

# An Adaptive Finite Volume Scheme for Solving Nonlinear Diffusion Equations in Image Processing

Zuzana Krivá and Karol Mikula

Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 813 68 Bratislava, Slovakia  
E-mail: mikula@vox.svf.stuba.sk

Received January 27, 2000; accepted November 16, 2001

We propose the coarsening strategy for the finite volume computational method given by K. Mikula and N. Ramarosy (*Numer. Math.* **89**, 2001, 561–590) for the numerical solution of the (modified in the sense of F. Catté *et al.* (*SIAM J. Numer. Anal.* **29**, 1992, 182–193)) Perona–Malik nonlinear image selective smoothing equation (called *anisotropic diffusion* in image processing). The adaptive approach is directly at hand because a solution tends to be flat in large subregions of the image, and thus it is not necessary to consider the same fine resolution of computations in the whole spatial domain. This access reduces computational effort, because the coarsening of the computational grid rapidly reduces the number of unknowns in the linear systems to be solved at discrete scale steps of the method. © 2002 Elsevier Science (USA)

**Key Words:** image processing; nonlinear partial differential equations; numerical solution; finite volume method; adaptivity; grid coarsening.

## 1. INTRODUCTION

An image can be modeled as a real function  $u_0(x)$  representing values of gray-level intensity, defined in some rectangular subdomain  $\Omega \subset \mathbb{R}^d$  (in practice  $d = 2$  or  $3$ ). Basic tasks of image processing and computer vision—such as selective image smoothing, enhancement, restoration, segmentation, edge detection, shape analysis, optic flow computations—have been recently modeled by the application of evolutionary PDE to  $u_0(x)$ . Such approach is known as *image multiscale analysis* [1, 3, 14], since the initial image  $u_0(x) = u(0, x)$  is associated with a sequence of images  $u(t, x)$ , depending on an abstract parameter  $t > 0$  called scale. The well-known examples are nonlinear diffusion equations of Perona–Malik type [24, and, e.g., 7, 10, 11, 27] and generalized mean curvature flow equations [2, 26, and, e.g., 15, 9, 16, 19, 20]. In this paper we are dealing with the Perona–Malik-type problem suggested by Catté *et al.* [7] in the form

$$\partial_t u - \nabla \cdot (g(|\nabla G_\sigma * u|)\nabla u) = f(u_0 - u) \quad \text{in } Q_T \equiv I \times \Omega, \quad (1)$$

$$\partial_\nu u = 0 \quad \text{on } I \times \partial\Omega, \quad (2)$$

$$u(0, \cdot) = u_0 \quad \text{in } \Omega, \quad (3)$$

where  $\Omega \subset \mathbb{R}^d$  is a rectangular domain,  $I = [0, T]$  is a scaling interval, and  $g(s)$  is a decreasing function,

$$g(0) = 1, 0 < g(s) \rightarrow 0 \quad \text{for } s \rightarrow \infty, \quad (4)$$

$$G_\sigma \in C^\infty(\mathbb{R}^d) \text{ is a smoothing kernel with } \int_{\mathbb{R}^d} G_\sigma(x) dx = 1 \quad (5)$$

$$\text{and } G_\sigma(x) \rightarrow \delta_x \quad \text{for } \sigma \rightarrow 0, \delta_x\text{-Dirac function at point } x, \\ f \text{ is a Lipschitz continuous, nondecreasing function, } f(0) = 0, \quad (6)$$

$$u_0 \in L^2(\Omega). \quad (7)$$

The diffusion process of (1) is governed by the shape of function  $g$  and by its dependence on  $\nabla u$ , which is in a sense an edge indicator. Using Eq. (1) with  $\nabla u$  instead of  $\nabla G_\sigma * u$  was suggested by Perona and Malik in [24]. Catté *et al.* in [7] slightly modified the equation (for  $\sigma$  small the models are close), making it possible to prove existence and uniqueness of a weak solution for the modified model, making the model less sensitive to noise, and keeping the practical advantages of the original formulation. Other approaches to Perona–Malik edge-enhancing diffusion are given, e.g., in [13, 21].

For the numerical solution of (1)–(3), we adjust a technique suggested and analyzed in [18]. It is based on semi-implicit discretization in scale and on the so-called finite volume method in space. Recently, the finite volume method has been widely used in computational sciences and engineering since it is based on physical principles as conservation laws, is local, and is easy implemented. Despite other methods (finite elements, finite differences) in the finite volume method, discrete approximations of a solution of partial differential equation are considered to be piecewise constant in control volumes (cells), which in image processing corresponds to the pixel structure of a discrete image. This makes analysis of convergence of the finite volume method to a weak solution of the continuous problem more difficult [18] but on the other hand, from a conceptual point of view such an approach seems to be *the most natural for image processing*. Semi-implicitness of the method means that nonlinearity of the equation is treated from the previous discrete scale step; i.e., the scheme is linear and leads to a solution of sparse linear systems in each discrete scale step of the algorithm. As the solution tends to be more flat with the increasing scale in large regions of the image, we can improve considerably the efficiency of the method using adaptivity, i.e., choosing nonuniform grids with much less finite volumes. Since the whole information about the image is contained in the initial grid and there is no spatial movement of the edges, no refinement is needed and we work just with grids, elements of which are obtained by the merging of pixels. This process is called *coarsening* in numerical methods for solving PDEs. In this paper, we present a coarsening strategy for rectangular grids and join such a strategy with the finite volume method for solving (1)–(3).

