Comput Visual Sci 1:53–61 (1997)

Computing and
Visualization in Science

A coarsening finite element strategy in image selective smoothing

Eberhard Bänsch[1], **Karol Mikula**[2]

[1] Institut für Angewandte Mathematik, Universität Freiburg, Herrmann-Herder-Strasse 10, D-79104 Freiburg, Germany
[2] Department of Mathematics, Slovak Technical University, Radlinskeho 11, SK-81368 Bratislava, Slovakia

Received: 2 September 1996 / Accepted: 24 February 1997
Communicated by W. Jäger

Abstract. We introduce the coarsening of finite element computational grid strategy into the method proposed by Kačur and Mikula in [11] for the numerical solution of the Perona–Malik model modified in the sense of Catté, Lions, Morel and Coll. It improves the efficiency of the method, while the solution tends to be more flat during the selective smoothing process applied to the image. The numerical approximation consists of Rothe's method in time and linear finite elements on unstructured adaptively coarsened meshes in space.

0 Introduction

An image can be modelled as a real valued function $u_0(x)$, representing the values of the greylevel intensity, defined in some rectangular subdomain $\Omega \subset I\!R^d$, in practice $d = 2$ or 3. In the last decade, models based on acting of evolutionary PDE's to $u_0(x)$ have been suggested for special purposes in image processing and computer vision such as selective image smoothing, enhancement, restoration, segmentation, edge detection or shape analysis. Such view is called "image multiscale analysis", see [1, 3, 13, 20]. It associates with $u(0,x) = u_0(x)$ a "sequence" of simplified images $u(t,x)$ depending on the abstract parameter $t > 0$, the scale. Then the scaling of the initial image is understood as a "running" through the sequence of physical or biological filters, what causes the extracting of information which may be relevant e.g. for human perception. Under some reasonable assumptions (see [1]), the family of nonlinear operators representing the filtering is the solution $u(t,x)$ of a nonlinear partial differential equation of degenerate parabolic type. Famous examples are nonlinear diffusion of Perona–Malik type ([9, 11, 12, 17, 18]) and generalized mean curvature motions ([1, 2, 3, 13, 15]). This approach for image processing problems yields the possibility to apply robust numerical techniques to image analysis.

In this paper, we are dealing with the following problem, suggested by Catté, Lions, Morel and Coll:

$$\partial_t u - \nabla(g(|\nabla G_\sigma * u|)\nabla u) = f(u_0 - u) \quad \text{in } Q_T \equiv I \times \Omega, \tag{0.1}$$

$$\partial_\nu u = 0 \quad \text{on } I \times \partial\Omega, \tag{0.2}$$

$$u(0,\cdot) = u_0 \quad \text{in } \Omega, \tag{0.3}$$

where $\Omega \subset I\!R^d$ is a bounded domain with Lipschitz continuous boundary, $I = (0,T)$ is a scaling (time) interval, and

$$g \text{ is a Lipschitz continuous function, } g(0) = 1 \text{ and } 0 < g(s) \to 0 \text{ for } s \to \infty, \tag{0.4}$$

$$G_\sigma \in C^\infty(I\!R^d) \text{ is a smoothing kernel with } \int_{I\!R^d} G_\sigma(x)dx = 1 \text{ and } G_\sigma(x) \to \delta_x \text{ for } \sigma \to 0, \ \delta_x \text{ the Dirac measure at point } x, \tag{0.5}$$

$$f \text{ is a Lipschitz continuous, nondecreasing function, } f(0) = 0, \tag{0.6}$$

$$u_0 \in L^2(\Omega). \tag{0.7}$$

(0.1)–(0.3) represent a slight modification of the original Perona–Malik model, where ∇u stands in the place of $\nabla G_\sigma * u$ in (0.1). The Perona–Malik model, called the "anisotropic diffusion" in image smoothing and edge detection analysis, selectively diffuses an image in regions, where the signal is of constant mean in contrast to those regions where the signal changes its tendency. Here the "time" t is an abstract time–scale monitoring the filter process, i.e. bigger values of t meaning stronger filtering. The diffusion process is governed by the shape of the function g and its dependence on ∇u which is in some sense an edge indicator, see [18].

The analysis of the 1d–case (see e.g. [9]) shows that if the function $g(s)s$ is decreasing, the Perona–Malik equation can behave locally like the backward heat equation, which is an ill–posed problem. So for g's used in practice, e.g. $g(s) = 1/(1+s^2)$ or $g(s) = e^{-s}$, both existence and uniqueness of a solution cannot be obtained.

One way how to preveal that "mathematical" disadvantage has been proposed by Catté, Lions, Morel and Coll in [9]. They introduce the convolution with the Gaussian kernel G_σ into the decision process for the value of the diffusion coefficient. This slight modification (for σ small, the models are close and in a sense $\nabla G_\sigma * u \to \nabla u$ for $\sigma \to 0$) allowed them to prove the existence and uniqueness of the solution

for the "modified" model. Moreover, the usage of the "Gaussian gradient", which is used also for other tasks in image processing [20], makes the process more stable and transparent in the presence of noise. It has made explicit some "implicit presmoothing" included in numerical realizations of the Perona–Malik equation, too.

By means of f on the right–hand side of (0.1), the solution u is forced to be close to u_0 and to tend to some nontrivial equilibrium, which can weaken the influence of the stopping time T. In [17], $f(s) \equiv s$ is proposed. In our experiments we choose $f(s) \equiv 0$.

