UCRL-PROC-208699

Wavelet-based multiresolution with n-th-root-of-2 Subdivision

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

December 20, 2004

Dagstuhl Seminar on Geometric Modeling
Dagstuhl , Germany
May 1, 2002 through May 6, 2002

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

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

Lars Linsen[1] Valerio Pascucci[2] Mark A. Duchaineau[2]
Bernd Hamann[1] Kenneth I. Joy[1]

[1] Center for Image Processing and
Integrated Computing (CIPIC)
Department of Computer Science
University of California, Davis
Davis, CA 95616-8562*

[2] Center for Applied Scientific
Computing (CASC)
Data Science Group
Lawrence Livermore National Laboratory
Livermore, CA 94550-9234†

Abstract

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.

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

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

1 Introduction

Multiresolution methods can be classified into regular and irregular ones according to 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 visualization of large 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 $\sqrt[n]{2}$-subdivision scheme that only doubles the number of vertices in each subdivision step regardless of dimension, which is a factor of $\sqrt[n]{2}$ in each of the n dimensions. This fact implies that the $\sqrt[n]{2}$-subdivision scheme, in general, will require less vertices to satisfy error bounds. In Section 2, we describe the $\sqrt[n]{2}$-subdivision scheme in general and provide more detail for up to four dimensions.

For time-varying volume representation, multiresolution representations with scalability in time and space need to be considered. The approach by Shen et al. [9] 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 $\sqrt[4]{2}$-subdivision hierarchy for representing time-varying volume data, we treat time as a real fourth dimension, dealing with spatial and temporal dimensions equally. Compared to the TSP approach, we have finer granularity, which is especially desirable in the spatial dimension.

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 and produce better approximation quality on coarser levels, we use a downsampling filter based on n-variate B-spline wavelets.

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 3, we describe a one-dimensional lifting scheme for B-spline wavelets applicable to multiresolution polygonal representations of curves. In Sections 4, 6, and 8, we describe how this approach can be generalized to the 2D, 3D, and 4D settings of multiresolution representations generated by $\sqrt[n]{2}$ subdivision (with $n = 2, 3, 4$). In Sections 5, 7, and 9, 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 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 were described by Cohen and Daubechies [2] for $n = 2$ and Maubach [6] 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). Velho and Zorin [12] completed the $\sqrt{2}$-subdivision scheme by adding an averaging step to the splitting step.

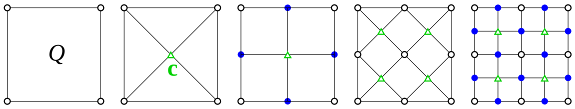

Figure 1: $\sqrt{2}$ subdivision.

This subdivision scheme can be generalized to arbitrary dimension. The splitting step of the $\sqrt[n]{2}$ 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 [8] for further details.

Figure 2 shows three $\sqrt[3]{2}$-subdivision splitting steps ($\alpha = 1$) for structured rectilinear volume data. Three kinds of polyhedral shapes (octahedron, octahedron with split faces, and cuboid) arise.

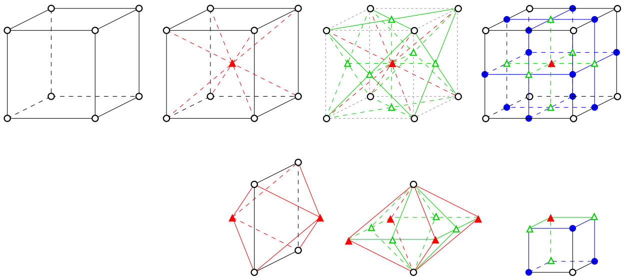

Figure 2: $\sqrt[3]{2}$ subdivision (upper row) and created polyhedral shapes (lower row).

With respect to the start configuration (first picture of Figure 2), the three subdivision steps are shown in Figure 2: The first step inserts the centroid of the cuboid, the second step inserts the centers of the faces of the original cuboid, and the third step inserts the midpoints of the edges of the original cuboid.

