Adv Comput Math
DOI 10.1007/s10444-015-9423-2

# Adaptive cell-centered finite volume method for diffusion equations on a consistent quadtree grid

**Zuzana Krivá[1] · Angela Handlovičová[1] · Karol Mikula[1]**

Received: 21 March 2014 / Accepted: 25 May 2015


**Abstract** Models applied in image processing are often described by nonlinear PDEs in which a good approximation of gradient plays an important role especially in such cases where irregular finite volume grids are used. In image processing, such a situation can occur during a coarsening based on quadtree grids. We present a construction of a deformed quadtree grid in which the connection of representative points of two adjacent finite volumes is perpendicular to their common boundary enabling us to apply the classical finite volume methods. On the other hand, for such an adjusted grid, the intersection of representative points connection with a finite volume boundary is not a middle point of their common edge and standard methods cannot achieve a good accuracy. In this paper we present a new cell-centered finite volume method to evaluate solution gradients, which results into a solution of a simple linear algebraic system and we prove its unique solvability. Finally we present numerical experiments for the regularized Perona-Malik model in which we applied this new method.

**Keywords** Image processing · The linear heat equation · Finite volume method · Adaptivity

Communicated by: Y. Xu

✉ Zuzana Krivá
kriva@math.sk

Angela Handlovičová
angela.handlovicova@stuba.sk

Karol Mikula
mikula@math.sk

[1] Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 810 05 Bratislava, Slovak Republic

Published online: 12 August 2015

**Mathematics Subject Classifications (2010)** 65M50 · 65M60 · 68U10

## 1 Introduction

In image processing many filtering algorithms are based on nonlinear diffusion PDEs which modify the linear diffusion (heat) equation by slowing down the diffusion in the vicinity of edges. The linear heat equation is not only the base of these reliable and mathematically approved methods but it can be applied also when evaluating the so called "Gaussian gradient", used for regularization and edge detection not only in the regularized Perona-Malik equation we solve, but also in some modifications of level set type PDEs models [2, 7, 8, 15, 22, 27, 28].

For some of these models adaptive methods have been developed [3, 4, 10, 17–19, 23, 24] benefitting from the fact that with the progress of denoising larger elements can be used in a computational grid. In this work we present a new finite volume scheme working on a deformed graded quadtree grid generalizing the method for an evaluation of gradients presented in [12].

This paper is organized as follows: first we deal with a numerical solution to the linear heat equation on the so called *consistent* adaptive grid. It is obtained by modification of a graded quadtree, i.e. the quadtree in which the difference in a level between adjacent cells is constrained, in our case to one. In a nonconforming situation the basic quadtree grid does not satisfy the classical orthogonality property which is mandatory in the standard finite volume method [11]. However, the graded quadtree can be deformed into a *consistent* quadtree grid in such a way that in the resulting polygonal grid the connection of two representative points of two adjacent finite volumes is perpendicular to their common boundary. Examples of basic quadtree and consistent grids are displayed in Fig. 1. The process of the deformation and geometric properties of resulting polygonal grid are presented in Section 2. The consistent grid enables us to solve the linear heat equation in a standard way: the corresponding

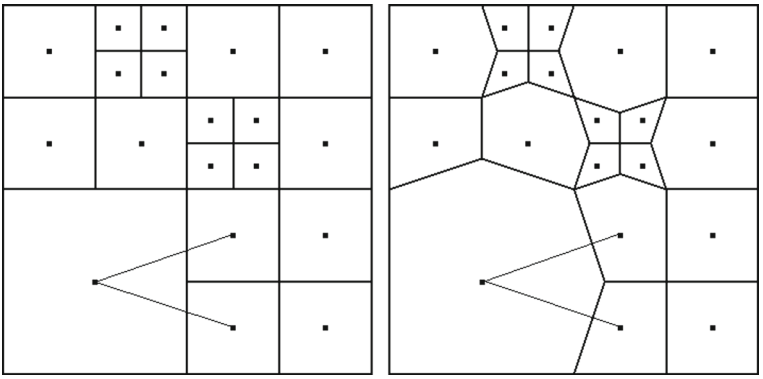

**Fig. 1** An example of the original quadtree grid together with the representative points of its elements (*on the left*). This grid is transformed into the consistent one (*on the right*)

implicit finite volume scheme and the experimental order of convergence (EOC) are presented in Section 3.

The linear heat equation can be a stepping stone for nonlinear diffusion equations requiring solution gradients and their norms. We have chosen to solve the regularized Perona-Malik equation [8, 22] for which several adaptive methods have already been elaborated [3, 4, 9, 17, 18], but working on triangular or rectangular grid elements. Our method to evaluate the gradients is based on [12] dealing with similar nonlinearities in the level set curvature driven equations. We consider also representative points on finite volume edges, which are updated using the conservation principle. Comparing with the diamond cell adaptive method [18] the advantage is that we work locally in the sense that we only need to access neighbors sharing a common edge, not a vertex only, because if the neighborhood information of the grid is not stored explicitly, finding the neighbors with a common vertex leads to too many tests and can be time consuming. In the standard approach based on [12] the represenative points are obtained as an intersection of a line segment connecting representative points of two adjacent finite volumes and their common edge which must be mutually orthogonal. Moreover this intersection must be a midpoint of the edge, but such a requirement is not fulfilled in the consistent adaptive grid. In the present work we generalize the gradient evaluation in such a way that this midpoint condition can be relaxed. The method leading to a simple algebraic linear system is described in Section 4, where also unique solvability of the linear system and properties of its coefficient matrix have been shown.

In Sections 5 and 6 we derive the semi-implicit finite volume scheme for the regularized Perona-Malik equation and present numerical experiments.

## 2 The adaptive grid

A region representation is an important issue in image processing and consequently a number of representations are currently in use among which quadtree techniques [1, 21, 25, 26] are well suited for large set of applications. To mention some of them - the finite difference schemes for the Poisson and heat equations on non-graded quadtrees [13, 19] and the multigrid finite element schemes for PDEs in image processing [10, 23, 24]. For the finite volume method, a cell centered finite volume discretization to Poisson equation using multigrid approach, has been used already in 1998 [14]. For the elliptic boundary value problems the higher order finite volume method have been studied in [5, 6].

Recently adaptive finite volume schemes working on quadtrees has been proposed in [17, 18]. However, our consistent grid is not Cartesian, but polygonal one, possessing the properties enabling us to use the methods based on [12].

In image filtering, the idea to use more and more "coarse" elements with progress of smoothing has been used for the first time in [3] for the regularized Perona-Malik equation. It benefits from the fact that 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 the algorithm works just with grids, elements of which are obtained by merging of pixels. This process has been called a *coarsening* in image

processing. We want to use these principles when working on the consistent finite volume adaptive grid. First, let us describe building the quadtree.

### 2.1 Building the quadtree

The initial image is given as a set of discrete grey (or RGB) values on cells of an initial regular - *nonadaptive* grid corresponding to the pixel structure of the image. We build the quadtree by a merging of the elements with similar intensities from smaller cells to larger cells, i.e. from leaves to the root. The old values are either unchanged, or replaced by an averaging the old values from the processed area. During this process the information about successful or unsuccessful merging is stored in a binary field with the size corresponding to the image. Moreover this information is stored in such a way that it enables us to create a graded quadtree with prescribed ratio of elements. It can be easily used as a stopping criterion during *traversing* the quadtree and to test the configurations of elements - the leaves of the quadtree.

As we have already mentioned, we build the graded quadtree, i.e. we require the ratio of sides of two adjacent squares to be $1:1$, $1:2$ or $2:1$. Grids associated with such trees are often used in order to produce procedures that are easier to implement, moreover, in our case it is an inevitable requirement to be able to build the consistent finite volume grid. The used technique of building the quadtree (and octree) adaptive grids is described in [16, 17] where the following **coarsening criterion** is used: the cells are merged if a difference in their intensities is below a prescribed tolerance $\varepsilon$.

### 2.2 Adjustment to the quadtree consistent grid

The quadtree grid (Fig. 1 left) is *inconsistent* in the sense, that we cannot find the unique representative points of the adjacent grid elements - finite volumes - such that the connection of their representative points is perpendicular to their common boundary. The adaptive grid fulfilling this condition is called *consistent* and it is an *admissible* mesh in the sense of [11]. However, the basic quadtree grid can be adjusted to a consistent one procedurally: we must adjust the shape, if two adjacent finite volumes $p$ and $q$ are of a different size. If we denote the length of a common edge in the original quadtree by $h$ and we shift the "hanging node" by $v = \frac{h}{3}$ (e.g. in Fig. 2 we shift $X$ to $X'$), then the connection of representative points is perpendicular to the shifted common boundary. This fact (and also the fact that $\frac{BX'}{PQ} = \frac{2}{3}$) follows from the similarity of triangles $\triangle AQP$ and $\triangle XX'B$ with the ratio of their adjacent sides 1:3. The area of $p$ is also evaluated procedurally - it depends on a configuration of its neighbors.

### 2.3 Notations

Let $\mathcal{T}_h$ be a polygonal adaptive grid with finite volumes $p$ of a measure $|p|$. Let every finite volume $p$ have a representative point $X_p$ lying in its center or in the center of the original square for an adjusted element of the consistent grid. Let $N_p$ denote the set of neighbors $q \in \mathcal{T}_h$ for which common interface of $p$ and $q$ is a line

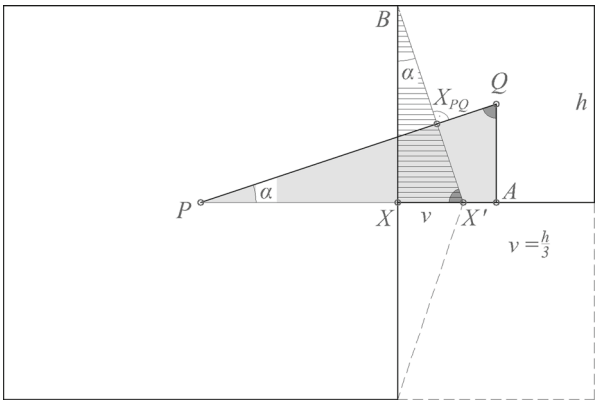


**Fig. 2** Adjustment to the consistent grid. $|XX'| = v = \frac{1}{3}h$. XB=$\frac{2}{3}$PA, hence $\frac{BX'}{PQ} = \frac{2}{3}$

segment - an edge $\sigma_{pq}$ with a nonzero measure in $I\!R$ denoted by $|\sigma_{pq}|$. For $q \in N_p$, $d_{pq} = |X_q - X_p|$ and $\mathbf{n}_{pq}$ is a unit normal vector outward to $p$. By $\mathcal{E}_p$ we denote a set of all edges $\sigma$ of $p$. When we speak about a unit normal vector to $\sigma \in \mathcal{E}_p$, we denote it by $\mathbf{n}_{p\sigma}$.

