Multiresolution and Hierarchical Methods for the Visualization of Volume Data

T. Ertl, R. Westermann, R. Grosso

Computer Graphics Group, Universität Erlangen-Nürnberg

Abstract

As three-dimensional data sets resulting from simulations or measurements become available at ever growing sizes the need for visualization tools which allow the inspection and the analysis of these data sets at interactive rates is increasing. One way to deal with the complexity is the compression of the data in such a way that the number of cells which have to be processed by the visualization mapping is reduced. Since this compression will be lossy, it is up to the user to choose between quality or speed. The decision will usually be made interactively requiring fast access to a complete hierarchy of representations of the data set at various levels of resolution. Two different approaches and visualization algorithms based upon them are presented in this paper: wavelet analysis deriving a hierarchy of coarser representations from the original data set and multilevel finite elements generating successively refined tetrahedral grids from an initially coarse triangulation.

1 Introduction

Scientific visualization is the process of generating a visual representation of the information contained in abstract data fields resulting from computer simulations or sensoric measurements. The standard model of this process comprises a pipeline of three stages. The *filter* stage is a preprocessing step converting the raw input data into visualization data which is usually reduced by operations like sampling, slicing, cropping, etc. The *mapper* stage performs a mapping of the abstract data fields into a visual representation consisting of geometric primitives like points, lines, surfaces or voxels and associated graphical attributes like color, transparency, texture, etc. The *renderer*, finally, uses this scene description to generate images by means of 3D graphics APIs such as OpenGL or OpenInventor, possibly exploiting 3D graphics hardware to achieve interactive frame rates. Many different mapping algorithms have been developed for various scenarios. A crude classification of these methods distinguishes between the dimensionality of the data set, the underlying data type,

Preprint submitted to Elsevier Preprint *4 August 2000*

such as scalar, vector, multivariate, and the supported grid structure, such as regular, curvi-linear or unstructured. In this paper we focus on scalar 3D data sets on Cartesian grids, although some of the ideas involve unstructured grids and can be extended to vectorial data sets.

Since volumetric data sets are intrinsically huge, a lot of efforts have been undertaken during the last years to come up with optimized visualization algorithms. The goal is to develop algorithms which react to changes of mapping parameters (e.g. varying the iso-value) by regenerating within seconds the corresponding geometrical representation which can then be rendered with several frames per second. Only with this type of real-time interaction and navigation is it possible to analyze an unknown data set and to compensate for the information lost during the projection of the 3D scene onto the screen. However, despite all the sophistication incorporated into these methods, does it seem that the data sets are growing faster than algorithmic progress is made. For example data volumes from 3D medical imaging like CT are approaching sizes of 512^3 which amounts to more than 100 million of voxel cells. It is obvious that visualization methods which essentially have to access each cell of a data set in order to derive a visual mapping might not catch up to the goal of interactive processing. Thus, we have to reduce the number of cells which have to be visually mapped, which means that we have to compress the data set in a preprocessing filtering step. Since we strive to reduce the number of cells by at least an order of magnitude only lossy compression schemes will be employed. However, this does not always have to lead to a significant loss of information. On the contrary, the compression scheme will be chosen in such a way, that only redundant or irrelevant information (e.g. CT voxels containing air) is discarded, while important features like high gradients, edges, etc. are retained or even emphasized.

Nevertheless, if an error is introduced in such a scheme, the user has to be given control over the threshold letting him choose between a fast visualization of a very crude approximation of the data and an almost perfect representation of the data which took perhaps minutes to compute. This requirement can only be met if not only one compressed version of the data, but a complete hierarchy of representations of the data set at different levels of resolution is available or can be generated on the fly. There are various ways to derive such a multiresolution hierarchy and this paper will focus on two of them which are well suited for scalar volumes on Cartesian grids: wavelet analysis and multilevel finite elements.

Developing efficient algorithms for generating a multiresolution hierarchy of a volumetric data set by compressing redundant information with respect to an error measure is only one side of the visualization pipeline. On the mapping side we need just as efficient algorithms which can take advantage of the hierarchical data structures. It turns out that this is by no means a trivial task as in general the traversal of a hierarchy is slow compared to full grid algorithms which have been optimized over years. However, ideally, a mapping algorithm exploiting the hierarchical representation of the data fields will automatically generate geometrical representations at various levels of detail, thus allowing incremental and progressive rendering. With respect to such adaptive visualization algorithms we will focus on the two fundamental techniques for scalar volumes: iso-surfaces and direct volume rendering.

The interactive visualization of very large datasets requires multiresolution analysis and hierarchical methods, new and highly sophisticated, possibly parallelized mapping algorithms, but also fast rendering of the resulting complex 3D scenes. One important aspect of accelerating the rendering phase, besides hardware support and scene graph optimization, is the simplification of polygonal meshes and the associated level-of-detail (LOD) generation, both of which have attracted significant attention in recent research (for a survey see [45]). While there definitely is a certain overlap in ideas, methods, and vocabulary with the topic of this paper, we like to point out some of the differences. Many of the proposed methods for mesh reduction are geometric in nature, which is reasonable because the goal is a coarse approximation of a given geometry. Usually, every single node of the mesh is analyzed with some local geometric error criterion like curvature or Hausdorff distance in order to decide whether it can be eliminated followed by a re-triangulation. In contrast, in volume simplification, we try to generate approximations of data sets which we consider to be samples of a continuous function. Thus, we prefer a functional approach which derives a hierarchy of approximations of the underlying function based on functional error measures like the L_2 norm.

The rest of the paper is organized as follows. In section 2 we will introduce some of the basic concepts and discuss related work. Our own contributions to hierarchical volume visualization are presented in section 3 based on multiresolution analysis and in section 4 based on adaptive mesh optimization with multilevel finite elements. Some conclusions and ideas for future work are summarized in section 5.

2 Related Work

2.1 Volume Visualization Algorithms

The problem of rendering visual representations of scalar volume data is still one of the main research areas of scientific visualization [46]. Besides a wide number of variations there are basically two main classes of techniques: iso-surfaces and direct volume rendering [12].

