An Adaptive Finite Element Method for Large Scale Image Processing

T. Preußer and M. Rumpf

Institut für Angewandte Mathematik, Universität Bonn,
Wegelerstr. 6, 53115 Bonn, Germany

E-mail: [tpreuss,rumpf]@iam.uni-bonn.de

Nonlinear diffusion methods have proved to be powerful methods in the processing of 2D and 3D images. They allow a denoising and smoothing of image intensities while retaining and enhancing edges. As time evolves in the corresponding process, a scale of successively coarser image details is generated. Certain features, however, remain highly resolved and sharp. On the other hand, compression is an important topic in image processing as well. Here a method is presented which combines the two aspects in an efficient way. It is based on a semi–implicit Finite Element implementation of nonlinear diffusion. Error indicators guide a successive coarsening process. This leads to locally coarse grids in areas of resulting smooth image intensity, while enhanced edges are still resolved on fine grid levels. Special emphasis has been put on algorithmical aspects such as storage requirements and efficiency. Furthermore, a new nonlinear anisotropic diffusion method for vector field visualization is presented.

1. INTR ODUCTION

Nonlinear diffusion methods in image processing have been known for a long time. In 1987 Perona and Malik [17] introduced a continuous diffusion model which allows the denoising of images together with the enhancing of edges. The diffusion driven evolution is started on an initial image intensity. In general, it is either noisy because of unavoidable measurement errors, or it carries partially hidden patterns which have to be intensified and outlined [9, 24]. Such an image smoothing and feature restoration process can be understood as a successive coarsening while certain structures are retained on a fine scale – an approach which is closely related to the major techniques in image compression.

Finite Element methods are widespread to discretize and appropriately implement the diffusion based models. Their general convergence properties were studied for instance by Kačur and Mikula [13]. Furthermore, Schnörr applied Finite Elements in a variational approach to image processing [20]. In various areas of scientific computing adaptive Finite

Element methods [6, 4] have been incorporated to substantially reduce the required degrees of freedom while conserving the approximation quality of the numerical solution. Thereby locally defined reliable error estimators or some error indicators steer the local grid refinement, respectively coarsening [23, 5]. The image intensities resulting from the nonlinear parabolic evolution are obviously well-suited to be resolved on adaptive grids. As time evolves, a successive coarsening in areas of smooth image intensity is near at hand. For instance in case of an d–dimensional image, where the image intensity is constant on piecewise smoothly bounded regions, we obtain the same image quality on a $O(N^{d-1}\log(N))$ complex adaptive grid as on a $O(N^d)$ regular grid. The cost of the numerical algorithm, the storage requirements, and the transmission time on computer networks scale with this complexity in terms of actual degrees of freedom.

These efficiency perspectives have first been studied by Bänsch and Mikula [3], who presented an adaptive Finite Element method. This method is based on simplicial grids generated by bisection and then again successively coarsened in the diffusion process. The major shortcoming of their approach is the enormous memory requirement for the data structures describing the adaptive grid and the sparse matrices used in the linear solver in each implicit time. Therefore, large 3D images – as they are widespread in medical imaging – are difficult to manage on moderately sized workstations.

Here we present an adaptive multilevel Finite Element method which avoids these shortcomings and comes along with minimal storage requirements. The specific ingredients of our method are:

- adaptive quad– and octrees, with accompanying piecewise bilinear, respectively trilinear Finite Element spaces are procedurally handled only,
- error indicators on grid nodes and a suitable threshold value implicitly describe the adaptive grid (no explicit adaptive grid structure is required),
- invoking a certain saturation condition for the nodal indicators, we ensure robustness and one level transitions only on the resulting adaptive grid,
- the adaptive Finite Element space is defined as an implicitly constrained discrete space on the full grid,
- the grid is completely handled procedurally,
- and instead of dealing with explicitly stored sparse matrices, the hierarchically preconditioned linear solver in each timestep uses ”on–the–fly” matrix multiplication based on efficient grid traversals.

Let us mention that this approach benefits from general and efficient multilevel data post processing methodology [16, 19] and is related to the multilevel methods discussed in [1, 25].

Finally, as a – to our knowledge – new area of application we will present a scale space method in vector field visualization. Flow visualization is an important task in scientific visualization. Simply drawing vector plots at nodes of some overlayed regular grid in general produces visual clutter. The central goal is to come up with inituitive methods with more comprehensible results. They should provide an overall as well as detailed view on the flow patterns. Several techniques generating such textures based on discrete models have been presented [8, 15, 21, 22]. We ask for a continuous model which leads to stretched streamline type patterns, which are aligned to the vector field. Furthermore, the possibility to successively coarsen this pattern is obviously a desirable property. For the generation of such field aligned flow patterns we apply anisotropic nonlinear diffusion. A matrix valued diffusion coefficient controls the anisotropy as in Weickert's method [26] to restore and enhance lower dimensional structures in images.