For the numerical solution of (0.1)–(0.3) we use the technique suggested and analysed in [11]. It is based on the Rothe's type approximation in time, using an implicit Euler scheme and a semi–implicit formulation of the non–linearity. This time discretization reduces the equation (0.1) to a sequence of linear elliptic problems

$$\frac{u^i - u^{i-1}}{\tau} - \nabla(g(|\nabla G_\sigma * u^{i-1}|)\nabla u^i) = f(u_0 - u^{i-1}) \quad \text{in } \Omega, \tag{0.8}$$

$$\partial_\nu u^i = 0 \quad \text{on } \partial\Omega, \tag{0.9}$$

$$u^0 = u_0 \tag{0.10}$$

for $i = 1, ..., n$, for the unknown functions u^i on each $t_i = i\tau, \tau = \frac{T}{n}, n \in I\!N$.

(0.8)–(0.9) are understood in the variational sense and on each discrete time–scale step t_i they are projected to a finite dimensional subspace (e.g. by the finite element method) and solved by the methods of numerical linear algebra. The convergence of such approximations to the unique weak solution of (0.1)–(0.3) was proved in [11].

In this paper we improve the efficiency of the method by the use of adaptively chosen grids on each time–scale step. Usually, for time dependent problems a modification consisting of refinement and coarsening steps is necessary to adjust the grid on a certain time step, see [7]. However, for our problem it is sufficient to coarsen the initial grid successively. The whole information about what is known of the image is contained in the initial grid. On the other hand there is no spatial *movement* of edges etc., hence no refinements of the grids are needed.

The basis for the coarsening is described for instance in [6, 10], see also [4]. This access may reduce the computational effort considerably, as the solution tends to be more flat with the increasing time–scale (see the experiments in Sect. 4). The coarsening of the computational grids rapidly reduces the number of unknowns in the linear systems to be solved at the discrete time–scale steps of the method. On the other hand, it leads to a finite element method on unstructured grids. Due to that fact, we must realize the convolution involved in the model in a special way.

Let us note that by the term $\nabla G_\sigma * u$ in (0.1) we mean $\int_{I\!R^d} \nabla_x G_\sigma(x - \xi)\tilde{u}(\xi)d\xi$, where $\tilde{u}$ is an extension of u, for which we assume

$$\|\tilde{u}\|_{W^{1,2}(I\!R^d)} \le C\|u\|_{W^{1,2}(\Omega)}. \tag{0.11}$$

Then the convergence results of [11] for the approximation scheme (0.8)–(0.9) hold true for kernels G_σ with

$$\int_{I\!R^d} |\nabla G_\sigma|^2 dx \le C_\sigma. \tag{0.12}$$

We use the fact that if

$$G_\sigma(x) = \frac{1}{(2\sqrt{\pi\sigma})^d} e^{-\frac{|x|^2}{4\sigma}} \tag{0.13}$$

the fundamental solution of the heat equation, for which (0.12) is clearly satisfied, then the term $\nabla G_\sigma * u^{i-1}$ in (0.8) is nothing else than the gradient of the solution at time σ of the heat equation in $I\!R^d$ with u^{i-1} as initial datum. Thus we replace the convolution by solving the heat equation for one time step with length σ on the same computational grid.

After making some comments on the notations in Sect. 1 we introduce the discrete equations for problem (0.1)–(0.3) in Sect. 2. In Sect. 3 the coarsening finite element strategy is explained. We conclude the presentation by a discussion on some numerical examples in Sect. 4.

In this presentation we restrict ourselves to the case $d = 2$, although our method can be applied also to the three dimensional case, see [6, 7]. In the three dimensional case the resulting discretized systems to solve are usually huge in image processing and therefore some additional attention has to be given to the problem of efficient linear solvers, such as multilevel preconditioning or multigrid techniques.

1 Notation

Let $\Omega \subseteq I\!R^2$ be a bounded domain. Denote by (u, v) the scalar product for $u, v \in L^2(\Omega)$. $W^{1,2}(\Omega)$ is the Sobolev space of L^2–functions with square integrable weak derivatives.

For $\Psi \in W^{-1,2} := \left(W^{1,2}\right)^*$ and $u \in W^{1,2}$ we denote by $\langle \Psi, v\rangle := \Psi(v)$ the corresponding dual pairing.

For $I \subset I\!R$ an interval and X a Banach (Hilbert) space we denote by $L^2(I, X)$ the Banach (Hilbert) space of measurable, square integrable functions from I to X.

A *triangulation* $\mathcal{T}$ of Ω is a set of (non–degenerate) triangles with $\bigcup_{T\in\mathcal{T}} T = \bar{\Omega}$.

A triangulation $\mathcal{T}$ is called *conforming* if the intersection of two non–disjoint, non–identical triangles consists either of a common vertex or a common edge.

$T \in \mathcal{T}$ is said to have a *non–conforming node*, if there is a vertex P of the triangulation which is not a vertex of T but $P \in T$.

A triangulation $\mathcal{T}$ has the property of *shape regularity* if $\max_{T\in\mathcal{T}} \{h_T/\rho(T)\} \le C$, with a constant C which is not too big. Here, h_T := diam(T) and $\rho(T) := \max\{r \mid B_r \subset T\}$ denotes the radius of the largest ball inscribed T.

2 Discretization scheme

Let (0.4)–(0.7), (0.11)–(0.12) be fulfilled. We are looking for a weak solution u of (0.1)–(0.3), i.e. for $u \in L^2(I, V)$ with $\partial_t u \in L^2(I, V^*)$, $u(0) = u_0$ (in the $L^2(\Omega)$ sense) such that the identity

$$\langle \partial_t u, v\rangle + (g(|\nabla G_\sigma * u|)\nabla u, \nabla v) = (f(u_0 - u), v) \tag{2.1}$$

holds for all $v \in V$ and for a.e. $t \in I$, where $V = W^{1,2}(\Omega)$ and $I = (0,T)$ for some $T > 0$.