Let us denote by $X_\sigma$ such a point of $\sigma_{pq}$ which represents the intersection of the line segment $X_pX_q$ and $\sigma_{pq}$. In our consistent grid, $X_pX_q$ is perpendicular to $\sigma_{pq}$, but the intersection $X_\sigma$ is not the midpoint of $\sigma$ in general case. Let us denote by $X^*_\sigma$ the **midpoint** of the edge $\sigma$.

## 3 Numerical scheme for the linear heat equation on a consistent adaptive grid

We solve the following problem:

$$\frac{\partial u(x,y,t)}{\partial t} - \Delta u(x,y,t) = r(x,y,t), \quad (x,y,t) \in Q_T \equiv \Omega \times I, \tag{1}$$

$$\frac{\partial u(x,y,t)}{\partial n} = 0, \quad (x,y,t) \in \partial\Omega \times I, \tag{2}$$

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

$$u_0 \in L_\infty(\Omega) \quad , \quad r(x,y,t) \in L_\infty(Q_T). \tag{4}$$

Here, $u(x,t)$ is an unknown function defined in $\Omega \subset R^2$ and a time interval $I = [0,T]$, $\mathbf{n}$ is a unit outer normal vector to $\partial\Omega$ and $u^0(x)$ is an initial condition. Having the grid, we integrate the diffusion equation over a finite volume $p$ and use the divergence theorem to obtain

$$\int_p \partial_t u \, dX - \int_{\partial p} \nabla u.\mathbf{n}_p \, dS = \int_p r \, dX. \tag{5}$$

We replace the time derivative by a finite difference using the uniform time step $\tau = t^n - t^{n-1}$, where $t^{n-1}$, $t^n$ are previous and current time steps. Let $u^n$ be the solution in the $n^{th}$ time step, $u_p^n$ denotes the solution over the finite volume $p$ in the $n^{th}$ time step and $r_p^n = \frac{1}{\tau|p|} \int\limits_{(n-1)\tau}^{n\tau} \int\limits_p r(x, y, t) dX dt$. Having the integral form (5) of the problem (1), let us denote by

$$f_{pq}^n = \int\limits_{\sigma_{pq}} \nabla u^n . \mathbf{n}_{pq} \, ds \tag{6}$$

the flux through boundary $\sigma_{pq}$ between $p$ and its neighbor $q$ at the time step $t^n$. Then the implicit scheme can be written in the following general form

$$\left(u_p^n - u_p^{n-1}\right) |p| = \tau \sum_{q \in N_p} f_{pq}^n + \tau |p| r_p^n. \tag{7}$$

The flux $f_{pq}^n$ contains a normal derivative of a solution at the time step $t^n$ evaluated on the boundary $\sigma_{pq}$ and we approximate it numerically by:

$$\nabla u^n \cdot \mathbf{n}_{pq} \approx \frac{\left(u_q^n - u_p^n\right)}{d_{pq}}. \tag{8}$$

Let us denote by $T_{pq}$ the term $\frac{|\sigma_{pq}|}{d_{pq}}$ and call it the transmissivity coefficient. Now the flux can be approximated by

$$f_{pq}^n \approx F_{pq} = T_{pq} \left(u_q^n - u_p^n\right), \tag{9}$$

where for the consistent adaptive grid $T_{pq}$ is defined as follows:

- for $q$ of the different size as $p$ (in the original quadtree grid), $T_{pq} = \frac{2}{3}$.
- for $q$ of the same size as $p$ (in the original quadtree grid): if $p$ and $q$ have the same parent in the quadtree structure and a common larger neighbor, $T_{pq} = \frac{2}{3}$ (the length of such a common edge is reduced, see e.g. Fig. 1). Otherwise $T_{pq} = 1$. In other words, if one edgepoint of the common edge is a hanging node in the original quadtree then $T_{pq} = \frac{2}{3}$, otherwise $T_{pq} = 1$.

Now we are ready to write the **implicit finite volume scheme** for solving problem (1)–(3):

Let $0 = t^0 \leq t^1 \leq \ldots \leq t^N = T$ denote the time discretization with $t^n = t^{n-1} + \tau$, where $\tau$ is the time step. For $n = 1, \ldots, N$ we look for $u_p^n$, $p \in \mathcal{T}_h$ satisfying

$$\left(u_p^n - u_p^{n-1}\right) |p| = \tau \sum_{q \in N_p} T_{pq} \left(u_q^n - u_p^n\right) + \tau |p| r_p^n \tag{10}$$

or rewritten

$$\left(\frac{|p|}{\tau}+\sum_{q\in N_p} T_{pq}\right)u_p^n-\sum_{q\in N_p} T_{pq}u_q^n=\frac{|p|}{\tau}u_p^{n-1}+|p|r_p^n; \tag{11}$$

with

$$u_p^0=\frac{1}{|p|}\int_p u_0(x)dx;\ \text{for all } p\in\mathcal{T}_h. \tag{12}$$

After standard application of Neumann boundary conditions, this scheme leads to a linear system of equations at every time step, which can be efficiently solved e.g. by the SOR method.

*Remark 1* Note that our consistent grid fulfils all conditions for admissible mesh defined in [11], Definition 10.1. Let us denote further

$$\text{size}(\mathcal{T}_h)=\sup\{\text{diam}(p),\ p\in\mathcal{T}_h\}.$$

**Theorem 1** *Let $u_h^n$ be on each finite volume $p\in\mathcal{T}_h$ a piecewise constant numerical solution given by numerical scheme (11)–(12). Then*

1. *There exists unique solution $u_h^n$ of the scheme (11)–(12) for every $n=1,...,N$.*
2. *This solution is $L_\infty$ stable.*
3. *Let solution of (1) be a function $u\in C^2(\bar{\Omega}\times R_+)$, $u_0\in C^2(\bar{\Omega})$, $r\in C^0(\bar{\Omega}\times R_+)$, $\mathcal{T}_h$ be an admissible mesh and $\tau\in(0,T)$. Let $e_p^n=u(x_p,y_p,t_n)-u_p^n$ for $p\in\mathcal{T}_h$ and $n=1,2,\ldots N$. Then there exists $C$ only depending on $u,T,\Omega$ and $r$ such that*

$$\left(\sum_{p\in\mathcal{T}}(e_p^n)^2|p|\right)^{\frac{1}{2}}\le C\,(size(\mathcal{T}_h)+\tau)\,,$$

*for all $n=1,2,\ldots N$.*

*Proof* 1. Note that resulting linear algebraic system of equations (11) has diagonally dominant M-matrix which implies existence of the unique solution for each $n=1,...,N$.
2. $L_\infty$ stability can be proved in a similar way as in [11] Lemma 18.1.
3. The error estimates are proven in [11] Theorem 17.1 for the non-homogeneous Dirichlet boundary condition. The result is valid for zero Neumann boundary condition as well.

□

The overall **adaptive algorithm** to solve (1)–(3) has three phases in every time step:

1. We build the graded quadtree adaptive grid and change it to a consistent one.

2. We compute the coefficients of the linear system during recursive traversal of the quadtree.
3. We solve the linear system (10).

### 3.1 EOC for the adaptive scheme on the consistent grid

For numerical tests, let us consider the linear diffusion equation with a right hand side

$$u_t - \Delta u = r, \tag{13}$$

where $u = u(x, y, t)$, $r = r(x, y, t)$, $(x, y) \in (0, 1) \times (0, 1) = \Omega$ and $t \in [T_1, T_2]$. For the function $r(x, y, t) = \cos(2\pi x)\cos(2\pi y)(1 + 8\pi^2 t)$, the exact solution is given by $u(x, y, t) = \cos(2\pi x)\cos(2\pi y)t$ and we solve the problem in a time interval $I = [0.5, 0.6]$. We compute $L_2(I, L_2(\Omega))$ norm of the error by the formula

$$E(h) = \sqrt{\sum_{n=1}^{N} \tau \sum_{p} (u(x_p, y_p, t^n) - u_p^n)^2 |p|}. \tag{14}$$

The initial uniform grid is of the size $16 \times 16$ $\left(h = \frac{1}{16}\right)$. Over this grid we construct two initial nonuniform quadtree grids depicted in Fig. 3 and perform the experiments. We take the initial nonuniform adaptive grid and solve the linear heat equation on this grid adjusted to the consistent one, for a given time interval $I$. Then we refine the initial quadtree adaptive grid by dividing every finite volume into four subvolumes, adjust it to the consistent one and solve the linear heat equation again (see Fig. 4). The experimental order of convergence is then evaluated by $EOC = log_2 \frac{E(h)}{E(h+1)}$. Let us note that the grid is not changing in time, though we solve the time dependent problem. To study EOC, we refine each grid three times. We performed $N$ time steps (the third column of Table 1) for each grid and evaluated the error given by (14). The errors and EOCs are shown in Table 1. As one can see, the method is the second order accurate.

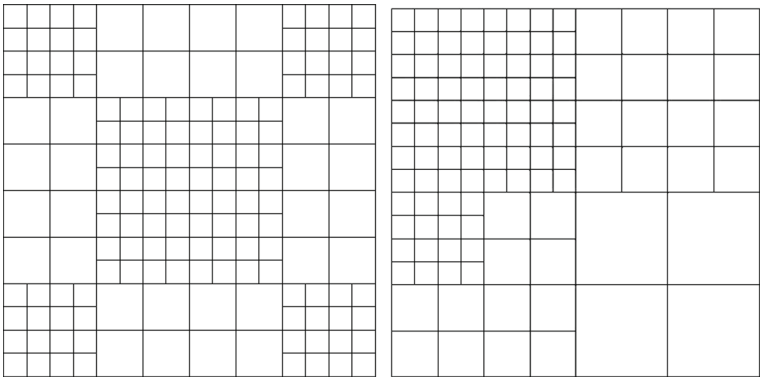

**Fig. 3** The initial coarse quadtree grids for EOC computation, *left*: Grid **a**, *right*: Grid **b**

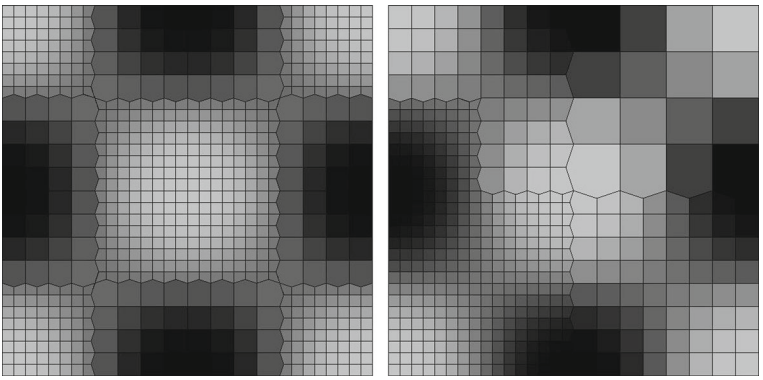

**Fig. 4** *Left*: the consistent grid with $h = \frac{1}{32}$ obtained by refinement of the grid **a** and the corresponding intensity function. *Right*: the consistent grid with $h = \frac{1}{32}$ obtained by refinement of the grid **b** and the corresponding intensity function. See also the second row of the Table 1

### 3.2 Computational complexity of the adaptive scheme

In this section we compare the complexity of the adaptive and nonadaptive algorithms for the linear heat equation. While in the nonadaptine algorithm working on regular square or rectangular grids there are two main phases; the evaluation of coefficients of the linear system and solving the linear system, the adaptive algorithm has an additional step - building the grid.