2. FE-DISCRETIZATION OF NONLINEAR DIFFUSION

Let us look at the modified Perona-Malik [17] model proposed by Catté, Lions, Morel, and Coll [9]. Without any restriction we consider the domain $\Omega := [0,1]^d$, $d = 2, 3$ and ask for solution of the following nonlinear parabolic, boundary and initial value problem: Find $\rho : \mathbb{R}^+ \times \Omega \to \mathbb{R}^m$ such that

$$\begin{aligned} \tfrac{\partial}{\partial t}\rho - \operatorname{div}\left(A(\nabla\rho_\epsilon)\nabla\rho\right) &= f(\rho)\,, &&\text{in } \mathbb{R}^+ \times \Omega\,,\\ \rho(0,\cdot) &= \rho_0 \quad , &&\text{on } \Omega\,,\\ \tfrac{\partial}{\partial\nu}\,\rho &= 0 \quad\;\, , &&\text{on } \mathbb{R}^+ \times \partial\Omega. \end{aligned}$$

where in the basic model $A = g$ for a non negative monotone decreasing function $g : \mathbb{R}_0^+ \to \mathbb{R}^+$ satisfying $\lim_{s\to\infty} g(s) = 0$, e. g. $g(s) = (1+s^2)^{-1}$, and ρ_ϵ is a mollification of ρ with some smoothing kernel. We interpret the solution ρ for increasing $t \in \mathbb{R}^+$ to be a successively filtered version of ρ_0. With respect to the shape of g, the diffusion is of regularized backward type [14] in regions of high image gradients, while noisy regions of ρ_0 will be smoothed by dominant diffusion.

We solve this problem numerically by applying a bilinear, respectively trilinear conforming Finite Element discretization on an adaptive quadrilateral, respectively hexahedral grid. In time a semi-implicit second order Euler scheme is used. As it has become standard the scheme is semi-implicit with respect to the evaluation of the nonlinear diffusion coefficient g and the right hand side. The computation of the mollified intensity ρ_ϵ is based on a single short timestep of the corresponding heat equation (linear diffusion) with given data ρ [13]. In the ith timestep we have to solve the linear system

$$(M + \tau L(\rho_\epsilon))\bar{\rho}^i = M\bar{\rho}^{i-1} + F\,,$$

where $\bar{\rho}^i$ is the corresponding solution vector consisting of the nodal values, τ the current timestep, M is the lumped mass matrix, $L(\rho_\epsilon)$ the weighted stiffness matrix and F the vector representation of the right hand side. The growth of F in the application is moderate compared to chemical reaction diffusion equations. Therefore we have not recognized any instabilities with this source term. The stiffness matrix and the right hand side are computed by applying the midpoint quadrature rule.

The above linear system as well as the linear system resulting from the mollification by the heat equation kernel is solved by a preconditioned conjugate gradient method. We use the Bramble–Pasciak–Xu preconditioning [7], thus making appropriate use of the given grid hierarchy.

As already mentioned above, a peculiarity of our scheme is that no matrices are stored explicitly. Instead, the multiplication of the mass, respectively the stiffness matrix with a coefficient vector consisting of nodal values is done procedurally. Therefore, in each step the hierarchical and adaptive grid is traversed and element wise local contributions are evaluated and successively assembled on the resulting coefficient vector. Thus we avoid storing the matrices explicitly. Otherwise we would have been unable to manage typical 3D applications with more than 10 million nodes. Furthermore, this procedural access

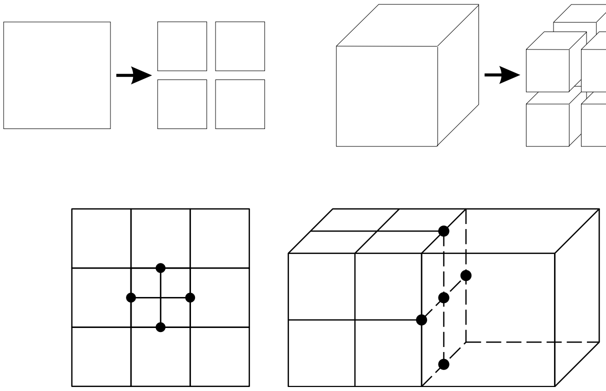

FIG. 1. On the top element types in two and three dimensions and their refinements are shown and on the bottom grid configurations with hanging nodes are depicted.

carries strong provisions for code optimization with respect to a cache optimal numbering of the nodes.

3. GRID ADAPTIVITY AND ERROR INDICATORS

In this section we will discuss an adaptive approach to the problem of nonlinear diffusion. We will especially focus on the choice and the handling of error indicator values on the grid nodes which steer the adaptive algorithm. It will be outlined that *saturation* plays an essential role in the robustness and implementability of the proposed algorithm. In fact, solely referring to saturated error indicator information and not to some explicit grid hierarchy enables us to define and handle appropriate adaptive meshes for the nonlinear diffusion algorithm. Let us assume the dimension of our image to be $(2^{l_{\max}} + 1)$ in each direction for some $l_{\max} \in \mathbb{N}$. The degrees of freedom are interpretated as nodal values of a regular grid with $2^{l_{\max} d}$ elements for $d = 2, 3$. Above this fine grid level we define a quadtree, respectively octree hierarchy of elements with $l_{\max} + 1$ grid levels. In each local refinement step an element E is subdivided into a set $\mathcal{C}(E)$ of 2^d child elements (cf. Fig. 1). Vice versa we denote by $\mathcal{P}(E)$ the ancestor of an element E. In each refinement step new grid nodes x appear. They are expressed by weighted sums over their parent nodes $x_{\mathcal{P}} \in \mathcal{P}(x)$ from the set of coarser grid level nodes:

$$x = \sum_{x_{\mathcal{P}} \in \mathcal{P}(x)} \omega(x, x_{\mathcal{P}}) x_{\mathcal{P}}.$$

The weights $\omega(x, x_{\mathcal{P}}) \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$ depend on the type of the new node, which might be the center of a 1D edge, a 2D face, or a 3D hexahedron. Let us denote by $\mathcal{N}_{\mathcal{C}}(E)$ the set of new nodes on an element E.

