Proceedings of ALGORITMY 2009
pp. 121–133

ADAPTIVE DIAMOND CELL FINITE VOLUME METHOD IN IMAGE PROCESSING

ZUZANA KRIVÁ * AND KAROL MIKULA†

Abstract. New adaptive finite volume scheme for image processing applications is presented. It is based on a combination of primal, dual and diamond cell grids which leads to an efficient method built for the quadtree structures obtained by a coarsening process. The method is applied to the regularized Perona-Malik equation in image filtering.

Key words. Image processing, Perona-Malik equation, finite volume method, adaptivity

AMS subject classifications. 65M50, 65M60, 68U10

1. Introduction.

The regularized Perona-Malik nonlinear anisotropic diffusion is one of the best known models for removing the noise from the image data sets [13, 4, 1]. For this model, several adaptive techniques were suggested, e.g. in [2, 3, 10, 11, 14]. The adaptive approach is especially efficient if there are large areas of approximately constant intensity in the image domain, because the underlying grid and thus the number of unknowns in the resulting linear systems can be significantly decreased. Here we present a new adaptive method based on the finite volume discretization (FVM) of nonlinear diffusion equations [9, 12, 7] where the so called diamond cell technique, see e.g. [5, 6, 8, 7], is used for evaluation of gradients in conformal as well as nonconformal vertices of the adaptive quadtree grid.

2. Perona-Malik nonlinear anisotropic diffusion.

Since the end of the 80s, the nonlinear diffusion equations have been used for processing of 2D and 3D images. The pioneering work of Perona and Malik [13] who modified the linear heat equation to nonlinear diffusion preserving edge positions, was followed by the nonlinear PDE suggested by Catté, P.L.Lions, Morel and Coll [4] which is called the regularized Perona-Malik equation

$$u_t - \nabla.(g(|\nabla G_\sigma * u|)\nabla u) = 0. \tag{1}$$

Here, $u(t,x)$ is an unknown function representing smoothed (filtered) image intensity defined in $Q_T \equiv [0,T] \times \Omega$. The equations are accompanied by zero Neumann boundary conditions and the initial condition

$$\frac{\partial u}{\partial \nu} = 0 \quad \text{on} \quad I \times \partial\Omega, \tag{2}$$

$$u(0,x) = u^0(x) \quad \text{in} \quad \Omega, \tag{3}$$

where ν is the unit normal vector to the boundary of Ω. We assume that $g : I\!R_0^+ \to I\!R^+$ is a nonincreasing function , $g(0) = 1$, and we admit $g(s) \to 0$ for $s \to \infty$, $G_\sigma \in C^\infty(I\!R^d)$ is a smoothing kernel (e.g. the Gauss function), $u^0 \in L_\infty(\Omega)$. The

[*] Department of Mathematics, Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovak Republic, `kriva@math.sk`

[†] Department of Mathematics, Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovak Republic, `mikula@math.sk`

convolution is defined by $\nabla G_\sigma * u = \int\limits_{I\!R^d} \nabla G_\sigma(x-\xi)\tilde{u}(\xi)\,d\xi$, where $\tilde{u}$ is an extension of u to $I\!R^d$, e.g. a reflective periodic extension of the image [4].

3. Adaptive grid and finite volume discretization. Using FVM, we may subdivide the continuous image domain into rectangular regions and look for a solution, which will be constant over each such region - *finite volume*. In this paper, these regions will be given by an *adaptive grid*.

Adaptive grid. The initial image is given as a set of discrete grey (or RGB) values on cells of an initial regular - *nonadaptive* grid, which corresponds to the pixel structure of the image. In a nonadaptive method every element of such a grid will correspond to one unknown in a resulting linear system. To decrease the number of unknowns, we can decrease number of elements: at the beginning and especially with progress of smoothing algorithms, we can merge cells using some "coarsening" criterion and instead on the regular mesh, we will work on the irregular *adaptive* one. For its generation we chose an approach based on quadtrees in 2D (eventually octrees in 3D), which are the most convenient way to produce *graded* grids. In our case it means, that we have small elements, where the image information changes, i.e. in 2D near edges, and we have large ones, where it is of constant mean. Moreover, the quadtree itself may be computed in integer arithmetic and we will be able to suppress disadvantages of irregular grids: the storing of the grid can be performed in a very economic way. The technique of building the quadtree (and octree) adaptive grids is described in [10, 11] and we use the same **coarsening criterion**: the cells are merged if a difference in their intensities is below a prescribed tolerance ε. Moreover, the adaptive grids are **balanced**, it means that the ratio of the sides of adjacent cells is 1:1, 1:2 or 2:1. During the computations, the access to the neighbours of a processed finite volume is needed and balancing of a grid makes it easier. It also enables to form the diamond cells for computing the gradients.

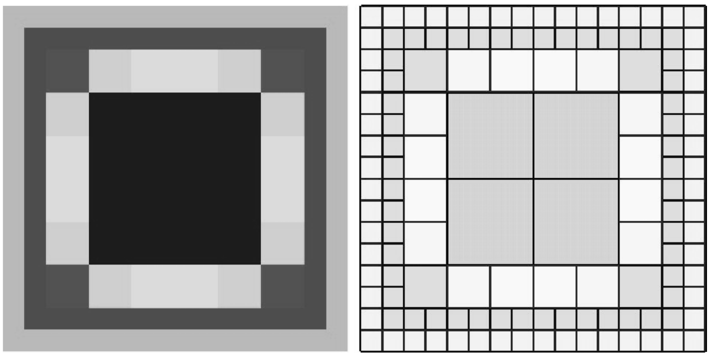

FIG. 1. *An example of an adaptive grid (on the right) for the image plotted on the left for $\varepsilon = 0$.*

Let $\mathcal{T}_h$ be an adaptive grid with finite volumes p of measure $m(p)$ and let $N(p)$ be the set of neigbors $q \in \mathcal{T}_h$ for which common interface of p and q is a line segment e_{pq} with nonzero measure $m(e_{pq})$. Having the grid, we can integrate the diffusion equation over a finite volume p and we use the divergence theorem to obtain

$$\int\limits_p \partial_t u dx - \int\limits_{\partial p} g(|\nabla G_\sigma * u|)\nabla u.\vec{n}_p \; ds = 0, \tag{4}$$

where $\vec{n}_p = (n_x, n_y)$ is the outward unit normal vector to ∂p. The equation (4) is called the integral form of the diffusion equation (1).

