Title

Hierarchical Representation of Time-varying Volume Data with 4th-root-of-2 Subdivision and Quadrilinear B-spline Wavelets

Authors

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

Publication Date

2002

Peer reviewed

Hierarchical Representation of Time-varying Volume Data with $\sqrt[4]{2}$ Subdivision and Quadrilinear B-spline Wavelets

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

* Center for Image Processing and Integrated Computing (CIPIC)
Department of Computer Science
University of California, Davis‡

† Center for Applied Scientific Computing (CASC)
Data Science Group
Lawrence Livermore National Laboratory

Abstract

Multiresolution methods for representing data at multiple levels of detail are widely used for large-scale two- and three-dimensional data sets. We present a four-dimensional multiresolution approach for time-varying volume data. This approach supports a hierarchy with spatial and temporal scalability.

The hierarchical data organization is based on $\sqrt[4]{2}$ subdivision. The $\sqrt[n]{2}$-subdivision scheme only doubles the overall number of grid points in each subdivision step. This fact leads to fine granularity and high adaptivity, which is especially desirable in the spatial dimensions.

For high-quality data approximation on each level of detail, we use quadrilinear B-spline wavelets. We present a linear B-spline wavelet lifting scheme based on $\sqrt[n]{2}$ subdivision to obtain narrow masks for the update rules. Narrow masks provide a basis for out-of-core data exploration techniques and view-dependent visualization of sequences of time steps.

1 Introduction

Due to the improvements in the performance of computing power and storage capacity achieved over the last decade, today's data-intensive scientific applications are capable of quickly generating and storing huge amounts of data. Downsampling can be used to reduce the data to a manageable amount. The reduced data can be examined by scientists to spot regions of interest, for which more detailed examinations can be performed. Today, visualization applications have to deal with large-scale data in the spatial as well as temporal dimensions and their representation at multiple levels of detail.

Multiresolution methods for representing data at multiple levels of detail are widely used for large-scale two- and three-dimensional data sets. Furthermore, for time-varying data sets techniques have been developed that make use of temporal coherence of, for example, numerically simulated data. We present a four-dimensional multiresolution approach, where time is treated as fourth dimension. We deal with large scales in spatial and temporal dimensions in a single hierarchical framework.

For large-scale volume representation, one should use regular rather than irregular data formats, since grid connectivity should be implicit and data should be easily and quickly accessible. To overcome regular data structures' disadvantage of coarse granularity, we have developed a $\sqrt[3]{2}$-subdivision scheme [14]. Every $\sqrt[n]{2}$-subdivision step only doubles the number of vertices, which is a factor of $\sqrt[n]{2}$ in each of the n dimensions. We briefly review the $\sqrt[n]{2}$-subdivision scheme for $n = 2$ in Section 3 and generalize it to the four-dimensional case.

Another sacrifice when using regular data structures is that downsampling is done based purely on grid structure and without considering data values. Therefore, some scientifically interesting details in a data set can get lost and be overseen for further examinations. To avoid this, we use a linear B-spline wavelet scheme: 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 as well as an average value of a certain region around $\mathbf{p}$. This approach leads to better approximations on coarser levels.

Quadrilinear 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 in the spatial and temporal dimensions. Thus, when using out-of-core techniques to operate on or visu-

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

alize large-scale data, substantial amounts of data must be loaded from external memory, with low I/O-performance. Lifting schemes with narrow filters can be used to overcome this problem. In Section 4, we describe the one-dimensional lifting scheme from [4] and generalize it to four dimensions based on a hypercube refinement approach. In Section 5, we adjust the quadrilinear B-spline wavelet lifting scheme to $\sqrt[4]{2}$ subdivision. We provide results in Section 6.

2 Related work

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 update in time [1, 27, 29, 31]. These approaches consider scaling in time but not in space. An approach dealing with large-scale data in time and space was described by Shen et al. [28]. 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. We treat time as a real fourth dimension.

Octrees are a common data structure used for multiresolution volume representation [15, 19, 23, 26, 37]. Compared to irregular data structures, as discussed in [5, 8, 9], regular structures like octrees have the advantage that grid connectivity is implicit and data is easily and quickly accessed. However, the refinement steps have to conform to the topological constraints, which can make regular structures less adaptive. To overcome this disadvantage, we developed the $\sqrt[3]{2}$-subdivision scheme, a regular data organization supporting finer granularity [14]. For example, an octree refinement step doubles the number of vertices in every dimension, which leads to a growth factor of eight; a $\sqrt[3]{2}$-subdivision step only doubles the overall number of vertices. Therefore, $\sqrt[3]{2}$ subdivision, in general, requires less vertices than octrees to satisfy specified approximation or image quality error bounds. Since finer granularity leads to higher adaptivity, this fact still holds when using adaptive refinement techniques.

Considering time-varying volume visualization, isosurface extraction [10, 27, 31] and volume rendering [1, 29] of single time steps are common, and distributed computing can be used to speed it up [16, 20]. A comparison of different visualization techniques is provided in [33]. However, for large-scale data, the visualization techniques can only operate in real time after downsampling the data. Therefore, large-scale time-varying volume visualization requires us to utilize multiresolution representations with scalability in time and space. Such a representation is discussed in this paper.

The splitting step of the $\sqrt[n]{2}$-subdivision scheme goes back to Cohen and Daubechies [6] for $n = 2$ and Maubach [17] for arbitrary n. It can be described by using triangular or quadrilateral meshes ($n = 2$), or their generalizations for higher dimensions. For tetrahedral meshes the splitting step of the $\sqrt[3]{2}$-subdivision scheme is equivalent to longest-edge bisection [7, 22, 38]. In the following, we consider the quadrilateral case and its generalization, e. g., cuboids for $n = 3$ and hypercuboids for $n = 4$. Figure 1 depicts two different illustrations of a hypercube. Figure 1(a) shows the symmetry in all four dimensions. We use the illustration in Figure 1(b) that "stretches" the hypercube in the temporal dimension.