The adaptivity for the finite element method [10] in image processing applications has been suggested in [5] (and generalized to the 3D case in [6]). This adaptive finite element method has been based on triangular grids generated by bisection [4] which are successively coarsened during the diffusion process. The approach given in [5] has been modified for

bilinear finite elements on rectangular grids in 2D and 3D by Preusser and Rumpf in [25]. They also improves storage requirements of the method by procedural handling of adaptive grids and applied nonlinear anisotropic diffusion in 2D and 3D flow field visualization.

The rest of this paper is organized as follows. In Section 2 we present the idea of coarsening. Section 3 is devoted to the finite volume method on nonuniform grids based on coarsening. Section 4 describes application of the adaptive finite volume method to real and artificial images and gives some computational comparisons.

## 2. COARSENING STRATEGY BASED ON QUADTREES

In this section we describe how to generate adaptively coarsened grids used in discrete scale steps of the computational method. The initial image is given as a set of discrete gray values on pixels of the uniform grid. At the beginning and especially with increasing scale, we can merge cells using some coarsening criterion, and instead on the regular grid we can work on the irregular adaptive structure. For its construction we chose an approach based on quadtrees, where the adaptive grid is represented by the leaves of quadtree structure. However, instead of organizing the resulting structure into a tree (which is known as being inconvenient when access to neighbors is needed) we use a procedural approach and maintain the field of *indicators* (see also [25]), which enable us to find out easily whether a given cell or its neighbor can be merged. Traversing this structure we stop on a higher level of hierarchy (i.e., on a coarser grid) if the so-called *coarsening criterion* is fulfilled for the cell. We consider the following coarsening criterion: *The cells are merged if a difference in intensities is below a prescribed tolerance  $\varepsilon$ .*

After creating the structure by setting the indicator field we calculate diffusion coefficients by its recursive traversing. In such way we create a system of linear equations that is then solved using the iterative method with low memory requirements. In order to simplify creating the matrix of the linear system we require that the ratio of the sides of two neighboring squares be 1 : 1, 1 : 2, or 2 : 1. Later, such a structure is called *balanced*.

### 2.1. Creating the Adaptive Grid

Without loss of generality, let us have an image with  $2^n \times 2^n$  pixels. Then the indicator field has dimensions  $(2^n + 1) \times (2^n + 1)$  (see Fig. 1). Setting the values of the indicator

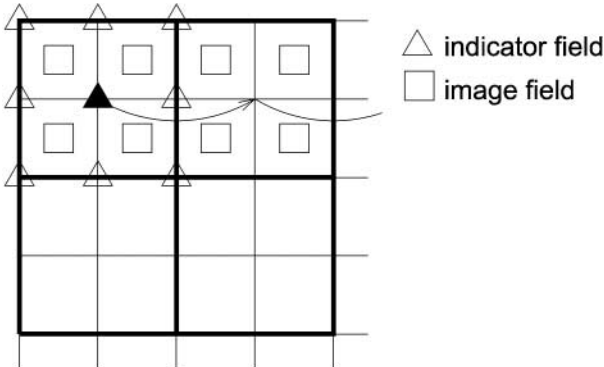
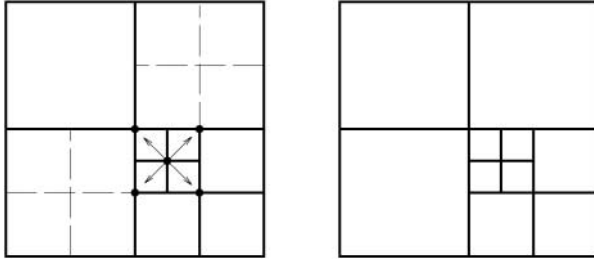


FIG. 1. Image field and indicator field together with image stencil and corresponding stencil in the indicator field.



**FIG. 2.** The role of the indicator field in balancing the structure. The grid on the right is not created in the coarsening process.

field at the beginning we start on the lowest level of the structure (i.e., on the pixel structure of the image). We try to merge the cells into  $2 \times 2$  cells according to the coarsening criterion. In order to perform merging we can use two stencils. One stencil ( $2 \times 2, 4 \times 4, \dots$ ) moves across the image and the other ( $3 \times 3, 5 \times 5, \dots$ ) stencil moves across the indicator field. Every  $2^j \times 2^j$  image stencil has the corresponding  $(2^j + 1) \times (2^j + 1)$  stencil in the indicator field. While neighboring image stencils are not overlapping, their corresponding indicator stencils share the side. With the help of stencils the values in the indicator field are set in such a way that:

1. they indicate whether the inspected cell on the higher level contains a quadruple suitable for merging. If intensities in the quadruple are not within the range of  $\varepsilon$ , then the position in the center of the indicator stencil is set to 1; otherwise it is left 0;
2. they help to maintain the structure as balanced. More precisely, after finding out that the inspected quadruple cannot be merged, not only the central node of the indicator stencil is set to 1, but this value is set also to the corners of the stencil.