To discretize in time, we replace (2.1) by the following problem (see also [11]): Let τ, σ be given numbers (usually $\sigma < \tau$). For $i = 1, \dots n$ we are looking for $u^i \in V$ approximating the image intensity function at the discrete time–scale point t_i, $i\tau$, $u^i \approx u(i\tau, \cdot)$, such that

$$(u^i, v) + \tau(g(|\nabla u^c|)\nabla u^i, \nabla v) = (u^{i-1} + \tau f(u_0 - u^{i-1}), v) \quad \forall v \in V \tag{2.2}$$

where $u^c \in V$ replaces the convolution $G_\sigma * u^{i-1}$ and is solution of the problem

$$(u^c, v) + \sigma(\nabla u^c, \nabla v) = (u^{i-1}, v), \quad \forall v \in V. \tag{2.3}$$

From the Lax–Milgram theorem it is clear that the solutions of (2.2)–(2.3) exist for all $i = 1, \dots n$.

We approximate (2.2)–(2.3) by the finite element method. For that let $\mathscr{T}$ be a conforming triangulation of Ω, which has the property of shape regularity. How to generate and coarsen such a triangulation is the topic of Sect. 3.

Given a triangulation $\mathscr{T}$, we define the set $V_h \subseteq V$ of piecewise linear finite elements, $V_h = V_h(\mathscr{T}) := \{v \in C^0(\Omega) | v_{|T} \text{ is linear for all } T \in \mathscr{T}\}$.

To derive a fully discrete scheme we choose triangulations $\mathscr{T}_i$ for each time–scale point $i = 1, \dots n$. We are looking for a function $u_h^i \in V_h(\mathscr{T}_i)$ fulfilling

$$(u_h^i, v_h) + \tau(g(|\nabla u_h^c|)\nabla u_h^i, \nabla v_h) = (u_h^{i-1} + \tau f(u_0 - u_h^{i-1}), v_h) \tag{2.4}$$

for all $v_h \in V_h(\mathscr{T}_i)$ where $u_h^c \in V_h(\mathscr{T}_i)$ is the solution of

$$(u_h^c, v_h) + \sigma(\nabla u_h^c, \nabla v_h) = (u_h^{i-1}, v_h) \quad \forall v_h \in V_h(\mathscr{T}_i). \tag{2.5}$$

By introducing the usual Lagrangian bases of hat functions $\varphi_j \in V_h$, determined by

$$\varphi_j(x_k) = \delta_{j,k}$$

for all vertices x_k, $k = 1, \dots, N$ of $\mathscr{T}$, N the number of vertices, u_h^i, u_h^c are given by $u_h^i = \sum_{k=1}^N u_k^i \varphi_k = \sum_{k=1}^N u^i(x_k)\varphi_k$ and $u_h^c = \sum_{k=1}^N u_k^c \varphi_k = \sum_{k=1}^N u^c(x_k)\varphi_k$.

Then (2.4–2.5) may be written in the form

$$\sum_{k=1}^N \Big\{(\varphi_k, \varphi_j) + \tau(g(|\nabla u_h^c|)\nabla\varphi_k, \nabla\varphi_j)\Big\} u_k^i = (u_h^{i-1} + \tau f(u_0 - u_h^{i-1}), \varphi_j)$$

and

$$\sum_{k=1}^N \Big\{(\varphi_k, \varphi_j) + \sigma(\nabla\varphi_k, \nabla\varphi_j)\Big\} u_k^c = (u_h^{i-1}, \varphi_j)$$

for all $j = 1, \dots N$.

Thus solving (2.4–2.5) means inverting two linear systems with matrices

$$\mathbf{M} + \tau\mathbf{A}(g(|\nabla u_h^c|)),$$

$$\mathbf{M} + \sigma\mathbf{A}(\mathbb{1}),$$

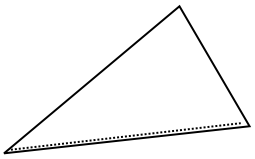

Fig. 1. Triangle with refinement edge

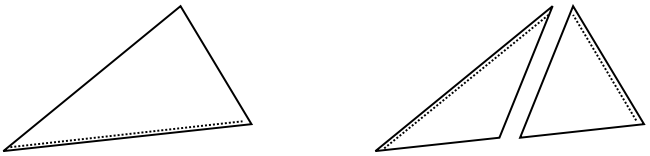

Fig. 2. Bisection of a single triangle

with $\mathbf{M}_{j,k} = (\varphi_k, \varphi_j)$ the mass matrix and $\mathbf{A}(w)_{j,k} = (w\nabla\varphi_k, \nabla\varphi_j)$ the stiffness matrix. Since $\mathbf{M}$, $\mathbf{A}(g(|\nabla u_h^c|))$ and $\mathbf{A}(\mathbb{1})$ are symmetric and positive definite it is clear that the discrete solutions exist and may be computed by a preconditioned conjugate gradient method quite efficiently. In our computations we used a simple diagonal scaling, which is not an optimal method. More appropriately one should use multilevel preconditioning (see for instance [8, 21]) or multigrid techniques, see for example [14].

3 Coarsening strategy

In this section, we describe how to generate the initial triangulation on which the initial image is given and how to locally coarsen this triangulation according to a certain coarsening criterion.