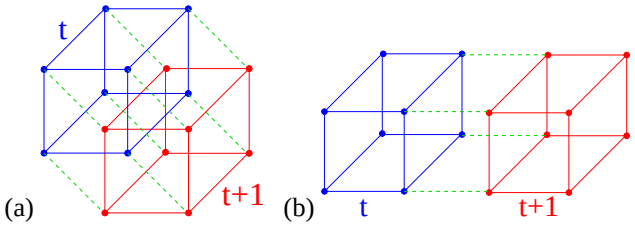


Figure 1. Illustrations of a hypercube.

Velho and Zorin [35] introduced $\sqrt{2}$-subdivision surfaces ($n = 2$) by adding an averaging step to the $\sqrt{2}$-subdivision splitting step. They 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 [36].)

When downsampling time-varying volume data in a regular fashion, as it is done by $\sqrt[4]{2}$-subdivision and the method of Shen et al. [28], data is not grouped due to changes in time or space. Thus, on coarse levels, some important details may be missing. We overcome this problem by using wavelets. Wavelet schemes provide a means to generate best approximations in a multiresolution hierarchy. Stollnitz et al. [30] described how to generate wavelets for subdivision schemes. However, $\sqrt[n]{2}$-subdivision wavelets can lead to over- and undershoots, which are disturbing during visualization, e. g., when extracting isosurfaces from different levels of approximation. They can cause changes of isosurface topology when changing the level of resolution. We would like to preserve isosurface topology as much as possible when changing between approximation levels. Therefore, we generate linear B-spline wavelets for the $\sqrt[n]{2}$-subdivision scheme. Linear B-spline wavelets are known to produce high-quality approximations and they have interpolating scaling functions, which guarantees interpolating refinement filters, see [12], i. e., no over- and undershoots can appear. (For an introduction to B-spline techniques, we refer to [24].)

The computation of wavelet coefficients at a certain vertex for wavelets with good approximation quality, e. g., for linear B-spline wavelets, is not limited to only adjacent vertices. Localization, however, is strongly desirable when we want to apply a wavelet scheme to adaptive refinement and

to out-of-core visualization techniques. Lifting schemes as introduced by Sweldens [32] decompose wavelet computations into several steps, but they assert narrow filters, see Figure 6. Bertram et al. [3, 4] defined a lifting scheme for one- and two-dimensional B-spline wavelets using a quadtree organization of the vertices.

Wavelets for general dilation matrices go back to Riemenschneider and Shen [25] who used a box-spline approach for their construction. Kovačević and Vetterli [13] and, more recently, Uytterhoeven [34], Kovačević and Sweldens [12], and Linsen et al. [14] developed lifting schemes that can be applied to $\sqrt[n]{2}$-subdivision data structures. Uytterhoeven's method [34] addresses the two-dimensional case, Kovačević and Sweldens' approach [12] as well as Linsen et al.'s technique [14] deal with the two- and three-dimensional cases. The filters used in [12] that produce good approximations are not narrow enough for our purposes. On the other hand, the update rule for the narrow filters in [12] is the identity, which does lead to high-quality approximations.

Another main difference between the non-separable filters used in [34] and [12] and the approach in [14] is the update rule. Following the approach from [14], we update the vertices in, for example, a $\sqrt[3]{2}$-subdivision scheme by applying first the three-, then the two-, and finally the one-dimensional update rules. This approach automatically includes the boundary cases, which are not sufficiently addressed in [34] and [12]. Moreover, in this paper we show how our lifting scheme can be generalized to four dimensions.

3 The $\sqrt[n]{2}$-subdivision scheme

We first describe the case $n = 2$. For a $\sqrt{2}$-subdivision step of a quadrilateral Q, we compute its centroid $\mathbf{c}$ and connect $\mathbf{c}$ to all four vertices of Q. The "old" edges of the mesh are removed (except for the edges determining the mesh/domain boundary). Figure 2 illustrates four $\sqrt{2}$-subdivision steps.

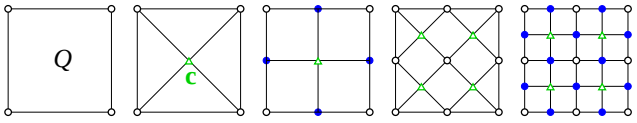


Figure 2. Illustration of $\sqrt{2}$ subdivision.

The mask used for the computation of the centroid $\mathbf{c}$ is given in Figure 3(a). Figure 3(b) shows the mask of the averaging step according to [35]. A $\sqrt{2}$-subdivision step is executed by first applying the mask shown in Figure 3(a), which inserts the new vertices, and then (after the topological mesh modifications) applying the mask shown in Figure 3(b), which adjusts the old vertices.

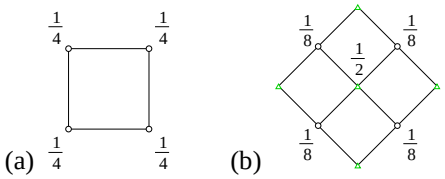


Figure 3. Masks of $\sqrt{2}$-subdivision step: (a) inserting centroid; (b) adjusting old vertices.

This subdivision scheme for quadrilaterals is analogous to the $\sqrt{3}$-subdivision scheme of Kobbelt [11] for triangles. 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 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.