Since two corners on the lower level become middle points on the side of the stencil, on the higher level we can control the merging of the neighboring cells and thus keep the structure balanced. For example, a structure as in the right half of Fig. 2 is not created in the coarsening process.

If four cells are merged into a larger one, then a new value, given by the average of old values, is stored in the left lower corner of the image stencil corresponding to the cell. This becomes the value representing the intensity of the merged pixels. Moreover, we remember maximal and minimal values for the merged quadruple in auxiliary fields (they can be free after creating the structure). Testing just the intensity differences in the coarsening criterion using the recursive process could cause a cumulation of errors, and in special cases the resulting difference could be greater than  $\varepsilon$ . Such a situation is depicted in Fig. 3. All  $2 \times 2$

$\bar{3}$	2	4	2	2	$\bar{2}$
	4	2	2	2	eps=2
	0	2	1	1	diff=4
$\bar{1}$	0	2	1	1	$\bar{1}$

**FIG. 3.** Possible cumulation of errors, which is suppressed in the coarsening process.

cells fulfill a coarsening criterion for  $\varepsilon = 2$ , and their new values are set to the average. If we test just these new values for processing the higher level, the coarsening criterion is fulfilled again. However, the intensity difference of the original pixels is 4, twice behind the tolerance. Thus, working on higher levels we calculate the minimum of all minimal values and the maximum of all maximal values for given cells and test their difference.

### 3. FINITE VOLUME SCHEME ON ADAPTIVE GRID

In this chapter we introduce the finite volume computational scheme for solving (1)–(3) on an adaptive grid obtained by means of the coarsening algorithm described in the previous section. The finite volume method given in [18] cannot be used straightforwardly in such a case. We will explain our strategy on how to adjust the scheme for adaptive nonuniform grids. First we recall the method from [18] and then present adjustments due to adaptivity.

Let  $\tau_h$  be a uniform mesh of  $\Omega$  with cells  $p$  of measure  $m(p)$  (we assume rectangular cells here). For every cell  $p$  we consider a set of neighbors  $N(p)$  consisting of all cells  $q \in \tau_h$  for which the common interface of  $p$  and  $q$ , denoted by  $e_{pq}$ , is of nonzero measure  $m(e_{pq})$ .

In the numerical scheme we will provide computations in the series of scale steps starting with  $\bar{u}_p^0$ ,  $p \in \tau_h$ , corresponding to the given intensities on the pixel structure of the initial discrete image. We assume that

$$\bar{u}_p^0 = \frac{1}{m(p)} \int_p u_0(x) dx, \quad p \in \tau_h; \quad (8)$$

i.e., the discrete image intensity represents the average cell value of the continuous intensity function  $u_0(x)$ . In the finite volume method, in every subsequent discrete scale step we get again piecewise constant approximations  $\bar{u}_p^n$ ,  $p \in \tau_h$ ,  $n = 1, 2, \dots$  of continuous solution (with possibly the same interpretation as cell averages of continuous solution). Convergence of such approximations to a weak solution of (1)–(3) provided the length of the scale step and the size of the pixel tend to zero is given in [18]. In [18], it is assumed that for every  $p$ , there exists a representative point  $x_p \in p$ , such that for every pair  $p, q$ ,  $q \in N(p)$ , the vector  $(x_q - x_p)/|x_q - x_p|$  is equal to the unit vector  $n_{pq}$ , which is normal to  $e_{pq}$  and oriented from  $p$  to  $q$ . (Let us note, that this assumption is not fulfilled for adaptive grids given by the coarsening algorithm.) In the simple case of a uniform grid, we can take  $x_p$  just as the center of the pixel. Then let  $x_{pq}$  be the point of  $e_{pq}$  intersecting the segment  $\overline{x_p x_q}$ . Then we define the coefficients

$$T_{pq} := \frac{m(e_{pq})}{|x_q - x_p|} \quad (9)$$

and

$$g_{pq}^{\sigma, n} := g(|\nabla G_\sigma * \tilde{u}(x_{pq})|), \quad (10)$$

where  $\tilde{u}$  is a periodic extension of the discrete image computed in the  $n$ th scale step. The finite volume scheme on the uniform grid is then written as follows:

Let  $0 = t_0 \leq t_1 \leq \dots \leq t_{N_{\max}} = T$  denote the scale discretization steps with  $t_n = t_{n-1} + k$ , where  $k$  is the discrete scale step. For  $n = 0, \dots, N_{\max} - 1$  we look for  $\bar{u}_p^{n+1}$ ,  $p \in \tau_h$ ,

satisfying the system of linear equations

$$\left( \frac{m(p)}{k} + \sum_{q \in N(p)} g_{pq}^{\sigma, n} T_{pq} \right) \bar{u}_p^{n+1} - \sum_{q \in N(p)} g_{pq}^{\sigma, n} T_{pq} \bar{u}_q^{n+1} = \frac{m(p)}{k} \bar{u}_p^n + f(\bar{u}_p^0 - \bar{u}_p^n) m(p). \quad (11)$$

The scheme (11) is linear *semi-implicit* in scale, since the scale derivative is replaced by the backward difference and the nonlinear terms of Eq. (1) are treated from the previous scale step while the linear terms are discretized on the current scale level. After such scale discretization, (11) is derived by integrating the corresponding elliptic equation over the cell, applying the divergence theorem and approximating the normal derivative on the boundary of the cell by  $(\bar{u}_q - \bar{u}_p)/|x_q - x_p|$ .