We suppose the grid to be adaptive. I. e. depending on data the recursive refinement is stopped locally on elements of different grid levels. Thereby a sequence of nested successively refined grids $\{\mathcal{M}^l\}_{0 \leq l \leq l_{\max}}$ is generated. On this sequence we define discrete function spaces $\{\mathcal{V}^l\}_{0 \leq l \leq l_{\max}}$ consisting of continuous piecewise bilinear, respectively trilinear functions, which are ordered by set inclusion:

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

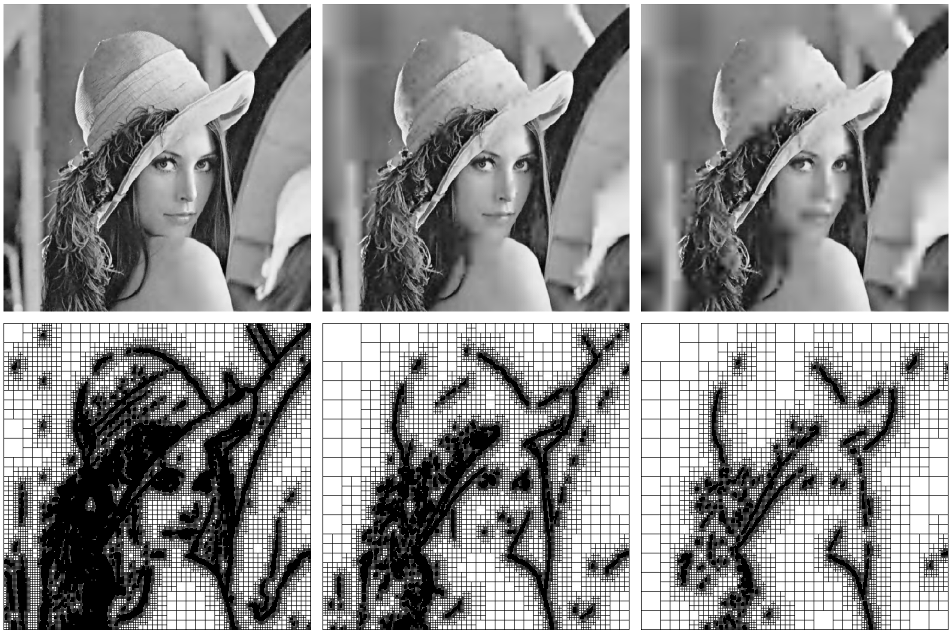

FIG. 2. Several adaptive grids of a smooth image. From left to right the number of unknowns is 85481, 44256, and 21600 respectively. A full grid would contain 263169 unknowns.

Let $\{\phi_i^l\}_i$ denote the basis of $\mathcal{V}^l$ consisting of hat-functions, i.e. if $\{x_1,\ldots,x_N\}$ denotes the set of non constrained vertices of $\mathcal{M}^l$, we have $\phi_i^l(x^j) = \delta_{ij}, j = 1,\ldots,N$. Thereby a vertex is called *constrained*, or a *hanging node*, if it is not generated by refinement on every adjacent element (cf. Fig. 1). On adaptive quadtrees, respectively octrees such hanging nodes are unavoidable. The handling of the corresponding nodal values is crucial for the efficiency of the resulting adaptive numerical algorithm. We choose an efficient implicit processing which will be described below.

Usually, for timedependent problems a grid modification consisting of the refinement and coarsening of elements is necessary at certain time steps. In our setting we start on the initial fine grid $\mathcal{M}^{l_{\max}}$ and it suffices to coarsen elements, since there is in general no spatial movement of the image edges and complete information of the image is coded on the initial grid. (cf. Fig. 2). This coarsening is obtained by prescribing a data dependent, boolean valued stopping criterion $S(E)$ on elements, which implies local stopping in a recursive depth first traversal of the hierarchical grid. It turned out to be suitable to let this element stopping criterion depend on a corresponding criterion $S(x)$ on the nodes, respectively basis functions, i. e. we define

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

$\mathcal{S}(\cdot)$ distinguishes which degrees of freedom are actually important, respectively which nodal values can be generated by interpolation of some coarse grid function. If $\eta(x)$ is some error indicator on the nodes x and ϵ is a prescribed threshold value, we obtain such an interpolation criterion by