Replacing the time derivative by backward difference using the uniform time step $\tau = t^n - t^{n-1}$, where t^{n-1}, t^n are previous and current time steps, respectively, and denoting by

$$F_{pq} = \int\limits_{e_{pq}} g(|\nabla G_\sigma * u^{n-1}|)\nabla u^n . \vec{n}_{pq}\, ds \tag{5}$$

the semi-implicit flux through boundary e_{pq} between p and its neighbor q, the semi-implicit scheme can be rewritten in the following general form

$$(u_p^n - u_p^{n-1})\ m(p) - \tau \sum_{q \in N(p)} F_{pq} = 0, \tag{6}$$

where u_p^n is a representative value of solution in the finite volume p at time step t^n.

The flux F_{pq} contains a gradient of the smoothed solution in t^{n-1} and a normal derivative of solution at time step t^n evaluated on the boundary e_{pq} and we have to approximate them numerically. To that goal we will work with three types of grids.

- **Primal grid.** It is the adaptive grid with cells - finite volumes - obtained by merging controlled by the coarsening criterion, see fig 1. The numerical solution u_p^n will be computed for every finite volume of the primal grid and can be understood as the mean value on the cell or value at the representation point of the cell which is given by its center of the mass. The vertices of the primal grid will be also used in our adaptive method, but they do not represent degrees of freedom in our computations.
- **Diamond cell grid.** It is used for evaluation of gradients on the boundaries of finite volumes, i.e. on the boundaries of cells of the primal grid. The diamond cell is a quadrilateral with diagonals connecting the centers of finite volumes and endpoints of the boundary edge. The example of such a grid is in fig. 2 on the left.
- **Dual grid for bilinear approximation.** This grid is used for interpolation of solution values in the vertices of the primal grid which will be used in the gradient approximation by the diamond cell method on the diamond cell grid. The example of the dual grid for vertices of the primal grid is plotted in fig. 2 on the right.

4. Diamond cell method for evaluation of gradients. In (5) we can see, that we must evaluate the gradient on the boundaries of the finite volumes. To do so, we use the diamond cell method (see e.g. [5, 6, 7]). For every edge, the diamond cell is constructed in such way, that it connects representative points of the finite volumes and endpoints of the boundary edges, on which we evaluate the gradient (see fig. 2).

To get the approximation of the averaged gradient $\overline{\nabla_D u}$ on the diamond cell D, we use the divergence theorem to obtain

$$\overline{\nabla_D u} = \frac{1}{m(D)} \int\limits_D \nabla u dx = \frac{1}{m(D)} \int\limits_{\partial D} u \vec{\mathbf{n}}_D ds, \tag{7}$$

where $\vec{\mathbf{n}}_D$ is the outward normal unit vector to the boundary of D and $m(D)$ is the area of D.

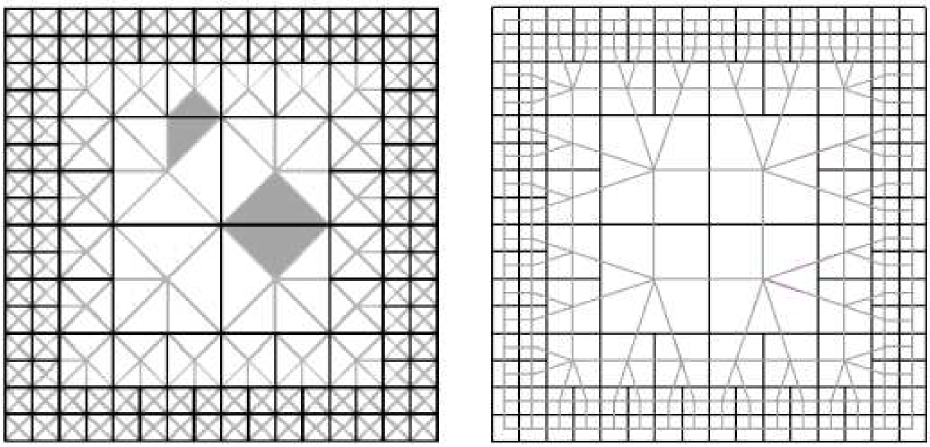

FIG. 2. *The diamond cell grid (grey lines on the left) and the dual grid (grey lines on the right) superimposed on the primal grid (bold black lines in both subfigures).*

Let σ be the edge of ∂D, then (7) is approximated by $\nabla_D u$ where

$$\nabla_D u = \frac{1}{m(D)} \sum_{\sigma \in \partial D} m(\sigma) \frac{1}{2} \left(u_{N_1(\sigma)} + u_{N_2(\sigma)} \right) \mathbf{\vec{n}}_\sigma, \tag{8}$$

where $N_1(\sigma)$ and $N_2(\sigma)$ are the endpoints of the edge σ with the length $m(\sigma)$.

4.1. The formula for evaluating the gradient in the local basis. Let the diamond cell D be constructed to an edge e which forms the boundary of two adjacent control volumes p and q. The gradient and the flux on e can be expressed in the local basis determined by the couple of vectors $(\vec{n}_e, \vec{t}_e)$, which are normal and tangential to e, by a general formula, cf. [5, 6, 8].