In order to generate a four-dimensional hierarchy, we start with a hypercube (or hypercuboid). In Figure 3, four $\sqrt[4]{2}$-subdivision steps ($\alpha = 1$) are shown.[1]

Figure 3(a) shows the initial hypercuboid, which consists of two cuboids at two time steps, t_1 and t_3. The first subdivision step inserts the centroid of the hypercuboid (Figure 3(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 (Figure 3(c)), the third step inserts the centers of their faces (Figure 3(d)), and the fourth step inserts the midpoints of their edges (Figure 3(e)). The geometric structure shown in Figure 3(e) consists of 16 hypercuboids.

3 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

[1] Although the hypercube is symmetric in all four dimensions, we use illustrations where the hypercube is stretched in the temporal dimension. We only show the spatial connectivities within the time steps and omit the connectivity information between time steps.

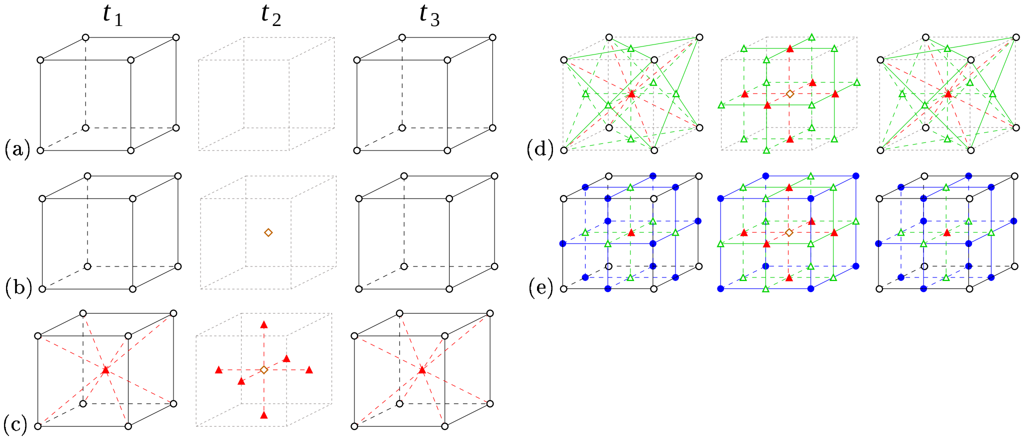

Figure 3: $\sqrt[4]{2}$ subdivision.

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. [10] described how to generate wavelets for subdivision schemes.

A family of filters can be derived by using B-splines of various degrees for wavelet generation. 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 [11] decompose wavelet computations into several steps, but they assert narrow filters.

The idea of a lifting scheme is shown in Figure 4, 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 $\mathcal{L}_{n-1}$ (*even* vertices) and the ones that belong to $\mathcal{L}_n \setminus \mathcal{L}_{n-1}$ (*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.

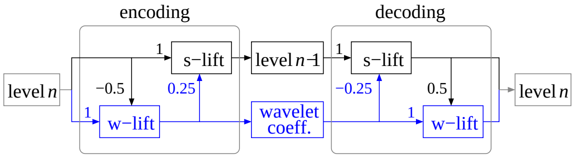

Figure 4: One-dimensional linear B-spline wavelet lifting scheme.

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

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

$$\text{w-lift}(a,b): \quad \left(\begin{array}{ccc} a & b & a \end{array} \right). \tag{2}$$

Using the s-lift and w-lift masks, a linear B-spline wavelet encoding step is defined by sequentially executing the two operations w-lift$(-\frac{1}{2},1)$ and s-lift$(\frac{1}{4},1)$. A linear B-spline wavelet decoding step is defined by executing the inverse operations in reverse order, i. e., s-lift$(-\frac{1}{4},1)$ and w-lift$(\frac{1}{2},1)$.

