

# UC Davis

## IDAV Publications

### Title

Wavelet-based Multiresolution with  $n$ th-root-of-2 Subdivision

### Permalink

<https://escholarship.org/uc/item/5d21c5dt>

### Journal

Journal on Computing, special edition

### Authors

Linsen, Lars  
Pascucci, Valerio  
Duchaineau, Mark A.  
et al.

### Publication Date

2004

Peer reviewed

# Wavelet-based Multiresolution with $\sqrt[n]{2}$ Subdivision

Lars Linsen<sup>1</sup>, Valerio Pascucci<sup>2</sup>, Mark A. Duchaineau<sup>2</sup>, Bernd Hamann<sup>1</sup>,  
and Kenneth I. Joy<sup>1</sup>

<sup>1</sup> Center for Image Processing and Integrated Computing (CIPIC)

Department of Computer Science

University of California, Davis

Davis, CA 95616-8562

<http://graphics.cs.ucdavis.edu/>

llinsen@ucdavis.edu, {hamann, joy}@cs.ucdavis.edu

<sup>2</sup> Center for Applied Scientific Computing (CASC)

Data Science Group

Lawrence Livermore National Laboratory

Livermore, CA 94550-9234

<http://www.llnl.gov/casc/>

{pascucci1, duchaineau1}@llnl.gov

**Summary.** Multiresolution methods are a common technique used for dealing with large-scale data and representing it at multiple levels of detail. We present a multiresolution hierarchy construction based on  $\sqrt[n]{2}$  subdivision, which has all the advantages of a regular data organization scheme while reducing the drawback of coarse granularity. The  $\sqrt[n]{2}$ -subdivision scheme only doubles the number of vertices in each subdivision step regardless of dimension  $n$ . We describe the construction of 2D, 3D, and 4D hierarchies representing surfaces, volume data, and time-varying volume data, respectively. The 4D approach supports spatial and temporal scalability. For high-quality data approximation on each level of detail, we use downsampling filters based on  $n$ -variate B-spline wavelets. We present a B-spline wavelet lifting scheme for  $\sqrt[n]{2}$ -subdivision steps to obtain small or narrow filters. Narrow filters support adaptive refinement and out-of-core data exploration techniques.

## 1 Introduction

Due to substantial improvements in computing power and storage capacity over the last decade, today's data-intensive applications are generating huge amounts of data in shorter and shorter time frames. Downsampling can be used to reduce the data to a manageable amount. The reduced data can be examined by scientists to identify regions of interest, for which more detailed exploration should be applied. Rendering and visualization applications must deal with large data sets and their representation at multiple levels of detail.

Multiresolution methods for curves and surfaces are widely used in computer graphics [13]. Representing volume data hierarchically is important in

the context of “volume modeling” and visualization of volume data, e.g., of scalar or vector fields defined over volumetric domains. Visualizing inherently trivariate phenomena requires one to apply rendering algorithms to volumetric data - examples being volume slicing via a cutting plane, isosurface extraction through marching-cubes-like algorithms, and ray casting. For time-varying volume data, multiresolution methods have to scale in spatial and temporal dimensions.

Multiresolution methods can be classified according to (regular and irregular) data formats and refinement rules. Regular refinement schemes’ main advantages over irregular refinement schemes are that grid connectivity and vertex locations are implicitly defined and data can be easily accessed, which is of particular importance for large-scale data. The main disadvantage of regular refinement schemes is their coarse granularity and thus low adaptivity. For example, in one quadtree or octree refinement step the number of vertices is multiplied by a factor of four or eight, respectively. We use the  $\sqrt[3]{2}$ -subdivision scheme that only doubles the number of vertices in each subdivision step regardless of dimension, which is a factor of  $\sqrt[3]{2}$  in each of the  $n$  dimensions. In Section 3, we describe the  $\sqrt[3]{2}$ -subdivision scheme in general and provide more detail for up to four dimensions.

Another drawback of using regular data structures is that downsampling is done based purely on grid structure, without considering data values. Therefore, aliasing artifacts occur and scientifically interesting details in a data set can get lost and be overlooked. To avoid this effect, we use a downsampling filter based on  $n$ -variate B-spline wavelets: The data value at a vertex  $\mathbf{p}$  is updated when changing the level of detail, i. e., the value varies with varying level of detail. On a coarse level, the value represents the value at  $\mathbf{p}$  itself and an average value of a certain region around  $\mathbf{p}$ . This “filtering” removes the aliasing artifacts and leads to better approximation quality on coarser levels.

Non-constant B-spline wavelets have the property that the computation of the wavelet coefficient at a vertex  $\mathbf{p}$  is not only based on the neighbors of  $\mathbf{p}$  but also on vertices that are farther away. Such large filters reduce the adaptivity of the multiresolution representation. Moreover, when using out-of-core techniques to operate on or visualize large-scale data, substantial amounts of data must be loaded from external memory with typically low I/O-performance for applying such large filters. Lifting schemes with narrow filters can be used to overcome this problem. In Section 4, we describe a one-dimensional lifting scheme for B-spline wavelets applicable to multiresolution polygonal representations of curves.

In Sections 5, 7, and 9, we describe how this approach can be generalized to the 2D, 3D, and 4D settings of multiresolution representations generated by  $\sqrt[3]{2}$  subdivision (with  $n = 2, 3, 4$ ). In Sections 6, 8, and 10, we apply these techniques to surfaces, volume data, and time-varying volume data, respectively. We provide several examples and visualize data using standard

visualization methods such as isosurface extraction, cutting planes, and volume rendering.

## 2 Related work

Multiresolution volume representation is based on a hierarchical data organization of irregular or regular type. Irregular data structures use non-uniform refinement steps, which makes them highly adaptive. Unfortunately, grid information must be stored and data access is not straight forward. Especially for large-scale data, additional memory requirements and memory organization needs are major drawbacks of irregular structures. For regular structures, grid connectivity and vertex locations are implicit and data is easily accessed. However, the refinement steps must conform to the topological constraints, which makes regular structures less adaptive.

Multiresolution representations for curves and surfaces have been subject of research for many years. An overview of existing techniques (including regular as well as irregular schemes) is given in [13]. In terms of 3D structures, irregular schemes are discussed in [5, 10, 9, 34] and regular schemes in [18, 19, 22, 26, 29, 43]. A discussion of regular versus irregular schemes is given in [33].

Since the advantages of regular schemes are very important for visualization applications, the  $\sqrt[3]{2}$ -subdivision scheme we describe is of regular type. To our knowledge, the  $\sqrt[3]{2}$ -subdivision scheme is a regular scheme with finest granularity, doubling the number of vertices in each subdivision step regardless of dimension. In the 3D case, this fact implies that the  $\sqrt[3]{2}$ -subdivision scheme, in general, will require less vertices than, for example, octrees to satisfy error bounds. Since finer granularity leads to higher adaptivity, this fact still holds when using adaptive refinement techniques. Our 3D approach has been described in [16].

For time-varying volume representation, sophisticated approaches make use of the data's temporal coherence and focus on the detection of spatial/structural changes and updates in time [1, 30, 32, 36]. These approaches consider scaling in time but not in space. However, applications in large-scale time-varying volume visualization using isosurface extraction [11, 30, 36] or volume rendering [1, 32] (a comparison is drawn in [39]) requires us to utilize multiresolution representations with scalability in time and space. An approach dealing with large-scale data in time and space was described by Shen et al. [31]. Their approach combines an octree with a binary tree to a Time-Space Partition (TSP) tree, where the octree is used for the spatial and the binary tree for the temporal hierarchy.