In the scheme (11) we must compute term (10), i.e., the vector

$$\nabla G_\sigma * \tilde{u}(x_{pq}) = \left( \frac{\partial(G_\sigma * \tilde{u})}{\partial x}(x_{pq}), \frac{\partial(G_\sigma * \tilde{u})}{\partial y}(x_{pq}) \right),$$

which is an input of the Perona–Malik function  $g$ . For that goal, we use the property of convolution

$$\frac{\partial(G_\sigma * \tilde{u})}{\partial x}(x_{pq}) = \left( \frac{\partial G_\sigma}{\partial x} * \tilde{u} \right)(x_{pq}).$$

Then one gets

$$\left( \frac{\partial G_\sigma}{\partial x} * \tilde{u} \right)(x_{pq}) = \int_{\mathbb{R}^d} \frac{\partial G_\sigma}{\partial x}(x_{pq} - s) \tilde{u}(s) ds = \sum_r \bar{u}_r^n \int_r \frac{\partial G_\sigma}{\partial x}(x_{pq} - s) ds \quad (12)$$

and thus

$$\nabla G_\sigma * \tilde{u}(x_{pq}) = \sum_r \bar{u}_r^n \int_r \nabla G_\sigma(x_{pq} - s) ds, \quad (13)$$

where the sum is restricted to control volumes  $r$  inside  $B_\sigma(x_{pq})$ , the ball centered at  $x_{pq}$  with radius  $\sigma$ . The ball  $B_\sigma$  is given either by a support of compactly supported smoothing kernels or it can represent a “numerical support” of the Gauss function (a domain in which values of the Gauss function are above some threshold given, e.g., by a computer precision). In any case just a finite sum in (13) is evaluated and coefficients of this sum, namely  $\int_r \nabla G_\sigma(x_{pq} - s) ds$ , can be precomputed in advance using a computer algebra system, e.g., Mathematica. It is worth noting that such an approach for evaluation of a diffusion coefficient  $g_{pq}^{\sigma, n}$  avoids explicit computation of gradients. We use this fact also in an adaptive scheme where computation of gradients on a nonuniform grid with the so-called “hanging nodes” could cause some difficulties.

As we have already noted, it is not possible to apply the previous scheme straightforwardly to adaptive nonuniform grids obtained by the coarsening algorithm. However, it is possible to modify it. For that goal, we will change a meaning of  $x_{pq}$  in (10) and definition (9) of

$T_{pq}$ . Let in the sequel  $x_{pq}$  be the middle point of the common boundary of two neighboring cells (with possibly nonequal measures). The definition of  $g_{pq}^{\sigma,n}$  will then remain the same. The only practical difference will be that the sum in (13) can be evaluated over nonequal control volumes. However, one can precompute all possible coefficients of the sum again in advance for every candidate larger cell on higher levels of hierarchy.

In the definition of  $T_{pq}$  in (9), the value  $|x_p - x_q|$  represents the distance used for approximation of the normal derivative  $(\bar{u}_q - \bar{u}_p)/|x_q - x_p|$ . Of course, in the case of a uniform rectangular grid with a unified size of cells,  $T_{pq}$  is equal 1. In the case of nonuniform rectangular grids, we can set this parameter using the average length of sides of two neighboring cells. Since our grids are balanced we put

$$\begin{aligned} T_{pq} &:= 1 && \text{if two inspected adjacent cells } p, q \text{ are of equal size,} \\ T_{pq} &:= \frac{2}{3} && \text{otherwise.} \end{aligned} \quad (14)$$

Let us note that in this case we obtain a scheme equivalent to (11) for a uniform grid with cells larger than the size on the finest level. The second possibility that we consider is given by

$$T_{pq} = \min\{l_p, l_q\}, \quad (15)$$

where  $l_p$  and  $l_q$  are the lengths of the sides of two adjacent cells  $p, q$  (of possibly nonequal measure). With this adjustment, for a uniform grid but with larger cells we do not get a scheme equivalent to (11). It is likely we assume an exchange of intensity between neighboring cells just in a strip of unit thickness along a boundary of a cell.

As our adaptive finite volume schemes we will consider system (11) where  $x_{pq}$  represents the middle point of the common boundary of two neighboring cells and  $T_{pq}$  is given by either (14) or (15). In the next section we will use both schemes for computations and present their results. They both represent reasonable modification of the basic scheme from [18].

In every discrete scale step, both schemes give a linear system that is symmetric and strictly diagonally dominant (with positive diagonal and negative numbers out of diagonal), which guarantees the existence of its unique solution, for which also  $L_\infty$  stability can be easily proved provided  $f \equiv 0$ .