The literature currently provides no analysis of averaging steps for dimensions larger than two. Thus, at present, we cannot provide a solution for the "optimal choice" of α used in the update rule. (Some investigations were made in [21].) However, when using the $\sqrt[4]{2}$-subdivision scheme for large-scale time-varying volume data, we deal with rectilinear grids with all hypercuboids having the same size. Thus, the update rule does not affect the position of the vertices regardless of the specific α value, but it only affects the dependent function values at the vertices. In [14], we showed that the $\sqrt{2}$-subdivision wavelets are not appropriate for our purposes. Thus, we replace them by B-spline wavelets and do not need to choose a value for α.

In Figure 4, four $\sqrt[4]{2}$-subdivision steps are shown. In each step, the centroids of the polyhedral shapes are inserted, and the connectivity is adjusted. In Figure 4, we only show the spatial connectivities within the time steps and do not show the connectivity information between time steps.

The four subdivision steps can be described in the following way: Figure 4(a) shows the initial hypercuboid, which consists of two cuboids at two time steps, say t_1 and t_3. The two cuboids are connected according to Figure 1(b). The first subdivision step inserts the centroid of the hypercuboid, shown in Figure 4(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 4(c). The third step inserts the centers of the faces of these eight cuboids or of the original hypercuboid, respectively, shown in Figure 4(d). Finally, the fourth step inserts the midpoints of the edges of the eight cuboids or of the original hyper-

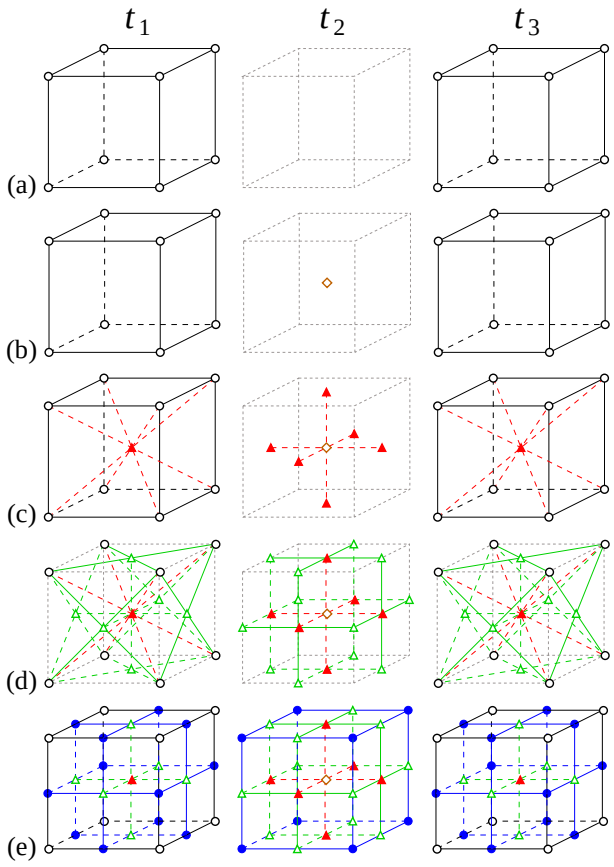


Figure 4. Steps of $\sqrt[4]{2}$ subdivision.

cuboid, respectively, shown in Figure 4(e). The geometric structure shown in Figure 4(e) consists of 16 hypercuboids. Thus, it is topologically equivalent to the one shown in Figure 4(a). Four $\sqrt[4]{2}$-subdivision steps produce the same result as one "hexadectree" refinement step, where a hexadectree is the generalization of an octree to four dimension, i. e., nodes of hexadectrees represent hypercuboids. Fine granularity can therefore be supported for multiresolution data visualization purposes by using a $\sqrt[4]{2}$-subdivision approach.

4 The Linear B-spline wavelet lifting scheme

We briefly review the one-dimensional lifting scheme discussed in [3] and generalize it to the four-dimensional case. We adjust the four-dimensional lifting scheme to be suitable for $\sqrt[4]{2}$ subdivision.

The one-dimensional B-spline wavelet lifting scheme uses 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 ∘ and their new neighbors •, whereas the w-lift mask is applied to the new vertices • and their neighbors ∘, see Figure 5(a). For a detailed derivation of the lifting scheme that we use, as well as for its analysis (smoothness, stability, approximation order, and zero moments), we refer to [2].

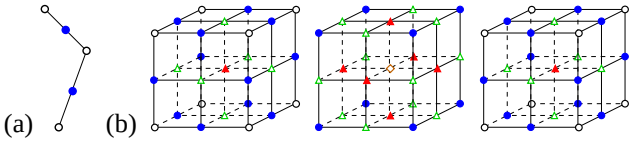


Figure 5. Refinement step for one- and four-dimensional meshes.

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

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

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

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

Figure 6 illustrates the one-dimensional linear B-spline wavelet lifting scheme.

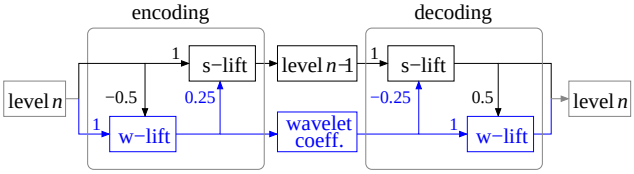
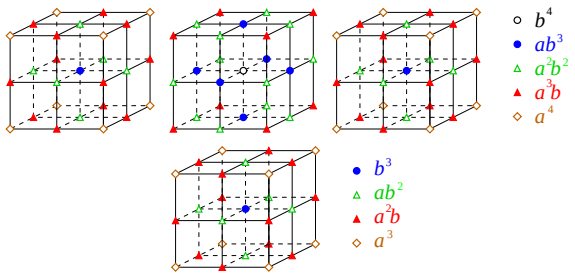


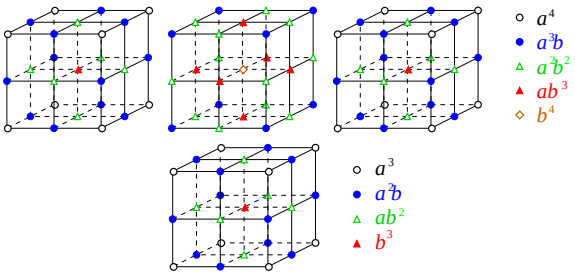
Figure 6. One-dimensional linear B-spline wavelet lifting scheme.