By using a  $\sqrt[3]{2}$ -subdivision hierarchy for representing time-varying volume data, we treat time as a real fourth dimension. Compared to the TSP approach, we have finer granularity, which is especially desirable in the spa-

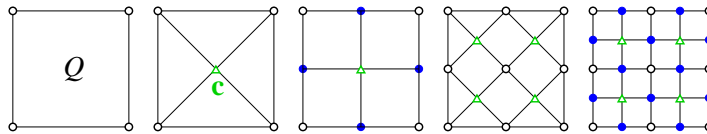
tial dimension. Our framework deals with spatial and temporal dimensions equally. Our 4D approach has been described in [17].

### 3 Multiresolution with $\sqrt[n]{2}$ subdivision

A multiresolution hierarchy based on  $\sqrt[n]{2}$  subdivision is constructed by starting with the coarsest resolution of a given mesh and iteratively applying  $\sqrt[n]{2}$ -subdivision steps. The subdivision steps are performed simultaneously for all mesh elements.

The splitting step of the  $\sqrt[n]{2}$ -subdivision scheme can be described by using triangular or quadrilateral meshes (for  $n = 2$ ) or their generalizations to higher dimensions (for  $n > 2$ ). In the 3D case, the splitting step of the  $\sqrt[3]{2}$ -subdivision scheme is equivalent to the longest-edge bisection of tetrahedral meshes [7, 24, 44]. We consider quadrilateral meshes and their generalizations, which makes the derivation and the description of the techniques more comprehensible. For implementation purposes, we have used triangular meshes and their generalizations, since they support application of existing visualization tools.

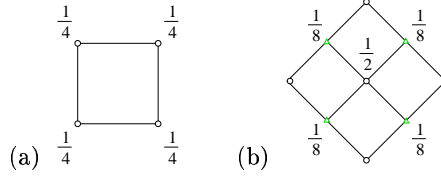
The splitting step of the  $\sqrt[n]{2}$ -subdivision scheme were described by Cohen and Daubechies [6] for  $n = 2$  and Maubach [20] for arbitrary  $n$ . Figure 1 illustrates four splitting steps of a  $\sqrt{2}$  subdivision ( $n = 2$ ). To split the quadrilateral  $Q$ , we compute its centroid  $\mathbf{c}$  and connect  $\mathbf{c}$  to the four vertices of  $Q$ . The “old” edges of the mesh are removed (except for the edges determining the mesh/domain boundary).



**Fig. 1.**  $\sqrt{2}$  subdivision.

Velho and Zorin [41] completed the  $\sqrt{2}$ -subdivision scheme by adding an averaging step to the splitting step. Figure 2 shows the masks for the splitting (a) and the averaging step (b). A  $\sqrt{2}$ -subdivision step is executed by first applying the mask shown in Figure 2(a), which inserts the centroids  $\mathbf{c}$  as new vertices, and then (after topological mesh modifications) applying the mask shown in Figure 2(b), which adjusts the old vertices.

Velho and Zorin showed that the produced surfaces are  $C^4$ -continuous at regular and  $C^1$ -continuous at extraordinary vertices. This subdivision scheme for quadrilaterals is analogous to the  $\sqrt{3}$ -subdivision scheme of Kobbelt for triangles [12]. Therefore, we call it  $\sqrt{2}$  subdivision. (For an introduction to subdivision methods, we refer to [42].)



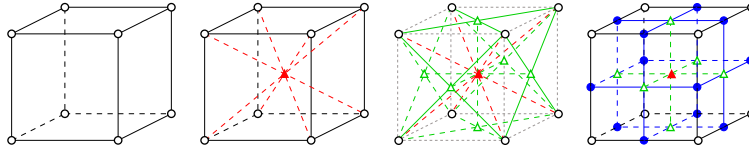
**Fig. 2.** Masks of  $\sqrt{2}$ -subdivision step: (a) inserting centroid; (b) adjusting old vertices.

This subdivision scheme can be generalized to arbitrary dimension. The splitting step of the  $\sqrt[3]{2}$  subdivision is executed by inserting the centroid of the  $n$ -dimensional geometrical shapes and adjusting vertex connectivity. The averaging step applies to every old vertex  $\mathbf{v}$  the update rule

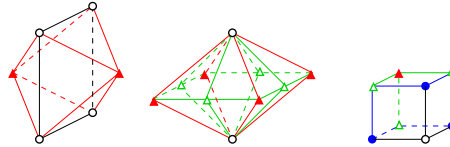
$$\mathbf{v} = \alpha \mathbf{v} + (1 - \alpha) \mathbf{w} ,$$

where  $\mathbf{w}$  is the centroid of the adjacent new vertices and  $\alpha \in [0, 1]$ . See [23] for further details.

Figure 3 shows three  $\sqrt[3]{2}$ -subdivision splitting steps ( $\alpha = 1$ ) for structured rectilinear volume data. Three kinds of polyhedral shapes arise, shown in Figure 4.



**Fig. 3.**  $\sqrt[3]{2}$  subdivision.

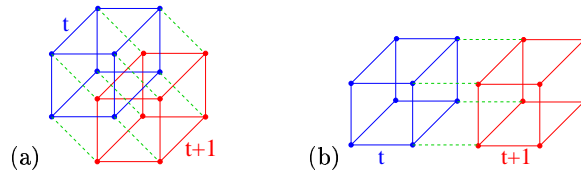


**Fig. 4.** Polyhedral shapes created by  $\sqrt[3]{2}$  subdivision: octahedron, octahedron with split faces, and cuboid.

With respect to the start configuration (first picture of Figure 3), the three subdivision steps can be described in the following way: The first step inserts the centroid of the cuboid (second picture of Figure 3), the second step inserts the centers of the faces of the original cuboid (third picture of Figure 3), and the third step inserts the midpoints of the edges of the original cuboid (fourth picture of Figure 3).

Thus, three  $\sqrt[3]{2}$ -subdivision steps produce the same result as one octree refinement step. The two steps in between provide finer granularity. For visualization purposes, this leads to higher adaptivity in error-bounded, view-dependent settings. Therefore, it is likely that one must render less data to obtain a desired image / visualization quality.

In order to generate a four-dimensional hierarchy, we start with a hypercube (or hypercuboid), depicted in Figure 5. Figure 5(a) shows its symmetry in all four dimensions. In the following, we use illustrations as in Figure 5(b), where the hypercube is stretched in one dimension. When considering time-varying volume data, this dimension will represent the temporal direction.



**Fig. 5.** Hypercube.

In Figure 6, four  $\sqrt[4]{2}$ -subdivision steps ( $\alpha = 1$ ) are shown. We only show the spatial connectivities within the time steps and omit the connectivity information between time steps.

Figure 6(a) shows the initial hypercuboid, which consists of two cuboids at two time steps,  $t_1$  and  $t_3$ . The two cuboids are connected according to Figure 5(b). The first subdivision step inserts the centroid of the hypercuboid, shown in Figure 6(b), which can be interpreted as the centroid of a cuboid at time step  $t_2 = \frac{t_1+t_3}{2}$ . The second subdivision step inserts the centroids of the eight cuboids within the original hypercuboid, shown in Figure 6(c). The third step inserts the centers of the faces of these eight cuboids or of the original hypercuboid, respectively, shown in Figure 6(d). Finally, the fourth step inserts the midpoints of the edges of the eight cuboids or of the original hypercuboid, respectively, shown in Figure 6(e). The geometric structure shown in Figure 6(e) consists of 16 hypercuboids.

## 4 Lifting of B-spline wavelets

When downsampling time-varying volume data in a regular fashion, data is not grouped due to changes in time or space. Thus, aliasing artifacts occur and important details may be missing on coarse levels of resolution. We overcome this problem by using downsampling filters. In image processing, such downsampling filters are commonly employed with wavelets. Stollnitz et al. [35] described how to generate wavelets for subdivision schemes.