Usually, the initial image is given as a set of discrete grey (or RGB) values. That means that the initial triangulation $\mathscr{T}_0$ is a uniform grid where u_0 is given on grid cells or grid points associated with the corresponding grey values. We generate this initial triangulation by refining a coarse grid, the *macro triangulation*. We choose the so called *bisection method*, which allows for coarsening quite easily and is explained for instance in [19] and also in [5, 16]. Such refinement strategies are commonly used in connection with multi–grid or multi–level methods, see i.g. [4].

The bisection method may be introduced in the following way:

3.1 Refinement by bisection

Denote by $\mathscr{T}^0$ the macro triangulation.

Before starting the refinement process one edge of every triangle of the macro–triangulation is marked (see Fig. 1). This edge is called *refinement edge*. To divide a single triangle, it is cut through the midpoint of the refinement edge and the vertex opposite to the refinement edge. The new refinement edges are choosen opposite to the new vertex (see Fig. 2).

This very simple process describes how to refine a single triangle. Now starting with a macro triangulation $\mathscr{T}^0$ we want to generate a sequence of successively finer triangulations $\{\mathscr{T}^k\}_k$ until the desired fineness is reached for a certain $\mathscr{T}^{k_0}$ and we may then define $\mathscr{T}_0 := \mathscr{T}^{k_0}$.

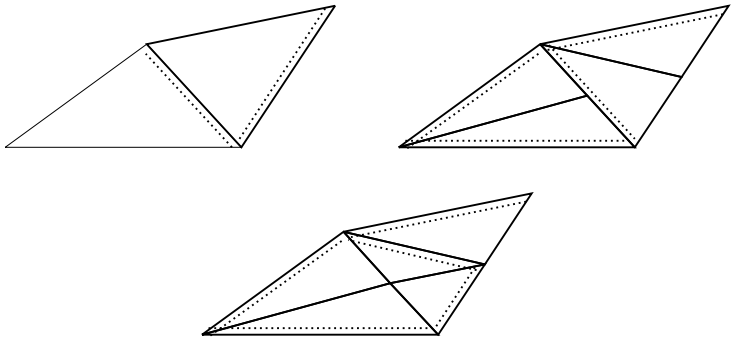

Fig. 3. $\mathscr{T}^0, \hat{\mathscr{T}}^0$ and $\mathscr{T}^1$

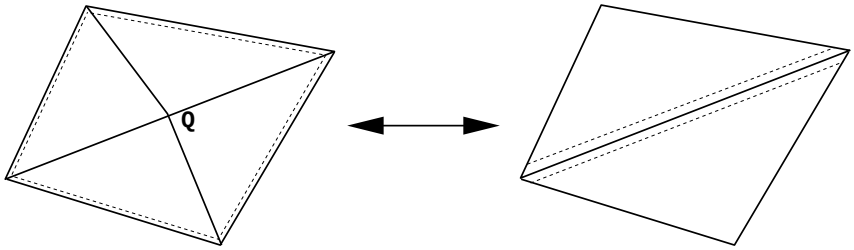

Fig. 4. Resolvable patch M with coarsening node Q and coarsened patch

For that we define the following (global) algorithm:

Refinement algorithm: Start with $\mathscr{T}^0$. Then for every k let Σ^+ be the set of those triangles, which have to be divided ($\Sigma^+ = \mathscr{T}^k$ in case of uniform refinement). Then one bisection step is given by:

while $\Sigma^+ \neq \emptyset$ **do**

bisect all $T \in \Sigma^+$ as described above, obtain the intermediate triangulation $\hat{\mathscr{T}}^k$ (possibly non-conforming) let now Σ^+ be the set of those triangles with a non-conforming node.

endwhile

$\mathscr{T}^{k+1} := \hat{\mathscr{T}}^k$

Figure 3 shows an example $\mathscr{T}^0 \longrightarrow \mathscr{T}^1$ with initial $\Sigma^+ := \mathscr{T}^0$.

The crucial point is the question whether the algorithm terminates and generates a sequence of triangulations $\mathscr{T}^k$ which has the property of shape regularity with a constant independent of k. This question is answered by the following Proposition 1.

Proposition 1 *The above algorithm stops in a finite number of steps, $\mathscr{T}^{k+1}$ is conforming and the sequence $\{\mathscr{T}^k\}_k$ is shape regular independent of k.*

Proof: The proof is very simple and can be found for instance in [5].

Remark : Note that there is no compatibility condition for the initial position of the refinement edges of neighbouring triangles. A good choice would be the longest edge of each triangle.

3.2 Local coarsening

We choose the bisection method to generate the starting triangulation $\mathscr{T}_0$ because a triangulation which was derived by a successive application of bisection steps can be derefined very easily.

We make the following definitions:

Definition:

i.) A simplex $T \in \mathscr{T}$ has level l if T was obtained after l refinement steps.

ii.) A simplex T is said to have *locally finest level* if the levels of all neighbours are less than or equal to the level of T.

iii.) Let $T \in \mathscr{T}$ and let T' be the father of T. A vertex P of T which was inserted while bisecting T' is called the *coarsening node* of T.

iv.) Let K be an edge of the triangulation $\mathscr{T}$ and K' the "father"–edge of K with midpoint Q. Set $M := \{T \in \mathscr{T} | T \cap K' \neq \emptyset\}$. If Q is the coarsening node for all $T \in M$ then M is called a *resolvable patch*.

Figure 4 shows a resolvable patch and the coarsened patch.

If M is a resolvable patch, then all $T \in M$ can be coarsened without interfering with $T' \in \mathscr{T}$ outside of M. Therefore resolvable patches are the configurations which we allow to be coarsened. This guarantees that the coarsening process stays local.