Let the quadtree be built over an image with $M = 2^N \times 2^N$ pixels and it has $m$ leaves corresponding to elements of the computational grid. Then we can say that the computational grid is constructed in $O(M)$ time, the coefficients of the linear system are evaluated in $O(m)$ time and the linear system is solved in $O(m)$ time (in [16], the computational complexity was discussed in more details).

We have performed an experiment exploring the time behaviour with increasing $M$, i.e., the adaptive grid was subsequently formed over images with the size $2^N \times 2^N$, $N = 8, \ldots, 11$. We have created a double-valued image of an object formed in such

**Table 1** EOC calculations, from the left: $h$ means the size of finite volumes in the initial grid if there would be no coarsening, $\tau$ is the time step, $N$ is the number of time steps performed over $I = [0.5, 0.6]$

| | | | Grid a) | Grid a) | Grid b | Grid b) |
|---|---|---|---|---|---|---|
| $h$ | $\tau$ | $N$ | $E(h)$ | $EOC$ | $E(h)$ | $EOC$ |
| $\frac{1}{16}$ | 0.003906 | 25 | 0.0188 | | 0.01302 | |
| $\frac{1}{32}$ | 0.000977 | 102 | 0.00465 | 2.015 | 0.003459 | 1.912 |
| $\frac{1}{64}$ | 0.000244 | 409 | 0.001154 | 2.010 | 0.00089 | 1.958 |
| $\frac{1}{128}$ | 0.000061 | 1638 | 0.000288 | 2.002 | 0.000109 | 1.996 |

Then the errors $E(h)$ and $EOC$ for both Grid a) and Grid b) are given

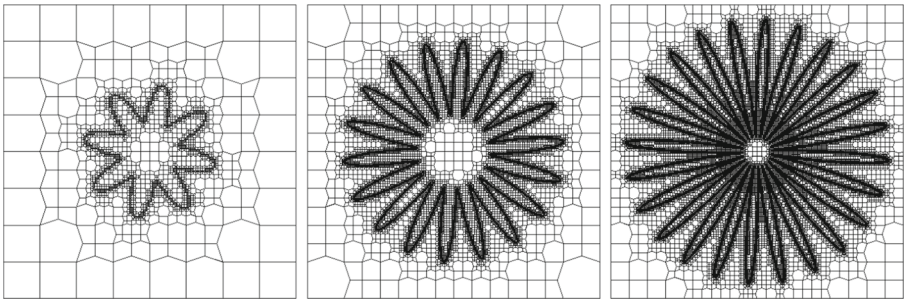

**Fig. 5** Examples of adaptive grids built on various refinement levels. The number of elements in the adaptive grid is about five percent of the original uniform grid

a way that the number of elements in an adaptive grid is about 5 % of the original regular grid (examples of such grids are in Fig. 5). In the experiment we explore the running times of particular phases of both nonadaptive and adaptive algorithms for increasing values of $N$. The results are displayed in Table 2 for regular grids and Table 3 for corresponding adaptive grids. The tables show linear scaling of both algorithms with respect to number of elements in the grid.

In the next experiment we take the grid on the level $N = 11$ with 223003 elements reduced from 4 194298 and we estimated the effectiveness of the algorithm by measuring the times of particular operations.

By measurements we get that an average cost to get and to store the coefficients of one equation of the linear system is 5.6 times higher in the adaptive grid than in the regular grid and for decrease above 16 % the coefficients are more efficiently evaluated on a regular grid (we must also realize that in a regular grid the elements are processed in loops and in our implementation of an adaptive algorithm with help of recursive procedures.) Usually we can save computational time in this phase only in the last steps of a time evolution, but mostly the dominant time is the time to solve the linear system.

The coefficients of a linear system are stored in a list and then the linear system is solved in simple loops without a help of a recursion. The semi-implicit methods we

**Table 2** Running times for particular phases of one time step of the **nonadaptive** algorithm for images of the size $2^N \times 2^N$

| $N$ | Number of elements | Time for LS coefficient | Time for linear system | Time for coeff+LS |
|---|---|---|---|---|
| 8 | 65536 | 0.016 | 0.1872(16it) | 0.2032 |
| 9 | 262144 | 0.03 | 0.531(13it) | 0.561 |
| 10 | 1048576 | 0.125 | 2.14(14it) | 2.265 |
| 11 | 4194298 | 0.517 | 8.56(14it) | 9.077 |

**Table 3** Running times for particular phases of one time step of the **adaptive** algorithm for the initial images of the size $2^N \times 2^N$

| N | Number of elements | % of orig. elements | Time to build a grid | Time for LS coeff. | Time for lin. system | Grid+ coeff+LS |
|---|---|---|---|---|---|---|
| 8 | 3277 | 5 % | 0.015 | – | 0.015(16it) | 0.03 |
| 9 | 13399 | 5.11 % | 0.047 | 0.016 | 0.031 (11it) | 0.094 |
| 10 | 53572 | 5.1 % | 0.125 | 0.046 | 0.125 (10it) | 0.296 |
| 11 | 223003 | 5.32 % | 0.547 | 0.156 | 0.5 (10it) | 1.203 |

use lead to linear systems with number of unknowns corresponding to the number of grid elements. However the efficiency of the linear system solution does not depend on the decrease of elements directly, because in one equation of the adaptive method the number of elements depends on a number of neighbors of a processed (diagonal) element and the total number of neighbors depends on a structure of a grid. Moreover, the number of iterations of a linear system depends also on the solution itself, on the stopping criterion (in all performed experiments it is the same), on the linear system matrix depending e.g. on the scale step size, the size of element (the higher, the better diagonal dominance is) and if we use SOR, the suitable choice of a relaxation parameter.

As to the stopping criterion, the absolute error of a residual $||Ax - b|| < tol$ is used with the same value of $tol$ for all refined grids. It is known that when solving an elliptic equation the number of iterations is increasing with the refinement level. On the other hand, when solving the parabolic equations, as in our case, the number of iterations does not increase when refining the grid. Let us note that we start the SOR iterations at the current time step by using numerical solution from the previous time level. Due to the standard parabolic space-time coupling ($\tau = h^2$), which means that halving the space step we refine the time step four times, the initial iteration (solution from the previous time step) at a fine grid is closer to the final solution of the linear system than it is on the one level coarser grid. In our opinion this is the reason why the number of SOR iterations remains stable when refining the grid.

Our experience is that with the grid with larger elements the number of iterations is lower, so at the end we can say that the time is proportional to decrease of elements and with a sufficient decrease of elements the adaptive algorithm is much faster in the final scale steps. Using Table 3 we estimate an average cost incurred by processing of one equation in the adaptive algorithm (on every level we divide the time (the 7th column) by a number of equations times a number of iterations (the 2nd and 6th columns). If we perform the same evaluation of the cost for processing one equation for the nonadaptive grid (Table 2) we can see that the cost in the adaptive grid is 1.12 times higher than in the nonadaptive one. For a number of iterations we use a linear interpolation between 5 % of elements (10 iterations,the 6th column of Table 2) and 100 % of elements (14 iterations, the 4th column of Table 3) we can estimate CPU times of an adaptive method for various decrease level of elements expressed as percentage of the original number of elements. These estimates are presented in

Fig. 6. At the top there is a contribution of particular phases while at the bottom we have a comparison of estimated overall CPU times.

## 4 Approximation of the gradient on the consistent grid

To solve more general diffusion equations we need to approximate the gradients over the finite volumes and their boundaries. In the adaptive methods for the Perona-Malik

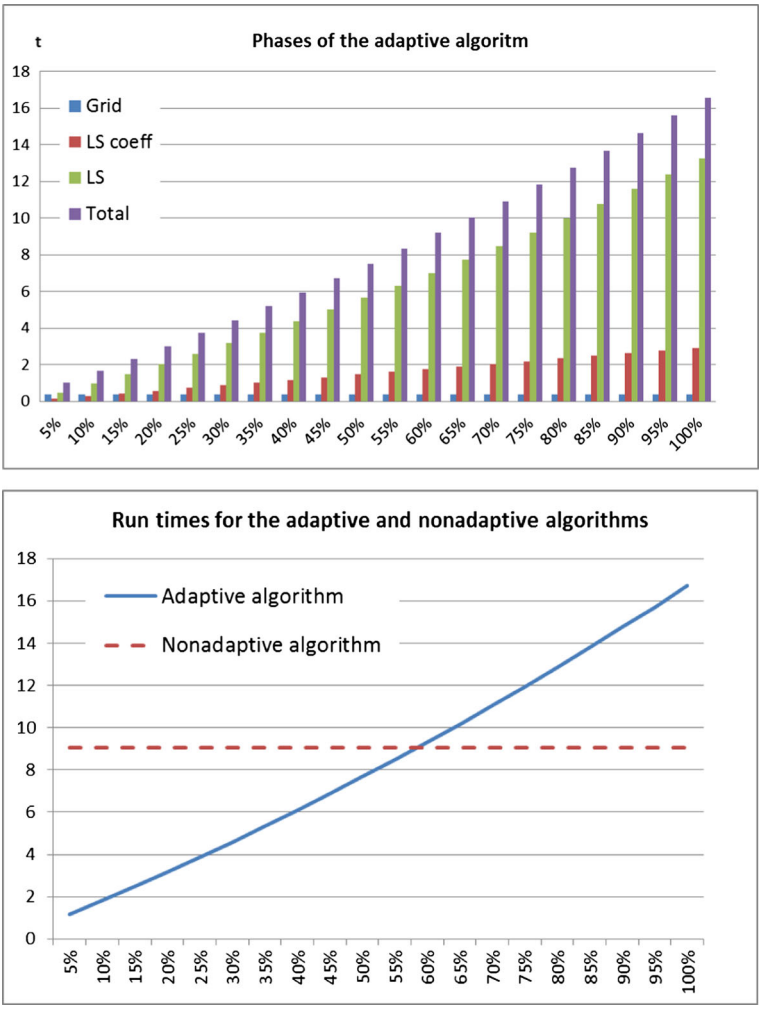


**Fig. 6** *Top*: Estimate of run time for three phases of the adaptive algorithm). *Bottom*: The running time estimates (axis y) for a decrease of elements (expressed in a percentage of the original image (axis x) for nonadaptive and adaptive algorithms. The original number of elements is 4194298. The times are given in seconds

equations [17, 18], both working on quadtree grids, the (presmoothed) gradients are evaluated on boundaries of the finite volumes, either convolving the solution with the partial derivatives of the smoothing kernel [17] or using the diamond cell method requiring solution approximations in the corners of the squares obtained by bilinear approximation. With the shapes occurring in our consistent adaptive grid too many possibilities would be tested and the algorithms hardly can be efficient. That is the reason why we have chosen the method based on [12] where we consider also representative points on finite volume edges, but not values at the corners. Then, with a help of these points we work locally, we only need access to neighbors sharing a common edge and update values in these points with a help of conservation principle.