Isosurfaces are an indirect way of visualizing a scalar volume. An opaque surface which passes through all cells which include a specific function value only represents one aspect of the data set, since volume features at other values are ignored. This is a good approach for objects with sharply determined borders (like bones in CT), where the illumination of the surface greatly enhances its 3D structure, but it is inadequate for amorphous objects with small value gradients which can hardly be represented by mathematically thin surfaces. The structure of a unknown data set, e.g. of a smoothly varying function, can only be understood if surfaces can be extracted for many iso-values and viewed from all directions at interactive rates. The standard Marching Cubes algorithm [31] traverses all cells and determines the triangulation within each cell based on trilinear interpolation of the values of the cell vertices. Special treatment of ambiguities is required to avoid inconsistencies visible as holes [40]. While the generated polygonal iso-surfaces seem to be an appropriate volume visualization technique when using graphics workstations, it turns out, that this is not the case for data set sizes typically found in medical applications. Here, the surface extraction takes on the order of minutes and generates up to a million triangles and more which both severely restrict interactive manipulation. Various methods to deal with these problems include discretized algorithms [36], efficient cell search with interval data structures [30] and polygon reduction [48].

Direct volume rendering tries to convey a visual impression of the complete 3D data set by assigning different color and opacity values to different objects or value ranges within the volume. The resulting image is then computed by taking into account the so defined emission and absorption effects as seen by an outside viewer. The underlying theory of the physics of light transport is simplified to the well known volume rendering integral in the case of neglecting scattering and frequency effects [23,24]. Given the emission q and the absorption κ the intensity I along the ray s can be computed from:

$$I(s) = \int_{s_0}^{s} q(s') e^{-\int_{s}^{s'} \kappa(s'')ds''} ds'$$

The discretization of this integral together with the assumption that the mapping from scalars to RGBA values can be described by transfer functions results in the compositing formulas for computing the intensity contribution along one ray of sight:

$$I = \sum_{k=1}^{n} C_k \alpha_k \prod_{i=0}^{k-1} (1 - \alpha_i)$$

The color of the voxel C_k and its opacity α_k are derived by a table lookup after tri-linear interpolation of the scalar value from the discrete sample points.

Again, a few standard algorithms exist enhanced by a wide variety of optimization strategies. The basic ray tracing idea [27] is to shoot a ray of sight through every pixel into the volume, reconstructing the function value at appropriately chosen sample points along the ray and blending the mapped color and opacity values. Acceleration of this expensive technique is achieved by adaptive sampling [11,9], by exploiting coherence [25], by parallelizing in image and object space, and by exploiting hardware in graphics workstations [3] or in special purpose architectures [43].

2.2 Hierarchical Approaches

The first hierarchical approaches applied in volume visualization were based on octrees. The basic idea is to recursively split a cubed volume into eight sub-volumes working bottom-up from the original data set and to store additional information at each node which allows to skip uninteresting parts of the volume. The octree is seldomly used for the mapping itself, i.e. the iso-surface extraction, because neighboring octree leaves at different levels of resolution exhibit hanging nodes which lead to interpolation discontinuities.

Levoy [28] exploited a pyramid of binary volumes to encode the presence of non-transparent material in order to accelerate ray tracing through empty space. Laur and Hanrahan [26] stored averaged RGBA information as well as an error indicator in the nodes of a pyramid in order to eventually stop the traversal before splatting cells with only marginal contributions to the final image. Wilhelms and van Geldern [54] accelerated the marching cubes algorithm for iso-surface extraction by speeding up the search for cells which are passed by the surface. At each level of an octree they store the minimum and maximum value of all the vertices beneath allowing a fast skip over regions which cannot contain the surface. The simplicity of the pyramidal structure still attracts researchers. Ghavamnia and Yang [13] use a simple Laplacian pyramid to compress a volume data set and to reconstruct the voxel values on the fly while integrating along the ray. Haley and Blake [21] employ an octree to reduce the memory overhead of the shear-warp volume rendering approach.

So far the hierarchical decomposition was space-based, i.e the decision to which subtree a voxel belongs was derived from the vertex coordinates. Alternatively, range-based approaches classify the voxels according to the range of the scalar values present in the cell [49]. The resulting span space is again accessed through hierarchical data structures: Livnat et al. [30] use *kd*-trees, Cignoni et al. [7] adopt an interval tree .

Although we focus in this paper on volumes originally given on regular grids we will see in section 4 that the conversion of such a data set into an unstructured i.e. a tetrahedral representation provides many advantages. The most important aspect here is the possibility of adaptive refinement without introducing hanging nodes. Thus, we will also look shortly at related work in this area. Cignoni [8] gives a good overview of tetrahedra based volume visualization. He also introduced a multiresolution algorithm for simplical complexes by iteratively inserting vertices and performing a Delauney triangulation [6]. Lürig [32] extended this basically geometric idea into a two-step procedure: structural or reconstruction analysis and tetrahedrization. Rumpf et al. [39,42,47] describe hierarchical and adaptive visualization algorithms on nested grids which are the result of modern numerical simulations like hierarchical finite element and multi-grid methods. They derive an error indicator which relates the gradient of the function to the curvature of an iso-surface, giving an estimate of the lower and upper bounds of the function for the hierarchical traversal.

2.3 Multiresolution Analysis and Wavelets

One benefit of a hierarchical data representation is the possibility to localize features at increasingly coarser resolution. Unnecessary evaluations of data samples can be avoided by appropriately selecting the desired level of detail. The error that is introduced by reconstructing the signal from a certain resolution level strongly reflects in the algorithm used to generate the hierarchy.

In typical octree based approaches data samples are simply averaged and every other sample is pushed up to the next level in the tree. Thus, the resolution at each level is decimated by a factor of two thereby increasing the size of details which can be reconstructed accordingly.

Formally, this is equivalent to projecting the original signal into a hierarchy of appropriately scaled piece-wise constant basis functions. Then, the underlying function basis is the Haar system, which in the univariate case is built from scales and dilates of the box function $\Phi(x)$ defined to be 1 on the interval $[0,1)$ and 0 otherwise. Since the Haar family $\Phi_k^j(x) = 2^{j/2}\Phi(2^j x - k)$ defines an orthonormal basis of $L_2(\mathbb{R})$, arbitrary functions $f \in L_2(\mathbb{R})$ can be written as linear combinations of the basis functions: $f(x) = \sum_{j,k}\langle f, \Phi_k^j\rangle\Phi_k^j$. Additionally, since the Haar functions are self-similar in the sense that $\Phi(x-k) = \Phi(2(x-k)) + \Phi(2(x-k)-1)$, the vector space spanned by translates of $\Phi(x)$ is contained in the vector space spanned by translates of $\Phi(2x)$. If only coefficients $\langle f, \Phi_k^j\rangle$ up to a certain level J are used to reconstruct the original function f, then an approximation of f with no details smaller than 2^{-J} is generated.

Despite the advantages of the Haar series it can only be used to a limited extend for the analysis of arbitrary signals. Particularly this representation does not allow for a precise localization of features in the frequency domain due to the infinite support of the box function's spectrum. Wavelet theory and multiresolution analysis provide a theoretical framework for the hierarchical decomposition of signals into properly designed function bases which enable the localization of signals in both time and space.