When applying four-dimensional B-spline wavelets to a hexadectree-organized set of vertices, four kinds of new vertices are obtained when executing a refinement step: the new vertices inserted at the midpoints of old edges •, the new vertices inserted at the centers of old faces △, the new vertices inserted at the centroids of old cuboids ▲, and the new vertices inserted at the centroids of old hypercuboids ◇, see Figure 5(b). Therefore, we must apply four different masks. For three- and four-dimensional masks, we show the structures of the masks and separately define the values for the different kinds of vertices. We derive the needed n-dimensional masks by convolution of the one-dimensional masks in the various coordinate directions. The s-lift(a,b) masks are defined by this depiction:



$$\begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix} \text{ and } \begin{pmatrix} a & b & a \end{pmatrix}.$$

The w-lift(a, b) masks are defined by this depiction:



$$\begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix} \text{ and } \begin{pmatrix} a & b & a \end{pmatrix}.$$

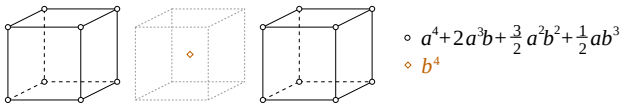
5 A lifting scheme for $\sqrt[4]{2}$ subdivision

Recalling the steps of a $\sqrt[4]{2}$-subdivision scheme, which are depicted in Figure 4, after the execution of the different steps different configurations arise. Therefore, we have to distinguish between the different steps. The following description starts with the situation shown in Figure 4(b) (*hypervolume case*), proceeds with the situation shown in Figure 4(c) (*volume case*), continues with the situation shown in Figure 4(d) (*face case*), and finally treats the situation shown in Figure 4(e) (*edge case*), which is topologically equivalent to the situation shown in Figure 4(a).

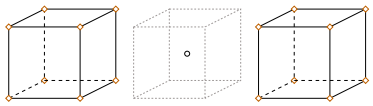
The hypervolume case

In the hypervolume case, we perform linear B-spline wavelet encoding according to the situation shown in Figure 4(b). Due to the lifting scheme discussed in Section 4, we must start with a w-lift operation. Therefore, we apply four masks similar to the four w-lift masks in Section 4, subject to the constraint that no values are available at the vertices •, △, and ▲.

Regarding the structures of the four-dimensional w-lift masks described in Section 4, we assume (i) that the value at a vertex • is defined by linear interpolation of the values at the two vertices ∘ (with which the vertex • shares an edge); (ii) that the value at a vertex △ is defined by bilinear interpolation of the values at the four vertices ∘ (with which the vertex △ shares a face); and (iii) that the value at a vertex ▲ is defined by trilinear interpolation of the values at the eight vertices ∘ (with which the vertex ▲ shares a cuboid). Since we are using linear B-spline wavelets, linear interpolation is appropriate. Consequently, one obtains the mask w-lift$_{encode}(a, b)$:

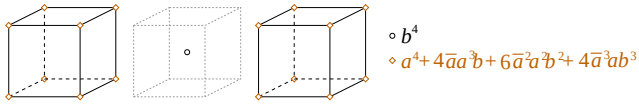


Since the vertices •, △, and ▲ are not available, no masks analogous to the one-, two-, and three-dimensional w-lift masks, as described in Section 4, have to be applied. However, theoretically these operations were executed, which must be considered in the next step. Since (i) the values at the vertices • are assumed to be linear interpolations of the values at the vertices ∘, (ii) the values at the vertices △ are assumed to be bilinear interpolations of the values at the vertices ∘, and (iii) the values at the vertices ▲ are assumed to be trilinear interpolations of the values at the vertices ∘, the values at the vertices •, △, and ▲ vanish. Therefore, the mask defining the next s-lift operation, which is an analogue of the four-dimensional s-lift mask defined in Section 4, reduces to the mask s-lift$_{encode}(a, b)$:



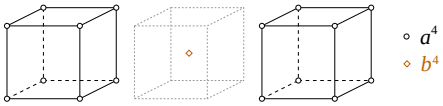
Again, the analogous versions of the one-, two-, and three-dimensional s-lift masks from Section 4 are only applied theoretically.

For the decoding step, we start with the s-lift operation, i. e., we adjust the four-dimensional s-lift mask from Section 4. Having (theoretically) applied the one-, two-, and three-dimensional s-lift masks from Section 4 with vanishing values at the vertices •, △, and ▲, (i) the values at the vertices ▲ are linear interpolations of the values at the neighbor vertices ⋄, multiplied by the factor $2a$, (ii) the values at the vertices △ are bilinear interpolations of the values at the neighbor vertices ⋄, multiplied by the factor $4a^2$, and (iii) the values at the vertices • are trilinear interpolations of the values at the neighbor vertices ⋄, multiplied by the factor $8a^3$. By renaming the factor a to $\overline{a}$, we obtain the mask s-lift$_{decode}(a, b)$:



Again, the analogous versions of the one-, two-, and three-dimensional s-lift masks from Section 4 are only applied theoretically. Since these one-, two-, and three-dimensional decoding s-lift operations are the inverse of the one-, two-,

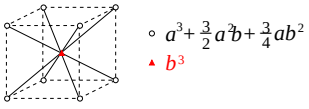
and three-dimensional encoding s-lift operations, the vertices •, △, and ▲ have their former values assigned again, i. e., the values vanish at these vertices. Hence, the mask for the final w-lift operation, which is the mask analogous to the four-dimensional w-lift mask defined in Section 4, reduces to the mask w-lift$_{encode}(a, b)$:



All the masks are as narrow as they can be.

