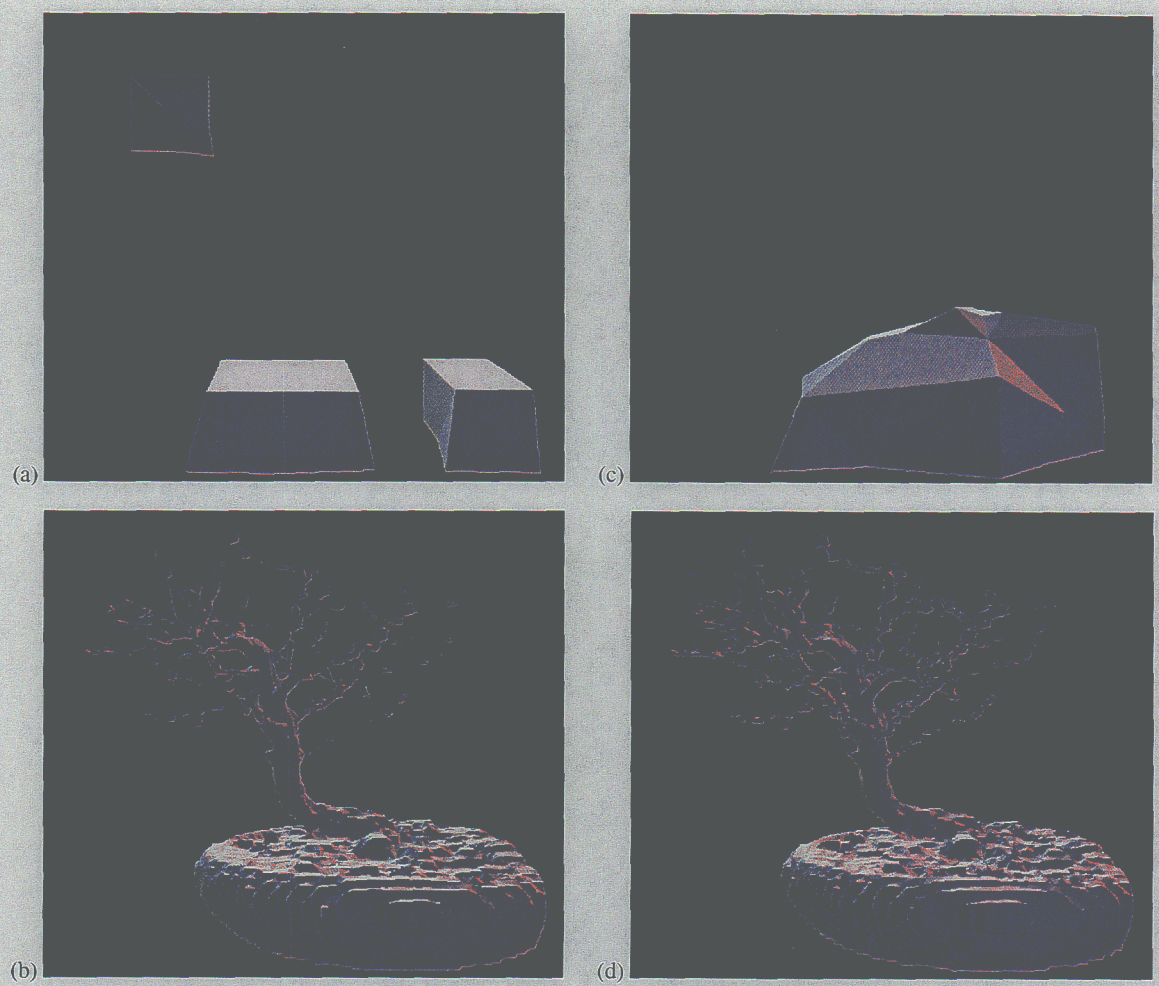


Figure 7: Comparing $\sqrt[3]{2}$-subdivision hierarchy without, (a) and (b), and with, (c) and (d), trilinear B-spline wavelets. Shown is the same isosurface extracted from the levels of detail with downsampling ratios 2^6 and 2^{15}. (Data set courtesy of S. Roettger, Abteilung Visualisierung und Interaktive Systeme, University of Stuttgart, Germany)

Considering the provided examples, we conclude that our approach provides a valuable tool for the interactive exploration of volumetric data at multiple level of resolution.

