Adaptive Projection Operators in Multiresolution Scientific Visualization

Mario Ohlberger and Martin Rumpf

Abstract—Recently, multiresolution visualization methods have become an indispensable ingredient of real-time interactive postprocessing. The enormous databases, typically coming along with some hierarchical structure, are locally resolved on different levels of detail to achieve a significant savings of CPU and rendering time. Here, the method of adaptive projection and the corresponding operators on data functions, respectively, are introduced. They are defined and discussed as mathematically rigorous foundations for multiresolution data analysis. Keeping in mind data from efficient numerical multigrid methods, this approach applies to hierarchical nested grids consisting of elements which are any tensor product of simplices, generated recursively by an arbitrary, finite set of refinement rules from some coarse grid. The corresponding visualization algorithms, e.g., color shading on slices or isosurface rendering, are confined to an appropriate depth-first traversal of the grid hierarchy. A continuous projection of the data onto an adaptive, extracted subgrid is thereby calculated recursively. The presented concept covers different methods of local error measurement, time-dependent data which have to be interpolated from a sequence of key frames, and a tool for local data focusing. Furthermore, it allows for a continuous level of detail.

Index Terms—Adaptive projection operators, multiresolution, efficient data analysis, error indicators, hierarchical grids, visualization of large data sets.

1 Introduction

Today's computing hardware and the rapid development of efficient numerical algorithms allow the successively finer approximation of physical quantities in scientific computing. Sophisticated multigrid methods [30], [1], [24], [27], [47] are especially capable nowadays of resolving complex solution structures. In a postprocessing step, the user wants to interactively explore the corresponding large amount of data with typically millions of unknowns to improve his understanding of interesting features. The numerical methods are mostly based on a variety of domain discretizations, such as structured or unstructured Finite Difference, Finite Element, or Finite Volume grids, which are, in general, supplied with a natural hierarchical structure. The corresponding meshes may consist of a single or of mixed element types, e.g., simplicial, prismatic, rectangular, or cubic ones. The recursive generation of elements is, in general, described by a finite set of refinement rules. Furthermore, very often in the implementation of numerical methods, nonstandard and application dependent data structures are essential for an efficient implementation of the simulation algorithm.

Data analysis, especially by suitable visualization methods, is an indispensable tool to study and understand the simulation results. Typical basic tools are the drawing of isolines, respectively, the color-shading or texturing on 2D domains, on surfaces, or on arbitrary slices in 3D, and the rendering of isosurfaces in 3D. Efficiency of the visualization is requested to extract the required information from the enormous database at a high frame rate. A variety of multiresolution visualization methods has been designed to serve this purpose for certain grid types in two and three dimensions. For a more detailed overview, we refer to the next paragraph. These methods correspond to a specific local error measurement. The considered error type, e.g., in the L^∞, L^2 norm, or in terms of wavelet coefficients, is, in general, closely related to the physical problem underlying the simulation. If error indicator values are below a certain threshold, the algorithm locally stops on coarser levels of detail. Algorithmic effort is needed to avoid cracks in the resulting approximation of surface graphs, isolines, and isosurfaces, or jumps in color and texture values.

We present here a unified approach to multiresolution visualization on nested grids which covers multiple types of error indication and comes along with a robust and efficient solution of the above continuity problem for a large class of visualization applications. The approach is based on the definition of an appropriate *adaptive projection* of the considered discrete function. Throughout the paper, the term projection is always meant in the sense of the mathematical operation which can be applied to some function. Here, a discrete function, the data given on a Finite Element mesh, is projected onto an adaptive grid that consists of different levels of mesh elements. The projection is recursively defined on the grid hierarchy and depends on some error indicator given on the grid nodes. It is guaranteed to be continuous if a natural *saturation condition* is fulfilled by the error indicator. Various types of indicators are supported. One possibility is that the saturation condition may fail. This typically happens

EDITORS' NOTE: This paper originally appeared in TVCG, Vol. 4, No. 4. Unfortunately, due to an extraordinary problem that occurred at press time, numerous errors were introduced in the final printed copy. We are therefore reprinting the paper in its entirety in the present issue.

- *M. Ohlberger is with the Institut für Angewandte Mathematik, Universität Freiburg, Herrmann-Herder-Str. 10, 79104 Freiburg, Germany. E-mail: mario@mathematik.uni-freiburg.de.*
- *M. Rumpf is with the Institut für Angewandte Mathematik, Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany. E-mail: rumpf@iam.uni-bonn.de.*

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 108126.

on coarse grid levels. Therefore, in a preroll step, a slight modification is introduced which again "saturates" the indicator values. Furthermore, we point out that the use of hierarchical offset values as an error indicator in addition especially allows an estimation of min / max values on grid cells, which is essential, e.g., for hierarchical isosurface generation.

The concept applies to time-dependent data as well, where a finite number of key frames is given and, in between, some interpolation is used. It is explained how to interpolate error indicator values to obtain an appropriate adaptive projection at any time.

Furthermore, a *continuous level of detail*, in our concept, the continuity of the adaptive data projection with respect to the user prescribed threshold, can easily be obtained by a slight generalization of the projection criterion.

Finally, in case of Finite Element spaces of *higher-order polynomial degree*, the presented methods can be adapted to enable a "virtual," adaptive refinement of the finest grid cells to resolve the local function with arbitrary prescribed precision.

The paper is organized as follows: First, we review related work on multiresolution visualization in Section 1.1 and, in order to especially clarify the notation, we give a brief overview on nested grids and function spaces in Section 1.2. In Sections 2 and 3, we introduce adaptive data projections depending on some type of error measurement and explain a general adaptive visualization algorithm. Next, various types of projection criteria, respectively, error indicators, are discussed in Section 4. In Section 5, we derive additional projection criteria for geometric shapes, as well as time-dependent and vector-valued functions. In Section 6, we explain how to guarantee a continuous level of detail with respect to the user prescribed threshold value and Section 7 discusses adaptive projection in case of higher-order polynomial degree of the discrete function spaces. Although the discussion of algorithmic aspects is not this paper's main intention, in Section 8, we comment on some aspects concerning the implementation and, in Section 9, we discuss the efficiency of the presented approach and compare it with other methods. Finally, in Section 10, we draw conclusions.

Let us remark that, although the presented concept applies to 2D and 3D visualization applications, with a strong focus on the 3D case, most of the schematic figures deal with the 2D case. This is solely to simplify the presentation.

1.1 Related Work

As already described, improving the efficiency of visualization methods with respect to very large data sets in two and three dimensions is a key issue in recent research. A variety of applications, such as terrain visualization, surface modeling, medical imaging, and, especially, numerical simulations, deliver enormous amounts of data. An interactive exploration is indispensable to analyze the output, understand solution features, and modify input parameters. Multiresolutional techniques have proven to be the adequate solution for a large class of applications. Authors have approached them in a multitude of ways. Here, we give a brief and, naturally, incomplete overview.

The efficient rendering of height fields in geographic imaging, especially for flight simulation purposes, has been studied, e.g., by Certain et al. [7], Faust et al. [20], and De Floriani et al. [14]. Applying some hierarchical algorithm, they adaptively extract conforming triangular meshes from the underlying regular database. Data in the center of the typically moving viewpoint is thereby resolved finer than in outer areas. Errors are measured in the L^∞ norm, respectively, in pixels in image space. Gross and Staadt [25] consider a wavelet compressed data representation and use wavelet coefficients as an error indicator. Cohen-Or and Levanoni [12] study continuous level of detail rendering in case of Delaunay triangulated terrain.

For arbitrary triangular surfaces, e.g., isosurfaces in numerical data fields, surfaces generated by some 3D scanning process, or shapes in geometric modeling and adaptive coarsening strategies, have been presented by Turk [58], Hamann [28], and Schroeder et al. [53]. In a noninteractive preparatory step, requested surface reduction rates are achieved by successive elimination of vertices. Hoppe [31] introduced so-called progressive meshes, which allow an efficient complexity reduction and fast transmission of data over the net at any prescribed resolution. A conceptional overview on hierarchical triangulations is given by De Floriani and Puppo [15]. For a comparison of different mesh simplification algorithms, we refer to Cignoni et al. [11].

In 3D, the efficient storing and handling of hierarchical data coded in octrees has been studied by, among others, Gargantini [21], Williams [61], Tamminen and Samet [57], and Levoy [38]. Ghavamnia and Yang [22] have discussed how to address hierarchically compressed data in fast volume rendering.

Hierarchical searching for isosurfaces was first considered by Wilhelms and Van Gelder [59] on hexahedral meshes. They thereby speed up the classical marching cube algorithm introduced by Lorensen and Cline [40]. The grid elements are encoded in an octree, which allows the recursive search for isosurface intersections starting on the coarse grid elements. This method is especially efficient in case of smooth functions at the expense of extra storage for the min and max values on each node in the octree structure (cf. Section 4.4). The hierarchical approach competes with other efficient isosurface extraction methods which use some efficient presorting [23], [55], [39] or seed cell algorithms, such as the extremal graph methods by Itoh et al. [32], [33]. These approaches are preferable if the data is governed by high frequencies.

But, in contrast to these approaches, the hierarchical data access, as, for instance, in 2D, can be combined with an adaptive choice of the desired data resolution. A fast and adaptive visualization of volume data is implemented in the hierarchical splatting algorithm by Laur and Hanrahan [37]. They have used an L^2-type error indicator on an octree encoded voxel database to speed up rendering substantially. Wilhelms et al. [60] use such a hierarchical speed up in a scan-plane-type approach to volume rendering, especially for preview purposes. Cignoni et al. [8] have applied a successive adaptive refinement of volumes by Delaunay methods, which leads to nonnested hierarchical meshes. They discuss further issues in [9]. Additional points are successively inserted in areas where an L^∞–type error indicator measures differences above a certain threshold value. Adaptive isosur-

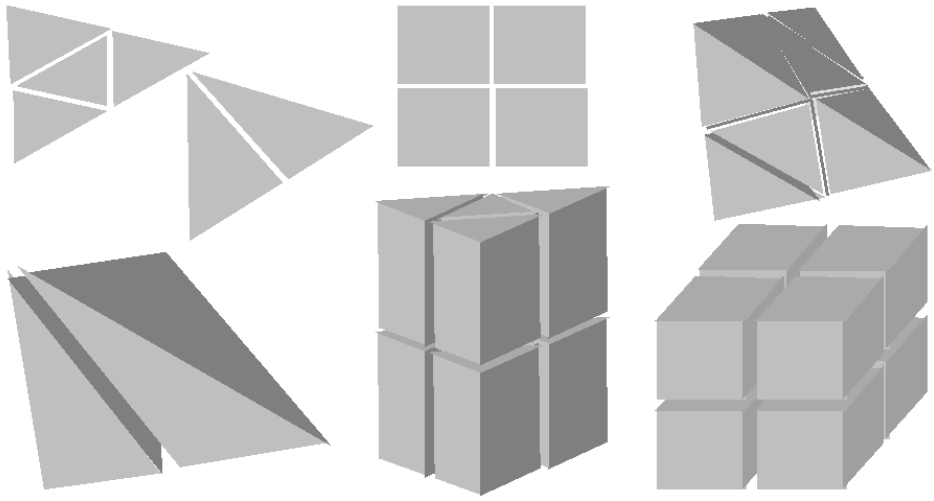

Fig. 1. Basic element types in two and three dimensions with possible refinements.

faces on regular data fields are treated by the octree based decimation algorithm presented by Shekhar et al. [54]. Based on error indicator values on the octree cells, the recursive tree traversal is stopped locally on coarser grid levels. They thereby enforce at most one-level transitions between cells on which they definitely draw local isosurfaces.

Different approaches have been presented to solve the outstanding continuity problem, e.g., to avoid cracks in adaptive isosurfaces. In the Delaunay approach by Cignoni et al. [8] and in the nested mesh method by Grosso et al. [26], the successive remeshing during the refinement guarantees the continuity. On the other hand, Shekhar et al. [54] rule out hanging nodes by inserting additional points on faces with a transition from finer to coarser elements due to an adaptive stopping criteria.

Any adaptive visualization algorithm in 2D and 3D is based on a specific local error measurement. Different approaches to measure errors have been considered. Often, the hierarchical data offset from one grid level to the next finer one is measured in the L^∞ norm (cf., e.g., [31], [54], Klein et al. [36]) to achieve higher reduction rates measuring surface distances in the more natural Hausdorff norm. Grosso et al. used L^2, respectively, $H^{1,2}$ [26], projections of regular field data onto adaptive unstructured meshes. Finally, wavelet coefficients are often appropriate error indicators. Among many other contributions, we here especially cite Gross and Staadt [25] and Certain et al. [7]. Bonneau et al. [5] studied a problem dependent blending of different wavelet based error indication. Eck et al. [17] and Schröder and Sweldens [52] have worked out multiresolution visualization methods on triangulated surfaces based on local error measurement in wavelet spaces. Compare Sections 4 and 5 for a comparison of different types of error measurement.

1.2 Data on Nested Grids

In the following section, we will discuss a general approach to adaptive projection methods based on nested grids. Let us, therefore, briefly introduce some basic concepts of nested grids and of function spaces defined on them. Let $\Sigma^m \subset \mathbb{R}^m$ be the set of simplices of dimension m, e.g., Σ^1 the set of line segments, Σ^2 the set of triangles, and Σ^3 the set of tetrahedrons. Here, we consider all elements $E \subset \mathbb{R}^n$ consisting of tensor products of simplices, i.e., for some integer k

$$E = \bigotimes_{i=1}^{k} \sigma_i,$$

with $\sigma_i \in \Sigma^{m_i}$, where m_i denotes the dimension of σ_i and k, $m_1, \cdots, m_k$ have to be chosen such that $\sum_{i=0}^{k} m_i = n$ (cf. [46] for the definition of simplices and tensor products).

Examples are triangles ($E = \Sigma^2$), rectangles ($E = \Sigma^1 \times \Sigma^1$) in 2D, and tetrahedrons ($E = \Sigma^3$), prisms ($E = \Sigma^2 \times \Sigma^1$), or hexahedrons ($E = \Sigma^1 \times \Sigma^1 \times \Sigma^1$) in 3D (cf. Fig. 1).