Let the representative point of the processed finite volume p be denoted x_W and the representative point of its neighbour q be denoted by x_E. The endpoints of the edge e are denoted by x_S and x_N in such way that the notation corresponds to the cardinal directionality, see fig.3, and let u_W, u_E, u_S, u_N denote the values of u in those points. Let $\vec{n}_e$ be the normal unit vector to e outward to p and $\vec{t}_e$ be a unit vector parallel to e such that $(x_N - x_S).\vec{t}_e > 0$.

In the local basis, the general formula for evaluation of the gradient of u on e corresponding to D by the the diamond cell approach (8) is given by

$$\nabla_D u = \left(\frac{u_E - u_W}{(x_E - x_W).\vec{n}_e} - \alpha_e \frac{u_N - u_S}{|x_N - x_S|} \right) .\vec{n}_e + \left(\frac{u_N - u_S}{(x_N - x_S).\vec{t}_e} \right) .\vec{t}_e, \text{ where}$$

$$\alpha_e = \frac{(x_E - x_W).\vec{t}_e}{(x_E - x_W).\vec{n}_e}. \tag{9}$$

Example 1. Let us apply the formula (9) to the case of finite volumes of nonequal size depicted in fig.3. The length of the common boundary is h. First let us take the

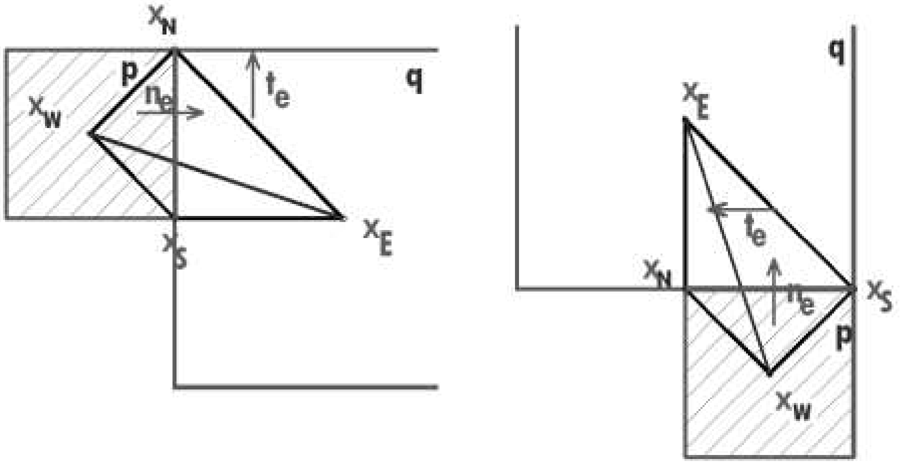

FIG. 3. *The notation for (9).*

case of the diamond cell depicted on the left. We get

$$\begin{aligned}\nabla_D u &= \left(\frac{u_E - u_W}{\frac{3}{2}h} - \left(-\frac{1}{3}\frac{u_N - u_S}{h}\right)\right).\vec{n}_e + \left(\frac{u_N - u_S}{h}\right).\vec{t}_e = \\ &= \left(\frac{2}{3}\frac{u_E - u_W}{h} + \frac{1}{3}\frac{u_N - u_S}{h}\right).\vec{n}_e + \left(\frac{u_N - u_S}{h}\right).\vec{t}_e\,.\end{aligned}\tag{10}$$

In the case of the diamond cell D on the right, we have

$$\begin{aligned}\nabla_D u &= \left(\frac{u_E - u_W}{\frac{3}{2}h} - \frac{1}{3}\frac{u_N - u_S}{h}\right).\vec{n}_e + \left(\frac{u_N - u_S}{h}\right).\vec{t}_e = \\ &\left(\frac{2}{3}\frac{u_E - u_W}{h} - \frac{1}{3}\frac{u_N - u_S}{h}\right).\vec{n}_e + \left(\frac{u_N - u_S}{h}\right).\vec{t}_e\,.\end{aligned}\tag{11}$$

The previous Example leads to the following observations:

- $(x_E - x_W).\vec{n}_e > 0$ and equal to h (for the finite volumes of the same size) or $\frac{3}{2}h$ (for the finite volumes of the nonequal size). Thus the first part of the normal derivative term is multiplied by 1 or $\frac{2}{3}$.
- α_e includes projections of $\overrightarrow{x_W x_E}$ onto $\vec{n}_e$ and $\vec{t}_e$. In the case of finite volumes of the same size, $\overrightarrow{x_W x_E} \perp \vec{t}_e$ and $\alpha_e = 0$. Otherwise, α_e is $\pm\frac{1}{3}$. The signum of α_e is given by the numerator, because the denominator containing $(x_E - x_W).\vec{n}_e$ is always positive. Thus it depends on the angle between $\overrightarrow{x_W x_E}$ and $\overrightarrow{x_S x_N}$ (resp. between $\overrightarrow{x_W x_E}$ and $\vec{t}_e$.)

4.2. Interpolation of u in the vertices of the primal grid. One diagonal of the diamond cell always connects a couple of vertices x_S, x_N of the primal grid. In these vertices the values of the function u will be approximated using the dual grid and bilinear interpolation.

Every internal vertex of the primal grid is a corner of three (in case of a hanging vertex) or four finite volumes (in conformal situation). If we connect the centers of these finite volumes, we get a cell of dual grid. For any vertex, such cell is a triangle or a quadrilateral, see fig.2 right. If we construct a bilinear (eventually linear) function using values of the solution in vertices of the quadrilateral (eventually triangle) given by degrees of freedom of solution computed on the primal grid, we get the weights necessary for interpolation in x_S, x_N, respectively, see fig.4 for all possible cases arising in our method.