Acknowledgments

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the National Institute of Mental Health and the National Science Foundation under contract NIMH 2 P20 MH60975-06A2; the Army Research Office under contract ARO 36598-MA-RIP; and the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159. We also acknowledge the support of ALSTOM Schilling Robotics and SGI. We thank the members of the Visualization and Graphics Research Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis. We especially thank Peer-Timo Bremer for supplying us with an implementation of the $\sqrt{2}$-subdivision wavelets.

References

BERTRAM, M., DUCHAINEAU, M. A., HAMANN, B., AND JOY, K. I. 2000. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In *Proceedings of IEEE Conference on Visualization 2000*, IEEE Computer Society Press, T. Ertl, B. Hamann, and A. Varshney, Eds., IEEE, 389–396.

BERTRAM, M., LANEY, D. E., DUCHAINEAU, M. A., HANSEN, C. D., HAMANN, B., AND JOY, K. I. 2001. Wavelet representation of contour sets. In *Proceedings of IEEE Conference on Visualization 2001*, IEEE Computer Society Press, T. Ertl, K. Joy, and A. Varshney, Eds., IEEE, 303–310.

BERTRAM, M. 2000. *Multiresolution Modeling for Scientific Visualization*. PhD thesis, University of California - Davis.

GUÉZIEC, A., AND HUMMEL, R. 1995. Exploiting triangulated surface extration using tetrahedral decomposition. *IEEE Transactions on Visualization and Computer Graphics 1*, 4, 328–342.

$\mathcal{E}_{RMS}$	Figure 7		Figure 8		Figure 9	
downsampling ratio	w/o wavelets	wavelets	w/o wavelets	wavelets	w/o wavelets	wavelets
2^3	0.0179	0.0159	0.0284	0.0258	0.0177	0.0157
2^6	0.0363	0.0313	0.0384	0.0346	0.0429	0.0390
2^9	0.0602	0.0505	0.0507	0.0461	0.0760	0.0698
2^{12}	0.0899	0.0751	0.0684	0.0625	0.1094	0.0990
2^{15}	0.1226	0.0964	0.0923	0.0872	0.1435	0.1271

Table 1: Root-mean-square errors for three examples at different levels of resolution without (w/o) and with trilinear B-spline wavelets.