A conforming mesh $\mathcal{M}$ is a set of elements E such that any two elements of $\mathcal{M}$ are disjoint or they intersect in a boundary simplex, e.g., a common face, edge or vertex. A family of conforming meshes $\{\mathcal{M}^l\}_{0 \le l \le l_{\max}}$ is called a nested grid if, for all $E^{l+1} \in \mathcal{M}^{l+1}$, there exists an $E^l \in \mathcal{M}^l$ with $E^{l+1} \cap E^l = E^{l+1}$ and $\bigcup_{E^{l+1} \in M^{l+1}} E^{l+1} = \bigcup_{E^l \in M^l} E^l$. These kinds of grids are mostly recursively generated by refinement of certain elements of the preceding coarser mesh. Corresponding to a nested grid $\{\mathcal{M}^l\}_{0 \le l \le l_{\max}}$, we consider a family $\{\mathcal{V}^l\}_{0 \le l \le l_{\max}}$ of discrete function spaces which, in most applications, are ordered by set inclusion:

$$\mathcal{V}^0 \subset \mathcal{V}^1 \subset \cdots \subset \mathcal{V}^l \subset \mathcal{V}^{l+1} \subset \cdots \subset \mathcal{V}^{l_{\max}}.$$

Since we consider only tensor product elements, we assume a corresponding tensor product structure for the function spaces as well. That is, if $U^l \in \mathcal{V}^l$ and $E^l \in \mathcal{M}^l$ with $E^l = \bigotimes_{i=1}^{m} \sigma_i$, then $U^l\big|_{E^l}$ is in the span of functions $\prod_{i=1}^{m} U_i$ with functions U_i defined on σ_i, e.g., in the case of a rectangle $[x_0, x_1] \times [y_0, y_1]$, the bilinear function

$$U(x,y) = U_{00}\frac{x-x_1}{x_0-x_1}\frac{y-y_1}{y_0-y_1} + U_{01}\frac{x-x_1}{x_0-x_1}\frac{y-y_0}{y_1-y_0} + U_{10}\frac{x-x_0}{x_1-x_0}\frac{y-y_1}{y_0-y_1} + U_{11}\frac{x-x_0}{x_1-x_0}\frac{y-y_0}{y_1-y_0}$$

is the prototype of a tensor product function. The coeffients coincide with the values at the nodes of the rectangle.

Moreover, for the time being, we restrict ourselves to conforming Lagrangian finite element function spaces generated by tensor products of linear functions U_i on simplices σ_i. This especially includes the elements sketched in Fig. 1 with, for instance, linear, bilinear, or trilinear functions defined on them, i.e., data is prescribed on the nodes of the elements and there are no further degrees of freedom. In case of higher-order function spaces, the method presented here works on the embedded tensor product subspaces. Nevertheless a generalization seems possible, cf. Section 7. In multigrid applications, function values on vertices may vary from one grid level to the other because of the so-called coarse grid correction [27]. For visualization purposes, we suppose that the unique finest level value is always given on each vertex.

Let us finally introduce some further useful notation. We define $U^l := P_{\mathcal{V}^l} U$, where $P_{\mathcal{V}^l}$ denotes the nodal interpolation operator on the grid $\mathcal{M}^l$ and U is a given, mostly discrete, continuous function. For $E \in \mathcal{M}^l$, the set of child elements in $\mathcal{M}^{l+1}$ is denoted by $C(E)$ (cf. Fig. 1). Furthermore, let us denote the set of nodes of E and $C(E)$ by $\mathcal{N}(E)$, and $\mathcal{N}(C(E))$, respectively, and define

$$\mathcal{N}_C^{l+1}(E) := \mathcal{N}\big(C(E)\big) \setminus \mathcal{N}(E),$$

$$\mathcal{N}_C(E) := \bigcup_{\tilde{E} \subset E} \mathcal{N}\big(\tilde{E}\big) \setminus \mathcal{N}(E),\ \mathcal{N}\big(\mathcal{M}^l\big) := \bigcup_{E \in \mathcal{M}^l} \mathcal{N}(E).$$

Thus, $\mathcal{N}_C^{l+1}(E)$ denotes the set of new nodes which are created when refining an element E once, whereas $\mathcal{N}_C(E)$ collects all nodes of elements generated form E by the recursive refinement up to the finest level. Finally, $\mathcal{N}(\mathcal{M}^l)$ denotes the set of all nodes corresponding to elements of the refinement level l.

The set of (open) faces F of a specific element E is denoted by $\mathcal{F}(E)$. It is noteworthy that every vertex x^{l+1} in $\mathcal{N}_C^{l+1}(E)$ can be evaluated as a weighted sum over the coordinate vectors of its thereby defined parent vertices $x^l \in \mathcal{P}(x^{l+1}) \subset \mathcal{N}(E)$ with weights $\omega_{x^{l+1}}(x^l)$:

$$x^{l+1} = \sum_{x^l \in \mathcal{P}\left(x^{l+1}\right)} \omega_{x^{l+1}}\left(x^l\right) x^l . \qquad (1)$$

The weights are assumed to depend solely on the refinement rule and on a numbering of child and parent vertices. In general, the number of refinement rules is small such that element and vertex production rules, including the weights, can easily be stored in a lookup table. Different refinement rules such as the bisection strategy, the so-called red-green refinement or the refinement of prismatic grids are, for instance, discussed in [3], [50], [18], and [56], respectively.

2 Adaptive Data Projection

Before we develop a rigorous concept of adaptive projection methods in multiresolution visualization, let us introduce the basic idea with some simple considerations. First of all, let us stress that finding appropriate data projections is a key issue in this field. Typical visualization methods, such as the extraction of isosurfaces, the color shading on slices in 3D, or the drawing of height fields on 2D domains, successively visit cells and invoke local rendering operations. In the hierarchical context, we process all grid cells on the coarsest level and, depending on certain user defined criteria, recursively pass over to child cells or confine ourselves with stopping at the current cell and the corresponding data resolution. The criterion whether to stop or to proceed is mostly related to some error measurement, i.e., if the true data is already sufficiently approximated on a coarse cell, then we can skip the expensive search for detailed features to be visualized on the child level. A very first, preliminary version of such a recursive visualization algorithm *Inspect()* applied to any macro element is sketched in the following pseudocode

```
Inspect(E) {
    if C(E) ≠ ∅ ∧ ¬S(E)
        for all Ẽ ∈ C(E)
            Inspect(Ẽ);
    else Extract(E);
}
```

where $S(E)$ is the Boolean valued stopping criterion and the procedure *Extract()* finally performs the local rendering on the element E. If $\eta(E)$ is some error evaluation on E and ϵ is a user prescribed error tolerance, then one possible stopping criterion is $S(E) := (\eta(E) \le \epsilon)$. Let us remark that, in the above algorithm, we have also skipped local search restrictions, such as, in the case of isosurface extraction, the consideration of min / max bounds for some hierarchical guidance (cf. [59]).

It is obvious that this rudimentary strategy comes along with the drawback of cracks in isolines or jumps in the color intensity at edges in 2D or on slices in 2D, cracks in isosurfaces in 3D, respectively. They occur because of the nonconformity of the resulting triangulation (cf. Fig. 2). In explicit, at transition faces between leaf elements of the recursive traversal on different grid levels, different approximations of the true function U are taken into account. On the one element, additional finer-level nodal values have to be consid-

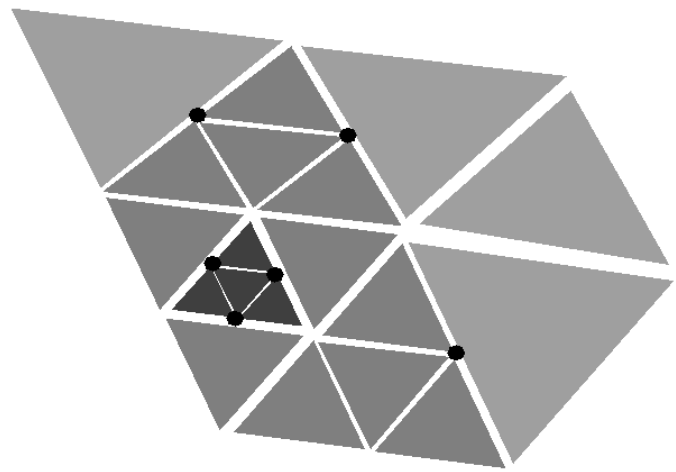

Fig. 2. An adaptive traversal of a 2D grid leads to nonconforming grids, respectively, hanging nodes.

ered, whereas, on the other, adjacent element, an interpolation of coarser nodal values defines the actually considered data approximation. In order to achieve an appropriate visual output from multiresolution visualization methods, we have to guarantee consistent data projections in case an adaptive stopping criterion is applied during the mesh traversal.

At first, one might ask for some adjustment procedure which explicitly refers to adjacency information among elements on the same, or on different grid levels (cf. [54]). But, in this case, the adaptive visualization method can no longer be coded as a strict, easy to implement, and fast depth-first hierarchical tree traversal. Even worse, in practical application, especially on economically stored unstructured grids [43], adjacency information is often not stored, but has to be retrieved from the grid hierarchy and the knowledge of the refinement process, i.e., depending on the refinement rule, we have to express adjacent elements of children as children of adjacent elements. If we only store neighborhood relations on the macro grid, this allows a recursive, but, in general, expensive, evaluation of adjacency [16], [43].

Therefore, we ask for a different approach which does not refer to adjacency information explicitly. It motivates the introduction of an appropriate adaptive data projection which turns out to be a mathematically rigorous and algorithmically flexible and efficient tool. We are thus able to formulate various multiresolution visualization operations. They can be applied to a large class of computational, hierarchical grids, consisting of elements which are tensor products of simplices with at least the corresponding tensor products of linear functions as the accompanying discrete function spaces. Furthermore, there are provisions for much more general discrete functions, such as those from general hp-Finite Element methods (cf. Section 7).

First of all, we replace the stopping criterion on elements by some *projection criterion* $S(x^l)$ for every vertex $x^l \in \mathcal{N}(\mathcal{M}^l)$ with $l \le l_{\max}$. For the time being, we assume S to attain values FALSE (0) and TRUE (1). If $\eta(x^l)$ is some error indicator on x^l and ϵ is a user prescribed threshold, then we define

$$S_\eta\left(x^l\right) := \left(\eta\left(x^l\right) \le \epsilon\right).$$

A variety of different projection criteria will be discussed in Sections 4 and 5. In Section 6, we will slightly generalize this to ensure a continuous level of detail in the animation of parameters such as the above threshold value. Now, we uniquely define the adaptive projection operator P_S corresponding to the above point-wise defined projection criterion S. It maps a discrete function $U \in \mathcal{V}^{l_{\max}}$ to a continuous, but now adaptive, function $P_S U$. Here, we take the tensor product structure of the local function space into account and obtain by (1) the following recursive formula for values of $P_S U$ on vertices $x^l \in \mathcal{N}(\mathcal{M}^{l_{\max}}) \setminus \mathcal{N}(\mathcal{M}^0)$:

(Projection Operator)

$$\left(P_S U\right)\left(x^l\right) := S\left(x^l\right) \sum_{x^{l-1} \in \mathcal{P}\left(x^l\right)} \omega_{x^l}\left(x^{l-1}\right)\left(P_S U\right)\left(x^{l-1}\right) + \left(1 - S\left(x^l\right)\right) U\left(x^l\right). \quad (2)$$

Furthermore, on the coarsest grid, $(P_S U)(x^0) = U(x^0)$ for $x^0 \in \mathcal{N}(\mathcal{M}^0)$. We choose the interpolated values if the projection criterion is fulfilled, else the true values are retrieved from the database. If $S(x)$ is true for all $x \in \mathcal{NC}(E)$, where $E \in \mathcal{M}^l$, then

$$P_S U\big|_E = P_S U^l\big|_E,$$

that is, the projection remains unchanged if we recursively process elements and vertices on finer grid levels. This implies a deduced natural *stopping criterion* on elements

$$S(E) := \bigwedge_{x \in \mathcal{N}_C(E)} S(x).$$

Although the adaptive projection is continuous by definition, in case of isosurfaces on specific grid types, we have to carefully handle the restriction of $P_S U$ at transition faces between different levels of resolution, on which bilinear discrete functions are involved. For a detailed discussion, we refer to [45]. Checking for the element stopping criterion, that is, testing the nodal projection criterion at all nodes $x \in \mathcal{N}_C(E)$ involves a look ahead onto all fine grid details on element E. But, this is computationally expensive and not very handsome. Therefore, we require a natural saturation condition for the projection criterion:

(Saturation Condition) *If the Projection criterion $S(x^l)$ is true for a node $x^l \in \mathcal{N}(E)$, then $S(x^{l+1})$ is true for all nodes $x^{l+1} \in \mathcal{N}_C^{l+1}(E)$.*

Based on this condition the stopping criterion simplifies to

$$S(E) := \bigwedge_{x \in \mathcal{N}_C^{l+1}(E)} S(x).$$

If the saturation condition is not fulfilled for a specific type of projection criterion, then we can adjust the criterion in a preprocessing step. In case of a typical error indicator, this generally turns out to be necessary only on coarse grid levels. On finer grid levels, on the other hand, we are already in a saturated state, except at singularities approximated in the data which are still not well resolved. For a detailed numerical background, we refer to [2], [18]. Such a saturation condition is very often implicitly assumed in multiresolution visualization. Here, we state it explicitly. This especially prevents us from overlooking details on fine grid levels. If a certain error indicator does not fulfill the above condition, a slight modification leads to a properly saturated indicator and an induced projection criterion, respectively. A simple update algorithm for an error indicator η and, thereby, the corresponding projection criterion S_η is the following level-wise traversal of the grid hierarchy, starting on the second-finest level and ending on the macro grid (cf. Fig. 2).

```
for (l = l_max − 1; l ≥ 0; l − −)
    for all E ∈ M^l {
        η* = max_{x^{l+1} ∈ N_C^{l+1}(E)} η(x^{l+1});
        for all x^l ∈ N(E)
            if (η(x^l) < η*)   η(x^l) = η*;
    }
```

Let us emphasize that a depth first traversal of the hierarchy in the adjustment procedure would not be sufficient.

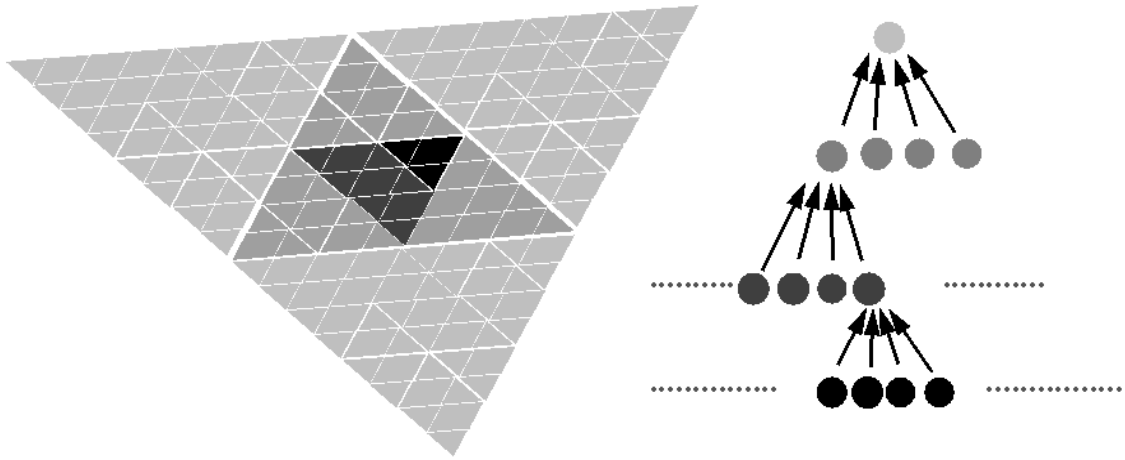

Fig. 3. A schematic sketch of the preroll to adjust indicator values and ensure the saturation condition. On the left, different grid levels of a triangular mesh are indicated by color. On the right, it is indicated that the sets of four child elements are taken into account to adjust the error indicator values on the parent elements.