Using the same masks, a cubic B-spline wavelet encoding step is defined by the three lifting operations s-lift$(-\frac{1}{2},2)$, w-lift$(-\frac{1}{2},1)$, and s-lift$(\frac{3}{8},1)$. 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 [1].

4 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.,

$$\left(\begin{array}{ccc} a & b & a \end{array} \right) * \left(\begin{array}{c} a \\ b \\ a \end{array} \right) = \left(\begin{array}{ccc} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{array} \right). \tag{3}$$

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 △ of old faces;

at the midpoints • 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.

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 • (see third picture of Figure 1). Since the mask (3) requires data at the positions •, we compute them by linearly interpolating the values at the vertices ∘. Linear interpolation is appropriate, since we are using linear wavelets. This approach changes mask (3) to

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

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

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

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 • have vanished, but the s-lift encoding operation (theoretically) executed the 1D mask to update the vertices •. Hence, the values at the vertices • are given by linear interpolation of the values at the neighbor vertices △, multiplied by the factor $2a$ of the 1D mask. We rename the factor a to $\overline{a}$ and obtain the new mask

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

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

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

5 Surfaces

In Figure 5, we provide an example for a $\sqrt{2}$-subdivision hierarchy combined with 2D wavelet filters. The original surface shown in Figure 5(a) results from sampling a 2D Gaussian function at 64^2 vertices. The surface is encoded and decoded again. In Figure 5(b), we show a coarse level of detail obtained by applying $\sqrt{2}$-subdivision wavelets filters. In Figure 5(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.

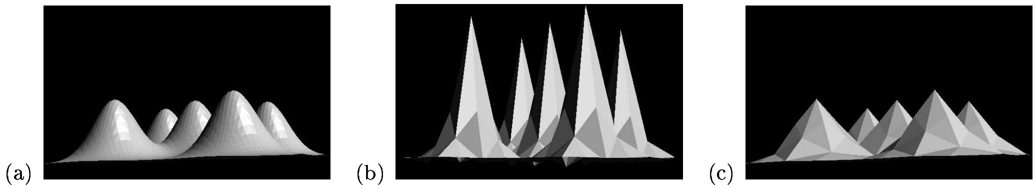

Figure 5: (a) $\sqrt{2}$-subdivision surfaces; (b) encoded and decoded by $\sqrt{2}$-subdivision wavelets; and (c) by bilinear B-spline wavelets.

In Figure 5(b), obvious over- and undershoots caused by the $\sqrt{2}$-subdivision wavelet filters can be recognized. We also observed less pronounced over- and undershoot when using bicubic B-spline wavelet filters. Only bilinear B-spline wavelet filters ensure no over- and undershoots, see Figure 5(c), since linear B-spline wavelets have interpolating scaling functions, which guarantees interpolating refinement filters [4]. 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.

Moreover, the 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.

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

In a $\sqrt[3]{2}$-subdivision hierarchy, three different kinds of polygonal shapes appear. 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 2 (*volume case*), proceed with the situation shown in the third picture of Figure 2 (*face case*), and finally treat the situation shown in the fourth picture of Figure 2 (*edge case*), which is topologically equivalent to the situation shown in the first picture of Figure 2.

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 2, we have no data values available at the vertices ● and △ (see fourth picture of Figure 2).

Again, we assume 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), and 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). One obtains the encoding w-lift mask depicted in Figure 6.

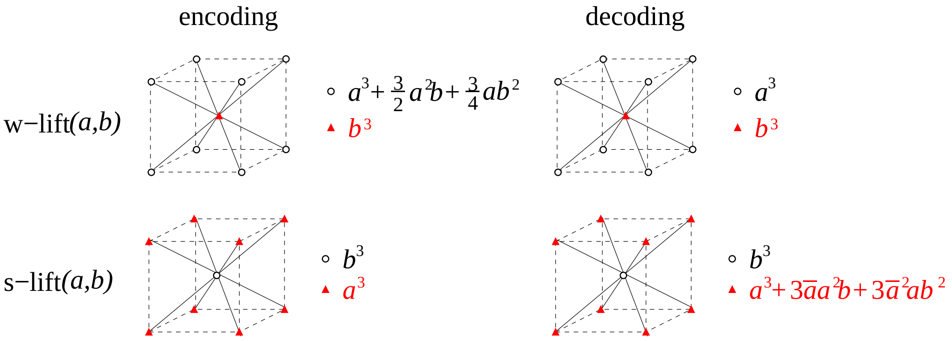

Figure 6: 3D lifting masks.