$$\mathcal{S}(x) := (\eta(x) \leq \epsilon\,)\,.$$

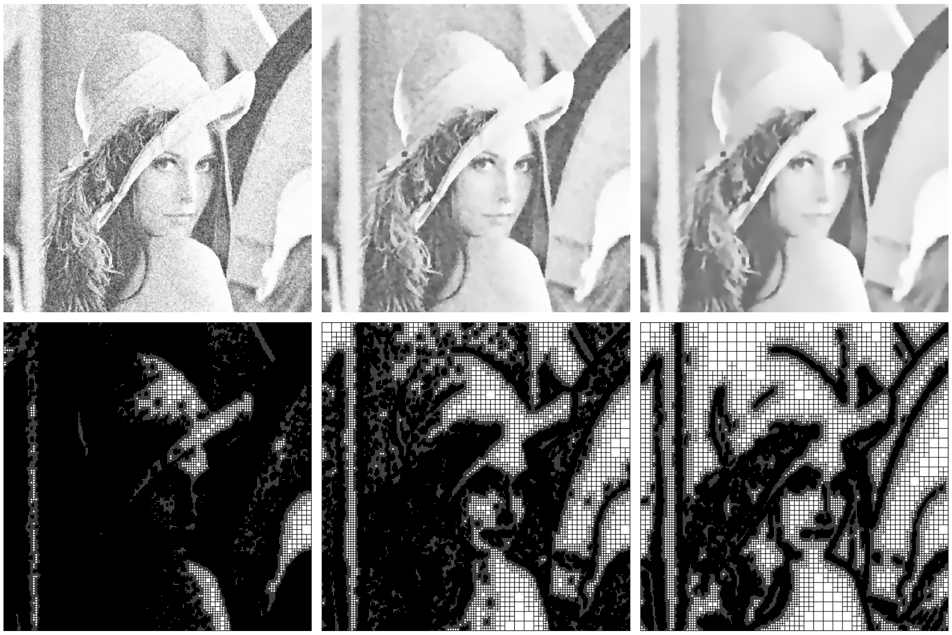

FIG. 3. From left to right several timesteps of the selective image smoothing on adaptive grids are shown.

Given an image intensity $\rho \in \mathcal{V}^{l_{\max}}$ an intuitive choice for an error indicator is $\eta(x) := |\nabla \rho(x)|$, because the gradient of an image ρ acts like an edge indicator. Hence in regions with nearly constant intensity the grid will be coarsened substantially, whereas in the vicinity of high gradients, indicating preservable edges, the grid size is kept fine.

The stopping criterion on elements is motivated by the fact that in the next refinement step only interpolated nodal values would appear. To ensure every descendent nodal value on such an element – also those on finer grid levels – to be interpolated we require the following natural saturation condition on the error indicator:

(Saturation Condition) *An error indicator value $\eta(x)$ for $x \in \mathcal{N}(E)$ is always greater than every error indicator $\eta(x_{\mathcal{C}})$ for $x_{\mathcal{C}} \in \mathcal{N}_{\mathcal{C}}(E)$.*

In general the saturation condition is not fulfilled, but we can modify the error indicator in a preprocessing step. Typically, this turns out to be necessary only on coarse grid levels. A simple update algorithm for an error indicator η and thereby the corresponding projection criterion $\mathcal{S}$ is the following bottom-up traversal of the grid hierarchy, starting on the second finest level and ending on the macro grid.

```
for l= $l_{\max}$-1 to 0 step -1 do
   for each element $E$ of $\mathcal{M}^l$ do
      $\eta^*$ := max$_{x \in \mathcal{N}_{\mathcal{C}}(E)}$ $\eta(x)$;
      for all $x \in \mathcal{N}(E)$ do if($\eta(x)$ < $\eta^*$) $\eta(x)$ = $\eta^*$;
```

Let us emphasize that a depth first traversal of the hierarchy in the adjustment procedure would not be sufficient. This saturation process "transports" fine grid error information up to coarse grid level and prevents us from overlooking important fine grid details [16]. Furthermore, the saturation condition comes along with another desirable property. The corresponding element stopping criterion implies only one level grid transitions at element

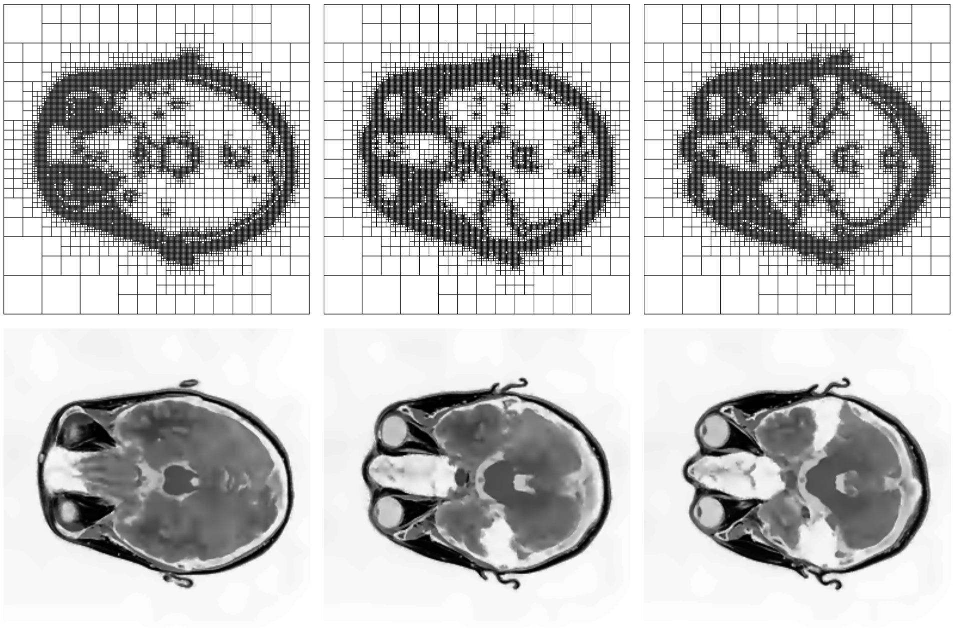

FIG. 4. Nonlinear diffusion has been applied to a 3d medical data set. Several slices through the adaptive grid are depicted showing the corresponding image intensity as well as the intersection lines with element faces.