By a wavelet we will mean a function Ψ which has exponential fast decay at infinity and for which $\int_{-\infty}^{+\infty} \Psi(x)dx = 0$. For a family $\Psi_k^j(x)$ to define a basis of $L_2(\mathbb{R})$ it is build from scales and dilates of Ψ in such a way that any function $f \in L_2(\mathbb{R})$ has a series expansion of the form $f(x) = \sum_{j,k} c_{j,k}\Psi_k^j(x)$. The coefficients $c_{j,k}$ represent the wavelet transform of f at level j. If the family $\Psi_k^j(x)$ provides an orthogonal basis then the coefficients are obtained by computing the inner products $\langle f, \Psi_k^j(x)\rangle$ between f and Ψ. Throughout this paper we take the translations to be integer values and the scales to be of the form 2^j, $j \in \mathbb{Z}$, so that Ψ has the form $\Psi_k^j = 2^{j/2}\Psi(2^jx - k)$.

Multiresolution analysis (MRA) [5,10] comes up with a general theory to derive wavelets $\Psi \in L_2(\mathbb{R})$ from a so called scaling function Φ. A MRA of $L_2(\mathbb{R})$ is defined as a sequence of closed subspaces $V_j \subset L_2(\mathbb{R})$, $j \in \mathbb{Z}$, with $V_j \subset V_{j+1}$ and $\bigcup_{j=-\infty}^{+\infty} V_j$ is dense in $L_2(\mathbb{R})$, and a unique scaling function $\Phi \in L_2(\mathbb{R})$. Again, the family $\Phi_k^j(x) = 2^{j/2}\Phi(2^jx - k), k \in Z$, is derived by translating and dilating Φ. For a fixed j each series generates a basis of V_j. To each of the vector spaces V_j there exists the complementary vector space W_j with $V_{j+1} = W_j \oplus V_j$. The sequence W_j defines the so called difference spaces, in which all the information is retained that is lost from one approximation space V_j to the next coarser one (see Figure 1). The wavelets Ψ_j are derived from Ψ in such a way that they span the difference spaces W_j. Since both functions Ψ_j and Φ_j are $\in V_{j+1}$ they can be expanded in the basis Φ_{j+1}. Sub-band filter

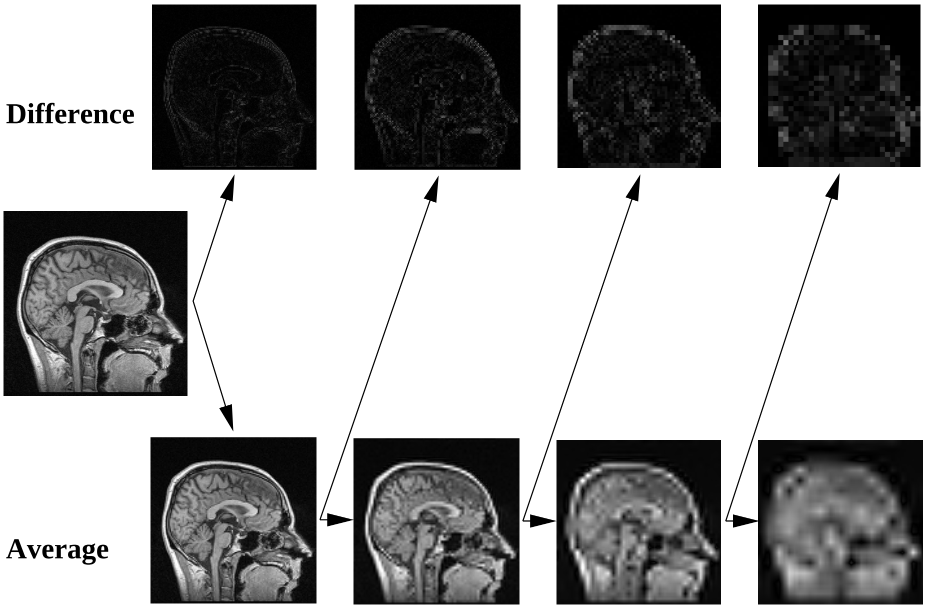

Fig. 1. Original signal and it's reconstruction from different multiresolution spaces.

pairs h and g describe exactly how to get from the projection into V_{j+1} to the projections into V_j and W_j and vice versa. The filter h is used to select input frequency components up to a band-pass frequency, whereas g selects frequencies from V_j which are larger than this frequency. Usually, the pyramid algorithm [35] is applied to perform the projection in linear operations with the number of function samples.

Especially in the case of a separable MRA the theory can be extended straight forwardly to higher dimensions. In three dimensions we construct scaling functions of the form $\Phi^j_{l,m,n}(x,y,z) = \Phi^j_l(x)\Phi^j_m(y)\Phi^j_n(z)$. Three dimensional wavelets are build by taking all other products of scaling functions and wavelets at a certain scale, thus providing a basis of $L_2(\mathbb{R}^3)$.

Although an infinite number of wavelets exist which can be distinguished based on certain properties only some of them are of major relevance in volume visualization. The smaller the support of the basis functions the less coefficients have to be considered for the reconstruction of a function value. Smoothness of the involved basis functions corresponds to better frequency localization of the used filters. And finally, the expected compression ratios strongly depend on the number of vanishing moments provided by the used wavelets. This is defined to be the largest integer number p for which all integrals $\langle x^n \rangle_\Psi = \int_{-\infty}^{\infty} \Psi(x) x^n dx, n = 0, \ldots, p-1$ vanish, and it thus determines the convergence rate of wavelet coefficients.

3 Multiscale Volume Visualization and Segmentation

3.1 Compression Domain Volume Rendering

Due to the approximation properties of wavelets, the expansion of discrete signals into a wavelet basis result in sparse representations. It is thus of particular interest to study wavelet transforms in the field of volume visualization where we have to deal with intrinsically large data sets.

The core ideas for exploiting wavelet transforms in volume rendering applications have first been introduced by Muraki [37]. Thus, this work can be seen as the forerunner to a variety of wavelet based techniques [51,29,14] specifically designed to take advantage of the hierarchical nature of wavelet transforms and the resulting properties. Particularly in volume rendering applications wavelet techniques show up as a remedy to locally reconstruct the original signal within arbitrary approximation spaces. As a consequence, the volume rendering integral can be directly evaluated on the signal now expanded into a wavelet basis. Encoding the data in its entirety is avoided thus providing an effective way to render large scale data sets on a compressed domain.