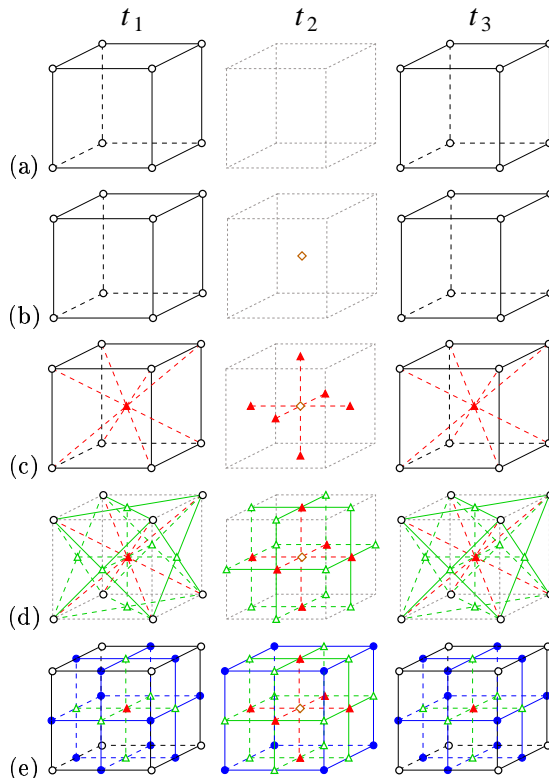


Fig. 6.  $\sqrt[3]{2}$  subdivision.

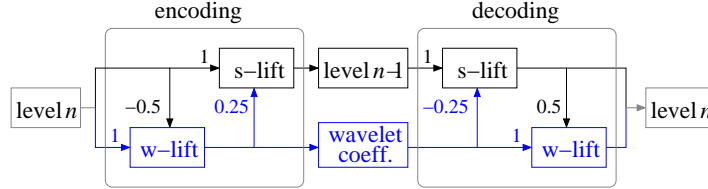
A family of filters can be derived by using B-splines of various degrees for wavelet generation. (For an introduction to B-spline techniques, we refer to [27].) However, when using non-constant B-splines, the size of the wavelet filters is not limited to adjacent vertices. Localization is desirable when we want to apply the wavelet filter to adaptive refinement and out-of-core visualization techniques. Lifting schemes as introduced by Sweldens [37] decompose wavelet computations into several steps, but they assert narrow filters.

We review the idea of lifting in general and lifting of B-spline wavelets in particular. These concepts are then used to generate narrow wavelet filters for  $\sqrt[3]{2}$ -subdivision hierarchies.

The idea of a lifting scheme is shown in Figure 7, using the example of linear B-spline wavelets. For downsampling, the vertices of a level of resolution  $\mathcal{L}_n$  are split into two groups: the ones that belong to the next coarser level of resolution  $\mathcal{L}_{n-1}$  (often referred to as *even* vertices) and the ones that belong to  $\mathcal{L}_n \setminus \mathcal{L}_{n-1}$  (often referred to as *odd* vertices). Instead of applying a large downsampling filter to the vertices  $\in \mathcal{L}_{n-1}$ , the lifting scheme decomposes the large filter into two narrow ones and executes two steps. First, one



narrow filter (*w-lift*) is applied to the vertices  $\in \mathcal{L}_n \setminus \mathcal{L}_{n-1}$ . Second, the other narrow filter (*s-lift*) is applied to the vertices  $\in \mathcal{L}_{n-1}$ . This process is usually referred to as *encoding*, and the values at the vertices  $\in \mathcal{L}_n \setminus \mathcal{L}_{n-1}$  are called *wavelet coefficients*. The *decoding* step inverts the two encoding steps and reconstructs level  $\mathcal{L}_n$  from level  $\mathcal{L}_{n-1}$  using the wavelet coefficients.



**Fig. 7.** One-dimensional linear B-spline wavelet lifting scheme.

The lifting filters can be described by masks. For example, the one-dimensional B-spline wavelet lifting filters are given by:

$$\text{s-lift}(a, b): (a \ b \ a) , \quad (1)$$

$$\text{w-lift}(a, b): (a \ b \ a) . \quad (2)$$

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

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

A linear B-spline wavelet decoding step is defined by executing the inverse operations in reverse order. They are

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

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

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

A cubic B-spline wavelet decoding step is again defined by the inverse operations in reverse order.

For a detailed derivation of the B-spline lifting scheme that we use, as well as for its analysis (smoothness, stability, approximation order, and zero moments), we refer to [2, 3, 4].

## 5 Lifting for $\sqrt{2}$ -subdivision hierarchies

The 1D filters described in the previous section for polygons can be generalized to 2D filters for meshes representing tensor-product surfaces by convolution of the 1D masks in the two coordinate directions, e.g.,

$$(a \ b \ a) * \begin{pmatrix} a \\ b \\ a \end{pmatrix} = \begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}. \quad (3)$$

A mesh hierarchy for tensor-product surfaces can be generated using quadtree refinement. By using  $\sqrt{2}$  subdivision instead of a quadtree-based scheme, we have an additional level of resolution (see second picture in Figure 1). For this additional level, we only insert new vertices at the centers  $\triangle$  of old faces; at the midpoints  $\bullet$  of old edges, vertices are not inserted before the subsequent subdivision step (see third picture in Figure 1). Thus, in order to apply the wavelet lifting scheme to a  $\sqrt{2}$ -subdivision hierarchy, we have to adjust the mask (3) to the setting shown in the second picture of Figure 1.

Wavelets for general dilation matrices were discussed by Riemenschneider and Shen [28] who used a box-spline approach. Kovačević and Vetterli [15] and, more recently, Uytterhoeven [40] and Kovačević and Sweldens [14] developed lifting schemes that can be applied to  $\sqrt[3]{2}$ -subdivision mesh hierarchies. Uytterhoeven's method only addresses the two-dimensional case, Kovačević and Sweldens' approach deals with the two- and three-dimensional cases. The filters used in [14] (that do not degenerate to the identity) are not narrow enough for our purposes. We derive narrow B-spline wavelet lifting filters for  $\sqrt{2}$ -subdivision hierarchies and later describe their generalization.

For encoding with linear B-spline wavelets, the w-lift operation is executed first. In a  $\sqrt{2}$ -subdivision hierarchy, we have no data values available at the positions  $\bullet$  (see third picture of Figure 1). Since the mask (3) requires data at the positions  $\bullet$ , we compute them by linearly interpolating the values at the vertices  $\circ$ . Linear interpolation is appropriate, since we are using linear wavelets. This approach changes mask (3) to

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

Next, the s-lift operation is executed. Again, we have to determine data values at the positions  $\bullet$ . However, the w-lift operation has (theoretically) executed 1D masks to update the values at the positions  $\bullet$ . Since we assumed that the values at the vertices  $\bullet$  were linear interpolations of the values at the vertices  $\circ$ , the values at the vertices  $\bullet$  vanish by executing the 1D masks. The mask for the s-lift encoding step becomes

$$\text{s-lift}_{\text{encode}}(a, b) : \begin{pmatrix} a^2 & a^2 \\ & b^2 \\ a^2 & a^2 \end{pmatrix}. \quad (5)$$