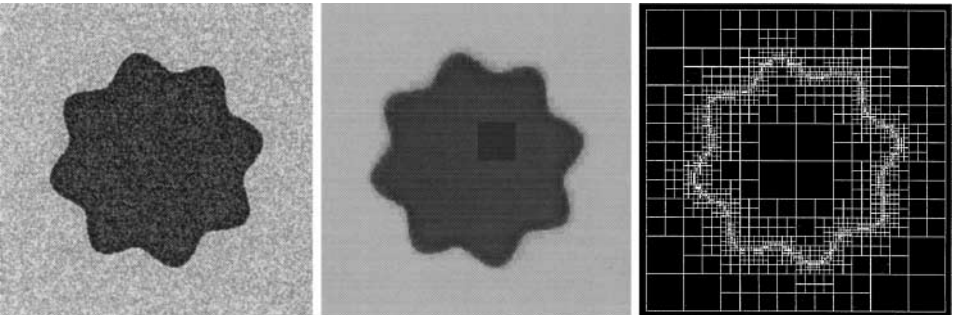


FIG. 4. Initial image (uniform noise), result of smoothing and adaptive grid (Example 1).

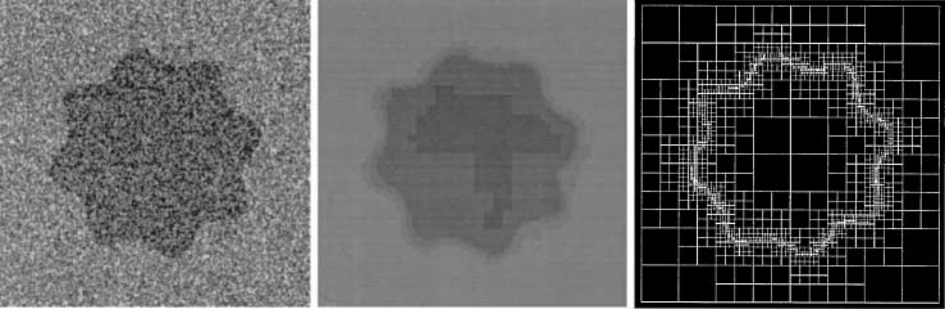


FIG. 5. Initial image (uniform noise), result of smoothing and adaptive grid (Example 2).

#### 4. NUMERICAL EXPERIMENTS

In this section we present experiments with some real as well as artificial images perturbed by various types of noise. In simulations, we use the function

$$g(s) = \frac{1}{1 + Ks^2},$$

and the convolution is realized with the kernel

$$G_\sigma(x) = \frac{1}{Z} e^{-\frac{|x|^2}{|x|^2 - \sigma^2}},$$

where the constant  $Z$  is chosen so that  $G_\sigma$  has unit mass. In order to compute the diffusion coefficient  $g_{pq}^{\sigma,n}$  we use the concept given in (13). In numerical experiments we have chosen  $\sigma = \frac{1}{2}$ , i.e., half size of the cell on the finest level, which is the fastest and simplest approach. In all examples, we work with gray-level images with intensity between 0 and 255, and before computations we transform intensity into the interval  $[0, 1]$ . In Examples 1–4, the results of the adaptive scheme (15) are presented (the results of the scheme (14) are practically very similar); in all further examples the scheme (14) has been used for presentation. In the figures we document the results of multiscale analysis (iterative filtering) as well as adaptive computational grids. In the tables we present a comparison of CPU times for adaptive/nonadaptive strategies and decreasing number of cells in the finite volume method. In every discrete scale step we first adjust the grid using the coarsening algorithm, and after

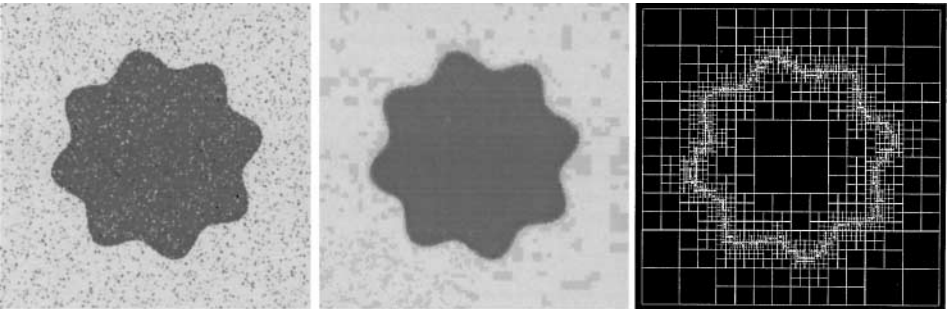
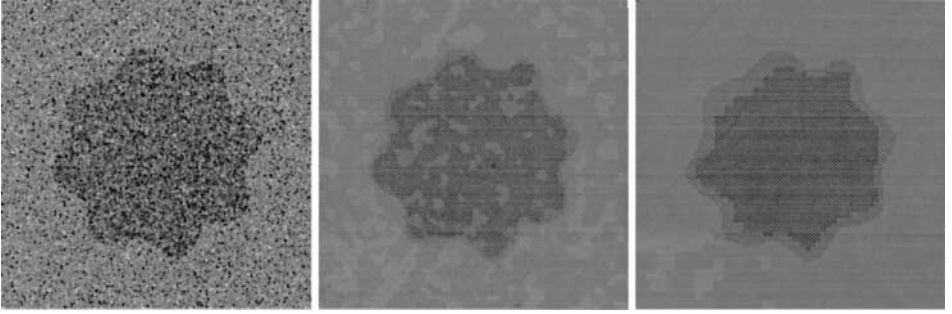