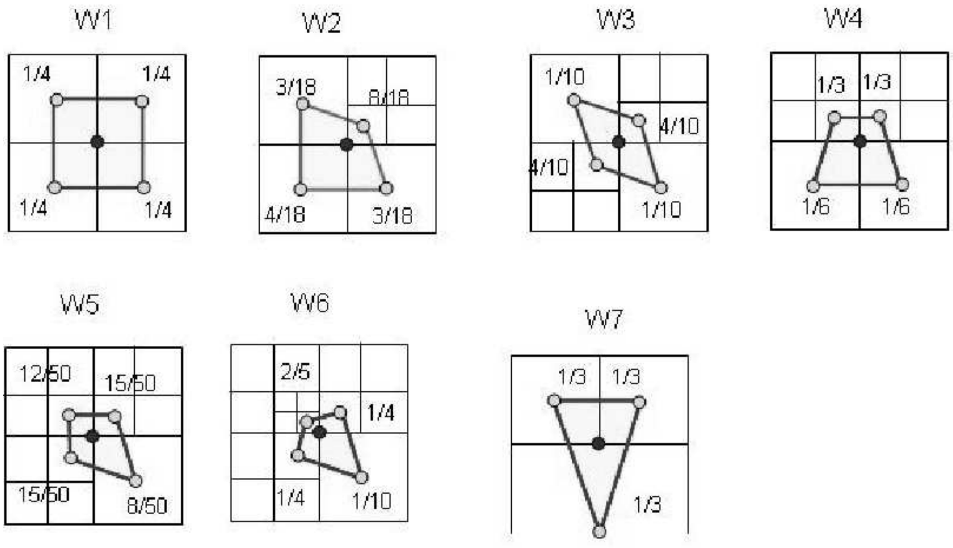

FIG. 4. *The weights for bilinear interpolation.*

4.3. Evaluation of the diffusion coefficient and the flux. In the equation (1), the gradient is used in two cases: its norm is an input of the Perona-Malik function g representing diffusion coefficient, and, we need the gradient on the boundary of finite volume multiplied by the outward unit normal in the flux evaluation, see (5).

4.3.1. Evaluation of the diffusion coefficient $g(|\nabla G_\sigma * u|)$. The convolution $G_\sigma * u$ can be replaced by the solution of the linear heat equation (i.e. the equation (1) with $g \equiv 1$), at time σ. The linear heat equation will be solved numerically by one implicit time step on the same adaptive grid. To that goal we consider σ instead of τ in (5) and the diffusion flux F_{pq} is replaced by the flux for the linear heat equation. Using (9) the linear diffusion flux can be approximated by

$$(12) \qquad F^c_{pq} = \nabla_D u.\vec{n}_{pq}\; m(e_{pq}) = \left(\frac{u_E - u_W}{(x_E - x_W).\vec{n}_{pq}} - \alpha_{e_{pq}} \frac{u_N - u_S}{|x_N - x_S|} \right) m(e_{pq}),$$

where D is the diamond cell for the edge e_{pq} between p and q.

To get a more compact formula for the flux (12), we change notation as follows. The value in p resp. q will be denoted by u_p resp. u_q, and, if the finite volumes are of different size, the endpoints of the common edge will be denoted by x_H in case of a hanging vertex (here, a vertex belonging to three finite volumes) and x_R for a regular (conformal) vertex (here, a vertex belonging to four finite volumes). Then $\tilde{u}_R$ and $\tilde{u}_H$ will be the values in x_R and x_H obtained by the bilinear approximation. The first part of the flux (12) will be written in the form $T^1_{pq}(u_q - u_p)$ and the second part in the form $T^2_{pq}(\tilde{u}_R - \tilde{u}_H)$. We will distinguish three cases:

- q is of the same size as p. Then $F^c_{pq} = u_q - u_p$, so $T^1_{pq} = 1$ and $T^2_{pq} = 0$.

- q is of the half size of p. The situation is depicted in fig. 5. On the left $\alpha = \frac{1}{3}$, on the right $\alpha = -\frac{1}{3}$, thus $F^c_{pq} = \frac{2}{3}(u_q - u_p) - \frac{1}{3}(\tilde{u}_R - \tilde{u}_H)$, $T^1_{pq} = \frac{2}{3}$ and $T^2_{pq} = -\frac{1}{3}$.
- q is of the double size of p. The situation is depicted in fig.6. On the left $\alpha = \frac{1}{3}$, on the right $\alpha = -\frac{1}{3}$, thus $F^c_{pq} = \frac{2}{3}(u_q - u_p) + \frac{1}{3}(\tilde{u}_R - \tilde{u}_H)$, $T^1_{pq} = \frac{2}{3}$ and $T^2_{pq} = \frac{1}{3}$.

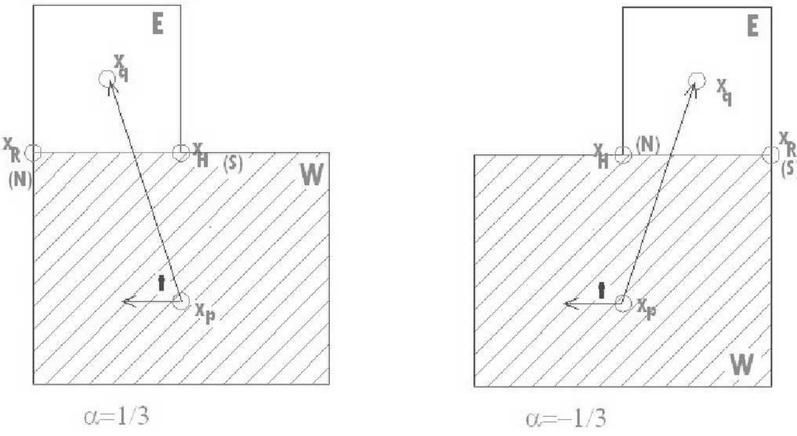

FIG. 5. *Evaluation of the flux in the case of a smaller neighbor. The signum of α depends on the angle between the vector $x_p x_q$ and $\vec{t}_e$.*