For decoding with linear B-spline wavelets, we first execute the s-lift operation. Prior to executing the s-lift encoding operation, the values at the vertices  $\bullet$  have vanished, but the s-lift encoding operation (theoretically) executed the 1D mask to update the vertices  $\bullet$ . Hence, the values at the vertices  $\bullet$  are given by linear interpolation of the values at the neighbor vertices  $\Delta$ , multiplied by the factor  $2a$  of the 1D mask. We rename the factor  $a$  to  $\bar{a}$  and obtain the new mask

$$\text{s-lift}_{\text{decode}}(a, b) : \begin{pmatrix} a^2 + 2\bar{a}ab & a^2 + 2\bar{a}ab \\ & b^2 \\ a^2 + 2\bar{a}ab & a^2 + 2\bar{a}ab \end{pmatrix}. \quad (6)$$

Finally, the w-lift operation is executed again. The s-lift decoding operation has (theoretically) applied a 1D mask again to update the vertices  $\bullet$ . Since the 1D s-lift decoding mask is the inverse of 1D s-lift encoding mask, the values at the vertices  $\bullet$  are the same as before the execution of these two s-lift operations, i. e., they vanish, leading to the new mask

$$\text{w-lift}_{\text{decode}}(a, b) : \begin{pmatrix} a^2 & a^2 \\ & b^2 \\ a^2 & a^2 \end{pmatrix}. \quad (7)$$

Considering cubic B-spline wavelets, the masks can be derived similarly. The cubic B-spline wavelet lifting scheme executes one more s-lift operation. For example, for encoding one can derive the first s-lift mask

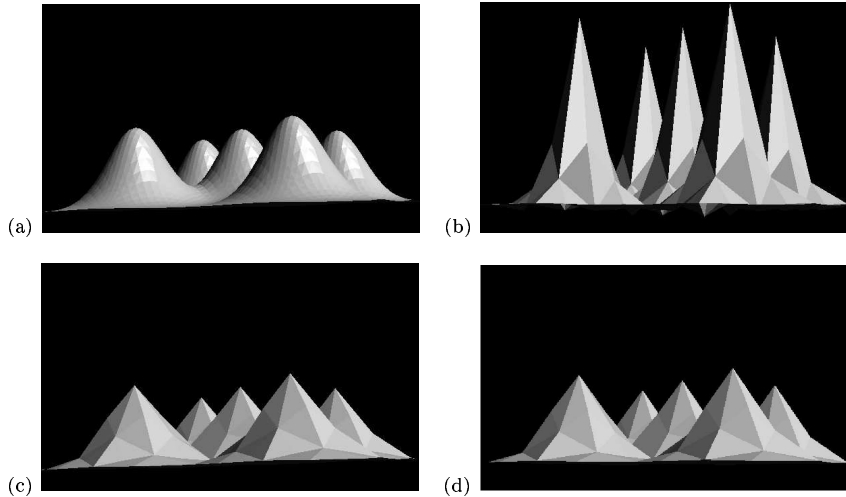
$$\text{s-lift}_{\text{encode1}}(a, b) : \begin{pmatrix} & & \frac{ab}{2} \\ & a^2 & a^2 \\ \frac{ab}{2} & b^2 + 2ab & \frac{ab}{2} \\ & a^2 & a^2 \\ & & \frac{ab}{2} \end{pmatrix},$$

while the other two encoding masks are the same as the linear encoding masks (4) and (5).

## 6 Surfaces

In Figure 8, we provide an example for a  $\sqrt{2}$ -subdivision hierarchy combined with 2D wavelet filters. The original surface shown in Figure 8(a) results from sampling a 2D Gaussian function at  $64^2$  vertices. The surface is encoded and decoded again. In Figure 8(b), we show a coarse level of detail obtained by applying  $\sqrt{2}$ -subdivision wavelets filters. In Figure 8(c) and (d), we show the

same level of detail obtained when establishing the  $\sqrt{2}$ -subdivision hierarchy using downsampling filters based on bilinear and bicubic B-spline wavelets, respectively.



**Fig. 8.** (a)  $\sqrt{2}$ -subdivision surfaces; (b) encoded and decoded by  $\sqrt{2}$ -subdivision wavelets; (c) by bilinear B-spline wavelets; and (d) by bicubic B-spline wavelets.

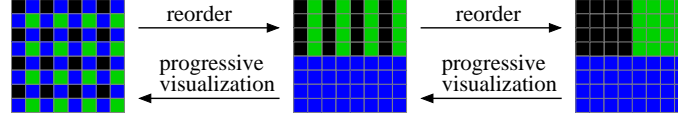
In Figure 8(b), obvious over- and undershoots caused by the  $\sqrt{2}$ -subdivision wavelet filters can be recognized. Over- and undershoots are disturbing during visualization, for example, when extracting isocontours from different levels of approximation. They can cause changes of contour topology when changing the level of resolution. We would like to preserve topology as much as possible when changing approximation levels.

In Figure 8(d), less pronounced over- and undershoot can be seen when using bicubic B-spline wavelet filters. Only bilinear B-spline wavelet filters ensure no over- and undershoots, see Figure 8(c), since linear B-spline wavelets have interpolating scaling functions, which guarantees interpolating refinement filters [14].

Moreover, the last section showed that lifting filters for linear B-spline wavelet were as narrow as they can be, whereas some of the lifting filters for cubic B-spline wavelets are larger. These drawbacks of cubic B-spline wavelets led to the decision to focus on linear B-spline wavelets when generalizing the schemes to higher dimensions.

For progressive visualization, i. e. when generating images progressively, the storage of values can be reorganized as shown in Figure 9 for three successive levels of resolution. Progressive visualization starts by using the upper-left block in the right picture, then adding the upper-right block, and, finally, adding the lower block. Reordering ensures that data can be read in a con-

tinuous stream without reading data multiple times. Moreover, the figure illustrates that the wavelet scheme does not require additional storage.



**Fig. 9.** Reordering data for progressive visualization.

## 7 Lifting for $\sqrt[3]{2}$ -subdivision hierarchies

In a  $\sqrt[3]{2}$ -subdivision hierarchy, three different kinds of polygonal shapes appear, see Figure 4. Therefore, three different kinds of masks have to be defined for the lifting filters. For deriving these masks based on trilinear B-spline wavelets, we start with the situation shown in the second picture of Figure 3 (*volume case*), proceed with the situation shown in the third picture of Figure 3 (*face case*), and finally treat the situation shown in the fourth picture of Figure 3 (*edge case*), which is topologically equivalent to the situation shown in the first picture of Figure 3.

**Volume case.** A convolution of 1D masks in the three coordinate directions leads to a generalization of mask (3) to a 3D mask, which can be used for mesh hierarchies based on octree refinement. In the situation shown in the second picture of Figure 3, we have no data values available at the vertices  $\bullet$  and  $\triangle$  (see fourth picture of Figure 3).

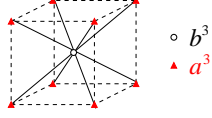
Again, we assume that the value at a vertex  $\bullet$  is defined by linear interpolation of the values at the two vertices  $\circ$  (with which the vertex  $\bullet$  shares an edge), and that the value at a vertex  $\triangle$  is defined by bilinear interpolation of the values at the four vertices  $\circ$  (with which the vertex  $\triangle$  shares a face). One obtains the mask  $w\text{-lift}_{\text{encode}}(a, b)$ , which can be depicted as

$$\circ \quad a^3 + \frac{3}{2}a^2b + \frac{3}{4}ab^2$$

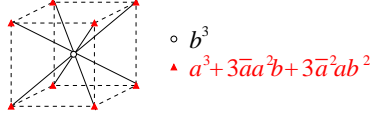
$$\triangle \quad b^3$$

The 1D and 2D masks for updating the vertices  $\bullet$  and  $\triangle$  are only “applied theoretically.” However, since the values at the vertices  $\bullet$  are assumed to be linear interpolations of the values at the vertices  $\circ$ , and since the values at the vertices  $\triangle$  are assumed to be bilinear interpolations of the values at the vertices  $\circ$ , the values at the vertices  $\bullet$  and  $\triangle$  vanish.