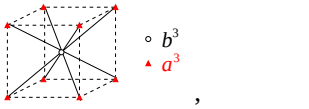
The volume case

When applying linear B-spline wavelet encoding to the situation depicted in Figure 4(c), we must make sure that we do not violate the assumptions made for the hypervolume case. We assume that the values at the vertices ▲ are trilinear interpolations of the values at the neighbor vertices ∘. Thus, when the values at the vertices △ are available, their values should be computed only from the values at the vertices ∘. Therefore, we are left with the three-dimensional case, which is examined in [14]. The construction is analogous to the four-dimensional (hypervolume) case. Encoding is performed by applying the masks w-lift$_{encode}(a, b)$, depicted as



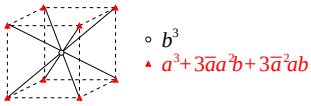
,

and s-lift$_{encode}(a, b)$, depicted as



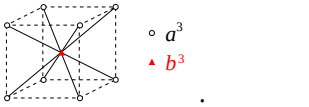
,

and decoding is performed by applying the masks s-lift$_{decode}(a, b)$, depicted as



,

and w-lift$_{decode}(a, b)$, depicted as



.

The face case

When applying linear B-spline wavelet encoding to the situation depicted in Figure 4(d), we must not violate the assumption that the values at the vertices △ are bilinear interpolations of the values at the neighbor vertices ∘. When the values at the vertices △ are available, their values should be computed only from the values at the vertices ∘. Thus, we are left with the two-dimensional case, which is also examined in [14]. Encoding is performed by applying the masks w-lift$_{encode}(a, b)$, depicted as

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

and s-lift$_{encode}(a, b)$, depicted as

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

and the decoding is performed by applying the masks s-lift$_{decode}(a, b)$, depicted as

$$\begin{pmatrix} a^2+2\overline{a}ab & & a^2+2\overline{a}ab \\ & b^2 & \\ a^2+2\overline{a}ab & & a^2+2\overline{a}ab \end{pmatrix},$$

and w-lift$_{decode}(a, b)$, depicted as

$$\begin{pmatrix} a^2 & & a^2 \\ & b^2 & \\ a^2 & & a^2 \end{pmatrix}.$$

The edge case

When applying linear B-spline wavelet encoding to the situation shown in Figure 4(e), which is topologically equivalent to the situation shown in Figure 4(a), we must not violate the assumption that the values at the vertices • are linear interpolations of the values at the neighbor vertices ∘. When the values at the vertices • are available, their values should be computed only from the values at the vertices ∘. Thus, we are left with the one-dimensional case, illustrated in Section 4. We can apply masks (1) and (2) to deal with the edge case.

It is a significant advantage of our scheme that the volume, face, and edge cases cover automatically boundary volumes, boundary faces, and boundary edges of the domain. Thus, no additional boundary case examination is necessary.

6 Results

We have applied our techniques to numerically simulated hydrodynamics data. The data set used to generate Figure 7 is the result of a three-dimensional time-varying simulation of the Richtmyer-Meshkov instability and turbulent mixing in a shock tube experiment [18]. The simulation result is stored in 274 time steps, and each time step has an associated 2048^3 rectilinear grid. For each vertex, an entropy value between 0 and 255 is stored. The figure shows one slice of the rectilinear grid for two different time steps. Figures 7(a)–(c) show time step 96 and Figures 7(d)–(f) time step 184. Figures 7(a) and 7(d) show the original slices at highest resolution (2048^2), whereas Figures 7(b), 7(c), 7(e), and 7(f) show the slices after downsampling the four-dimensional data using $\sqrt[4]{2}$ subdivision with a downsampling ratio of 2^{12}. (The downsampling ratio is defined as the original number of vertices divided by the number of

vertices at the used coarser resolution.) For the creation of Figures 7(b) and 7(e), we have applied the $\sqrt[4]{2}$-subdivision hierarchy without linear B-spline wavelet encoding, and for the creation of Figures 7(c) and 7(f), we have applied the $\sqrt[4]{2}$-subdivision hierarchy with linear B-spline wavelet encoding. For the wavelet encoding, we have only considered this single slice. Since spatial and temporal dimensions are treated equally, we have effectively performed a trilinear B-spline wavelet encoding.

Considering the temporal dimension, the wavelet encoding leads to an averaging over several time steps. Therefore, one time step represents changes of several time steps of the original data set. In Figure 7(c), we see that the "bubble" rising in the middle of the slice is already more clearly visible than in Figure 7(b).

Considering the spatial dimensions, the wavelet encoding leads to an averaging over several adjacent vertices within each grid of one particular time step. Therefore, detailed features do not get lost during downsampling. In Figure 7(f), we can still see where the bubbles next to the center bubble have their "offspring." The fine connections indicating the offspring are visible in Figure 7(d) but get lost during downsampling when not using wavelets, see Figure 7(e).

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

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

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 8 shows the RMS errors of the time-varying three-dimensional simulation of the Richtmyer-Meshkov instability for various levels of resolution. (We scaled the RMS errors to the interval $[0, 1]$.) For all resolutions, we have obtained smaller RMS errors when using linear B-spline wavelets. Furthermore, the benefits of using linear B-spline wavelets increase as resolutions decrease.