3 A Multiresolution Algorithm

The general multiresolution algorithm is based on the application of the above introduced adaptive projection operator. It computes a continuous function on an adaptive grid performing a depth first traversal of the mesh hierarchy. This can be written in pseudocode

```
Inspect(E) {
    P_S U = AdaptiveProjection(U, E);
    if ElementOfInterest(P_S U, E) {
        if C(E) ≠ ∅ ∧ ¬S(E)
            for all Ẽ ∈ C(E)
                Inspect(Ẽ);
        else Extract(E);
    }
}
```

where *AdaptiveProjection()* is the above introduced nodal projection operation (2) and *ElementOfInterest()* checks whether features to be visualized are possibly inside the element or not, e.g., it is verified if the element is a candidate for the intersection with an isosurface or if there are critical points where to place some icons. For an implementation of such a routine along the guidelines of adaptive error measurement, see Section 4.4. Let us emphasize that the saturation condition is the key which prevents us from having to check complex adjacency information.

This saturation condition comes along with another desirable and straightforward consequence. Performing the adaptive visualization algorithm, we end up with at most one level transitions at faces of elements on which the local rendering takes place, i.e., on each such face, vertices of only two, not necessarily successive, levels will occur (cf. Fig. 2). We prove this by contradiction.

Let us suppose that two elements E, $\tilde{E}$ meet at a certain face F. On $\tilde{E}$, the above algorithm already stops, i.e., $S(\tilde{E})$ is true, whereas on E elements, $E^1 \subset E^2 \subset E$ of two different finer levels are traversed, i.e., $S(E)$, $S(E^2)$ are false (cf. Fig. 4). Furthermore, we assume that E^1, E^2 have faces F^1, respectively F^2, with $F^1 \subset F^2 \subset F$. By assumption, there exists a node $x^1 \in \mathcal{N}(C(E^2)) \setminus \mathcal{N}(E^2)$ for which the projection criterion $S(x^1)$ fails. We then know by means of the saturation condition that $S(x^2)$ also fails for all nodes $x^2 \in \mathcal{N}(E^2)$, especially for those on $F^2 \subset F$. Therefore, $S(x)$ fails at least on one node $x \in \mathcal{N}_C(\tilde{E})$. But, this contradicts our assumption that $S(\tilde{E})$, is true, once more because of the saturation condition.

If we run the adaptive algorithm, the full grid hierarchy is partially traversed. Obviously, we cannot do better, i.e., resolve the considered physical quantity finer than provided by the actual local depth of the hierarchical database. Let us suppose that the projection criterion is related to the same error estimator η that was originally used in the adaptive numerical algorithm to compute the data. Then, the adaptive projection in the postprocessing resamples the computational grid history for decreasing threshold value ϵ down to the threshold value ϵ^* at which the computation was finally stopped. Fig. 5 depicts this schematically and compares it with a simple cut-off at some level of the grid hierarchy. If we use a different projection criterion, the computational grid hierarchy and the portion of it traversed during the visualization algorithm will not match properly. In certain

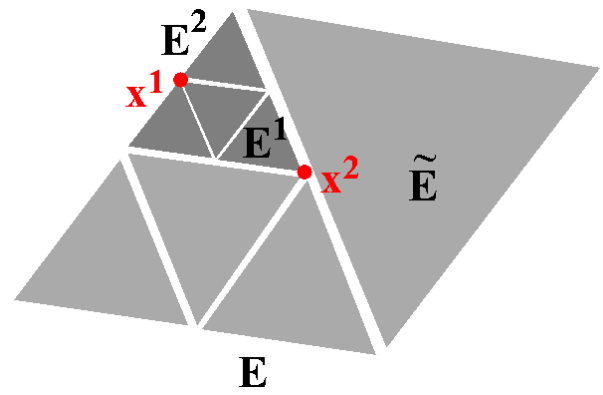

Fig. 4. In the adaptive traversal, at most one level transitions occur. Thus, child element of E^2 would not be visited, if the algorithm stops on $\tilde{E}$.

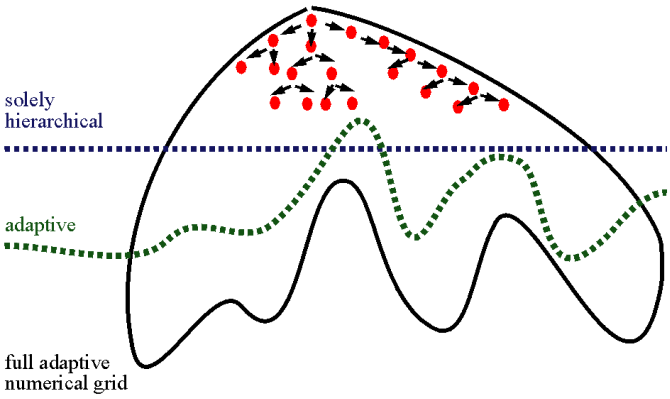

Fig. 5. A solely hierarchical traversal of the grid would stop at a certain level of the hierarchy, whereas an adaptive traversal allows a stopping criterion depending on the data.

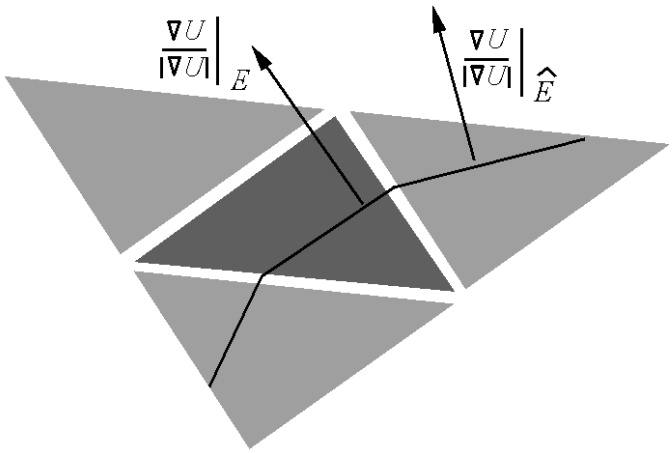

Fig. 6. A schematic sketch of the jump of the normalized gradients across an edge in 2D.

local areas, a recomputing would be necessary to overcome this shortcoming.

4 Primary Projection Criteria

Up to now, the projection criterion $S(x)$ on nodes x on different grid levels is still some abstract Boolean valued function which is admissible if the saturation condition is fulfilled. The almost trivial choice is the level wise post processing which is induced by the projection criterion $S(x) = (l > l^*)$ for $x \in \mathcal{N}(\mathcal{M}^l) \setminus \mathcal{N}(\mathcal{M}^{l-1})$, where l^* is the considered recursion depth. We will now discuss further suitable and more advanced projection criteria, corresponding to different aims of a multiresolution strategy.

4.1 Visual Error Indicator

The visual impression and a sufficient resolution of numerical data in the visualization process is closely related to curvature, for instance, curvature of isosurfaces or isolines on slices. Therefore, we ask for a discrete curvature quantity which locally measures the quality of the data approximation from the viewpoint of the visual appearance [45]. One thing we can easily recognize in images consisting of isosurfaces are folds at surface edges or, in case of isolines on slices, folds at polygon vertices. In each element, the data gradient ∇U^l is always perpendicular to an isosurface or an isoline on any chosen plane. Therefore, at any point x on an element face F, the normal component of the jump of the normalized gradient, denoted by $\left[\frac{\nabla U^l}{|\nabla U^l|}\right]_F$ locally measures the fold in the data function (cf. Fig. 6). Here, the jump operator applied to some function W is defined as

$$[W]_F := \lim_{i\to\infty} \left| W\left(x_i^+\right) - W\left(x_i^-\right) \right|$$

for sequences $\{x_i^-\}$ and $\{x_i^+\}$ converging to x from different sides of F. Let us remark that, for linear functions u on simplices, the gradients are constant on elements. This jump obviously serves as a well-founded graphical error criterion and motivates the following definition of an error indicator for a node $x \in \mathcal{N}(\mathcal{M}^l)$

$$\eta_V(x) := \max_{F\in\mathcal{N}(\mathcal{M}^l)\wedge x\in F} \left[\frac{\nabla U^l(x)}{|\nabla U^l(x)|}\right]_F$$

and the corresponding projection criterion $S_V(x) = (\eta_V(x) \le \epsilon)$ for a threshold value ϵ (cf. [45]). Fig. 7 shows isosurfaces for a test data set resulting from applications of the adaptive algorithm for different values of ϵ and, in Fig. 8, we sketch the statistical behavior in a diagram. Fig. 9 demonstrates the applicability of the method for simulation data.

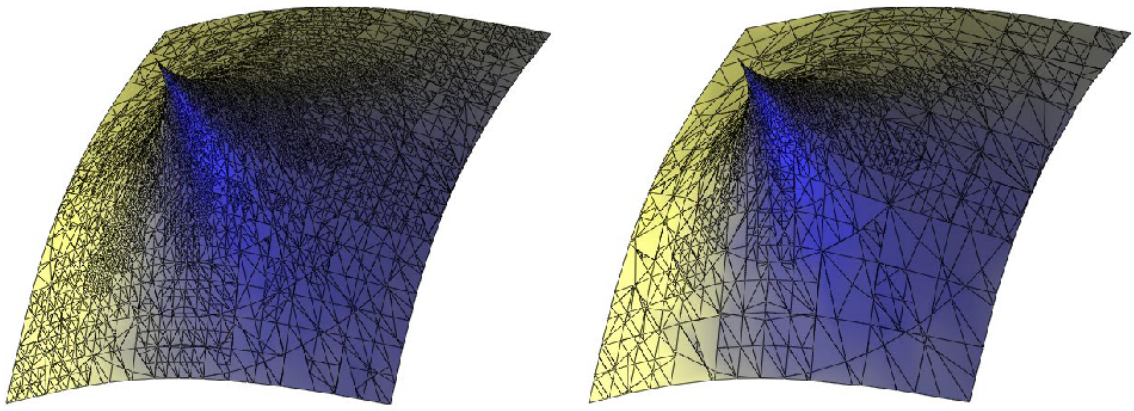

Fig. 7. Adaptive isosurface extraction on a test data set, the grid of which consists of 12 million tetrahedrons, for different threshold values.

4.2 Numerical Error Indicator and Wavelet Coefficients

Adaptive numerical methods have become popular, especially in the last decade, and proven to be efficient strategies to adequately resolve solution features in simulation computations. Features of significant interest are, for instance, general singularities, boundary layers, or vortices which cannot be sufficiently resolved by numerical methods on standard uniform grids. Many of these adaptive methods have in common that they successively cycle over the following three steps: Compute an approximate solution on the current grid, calculate local error estimator or indicator values $\eta_N(x)$ for grid nodes x, adapt the grid applying local refinement or for time-dependent problems also local coarsening. This cycle is stopped if a prescribed error tolerance is falling short. Depending on a given norm $\|\cdot\|$, for some partial differential equations true error estimators are known such that a reliable a posteriori error estimate

$$\|u - U\| \le C \left\| \left(\eta_N\left(x^l\right) \right)_{x^l \in \mathcal{N}\left(\mathcal{M}^l\right)} \right\|$$

holds [44], where u, U are the continuous, respectively numerical, solution on the grid $\mathcal{M}^l$ and $\|\!|\cdot|\!\|$ is an appropriate norm on the space $\mathbb{R}^{\#\mathcal{N}\left(\mathcal{M}^l\right)}$, e.g., for Poisson's problem we are led to

$$\eta_N(x) := h_F \left\| \left[\nabla U^l(x) n_F \right]_F \right\|_{L^\infty(F)}$$

as an error indicator on faces F, respectively on vertices x, which are going to be created on theses faces after some local refinement, where h_F is the diameter of face F.

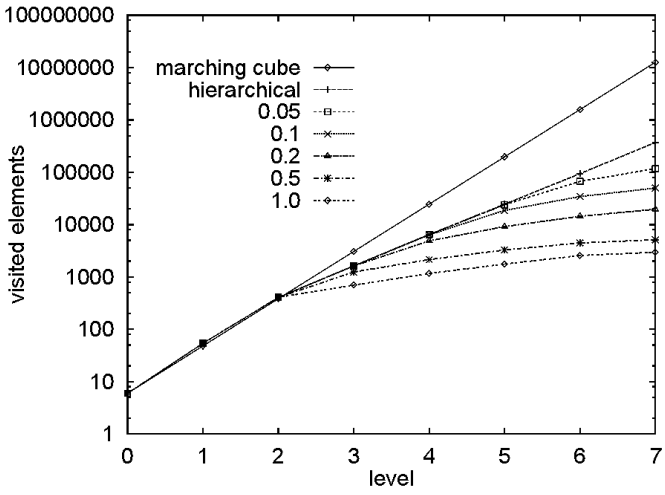

Fig. 8. On a logarithmic scale, we compare different visualization strategies concerning the overall number of visited cells for increasing grid level. Compare with Fig. 7 for the corresponding data set. We expect the purely hierarchical isosurface extraction to distinguish from the marching cube method in the slope of the curve by a factor of $\frac{2}{3}$. This is obviously reflected by the above diagram. Furthermore, for successively increased threshold value, the method reaches a saturated state successively earlier. The behavior at the singularity is not visible in the diagram.

Seen against this background, it appears convenient to use these estimator values also for the visual post processing and define the error projection criterion $S_N(x) = (\eta_N(x) \le \epsilon)$, where ϵ is again some user defined threshold.

Wavelet-based methods have also become very popular and effective instruments in numerical methods [34], data compression [62], and, especially, in multiresolution visualization [25], [42], [19]. They are especially powerful when considered on regular, structured grids. A key issue of wavelet type multiresolution visualization is the error measurement in terms of local frequencies, which is often a desirable feature, e.g., for geographical maps. At first, if data is not already given in wavelet space, it is analyzed and a hierarchy of wavelet coefficients is extracted from the input data set. These wavelet coefficients correspond to wavelets or prewavelets $\psi^l(\cdot)$ with $1 \le l \le l_{\max}$ evaluated at the nodes $x \in \mathcal{N}(\mathcal{M}^l)$ of a specific hierarchical depth. In terms of our approach, during a recursive wavelet synthesis which locally converts back to the standard function basis, the wavelet coefficients can serve as the appropriate error indicator $\eta_w(x)$ if we ensure saturation. Therefore, this important class of methods also fits into the presented frame. For a detailed discussion of this topic, we refer to the variety of efficient and specialized methods in the literature.