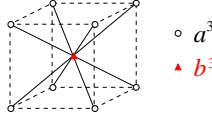
Therefore, the mask for the s-lift encoding operation reduces to the mask  $\text{s-lift}_{\text{encode}}(a, b)$ :



For the decoding, we can proceed in the same way (and analogously to the 2D case) and derive the mask  $\text{s-lift}_{\text{decode}}(a, b)$ :



where  $\bar{a}$  is the parameter  $a$  from the s-lift encoding mask, and the mask  $\text{w-lift}_{\text{decode}}(a, b)$ :



These 3D masks are as narrow as they can be.

**Face case.** In the next  $\sqrt[3]{2}$ -subdivision step, we have to deal with the situation shown in the third picture of Figure 3. We have to ensure that we do not violate the assumptions made for the volume case. We assumed that the values at the vertices  $\Delta$  are bilinear interpolations of the values at the neighbor vertices  $\circ$ . Thus, when the values at the vertices  $\Delta$  are available, their values should be computed only from the values at the vertices  $\circ$ . This insight leaves us with the 2D case, and we can apply the 2D masks of Section 5.

**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 vertices  $\bullet$  are linear interpolations of the values at the neighbor vertices  $\circ$ . When the values at the vertices  $\bullet$  are available, their values must be computed only from the values at the vertices  $\circ$ . This fact leaves us with the one-dimensional case, and we can apply the 1D masks of Section 4.

While in [14] and [40] the boundary cases for the non-separable filters are not addressed sufficiently, the face and edge cases of our scheme cover naturally boundary faces and boundary edges of the domain.

## 8 Volume data

We use our techniques for scientific visualization of volume data. We compare the results obtained by applying a  $\sqrt[3]{2}$ -subdivision multiresolution scheme with and without applying trilinear B-spline wavelet filters for downsampling. Since we want to show how our wavelets improve image quality at a low resolution, all examples are provided at a coarse level of resolution.

Our first example data set is of a human brain data set. It is given as 753 slices, and each slice has a resolution of  $1050 \times 970$  points, where 24-bit RGB-color information is stored. The original data set was preprocessed with a segmentation algorithm described in [38] to eliminate noise. We generated a mesh hierarchy based on  $\sqrt[3]{2}$  subdivision and applied the trilinear B-spline wavelet filters to each color channel independently.

Since the data was too large to be stored in main memory, we used out-of-core techniques. Due to the narrow masks of our lifting scheme, at most three slices were used simultaneously.

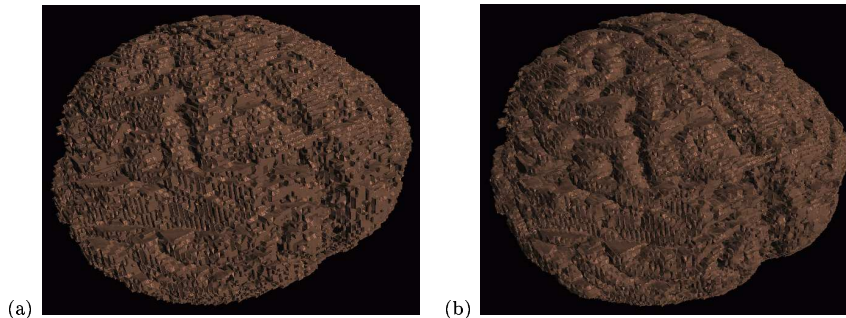


**Fig. 10.** (a) Slice through 3D brain data set at full resolution; (b) slice at resolution of 1.6% without and (c) with B-spline wavelet filters on a  $\sqrt[3]{2}$ -subdivision scheme. (Data set courtesy of A. Toga, Ahmanson-Lovelace Brain Mapping Center, University of California, Los Angeles)

For the generation of Figure 10, we used an interactive progressive slicing visualization tool, see [25], to generate an arbitrary cutting plane through the brain data set. Figure 10(a) shows a part of the slice at highest resolution, Figure 10(b) after downsampling to a resolution of 1.6% with  $\sqrt[3]{2}$  subdivision, and Figure 10(c) after downsampling to a resolution of 1.6% with  $\sqrt[3]{2}$  subdivision and trilinear B-spline wavelet filters.

Compared to Figure 10(b), the contours of the brain in Figure 10(c) are much smoother. Moreover, the slice in Figure 10(c) does not only contain information of the slice in Figure 10(a) but also of the full-resolution slices next to it. Without the averaging performed by the wavelet filter, some detail information of the neighbored slices can get lost.

Figure 11 shows an isosurface for the value 78 extracted from the same data set at a resolution of 0.2%. For isosurface extraction, we converted the RGB data to the HSV color model and used the value  $V$ . For Figure 11(a),



**Fig. 11.** Hierarchical visualization of brain data set, (a) based on  $\sqrt[3]{2}$ -subdivision without and (b) with B-spline wavelet filters. (Data set courtesy of A. Toga, Ahmanson-Lovelace Brain Mapping Center, University of California, Los Angeles)

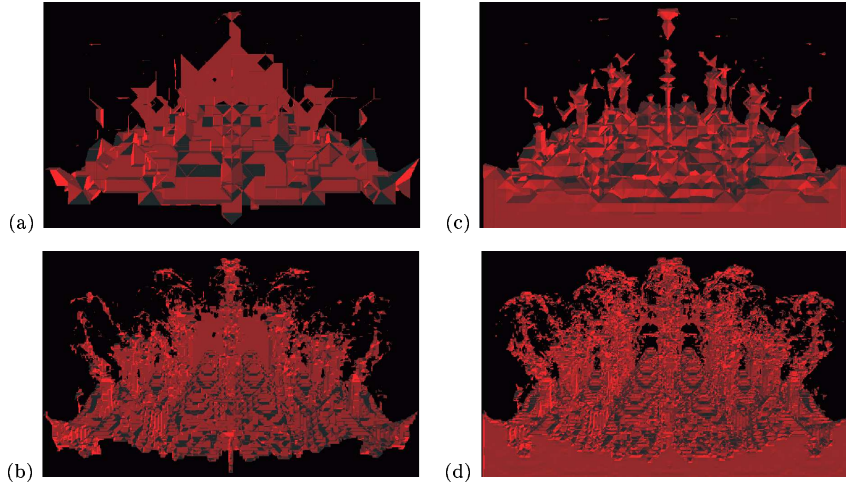
we used a  $\sqrt[3]{2}$ -subdivision hierarchy without using downsampling filters, and, for Figure 11(b), we combined the  $\sqrt[3]{2}$ -subdivision hierarchy with trilinear B-spline wavelet filters. Figure 11(b) exhibits much more detail than Figure 11(a).

For the generation of Figure 12, we applied our techniques to numerically simulated hydrodynamics data. The data set is the result of a 3D simulation of the Richtmyer-Meshkov instability and turbulent mixing in a shock tube experiment [21]. For each vertex of a  $1024^3$  structured-rectilinear grid (one time step considered only), an entropy value between 0 and 255 is stored. The figure shows the isosurface corresponding to the value 225 extracted from two different levels of resolution (one time step). Again, we contrasted the results of the  $\sqrt[3]{2}$ -subdivision hierarchy without (left column) and with (right column) trilinear B-spline wavelet filters.