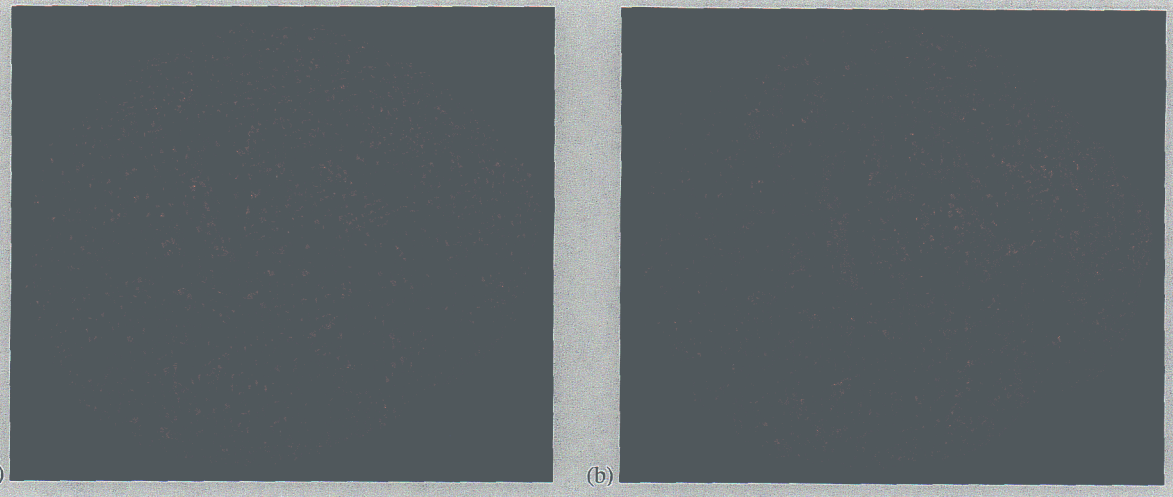


Figure 8: Hierarchical visualization of brain data set, based on $\sqrt[3]{2}$-subdivision without (a) and with (b) B-spline wavelets. (Data set courtesy of A. Toga, Ahmanson-Lovelace Brain Mapping Center, University of California - Los Angeles)

KOBBELT, L. 2000. $\sqrt{3}$-subdivision. In *Proceedings of SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, K. Akeley, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 103–112.

KOBBELT, L. 2002. Multiresolution techniques. In *Handbook of Computer Aided Geometric Design*, Elsevier Science Publishing, Amsterdam, The Netherlands, Farin, Hoschek, and Kim, Eds., to appear.

MACCRACKEN, R. A., AND JOY, K. I. 1996. Free-form deformations with lattices of arbitrary topology. In *Proceedings of SIGGRAPH 1996*, ACM Press / ACM SIGGRAPH, H. Rushmeier, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 181–188.

MAUBACH, J. M. 1995. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM J. Scientific Computing 16*, 210–227.

PASCUCCI, V. 2001. Slow-growing subdivisions in any dimension: Towards removing the curse of dimensionality. *Lawrence Livermore National Laboratory technical report UCRL-ID-144257*.

PINSKIY, D., BRUGGER, E., CHILDS, H., AND HAMANN, B. 2001. An octree-based multiresolution approach supporting interactive rendering of very large volume data sets. In *Proceedings of the 2001 International Conference on Imaging Science, Systems, and Technology (CISST 2001), Volume 1*, Computer Science Research, Education, and Applications Press (CSREA), Athens, Georgia, H. Arabnia, R. Erbacher, X. He, C. Knight, B. Kovalerchuk, M. Lee, Y. Mun, M. Sarfraz, J. Schwing, and H. Tabrizi, Eds., 16–22.

PRAUTZSCH, H., BOEHM, W., AND PALUSZNY, M. 2002. *Bézier and B-spline Techniques*. Springer-Verlag, Heidelberg, Germany, to appear.

STOLLNITZ, E. J., DEROSE, T. D., AND SALESIN, D. H. 1996. *Wavelets for Computer Graphics: Theory and Applications*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Brian A. Barsky (series editor), Morgan Kaufmann Publishers, San Francisco, U.S.A.

TAKANASHI, I., LUM, E., MA, K.-L., MEYER, J., HAMANN, B., AND OLSON, A. J. 2002. Segmentation and 3d visualization of high-resolution human brain cryosections. In *Proceedings of SPIE Visualization and Data Analysis 2002*, to appear, SPIE.

VELHO, L., AND ZORIN, D. 2001. 4-8 subdivision. *Computer-Aided Geometric Design 18*, 5, 397–427.

WARREN, J., AND WEIMER, H. 2002. *Subdivision Methods for Geometric Design*. Morgan Kaufmann Publishers, San Francisco, U.S.A.

ZHOU, Y., CHEN, B., AND KAUFMAN, A. E. 1997. Multiresolution tetrahedral framework for visualizing regular volume data. In *Proceedings of IEEE Conference on Visualization 1997*, IEEE Computer Society Press, R. Yagel and H. Hagen, Eds., IEEE, 135–142.

Figure 9: Entropy in a threedimensional simulation of the Richtmyer-Meshkov instability, visualized by isosurface extraction from a $\sqrt[3]{2}$-subdivision hierarchy without (left column) and with (right column) B-spline wavelets at the resolutions with downsampling ratios 2^9, 2^{12}, and 2^{15}.