Instead of solving the volume rendering integral over the original domain the signal is first projected into the sequence of difference spaces $W_1, \ldots, W_J$ and the final smooth approximation space V_J. Consequently only an approximation to the final pixel intensity is obtained. The accuracy of the integral evaluation strongly depends on the size of details present in the subspaces from which the density distribution is reconstructed. By expressing the signal as a linear combination of wavelet coefficients at different scales the integration takes place only where coefficients don't vanish and where the influence of the corresponding basis functions is not zero. The more sparse the representation is the less coefficients have to be involved. Compression domain rendering of large scale data sets as well as progressive rendering and transmission is accomplished easily by exploiting the multiresolution representation.

Apparently it is not possible to directly expand the exponential attenuation term into a wavelet series since it depends on the direction of the line of sight. However, it acts as a smoothing operator which does not introduce high frequencies. Additionally, since both terms, the emission and the absorption, are derived from the available density distribution, it seems to be fairly acceptable to correlate the accuracy of the integration process to the distribution originally provided. Even more efficiently, when absorption effects are completely neglected, also an analytic solution of the integral based on a wavelet expansion of the volume emission can be found [14]. This is achieved by constructing piecewise polynomial splines with knot sequences directly derived from the wavelet coefficients.

Solving the integral as described exhibits several desired features. Particularly, in rather smooth areas only a limited number of coefficients have to be considered since they are small in magnitude. This is due to the approximation convergence determined by the number of vanishing moments the wavelet provides. Additionally, when orthonormal wavelets are used the L_2 error introduced by neglecting certain coefficients can be determined, but more accurately, as a consequence of the spatial localization of the coefficients also an upper bound of the pixel-wise error accumulated along a particular ray can be predicted. Furthermore, the adaptive nature of the wavelet representation is considered by adjusting the integration step size to the scale of the underlying basis functions. In regions where only low frequency details are present the step size is increased accordingly whereas a finer sampling frequency is chosen in regions of high density variations.

As can be seen in Figure 2, wavelets provide a powerful tool for compressing large scale volume data and for visualizing them in an adaptive manner thereby controlling the introduced error. Faster rendering can be achieved using object space approaches [29] where weighted footprints of wavelet coefficients, so

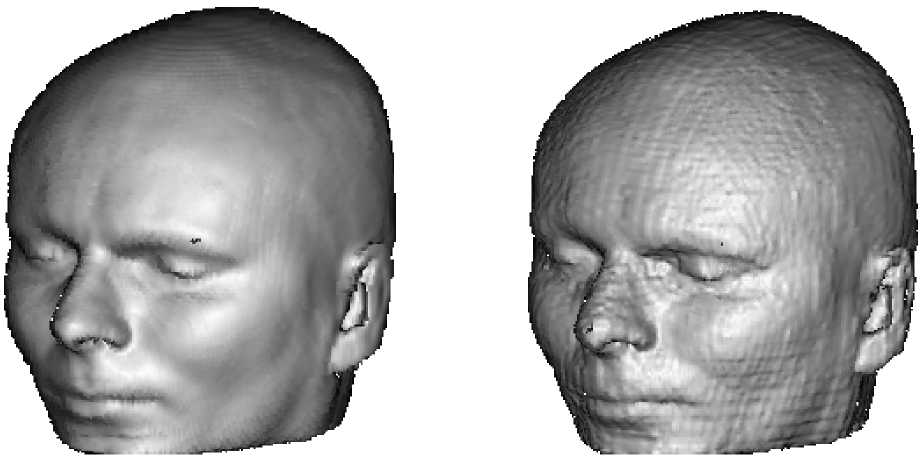

Fig. 2. Volume ray-casting of an iso-surface from a human head MRI-Scan. On the left the surface was rendered from the original data set with 32 MB. On the right the same surface was rendered from the compressed data set with 0.58 MB.

called wavelet splats, are projected to the image plane. The advantage of these techniques shows up in applications where progressive transmission with respect to the importance of wavelet coefficients is requested. This is particular useful in client-server applications where on-the-fly visualization at a coarse resolution level is often required for preview purposes [15]. Efficiently coding the hierarchical representation and building data structures allowing for fast access [17] is one of the major challenges in these kinds of applications. However, in projective approaches attenuation effects have to be completely abandoned since the accurate ordering of coefficients with respect to the present viewing definition can not be achieved in an acceptable amount of time.

3.2 Structural Analysis

As outlined in the previous section, wavelet techniques can be effectively used to decompose the original signal into a hierarchy in which copies of the data at ever coarser scales are included and where the size of details maintained within these copies scales up accordingly. The hierarchical representation allows locating the significant structures of the original signal at arbitrary scales. Moreover, the signal's local characteristics can be determined quite accurately from the variation of these structures across scales.

One key technique often applied in image processing is to determine the local sharp variation points in a smoothed version of the original signal by detecting the zero crossings of the signal convolved with the second derivative of the smoothing function [33,4]. Small fluctuations are removed in the blurred version, while the significant structures are retained. Successively performing this procedure with appropriately scaled versions of the smoothing function yields the edges at multiple scales. Now a characterization of the relevant features can be obtained by combining the edge information across all levels.

Obviously, the accuracy by which edges can be detected strongly depends on the kernel functions used. Traditionally, a Gaussian kernel is employed. On the other hand, since it is the high frequency information in particular which is retained in the difference spaces of a MRA, wavelet transforms in general provide a powerful tool to localize and analyze the multiscale edges. In [38] DOG wavelets were exploited to decompose the signal into a hierarchy of 3D edges, which were then used to enhance the significant structures by appropriately blending edges from multiple scales. More generally, assuming θ to be the integral over a wavelet Ψ, $\theta(x) = \int_{-\infty}^{x} \Psi(t)\, dt$ and setting $\theta^j(x) = 2^{j/2}\theta(2^j x)$ and $\Psi^j(x) = 2^{j/2}\Psi(2^j x)$, then the wavelet transform at scale 2^j can be written as $W^j f(x) = f * \Psi^j(x) = f * (2^j \frac{d\theta^j}{dx}) = 2^j \frac{d}{dx}(f * \theta^j)(x)$. It is thus proportional to the derivative of the original signal smoothed by $\theta^j(x)$. As a result, the multiscale edges of f can be completely determined by repeatedly convolving Ψ^j with the smoothed version $f * \theta^j(x)$ and by locating the local maxima of $W^j f(x)$ [34].