The method from [12] works in this way:

1. for edges $\sigma$ of finite volume $p$ we define representative points $X_\sigma^*$ - their midpoints,
2. with help of these points, we evaluate the norm of gradient locally on $p$ (using the consequence of the Stokes formula, see (19)–(20)),
3. equation for the finite volume $p$ is derived locally,
4. values of solution in $X_\sigma^*$ are obtained by using the conservation principle.

In the consistent adaptive grid, if we set $X_\sigma$ to be the intersection of the connection of representative points $X_p$ and $X_q$ and the edge $\sigma_{pq}$, the line segments $X_pX_\sigma$ and $X_qX_\sigma$ are perpendicular to $\sigma$, but $X_\sigma$ is not the midpoint of $\sigma$ on every grid element. Such a situation occurs on edges containing a hanging node in the original quadtree grid. The most critical shape in this sense is the sharp element, where $X_\sigma$ is not the midpoint on any of the edges (Fig. 7 right).

Let us suppose the linear approximation of the solution over the finite volume $p$. Any linear function can be written as

$$u(X) = u(X_p) + \nabla u \cdot (X - X_p) = u_p + \nabla u \cdot (X - X_p). \tag{15}$$

If $X = X_\sigma$, it holds

$$u_\sigma - u_p = \nabla u \cdot (X_\sigma - X_p), \tag{16}$$

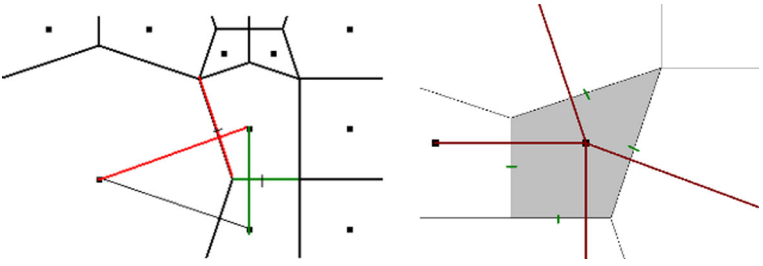

**Fig. 7** Examples of the shapes where the intersection of the connection of representative points and a common edge $\sigma$ is not the midpoint of $\sigma$

where $u_\sigma$ is a value of the solution at a point $X_\sigma$. The gradient of the linear function is a constant vector in $\mathbb{R}^2$, thus also over a control volume $p$. Then it holds

$$\nabla u = \frac{1}{|p|}\int_p \nabla u dx = \frac{1}{|p|}\int_{\partial p} u\mathbf{n}_{p\sigma} dS \tag{17}$$

$$= \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}\int_\sigma (u_p + \nabla u\cdot(X - X_p))\mathbf{n}_{p\sigma} dS \tag{18}$$

$$= \frac{1}{|p|}u_p\sum_{\sigma\in\mathcal{E}_p}|\sigma|\mathbf{n}_{p\sigma} + \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|\nabla u\cdot(X^*_\sigma - X_p)\mathbf{n}_{p\sigma}. \tag{19}$$

The term $\sum_{\sigma\in\mathcal{E}_p}|\sigma|\mathbf{n}_{p\sigma} = \mathbf{0}$ and the expression $|\sigma|\nabla u(X^*_\sigma - X_p)\mathbf{n}_{p\sigma}$ represents the precise integration of a linear function over the edge $\sigma$. Thus we have

$$\nabla u = \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|\nabla u\cdot(X^*_\sigma - X_p)\mathbf{n}_{p\sigma}. \tag{20}$$

On the edges, where $X_\sigma \neq X^*_\sigma$, we can express

$$X^*_\sigma - X_p = (X_\sigma - X_p) + (X^*_\sigma - X_\sigma). \tag{21}$$

The second part of the right hand side of (21) will be called correction. Then $\nabla u$ can be split into two parts

$$\nabla u = \underbrace{\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|\nabla u\cdot(X_\sigma - X_p)\mathbf{n}_{p\sigma}}_{=A} + \underbrace{\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|\nabla u\cdot(X^*_\sigma - X_\sigma)\mathbf{n}_{p\sigma}}_{=B} \tag{22}$$

The part of $\nabla u$ given by $A$ and denoted as $(\nabla u)^A$ represents the gradient without correction and due to (16) it can be evaluated as

$$(\nabla u)^A = \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|(u_\sigma - u_p)\mathbf{n}_{p\sigma}. \tag{23}$$

The second term $B$ - the correction of $(\nabla u)^A$, depends on the unknown gradient. The practical evaluation will be explained later

In the following text we present three experiments performed on a specific consistent grid (see Fig. 10, the grid is built over a uniform grid with $32 \times 32$ elements) and the specific function $u(x, y)$, defined on $< -1.25, 1.25 > \times < -1.25, 1.25 >$,

on which we evaluate the norm of the gradient on $p$:

$$u(x, y) = \frac{1}{2}\left(x^2 + y^2\right). \tag{24}$$

**Experiment 1** We evaluate the norm of the gradient on $p$ in its representative point analytically, i.e. $\nabla u(\mathbf{X_p}) = (x_p, y_p)$, $|\nabla u(\mathbf{X_p})| = \sqrt{{x_p}^2 + {y_p}^2}$. The result is graphed in Fig. 8 left.

**Experiment 2** We evaluate the approximated gradient using only $A$ and neglecting $B$ in (22). The values $u_\sigma$ were obtained by a linear interpolation. The result is shown in Fig. 8 right. The errors, significant on sharp elements with latrge gradients, indicates a necessity for including the part $B$ into computations.

### 4.1 Evaluation of the gradients with corrections

*Remark* In the following text, we use subscripts in two ways: if they represent derivatives, we use $x$ resp. $y$ and if they represent the vector components, we use 1 resp. 2.

Let us denote the correction vector $(X^*_\sigma - X_\sigma)$ by $\mathbf{c}_\sigma = ((c_\sigma)_1, (c_\sigma)_2)$. We will work with $(\nabla u)^A = ((u_x)^A, (u_y)^A)$, $\mathbf{n_{p\sigma}} = ((n_{p\sigma})_1, (n_{p\sigma})_2)$ and the unknown vector $\nabla u = (u_x, u_y)$. Now we have

$$\nabla u = (\nabla u)^A + \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p} |\sigma|(\nabla u \cdot \mathbf{c}_\sigma)\mathbf{n}_{p\sigma}, \tag{25}$$

i.e.,

$$(u_x, u_y) = \left((u_x)^A, (u_y)^A\right) + \frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p} |\sigma|((c_\sigma)_1 u_x + (c_\sigma)_2 u_y)((n_{p\sigma})_1, (n_{p\sigma})_2). \tag{26}$$

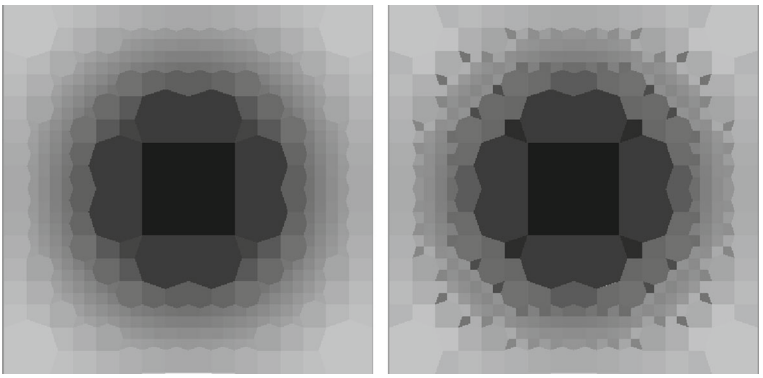

**Fig. 8** *Left*: Experiment 1. - the norm of gradient evaluated analytically. *Right*: Experiment 2. Omitting part B in (22) leads to the discontinuities in the norms of gradients

We see that (26) represents a linear system of 2 equations with 2 unknowns $u_x$ and $u_y$, which can be adjusted to the following form:

$$\begin{aligned} u_x\left(1-\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|(c_\sigma)_1(n_{p\sigma})_1\right) &+ u_y\left(-\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|(c_\sigma)_2(n_{p\sigma})_1\right) = (u_x)^A, \\ u_x\left(-\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|(c_\sigma)_1(n_{p\sigma})_2\right) &+ u_y\left(1-\frac{1}{|p|}\sum_{\sigma\in\mathcal{E}_p}|\sigma|(c_\sigma)_2(n_{p\sigma})_2\right) = (u_y)^A. \end{aligned} \tag{27}$$

The elements of the coefficient matrix of (27) can be evaluated procedurally, traversing the quadtree and the system can be solved e.g. by the Cramer rule.

### 4.2 Practical evaluation of gradients

For a particular element shape, the matrix $B$ can be precalculated in advance. Let us denote:

$$\mathbf{N}_{p\sigma} = \frac{|\sigma|\mathbf{n}_{p\sigma}}{l} \text{ and } \mathbf{C}_\sigma = \frac{\mathbf{c}_\sigma}{l},$$

where $l$ is the length of the square edge in the original quadtree. To evaluate these vectors we start with the edge vector $\vec{\sigma}$ (oriented from the shifted node to the quadtree corner), $\mathbf{N}_{p\sigma}$ is obtained by its rotation by 90 or $-90$ degrees (i.e. by switching its coordinates and negating one of them) to create the outer normal and extracting $l$ - the length of the square in the original quadtree. $\mathbf{C}_\sigma$ is parallel to $\vec{\sigma}$, but shortened according to the geometrical situation and $l$ is extracted. The nonzero corrections occur only if one of the edgepoints of $\sigma$ is the shifted node. It can be shown that on the aligned edge $\sigma$, the correction $\mathbf{c}_\sigma$ can be expressed as $\mathbf{c}_\sigma = \frac{\vec{\sigma}}{10}$, on the vertical or horizontal edge $\mathbf{c}_\sigma = \frac{\vec{\sigma}}{4}$. For example, on a shape like in Fig. 9c), we have $\vec{\sigma} = \left(\frac{1}{3}h, h\right)$, $\mathbf{N}_{p\sigma} = \left(1, -\frac{1}{3}\right)$, $\mathbf{C}_\sigma = \left(\frac{1}{30}, \frac{1}{10}\right)$ for an aligned edge and $\vec{\sigma} = \left(-\frac{2}{3}h, 0\right)$, $\mathbf{N}_{p\sigma} = \left(0, -\frac{2}{3}\right)$, $\mathbf{C}_\sigma = \left(-\frac{1}{6}, 0\right)$ for a horizontal edge. In all other configurations of this shape we just switch or negate these coefficients.