For data organization, the storage of values can be (re-) organized as shown in Figure 9 for the two-dimensional case. The depicted scheme scales to arbitrary dimensions. Reorganization leads to spatial locality of data belonging to the same level of detail, and spatial locality leads to fast data access. This fact can be used for progressive visualization, e. g., for generating images progressively by loading data from slower external memory, which is inevitable when dealing with large-scale data sets. 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 continuous stream without reading data multiple times.

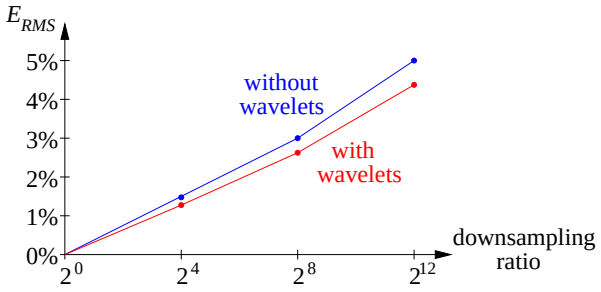


Figure 8. RMS errors for entropy in a four-dimensional simulation of Richtmyer-Meshkov instability for different levels of resolution, without and with linear B-spline wavelets.

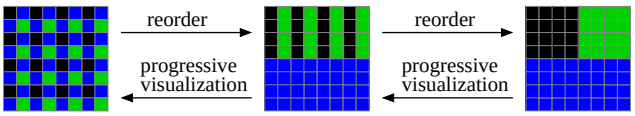


Figure 9. Reordering data for progressive visualization.

The time-varying volume data used for the examples shown in Figures 10 and 11 represents the evolution of an argon bubble disturbed by a shock wave. (The data set is courtesy of The Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, see http://seesar.lbl.gov/ccse.) 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 with quadrilinear B-spline wavelets. We have used slicing for generation of Figure 10 and volume rendering for the creation of Figure 11.

In Figure 10, we show a $\sqrt[4]{2}$-subdivision hierarchy, generated in combination with quadrilinear B-spline wavelets, at three levels of downsampling. Figure 10 shows a slice through the volume for time step 196. Since this data set is not very large in spatial dimensions, downsampling by a factor of two in every spatial dimension can be sufficient, whereas further downsampling in the temporal dimension may be desired. The fact that our four-dimensional

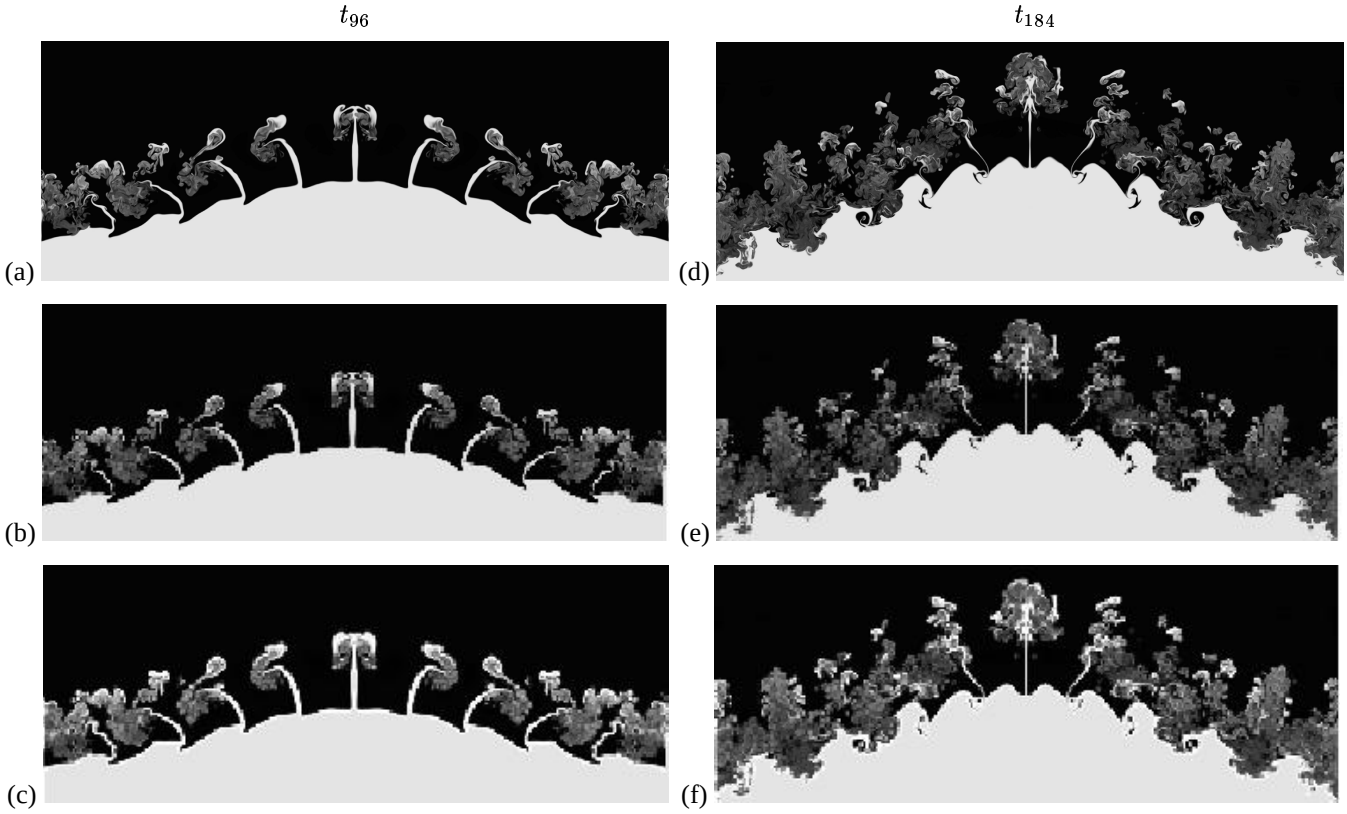


Figure 7. Entropy of a time-varying simulation of Richtmyer-Meshkov instability, visualized by a slice for two time steps of the original data, shown in (a) and (d), after $\sqrt[4]{2}$-subdivision downsampling without linear B-spline encoding, shown in (b) and (e), and after $\sqrt[4]{2}$-subdivision downsampling with linear B-spline encoding, shown in (c) and (f).