By constructing derivative wavelets $\Psi_1(\vec{x}), \Psi_2(\vec{x})$ and $\Psi_3(\vec{x})$ which are the partial derivatives of a three dimensional smoothing function $\theta(\vec{x})$ along x, y and z, respectively, the approach can be lifted to higher dimensions. This procedure was successfully applied to images [34], and extended to discrete scalar volume data in [53,20]. Similar to the one dimensional case, the gradient vector at scale 2^j can then be written as

$$\vec{\nabla}(f * \theta^j)(\vec{x}) = \begin{pmatrix} W_1^j f(\vec{x}) \\ W_2^j f(\vec{x}) \\ W_3^j f(\vec{x}) \end{pmatrix} = 2^j \begin{pmatrix} \frac{\partial}{\partial x}(f * \theta^j)(\vec{x}) \\ \frac{\partial}{\partial y}(f * \theta^j)(\vec{x}) \\ \frac{\partial}{\partial z}(f * \theta^j)(\vec{x}) \end{pmatrix}.$$

By computing the magnitude from the squares of the gradient components we are now ready to determine those points where the gradient magnitude is locally maximum along the direction the gradient is pointing to. In practice, however, since it is only possible to consider a finite number of directions spanned by $[f * \Psi_1(\vec{x}), f * \Psi_2(\vec{x}), f * \Psi_3(\vec{x})]$ a discretization has to be applied.

Although the significant structures are maintained in the difference spaces across the multiscale hierarchy, the entropy of the decoded information slightly decreases since the gradient maxima are computed from the low frequent parts. Less significant structures disappear at one of the coarser scales. On the other hand, it can be shown that from the evolution of the gradient maxima across scales the original signal can be locally characterized. In particular, it is possible to estimate the signal's smoothness, which is commonly measured by the Lipschitz regularity from the asymptotic decay of the wavelet coefficients to the finest scale [22]. For this method to proceed properly, those maxima which proceed from a certain scale to the next coarser one have to be separated. By connecting neighboring locations across scales so called maxima chains are constructed. By the decay, then, of the magnitude of wavelet coefficients along these chains the smoothness of the original signal at points the chains are pointing to can be determined.

In [20,52,53] these procedures were applied to determine the sharp variations of features in stationary 3D volume data and in time-resolved sequences. The extracted information can be efficiently used to enhance or suppress more or less significant structures. Furthermore, by adaptively traversing the data with respect to the multiscale representation of the 3D edges during the ray traversal the rendering process can be accelerated considerably. Figure 3 shows how this concept can applied to automatically render results which compare quite well to traditional segmentation techniques.

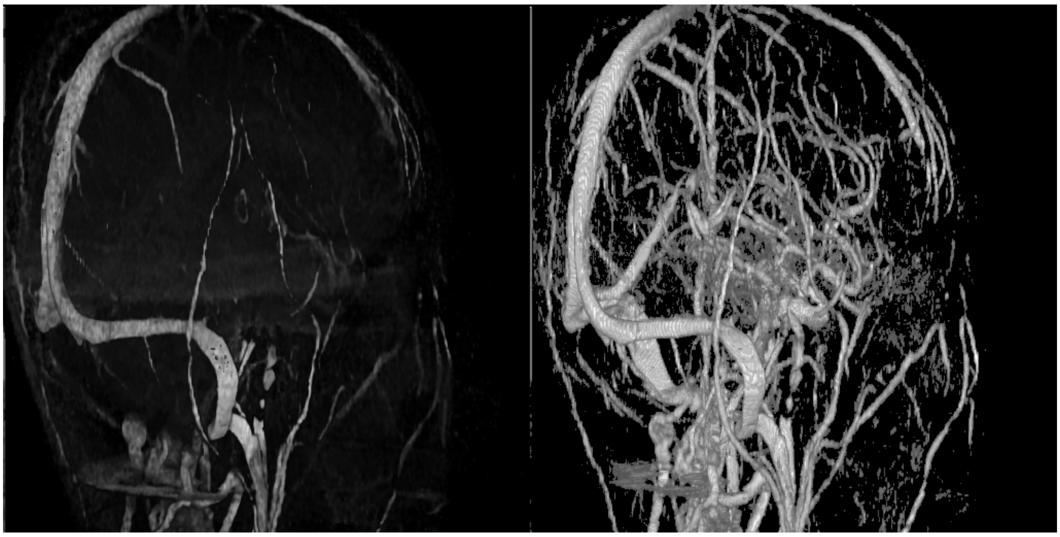

Fig. 3. Volume rendering of the vessel structures in a data set from brain angiography. The structural analysis leads to an automatic enhancement of the vessels and a significant noise reduction (right) compared to the standard linear transfer function (left).

4 Adaptive Mesh Optimization for Volume Visualization

It should have become obvious from the last section, that the wavelet based MRA exhibits many advantages in the area of signal analysis and compression, but that the evaluation of the hierarchy introduces some additional overhead.

Thus, we are looking for a representation of our data which consists of a hierarchy of *full* grids where we can apply the existing sophisticated algorithms on each level. It turns out that the numerical mathematics community has developed such an adaptive mesh optimization technique for the field of multilevel finite elements [1,55]. One important aspect of this approach for visualization is that it proceeds from a very coarse approximation to finer levels of resolution, stopping whenever a given error threshold is fulfilled. This is in contrast to the wavelet decomposition which always has to start at the finest level of resolution. In this section we will summarize our application of this technique to volume visualization [19].

4.1 Multilevel Finite Element Mesh Refinement

We start out with an intentionally very coarse triangulation of our domain Ω. Depending on the geometry of the domain it may be a complex task to generate such a grid, but for a cube this is the standard partition into five or six tetrahedra. We construct a N-dimensional finite element subspace $S \in L_2(\mathbb{R}^3)$ with basis functions $\{\phi_i\}_{i=1}^N$ consisting of all piecewise linear functions associated with the vertices and with their support restricted to the tetrahedra containing the vertex. Then we search for the best approximation $u \in S$ of the function $f \in L_2(\mathbb{R}^3)$ which is represented by our data set. This *least squares approximation* problem is reduced to the solution of a linear system $AU = F$ which is solved by a conjugate gradient method. The coefficients are given by $A_{ij} = \langle \phi_j, \phi_i \rangle_{L_2}$, $F_i = \langle f, \phi_i \rangle_{L_2}$ and $u = \sum U_i \phi_i$ with $\langle f, g \rangle_{L_2} = \int_\Omega f g dx$. The evaluation of the inner products and the assembling of the corresponding matrices and vectors are carried out using standard finite element techniques. The inner products $\langle f, \phi_i \rangle$ are computed using a five point third order integration formula. In order to evaluate the quadrature we use tri-linear interpolation of f from the original data. A global error of our best approximation can be found by summation of all the local errors $E_\Delta = \int_\Delta (f - u)^2 d\Omega$ computed in each tetrahedron. If the *local error* coefficients exceed a certain threshold the corresponding mesh elements are marked for refinement. After the error analysis, the mesh is refined and the iteration starts all over again determining the best approximation of the function on the new level. The algorithm terminates when the *global error* is less than a user-specified threshold.