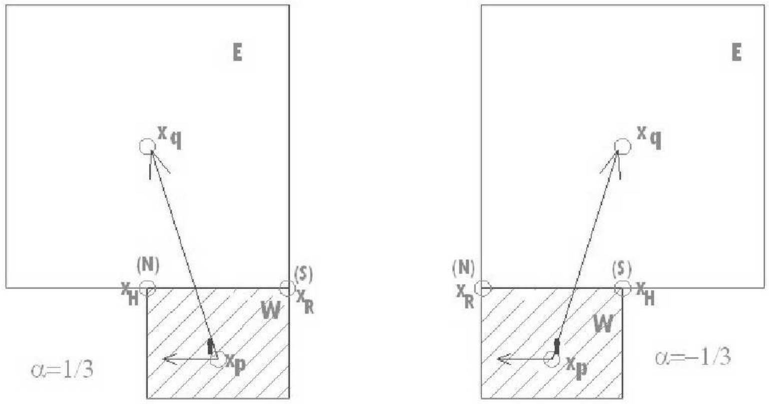

FIG. 6. *Evaluation of the flux in the case of a bigger neighbor. The signum of α depends on the angle between the vector $x_p x_q$ and $\vec{t}_e$.*

The solution of the linear heat equation at time σ will be denoted by u^c. Then the diffusion coefficient $g(|\nabla G_\sigma * u|)$ on any edge e is replaced by $g(|\nabla_D u^c|)$ where $g(s) = \frac{1}{1+K*s^2}$, $K \geq 0$ and D is a diamond cell corresponding to e.

4.3.2. Evaluation of the flux for the Perona-Malik equation. Now, the flux (5) can be approximated by

$$F_{pq} = g(|\nabla_D u^c|)T^1_{pq}(u_q - u_p) + g(|\nabla_D u^c|)T^2_{pq}(\tilde{u}_R - \tilde{u}_H), \tag{13}$$

where

- for q of the same size as p, $T^1_{pq} = 1$ and $T^2_{pq} = 0$.
- for q is of the half size of p, $T^1_{pq} = \frac{2}{3}$ and $T^2_{pq} = -\frac{1}{3}$.
- for q of the double size of p, $T^1_{pq} = \frac{2}{3}$ and $T^2_{pq} = \frac{1}{3}$.

5. The adaptive finite volume scheme for the regularized Perona-Malik equation. In order to define the scheme the approximated value in a hanging vertex of the edge e_{pq} will be denoted by $\tilde{u}_{H(e_{pq})}$ and in a regular one $\tilde{u}_{R(e_{pq})}$. Now, we can write the fully discrete **semi-implicit adaptive finite volume scheme** for solving regularized Perona-Malik problem:
Let $0 = t_0 \le t_1 \le ... \le t_N = T$ denote the time stepping with $t_n = t_{n-1} + \tau$, where τ is the time step. For $n = 1, ..., N$ we look for u^n_p, $p \in \mathcal{T}_h$ satisfying

$$\frac{u^n_p - u^{n-1}_p}{\tau} m\,(p) = \sum_{q \in N(p)} g(|\nabla_D u^c|) T^1_{pq} \left(u^n_q - u^n_p\right) + g(|\nabla_D u^c|) T^2_{pq} \left(\tilde{u}^n_{R(e_{pq})} - \tilde{u}^n_{H(e_{pq})}\right), \tag{14}$$

where u^c replaces the convolution $G_\sigma * u^{n-1}$ and is obtained by solving the system

$$\frac{u^c_p - u^{n-1}_p}{\sigma} m\,(p) = \sum_{q \in N(p)} T^1_{pq} \left(u^c_q - u^c_p\right) + T^2_{pq} \left(\tilde{u}^c_{R(e_{pq})} - \tilde{u}^c_{H(e_{pq})}\right), \tag{15}$$

for some σ (usually $\sigma < \tau$) and T^1_{pq} and T^2_{pq} are given in Section (4.3.2).

The **adaptive algorithm** has six phases in every time step:

1. *We build the primal - adaptive grid according to the coarsening criterion.*
2. *We interpolate the values in the vertices* of this grid using bilinear interpolation by recursive traversal of the quadtree. During this traversal, for every vertex of processed volume, all volumes containing the vertex must be detected and according to their size the corresponding set of weights must be determined. The pointers to volumes and weights are stored in the database (because our scheme is semi implicit and we use Gauss-Seidel method for solving the linear system).
3. *We compute the coefficients of the system (15)* by recursive traversal of the quadtree corresponding to the adaptive grid.
4. *We solve the linear system (15).*
5. *We compute the coefficients of the system (14)* by recursive traversal of the quadtree corresponding to the adaptive grid.
6. *We solve the linear system (14).*

6. Numerical experiments.

6.1. Experimental order of convergence (EOC) for the linear heat equation. Let us have the linear diffusion equation

$$u_t - \Delta u = f, \tag{16}$$

where $u = u(x, y, t)$, $f = f(x, y, t)$, $(x, y) \in (0, 1) \times (0, 1) = \Omega$ and $t \in [T_1, T_2]$. For the function $f(x, y, t) = \cos(2\pi x)\cos(2\pi y)(1 + 8\pi^2 t)$, the exact solution is given by $u(x, y, t) = \cos(2\pi x)\cos(2\pi y)t$. We add numerically evaluated right hand side f to the scheme (14) with $g \equiv 1$ and solve the problem in time interval $[0.5, 0.6]$. We compute $L_2(I, L_2(\Omega))$-norm of the error by the formula

$$\sqrt{\sum_{n=1}^{N} \tau \sum_{p} (u(x_p, t^n) - u_p^n)^2 m(p)}. \tag{17}$$

The experimental order of convergence is then evaluated by

$$EOC = \log_2 \left(\frac{E_i}{E_{i+1}} \right), \tag{18}$$

where E_i, E_{i+1} are the errors on subsequently refined (nonuniform adaptive) grids obtained by dividing every finite volume into four subvolumes. Let us note that the grid is not changing in time in the Experiments 1a, 1b and 2, although we solve the time dependent problem.