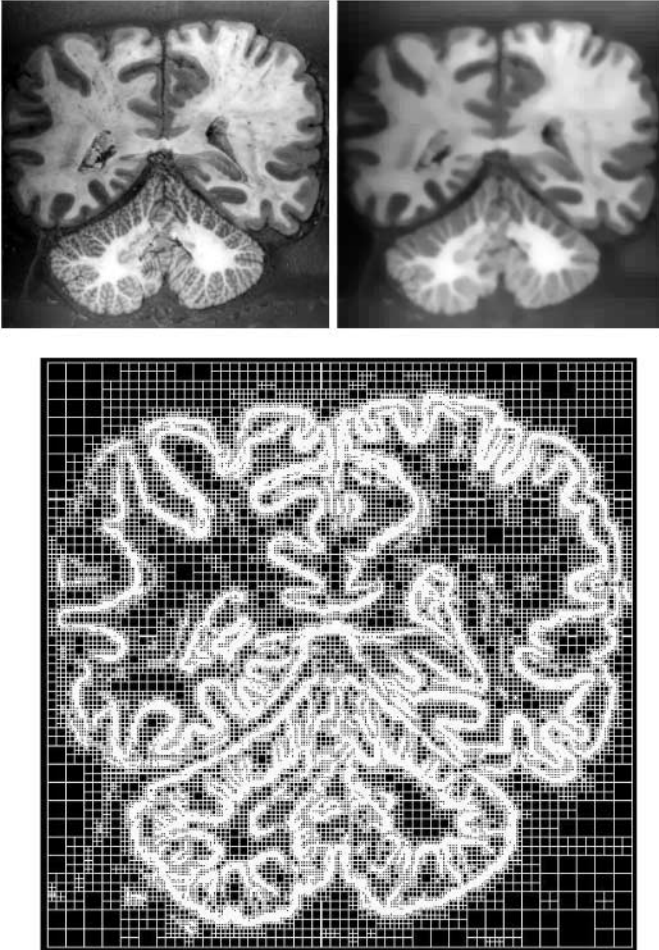
FIG. 6. Initial image (10% salt and pepper noise), result of smoothing and adaptive grid (Example 3).





**FIG. 7.** Initial image (50% salt and pepper noise) and results of successive smoothing (Example 4).

that we solve the linear system with a corresponding number of unknowns given in the tables. From experiments, one can see that the speed-up due to adaptivity is about three in rather general situations. All experiments were done on Pentium II (400 MHz) with a Linux operating system.



**FIG. 8.** Processing of medical image (Example 5); original (top left), nonlinear multiscale analysis after 25 scale steps (top right), final adaptive finite volume grid (down).

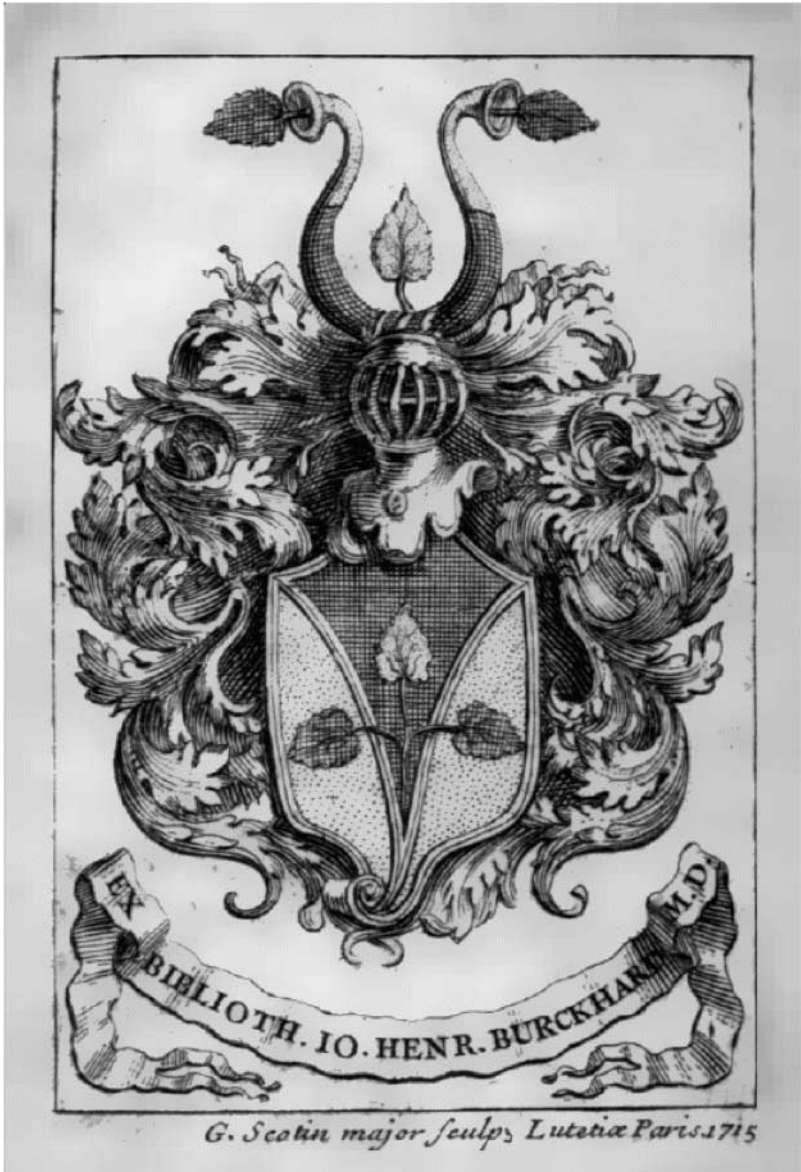


FIG. 9. Scanned *ex libris* from Lycean library in Kezmarok (Example 6).