At the core of the method is a local refinement algorithm which generates a hierarchy of adaptively refined meshes. Each triangulation in the sequence is required to be *conforming*, i.e. the intersection of two elements consists of a common face or a common edge or a common vertex or it is empty. This condition prevents *hanging nodes*, which are difficult to treat in finite element computations and which are problematic for rendering purposes. For the same reasons the triangulation sequence has to be *stable* with respect to some measure of degeneracy, e.g. all interior angles are bounded away from zero. Finally, in order to build a hierarchy of nested spaces the triangulation sequence has to satisfy the *nestedness* condition, which means that an element in a triangulation is obtained by subdividing an element in a coarser triangulation of the sequence.

We extend an algorithm which combines *regular* and *irregular* mesh refinement rules [2]. Elements which are marked for refinement are split regularly: a tetrahedron is cut into four tetrahedrons at the corners and one octahedron inside. Instead of subdividing the octahedron immediately into tetrahedra we treat them as a regular element type and define another regular refinement rule for them by which they are split into six octahedra and eight tetrahedra. In this way we avoid ambiguities and reduce the number of cells. The regular refinement rules can be applied to neighbor elements without consistency problems. In the case of adaptive refinement, however, only a subset of the given elements will be regularly refined. Thus, the partition has to be *closed* by irregular refinement of neighboring elements in order to avoid hanging nodes. We restrict irregular refinement to tetrahedra and if an octahedron has to be irregularly refined, it will be first subdivided into eight tetrahedra by connecting the vertices with the barycenter. All elements generated by the irregular refinement are constructed using the vertices introduced by the regular refinement of neighbor elements. We have implemented a *full set of refinement rules* which fits all possible edge and face refinement patterns. In order to avoid stability problems irregularly refined tetrahedra must not be refined again. If a subdivision is required, the originally refined tetrahedron must be re-refined with the regular rule. Finally, these local rules are combined and rearranged into a *global refinement algorithm* which guarantees for stability and conformity. Figure 4 demonstrates the algorithm and the quality of the generated meshes for a 2D example of a MRI slice of a medical data set.

4.2 Adaptive Volume Visualization

If the described algorithm is applied to a Cartesian scalar volume, an adaptively refined tetrahedral mesh is computed which is coarse in homogeneous regions of the volume but fine in regions with a strong variation of the underlying function, i.e. around edges or surfaces. Depending on the error threshold, which gives a measure of the quality of the approximation, we achieve a significant reduction in the number of cells compared to the original voxels and thus an enormous compression of our data set while retaining relevant features. We point out that the algorithm is progressive in the sense that the coarser the desired approximation is, the faster can it be computed. Furthermore, the amount of work to find the approximation to a certain error threshold is nearly independent of the data size, thus very large data sets can be processed as

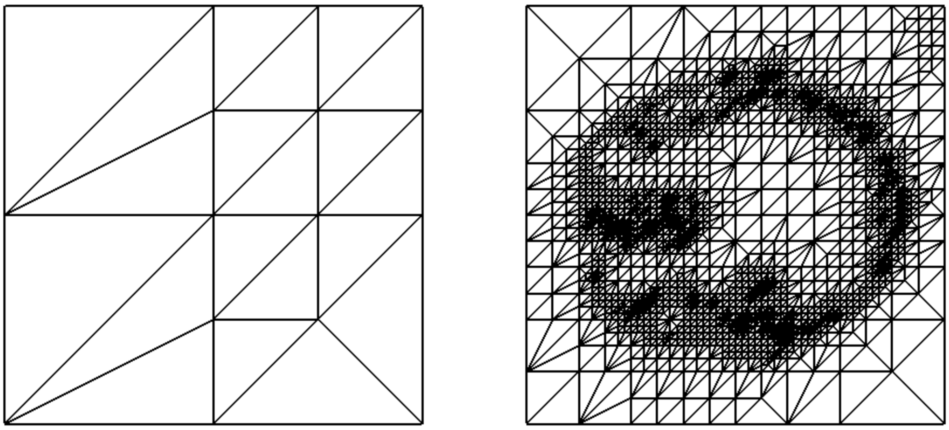

Fig. 4. Adaptive mesh refinement for a 512^2 medical data set. The left image shows regular refined triangles and the irregular closure at level 3, the right image demonstrates the quality of the triangulation.

long as they can be interpolated at any point in the domain.

Arbitrary visualization algorithms for unstructured meshes can be used to derive visual mappings from the approximated fields. They will be much faster than the visualization of the original data set for several reasons. First, the number of cells to be traversed will be reduced by at least one order of magnitude. This automatically reduces the number of output primitives. Second, no special algorithms are necessary, since we deal with full grids on each level of the hierarchy. Interpolation within a tetrahedron is linear and thus even simpler, but normal estimation is a little bit more involved. Third, the adaptivity of the mesh automatically leads to adaptive visualization algorithms. This is obvious for the search of iso-surface cells which proceeds fast through homogeneous regions with large tetrahedrons. For ray-casting we integrate from cell boundary to cell boundary which automatically leads to an adaptive step size.

Figure 5 shows iso-surfaces of a CT data set of a human abdomen. The original volume is $512^2 \times 181$ which is about 47 million voxels. A standard marching cubes algorithm runs for about a minute to generate an iso-surface consisting of 1,4 million triangles. In the left image an iso-surface with only 70,000 triangles is extracted from an adaptively refined mesh at level 5 (50,000 vertices) in 1.2 seconds. The right image shows the result for level 7 (180,000 vertices) with 340,000 triangles extracted in 4.4 seconds. All the relevant structures of the iso-surface are already clearly visible at level 5 and even lower levels would be sufficient for fast previewing. Comparing the surface at level 7 with the iso-surface extracted from the full resolution exhibits some approximation errors in the smoothness of the surface, however this was to be expected when reducing the number of cells to about 2% and this can be improved by incorporating the Sobolev norm which brings gradient information into the