The system can be written in the following form:

$$\begin{aligned} u_x\overbrace{\left(1-\frac{l^2}{|p|}\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)}^{b_{11}} &+ u_y\overbrace{\left(-\frac{l^2}{|p|}\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1\right)}^{b_{12}} = (u_x)^A, \\ u_x\overbrace{\left(-\frac{l^2}{|p|}\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_2\right)}^{b_{21}} &+ u_y\overbrace{\left(1-\frac{l^2}{|p|}\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_2\right)}^{b_{22}} = (u_y)^A. \end{aligned} \tag{28}$$

The coefficients of the linear system (28) depend on $\mathbf{N}_{p\sigma}$, $\mathbf{C}_\sigma$ and the ratio $\frac{l^2}{|p|}$. All of them are determined only by the shape of element $p$, not by its level (size), so we can store the inverse of the coefficient matrix $B$ for every shape and evaluate $\nabla u = B^{-1}(\nabla u)^A$. This approach is suitable, if there is only limited number of shapes in the polygonal grid like in our consistent adaptive grid and if we work on a fixed grid like in Experiments 4 and 5.

*Example 1* Let us show the coefficient matrix $B$ of the linear system (28) for some configurations of elements (cf. Fig. 9):

a) If an element $p$ has a couple of hanging nodes on neighboring edges in the original quadtree (i.e. on edges sharing a common vertex), or on all of them, the effects of corrections cancel mutually and $B$ is the unit matrix.
b) If the hanging node is on one or three edges of $p$, then

$$B = \begin{pmatrix} 0.985 & 0 \\ 0 & 1.015 \end{pmatrix}$$

c) If $p$ has one larger neighbor on its edge, then $B$ is the matrix

$$B = \begin{pmatrix} 0.96 & -0.12 \\ -0.12 & 1.04 \end{pmatrix}$$

d) In the case of the sharp element, the coefficient matrix is

$$B = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}.$$

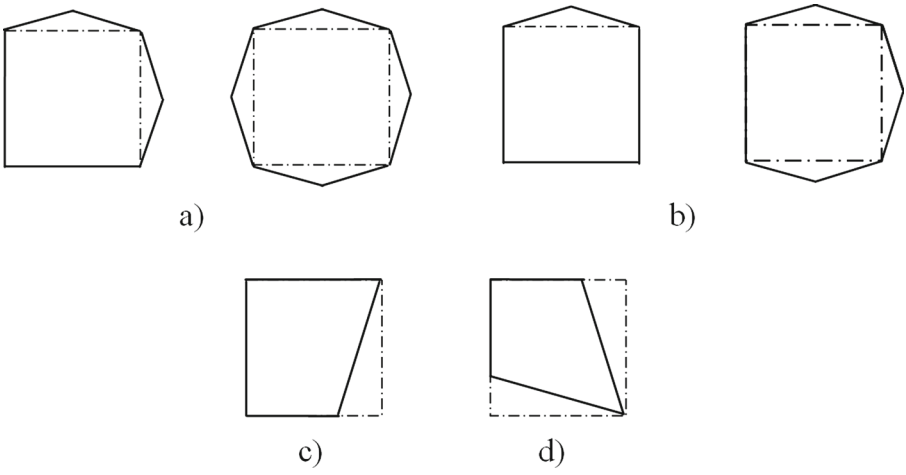


**Fig. 9** Example 1. Shapes for particular cases of Example 1

**Experiment 3** We take the function $u(x, y) = \frac{1}{2}(x^2 + y^2)$ and the grid from the Experiment 1 (Fig. 10) and compare the gradients $(\nabla u)^A$ and $\nabla u$ for an element far from the center. We take the sharp element highlighted in red with the representative point $(x_p, y_p) = (0.742, -0.89)$. First $u_\sigma$ is set to the exact value evaluated using $u(x, y)$. The gradient vector $(\nabla u)^A$ evaluated without correction with a value $(-1.711, -1.801)$ was corrected using (28) to $(0.860, -1.03)$.

In practical tasks $u_\sigma$ is obtained by interpolation, here the values of $u_\sigma$ (like in the Experiment 2) are obtained by linear interpolation between $u_p$ and $u_q$ of its neighbor. It is interesting that in such a case the gradient of $u(x, y)$ obtained by (28) is equal to the analytical gradient. On a square element, this fact can be easily verified. For an element with a nonzero correction, let us take, for example, the element with the same shape like the selected element ( i.e. the element depicted in Fig. 9d) with its matrix $B$ shown in Example 1 d)). We get, using (23)

$$\left((u_x)^A, (u_y)^A\right) = \left(x_p + 0.3y_p, y_p + 0.3x_p\right).$$

After correcting this gradient by multiplying by $B^{-1}$, we have:

$$\frac{1}{1-0.3^2}\begin{pmatrix} 1 & -0.3 \\ -0.3 & 1 \end{pmatrix}\left(x_p + 0.3y_p, y_p + 0.3x_p\right)^T =$$

$$\frac{1}{1-0.3^2}\left(\left(1-0.3^2\right)x_p, \left(1-0.3^2\right)y_p\right)^T = \left(x_p, y_p\right)^T.$$

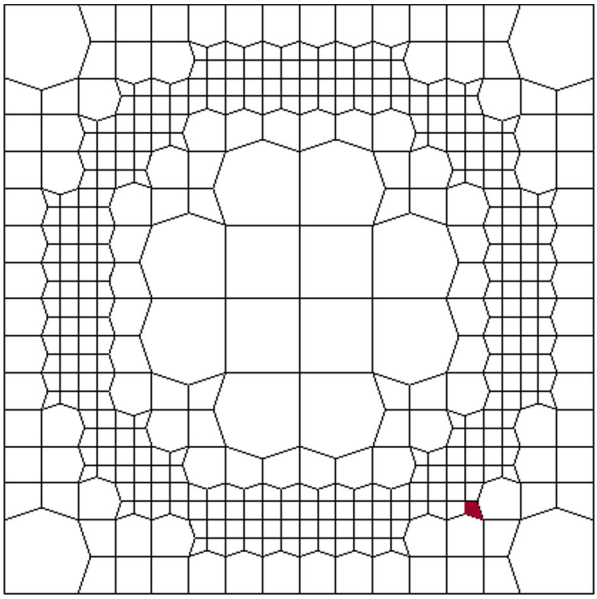

**Fig. 10** The grid used in Experiments 1-3. The selected shape for Experiment 3. The dimension of the original nonadaptive was $32 \times 32$, $h = 0.03125$

Finally, let us take the function $u_2(x, y) = \frac{1}{3}\left(x^3 + y^3\right)$ and the selected shape ($u_\sigma$ is obtained by linear interpolation). The analytical value of the gradient is $(0.551, 0.807)$, using (23) we get $(\nabla u)^A = (0.813, 0.987)$, after correction $\nabla u = (0.5572, 0.813)$.

### 4.3 Properties of the coefficient matrix $B$

In the following we want to show symmetry and regularity properties of the matrix $B$. In the proofs we show some rules which can be helpful during construction of $B$.

**Lemma** *It holds:*

a) $\sum_{\sigma\in\mathcal{E}_p} (C_\sigma)_1 (N_{p\sigma})_1 = -\sum_{\sigma\in\mathcal{E}_p} (C_\sigma)_2 (N_{p\sigma})_2,$

b) $\sum_{\sigma\in\mathcal{E}_p} (C_\sigma)_1 (N_{p\sigma})_2 = \sum_{\sigma\in\mathcal{E}_p} (C_\sigma)_2 (N_{p\sigma})_1.$

*Proof* We explore only edges with nonzero corrections. The first property follows from the fact that $\mathbf{C}_\sigma \perp \mathbf{N}_{p\sigma}$. To show the symmetry property in b), we use the fact that for the parallel codirectional vectors $\mathbf{u}$ and $\mathbf{v}$ it holds $\mathbf{u}\cdot\mathbf{v} = |\mathbf{u}||\mathbf{v}|$ and for the opposite directional vectors $\mathbf{u}$ and $\mathbf{v}$, $\mathbf{u}\cdot\mathbf{v} = -|\mathbf{u}||\mathbf{v}|$. Let $\mathcal{E}_p{}^* = \{\sigma^1, \sigma^2\}$ denote two edges sharing a shifted node * with the corresponding vectors $\mathbf{N^1}$, $\mathbf{C^1}$ and $\mathbf{N^2}$, $\mathbf{C^2}$ (for examples, see Fig. 11). Let us rotate $\mathbf{N^1}$ and $\mathbf{N^2}$ by 90°, we get vectors $\left(N_2^1, -N_1^1\right)$ and $\left(N_2^2, -N_1^2\right)$. Without losing generality, one is codirectional and the

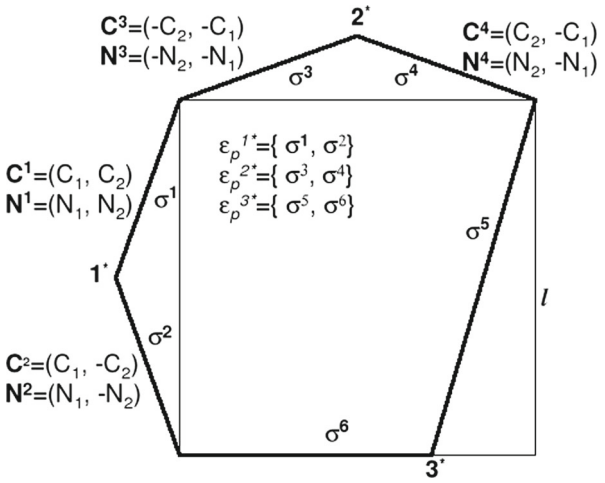


**Fig. 11** Examples of the sets $\mathcal{E}_p^*$. If $\mathcal{E}_p^{1*}$ and $\mathcal{E}_p^{2*}$ correspond to two adjacent edges in the original quadtree, then $\sum_{\sigma\in\mathcal{E}_p^{1*}\cup\mathcal{E}_p^{2*}} (C_\sigma)_1 (N_{p\sigma})_1 = 2C_1N_1 + 2C_2N_2 = 2\mathbf{C^1}\cdot\mathbf{N^1} = 0$, because $\mathbf{C^1}\perp\mathbf{N^1}$

other is opposite directional to $\mathbf{C^1}$ and $\mathbf{C^2}$, respectively. Using the property of the scalar product mentioned above, we have

$$C_1^1 N_2^1 - C_2^1 N_1^1 = |\mathbf{C^1}||\mathbf{N^1}|,$$
$$C_1^2 N_2^2 - C_2^2 N_1^2 = -|\mathbf{C^2}||\mathbf{N^2}|.$$