We may write the coarsening algorithm in the following form:

Coarsening algorithm: Let $\mathscr{T}_i$ be a triangulation obtained by refinement and coarsening steps. Let $\Sigma^- \subset \mathscr{T}_i$ be the set of triangles to be derefined. Then one coarsening step consists of:

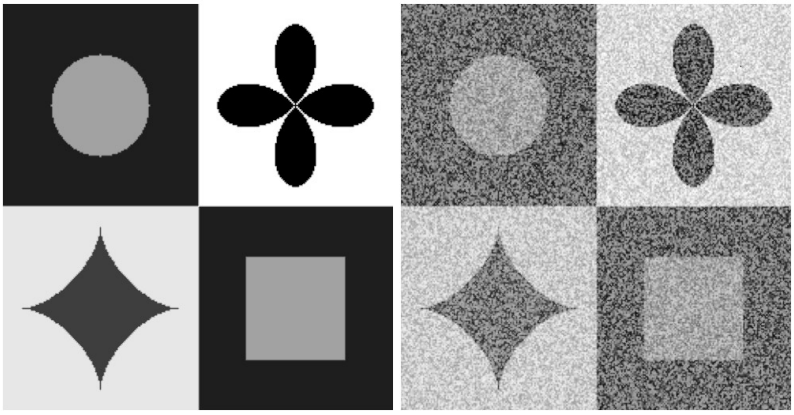

Fig. 5. Original and noisy initial image

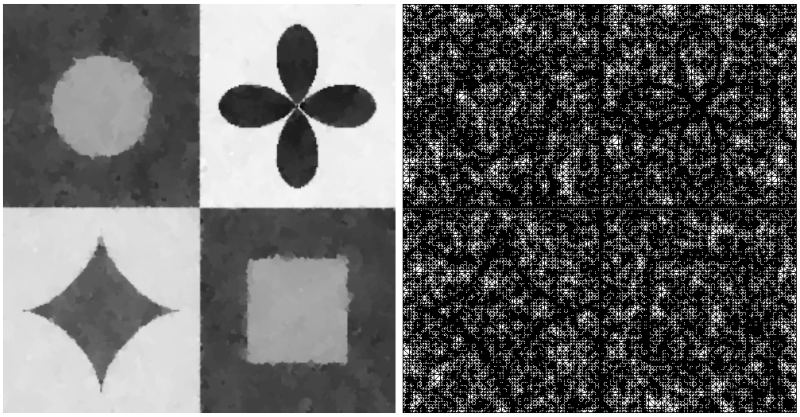

Fig. 6. Solution and grid after 4 time–scale steps

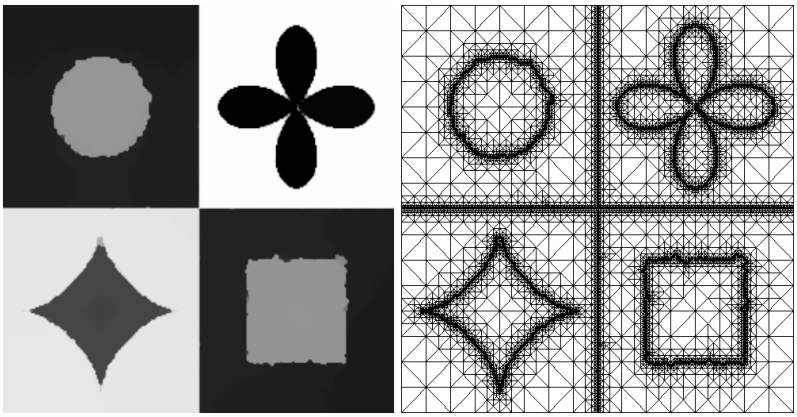

Fig. 7. Solution and grid after 8 time–scale steps

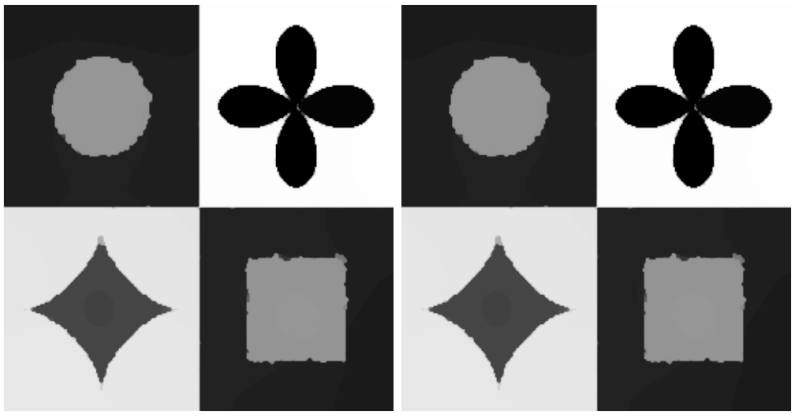

Fig. 8. Image after 8 time–scale steps, adaptive calculation (*left*), calculation on a uniform grid (*right*)