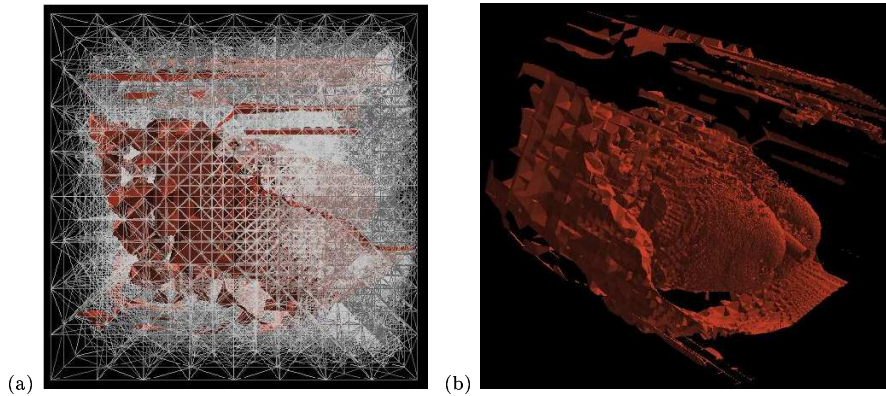
Considering the example shown in Figure 12, when using the wavelet approach low-resolution visualizations suffice to see where turbulent mixing takes place. For example, Figure 12(c) shows clearly the big “bubble” rising in the middle of the data set. The bubble can hardly be seen in Figure 12(a). The averaging steps of the wavelet filters lead to a much better approximation.

Since the generated wavelet filters are narrow (as a result of the lifting scheme), they do not restrict the application of the  $\sqrt[3]{2}$ -subdivision hierarchy to an adaptive setting. For adaptive mesh refinement in a  $\sqrt[3]{2}$ -subdivision hierarchy, we refer to [8]. In Figure 13, we show a view-dependent visualization based on an adaptively refined  $\sqrt[3]{2}$ -subdivision mesh. The data set we used is a computerized tomography (CT) scan of a primate lung consisting of  $512 \times 512 \times 266$  sample points. Figure 13(a) shows the adaptively refined mesh, where the viewpoint is positioned at the center of the right face of the bounding box. Figure 13(b) shows an isosurface extracted from the adaptively refined mesh for isovalue 86, chosen from the interval  $[0, 255]$ .





**Fig. 12.** Entropy in a 3D simulation of Richtmyer-Meshkov instability, visualized by isosurface extraction from a  $\sqrt[3]{2}$ -subdivision hierarchy without (left column) and with (right column) B-spline wavelet filters (at resolutions of 0.003% and 0.2%).

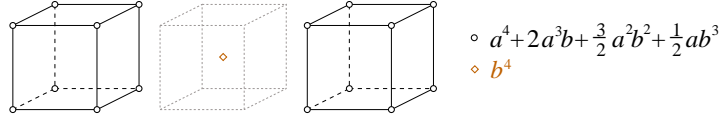


**Fig. 13.** View-dependent visualization of lung data set using an adaptively refined  $\sqrt[3]{2}$ -subdivision mesh and B-spline wavelet filters. (Data set courtesy of Erik R. Wisner, Department of Radiology, University of California, Davis)

## 9 Lifting for $\sqrt[4]{2}$ -subdivision hierarchies

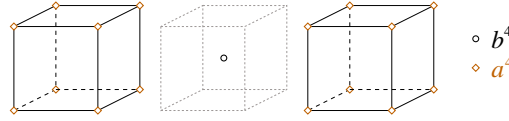
For the 4D case, we have to distinguish between four different cases referring to the four different configurations shown in Figure 6. We start with the situation shown in Figure 6(b). By convolution of the 1D masks in four coordinate directions we obtain a generalization of mask (3) to a 4D mask. We adjust it to the  $\sqrt[4]{2}$ -subdivision setting of Figure 6(b) by assuming (i) that the value at a vertex  $\bullet$  is defined by linear interpolation of the values

at the two vertices  $\circ$  (with which the vertex  $\bullet$  shares an edge); (ii) that the value at a vertex  $\triangle$  is defined by bilinear interpolation of the values at the four vertices  $\circ$  (with which the vertex  $\triangle$  shares a face); and (iii) that the value at a vertex  $\blacktriangle$  is defined by trilinear interpolation of the values at the eight vertices  $\circ$  (with which the vertex  $\blacktriangle$  shares a cuboid). Consequently, one obtains the mask  $w\text{-lift}_{\text{encode}}(a, b)$ :



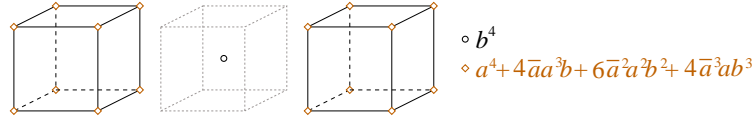
$$\begin{aligned} \circ & a^4 + 2a^3b + \frac{3}{2}a^2b^2 + \frac{1}{2}ab^3 \\ \diamond & b^4 \end{aligned}$$

We proceed as for the lower-dimensional cases and obtain the mask  $s\text{-lift}_{\text{encode}}(a, b)$



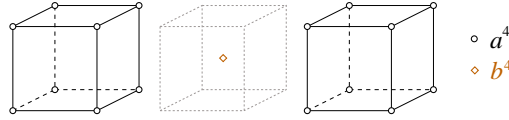
$$\begin{aligned} \circ & b^4 \\ \diamond & a^4 \end{aligned}$$

the mask  $s\text{-lift}_{\text{decode}}(a, b)$



$$\begin{aligned} \circ & b^4 \\ \diamond & a^4 + 4\bar{a}a^3b + 6\bar{a}^2a^2b^2 + 4\bar{a}^3ab^3 \end{aligned}$$

where  $\bar{a}$  is the parameter  $a$  from the s-lift encoding mask, and the mask  $w\text{-lift}_{\text{encode}}(a, b)$

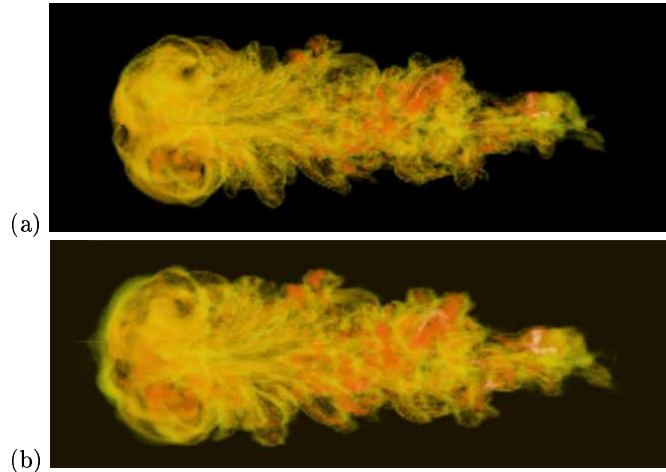


$$\begin{aligned} \circ & a^4 \\ \diamond & b^4 \end{aligned}$$

When inserting the vertices  $\blacktriangle$ , the vertices  $\triangle$ , and the vertices  $\bullet$  as done in the situations in Figure 6(c), (d), and (e), respectively, we must follow the assumptions (i)-(iii) described above. The filters reduce to the 3D, 2D, and 1D filters, respectively.

## 10 Time-varying volume data

The time-varying volume data used for the example shown in Figure 14 represents the evolution of an Argon bubble disturbed by a shock wave. The simulated data consists of 450 time steps, each one having an associated  $640 \times 256 \times 256$  rectilinear grid. For each vertex, a density value between 0 and 255 is stored. We have constructed a  $\sqrt[4]{2}$ -subdivision hierarchy combined



**Fig. 14.** Density of a time-varying simulation of an interaction of a shock with an Argon bubble, visualized by volume rendering time step 192. Downsampled to 0.78% using combined  $\sqrt[4]{2}$ -subdivision hierarchy in four dimensions and one dimension without (a) and with (b) linear B-spline wavelets.

with quadrilinear B-spline wavelet filters. For visualizing the results, we have used volume rendering.