Experiment 1a. The uniform grid is of the size 16×16 ($h = \frac{1}{16}$) and by a coarsening we construct our initial nonuniform grid with 160 finite volumes depicted in fig. 7. To study EOC, we refine this grid three times. The evaluation of the errors is shown in Table 1. The finest nonuniform grid and the corresponding numerical solution at time 0.6 are depicted in fig. 8.

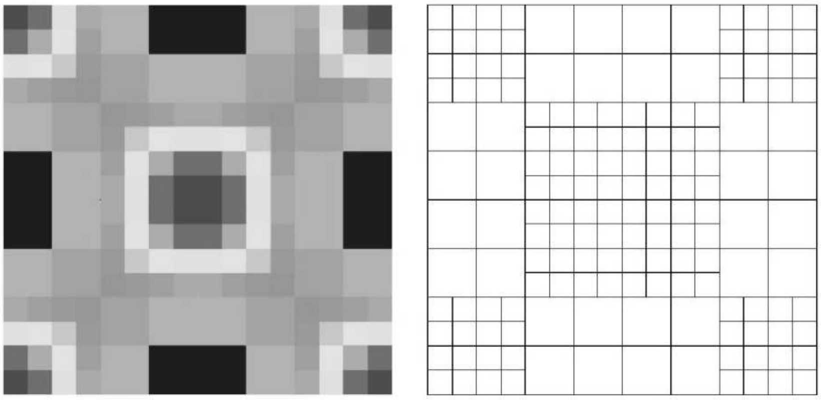

FIG. 7. *The initial coarse grid and u^0 for Experiment 1a.*

Experiment 1b. In the second experiment we used a different initial nonuniform grid, which is more irregular. The initial state and the numerical results are depicted in fig.9 and fig.10. The evaluation of the errors is shown in the Table 1. In both cases we observed the second order accuracy of the scheme.

Experiment 2. In this case we created an adaptive grid from a uniform one with 128×128 finite volumes using various values of the coarsening threshold ε, see section 3. Coarsening was adapted to the solution $u(x, y, t) = \cos(2\pi x)^{20} \cos(2\pi y)t$ at time $t = 0.5$, since the function has several "flat" regions suitable for using adaptivity. We performed one time step with $\tau = (\frac{1}{128})^2$ and evaluated the error for different values of ε. The results are displayed in Table 2 and show that no significant error increase can be observed using adaptivity, while the number of finite volumes is decreased which results in higher computation efficiency of the adaptive method.

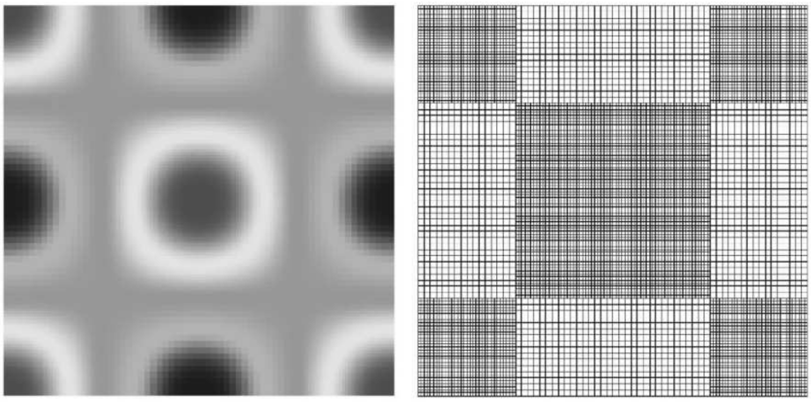

FIG. 8. *The finest grid and numerical solution at time 0.6 for Experiment 1a.*

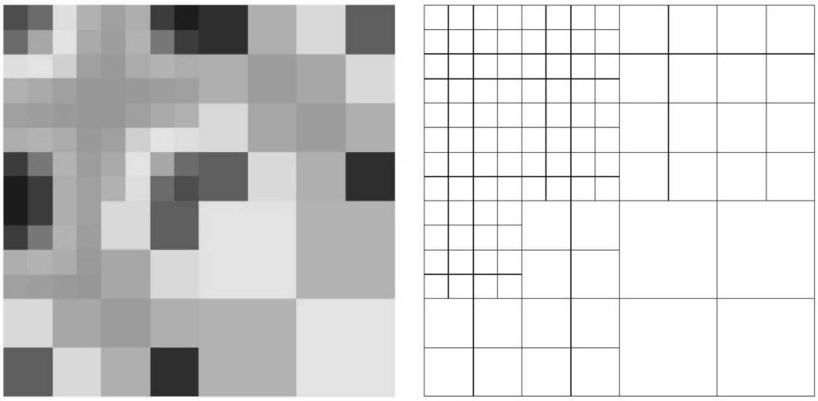

FIG. 9. *The initial coarse grid and u^0 for Experiment 1b.*

6.2. Applying the Perona-Malik adaptive algorithm to image filtering. **Experiment 3.** Figure 12 shows the original image perturbed by an additive noise, the result of smoothing and the resulting mesh in the upper row and the details are presented in the lower row. We used $g(s) = \frac{1}{1+Ks^2}$, $K = 2$, $\tau = 0.0004$, $\sigma = 0.0001$, $h = 0.01$ and we performed 25 time steps. The initial image was of the size 256×256 pixels and thus contains 65536 finite volumes. After 5 time steps, the number of finite volumes decreased to 38215, after 10 time steps to 5824 and at the end we had 4423 finite volumes.