We sum these two equations and show that the right-hand side of the sum is equal to zero. This implies that b) holds for one couple of edges with nonzero corrections ${\mathcal{E}_p}^*$, thus also for all edges in $\mathcal{E}_p$. To show that $\left(C_1^1 N_2^1 + C_1^2 N_2^2\right) - \left(C_2^1 N_1^1 + C_2^2 N_1^2\right) = 0$, we must show that $|\mathbf{C^1}||\mathbf{N^1}| = |\mathbf{C^2}||\mathbf{N^2}|$. For the elements like in Fig. 9a and b, it is obvious, because the lengths of both edges are the same, so we only need to prove it for shape in the case c) what implies the result for all possible shapes in the consistent grid. Let $\mathbf{N^1}$, $\mathbf{C^1}$ are related to the horizontal edge and $\mathbf{N^2}$, $\mathbf{C^2}$ to the skew one. Since $|\mathbf{C^1}| = \frac{1}{4}|\mathbf{N^1}|$ and $|\mathbf{C^2}| = \frac{1}{10}|\mathbf{N^2}|$ (see section Practical evaluation of gradients), we get

$$|\mathbf{C^1}||\mathbf{N^1}| - |\mathbf{C^2}||\mathbf{N^2}| = \frac{1}{4}|\mathbf{N^1}|^2 - \frac{1}{10}|\mathbf{N^2}|^2 = \frac{1}{4}\left(\frac{2}{3}\right)^2 - \frac{1}{10}\left(\frac{\sqrt{10}}{3}\right)^2 = 0,$$

so the symmetry property is fulfilled for $\sigma \in {\mathcal{E}_p}^*$ and thus also for all $\sigma \in \mathcal{E}_p$. □

**Theorem 3** *The matrix B is regular and the system* (28) *always has a unique solution.*

*Proof* Using the properties of the Lemma, we can write

$$det(B) = 1 - \frac{l^4}{|p|^2}\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)^2 - \frac{l^4}{|p|^2}\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1\right)^2 =$$
$$1 - \frac{l^4}{|p|^2}\left(\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)^2 + \left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1\right)^2\right).$$

We can show that $det(B) > 0$, i.e.:

$$\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)^2 + \left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1\right)^2 < \frac{|p|^2}{l^4}. \tag{29}$$

For a square element, $B$ is the identity matrix. Now let us take the element having only smaller neighbors or neighbors of the same size like in Fig. 9a and b. Like in the proof of the case b) of the previous lemma, let us evaluate the sums in (29) for couples of edges $\mathcal{E}_p^*$ sharing a shifted node. Let us chose some edge and denote $\mathbf{C_{\sigma_1}} = (C_1, C_2)$ and $\mathbf{N_{\sigma_1}} = (N_1, N_2)$. Using the geometrical properties of the element all other $C_i$ and $N_j$ corresponding to small neighbors can be expressed using $C_1, C_2, N_1, N_2$, see Fig. 11 for points 1* and 2*. It is easy to

see that $\sum_{\sigma\in\mathcal{E}_p^*}(C_\sigma)_2(N_{p\sigma})_1 = \sum_{\sigma\in\mathcal{E}_p*}(C_\sigma)_1(N_{p\sigma})_2 = 0$. If $\mathcal{E}_p^{1*}$ and $\mathcal{E}_p^{2*}$ correspond to two shifted nodes $*$ on two adjacent edges in the nonadjusted quadtree (again see $1^*$ and $2^*$ in Fig. 11), then due to orthogonality of $\mathbf{C}_{\sigma\mathbf{1}}$ and $\mathbf{N}_{\sigma\mathbf{1}}$, we have $\sum_{\sigma\in\mathcal{E}_p^{1*}\cup\mathcal{E}_p^{2*}}(C_\sigma)_1(N_{p\sigma})_1 = 2C_1N_1 + 2C_2N_2 = 0$ and

$$\left(\sum_{\sigma\in\mathcal{E}_p^{1*}\cup\mathcal{E}_p^{2*}}(C_\sigma)_1(N_{p\sigma})_1\right)^2 = 0. \tag{30}$$

As a consequence it holds that the sum $\sum(C_\sigma)_1(N_{p\sigma})_1$ is nonzero only for one couple of smaller neighbors or two couples on opposite sides. Without losing generality let us take the edge vector $\sigma^{\mathbf{1}} = (\frac{1}{3}\frac{l}{2}, \frac{l}{2})$ from Fig. 11, where $l$ is the length of edge in the corresponding square and $\mathbf{C} = \frac{\sigma^{\mathbf{1}}}{10l} = \left(\frac{1}{60}, \frac{1}{20}\right)$ and $\mathbf{N} = \left(-\frac{1}{2}, \frac{1}{6}\right)$ are corresponding correction and outer normal. The sum $\sum(C_\sigma)_1(N_{p\sigma})_1$ is equal to $\pm\left(\frac{1}{120} + \frac{1}{120}\right) = \pm\frac{1}{60}$ for one couple (or three couples) of smaller neighbors and to $\pm\frac{1}{30}$ for two couples on opposite sides. So we have

$$\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)^2 + \left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1\right)^2 < \left(\frac{1}{30}\right)^2 < 1 < \frac{|p|^2}{l^4}. \tag{31}$$

□

**Now let us take the element with a single greater neighbor** For an example let us take the case corresponding to Fig. 9c with $\mathcal{E}_p^*$ corresponding to $\mathcal{E}_p^{3*}$ in Fig. 11. Generally, $\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1$ is nonzero only on the skew edge. In our example, on this edge $\sigma^{\mathbf{6}} = l(\sigma_1, \sigma_2) = l(\frac{1}{3}, 1)$, $C = \frac{1}{10}(\sigma_1, \sigma_2) = (\frac{1}{30}, \frac{1}{10})$ and $N = (-\sigma_2, \sigma_1) = (-1, \frac{1}{3})$, thus $C_1N_1 = -\frac{1}{30}$ and to $\pm\left(\frac{1}{10}\sigma_1\sigma_2\right)$ generally. To evaluate the mixed term, because $\sum_{\sigma\in\mathcal{E}_p{}^*}(C_\sigma)_2(N_{p\sigma})_1 = \sum_{\sigma\in\mathcal{E}_p{}^*}(C_\sigma)_1(N_{p\sigma})_2$, we can select that case for which $C_iN_j$ is zero on the horizontal or vertical edge. In our case on the horizontal edge $C_2N_1 = 0$ and we evaluate the expression $C_2N_1$ only on a skew one. We have $|C_2N_1| = \frac{1}{10}$ and finally

$$\left(\frac{1}{30}\right)^2 + \left(\frac{1}{10}\right)^2 < \left(\frac{5}{6}\right)^2, \tag{32}$$

where $\frac{5}{6}$ is the ratio of $|p|$ and $l^2$ for this shape.

If some element has one greater neighbor, then on vertical and horizontal edges it can have one or two couples of smaller neighbors, but only one couple of smaller neighbors contributes to the left hand side of (29) by the same argument as in (30), again see Fig. 11. At the same time the right hand side is enlarged, so (29) holds.

In the case that the element has a couple of greater neighbors (Fig. 9d), due to the diagonal symmetry of the element

$$\left(\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_1(N_{p\sigma})_1\right)^2 = 0 \text{ and } |\sum_{\sigma\in\mathcal{E}_p}(C_\sigma)_2(N_{p\sigma})_1| = \frac{2}{10}, \tag{33}$$

thus (29) is satisfied also on this element.

## 5 Numerical solution of the regularized Perona-Malik equation on the consistent adaptive grid

In this section, we deal with the initial - boundary value problem:

$$\partial_t u - \nabla.(g(|\nabla G_s * u|)\nabla u) = 0, \quad \text{in } Q_T \equiv I \times \Omega, \tag{34}$$

$$\frac{\partial u(x,t)}{\partial n} = 0, \quad \text{in } \partial\Omega \times I, \tag{35}$$

$$u(x,0) = u^0(x), \quad x \in \Omega. \tag{36}$$

where $\Omega \subset I\!R^2$ is a rectangular domain, $I = [0, T]$ is a scaling interval, and

$g(v)$ is a decreasing smooth function,

$$g(0) = 1, 0 < g(v) \to 0 \text{ for } v \to \infty, \tag{37}$$

$$G_s \in C^\infty(I\!R^2) \text{ is a smoothing kernel with } \int_{I\!R^2} G_s(x)dx = 1 \tag{38}$$

and $G_s(x) \to \delta_x$ for $\sigma \to 0$, $\delta_x$ - Dirac function at point $x$,

$$u^0 \in L_2(\Omega). \tag{39}$$

Let us use our consistent adaptive grid $\mathcal{T}$ for the space discretization in the finite volume method. Let us denote by $u_\sigma^n$ the value of solution in $X_\sigma$ at the time step $t^n$ and by $d_{p\sigma}$ the distance of representative point $X_p$ from $X_\sigma$. Because the line segment $X_pX_\sigma \perp \sigma$ for all $\sigma \in \mathcal{E}_p$ for all $p \in \mathcal{T}$ the derivative in the direction $\mathbf{n}_{p\sigma}$ can be approximated by

$$\nabla u^n \cdot \mathbf{n}_{p\sigma} \approx \frac{\left(u_\sigma^n - u_p^n\right)}{d_{p\sigma}}. \tag{40}$$

The diffusion coefficient denoted by $g_p^{s,n-1}$ is constant all over $p$ (it is explained in more details later), thus the flux over $\sigma \in \mathcal{E}_p$ can be approximated by

$$f_{p\sigma}^n \approx F_{p\sigma}^n = g_p^{s,n-1}\frac{|\sigma|}{d_{p\sigma}}\left(u_\sigma^n - u_p^n\right). \tag{41}$$

The semi-implicit scheme using the local values can be written in the following general form

$$\left(u_p^n - u_p^{n-1}\right) |p| = \tau \sum_{\sigma \in \mathcal{E}_p} F_{p\sigma}^n. \tag{42}$$

To get the diffusion coefficient $g_p^{s,n-1}$, in the regularized Perona-Malik equation we evaluate the gradient of the presmoothed solution $u^{n-1}$ from the previous time step. With a help of the system (28) we evaluate the gradient using the values of $u_\sigma$ for all $\sigma$ of $p$. To perform presmoothing we can use the property of convolution with respect to the derivative, i.e.

$$|\nabla G_s * u^{n-1}| = |G_s * \nabla u^{n-1}|,$$

In our algorithm we realize the convolution by applying several steps of the adaptive explicit scheme for time corresponding to $s$ to both $x$ and $y$ coordinates of the gradients, then we evaluate the norm of the gradients and apply the Perona-Malik function $g$.