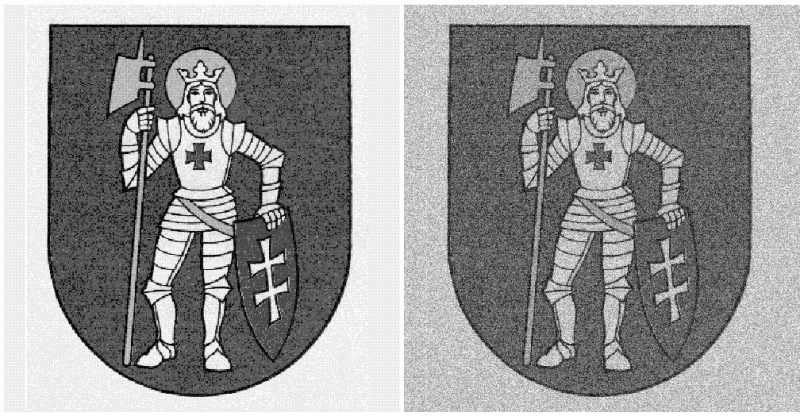

Fig. 9. Original and noisy initial image

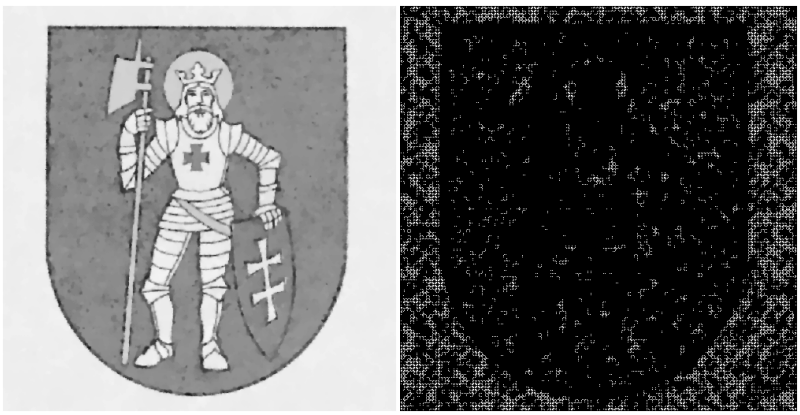

Fig. 10. Solution and grid after 3 time–scale steps

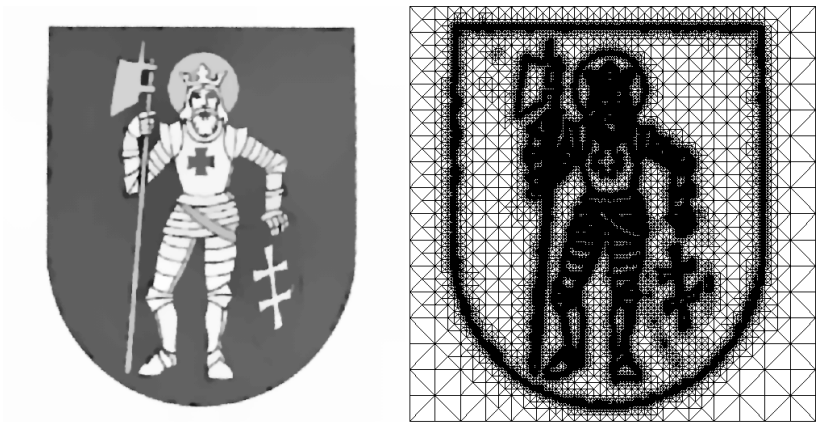

Fig. 11. Solution and grid after 6 time–scale steps

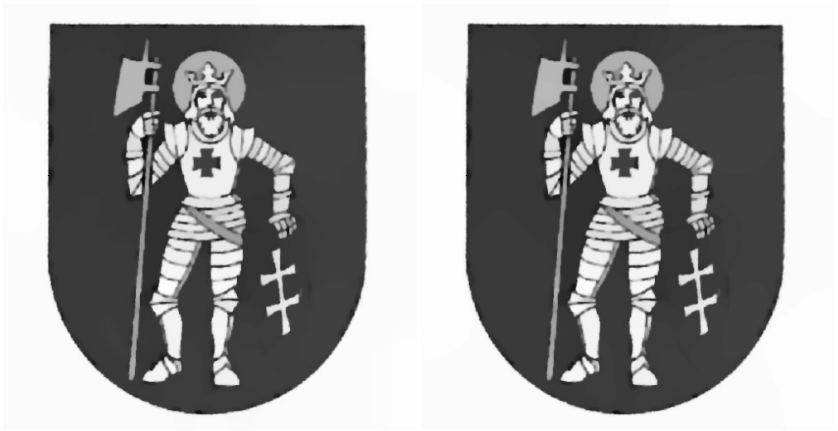

Fig. 12. Image after 6 time–scale steps, adaptive calculation (*left*), calculation on a uniform grid (*right*)

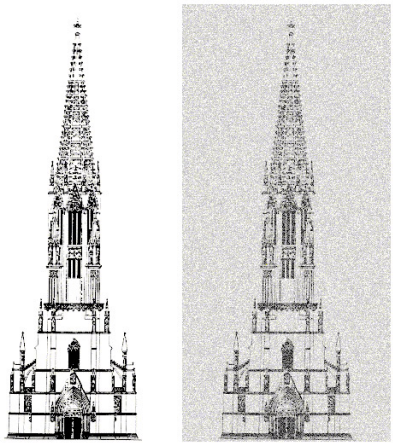

Fig. 13. Original and noisy initial image

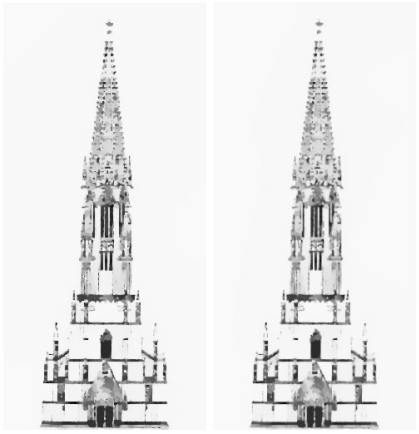

Fig. 15. Image after 9 time–scale steps, adaptive calculation (*left*), calculation on a uniform grid (*right*)

for each $T \in \Sigma^-$ **do**
  if T belongs to a resolvable patch M
    if $T' \in \Sigma^-$ for all $T' \in M$
      derefine M, see Fig. 4
    endif
  endif
enddo

Since we only derefine resolvable patches the question arises whether there are "enough" resolvable patches in an arbitrary triangulation. In [10] it is shown that under reasonable assumptions on the distribution of the refinement edges for $\mathscr{T}^0$ triangles of locally finest levels always belong to resolvable patches.