Proceeeding analogously to the 2D case, we can derive the encoding s-lift mask and the decosing masks shown in 6.

Face case. In the next $\sqrt[3]{2}$-subdivision step, we have to deal with the situation shown in the third picture of Figure 2. We have to ensure that we do not violate the assumptions made for the volume case that the values at the vertices △ are bilinear interpolations of the values at the neighbor vertices $\circ$. Thus, when the values at the vertices △ 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 4.

Edge case. When applying linear B-spline wavelet encoding to the situation illustrated in the fourth picture of Figure 2, 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 3.

All masks are as narrow as they can be. The face and edge cases naturally cover boundary faces and boundary edges of the domain.

7 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.[2] It is given as 753 slices, and each slice has a resolution of 1050 × 970 points, where 24-bit RGB-color information is stored. 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.

Figure 7(a) shows a part of a cutting plane at highest resolution, Figure 7(b) after downsampling to a resolution of 1.6% with $\sqrt[3]{2}$ subdivision, and Figure 7(c) after downsampling to a resolution of 1.6% with $\sqrt[3]{2}$ subdivision and trilinear B-spline wavelet filters.

Compared to Figure 7(b), the contours of the brain in Figure 7(c) are

[2] Data set courtesy of A. Toga, Ahmanson-Lovelace Brain Mapping Center, University of California, Los Angeles

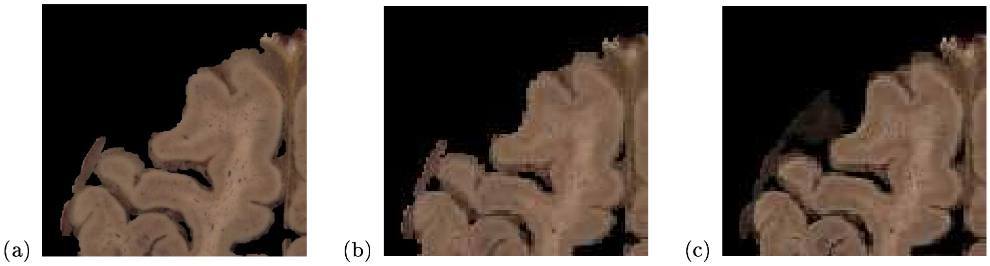

Figure 7: (a) Slice through 3D brain data set at full resolution; (b) slice at resolution of 1.6% without and (c) with B-spline wavelet filters on a $\sqrt[3]{2}$-subdivision scheme.

much smoother. Moreover, the slice in Figure 7(c) does not only contain information of the slice in Figure 7(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.

For the generation of Figure 8, 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 [7]. 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.

When using the wavelet approach, low-resolution visualizations suffice to see where turbulent mixing takes place. For example, Figure 8(c) shows clearly the big "bubble" rising in the middle of the data set. The bubble can hardly be seen in Figure 8(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 [3, 5]. In Figure 9, 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.[3] Figure 9(a) shows the adaptively refined

[3] Data set courtesy of Erik R. Wisner, Department of Radiology, University of Califor-