faces of the actual adaptive grid (cf. Fig. 1). Thus, the possible hanging node configurations confine to the basic one level cases. I. e. any open face and any edge of an element E contains at most one hanging node (cf. [11] for a general treatment of hanging nodes). Finally, this has straightforward implications on the assurance of continuity of discrete Finite Element functions and the corresponding matrix assembly in the implementation of our nonlinear diffusion algorithm. In general on regular grids the continuity is guaranteed by identifying each local degree of freedom (dof) with the global dof in the assembly of the global stiffness matrices and the right hand side of the corresponding discrete linear problem. However, hanging nodes of the adaptive grid do not represent dofs, due to their dependence upon other dofs. Therefore, when assembling the global stiffness matrices, we have to distribute the contribution of the hanging nodes onto the constraining dofs. This is nothing else but procedurally respecting the appropriate interpolation conditions. For future use let us introduce the following notation:

- `NDEP`(i) = Number of constraints of the node with local index i of an element. We define `NDEP`(i):=1 if the node is not constrained.
- `CCOEF`(i,j) = List of constrained coefficients. In our case we always have `CCOEF`$(i,j) = 1/$`NDEP`(i) for $j = 1, \ldots,$ `NDEP`(i).
- `CDOFM`(i,j) = List of global dofs that constrain the node i, $j = 1, \ldots,$ `NDEP`(i). For non-hanging nodes `CDOFM`$(i,1)$ coincides with the global dof of node i.

The `CCOEF`–values are identical to the weights in the above node generation rule. Figure 3 shows the application of the resulting adaptive algorithm to selectively smoothen some noisy image. In Figure 4 and 5 we have applied the algorithm to a 3D data set [12].

4. PROCEDURAL GRID HANDLING AND MATRIX MULTIPLICATION

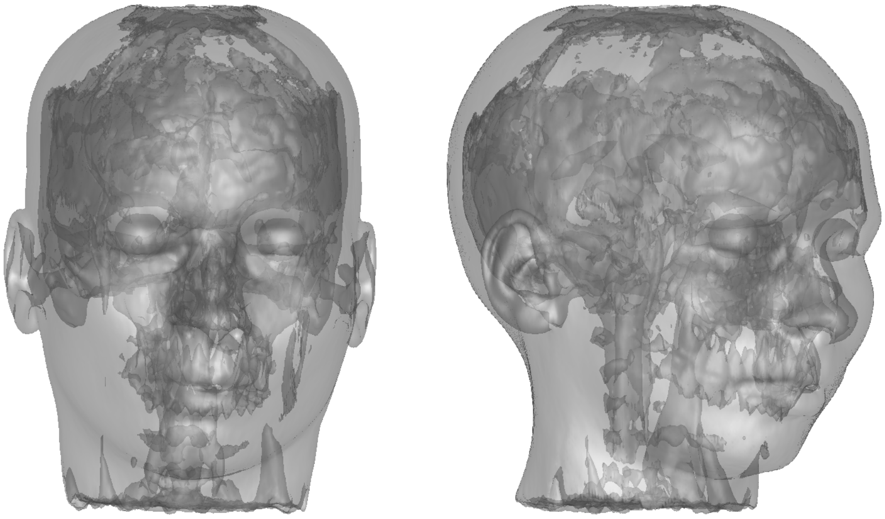

FIG. 5. A transparent isosurface visualization of the brain data set, smoothed by nonlinear diffusion (cf. Fig. 4).

As already mentioned in Section 2 the hierarchical grid is handled solely procedurally and necessary matrix multiplications in the linear system solver are performed on–the–fly traversing the adaptive grid recursively. Let us describe this now in more detail.

Traversing the grid, information that is needed to identify an element E will be generated recursively. If this recursive traversal routine reaches a leaf element of the adaptive grid, i. e. an element for which $S(E)$ is true, a callback-method will perform some action on that element. For instance it calculates the local right hand side. An element E is identified by the index vector of its lower left corner, its grid level and its refinement-type. Every other information like the element's size, the mapping of local dofs to global dofs, and the constrained dofs will be stored in lookup tables as already mentioned in Section 3. In 2D the hierarchical traversal can be formulated in pseudo code as follows:

```
sub traverse(i, j, lev, refType, callback, params)
   if (lev $\neq l_{\max}$) and $\neg S($element$)$ do
      offset = $2^{l_{\max}-\mathtt{lev}-1}$;
      traverse(i, j, lev+1, 0, callback, params);
      traverse(i + offset, j, lev+1, 1, callback, params);
      traverse(i + offset, j + offset, lev+1, 2, callback, params);
      traverse(i, j + offset, lev+1, 3, callback, params);
   else callback(i, j, lev, refType, params);
```

We can also formulate the “on-the-fly” matrix-vector multiplication using this traversal with callback. Multiplying a given vector u with the matrix $M + \tau L(\rho_\epsilon)$ and assembling the result in a vector w requires the following local callback procedure:

```
sub matrixProduct(i, j, lev, refType, (u,w))
   for each pair l,k of local dofs
      for lc=1 to NDEP(l), kc=1 to NDEP(k)
         w(CDOFM(l,kc)) += localMatrix(l,k) *
```

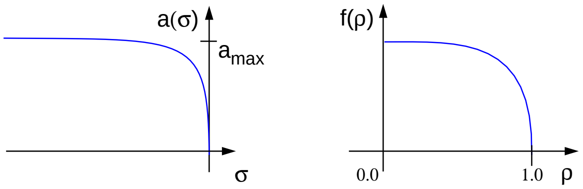

FIG. 6. The diffusion coefficient and the right hand side in the continuous segmentation algorithm.

```
                                   CCOEFF(l,kc) * CCOEFF(k,lc) * u(CDOFM(k,lc));
```

Similarily the adaptive BPX preconditioning can be implemented.