In particular we have the following result (see [10]):

Proposition 2 *Let the refinement edges in $\mathscr{T}^0$ be choosen as the longest edges in each triangle. Then, if $T \in \mathscr{T}^k$ for $k \in \mathbb{N}$ has locally finest level, T belongs to a resolvable patch.*

By Proposition 2 it is clear that by successive application of the coarsening algorithm it is always possible to go back to the macro triangulation.

3.3 Coarsening criterion and adaptive method

As the local behaviour of ∇u determines the evolution process and is an indicator for edges, the coarsening criterion is based on this value. More precisely, let $\epsilon > 0$ be a given tolerance. For i a time–scale step and u_h^i the corresponding numerical solution on the grid $\mathscr{T}_i$ we allow all triangles $T \in \mathscr{T}_i$ to be coarsened, if

$$h_T |\nabla u_h^i| \le \epsilon \quad \text{on } T. \tag{3.1}$$

Note that since u_h^i is piecewise linear, ∇u_h^i is constant on each triangle.

Thus we have the following adaptive scheme to approximate (0.1–0.3):

Let $\mathscr{T}_0, u_0$ be given.

for $i = 1, 2, \ldots n$ **do**
  set up the matrix $\mathbf{M} + \sigma\mathbf{A}(\mathbb{1})$
  compute $u_h^c \in V_h(\mathscr{T}_i)$ solving (2.5)
  set up the matrix $\mathbf{M} + \tau\mathbf{A}(g(|\nabla u_h^c|))$
  compute $u_h^i \in V_h(\mathscr{T}_i)$ solving (2.4)
  set $\Sigma^- := \{T \in \mathscr{T}_i \mid h_T |\nabla u_h^i| \le \epsilon \text{ on } T\}$
  derefine $\mathscr{T}_i$ according to Σ^- to obtain $\mathscr{T}_{i+1}$
enddo

4 Discussion on numerical experiments

In this section, we present some examples with an initial image u_0 which is derived from an original image $\hat{u}_0$ by adding some noise:

$$u_0(x) = (1 + \delta\psi(x))\hat{u}_0(x),$$

with $\delta \in [0, 1]$ and ψ a (pseudo) random function with values in $[0, 1]$.

Throughout this section we use $g(s) = \dfrac{1}{1 + s^2}$ and $f \equiv 0$.

The **first example** consists of 4 simple patterns which are perturbed by 60% noise, i.e. $\delta = 0.6$. Figure 5 shows the original image and the noisy one, which are given on a 257×257 mesh that is, the initial grid $\mathscr{T}_0$ consists of 66049 nodes. The following parameters were used for the computation:

$$\tau = 0.01, \quad \sigma = 0.001, \quad \epsilon = 0.07$$

for the coarsening criterion in 3.1.

Figures 6–7 show the solution after 4 and 8 time–scale steps together with the corresponding grids. A comparison of the solution obtained by the adaptive procedures and by a calculation on the fixed initial grid after 8 time–scale steps is given in Fig. 8. The corresponding CPU–times on a SGI workstation are reported in Table 1. Table 2 shows the decrease of the number of unknowns during the time evolution.

The **second example** is less academic. The initial picture is given on a 513×513 mesh (see Fig. 9).

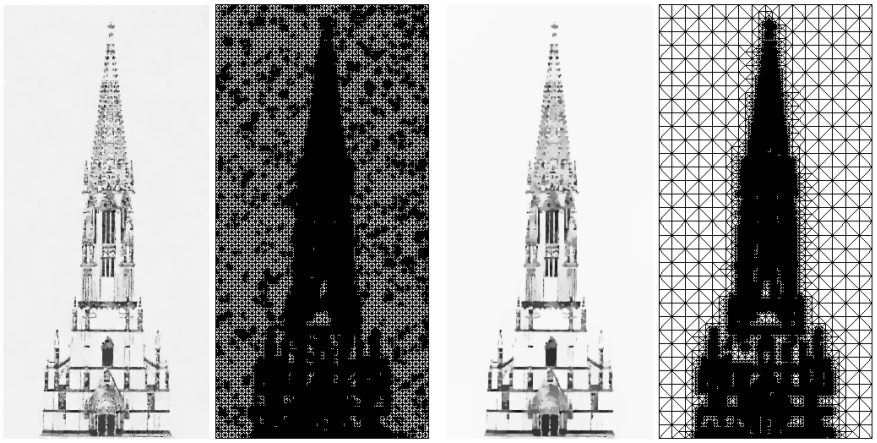

Fig. 14. Solutions and grids after 5 and 9 time–scale steps

Table 2. Decrease of unknowns, Example 1

time step	1.	2.	3.	4.	5.	6.	7.	8.
♯ unknowns	66049	65539	61490	41939	12225	8943	8002	7797

Table 1. Comparison of CPU time for adaptive/fixed mesh strategies, Example 1

SGI RS 4400	
adaptive	152 sec
fixed mesh	418 sec

Table 3. Comparison of CPU time for adaptive/fixed mesh strategy, Example 2

SGI RS 4400	
adaptive	570 sec
fixed mesh	1875 sec

We used the parameters

$$\tau = 0.004, \quad \sigma = 0.002, \quad \epsilon = 0.07, \quad \delta = 0.6.$$

The results are shown in Figs. 10 and 11 after 3 and 6 time–scale steps respectively. Clearly some fine details of the image are lost, however, the essential information of the image is kept during the time evolution.