*Examples 1 and 2.* In these two numerical experiments we add to a double-valued  $256 \times 256$  image  $\hat{u}$  a kind of uniform noise by a transformation

$$u_0(x) = \text{MIN}(255, \text{MAX}(0, \hat{u}(x) - C + \psi)/255),$$

where  $\psi$  is a random function generating integer values in  $[0, 2C]$ . In Example 1 the intensity difference in  $\hat{u}$  was 150 and  $C = 40$ ; in Example 2, the intensity difference was only 50 and  $C = 100$ . Figures 4 and 5 and Tables 1–4 show results. We have computed 10, respectively 20, scale steps of the adaptive algorithm. In Example 2 the comparable smoothing results were obtained after 25 steps of the nonadaptive algorithm with the same parameters. Scale



**FIG. 10.** Result of image selective smoothing using the adaptive finite volume method (Example 6).

step  $k = 1$  and the coarsening criterion in Example 1 was  $\varepsilon = 0.025$ , and in Example 2  $\varepsilon = 0.012$ .  $K = 1$  in both experiments.

*Examples 3 and 4.* In these two numerical experiments we work with  $512 \times 512$  images perturbed by salt and pepper noise. In Example 3 the intensity difference in  $\hat{u}$  was 150, and this was perturbed by 10% noise. For  $\varepsilon = 0.012$  we need 10 unit scale steps of the adaptive algorithm and 15 steps of the nonadaptive algorithm to get the results given in Fig. 6. Table 5 and Table 6 show the report on the role of coarsening (number of pixels for the initial image was  $262 \times 144$ ). In Example 4 the intensity difference in  $\hat{u}$  was 50, and 50% salt and pepper noise was used. For  $\varepsilon = 0.012$  we need 19 unit scale steps of the adaptive algorithm and

**TABLE 1**  
**Comparison of CPU Time for**  
**Adaptive/Fixed Mesh Strategies**  
**in Example 1**

	Pentium
Adaptive	3.04 s
Nonadaptive	9.81 s

**TABLE 2**  
**Decreasing Number of Unknowns in Example 1**

Scale step	1	2	3	4	5	6	7	10
No. of unknowns	65437	61936	32542	9595	7051	4405	3832	3802

**TABLE 3**  
**Comparison of CPU Time for**  
**Adaptive/Fixed Mesh Strategies**  
**in Example 2**

	Pentium
Adaptive	7.03 s
Nonadaptive	18.15 s

**TABLE 4**  
**Decreasing Number of Unknowns in Example 2**

Scale step	1	3	5	8	10	14	16	20
No. of unknowns	65533	65218	57061	37060	16378	8278	7039	6301

**TABLE 5**  
**Comparison of CPU Time for**  
**Adaptive/Fixed Mesh Strategies**  
**in Example 3**

	Pentium
Adaptive	11.41 s
Nonadaptive	50.13 s

**TABLE 6**  
**Decreasing Number of Unknowns in Example 3**

Scale step	1	3	4	6	7	8	9	10
No. of unknowns	127618	104947	91678	56941	33928	15227	10774	9730

**TABLE 7**  
**Comparison of CPU Time for**  
**Adaptive/Fixed Mesh Strategies**  
**in Example 4**

	Pentium
Adaptive	26.98 s
Nonadaptive	86.51 s

**TABLE 8**  
**Decreasing Number of Unknowns in Example 4**

Scale step	1	4	6	8	9	10	13	16	19
No. of unknowns	233623	212041	133129	62701	49423	31375	14758	8812	7351

30 steps of the nonadaptive algorithm to get the results given in Fig. 7 (we plot the 10th and 19th scale steps of the adaptive algorithm). The CPU times and decreasing number of unknowns are reported in Tables 7 and 8.  $K = 1$  in both experiments.

*Example 5.* An image of brain ( $512 \times 512$  pixels) is processed in this example. We run a multiscale analysis process for 25 scale steps with length  $k = 0.02$  and with  $\varepsilon = 0.025$ ,  $K = 12$ . The result and grid is documented in Fig. 8. The number of unknowns decreased from 262144 to 68062. Computational speed-up due to adaptivity was about 2.5.

*Example 6.* Figure 9 shows scanned *ex libris* from the Lycean library in Kežmarok ( $690 \times 1024$  pixels). Since the picture contains small dark spots and thin lines that are difficult to preserve during smoothing, we restrict Perona–Malik-type diffusion only to values over a certain threshold (in our case for intensities greater than 80). Such models were studied mathematically in [11, 12]. Here, we cut the dark pixels off and for an arising “internal boundary” we ensure zero Neumann conditions. Practically it means that for pixels below the threshold, the equation in the linear system is omitted. Figure 10 shows the result of such a smoothing accompanied with coarsening strategy. Together with conservation of dark spots we conserve also some discrepancies that can be found easily now and removed by applying, e.g., the heat equation in specified subregions of the image (here we clean the borders of the image in such a way).