5. SEGMENTATION

Once a noisy 2D or 3D image has been processed by the nonlinear diffusion method with simultaneous grid coarsening we can consider efficient segmentation or Finite Element simulation tasks on the compressed data base. Here, we will briefly outline a continuous watershed algorithm, and show some results for large medical data sets.

Given a C^0 function $\sigma : \Omega \to \mathbb{R}$ indicating the segment boundary, and a seed point x, we define a corresponding segment by

$$\begin{aligned}\mathcal{S}(x,\sigma) \;&:=\; \{y \in \Omega \,|\, \exists\, C^0 \text{ curve } \gamma : [0,1] \to \Omega,\ \gamma(0) = x,\ \gamma(1) = y,\\ &\qquad \sigma(\gamma(t)) \leq 0 \,\forall t \in [0,1]\,\}\,.\end{aligned}$$

Examples are connected subsets of the preimage of a grey value interval ($\sigma_1(\rho) = -\chi_{[\rho^-,\rho^+]}(\rho)$), domains bounded by large gradients ($\sigma_2(\rho) = \|\nabla\rho\| - M$), or combined criteria ($\omega_1\sigma_1 + \omega_2\sigma_2$, $\max\{\sigma_1,\sigma_2\}$). Then a diffusive watershed algorithm consists of solving the following diffusion problem:

$$\begin{aligned}\frac{\partial}{\partial t}\rho - \operatorname{div}\left(a(\sigma)\nabla\rho\right) \;&=\; f(\rho) \text{ in } \mathbb{R}^+ \times \Omega\,,\\ \rho(0) \;&=\; \rho_0 \text{ on } \Omega\,,\\ \frac{\partial}{\partial \nu}\,\rho \;&=\; 0 \text{ on } \mathbb{R}^+ \times \partial\Omega\,.\end{aligned}$$

where ρ_0 is a seed "drop", i. e. a positive function with compact support in a small neighborhood of the seed point x. Thereby the diffusion coefficient $a(\sigma)$ is positive inside the segment and vanishes at the segment boundary. Considering in addition an anisotropic diffusion model [24], tangential smoothing along the segment boundary can be incorporated (cf. Fig. 6). Obviously the algorithm can easily be implemented based on the already available nonlinear diffusion code. Figure 7, 8 show results obtained by this method.

Let us finally emphasize that the segmentation benefits considerably of the adaptive resolution of edges or surfaces in the data set, which are typically identical with the segment boundaries.

6. APPLICATION TO FLOW VISUALIZATION

As already sketched in the introduction we will now apply nonlinear anisotropic diffusion to vector field visualization. Thereby we consider diffusive smoothing along streamlines and edge enhancing in the orthogonal directions. Applying this to some initial random

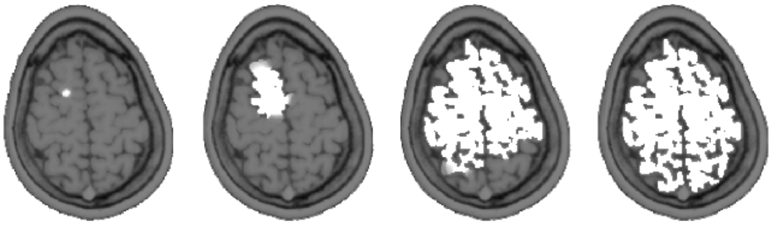

FIG. 7. Brain segmentation on slices of a MRT-image by nonlinear diffusion. Consecutive timesteps of the corresponding evolution are depicted.

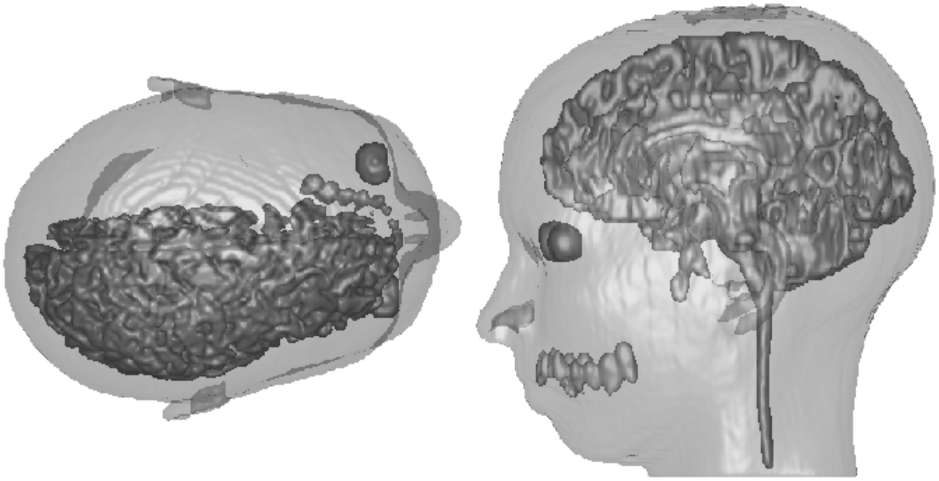

FIG. 8. Results of a brain segmentation in 3D with tangential smoothing based on a gradient segment criterion

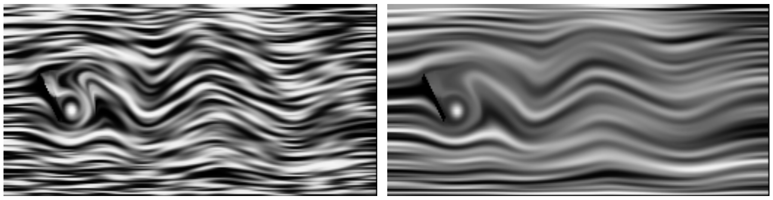

FIG. 9. A single timestep is depicted from the nonlinear diffusion method applied to the vector field describing the flow around an obstacle at a fixed time. A discrete white noise is considered as initial data. We run the evolution on the left for a small and on the right for a large constant diffusion coefficient α.

noise image we generate a scale of successively coarser patterns which represent the flow field.