Figure 12 shows a comparison of the results obtained by the adaptive strategy and a calculation on a fixed mesh. Again the results are quite similar. The reduction of the computational effort is reported in Tables 3–4.

The **third example** is quite a hard numerical test problem due to a lot of fine details. As before, Fig. 13 shows the original and noisy initial image, Fig. 14 the result of the computation after 5 resp. 9 time–scale steps and Fig. 15 the comparison with the numerical solution on a fixed mesh.

This computation is performed with

$$\tau = 0.004, \quad \sigma = 0.002, \quad \epsilon = 0.085, \quad \delta = 0.6.$$

Table 4. Decrease of unknowns, Example 2

time step	1.	2.	3.	4.	5.	6.
♯ unknowns	263169	250919	134729	69152	51891	44439

Table 5. Comparison of CPU time for adaptive/fixed mesh strategy, Example 3

SGI RS 4400	
adaptive	312 sec
fixed mesh	875 sec

5 Conclusions

We introduced a coarsening finite element strategy for the modified Perona–Malik model in image processing. The techniques stems from a general adaptive finite element approach for time dependent problems. This coarsening strategy is capable of improving the efficiency of image smoothing by nonlinear diffusion considerably. It was shown that the number of unknowns decreases rapidly even after only a few smoothing steps.

References

1. L. Alvarez, F. Guichard, P.L. Lions, J.M. Morel: Axioms and Fundamental Equations of Image Processing. Archive for Rat. Mech. Anal., **123**, 200–257 (1993)
2. L. Alvarez, P.L. Lions, J.M. Morel: Image selective smoothing and edge detection by nonlinear diffusion II. SIAM J. Numer. Anal., **29**, 845–866 (1992)
3. L. Alvarez, J.M. Morel: Formalization and computational aspects of image analysis. Acta Numerica, 1–59 (1994)
4. R.E. Bank: PLTMG: A software package for solving elliptic partial differential equations. Users Guide 6.0. SIAM, Philadelphia (1990)
5. E. Bänsch: Local Mesh Refinement in 2 and 3 Dimensions. IMPACT of Computing in Science and Engineering, **3**, 181–191 (1991)
6. E. Bänsch: Adaptive Finite–Element Techniques for Navier–Stokes Equations and other Transient Problems. In: Adaptive Finite and Boundary Elements, Computational Mechanics Publications und Elsevier, Amsterdam 1993
7. E. Bänsch: Mesh refinement in two and three dimensions. In: INRIA course “Calcul d’erreur a posteriori et adaptation de maillage”, Rocquencourt (France) 9/1995
8. F. Bornemann: An Adaptive Multilevel Approach to Parabolic Equations III. 2D Error Estimation and Mutlilevel Preconditioning. IMPACT of Computing in Science and Engineering **4**, 1–45 (1992)

Table 6. Decrease of unknowns, Example 3

time step	1.	2.	3.	4.	5.	6.	7.	8.	9.
♯ unknowns	131841	130516	121912	92539	45568	34040	29553	28524	27675

9. F. Catté, P.L. Lions, J.M. Morel, T. Coll: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **129**, 182–193 (1992)
10. D. Hempel: Local Mesh Adaptation in Two Space Dimensions. IMPACT of Computing in Science and Engineering, **5**, no. 4, 309–317 (1993)
11. J. Kačur, K. Mikula: Solution of nonlinear diffusion appearing in image smoothing and edge detection., Applied Numerical Mathematics, **17**, 47–59 (1995)
12. J. Kačur, K. Mikula: Slow and fast diffusion effects in image processing – Approximation schemes and numerical experiments. Preprint IWR 96–26, University of Heidelberg (1996)
13. P.L. Lions: Axiomatic derivation of image processing models. M^3AS, **4**, 467–475 (1994)
14. J.M. Laferte, P. Perez, F. Heitz: Global non linear multigrid optimization for image analysis tasks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, **5**, Los Alamitos, CA, IEEE 533–536 (1994)
15. K. Mikula: Solution of nonlinear flow governed by curvature of plane convex curves. Preprint IWR 96–08, University of Heidelberg (1996)
16. W.F. Mitchell: A Comparison of Adaptive Refinement Techniques for Elliptic Problems. ACM Trans. on Math. Software **15**, no. 4, 326–347 (December 1989)
17. K.N. Nordström: Biased anisotropic diffusion – A unified approach to edge detection. Preprint, Department of Electrical Engeneering and Computer Sciences, University of California, Berkeley (1989)
18. P. Perona, J. Malik: Scale space and edge detection using anisotropic diffusion. In Proc. IEEE Computer Society Workshop on Computer Vision (1987)
19. M.C. Rivara: Algorithms for refining triangular grids for adaptive and multigrid techniques. Int. J. Numer. Meth. Eng. **20**, 745–756 (1984)
20. B.M. ter Haar Romeny (Ed.): Geometry driven diffusion in computer vision. Dordrecht: Kluwer 1994
21. H. Yeserentant: Two preconditioners based on the multi–level splitting of finite element spaces. Numer. Math. **58**, 163–184 (1990)