4.3 Magnifying Glasses

Another desirable feature of multiresolution data processing is the focus on a specific domain in image or object space. Inside some lens domain Ω, we thus expect at least a certain fineness $h_{\min}$ of the grid on which we extract and visualize information. Outside a significantly coarser mesh width, $h_{\max}$ is supposed to be sufficient. To focus on certain details using a lens has already been discussed by Bier et al. [4] and by Cignoni et al. [10]. Here, we embed such an approach in the concept of adaptive projection operators. In what follows, we will restrict ourselves to lens domains Ω in object space with Lipschitz continuous boundary. For domains $\tilde{\Omega}$ in image space, we consider the pull back $\Omega = M^{-1}(\tilde{\Omega})$, where M is the affine transformation from object space to image space. We ask for a projection criterion, which leads to the requested behavior of the visualization method. Furthermore, it has to be admissible, that is, to fulfill the saturation condition, to ensure continuity of the adaptive projection, and result in an appealing graphical output. Here, the saturation condition can be weakened: Projection criteria on parent nodes have to imply projection criteria on nodes. Let us define the lens projection criterion

$$S_L(x) := \left(h(x) \le \min\left\{ h_{\max}, C_{\mathcal{M}} \operatorname{dist}(x, \Omega) + h_{\min} \right\} \right),$$

where $h(x) = \operatorname{dist}(x, \mathcal{P}(x))$, $\mathcal{P}(x)$ is again the set of parent nodes of x, and $C_{\mathcal{M}}$ is some constant which solely depends on the type of refinement rules and will be fixed later. For a variety of domains Ω, this criterion is obviously easy to calculate and we are left to prove that the saturation condition holds.

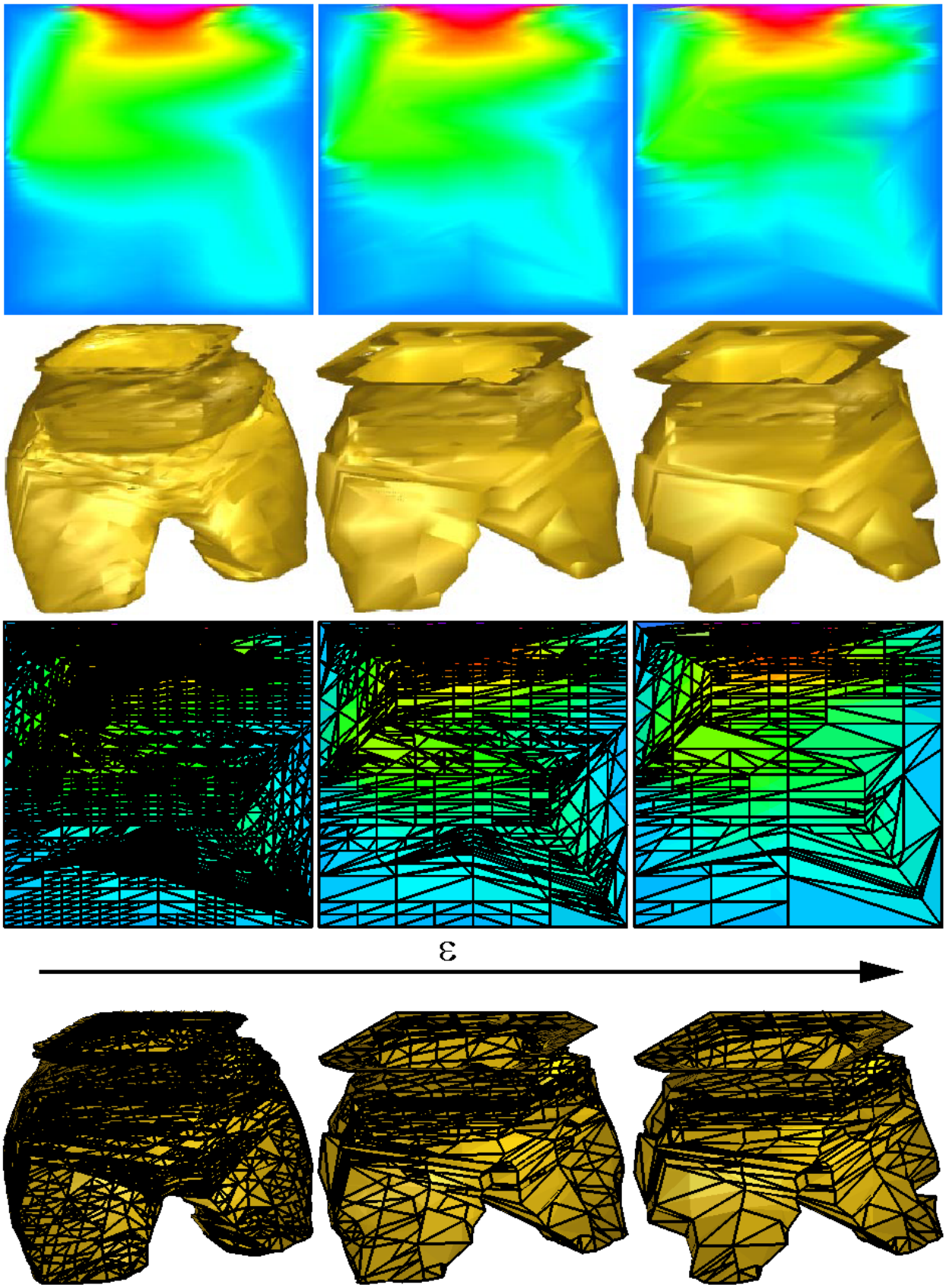

Fig. 9. On the top, color shading on slices and isosurfaces for increasing threshold values is applied to a porous media data set. The isosurface corresponds to the interface between fresh and salt water in this two-phase flow calculation. On the bottom, the intersections with element faces are outlined in black.

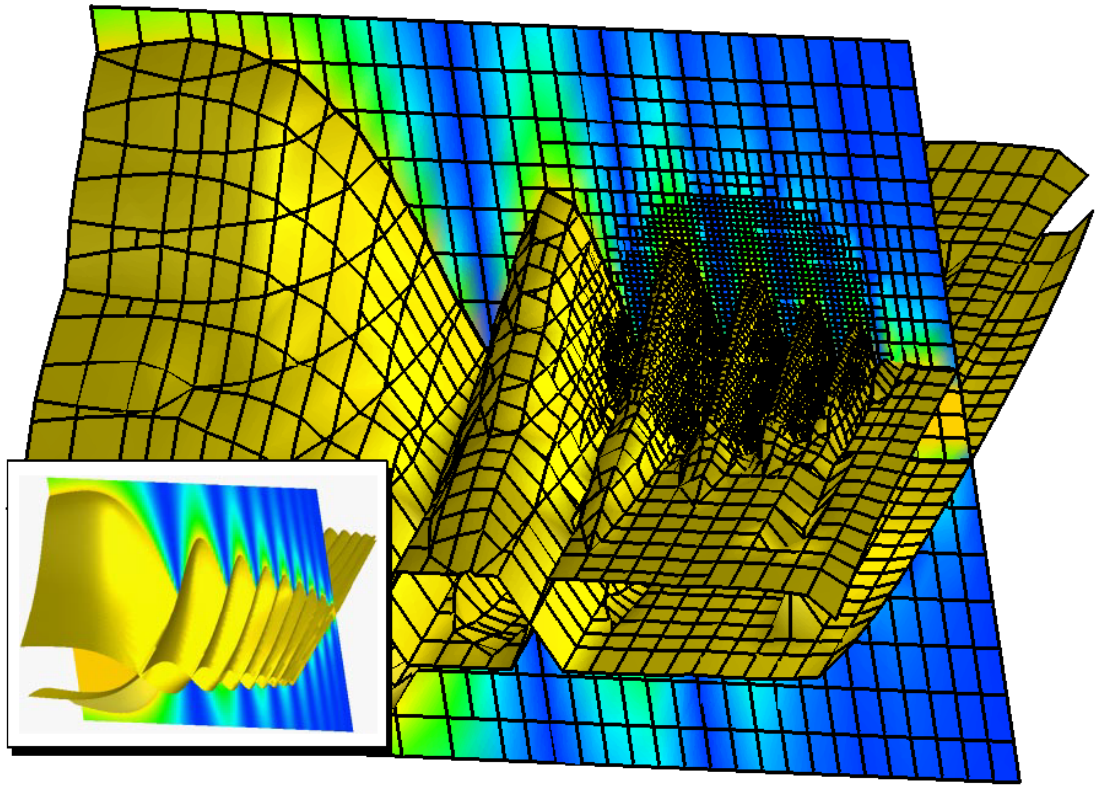

Fig. 10. Application of a magnifying lens to a 3D data set on a hexahedral grid. An adaptive isosurface and color shading on a slice are drawn for a ball-shaped magnifying lens. Outside the lens domain, isosurface and color shading are only resolved on coarse grid levels. In the lower-left corner, a full resolution image is added for comparison.

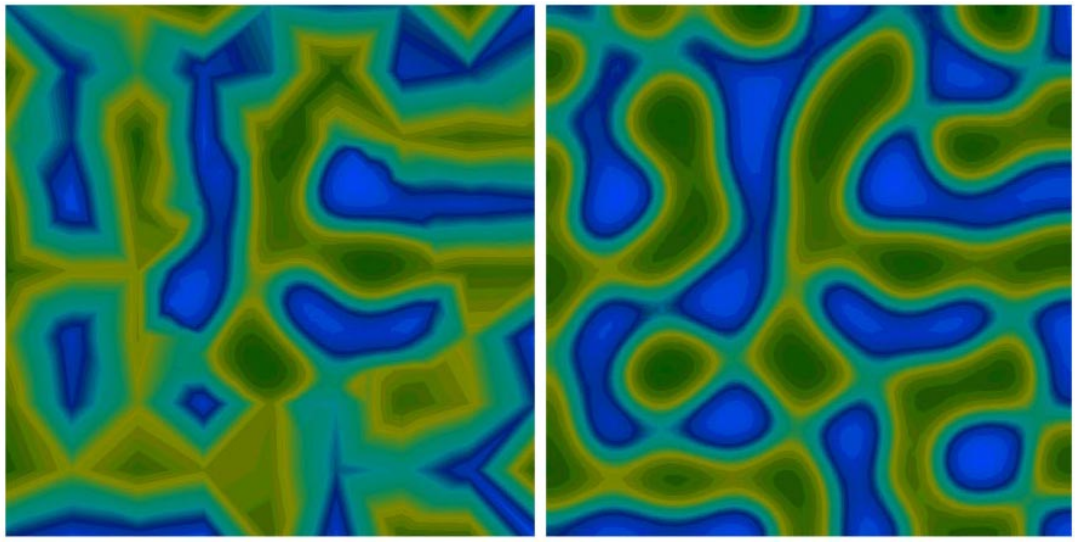

Fig. 11. Color shading of the density in a 2D phase transition simulation is shown on the left with respect to an applied circular lens and, on the right, equally fine on the whole domain. Data is thereby given on a uniform triangular grid.

Let us assume that $S_L(x^l)$ is true for one $x^l \in \mathcal{N}(E)$ on some element E. For any $x^{l+1} \in \mathcal{N}_C^{l+1}(E)$ with $x^l \in \mathcal{P}(x^{l+1})$

$$h\left(x^{l+1}\right) \le \left|x^{l+1} - x^l\right| \le \alpha\, h\left(x^l\right)$$

for a fixed constant $\alpha \in (0, 1)$ depending on the refinement rules used in the grid generation. Then, taking into account that dist(x, Ω) is Lipschitz continuous in x with Lipschitz constant 1, we obtain for $C_{\mathcal{M}} \le \frac{1-\alpha}{\alpha}$

$$\left|\operatorname{dist}\left(x^l, \Omega\right) - \operatorname{dist}\left(x^{l+1}, \Omega\right)\right| \le \left|x^{l+1} - x^l\right|$$

$$\le \alpha\, h\left(x^l\right) \le \frac{1-\alpha}{C_{\mathcal{M}}} h\left(x^l\right).$$

This immediately yields

$$C_{\mathcal{M}} \operatorname{dist}\left(x^l, \Omega\right) - (1-\alpha) h\left(x^l\right) \le C_{\mathcal{M}} \operatorname{dist}\left(x^{l+1}, \Omega\right).$$

Using the above estimates, we finally observe

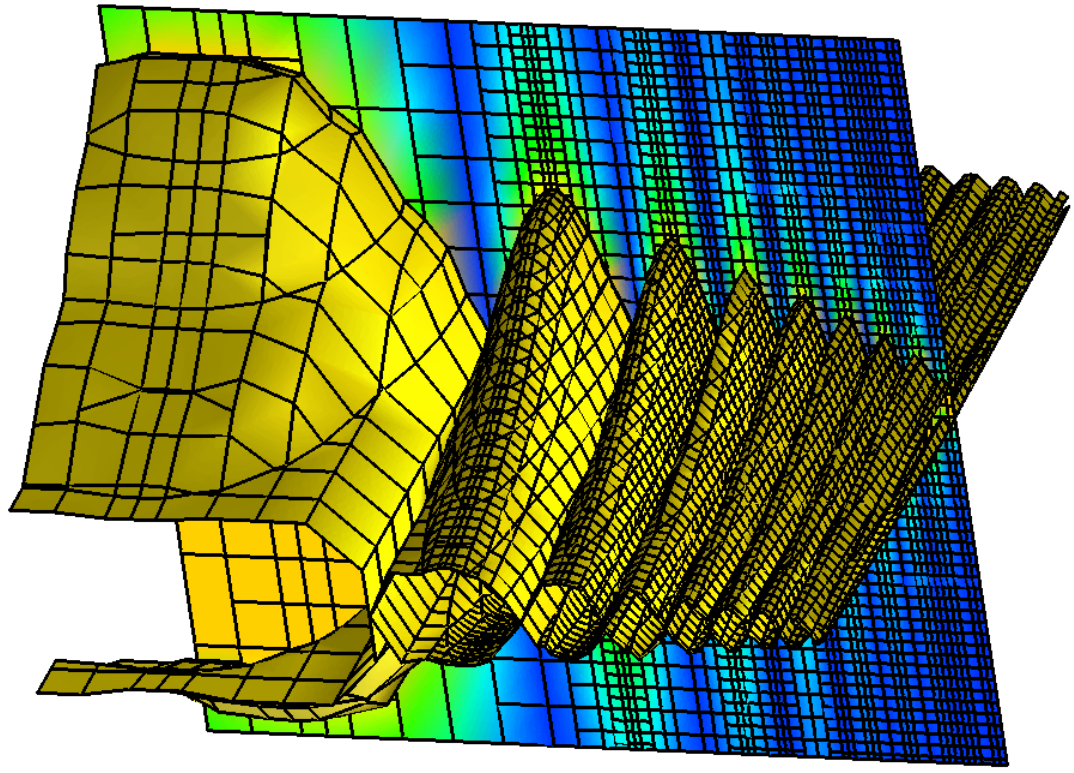

Fig. 12. Adaptive color shading and isosurface extraction based on the hierarchical error indicator are applied on a hexahedral grid.

$$
\begin{aligned}
& h\left(x^{l+1}\right) \le \alpha\, h\left(x^{l}\right) \\
& \le \min\left\{h_{\max}, C_{\mathcal{M}}\operatorname{dist}\left(x^{l}, \Omega\right) + h_{\min}\right\} - (1-\alpha)h\left(x^{l}\right) \\
& \le \min\left\{h_{\max}, C_{\mathcal{M}}\operatorname{dist}\left(x^{l}, \Omega\right) - (1-\alpha)h\left(x^{l}\right) + h_{\min}\right\} \\
& \le \min\left\{h_{\max}, C_{\mathcal{M}}\operatorname{dist}\left(x^{l+1}, \Omega\right) + h_{\min}\right\}.
\end{aligned}
$$

Therefore, $S_L(x^{l+1})$ holds, which had to be proven.