For a given smooth vector field $v : \Omega \to \mathbb{R}^n$ we define a family of continuous orthogonal mappings $B(v) : \mathbb{R}^n \to SO(n)$ such that $B(v)v = e_0$, where $\{e_i\}_{i=0,\cdots,n-1}$ is the standard base in $\mathbb{R}^n$. We consider a diffusion matrix $A = A(v, \nabla\rho_\epsilon)$ and define

$$A(v,d) = B(v)^T \begin{pmatrix} \alpha(\|v\|) & 0 \\ 0 & g(d) \end{pmatrix} B(v)$$

where $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ controls the linear diffusion in vector field direction, i. e. along streamlines, and the above introduced edge enhancing diffusion coefficient $g(\cdot)$ acts in the orthogonal directions. We may either choose a linear function α or in case of a veloc-

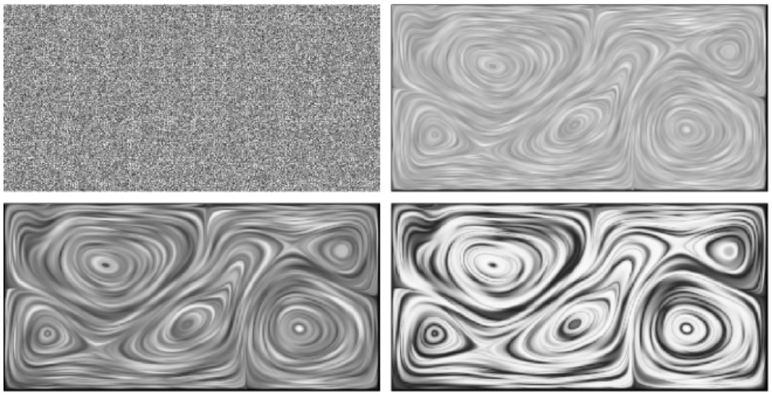

FIG. 10. Several timesteps are depicted from the nonlinear anisotropic evolution applied to a convective flow field in a 2D box.

ity field, which spatially varies over several orders of magnitude, we select a monotone function α with $\alpha(0) > 0$ and $\lim_{s\to\infty} \alpha(s) = \alpha_{\max}$.

Different to the problems studied by Weickert in [26] in our case no canonical initial data is given. To avoid aliasing artifacts we thus choose some random noise ρ_0 of an appropriate frequency range. During the evolution the random pattern will grow upstream and downstream, whereas the edges tangential to these patterns are successively enhanced. Still there is some diffusion perpendicular to the field which supplies us for evolving time with a scale of progressively coarser representation of the flow field. Running the evolution for vanishing right hand side f the image contrast will unfortunately decrease successively. Therefore, we select an appropriate contrast enhancing right hand side $f : [0,1] \to \mathbb{R}^+$ with $f(0) = f(1) = 0$, $f > 0$ on $(0.5, 1)$, and $f < 0$ on $(0, 0.5)$ (cf. reaction diffusion problems in image analysis studied in [2, 10]). Finally we end up with the method of nonlinear anisotropic diffusion to visualize complex vector fields.

We expect an almost everywhere convergence to $\rho(\infty, \cdot) \in \{0, 1\}$ due to the choice of the contrast enhancing function $f(\cdot)$. The set of asymptotic limits significantly influences the richness of the developing pattern. One way to enrich this set significantly is to consider a vector valued diffusion problem, where the contrast enhancing right hand side leads to asymptotic states which are either 0 or located on the sphere in the intensity space. For details we refer to [18]. This method is capable to nicely depict the global structure of flow fields, including saddle points, vortices, and stagnation points on the boundary. Results are shown in Figure 10. Here the anisotropic diffusion method is applied to an incompressible Bénard convection problem in a rectangular box with heating from below and cooling from above. The formation of convection rolls leads to an exchange of temperature.

The anisotropic nonlinear diffusion problem has already been formulated in Section 2 for arbitrary space dimension. Differing from 2D in 3D we have somehow to break up the volume and open up the view to inner regions. Here a further benefit of the vector valued diffusion comes into play. The asymptotic limits - which differ from 0 - are distributed on $S^1 \cap [0,1]^2$. Hence, we reduce the informational content and focus on a ball shaped neighbourhood $B_\delta(\omega)$ of a certain point $\omega \in S^1 \cap [0,1]^2$ (cf. Fig. 11).