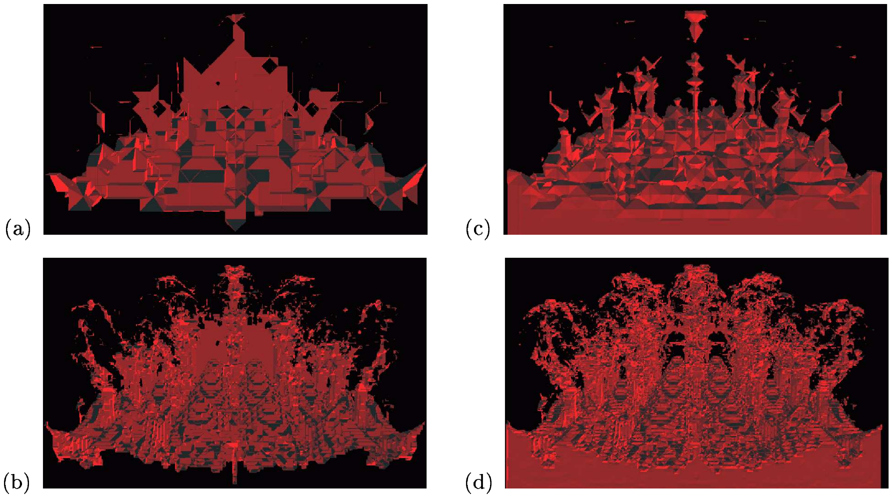

Figure 8: Entropy in a 3D simulation of Richtmyer-Meshkov instability, visualized by isosurface extraction from a $\sqrt[3]{2}$-subdivision hierarchy without (left column) and with (right column) B-spline wavelet filters (at resolutions of 0.003% and 0.2%).

mesh, where the viewpoint is positioned at the center of the right face of the bounding box. Figure 9(b) shows an isosurface extracted from the adaptively refined mesh for isovalue 86, chosen from the interval $[0, 255]$.

8 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 3. We start with the situation shown in Figure 3(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 3(b) by assuming (i) that the value at a vertex ● is defined by linear interpolation of the values at the two vertices $\circ$ (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 $\circ$ (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 $\circ$ (with which the vertex ▲ shares a cuboid). Consequently, one

nia, Davis

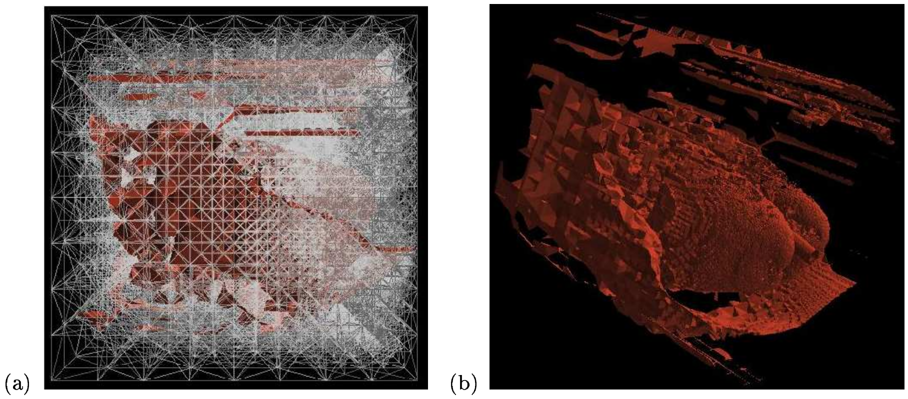

Figure 9: View-dependent visualization of lung data set using an adaptively refined $\sqrt[3]{2}$-subdivision mesh and B-spline wavelet filters.

obtains the mask w-lift$_{encode}(a, b)$ shown in Figure 10. We proceed as for the lower-dimensional cases and obtain the masks s-lift$_{encode}(a, b)$, s-lift$_{decode}(a, b)$, where $\overline{a}$ is the parameter a from the s-lift encoding mask, and w-lift$_{encode}(a, b)$, see Figure 10.