In case of a regular hexagonal octree $\alpha = \frac{1}{2}$ (cf. Fig. 1). Therefore, 1 is an admissible value for $C_{\mathcal{M}}$. If we apply triangular or tetrahedral bisection [50], [41] or the so-called red refinement of simplices [18], where triangles and tetrahedrons are divided into four, respectively eight, child elements, α depends on the regularity of the initial mesh. Let us remark that, for decreasing values of the constant $C_{\mathcal{M}}$, we obtain an increasing thickness of the transition zone between fine and coarse grid granularity. Figs. 10 and 11 depict examples for 2D and 3D meshes, where we have chosen a ball-shaped lens domain in object space, whereas Fig. 13 points out that also nonstandard domains can be handled as lens domains. In the application, the lens domain will be parameterized to ensure an effective and interactive exploration of the database. To underline the close relation to the projection criterion derived from error indicators and a corresponding user defined threshold, here we define $\eta_L(x) := h(x)$ as indicator and $\epsilon(x) := \min\{h_{\max}, C_{\mathcal{M}}\operatorname{dist}(x, \Omega) + h_{\min}\}$ as threshold. In what follows, we refer to this analogy.

4.4 Hierarchical Values

Frequently, the data values stored on higher-order grid nodes x^{l+1} are not the original function values $U(x^{l+1})$, but the offset $U_\delta(x^{l+1})$ at the nodes corresponding to the canonical nodal projection operator onto $\mathcal{V}^l$ applied to U. They are related to the U-values by the following recursive formula

$$
U\left(x^{l+1}\right) = \sum_{x^l \in \mathcal{P}\left(x^{l+1}\right)} \omega_{x^{l+1}}\left(x^{l}\right) U\left(x^{l}\right) + U_\delta\left(x^{l+1}\right).
$$

The U_δ-values allow an economical δ-compression of the data and the original values can easily be retrieved, if the above recursion is applied during the mesh traversal in a visualization method. Furthermore, we can choose

$$
S_H(x) := \left(\left|U_\delta(x)\right| \le \epsilon\right)
$$

as a projection criterion. As before, it is admissible if the saturation condition is fulfilled. For smooth data, e.g., $U(x) = u(x)$ for all $x \in \mathcal{N}(\mathcal{M}^{l_{\max}})$ with $u \in C^2$, $|U_\delta(x^{l+1})| = O(\operatorname{diam}(E)^2)$ for $x^{l+1} \in \mathcal{N}_C^{l+1}(E)$, which implies the saturation condition asymptotically on grids $\mathcal{M}^l$ for l sufficiently large (cf. Fig. 12). We can apply the adjustment algorithm from Section 2 to precompute an admissible hierarchical L^∞-error indicator. Alternatively, we can compute a robust upper bound for the offset values on elements by the recursive formula

$$
\eta_H(x) := \left|U_\delta(x)\right| + \max_{\{E \mid x \in \mathcal{N}(E)\}} \max_{x^{l+1} \in \mathcal{N}_C^{l+1}(E)} \eta_H\left(x^{l+1}\right),
$$

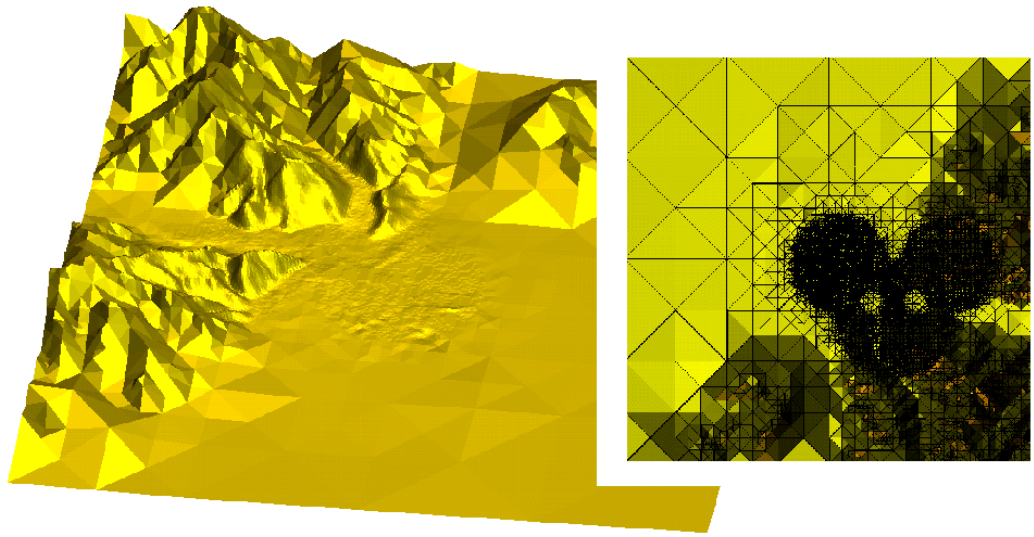

Fig. 13. The application of a magnifying glass for a nonstandard lens domain combined with a geometry error control.

where, on the second-finest grid level, $\eta_H(x) := |U_\delta(x)|$ for all $x \in \mathcal{N}(\mathcal{M}^{l_{\max}})$. These values can also be used to perform the necessary intersection test during the hierarchical extraction of an isosurface. We thereby avoid the expensive storing of min / max values as discussed in [59] (cf. also Section 5.3).

For the sake of completeness, let us sketch the corresponding *ElementOfInterest()* routine (cf. Section 3):

```
ElementOfInterest(P_S U, E) {
    if(min_E P_S U − η_H(E) ≤ C ≤ max_E P_S U + η_H(E))
        return true;
    else
        return false;
}
```

whereby $\eta_H(E) := \max_{x \in \mathcal{N}(C(E)) \setminus \mathcal{N}(E)} \eta_H(x)$ and C denotes the isovalue currently of interest.

5 Derived Projection Criteria

Up to now, different projection criteria have been discussed mainly for stationary and scalar discrete functions on 2D and 3D hierarchical grids. How to derive appropriate criteria from them, especially for vector-valued, time-dependent functions or for geometric shapes will be discussed in what follows.

5.1 Combining Different Criteria

In the previous paragraphs, we have discussed several types of projection criteria which we apply to define adaptive data projections in multiresolution visualization. They all ensure a sufficient resolution of the visual image with respect to some quality criterion, e.g., an acceptable error for the considered physical quantity, a suitable resolution of the geometric shape, or a detailed data enhancement in a user defined focus. It is frequently required to fulfill several criteria at the same time. Therefore, we combine the set of corresponding projection criteria $\{S_i\}_{1 \le i \le m}$ to one criterion

$$S := S_1 \wedge S_2 \wedge \cdots \wedge S_m. \tag{3}$$

Again, the saturation property for S is inherited from those for the different S_i. Fig. 13 presents a combination of magnifying lens and geometric error indicator for a geographical map.

A different combination of error indicators is needed in case of vector valued function data, where we already have error indicator values at hand for the components of the data function. Therefore, we can collapse a vector of corresponding error indicators η_i to one error indicator η for each node defining

$$\eta(x) := \gamma\big(\eta_1(x), \ldots, \eta_d(x)\big)$$

for a function $\gamma: \mathbb{R}^d \to \mathbb{R}$ which is increasing in all its components. It can easily be proved that this induces the admissibility of the adjoint projection criterion $S(x) = (\eta(x) \le \epsilon)$. Especially, every norm on the space $\mathbb{R}^d$, such as the maximum norm, is well-suited.

5.2 Geometry Error Indicator

Up to now, we have considered discrete functions on domains in two and three dimensions. But, instead of planar domains in $\mathbb{R}^2$, we can similarly deal with surfaces $\mathcal{G}$ in $\mathbb{R}^3$ which are approximated by polygonal grids $\mathcal{M}^l$ for $l \le l_{\max}$ starting with a coarse initial approximation $\mathcal{M}^0$. Let us emphasize that $\mathcal{G}$ does not have to be a parameterized surface (cf. Fig. 18, which displays the deformation of an elastic cylindrical shell). Nevertheless, we can parameterize $\mathcal{M}^{l_{\max}}$ over $\mathcal{M}^0$ by some function G which is supposed to be closed to a parameterization g of $\mathcal{G}$ over $\mathcal{M}^0$ (cf. [17]). Following the guidelines for the adaptive projection of discrete functions, we can analogously define adaptive geometry projections

$$P_S\mathcal{G} := \big(P_S G\big)\big(\mathcal{M}^0\big),$$

applying the above results to the in general vector valued parameterization G of $\mathcal{G}$ (cf. Section 5.1). Fig. 14 shows results for some geographical map.

5.3 Time-Dependent Data

In most physical simulations and for many applications in geometric modeling, the discrete function U or the geometry

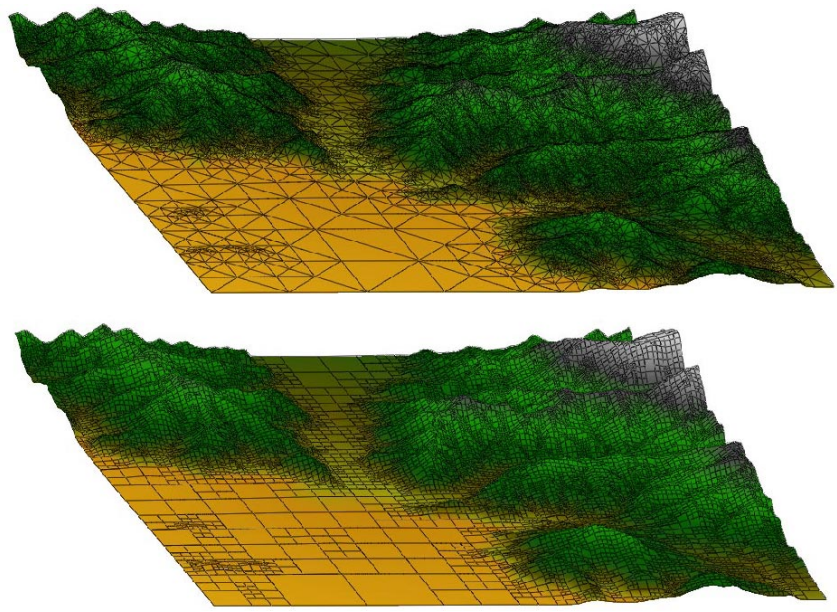

Fig. 14. Geometry error control for geographical maps. We compare data representation on a regular and on a triangular mesh. The elements on which the stopping criterion is fulfilled are outlined for the two grid types. The same type of error indicator and the same threshold value lead to slightly different results, as is visible, especially in the lower-left corner.

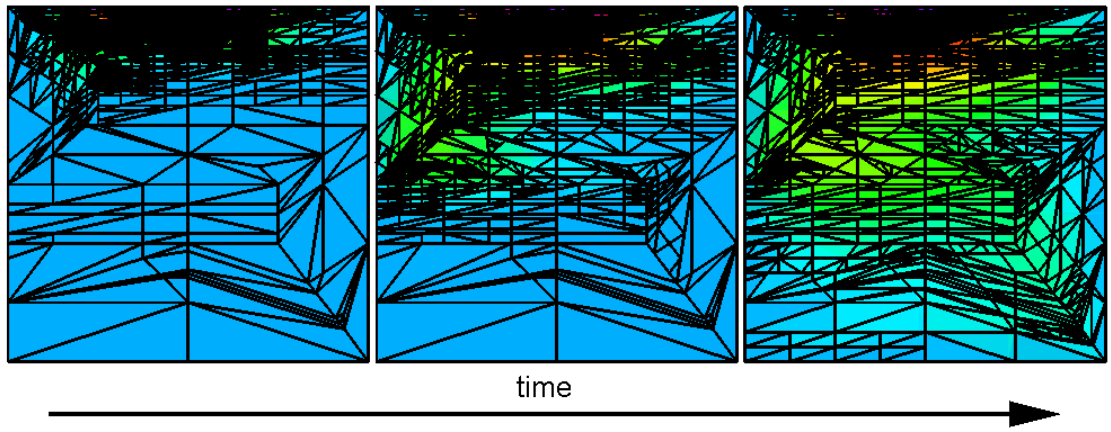

Fig. 15. Interpolation in time for adaptive color shading on slices: The pictures show the distribution of the concentration in a two phase flow calculation at different time steps, which do not coincide with timesteps from the computation.