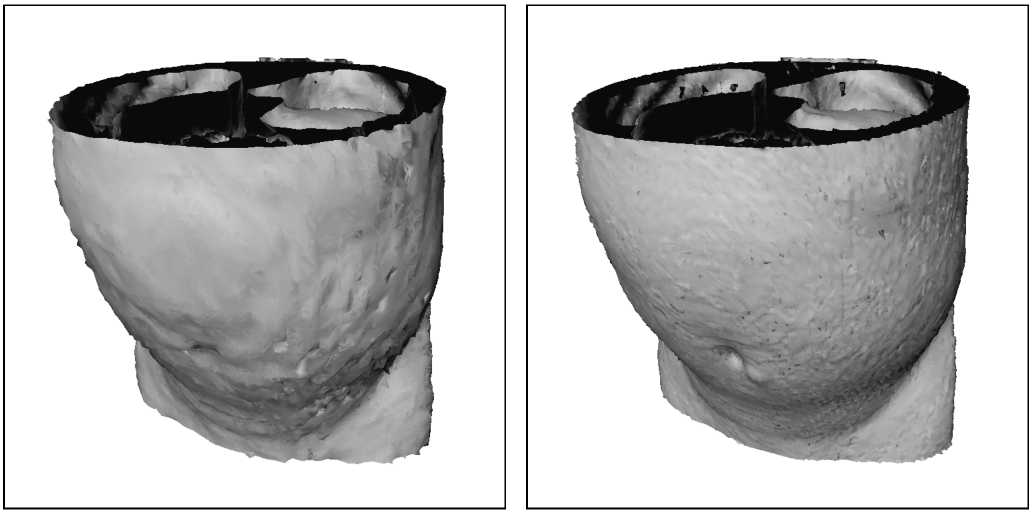

Fig. 5. Iso-surface extraction from an abdomen CT data set. The original 47 million voxels were compressed to adaptive tetrahedral meshes of various resolution (left level 5, right level 7) which allow the fast extraction of surfaces with less triangles.

approximation [18].

4.3 Progressive Iso-surface Extraction

So far, only one level of the multiresolution hierarchy of the volume was actually used during the visualization with standard algorithms for unstructured grids. We will now outline a new iso-surface algorithm which explicitly exploits the data hierarchy and generates iso-surfaces of varying resolution [16]. The scenario we are aiming at is the interactive visualization of huge data sets in a distributed environment where the user at the graphics workstation requests the multiresolution data set from a remote server. The server starts sending the coarsest resolution and incremental information on how to refine the mesh. The client begins with the reconstruction of the hierarchy from the data stream and at the same time with the iso-surface extraction on the already built levels. This iso-surface is locally refined as soon as higher levels of the volume hierarchy have been reconstructed up to a maximum level or a user interaction changes parameters like iso-value or resolution depth. The algorithm is based on a very efficient abstract representation of the mesh which consists of the coordinates of the vertices at the coarsest level, a set of integer indices describing the refinement, and the function values at the finest level of resolution. The abdomen data set refined up to level 7 can thus be stored in approximately 1 MB which corresponds to a compression factor of 50. The reconstruction of the hierarchy from this representation can be performed at interactive rates. This is in part due to the concept of virtual elements: irregularly refined elements, which are usually more than regularly refined elements, are not actually stored, but are computed on demand. Finally, the progressive iso-surface extraction which inserts and deletes only triangles resulting from a mesh refinement or coarsening also needs on the average less than one second when switching from one level to the next.

5 Conclusions

We have presented an overview of multiresolution and hierarchical methods for the visualization of volume data. Besides reviewing related work in this field we have described in more detail two complementary approaches: wavelet-based multiresolution analysis and finite-element-based adaptive mesh optimization. We have shown how compressed representations of the volume data can be exploited by a variety of visualization algorithms ranging from structural analysis and ray casting to progressive iso-surface extraction. The availability of increasingly huge data sets will promote exciting new research in the area of multiresolution modeling and visualization of volumes, similarly as it happened for polygonal models. We expect contributions from applying geometrical approaches to 3D [44], from extending wavelets to non-regular grids [41], and from other hierarchical bases representations like sparse grids [56,50].

References

[1] R. E. Bank. Hierarchical Bases and the Finite Element Method. *Acta Numerica*, pages 1–43, 1996.

[2] J. Bey. Tetrahedral Grid Refinement. *Computing*, 55(4):355–378, 1995.

[3] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. In A. Kaufman and W. Krüger, editors, *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, 1994.

[4] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(3):679–698, 1986.

[5] Chui, C. *Wavelet Analysis and its Applications I: An Introduction to Wavelets.* Academic Press, Inc., 1992.

[6] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution Modeling and Visualization of Volume Data based on Simplicial Complexes. In *1994 Symposium on Volume Visualization*, pages 19–26. ACM SIGGRAPH, 1994.

[7] P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno. Speeding Up Isosurface Extraction Using Interval Trees. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):158–170, 1997.

[8] P. Cignoni, C. Montani, and R. Scopigno. Tetrahedra Based Volume Visualization. In *Workshop Visualization and Mathematics 1997*, Berlin, 1997. in preparation.

[9] J. Danskin and P. Hanrahan. Fast Algorithms for Volume Ray Tracing. In *1992 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, 1992.

[10] Daubechies, Ingrid. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[11] B. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. *Computer Graphics (SIGGRAPH '88)*, 22(4):65–74, 1988.

[12] T.T. Elvins. A Survey of Algorithms for Volume Visualization. *Computer Graphics*, 26(3):194–201, 1992.

[13] M. Ghavamnia and X. Yang. Direct Rendering of Laplacian Pyramid Compressed Volume Data. In G.M. Nielson and D. Silver, editors, *Visualization '95*, pages 192–200. IEEE Computer Society Press, 1995.

[14] M. Gross, L. Lippert, A. Dreger, and R. Koch. A New Method to Approximate the Volume Rendering Equation Using Wavelets and Piecewise Polynomials. *Computers and Graphics*, 19(1):18–28, 1995.

[15] M. Gross, L. Lippert, and C. Kurmann. Compression Domain Volume Rendering for Distributed Environments. *Computers Graphics Forum (EUROGRAPHICS '97)*, 16(3):96–107, 1997.

[16] R. Grosso and T. Ertl. Progressive Iso-Surface Extraction from Hierarchical 3D Meshes. *Computers Graphics Forum (EUROGRAPHICS '98)*, 17(3), 1998.

[17] R. Grosso, T. Ertl, and J. Aschoff. Efficient Data Structures for Volume Rendering of Wavelet-Compressed Data. In N.M. Thalmann and V. Skala, editors, *WSCG '96 - The Fourth International Conference in Central Europe on Computer Graphics and Visualization*, volume I, pages 103–112, University of West Bohemia, Plzen, 1996.

[18] R. Grosso and G. Greiner. Multilevel Sobolev Approximations and Adaptive Mesh Reduction. In *Proc. Computer Graphics International 1998*, pages 761–771. IEEE Computer Society Press, 1998.

[19] R. Grosso, Ch. 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, *Visualization '97*, pages 387–394. IEEE Computer Society Press, 1997.