Experiment 4. The adaptive approach is especially efficient if we have large areas of approximately constant intensity. We present 2D experiment dealing with a slice of 3D volume acquired by a two-photon laser scanning microscopy. The data represent the nuclei of the zebra fish embryo, and many of the slices contain only small regions of image information. Figures 13 and 14 shows the original noisy data, the result of smoothing and the resulting mesh. We used $\varepsilon = 0.008$, $\tau = 0.0003$, $\sigma = 0.0002$, $K = 2$ and $h = 0.01$. We performed 20 time steps. The initial image is of the size 512×512 and contains 262144 pixels. We observed a very fast decrease of finite volumes, after one time step, it was 145358, after the second time step 10702 and at the end we

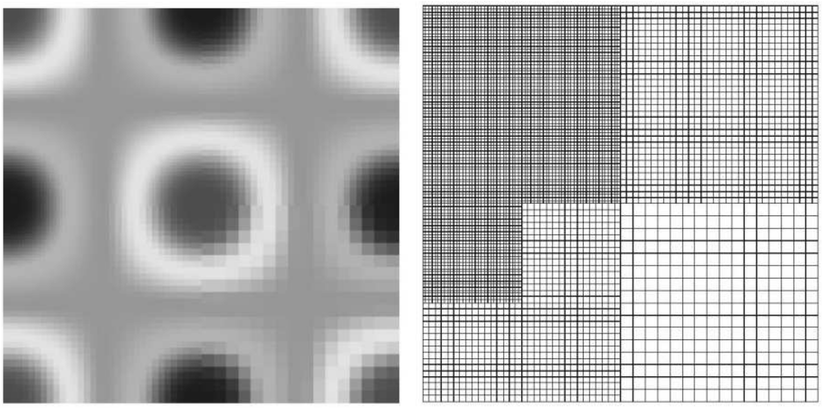

FIG. 10. *The finest grid and numerical solution at time 0.6 for Experiment 1b.*

TABLE 1

EOC calculations, from the left: h means the size of finite volumes in the initial grid if there would be no coarsening, τ is the time step, N is the number of time steps performed over $T =$ $[0.5, 0.6]$. *Then the errors $E(h)$ and EOC for both Experiments 1a and Experiments 1b are given.*

h	τ	N	Exper.1a $E(h)$	Exper.1a *EOC*	Exper.1b $E(h)$	Exper.1b *EOC*
$\frac{1}{16}$	0.003906	25	0.00708300		0.011175	
$\frac{1}{32}$	0.000977	102	0.00170000	2.06	0.002872	1.96
$\frac{1}{64}$	0.000244	409	0.00041700	2.03	0.000728	1.98
$\frac{1}{128}$	0.000061	1638	0.00010400	2.00	0.000182	2.00

had 8170 finite volumes. While at the beginning, the linear system solving took 0.5 seconds, the final one was solved only in 0.02s. The figure 14 shows the details of the data.

Acknowledgements. This work was supported by grants APVV-RPEU-0004-06, APVV-0351-07, APVV-LPP-0020-07, VEGA 1/0269/09 and European projects Embryomics and BioEmergences.

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 (1993) pp. 200–257.

[2] E. BÄNSCH, K. MIKULA, *A coarsening finite element strategy in image selective smoothing,* Computing and Visualization in Science, Vol.1, No.1 (1997) pp. 53–61.

[3] E. BÄNSCH, K. MIKULA, *Adaptivity in 3D image processing,,* Computing and Visualization in Science, Vol.4, No.1 (2001) pp. 21–30.

[4] V. CATTÉ, P.L. LIONS, J.M.. MOREL, T. COLL, *Image selective smoothing and edge detection by nonlinear diffusion,* SIAM J. Numer. Anal., Vol. 29 (1992) pp. 182–193.

[5] Y. COUDIÉRE, P. VILLEDIEU, *Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem,* ESAIM: M2AN, 33(3) (2000) pp. 493–516.

[6] Y. COUDIÉRE, J.P. VILA, P. VILLEDIEU, *Convergence rate of a finite volume scheme for the linear convection-diffusion equation on locally refined meshes,* ESAIM: M2AN, 34(6) (2000) pp. 1123–1149.

[7] O.DRBLIKOVÁ, K.MIKULA, *Convergence analysis of finite volume scheme for nonlinear tensor anisotropic diffusion in image processing,* SIAM Journal on Numerical Analysis, Vol. 46, No.1 (2007) pp. 37-60.

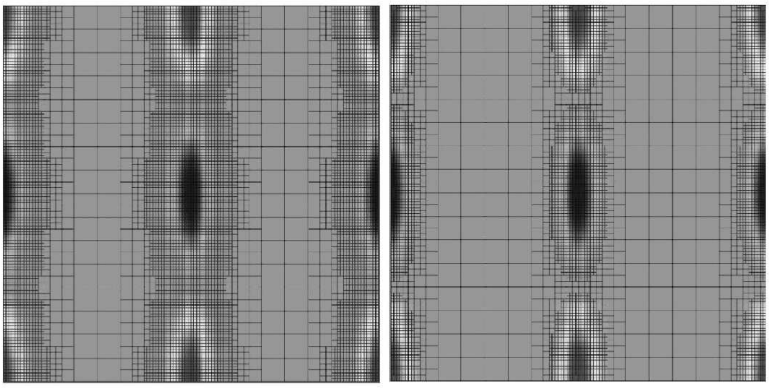

FIG. 11. *The adaptive grids for Experiment 2, on the left with* $\varepsilon = 0.001$ *on the right with* $\varepsilon = 0.025$.