$\mathcal{G}$ are time-dependent. Typically, a sequence of time steps, also called key frames, is given and an appropriate interpolation is used in between. We here restrict ourselves to the case of multiscalar functions. As already mentioned in Section 5.2, a geometric multiscale analysis works analogously. Let us denote by $\{U_{t_i}\}_{1\le i\le m}$ the sequence of time steps. An interpolation $U(t, x)$ is uniquely defined by a corresponding interpolation

$$U(t,x) := I\Big(t, U_{t_1}(x), \ldots, U_{t_m}(x)\Big)$$

on the nodes $x \in \mathcal{N}(\mathcal{M}^{l_{\max}})$. The concrete interpolation scheme, however, depends on the application. Here, we implicitly assume a uniform mesh $\mathcal{M}^{l_{\max}}$ and rule out adaptivity of the considered numerical grids in time. For a concept to handle adaptive data in time and space, we refer to [48]. On the set of time steps $\{U_{t_i}\}_{1\le i\le m}$, a corresponding set of admissible indicators $\{\eta_i\}_{1\le i\le m}$ is assumed to be given. We ask for an appropriate indicator $\eta(t, x)$ on every node $x \in \mathcal{N}(\mathcal{M}^{l_{\max}})$, which again should be admissible. As we already know from Section 5.1, $\gamma_t(\eta_1(x), \cdots, \eta_m(x))$ is admissible for any family of component-wise increasing functions γ_t with $t \in [t_1, t_m]$. If we suppose the interpolation to be defined as a weighted sum

$$I\Big(t, U_{t_1}(x), \ldots, U_{t_m}(x)\Big) := \sum_{i=1}^{m} \mu_{t_i}(t) U_{t_i}(x)$$

with continuous nonnegative weights $\mu_{t_i}(t)$, we gain an appropriate induced and admissible indicator

$$\eta(t,x) := \gamma\Big(\big(\mu_{t_i}(t)\eta_{t_i}(x)\big)_{1\le i\le m}\Big),$$

where γ is a standard norm in $\mathbb{R}^m$, fixed in time (cf. Fig. 15). For the linear interpolation in time, the weights coincide with the simple hat functions

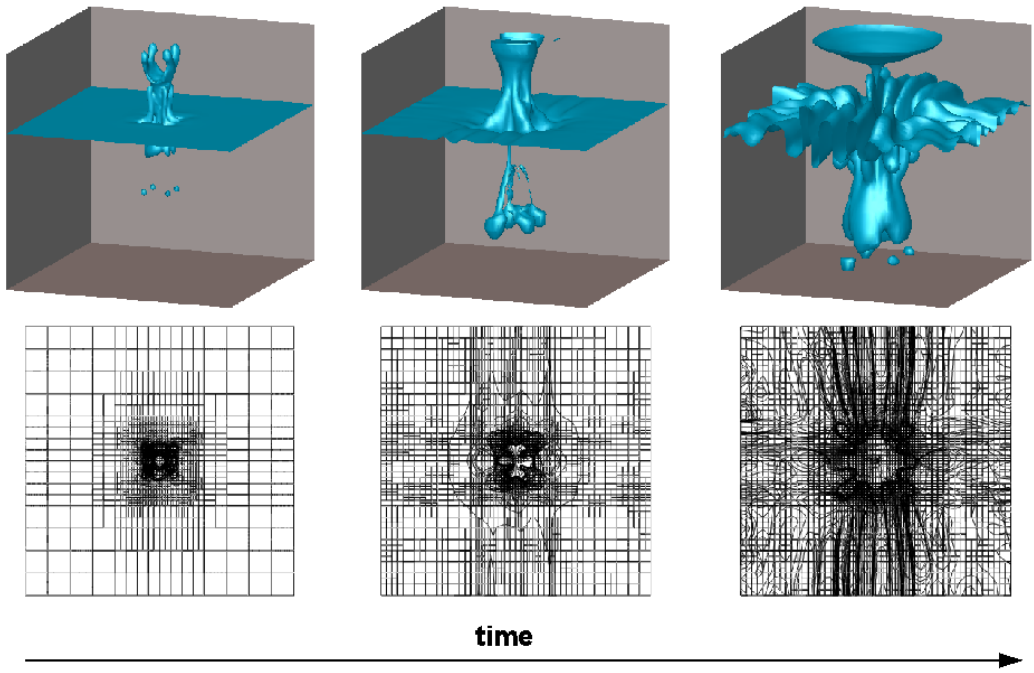

Fig. 16. Interpolation in time for adaptive isosurfaces. Above, at different times, an isosurface of a porous media density is extracted on a hexahedral grid. Below, in a projective view from the top, the edges of the cells on which the local isosurfaces are extracted show the adaptive approach.

$$\mu_{t_i}(t) := \max\left\{0, \min\left\{\frac{t-t_{i-1}}{t_i - t_{i-1}}, \frac{t-t_{i+1}}{t_i - t_{i+1}}\right\}\right\}.$$

Compare Fig. 15 for an adaptive color shading on slices on a tetrahedral mesh and Fig. 16 for adaptive isosurfaces on hexahedral grids, both extracted from interpolated data. Let us remark that if we take the indicator $\eta_H(x)$ as defined in Section 4.4 into account, the induced indicator

$$\eta_H(t,x) := \sum_{i=1}^{m} \mu_{t_i}(t)\eta_{t_i}(x)$$

can be used to calculate reliable data bounds for the interpolated function. To check this, we straightforwardly estimate the difference $U(t, x) - U^l(t, x)$, where $U^l(t, x)$ is the local restriction to a certain grid level l

$$\begin{aligned}
&\left|U(t,x) - U^l(t,x)\right| \\
&= \left|I\left(t, U_{t_1}, \ldots, U_{t_m}\right) - I\left(t, U^l_{t_1}, \ldots, U^l_{t_m}\right)\right| \\
&\le \sum_{i=1}^{m} \mu_{t_i}(t)\left|U_{t_i}(x) - U^l_{t_i}(x)\right| \le \sum_{i=1}^{m} \mu_{t_i}(t)\eta_{t_i}(x).
\end{aligned}$$

From our point of view, this important property points out a significant advantage of the hierarchical intersection test compared to other acceleration algorithms for isosurface extraction, including the span-space methods [55], the *k*-tree method [39], or the extremal graph approach [32]. Without any sophisticated adjustment, the expensive preparatory step which comes along with these algorithms has to be invoked on every new interpolation in time. This turns out to be a major drawback compared to almost no extra cost for the hierarchical strategy, provided time-dependent data is considered.

6 Continuous Level of Detail

For fixed projection criterion, we have so far obtained continuous, adaptive projections in space and time and, thereby, an appropriate visual output for a variety of visualization methods. If we apply the adaptive projection method corresponding to some nodal indicator η for varying threshold ϵ, and parameters of $\epsilon(x)$, respectively (cf. Section 4.3), continuity is no longer ensured. Indeed, every time the projection criterion switches on a node x, an immediate local transition between original data and projection will cause local jumps in the visual appearance of the animation. To overcome this shortcoming, we slightly generalize the projection criterion. Instead of the Boolean-valued projection criterion $S(x) = (\eta(x) \le \epsilon(x))$ for some indicator η and a threshold $\epsilon(x)$, we introduce a real-valued function

$$\tilde{S}(x) := \chi\left(\eta(x), \epsilon(x)\right),$$

where $\chi : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to [0, 1]$ is supposed to be a Lipschitz continuous function, monotone decreasing in η, with $\chi(\eta, \epsilon) = 1$ for $\eta \le \epsilon$. Now, we replace S by $\tilde{S}$ in the definition of the adaptive projection and obtain

$$\begin{aligned}
\left(P_{\tilde{S}}U\right)\left(x^{l+1}\right) := \tilde{S}\left(x^{l+1}\right) &\sum_{x^l \in P\left(x^{l+1}\right)} \omega_{x_i^{l+1}}\left(x^l\right)\left(P_{\tilde{S}}U\right)\left(x_i^l\right) \\
&+ \left(1 - \tilde{S}\left(x^{l+1}\right)\right)U\left(x^{l+1}\right).
\end{aligned}$$

By construction $P_{\tilde{S}}U$ continuously depends on η, ϵ. The corresponding stopping criterion $S(E)$ can be adapted in a straightforward manner

$$\tilde{S}(E) := \max_{x^{l+1} \in \mathcal{N}_C^{l+1}(E)} \tilde{S}\left(x^{l+1}\right),$$

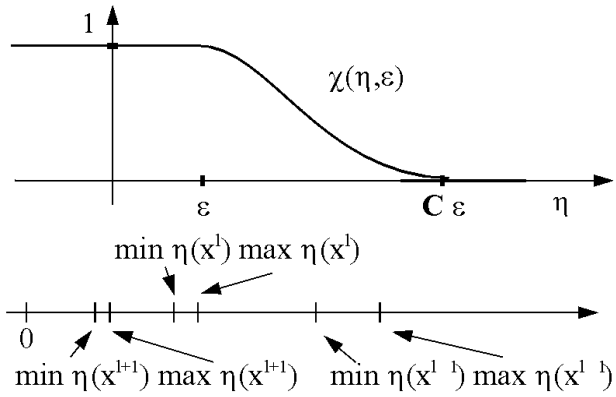

Fig. 17. On the top, a typical function χ is sketched and, on the bottom, the scale of indicator values on different grid levels is drawn schematically.

i.e., we test for $\tilde{S}(E)$. Moreover, in order to achieve an appropriate blending result for varying values of ϵ, we assume $\chi(\eta, \epsilon) = 0$ for $\eta \geq C\epsilon$ (cf. Fig. 17) and $C > 1$. Here, appropriate means that $P_{\tilde{S}}(U)$ on a finally extracted element E locally only depends on original U-values on nodes of at most two grid levels. This is a desirable property which is already satisfied for the original projection $P_S(U)$ due to the saturation condition (cf. Section 2), i.e., if

$$C \leq C^* := \frac{\min\limits_{x^l \in \mathcal{N}(E)} \eta\left(x^l\right)}{\max\limits_{x^{l+1} \in \mathcal{N}_C^{l+1}(E)} \eta\left(x^{l+1}\right)}$$

for all elements E up to the second-finest grid level this property also holds for the modified projection $P_{\tilde{S}}(U)$. We obtain smooth transitions between different levels of detail without any additional interaction on the hierarchy compared to the case of the nonmodified adaptive projection. As is typical for smooth data on sufficiently fine grid levels, C^* is strictly larger than 1 (cf. Section 4.4). But, in general, especially on coarse grid levels, we only have $C^* \geq 1$ because of the saturation condition. To overcome this drawback, we can introduce a modification of the suggested adjustment algorithm for indicators and replace the conditional blowup of the indicator (cf. Section 2) by the following pseudocode statement

if $\eta(x^l) < C\eta^*$ $\eta(x^l) = C\eta^*$;

where C is the constant later on used for the definition of χ. Otherwise, we have to accept additional interaction effects between different grid levels. In the case of the magnifying lens, the situation is simpler. Here, we have to decrease the constant $C_{\mathcal{M}}$.

Finally, the combined projection criterion (cf. (3)) for a set of different real valued projection criteria $\{\tilde{S}_i\}_{1 \leq i \leq m}$ can be redefined by

$$\tilde{S} := \frac{1}{m} \sum_{i=1}^{m} \tilde{S}_i.$$

To summarize, we have so far obtained a data projection continuously depending on a threshold value ϵ. Nevertheless, the visual appearance of the graphical results will only reflect this continuity if we guarantee that the parameters of the finally generated graphical primitives also depend continuously on $P_S(U)$ (cf., e.g., [20], [31]).

7 HP–Finite Element Data

Up to now, we have solely considered Finite Element data which is—for each element in the function space—spanned by tensor products of linear functions (cf. Section 1.2). A higher-order polynomial degree is often used in the numerical code to improve the approximation order whenever the approximate solution is smooth enough. Recently, adaptive methods, which adapt the grid size *and* the polynomial degree locally have become popular [6], [49]. In general, in areas where the solution points out higher-order

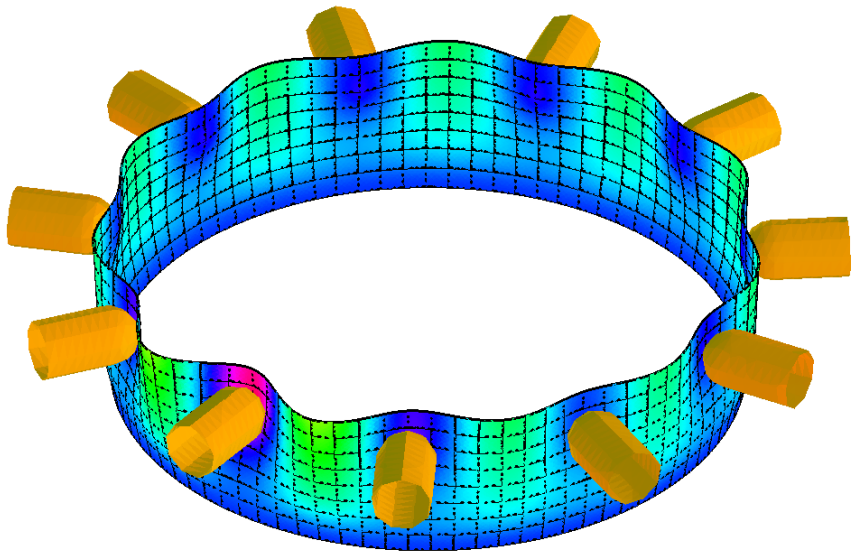

Fig. 18. Deformation of an elastic shell, which has been computed with a higher-order Finite Element method. The black lines indicate edges of the elements. Virtual refinement is used to obtain a much better approximation to the actual polynomial shape and the stress coded in color.

differentiability, the polynomial degree p is successively increased to obtain an exponential decay of the numerical error. In contrast, in areas where the solution properties indicate singularities, a refinement of the grid size h in general turns out to be the preferable strategy for error reduction. The combination of both is called the *hp*-Finite-Element approach. An efficient visualization of data from these effective discrete function spaces is a challenging task. Here, we discuss a generalization of the presented adaptive approach for meshes with successively refined grid size. Therefore, error information is measured on leaf elements of the grid hierarchy and, if necessary, additional "virtual" grid levels are introduced (cf. Fig. 18).

On the corresponding "virtual" elements E_{hp} we define standard adaptive projections $P_{S_{hp}}$ corresponding to some projection criterion S_{hp} on "virtual" nodes x_{hp}. The corresponding function spaces are again spanned by linear functions but, now, on the later-on refined grid. We can interpret this strategy as an *h-subsampling* of the actual polynomial data.

For the sake of simplicity, let us assume that the considered grid consists solely of one element type, which is triangular, or rectangular in 2D, respectively tetrahedral, prismatic, or hexahedral in 3D. For each of these element types, we consider a fixed refinement rule which decreases the element diameter by a factor of $\frac{1}{2}$, e.g., in the case of simplices, this is the red refinement rule [18], whereas, for right-angled cells the quad–, or octree construction is considered (cf. Fig. 1). These refinement rules come along with additionally created nodes $\mathcal{N}_{hp}(E)$ on element edges and faces and in the interior. Let us suppose some projection criterion $S_{hp}(x_{hp})$ to be defined on these nodes. On child elements E_{hp} of a "virtually" refined element, we introduce the stopping criterion $S(E_{hp})$ deduced from the projection criterion on the corresponding vertices (cf. Section 2). Starting on $E_{hp} = E$ as the initial "virtual" element with nodes x_{hp} and projection criterion $S_{hp}(x_{hp})$ inherited from the adaptive projection on the original grid the adaptive visualization algorithm can be defined analogously to the standard case. Finally, we ask for easy-to-compute projection criteria on nodes and for a replacement of the above saturation condition. It is much too expensive to store error indicators on all "virtual" nodes. They therefore have to be computed from the original data during runtime. To ensure efficiency of the final algorithm, only local information, which resides on the currently inspected element E_{hp}, should be taken into account for the definition of an error indicator. Furthermore, it is now ruled out to look ahead onto much finer grid levels in a preroll step in order to fulfill the saturation condition. To overcome this difficulty, we suppose $\eta_{hp}(x_{hp}^{l+1})$, respectively $S_{hp}(x_{hp}^{l+1})$, to be uniquely defined depending on $U(x_{hp}^{l})\forall x_{hp}^{l} \in \mathcal{P}(x_{hp}^{l+1})$ and $U(x_{hp}^{l+1})$. For instance, a suitable first choice for the error indicator would be the hierarchical offset value (cf. Section 4.4)

$$\tilde{\eta}_{hp}\left(x_{hp}^{l+1}\right) = \left| U\left(x_{hp}^{l+1}\right) - \sum_{x_{hp}^{l} \in \mathcal{P}\left(x^{l+1}\right)} \omega_{x_{hp}^{l+1}}\left(x_{hp}^{l}\right) U\left(x_{hp}^{l}\right) \right|.$$

With these restrictions, indicator values on edges, respectively faces, depend solely on data values on this edge or face and coincide with those evaluated on the adjacent element. Therefore, continuity of the induced adaptive projection is guaranteed if we do not apply the adaptive stopping criterion. In order to also allow an adaptive stopping, which is the actual aim of our considerations, a modification of the indicator values η_{hp} is necessary. We recursively define

$$\eta_{hp}\left(x_{hp}^{l+1}\right) := \min\left\{ \tilde{\eta}_{hp}\left(x_{hp}^{l+1}\right), \min_{x_{hp}^{l} \in \mathcal{P}\left(x^{l+1}\right)} \left\{ \eta_{hp}\left(x_{hp}^{l}\right) \right\} \right\}.$$

It is thereby especially ensured that "virtual" nodal values on element faces are always generated by interpolation whenever the algorithm stops at a coarser level on the corresponding adjacent element. This definition is necessary to overcome the saturation assumption. In a certain sense, we construct a somewhat saturated error indicator in a top-down manner instead of assuming the bottom up implication of the saturation condition. The construction also properly matches the original projection criterion S on true nodes x from the original grid hierarchy and the new projection criterion S_{hp}.