### 5.1 Other geometrical properties of the consistent grid

Though the term $\frac{|\sigma|}{d_{p\sigma}}$ can be evaluated locally, the better way to evaluate it is to consider the neighbor $q$ sharing $\sigma$ with $p$. From now, the edge between $p$ and its neighbor $q$ will be denoted by $\sigma_{pq}$. Then we can express (41) with a help of the transmitivity coefficient $T_{pq} = \frac{|\sigma_{pq}|}{d_{pq}}$ and the ratio of $d_{p\sigma}$ and $d_{pq}$. If $\sigma_{pq}$ is a horizontal or vertical edge the ratio of $d_{p\sigma}$ and $d_{pq}$ is equal to 1:2, otherwise, the situation corresponds to the configuration in Fig. 12: we see a triple of similar triangles, two grey ones and their union $PVQ$, adjacent sides of $PVQ$ being the half diagonals of quadtree

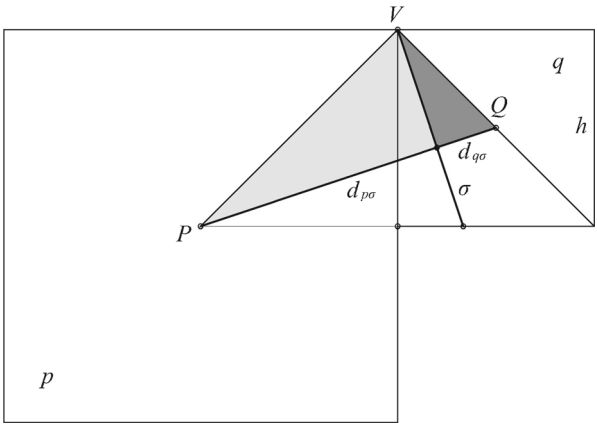


**Fig. 12** *Left*: Geometrical properties of the consistent grid: the *filled triangles* and the *triangle* PVQ are similar, ratio of their adjacent sides is 2:1, because adjacent sides of PVQ are half diagonals of squares with the same ratio of side lengths. It follows that $\frac{d_{p\sigma}}{d_{q\sigma}} = \frac{4}{1}$

squares with the ratio 2 : 1 (1 : 2). Thus we have $\frac{d_{p\sigma}}{d_{q\sigma}} = \frac{4}{1}$ and also $d_{p\sigma} = \frac{4}{5} d_{pq}$ and $d_{q\sigma} = \frac{1}{5} d_{pq}$.

Because $\frac{1}{T_{pq}} = \frac{d_{p\sigma}+d_{q\sigma}}{|\sigma_{pq}|}$ and $\frac{d_{p\sigma}}{|\sigma_{pq}|} = \frac{d_{p\sigma}}{|\sigma_{pq}|} \frac{1}{T_{pq}}$, the approximation of the flux (41) can be expressed in the form

$$F^n_{p\sigma} = g^{s,n-1}_p T_{pq} \frac{d_{pq}}{d_{p\sigma}} \left( u^n_\sigma - u^n_p \right). \tag{43}$$

Now we solve the linear system, where the set of equations for every finite volume $p$

$$\left( u^n_p - u^{n-1}_p \right) |p| = \tau \sum_{\sigma \in \mathcal{E}_p} F^n_{p\sigma}, \tag{44}$$

is accompanied by a set of equations for every $u^n_\sigma$, $\sigma \in \mathcal{E}_p$. The equation for $u^n_\sigma$ is obtained from the equations:

$$\begin{aligned} F^n_{p\sigma} &= -F^n_{q\sigma}, \\ g^{s,n-1}_p T_{pq} \frac{d_{pq}}{d_{p\sigma}} \left( u^n_\sigma - u^n_p \right) &= -g^{s,n-1}_q T_{pq} \frac{d_{pq}}{d_{q\sigma}} \left( u^n_\sigma - u^n_q \right), \end{aligned}$$

$$u^n_\sigma = \frac{d_{q\sigma} g^{s,n-1}_p u^n_p + d_{p\sigma} g^{s,n-1}_q u^n_q}{d_{q\sigma} g^{s,n-1}_p + d_{p\sigma} g^{s,n-1}_q}. \tag{45}$$

### 5.2 Eliminating $u_\sigma$ from the linear system

Because the values $u_\sigma$ can be excluded from the expression $u^n_\sigma - u^n_p$ of (43), we define also $F^n_{pq}$, which is used later to express different form of the semi-implicit scheme, leading to a linear system which does not contain values $u_\sigma$ and is formulated at the end of this section. If we replace $u^n_\sigma - u^n_p$ in (43) by

$$u^n_\sigma - u^n_p = \frac{d_{p\sigma} g^{s,n-1}_q (u^n_q - u^n_p)}{d_{q\sigma} g^{s,n-1}_p + d_{p\sigma} g^{s,n-1}_q},$$

we have

$$\begin{aligned} F^n_{p\sigma} &= \frac{T_{pq}}{d_{p\sigma}} \frac{g^{s,n-1}_p d_{pq} g^{s,n-1}_q d_{p\sigma}}{d_{q\sigma} g^{s,n-1}_p + d_{p\sigma} g^{s,n-1}_q} \left( u^n_q - u^n_p \right) \\ &= T_{pq} d_{pq} \frac{g^{s,n-1}_p g^{s,n-1}_q}{d_{q\sigma} g^{s,n-1}_p + d_{p\sigma} g^{s,n-1}_q} \left( u^n_q - u^n_p \right), \\ &= T_{pq} \frac{1}{\frac{d_{p\sigma}}{d_{pq}} \frac{1}{g^{s,n-1}_p} + \frac{d_{q\sigma}}{d_{pq}} \frac{1}{g^{s,n-1}_q}} \left( u^n_q - u^n_p \right). \end{aligned}$$

This flux, from now denoted as $F^n_{pq}$ can be expressed also in the form

$$F^n_{pq} = T_{pq} G^{s,n-1}_{pq} (u^n_q - u^n_p), \tag{46}$$

with the same values of $T_{pq}$ as in the case of the linear heat equation (1) and with $G_{pq}^{s,n-1}$ defined by

$$G_{pq}^{s,n-1} = \frac{1}{W_1 \frac{1}{g_p^{s,n-1}} + W_2 \frac{1}{g_q^{s,n-1}}}, \tag{47}$$

(48)

where $W_1 = \frac{d_{p\sigma}}{d_{pq}}$, $W_2 = \frac{d_{q\sigma}}{d_{pq}}$, $W_1 + W_2 = 1$ and their values are $\frac{1}{2}$, $\frac{1}{5}$ or $\frac{4}{5}$, depending on the configuration of elements.

If we take $g(s^2) = \frac{1}{1+Ks^2}$, then we can write

$$G_{pq}^{s,n-1} = \frac{1}{W_1(1 + K|\nabla u_p^{s,n-1}|^2) + W_2(1 + K|\nabla u_q^{s,n-1}|^2)} = \tag{49}$$

$$= \frac{1}{1 + K\left(W_1|\nabla u_p^{s,n-1}|^2 + W_2|\nabla u_q^{s,n-1}|^2\right)} = g\left(|\overline{\nabla} u_{pq}^{s,n-1}|^2\right). \tag{50}$$

Finally, one equation of the the linear system can be written as

$$\left(u_p^n - u_p^{n-1}\right) m(p) = \tau \sum_{q \in N_p} F_{pq}^n. \tag{51}$$

*Remark 2* Convergence of a finite volume scheme on a regular rectangular grid to solution of the regularized Perona-Malik equation was proven in [20]. However, the numerical scheme proposed in this paper for adaptive consistent grids uses different ideas proposed in [11] for solving curvature driven level set equation. Proof of convergence of this new scheme to the solution of regularized Perona-Malik problem is thus not straightforward and will be an objective of our future research.

## 6 Numerical experiments

The adaptive algorithm can be used in several ways and here we present two adaptive approaches:

- We start on a regular grid and continue to use it until the decrease of elements is sufficient. The number of elements corresponding to the current data is found out by building the adaptive grid by fast, linear in time algorithm: in the case of reaching undesired number of them, we can stop building the adaptive grid and continue on a regular one. As soon as the decrease of elements is sufficient we let the adaptive algorithm work until the stopping time on the same adaptive grid (Experiments 4 and 5). Advantage of this access is that for the fixed adaptive grid we can store all necessary information, e.g. configurations of neighbors, type of the matrix $B$ to evaluate the gradients, etc.
- After the sufficient decrease of elements we build a new adaptive grid in every subsequent times step (Experiment 6).

**Experiment 4** The data in this experiment represents artificial data of the size $128 \times 128$ disturbed by the additive noise. We performed 13 scale steps with $\tau = 1$, $K = 1000$ and the time of presmoothing $\tau_s = 0.6$. The grid elements were reduced to $\frac{1}{3}$ after 5 scale steps, and then we continued on the fixed grid. The parameter $\varepsilon$ used in the coarsening criterion is set to 0.01. Figure 13 shows the data itself, the filtered data and the adaptive grid fixed after 5 scale steps.

**Experiment 5** The data in this experiment is of the size $460 \times 512$ (cf. Fig. 14). They were embedded into the size $512 \times 512$ using the constant (picture background) color for the added elements. In the added part thus we have large grid elements and the efficiency is not influenced significantly. We performed 15 scale steps. The fixed grid is created after the 6th scale step, when number of elements decreased to one third of 262144. We use $\varepsilon = 0.03$, $\tau = 1$, $\tau_s = 0.6$ and $K = 1600$.