wavelet scheme is decomposed into a four-, three-, two-, and one-dimensional step allows us to integrate linear B-spline wavelet schemes of any dimension into one framework. For example, having downsampled the data set in four dimensions with the $\sqrt[4]{2}$-subdivision scheme combined with quadrilinear B-spline wavelets, we can continue to downsample only in the temporal dimension by using one-dimensional downsampling combined with linear B-spline wavelets.

Considering Figure 11(b), we have performed a $\sqrt[4]{2}$-subdivision downsampling combined with quadrilinear B-spline wavelets down to a level of detail with downsampling ratio 2^4, followed by one-dimensional downsampling steps with linear B-spline wavelets down to a level of detail with (total) downsampling ratio 2^7. One can compare this result to the one obtained when downsampling without wavelets, see Figure 11(a). Both pictures are the results of applying volume rendering to time step 192. Figure 11(a) only shows data from time step 192, whereas Figure 11(b) contains information of a short sequence of time steps close to time step 192, including all possibly significant changes.

7 Conclusions

We have introduced $\sqrt[4]{2}$ subdivision combined with quadrilinear B-spline wavelets for time-varying volume data representation. The approach provides a multiresolution hierarchy for four-dimensional data sets, where time is the fourth dimension. Temporal and spatial dimensions are treated equally in one framework.

The multiresolution data organization based on the $\sqrt[4]{2}$-subdivision scheme provides fine granularity by only doubling the overall number of data points in each subdivision step. In contrast, a generalization of an octree refinement to four dimensions increases the overall number of data points by a factor of 16 in each refinement step.

By integrating a wavelet scheme into the subdivision approach, we obtain, in general, much better approximations on each level of detail. We have developed a lifting scheme for quadrilinear B-spline wavelets. The lifting scheme uses narrow masks. This fact makes it possible to utilize the wavelet scheme for view-dependent, adaptive multiresolution visualization and facilitates out-of-core data explora-

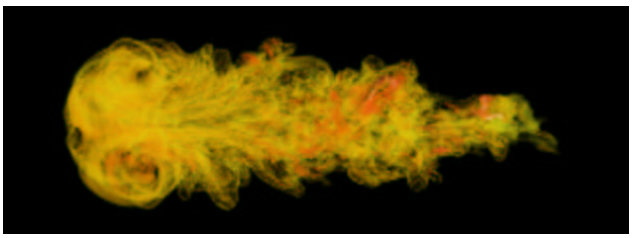
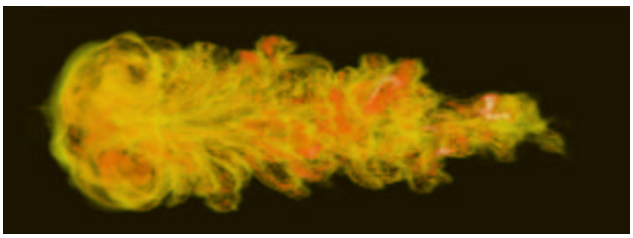
(a)  (b) 

Figure 11. Density of a time-varying simulation of an interaction of a shock with an argon bubble, visualized by volume rendering time step 192. Combined $\sqrt[4]{2}$-subdivision hierarchy in four dimensions and one dimension with downsampling ratio 2^7 without (a) and with (b) linear B-spline wavelets.

(a) 

(b) 

(c) 

Figure 10. Density of a time-varying simulation of an interaction of a shock with an argon bubble, visualized by slicing the volume data for time step 196. Shown are levels of the $\sqrt[4]{2}$-subdivision hierarchy with quadrilinear B-spline wavelet encoding for downsampling ratios 2^0 (a), 2^4 (b), and 2^8 (c).

tion techniques.

The wavelet encoding reorganizes data such that spatial locality of data belonging to the same level of detail is provided, which speeds up data access. No additional memory is required. The $\sqrt[4]{2}$-subdivision scheme also does not require us to store additional connectivity information.

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. Since the masks are narrow and linear B-spline wavelet operations are decomposed into only two steps, run time constants are small. We have applied our approach to large-scale time-varying data sets by using out-of-core techniques and combined our approach with various visualization methods. Considering the shown examples and the computed approximation errors, we conclude that our approach provides a valuable tool for hierarchical representation of time-varying volume data.

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, and the members of the Data Science Group at the Center for Applied Scientific Computing (CASC) at the Lawrence Livermore National Laboratory. We especially thank Oliver Kreylos for supplying us with his volume renderer.

References

[1] D. Anagnostou, T. J. Atherton, and A. E. Waterfall. 4d volume rendering with the shear warp factorization. In R. Crawfis and D. 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] M. Bertram. *Multiresolution Modeling for Scientific Visualization*. PhD thesis, Department of Computer Science, University of California, Davis, California, 2000.

[3] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representa-

tion and visualization. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings of IEEE Conference on Visualization 2000*, pages 389–396. IEEE, IEEE Computer Society Press, 2000.

[4] M. Bertram, D. E. Laney, M. A. Duchaineau, C. D. Hansen, B. Hamann, and K. I. Joy. Wavelet representation of contour sets. In T. Ertl, K. I. Joy, and A. Varshney, editors, *Proceedings of IEEE Conference on Visualization 2001*, pages 303–310. IEEE, IEEE Computer Society Press, 2001.