[20] B. Guo. A Multiscale Model for Structure-Based Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):291–301, 1995.

[21] M. Haley and H. Blake. Incremental volume rendering using hierarchical compression. *Computer Graphics Forum (EUROGRAPHICS '96)*, 15(3):45–55, 1996.

[22] S. Jaffarth. *Wavelets: Mathematics and Applications*, chapter Wavelets and Nonlinear Analysis, pages 467–505. ACM, 1993.

[23] J. T. Kajiya and B. P. Von Herzen. Ray Tracing Volume Densities. *Computer Graphics (SIGGRAPH '84)*, 18(3):165–174, 1984.

[24] Krüger, W. The Application of Transport Theory to the Visualization of 3-D Scalar Data Fields. In A. Kaufman, editor, *Visualization '90*, pages 273–280. IEEE Computer Society Press, 1990.

[25] P. Lacroute. Analysis of a Parallel Volume Rendering System Based on the Shear-Warp-Factorization. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):218–231, 1996.

[26] D. Laur and P. Hanrahan. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. *Computer Graphics (SIGGRAPH '91)*, 25(4):285–288, 1991.

[27] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[28] M. Levoy. Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics*, 9(3):245–261, 1990.

[29] L. Lippert and M. Gross. Fast Wavelet Based Volume Rendering by Accumulation of Transparent Texture Maps. *Computers Graphics Forum (EUROGRAPHICS '95)*, 14(3):432–443, 1995.

[30] Y. Livnat, H.-W. Shen, and C.R. Johnson. A Near Optimal Isosurface Extraction Algorithm using Span Space. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–84, 1996.

[31] W.E. Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (SIGGRAPH '87)*, 21(4):163–169, 1987.

[32] C. Lürig and T. Ertl. Adaptive Iso-Surface Generation. In B. Girod, H. Niemann, and H.-P. Seidel, editors, *3D Image Analysis and Synthesis '96*, pages 183–190. Graduiertenkolleg 3D Bildanalyse und Synthese, infix, 1996.

[33] D. Maar and E. Hildreth. Theory of Edge Detection. *Proceedings Royal Soc. London*, 207:187–217, 1980.

[34] S. Mallat and W. Hwang. Singularity Detection and Processing with Wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, 1992.

[35] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[36] C. Montani, R. Scateni, and R. Scopigno. Discretized Marching Cubes. In D. Bergeron and A. Kaufman, editors, *Visualization'94*, pages 281–287. IEEE Computer Society Press, 1994.

[37] S. Muraki. Volume Data and Wavelet Transforms. *IEEE Computer Graphics and Applications*, 13(4):50–56, 1993.

[38] S. Muraki. Multiscale 3D Edge Representation of Volume Data by a DOG Wavelet. In A. Kaufman and W. Krüger, editors, *1994 Symposium on Volume Visualization*, pages 35–42. ACM SIGGRAPH, 1994.

[39] R. Neubauer, M. Ohlberger, M. Rumpf, and R. Schwoerer. Efficient Visualization of Large-Scale Data on Hierarchical Meshes. In *Eighth Eurographics Workshop on Visualization in Scientific Computing*, pages 165–174, 1997.

[40] G. Nielson and B. Hamann. The Asymptodic Decider: Removing the Ambiguity in Marching Cubes. In G. Nielson and Rosenblum. L., editors, *Visualization '91*, pages 83–91. IEEE Computer Society Press, 1991.

[41] G. Nielson, L. Jung, and J. Sung. Haar-Wavelts over Triangular Domains. In R. Yagel and H. Hagen, editors, *Visualization '97*, pages 143–149. IEEE Computer Society Press, 1997.

[42] M. Ohlberger and M. Rumpf. Hierarchical and Adaptive Visualization on Nested Grids. *Computing*, 59 (4):269–285, 1997.

[43] H. Pfister and A. Kaufman. Cube-4 - A Scalable Architecture for Real-Time Volume Rendering. In R. Crawfis and Ch. Hansen, editors, *1996 Symposium on Volume Visualization*, pages 47–54. ACM SIGGRAPH, 1996.

[44] J. Popvic and H. Hoppe. Progressive Simplical Complexes. *Computer Graphics (SIGGRAPH '97)*, 31(4):217–224, 1997.

[45] E. Puppo and R. Scopigno. Simplification, LOD and Multiresolution - Principles and Applications. In *Eurographics 97 Tutorials.* Blackwell, 1997.

[46] L. J. Rosenblum. Research Issues in Scientific Visualization. *IEEE Computer Graphics and Applications*, 14(2):61–85, 1994.

[47] M. Rumpf, A. Schmidt, and K.G. Siebert. Functions Defining Arbitrary Meshes - A Flexible Interface between Numerical Data and Visualization. *Computer Graphics Forum*, 15(2):129–142, 1996.

[48] W. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of Triangle Meshes. *Computer Graphics (SIGGRAPH '92)*, 26(4):65–70, 1992.

[49] H. Shen and C. Johnson. Sweeping Simplices: A Fast Iso-Surface Extraction Algorithm for Unstructured Grids. In G.M. Nielson and D. Silver, editors, *Visualization '95*, pages 143–150. IEEE Computer Society Press, 1995.

[50] C. Teitzel, R. Grosso, and T. Ertl. Particle Tracing on Sparse Grids. In D. Bartz, editor, *Proc. 9th Eurographics Workshop on Visualization in Scientific Computing*, pages 132–142, 1998.

[51] R. Westermann. A Multiresolution Framework for Volume Rendering. In A. Kaufman and W. Krüger, editors, *1994 Symposium on Volume Visualization*, pages 51–58. ACM SIGGRAPH, 1994.

[52] R. Westermann. Compression Domain Rendering of Time-Resolved Volume Data. In G.M. Nielson and D. Silver, editors, *Visualization '95*, pages 51–58. IEEE Computer Society Press, 1995.

[53] R. Westermann and T. Ertl. A Multiscale Approach to Integrated Volume Segmentation and Rendering. *Computers Graphics Forum (EUROGRAPHICS '97)*, 16(3):96–107, 1997.

[54] J. Wilhelms and A. Van Gelder. Octrees for Faster Iso-surface Generation. In *ACM Transactions on Graphics*, pages 201–227, 1992.

[55] H. Yserentant. Hierarchical Bases. In R. E. O'Malley, editor, *ICIAM 91*. SIAM, 1992.

[56] Ch. Zenger. Sparse Grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, Proc. of the Sixth GAMM-Seminar Kiel, 1990, Vieweg, Braunschweig, Wiesbaden, Germany, 1991.