Considering Figure 14(b), we have performed a  $\sqrt[4]{2}$ -subdivision downsampling combined with quadrilinear B-spline wavelet filters down to a resolution of 6.25%, followed by 1D downsampling steps with linear B-spline wavelet filters down to a resolution of 0.78%. The fact that our 4D wavelet lifting scheme is decomposed into a 4D, 3D, 2D, and 1D step allows us to integrate linear B-spline wavelet schemes of any dimension into one framework. One can compare this result to the one obtained when downsampling without wavelet filters, see Figure 14(a). Both pictures are the results of applying volume rendering to time step 192. Figure 14(a) only shows data from time step 192, whereas Figure 14(b) contains information of a short sequence of time steps close to time step 192, including all possibly significant changes.

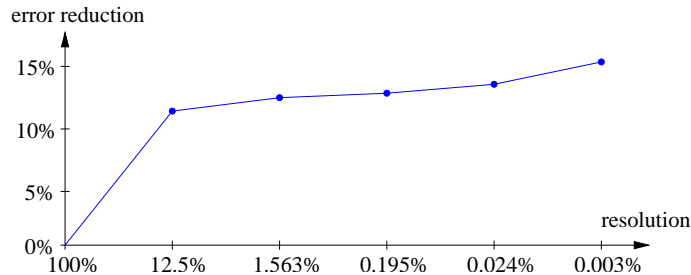
To quantify the improvement in approximation quality, we computed approximation errors for coarser levels of approximation by comparing them to the highest-resolution level. Given the original four-variate function  $F$ , represented discretely by sample values at locations  $\mathbf{x}_i$ ,  $i \in [1, n_x][1, n_y][1, n_z][1, n_t]$ , we have used the root-mean-square (RMS) error

$$E_{RMS} = \sqrt{\frac{1}{n_x n_y n_z n_t} \sum_{\mathbf{i}} \left( F(\mathbf{x}_i) - f(\mathbf{x}_i) \right)^2},$$

where  $f(\mathbf{x}_i)$  denotes the approximated function value obtained by quadrilinear interpolation applied to a hypercuboid in the coarser level of resolution.

In other words, the value of  $f(\mathbf{x}_i)$  is obtained by performing quadrilinear interpolation of the 16 function values associated with the corners of the hypercuboid containing the point  $\mathbf{x}_i$ .

Figure 15 shows the improvement when applying quadrilinear B-spline wavelet filters. The figure shows various levels of resolution (in a logarithmic scale) by how much the RMS error is reduced on average when using our wavelet filters.



**Fig. 15.** Average RMS error reduction when applying linear B-spline wavelet filters.

Since the masks of our lifting scheme are of constant size and the number of steps for our lifting scheme is constant, our algorithms run in linear time with respect to the number of original data points. Since the masks are narrow and linear B-spline wavelet operations are decomposed into only two steps, run-time constants are small.

## 11 Conclusion

We have presented multiresolution hierarchies based on  $\sqrt[n]{2}$  subdivision and  $n$ -variate B-spline wavelet filters. We have described the methods in a general setting with particular focus on the 1D, 2D, 3D, and 4D case, since they can be used for hierarchical representations of curves, surfaces, volume data, and time-varying volume data, respectively.

The  $\sqrt[n]{2}$ -subdivision scheme leads to a regular data organization. Mesh connectivity and vertex locations are implicitly defined (no additional storage space is necessary), and data access is simple and fast. In general, regular schemes have the drawbacks of coarse granularity and low adaptivity. The  $\sqrt[n]{2}$ -subdivision scheme only doubles the number of vertices in each subdivision step regardless of the dimension  $n$ .

For high-quality data approximation on each level of detail, we use down-sampling filters based on  $n$ -variate (linear) B-spline wavelets. The filters must be defined for the  $\sqrt[n]{2}$ -subdivision connectivity without restricting adaptivity. We derived B-spline wavelet lifting schemes for  $\sqrt[n]{2}$  subdivision leading to small / narrow filters.

We have demonstrated the benefit of using these filters by providing 2D, 3D, and 4D examples and applying various visualization tools (including view-dependent isosurface extraction from adaptively refined meshes) and even out-of-core data exploration techniques.

## Acknowledgments

This work was supported by the National Science Foundation under contracts ACI 9624034 (CAREER Award) and ACI 0222909, 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; 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, and the members of the Data Science Group at the Center for Applied Scientific Computing (CASC) at the Lawrence Livermore National Laboratory, Livermore, California.

## References

1. Dostas Anagnostou, Tim J. Atherton, and Andrew E. Waterfall. 4d volume rendering with the shear warp factorization. In Roger Crawfis and Daniel Cohen-Or, editors, *Proceedings of the ACM/IEEE Volume Visualization and Graphics Symposium 2000, Salt Lake City, Utah*, pages 129–137. ACM/IEEE, 2000.
2. Martin Bertram. *Multiresolution Modeling for Scientific Visualization*. PhD thesis, Department of Computer Science, University of California, Davis, California, <http://www.cipic.ucdavis.edu/publications>, 2000.
3. Martin Bertram, Mark A. Duchaineau, Bernd Hamann, and Kenneth I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In Thomas Ertl, Bernd Hamann, and Abitabh Varshney, editors, *Proceedings of IEEE Conference on Visualization 2000*, pages 389–396. IEEE, IEEE Computer Society Press, 2000.
4. Martin Bertram, Daniel E. Laney, Mark A. Duchaineau, Charles D. Hansen, Bernd Hamann, and Kenneth I. Joy. Wavelet representation of contour sets. In Thomas Ertl, Kenneth I. Joy, and Abitabh Varshney, editors, *Proceedings of IEEE Conference on Visualization 2001*, pages 303–310. IEEE, IEEE Computer Society Press, 2001.
5. Paolo Cignoni, Claudio Montani, Enrico Puppo, and Roberto Scopigno. Multiresolution modeling and visualization of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):352–369, 1997.