[5] P. Cignoni, C. Montani, E. Puppo, and R. Scopigno. Multiresolution modeling and visualization of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):352–369, 1997.

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

[7] T. Gerstner and R. Pajarola. Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings of IEEE Conference on Visualization 2000*, pages 259–266. IEEE, IEEE Computer Society Press, 2000.

[8] R. Grosso and G. Greiner. Hierarchical meshes for volume data. In F.-E. Wolter and N. M. Patrikalakis, editors, *Proceedings of CGI '98, Hanover, Germany*, 1998.

[9] R. Grosso, C. Lürig, and T. 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.

[10] L. Kettner and J. Snoeyink. Video: A prototype system for visualizing time-dependent volume data. In *In Video Proceedings of the 17th ACM Symposium on Computational Geometry*, 2001.

[11] L. Kobbelt. $\sqrt{3}$-subdivision. In K. Akeley, editor, *Proceedings of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 103–112. ACM, ACM Press / ACM SIGGRAPH, 2000.

[12] J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Transactions on Image Processing*, 9(3):480–496, 1999.

[13] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbf{r}^n$. *IEEE Transactions on Information Theory*, 38(2):533–555, 1992.

[14] L. Linsen, J. T. Gray, V. Pascucci, M. A. Duchaineau, B. Hamann, and K. 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.

[15] 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.

[16] K.-L. Ma and D. M. Camp. High performance visualization of time-varying volume data over a wide-area network. In R. Oldehoeft, editor, *Proceedings of Supercomputing 2000*. ACM, ACM Press, 2000.

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

[18] A. A. Mirin, R. H. Cohen, B. C. Curtis, W. P. Dannevik, A. M. Dimits, M. A. Duchaineau, D. E. Eliason, D. R. Schikore, S. E. Anderson, D. H. Porter, and P. R. Woodward. Very high resolution simulation of compressible turbulence on the ibm-sp system. In S. Howe, editor, *Proceedings of Supercomputing '99*. IEEE, IEEE Computer Society Press, 1999.

[19] M. Ohlberger and M. Rumpf. Hierarchical and adaptive visualization on nested grids. *Computing*, 59:365–385, 1997.

[20] S. Park, C. Bajaj, and I. Ihm. Effective visualization of very large oceanography time-varying volume dataset. *CS / TICAM Technical Report, University of Texas at Austin*, 2001.

[21] V. Pascucci. Slow growing subdivision (sgs) in any dimension: towards removing the curse of dimensionality. In *to appear in: Proceedings of Eurographics 2002*. 2002.

[22] V. Pascucci and C. Bajaj. Time critical adaptive refinement and smoothing. In R. Crawfis and D. Cohen-Or, editors, *Proceedings of the ACM/IEEE Volume Visualization and Graphics Symposium 2000, Salt Lake City, Utah*, pages 33–42. ACM/IEEE, 2000.

[23] D. Pinskiy, E. Brugger, H. R. Childs, and B. 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.

[24] H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and B-spline Techniques*. Springer-Verlag, Heidelberg, Germany, 2002.

[25] S. D. Riemenschneider and Z. Shen. Wavelets and pre-wavelets in low dimensions. *Journal Approximation Theory*, 71:18–38, 1992.

[26] R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill. Octree-based decimation of marching cubes surfaces. In R. Yagel and G. M. Nielson, editors, *Proceedings of IEEE Conference on Visualization 1997*, pages 335–342. IEEE, IEEE Computer Society Press, 1996.

[27] H.-W. Shen. Isosurface extraction from time-varying fields using a temporal hierarchical index tree. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *Proceedings of IEEE Conference on Visualization 1998*, pages 159–166. IEEE, IEEE Computer Society Press, 1998.

[28] H.-W. Shen, L.-J. Chiang, and K.-L. Ma. A fast volume rendering algorithm for time-varying fields using a time-space partitioning (tsp) tree. In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings of IEEE Conference on Visualization 1999*, pages 371–378. IEEE, IEEE Computer Society Press, 1999.

[29] H.-W. Shen and C. R. Johnson. Differential volume rendering: A fast algorithm for flow animation. In R. D. Bergeron and A. E. Kaufman, editors, *Proceedings of IEEE Conference on Visualization 1994*, pages 180–187. IEEE, IEEE Computer Society Press, 1994.

[30] E. J. Stollnitz, T. D. DeRose, and D. 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.

[31] P. Sutton and C. D. Hanson. Isosurface extraction in time-varying fields using a temporal branch-on-need tree (t-bon). In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings of IEEE Conference on Visualization 1999*, pages 147–153. IEEE, IEEE Computer Society Press, 1999.

[32] W. 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.

[33] M. Tory, N. Röber, T. Möller, A. Center, and M. S. Atkins. 4d space-time techniques: A medical imaging case study. In T. Ertl, K. Joy, and A. Varshney, editors, *Proceedings of IEEE Conference on Visualization 2001*, pages 473–476. IEEE, IEEE Computer Society Press, 2001.

[34] G. Uytterhoeven. *Wavelets: Software and Applications*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1999.

[35] L. Velho and D. Zorin. 4-8 subdivision. *Computer-Aided Geometric Design*, 18(5):397–427, 2001.

[36] J. Warren and H. Weimer. *Subdivision Methods for Geometric Design*. Morgan Kaufmann Publishers, San Francisco, U.S.A., 2002.

[37] R. Westermann, L. Kobbelt, and T. Ertl. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer*, pages 100–111, 1999.

[38] Y. Zhou, B. Chen, and A. E. Kaufman. Multiresolution tetrahedral framework for visualizing regular 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.