7. CONCLUSIONS

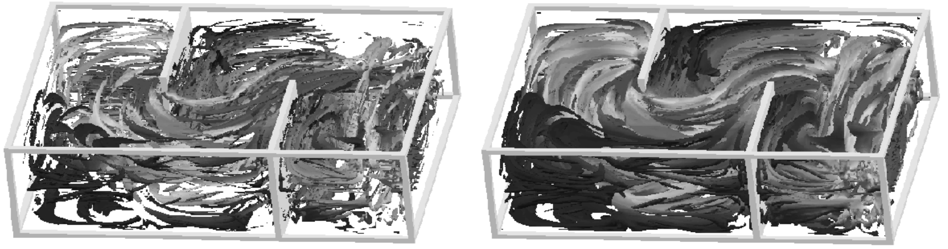

FIG. 11. The incompressible flow in a water basin with two interior walls and an inlet (on the left) and an outlet (on the right) is visualized by anisotropic nonlinear diffusion. Isosurfaces show the preimage of $\partial B_\delta(\omega)$ (for different values of δ) under the vector valued mapping ρ for some point ω on S^1. Color is indicating the velocity.

We have discussed an adaptive Finite Element method for the discretization of nonlinear diffusion methods in large scale image processing. Especially, we have introduced a new method to process adaptive grids and corresponding mass- and stiffness matrices procedurally with out storing any matrix or any graph structure for the hierarchical tree of elements. Thus the method enables the handling of large images (257^3 dofs and more) on moderately sized workstations. The adaptive algorithm is controlled by error indicators on grid nodes. Saturation of the indicator values plays an essential role in the reliability as well as in the concrete implementation of the algorithm. Furthermore corresponding segmentation results and a new method for 2D and 3D flow visualization have been presented.

From the authors' point of view exciting future research directions are the application of true multigrid methods which are able to manage oscillating coefficients. Beyond this, by exploiting cache optimization strategies with respect to an appropriate numbering of the dofs we expect a further significant speed up.

ACKNOWLEDGMENT

The authors would like to acknowledge Karol Mikula and Jarke van Wijk for inspiring discussions and many useful comments on image processing and flow visualization. Furthermore, they thank Eberhard Bänsch from Bremen University for providing the incompressible flow data sets.

REFERENCES

1. S. T. Acton. Multigrid anisotropic diffusion. *IEEE Trans. Image Proc.*, 7:280–291, 1998.
2. L. Alvarez and J. Esclarin. Image quantization using reaction-diffusion equations. *SIAM J. Appl. Math.*, 57:153–175, 1997.
3. E. Bänsch and K. Mikula. A coarsening finite element strategy in image selective smoothing. *Computing and Visualization in Science*, 1:53–63, 1997.
4. P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. Rentz-Reichert, and C. Wieners. Ug - a flexible software toolbox for solving partial differential equations. *Comput. Visual. Sci.*, 1:27–40, 1997.
5. R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.
6. Bornemann, F. and Erdmann, B. and Kornhuber, R. Adaptive multilevel methods in three space dimensions. *Int. J. Numer. Methods Eng.*, 36, No.18:3187–3203, 1993.
7. J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. of Comp.*, 55:1–22, 1990.
8. B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 263–272, Aug. 1993.
9. F. Catté, P. L. Lions, J. M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29:182–193, 1992.
10. G.-H. Cottet and L. Germain. Image processing through reaction combined with nonlinear diffusion. *Math. Comp.*, 61:659–673, 1993.

11. L. Demkowicz, K. Gerdes, C. Schwab, A. Bajer, and T. Walsh. HP90: A general and flexible fortran 90 hp – FE code. Technical Report 97-17, Seminar für Angewandte Mathematik, ETH Zürich, 1997.

12. T. Gerstner and M. Rumpf. Multiresolutional parallel isosurface extraction based on tetrahedral bisection. In *Proceedings of the Volume Visualization '99 workshop*, 1999.

13. J. Kačur and K. Mikula. Solution of nonlinear diffusion appearing in image smoothing ansd edge detection. *Appl. Numer. Math.*, 17 (1):47–59, 1995.

14. Kawohl, B. and Kutev, N. Maximum and comparison principle for one-dimensional anisotropic diffusion. *Math. Ann.*, 311 (1):107–123, 1998.

15. N. Max and B. Becker. Flow visualization using moving textures. In *Proceedings of the ICASE/LaRC Symposium on Time Varying Data, NASA Conference Publication 3321*, pages 77–87, 1996.

16. M. Ohlberger and M. Rumpf. Adaptive projection operators in multiresolutional scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 4 (4), 1998.

17. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *IEEE Computer Society Workshop on Computer Vision*, 1987.

18. T. Preußer and M. Rumpf. Anisotropic nonlinear diffusion in flow visualization. In *Proceedings Visualization '99, to appear*, 1999.

19. M. Rumpf. Recent numerical methods – a challenge for efficient visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15:43–58, 1999.

20. C. Schnörr. A study of a convex variational approach for image segmentation and feature extraction. *J. Math. Imaging and Vision*, 8:271–292, 1998.

21. H.-W. Shen and D. L. Kao. Uflic: A line integral convolution algorithm for visualizing unsteady flows. In *Proceedings Visualization '97*, pages 317–322, 1997.

22. J. J. van Wijk. Spot noise-texture synthesis for data visualization. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 309–318, July 1991.

23. R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *J. Comput. Appl. Math.*, 50:67–83, 1994.

24. J. Weickert. *Anisotropic diffusion in image processing.* Teubner, 1998.

25. J. Weickert, B. M. ter Haar Romeny, and Viergever. Efficient and reliable schemes for nonlinear diffusion. *IEEE Trans. Image Proc.*, 7:398–410, 1998.

26. Weickert, J. Foundations and applications of nonlinear anisotropic diffusion filtering. *Z. Angew. Math. Mech.*, 76:283–286, 1996.