## REFERENCES

1. L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, Axioms and fundamental equations of image processing, *Arch. Rat. Mech. Anal.* **123**, 1993, 200–257.
2. L. Alvarez, P. L. Lions, and J. M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.* **29**, 1992, 845–866.
3. L. Alvarez and J. M. Morel, Formalization and computational aspects of image analysis, *Acta Numer.* 1994, 1–59.
4. E. Bänsch, Local mesh refinement in 2 and 3 dimensions, in *IMPACT of Computing in Science and Engineering*, Vol. 3, pp. 181–191, 1991.
5. E. Bänsch and K. Mikula, A coarsening finite element strategy in image selective smoothing, *Comput. Visual. Sci.* **1**, 1997, 53–61.
6. E. Bänsch and K. Mikula, Adaptivity in 3D image processing, *Comput. Visual. Sci.* **4**, 2001, 21–30.
7. 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**, 1992, 182–193.
8. R. Eymard, T. Gallouet, and R. Herbin, Finite volume methods, in *Handbook for Numerical Analysis*, Vol. 7, pp. 715–1022, Elsevier, 2000.
9. A. Handlovičová, K. Mikula, and A. Sarti, Numerical solution of parabolic equations related to level set formulation of mean curvature flow, *Comput. Visual. Sci.* **1**, 1998, 179–182.
10. J. Kačur and K. Mikula, Solution of nonlinear diffusion appearing in image smoothing and edge detection, *Appl. Numer. Math.* **17**, 1995, 47–59.
11. J. Kačur and K. Mikula, Slow and fast diffusion effects in image processing, *Comput. Visual. Sci.* **3**, 2001, 185–195.

12. J. Kačur and K. Mikula, Slowed anisotropic diffusion, in *Proceedings of Scale-Space Theory in Computer Vision* (B. T. H. Romeny, L. Florack, J. Koenderink, and M. Viergever, Eds.), Lecture Notes in Computer Science 1252, pp. 357–360, Springer-Verlag, Berlin, 1997.
13. S. Kichenassamy, The Perona–Malik paradox, *SIAM J. Appl. Math.* **57**, 1997, 1328–1342.
14. P. L. Lions, Axiomatic derivation of image processing models, *Math. Models Methods Appl. Sci.* **4**, 1994, 467–475.
15. R. Malladi, J. Sethian, and B. Vemuri, Shape modeling with front propagation: a level set approach, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 1995, 158–174.
16. K. Mikula, Solution of nonlinear curvature driven evolution of plane convex curves, *Appl. Numer. Math.* **23**, 1997, 347–360.
17. K. Mikula, A. Sarti, and C. Lamberti, Geometrical diffusion in 3D-echocardiography, in *Proceedings of ALGORITMY'97—Conference on Scientific Computing, West Tatra Mountains—Zuberec*, 1997, pp. 167–181.
18. K. Mikula and N. Ramarosy, Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing, *Numer. Math.* **89**, 2001, 561–590.
19. K. Mikula and D. Ševčovič, Solution of nonlinearly curvature driven evolution of plane curves, *Appl. Numer. Math.* **31**, 1999, 191–207.
20. L. Moisan, Affine plane curve evolution: A fully consistent scheme, *IEEE Trans. Image Process.* **7**, 1998, 411–420.
21. M. Nitzberg and T. Shiota, Nonlinear image filtering with edge and corner enhancement, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1992, 826–833.
22. M. Ohlberger and M. Rumpf, Adaptive projection operators in multiresolutional scientific visualization, *IEEE Trans. Visual. Comput. Graphics* **4**, 1998, 344–364.
23. S. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.
24. P. Perona and J. Malik, Scale space and edge detection using anisotropic diffusion, in *Proc. IEEE Computer Society Workshop on Computer Vision*, 1987.
25. T. Preusser and M. Rumpf, An adaptive finite element method for large scale image processing, in *Proceedings of ScaleSpace'99*, 1999, pp. 223–234.
26. G. Sapiro and A. Tannenbaum, On affine plane curve evolution, *J. Funct. Anal.* **119**, 1994, 79–120.
27. A. Sarti, K. Mikula, and F. Sgallari, Nonlinear multiscale analysis of 3D echocardiographic sequences, *IEEE Trans. Medical Imaging* **18**, 1999, 453–466.
28. J. Weickert, B. M. t. H. Romeny, and M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Process.* **7**, 1998, 398–410.

---

ZUZANA KRIVA completed the study of theoretical cybernetics and mathematical informatics at the Faculty of Mathematics and Physics, Comenius University, in Bratislava. Currently she is an assistant professor in the Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, Slovakia. Her research interests include partial differential equations in image processing and visualization, computational geometry, and mesh generation.

KAROL MIKULA completed the study of computer science at the Faculty of Mathematics and Physics, Comenius University, in Bratislava. In 1993 he completed the Ph.D. in numerical analysis at the same university. Since 1991 he has been with the Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology, in Bratislava, Slovakia. He is the head of the Division of Computational Mathematics and an associate professor of applied mathematics. His research interests include numerical methods for nonlinear partial differential equations with applications in image processing, evolution of fronts, and porous media flow.