TABLE 2

Evaluation of the errors and report on number of finite volumes in the grids for the Experiment 2. Except of the error, and the threshold value ε*, the table shows the decrease of the finite volumes.*

ε	Errors	Number of FV
0.0000	0.0000356447	16380
0.0001	0.0000356450	9424
0.0010	0.0000356565	8080
0.0100	0.0000358390	6304
0.0250	0.0000368051	4768

[8] C. Erath, *Adaptive finite volume metode,* Diploma thesis (in German), Vienna University of Technology, 2005.

[9] R. Eymard, T. Gallouet, R. Herbin, *The finite volume method,* in: Handbook for Numerical Analysis, Vol.7 (Ph. Ciarlet, J. L. Lions, eds.), Elsevier, 2000.

[10] Z. Krivá, K. Mikula, *An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing,* J. Visual Communication and Image Representation, Vol. 13 (2002) pp. 22–35.

[11] Z. Krivá, K. Mikula, *Adaptive Finite Volume Schemes in Processing of 3D Data Sets,* Journal of Electrical Engineering Volume 12/S, 52 (2001) pp. 53–58.

[12] K. Mikula, N. Ramarosy, *Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing,* Numer. Math., Vol.89, No.3 (2001) pp. 561–590.

[13] P. Perona, J. Malik, *Scale space and edge detection using anisotropic diffusion,* Proc. IEEE Computer Society Workshop on Computer Vision (1987)

[14] T. Preusser,M. Rumpf, *An Adaptive Finite Element Method for Large Scale Image Processing,* J. Visual Communication Image Representation, 11 (2) (2000) pp. 183–195.

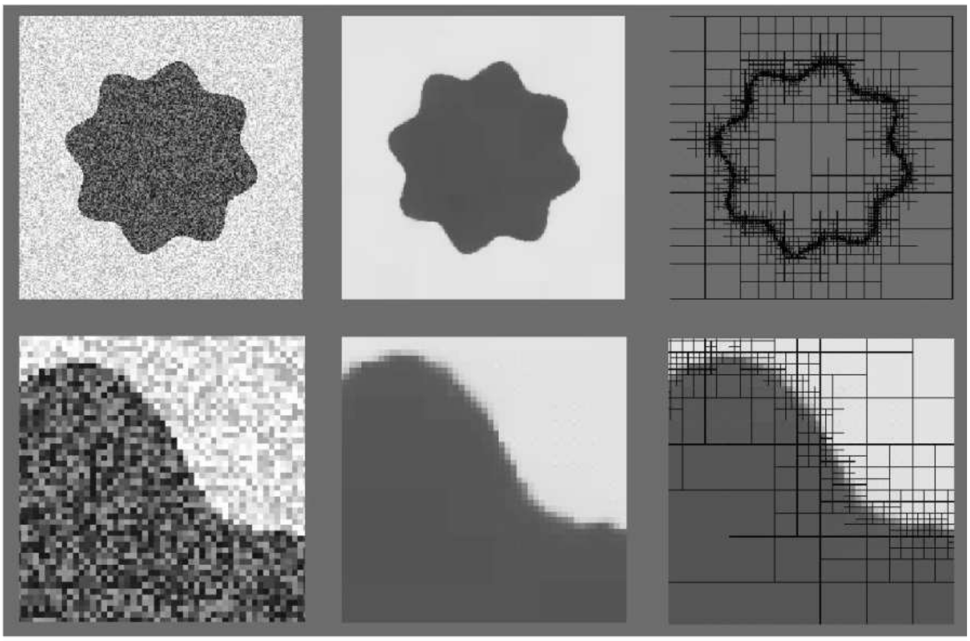

FIG. 12. *The adaptive smoothing of the regularized Perona-Malik PDE.*

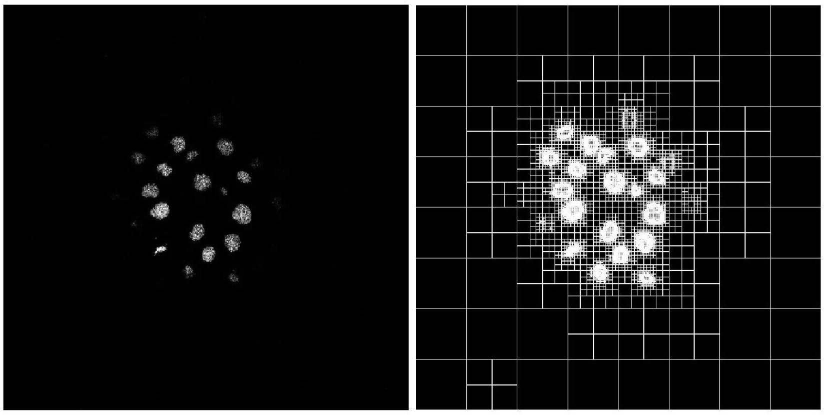

FIG. 13. *The adaptive smoothing of the regularized Perona-Malik PDE.*

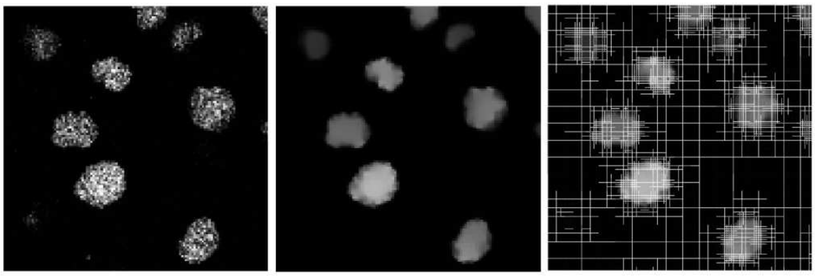

FIG. 14. *The adaptive smoothing of the regularized Perona-Malik PDE.*