6. Albert Cohen and Ingrid Daubechies. Nonseparable bidimensional wavelet bases. *Rev. Mat. Iberoamericana*, 9(1):51–137, 1993.
7. Thomas Gerstner and Renato Pajarola. Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In Thomas Ertl, Bernd Hamann, and Abitabh Varshney, editors, *Proceedings of IEEE Conference on Visualization 2000*, pages 259–266. IEEE, IEEE Computer Society Press, 2000.
8. Benjamin Gregorski, Mark A. Duchaineau, Peter Lindstrom, Valerio Pascucci, and Kenneth I. Joy. Interactive view-dependent rendering of large isosurfaces. In Robert Moorhead, Markus Gross, and Kenneth I. Joy, editors, *Proceedings of the IEEE Conference on Visualization 2002*, pages 475–482. IEEE, IEEE Computer Society Press, 2002.
9. Roberto Grosso and Günther Greiner. Hierarchical meshes for volume data. In F.-E. Wolter and N. M. Patrikalakis, editors, *Proceedings of CGI '98, Hanover, Germany*, 1998.
10. Roberto Grosso, Christoph Lürig, and Thomas Ertl. The multilevel finite element method for adaptive mesh optimization and visualization of volume data. In R. Yagel and H. Hagen, editors, *Proceedings of IEEE Conference on Visualization 1997*, pages 135–142. IEEE, IEEE Computer Society Press, 1997.
11. Lutz Kettner and Jack Snoeyink. Video: A prototype system for visualizing time-dependent volume data. In *In Video Proceedings of the 17th ACM Symposium on Computational Geometry*, 2001.
12. Leif Kobbelt.  $\sqrt{3}$ -subdivision. In Kurt Akeley, editor, *Proceedings of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 103–112. ACM, ACM Press / ACM SIGGRAPH, 2000.
13. Leif Kobbelt. Multiresolution techniques. In Farin, Hoschek, and Kim, editors, *Handbook of Computer Aided Geometric Design*. Elsevier Science Publishing, Amsterdam, The Netherlands, 2002.
14. Jelena Kovačević and Wim Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Transactions on Image Processing*, 9(3):480–496, 1999.
15. Jelena Kovačević and Martin Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for  $\mathbf{r}^n$ . *IEEE Transactions on Information Theory*, 38(2):533–555, 1992.
16. Lars Linsen, Jevan T. Gray, Valerio Pascucci, Mark A. Duchaineau, Bernd Hamann, and Kenneth I. Joy. Hierarchical large-scale volume representation with  $\sqrt[3]{2}$  subdivision and trivariate b-spline wavelets. *Technical Report Number CSE-2002-7, Department of Computer Science, University of California, Davis*, 2002.
17. Lars Linsen, Valerio Pascucci, Mark A. Duchaineau, Bernd Hamann, and Kenneth I. Joy. Hierarchical representation of time-varying volume data with  $\sqrt[4]{2}$  subdivision and quadrilinear b-spline wavelets. In Coquillart, Shum, and Hu, editors, *Proceedings of Tenth Pacific Conference on Computer Graphics and Applications – Pacific Graphics 2002*. IEEE, IEEE Computer Society Press, 2002.
18. L. Lippert, M. H. Gross, and C. Kurmann. Compression domain volume rendering for distributed environments. In *Proceedings of the Eurographics '97*, volume 14, pages 95–107. COMPUTER GRAPHICS Forum, 1997.
19. Donald Maegher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19:129–147, 1982.

20. Joseph M. Maubach. Local bisection refinement for  $n$ -simplicial grids generated by reflection. *SIAM J. Scientific Computing*, 16:210–227, 1995.
21. Arthur A. Mirin, Ron H. Cohen, Bruce C. Curtis, William P. Dannevik, Andris M. Dimits, Mark A. Duchaineau, D. E. Eliason, Daniel R. Schikore, S. E. Anderson, D. H. Porter, and Paul R. Woodward. Very high resolution simulation of compressible turbulence on the ibm-sp system. In Sally Howe, editor, *Proceedings of Supercomputing '99*. IEEE, IEEE Computer Society Press, 1999.
22. Mario Ohlberger and Martin Rumpf. Hierarchical and adaptive visualization on nested grids. *Computing*, 59:365–385, 1997.
23. Valerio Pascucci. Slow growing subdivision (sgs) in any dimension: towards removing the curse of dimensionality. In *Proceedings of Eurographics 2002*. COMPUTER GRAPHICS Forum, 2002.
24. Valerio Pascucci and Chandrajit Bajaj. Time critical adaptive refinement and smoothing. In Roger Crawfis and Daniel Cohen-Or, editors, *Proceedings of the ACM/IEEE Volume Visualization and Graphics Symposium 2000, Salt Lake City, Utah*, pages 33–42. ACM/IEEE, 2000.
25. Valerio Pascucci and Randall J. Frank. Global static indexing for real-time exploration of very large regular grids. In *Proceedings of Supercomputing 2001*. ACM, ACM Press, 2001.
26. Dmitriy Pinskiy, Erie Brugger, Henry R. Childs, and Bernd Hamann. An octree-based multiresolution approach supporting interactive rendering of very large volume data sets. In H. Arabnia, R. Erbacher, X. He, C. Knight, B. Kovalerchuk, M. Lee, Y. Mun, M. Sarfraz, J. Schwing, and H. Tabrizi, editors, *Proceedings of the 2001 International Conference on Imaging Science, Systems, and Technology (CISST 2001), Volume 1*, pages 16–22. Computer Science Research, Education, and Applications Press (CSREA), Athens, Georgia, 2001.
27. Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bézier and B-spline Techniques*. Springer-Verlag, Heidelberg, Germany, 2002.
28. Sherman D. Riemenschneider and Zuowei Shen. Wavelets and pre-wavelets in low dimensions. *Journal Approximation Theory*, 71:18–38, 1992.
29. Raj Shekhar, Elias Fayyad, Roni Yagel, and J. Fredrick Cornhill. Octree-based decimation of marching cubes surfaces. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of IEEE Conference on Visualization 1997*, pages 335–342. IEEE, IEEE Computer Society Press, 1996.
30. Han-Wei Shen. Isosurface extraction from time-varying fields using a temporal hierarchical index tree. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, *Proceedings of IEEE Conference on Visualization 1998*, pages 159–166. IEEE, IEEE Computer Society Press, 1998.
31. Han-Wei Shen, Ling-Jen Chiang, and Kwan-Liu Ma. A fast volume rendering algorithm for time-varying fields using a time-space partitioning (tsp) tree. In David Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of IEEE Conference on Visualization 1999*, pages 371–378. IEEE, IEEE Computer Society Press, 1999.
32. Han-Wei Shen and Christopher R. Johnson. Differential volume rendering: A fast algorithm for flow animation. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of IEEE Conference on Visualization 1994*, pages 180–187. IEEE, IEEE Computer Society Press, 1994.
33. Oliver G. Staadt. Multiresolution surface and volume representation. In Guido Brunnett, Bernd Hamann, and Heinrich Müller, editors, *Geometric Modeling for Scientific Visualization*. Springer-Verlag, Heidelberg, Germany, 2003.

34. Oliver G. Staadt and Markus H. Gross. Progressive tetrahedralizations. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, *Proceedings of IEEE Conference on Visualization 1998*, pages 397–402. IEEE Computer Society Press, 1998.
35. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. *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., 1996.
36. Philipp Sutton and Charles D. Hanson. Isosurface extraction in time-varying fields using a temporal branch-on-need tree (t-bon). In David Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of IEEE Conference on Visualization 1999*, pages 147–153. IEEE, IEEE Computer Society Press, 1999.
37. Wim Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proceedings of SPIE 2569, 1995.
38. Ikuko Takanashi, Eric Lum, Kwan-Liu Ma, Jörg Meyer, Bernd Hamann, and Arthur J. Olson. Segmentation and 3d visualization of high-resolution human brain cryosections. In Robert F. Erbacher, Philip C. Chen, Matti Gröhn, Jonathan C. Roberts, and Craig M. Wittenbrink, editors, *Proceedings of SPIE Visualization and Data Analysis 2002*, pages 55–61. Proceedings of SPIE 4665, 2002.
39. Melanie Tory, Niklas Röber, Torsten Möller, Anna Center, and M. Stella Atkins. 4d space-time techniques: A medical imaging case study. In Thomas Ertl, Ken Joy, and Amitabh Varshney, editors, *Proceedings of IEEE Conference on Visualization 2001*, pages 473–476. IEEE, IEEE Computer Society Press, 2001.
40. Geert Uytterhoeven. *Wavelets: Software and Applications*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1999.
41. Luiz Velho and Denis Zorin. 4-8 subdivision. *Computer-Aided Geometric Design*, 18(5):397–427, 2001.
42. Joe Warren and Henrik Weimer. *Subdivision Methods for Geometric Design*. Morgan Kaufmann Publishers, San Francisco, U.S.A., 2002.
43. Rüdiger Westermann, Leif Kobbelt, and Thomas Ertl. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer*, pages 100–111, 1999.
44. Yong Zhou, Baoquan Chen, and Arie E. Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In Roni Yagel and Hans Hagen, editors, *Proceedings of IEEE Conference on Visualization 1997*, pages 135–142. IEEE, IEEE Computer Society Press, 1997.