Therefore, we no longer need the stronger saturation condition. But, we can also no longer be certain that we do not overlook fine details in the data when stopping on insufficiently refined "virtual" elements. Furthermore, the useful property of finding at most one level transitions at faces no longer holds. Nevertheless, on higher-order polynomial data, the experimental results are satisfying, which seems to rely on the sufficient smoothness of the considered data function. If we apply the same strategy for general data on arbitrary grids, serious difficulties concerning image quality occur.

For a visualization method which draws isolines or displays some color shading on slices, a straightforward simplification is possible. Instead of subdividing the three dimensional elements which intersect the slice, we first compute the intersection polygons and, then, subdivide them into triangles (cf. Fig. 19). Finally, we apply the above algorithm on such triangles extracted on leaf elements of the original grid hierarchy. Fig. 20 demonstrates the significant improvement in data resolution obtained by the adaptive approach.

8 Some Algorithmic Aspects

We have implemented the concept of adaptive projections in 2D and 3D based on a general interface to data on arbitrary nested grids. This interface handles mesh elements procedurally. In detail, hierarchical access procedures address single elements and deliver information on the considered element in a prescribed data structure for a general element. A coarse grid, as well as a recursive depth-first traversal, is thereby supported by a set of specific interface routines. No conversion to a prescribed mesh format is nec-

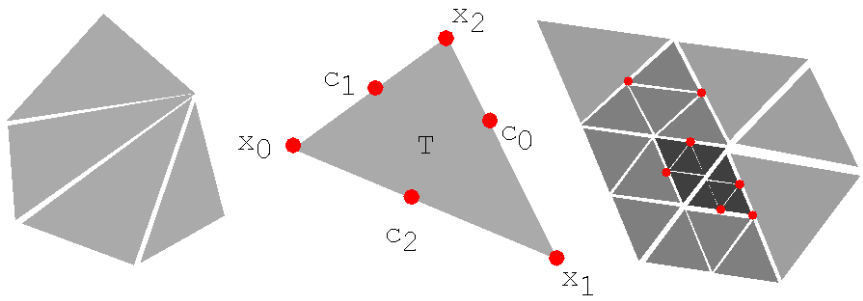

Fig. 19. On the left, the subdivision of a polygon into triangles is shown, whereas, in the middle picture, vertices and edge midpoints of a triangle are marked. The sketch on the right shows an extracted adaptive grid with more than *one* level transitions.

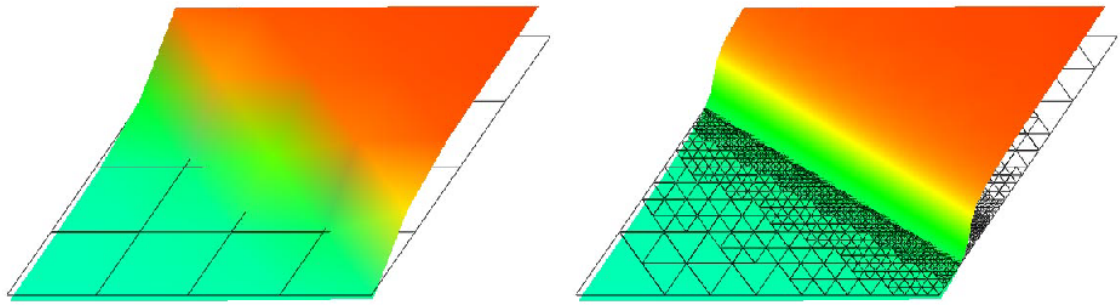

Fig. 20. On the left, the color shading on a slice applied to a 3D data set with an only *h*-refined grid and, on the right, the "virtual" *h*-subsampling on the same coarse grid.

essary in advance. The data mapping is performed only temporarily while running a visualization method. For details on the general element description, we refer to [51] and concerning the hierarchical access routines, especially, compare [43]. The images presented here are all generated applying this type of interface. Its major advantage is its generality. A large class of visualization methods, once implemented, and based on the procedural interface, immediately works on new nested-grid structures if an appropriate interface has been adapted to the specific user data structure. The visualization needs no significant extra memory. Very large hierarchical grids, especially, which are often stored economically, are thereby opened up for an effective post processing. Here, economical means that vertex and adjacency information is present only on the coarse grid elements. On the finer grid levels, we solely store references to refinement rules and references to new nodes. Complete data is then generated recursively during the procedural mesh traversal in the visualization (cf. [43]).

The presented multiresolution concept guarantees conformity of the extracted adaptive projection. If isosurfaces are considered, a local triangulation has to be generated on a leaf element in the adaptive mesh traversal algorithm and an overall smoothly shaded appearance is often required. We retrieve the local triangulation from a lookup table [40] which corresponds to the element type. For every element type, a lookup table is automatically generated whenever the algorithm picks up an element of this type for the first time.

As in the nonhierarchical case, if smooth shading is considered unique, surface normals have to be calculated at nodes. One approach is to interpret function gradients, which coincide with isosurface normals after normalization, as a vector valued discrete function. Then, we can apply the projection criterion already used for the original function and end up with continuous normals and smoothly shaded isosurfaces. An in advance calculation of interpolated gradients on all nodes, $x \in \mathcal{N}(\mathcal{M}^{l_{\max}})$ is often much too expensive concerning CPU time and storage requirement. We use hash tables to identify nodes on which a gradient has to be evaluated and which appear several times on different elements traversed in the isosurface methods [59]. The required hashing key depends on the coordinates of the nodes. On revisited nodes, we can then use the already calculated gradients. A presentation of the corresponding algorithmic details is beyond the scope of this paper and we refer to a forthcoming publication.

Let us finally comment on the use of color shading or texture mapping on discrete surfaces for data visualization when an adaptive projection has been applied. As long as the mapping from the function space into color or texture space is linear, the resulting appearance of color and texture is guaranteed to be continuous at the corresponding 2D element faces. If other mappings into color or texture space are considered, we have to perform the recursive adaptive projection not on the data function itself but on the resulting color and texture.

9 Remarks on Generality and Efficiency

The presented approach is restricted to nested grid hierarchies as they especially appear in numerical methods for partial differential equations describing physical phenomena in two or three dimensions. It is highly flexible in this mainly intended field of application, i.e., it is independent of the concrete element types, the refinement rules, and the possibly compressed user's data formats. Let us point out

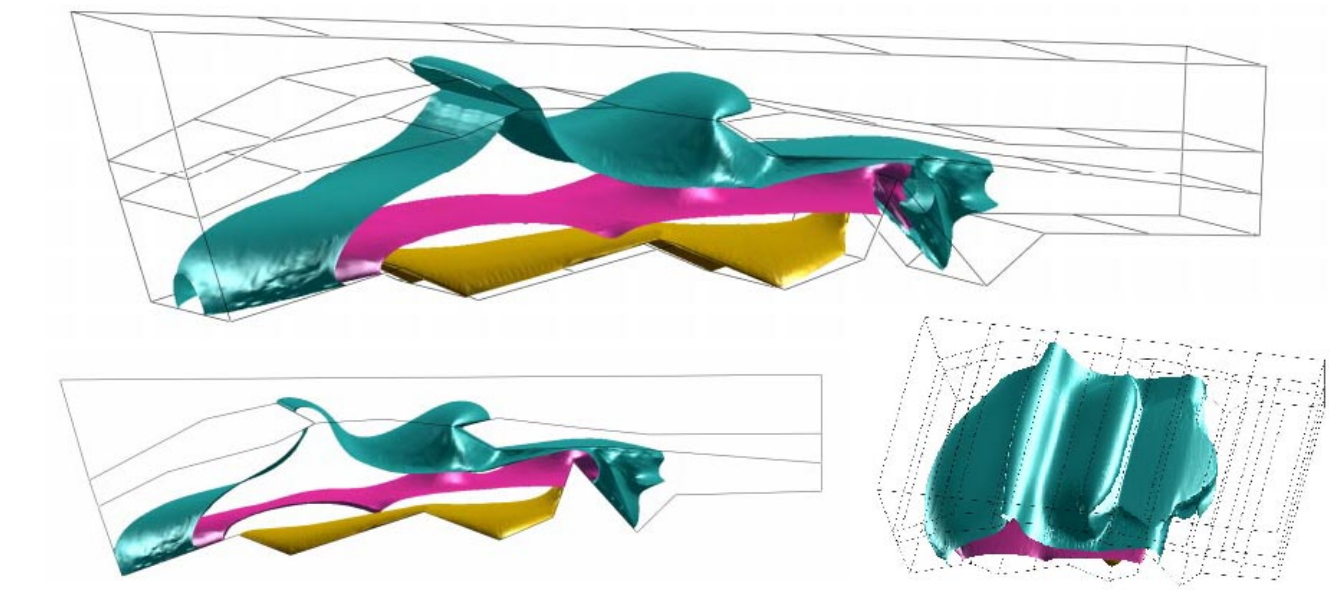

Fig. 21. Different isosurfaces of the salt concentration in a groundwater flow are extracted from a compressed 10 million element data set.

that there are other, more general, approaches especially for surfaces by De Floriani et al. [13] and Hamann and Chen [29] which also apply to nonnested grid hierarchies, but with a different focus concerning the field of applicability.

Concerning efficiency, we especially have to pay for the described flexibility in terms of CPU time if we settle the algorithms on the base of the described procedural data access (cf. Section 8). From our experience, there is a factor of about 2-3 compared to the same visualization method implemented and adapted on a specific data structure. Fig. 21 depicts a real-world problem from ground water flow on a hierarchical, unstructured mesh consisting of 10 million elements and 1.4 million elements (courtesy of K. Johannsen, ICA III, University of Stuttgart). The grid is adaptive and a set of different refinements rules is applied to generate it. Using the general procedural interface to address this data in highly compressed form, we still obtain about 93k triangles per second in the adaptive isosurface generation. Table 1 lists the number of triangles generated in the algorithm for an isosurface, the number of visited tetrahedra, respectively those on which we finally extract a local isosurface, and the resulting frame rate on an SGI Onyx2 with R10000 processor for different threshold values ϵ.

Here, we have taken the hierarchical error indicator η_H into account, including the adjustment procedure described in Section 4.4.

Finally, if optimal performance is required, the presented concept can easily be implemented on any optimized nested grid data structure which fits into our general frame. For instance, consider a regular hexahedral grid. If we apply the tetrahedral bisection strategy presented by Maubach [35] without storing tetrahedrons explicitly, but tracking the prescribed refinement rules in terms of quadtuples of index vectors for the vertices, we obtain a method similar to the one presented by Zhou et al. [63]. Fig. 22 shows results of the corresponding implementation of the isosurface algorithm for different threshold values. Again, we have applied the adjusted, hierarchical error indicator (cf. Section 4.4). The overall number of tetrahedrons is 12,582,912 and the grid consists of 2,146,689 nodes. Table 2 lists threshold values, triangle, respectively tetrahedra counts, and frame rates.

If we store min/max values on the tetrahedrons, the number of visited tetrahedron for $\epsilon = 0.0$ reduces to 1,216,638, which is a saving of only 3.4 percent , at the expense of an additional 3,145,728 floating-point values in storage (two for every tetrahedron up to second-finest level).

TABLE 1

ϵ	drawn triangles	visited tetrahedra	extracted tetrahedra	frames/sec
0.0	93k	204k	71k	1.0
0.01	34k	78k	26k	2.6
0.1	3k	10k	2.4k	21

10 Conclusions

A mathematically rigorous foundation of multiresolution data analysis is given here which applies to general hierarchical nested grids. The implementation of the corresponding visualization algorithms is confined to an appropriate depth-first traversal of the grid hierarchy, combined with the recursive calculation of continuous adaptive data projections. A corresponding stopping criterion, which indicates if the current data projection will locally remain unchanged on finer grid levels, allows a stopping on coarser grid levels and,

TABLE 2

ϵ	triangles drawn	visited tetrahedra	frames/sec
0.02	81,184	201,757	3.45
0.01	128,709	307,384	2.27
0.005	211,219	487,107	1.43
0.0025	315,440	727,419	0.98
0.00125	439,230	984,029	0.74
0.0	590,018	1,259,669	0.58

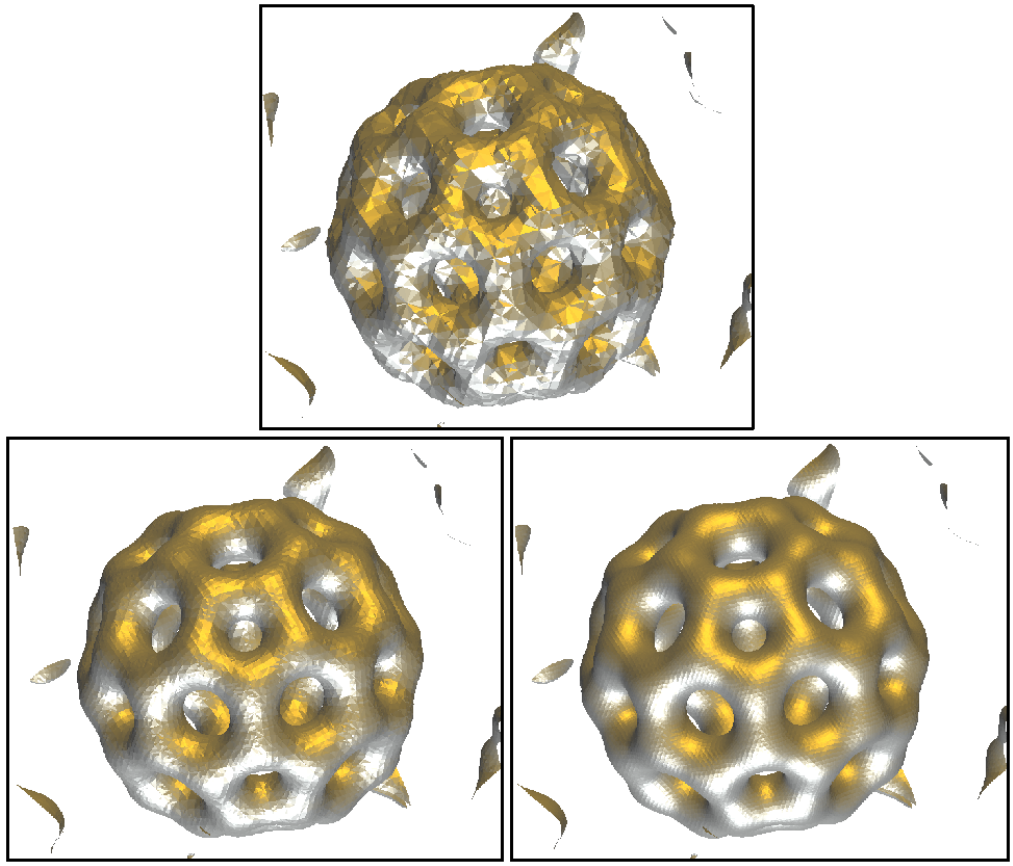

Fig. 22. Flat shaded adaptive isosurfaces are extracted from a 129^3 sized Bucky Ball data set (courtesy of AVS Internation Centre). We consider the hierarchical error indicator for threshold values $\epsilon = 0.02, 0.004, 0.0$.

thereby, a considerable saving of CPU and rendering time. This enables interactive visualization even for very large data sets. Combined with a procedural access to the user data, especially, economically stored hierarchies of millions of elements can be handled efficiently on standard workstations. The presentation is stimulated mainly by the strong relations to adaptive numerical methods and multiscale numerical analysis. The presented concept covers very general grid types, different methods of local error measurement, and local data focusing. It applies to time-dependent data as well and allows a continuous level of detail. Algorithmic details have been kept at a minimum to concentrate on a compact conceptual discussion. We especially regard hierarchical and adaptive methods for nonnested function spaces, gridless discretizations, and particle tracing type methods as interesting fields for future research. Furthermore, the application of related methods to direct volume rendering will be the subject of a forthcoming publication.