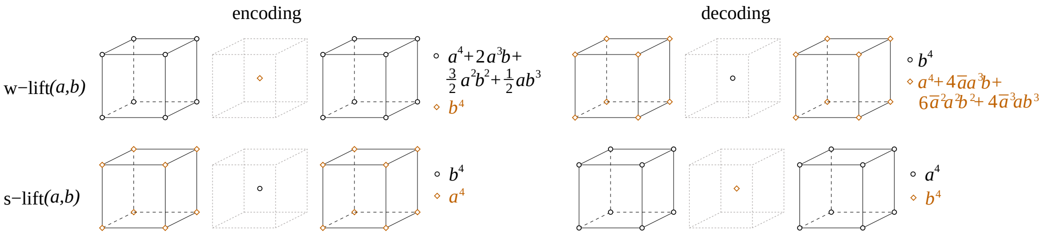
encoding
decoding
w−lift(a,b)
s−lift(a,b)
○ $a^4+2a^3b+\frac{3}{2}a^2b^2+\frac{1}{2}ab^3$
◇ b^4
○ b^4
◇ $a^4+4\overline{a}a^3b+6\overline{a}^2a^2b^2+4\overline{a}^3ab^3$
○ b^4
◇ a^4
○ a^4
◇ b^4

Figure 10: 4D lifting masks.

When inserting the vertices ▲, the vertices △, and the vertices ● as done in the situations in Figure 3(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.

9 Time-varying volume data

The time-varying volume data used for the example shown in Figure 11 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 640×256×256 rectilinear grid. For each vertex, a density value between 0 and 255 is stored.[4] We have constructed a $\sqrt[4]{2}$-subdivision hierarchy combined with quadrilinear B-spline wavelet filters. For visualizing the results, we have used volume rendering.

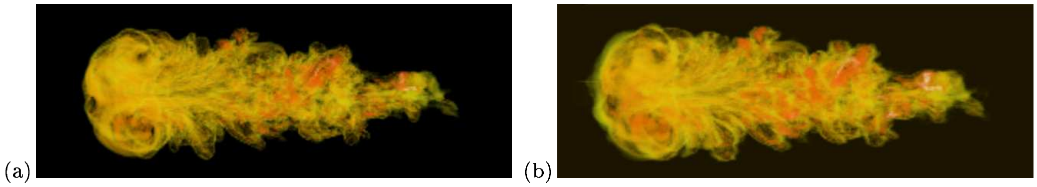

Figure 11: Volume rendering of density at time step 192 of Argon bubble simulation. Downsampled to 0.78% using combined $\sqrt[4]{2}$-subdivision hierarchy in four dimensions and one dimension without (a) and with (b) linear B-spline wavelets.

Considering Figure 11(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 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.

To quantify the improvement in approximation quality, we computed approximation errors for coarser levels of approximation by comparing them to the highest-resolution level. Figure 12 shows the average improvement for various levels of resolution (in a logarithmic scale) when applying quadrilinear B-spline wavelet filters.

[4] Data set courtesy of The Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory

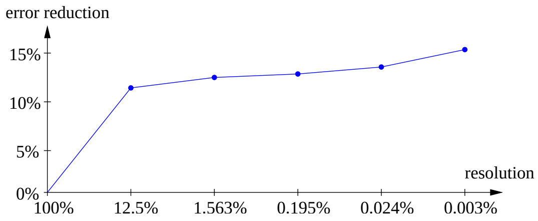

Figure 12: Average root-mean-square error reduction when applying linear B-spline wavelet filters.

10 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] 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.

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

[3] 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.

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

[5] Lars Linsen, Bernd Hamann, and Kenneth I. Joy. Wavelets for adaptively refined $\sqrt[3]{2}$-subdivision meshes. In *Proceedings of Sixth IASTED International Conference on Computers, Graphics, And Imaging (CGIM 2003).* The International Association of Science and Technology for Development (IASTED), 2003.

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

[7] 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.

[8] Valerio Pascucci. Slow growing subdivision (sgs) in any dimension: towards removing the curse of dimensionality. In *Proceedings of Eurographics 2002.* COMPUTER GRAPHICS Forum, 2002.

[9] 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.

[10] 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.

[11] 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.

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