ACKNOWLEDGMENTS

The authors thank W. Kinzelbach and S. Oswald from Zürich, G. Wittum and K. Johannsen from Stuttgart, and A. Schmidt and K.G. Siebert from Freiburg for providing numerical data on various grid types. They especially acknowledge many stimulating discussions and other, especially implementational, support from A. Dahr, T. Gerstner, R. Neubauer, M. Metscher, W. Rosenbaum, A. Schmidt, R. Schwörer, U. Weikard, and M. Wierse.

REFERENCES

[1] R.E. Bank, T. Dupont, and H. Yserentant, "The Hierarchical Basis Multigrid Method," *Numer. Math.*, vol. 52, pp. 427-458, 1988.

[2] R.E. Bank and A. Weiser, "Some A Posteriori Error Estimators for Elliptic Partial Differential Equations," *Math. Computation*, vol. 44, pp. 283-301, 1985.

[3] E. Bänsch, "Local Mesh Refinement in 2 and 3 Dimensions," *IMPACT of Computing in Science and Eng.*, vol. 3, pp. 181-191, 1991.

[4] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose, "Toolglass and Magic Lenses: The See–Through Interface," *Proc. SIGGRAPH '93*, pp. 73-80, Anaheim, Calif., 1-6 Aug. 1993.

[5] G.-P. Bonneau, S. Hahmann, and G.M. Nielson, "BLaC-Wavelets: A Multiresolution Analysis with Non-Nested Spaces," *Proc. Visualization*, 1996.

[6] C. Schwab and M. Suri, "The *p* and *hp* Versions of the Finite Element Method for Problems with Boundary Layers," *Math. Computation*, vol. 65, no. 216, pp. 1,403-1,429, 1996.

[7] A. Certain, J. Popovic, T. DeRose, T. Duchamp, D. Salesin, and W. Stuetzle, "Interactive Multiresolution Surface Viewing," *Proc. SIGGRAPH '96 Conf.*, pp. 91-98, 1996.

[8] P. Cignoni, L. De Floriani, C. Montoni, E. Puppo, and R. Scopigno, "Multiresolution Modeling and Visualization of Volume Data Based on Simplicial Complexes," *Proc. 1994 Symp. Volume Visualization*, pp. 19-26, 1994.

[9] P. Cignoni, C. Montani, E. Puppo, and R. Scopigno, "Optimal Isosurface Extraction From Irregular Volume Data," *Proc. 1996 Volume Visualization Symp.*, pp. 31-38, 1996.

[10] P. Cignoni, C. Montani, and R. Scopigno, "MagicSphere: An Insight Tool for 3D Data Visualization," *Computer Graphics Forum*, vol. 13, no. 3, pp. 317-328, 1994.

[11] P. Cignoni, C. Montani, and R. Scopigno, "A Comparison of Mesh Simplification Algorithms," *Computers and Graphics*, vol. 22, pp. 37-54, 1998.

[12] D. Cohen-Or and Y. Levanoni, "Temporal Continuity of Levels of Detail in Delaunay Triangulated Terrain," *Proc. Visualization*, 1996.

[13] L. De Floriani, P. Magillo, and E. Puppo, "Building and Traversing a Surface at Variable Resolution," *Proc. IEEE Visualization '97*, Oct. 1997.
[14] L. De Floriani, P. Marzano, and E. Puppo, "Multiresolution Models for Topographic Surface Description," *The Visual Computer*, vol. 12, no. 7, pp. 317-345, 1996.
[15] L. De Floriani and E. Puppo, "Hierarchical Triangulation for Multiresolution Surface Description Geometric Design," *ACM Trans. Graphics*, vol. 14, no. 4, pp. 363-411, 1995.
[16] M. Dellnitz, A. Hohmann, O. Junge, and M. Rumpf, "Exploring Invariant Sets and Invariant Measures," *Chaos*, vol. 7, pp. 221-228, 1997.
[17] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," *Proc. SIGGRAPH '95 Conf.*, pp. 173-182, 1995.
[18] F. Bornemann, B. Erdmann, and R. Kornhuber, "Adaptive Multilevel Methods in Three Space Dimensions," *Int'l J. Numerical Methods in Eng.*, vol. 36, no. 18, pp. 3,187-3,203, 1993.
[19] B. Falcidieno, S. Orgolesu, C. Pizzi, A. Sanguineti, and M. Spagnuolo, "High Fidelity Digital Terrain Modelling for the Reconstruction of the Antarctic Sea Floor," *J. Visualization and Computer Animation*, vol. 7, no. 3, pp. 177-188, 1996.
[20] N. Faust, L. Hodges, D. Koller, P. Lindstrom, W. Ribarsky, and G. Turner, "Real-Time, Continuous Level of Detail Rendering of Hight Fields," *IEEE Computer Graphics and Applications*, pp. 109-117, 1996.
[21] I. Gargantini, "Linear Octrees for Fast Processing of Three-Dimensional Objects," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 365-374, 1982.
[22] M.H. Ghavamnia and X.D. Yang, "Direct Rendering of Laplacian Compressed Volume Data," *Proc. Visualization*, pp. 192-199, 1995.
[23] M. Giles and R. Haimes, "Advanced Interactive Visualization for cfd," *Computing Systems in Eng.*, vol. 1, no. 10, pp. 51-62, 1990.
[24] M. Griebel, *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen*. Teubner, 1994.
[25] M.H. Gross and R.G. Staadt, "Fast Multiresolution Surface Meshing," *Proc. Visualization*, pp. 135-142, 1995.
[26] R. Grosso, C. Luerig, and T. Ertl, "The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data," *Proc. Visualization*, 1997.
[27] W. Hackbusch, *Multi-Grid Methods and Applications*, Berlin / Heidelberg: Springer, 1985.
[28] B. Hamann, "A Data Reduction Scheme for Triangulated Surfaces," *Computer Aided Geometric Design*, vol. 11, pp. 197-214, 1994.
[29] B. Hamann and J. Chen, "Data Point Selection for Piecewise Trilinear Approximation," *Computer Aided Geometric Design*, vol. 11, 1994.
[30] *On Robust and Adaptive Multi-Grid Methods*, P.W. Hemker et al., eds. Birkhaeuser, 1994.
[31] H. Hoppe, "Progressive Meshes," *Proc. SIGGRAPH '96 Conf.*, pp. 99-108, 1996.
[32] T. Itoh and K. Koyamada, "Automatic Isosurface Propagation Using an Extrema Graph and Sorted Boundart Cell Lists," *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 4, pp. 319-327, Dec. 1995.
[33] T. Itoh, Y. Yamaguchi, and K. Koyamada, "Volume Thinning for Automatic Isosurface Propagation," *Proc. Visualization '96*, 1996.
[34] J. Fröhlich and K. Schneider, "An Adaptive Wavelet-Vaguelette Algorithm for the Solution of pdes," *J. Computers and Physics*, vol. 130, no. 2, pp. 174-190, 1997.
[35] J. Maubach, "Local Bisection Refinement for *n*-Simplicial Grids Generated by Reflection," *SIAM J. Science and Computers*, vol. 16, pp. 210-227, 1995.
[36] R. Klein, G. Liebich, and W. Straßer, "Mesh Reduction With Error Control," *Proc. Visualization*, 1996.
[37] D. Laur and P. Hanrahan, "Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering," *Proc. Computer Graphics (SIGGRAPH '91)*, vol. 25, pp. 285-288, 1991.
[38] M. Levoy, "Efficient Ray Tracing of Volume Data," *ACM Trans. Graphics*, vol. 9, no. 3, pp. 245-261, 1990.
[39] Y. Livnat, H.W. Shen, and C.R. Johnson, "A Near Optimal Isosurface Extraction Algorithm Using the Span Space," *IEEE Trans. Visualization and Computer Graphics*, vol. 2, no. 1, pp. 73-83, Mar. 1996.
[40] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163-169, 1987.
[41] W.F. Mitchell, "A Comparison of Adaptive Refinement Techniques for Elliptic Problems," *ACM Trans. Math. Software*, vol. 15, no. 4, pp. 326-347, 1989.
[42] S. Muraki, "Volume Data and Wavelet Transform," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 50-56, 1993.
[43] R. Neubauer, M. Ohlberger, M. Rumpf, and R. Schwörer, "Efficient Visualization of Large Scale Data on Hierarchical Meshes," *Visualization in Scientific Computing '97*, W. Lefer and M. Grave, eds. Springer, 1997.
[44] R.H. Nochetto, "Pointwise A Posteriori Error Estimates for Elliptic Problems on Highly Graded Meshes," *Math. Computation*, vol. 64, pp. 1-22, 1995.
[45] M. Ohlberger and M. Rumpf, "Hierarchical and Adaptive Visualization on Nested Grids," *Computing*, vol. 59, pp. 269-285, 1997.
[46] E. Ossa, *Topologie*. Braunschweig / Wiesbaden: Vieweg, 1992.
[47] P.W. Hemker, "On the Structure of an Adaptive Multi-Level Algorithm," *BIT*, vol. 20, pp. 289-301, 1980.
[48] K. Polthier and M. Rumpf, "A Concept for Timedependent Processes," *Scientific Visualization*, M. Goebel, H. Mueller, and B. Urban, eds. Springer-Verlag, 1995.
[49] R. Stenberg and M. Suri, "Mixed *hp* Finite Element Methods for Problems in Elasticity and Stokes Flow," *Numer. Math.*, vol. 72, no. 3, pp. 367-389, 1996.
[50] M.C. Rivara, "Algorithms for Refining Triangular Grids Suitable for Adaptive and Multigrid Techniques," *Int'l J. Numerical Methods in Eng.*, vol. 20, pp. 745-756, 1984.
[51] M. Rumpf, A. Schmidt, and K.G. Siebert, "Functions Defining Arbitrary Meshes a Flexible Interface Between Numerical Data and Visualization Routines," *Computer Graphics Forum*, vol. 15, no. 2, pp. 129-141, 1996.
[52] P. Schröder and W. Sweldens, "Spherical Wavelets: Efficiently Representing Functions on the Sphere," *Proc. SIGGRAPH '95 Conf.*, pp. 161-172, 1995.
[53] W.J. Schroeder, J.A. Zarge, and W.A. Lorensen, "Decimation of Triangle Meshes," *Proc. Computer Graphics (SIGGRAPH '92)*, vol. 26, pp. 65-70, 1992.
[54] R. Shekhar, E. Fayyad, R. Yagel, and J.F. Cornhill, "Octree-Based Decimation of Marching Cubes Surfaces," *Proc. Visualization*, 1996.
[55] H.-W. Shen, C.D. Hansen, Y. Livnat, and C.R. Johnson, "Isosurfacing in Span Space with Utmost Efficiency (ISSUE)," *Proc. Visualization*, 1996.
[56] K.G. Siebert, "Local Refinement of 3-D-Meshes Consisting of Prisms and Conforming Closure," *IMPACT of Computing in Science and Eng.*, vol. 5, pp. 271-284, 1993.
[57] M. Tamminen and H. Samet, "Efficient Octree Conversion by Connectivity Labeling," *Computer Graphics*, vol. 18, no. 3, pp. 43-51, 1984.
[58] G. Turk, "Re-Tiling Polygonal Surfaces," *Computer Graphics*, vol. 26, no. 2, pp. 55-64, 1992.
[59] J.P. Wilhelms and A. Van Gelder, "Octrees for Faster Isosurface Generation," *Computer Graphics (San Diego Workshop Volume Visualization)*, vol. 24, no. 5, pp. 57-62, 1990.
[60] J.P. Wilhelms, A. Van Gelder, P. Tarantino, and J. Gibbs, "Hierarchical and Parallelizable Direct Volume Rendering for Irregular and Multiple Grids," *Proc. Visualization*, 1996.
[61] L. Williams, "Pyramidal Parametrics," *Computer Graphics*, vol. 17, no. 3, 1983.
[62] S. Yang and C. Cooke, "Data Compression Based on the Cubic *B*-Spline Wavelet with Uniform Two-Scale Relation," *Math. Computational Modeling*, vol. 23, no. 7, pp. 73-88, 1996.
[63] Y. Zhou, B. Chen, and A. Kaufman, "Multiresolution Tetrahedral Framework for Visualizing Volume Data," *Proc. IEEE Visualization '97*, 1997.

Mario Ohlberger received his Diplom degree in mathematics from Freiburg University in Germany in 1996. Currently, he is a research assistant at the Institute for Applied Mathematics at Freiburg University. His research interests are numerical methods for hyperbolic partial differential equations, porous media flow and, especially, a posteriori error analysis. Moreover, he is concerned with the development of efficient multiresolution methods for large numerical data sets.

Martin Rumpf received his Diplom degree and his PhD degree in mathematics from Bonn University in Germany in 1989 and 1992, respectively. He has been a professor of applied math-ematics at Bonn University since 1996. He was a postdoctoral research assistant at Freiburg University for four years. His research interests are numerical methods for nonlinear partial differential equations and, especially, adaptive finite element methods. Furthermore, he is concerned with new flow visualization techniques and efficient multiscale methods in scientific visualization.