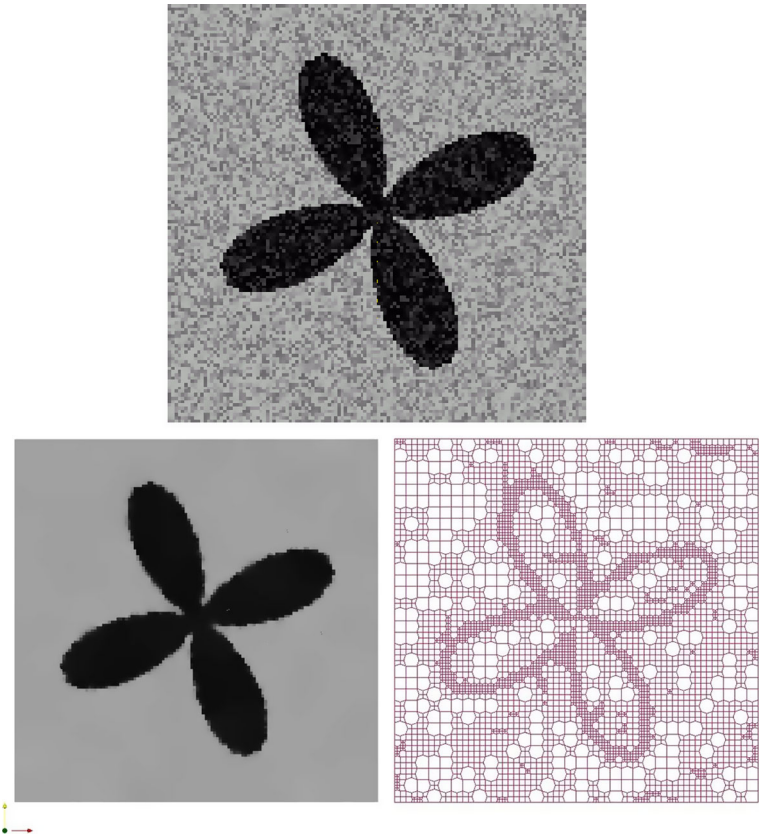

**Fig. 13** Experiment 4. *Top*: the artificial image with additive noise. *Bottom*: the filtered image (*left*) with the adaptive grid fixed after 5 scale steps

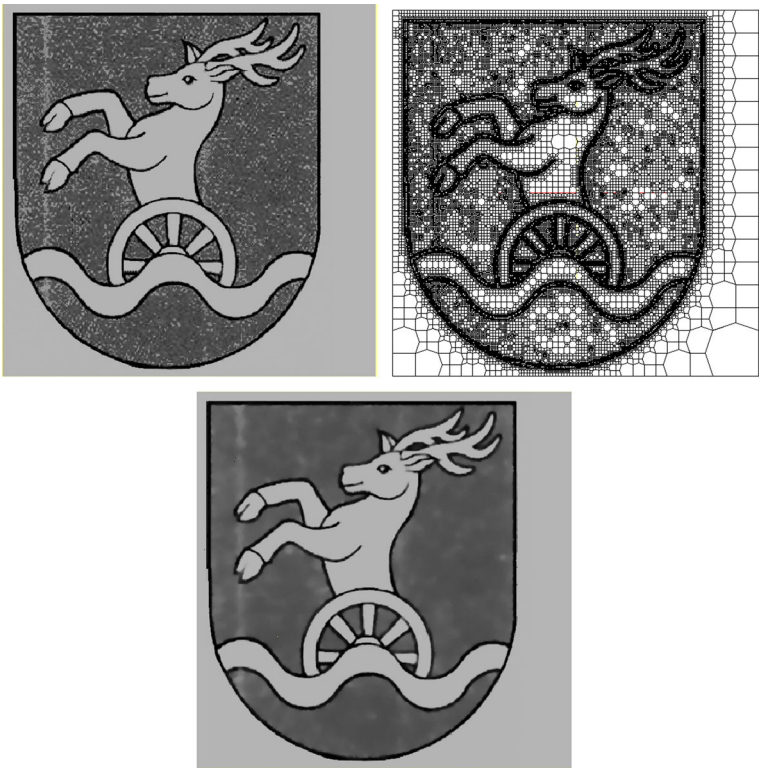

**Fig. 14** Experiment 5. From the *left*: the initial image, the fixed adaptive grid and the result

In the last experiment we work with the data from the Experiment 5, but we build the adaptive grid in every scale step. We use a simple coarsening criterion, during merging we compare the squared magnitude of the gradient with a given threshold $\varepsilon_2$. To compare with the grids built in previous examples, the difference is that the grid obtained in this way is dense around the edges, because the gradient depends on the boundary values in $x_\sigma$. The reason is that after building a new grid, we must evaluate $u_\sigma$ with the formula (45) using the values $g_p^{s,n-1}$ from the previous time step. This is one possibility ensuring also small difference of values in merged regions. (Another possibility is to demand, except of the small difference in solution values, also small difference of $g_p^{s,n-1}$ during merging regions.) The value of $\varepsilon_2$ depends on $K$ in the Perona - Malik function $g$.

**Experiment 6** The data in this experiment represents data of the size $460 \times 512$ from the Experiments 5. We performed 15 scale steps with $\tau = 1$, with $K = 1600$ and the time of presmoothing $\tau_s = 0.6$, $\varepsilon_2 = 0.001$. Figure 15 shows the images and corresponding grids after the 7th and 15th scale steps. The initial number of elements was 262144, the final number of elements was 65674.

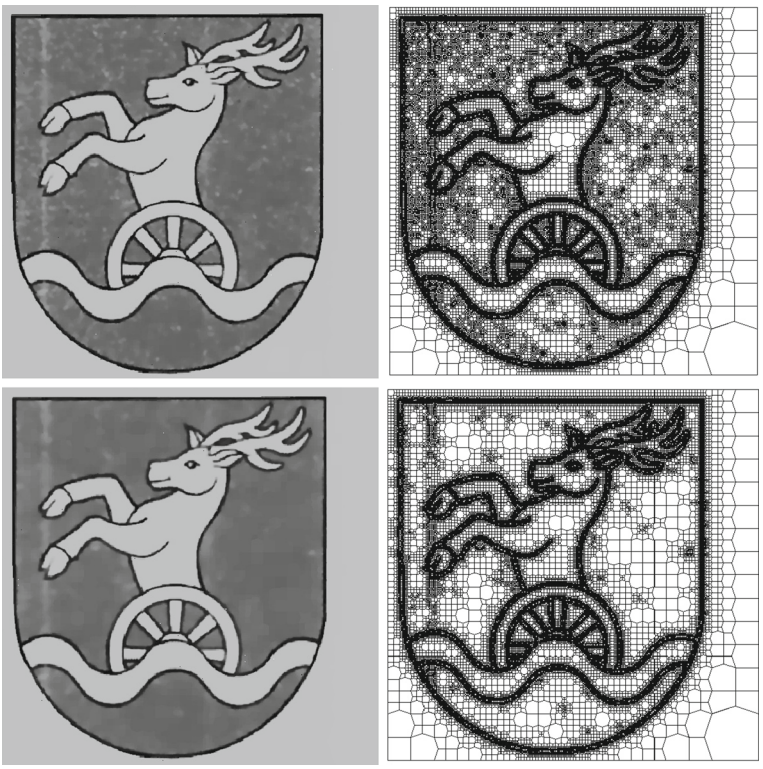

**Fig. 15** Experiment 6. The results with corresponding grids in 7th and 15th scale steps

**Acknowledgments** This work was supported by APVV-0184-10 and APVV-0161-12. We thank to Robert Eymard who pointed to us at ALGORITMY 2002 the construction of the consistent grid from the quadtree structure.

## References

1. Aftosmis, M.J., Berger, M.J., Melton, J.R.: Adaptive Cartesian mesh generation. Chapter 22 in Handbook of Grid Generation. CRC Press (1998)
2. Alvarez, L., Guichard, F., Lions, P.L., Morel, J.M.: Axioms and fundamental equations of image processing. Archive Rat. Mech. Anal. **123**, 200–257 (1993)
3. Bänsch, E., Mikula, K.: A coarsening finite element strategy in image selective smoothing. Comput. Visual. Sci. **1**(1), 53–61 (1997)
4. Bänsch, E., Mikula, K.: Adaptivity in 3D image processing. Comput. Visual. Sci. **4**(1), 21–30 (2001)
5. Chen, Z., Wu, J., Xu, Y.: Higher-order finite volume methods for elliptic boundary value problems. Adv. Comput. Math. **37**, 191–253 (2012)
6. Chen, Z., Xu, Y., Zhang, Y.: A construction of higher-order finite volume methods. Math. Comput. **S0025–5718**, 02891–3 (2014)
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. ICCV, 694–699 (1995)
8. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **129**, 182–193 (1992)

9. Coudiére, Y., Vila, J.P., Villedieu, P.: Convergence rate of a finite volume scheme for the linear convection-diffusion equation on locally refined meshes. ESAIM: M2AN **34**(6), 1123–1149 (2000)
10. Droske, M., Meyer, B., Rumpf, M., Schaller, C.: An Adaptive level set method for medical image segmentation. Lect. Notes Comput. Sci **4**(2001), 416–422 (2082)
11. Eymard, R., Gallouet, T., Herbin, R.: The finite volume method. In: Ciarlet, Ph., Lions, J. L. (eds.) Handbook for Numerical Analysis, vol. 7. Elsevier (2000)
12. Eymard, R., Handlovičová, A., Mikula, K.: Study of a finite volume scheme for the regularized mean curvature flow level set equation. IMA J. Numer. Anal **31**(3), 813–846 (2010)
13. Chen, H., Min, Ch., Gibou, F.: A supra-convergent finite difference scheme for the poisson and heat equations on irregular domains and non-graded Cartesian Grids. J. Sci. Comput **31**(1/2), 19–59 (2007)
14. Johansen, H., Colella, P.: A cartesian grid embedded boundary method for Poisson's equation on irregular domains. J. Comput. Phys. **147**(60), 60–85 (1998)
15. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Conformal curvature flows: from phase transitions to active vision. Arch. Rational Mech. Anal **134**, 275–301 (1996)
16. Krivá, Z.: Adaptive finite volume methods in image processing. Edicia vedeckých prác, Zošit č. 15, Vydavatelstvo STU Bratislava (2004)
17. Krivá, Z., Mikula, K.: An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing. J. Vis. Commun. Image Represent **13**, 22–35 (2002)
18. Krivá, Z., Mikula, K.: Adaptive diamond cell finite volume method in image processing. Proc. ALGORITMY 2009, Conf. on Scientific Computing, pp. 174–188. Podbanské (2009)
19. Min, Ch., Gibou, F.: A second order accurate level set method on non-graded adaptive cartesian grids. In: Journal of Computational Physics, vol. 225, pp. 300–321. Elsevier (2007)
20. Mikula, K., Ramarosy, N.: Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing. Numerische Mathematik **89**(3), 561–590 (2001)
21. Ohlberger, M., Rumpf, M.: Adaptive projection operators in multiresolutional scientific visualization. IEEE Trans. Visual. Comput. Graph. **4**(4), 344–364 (1998)
22. Perona, P.: J. Scale space and edge detection using anisotropic diffusion. In: Proc. IEEE Computer Society Workshop on Computer Vision, Malik (1987)
23. Preusser, T., Rumpf, M.: An adaptive finite element method for large scale image processing. J. Vis. Commun. Image Represent. **11**(2), 183–195 (2000)
24. Preusser, T., Rumpf, M.: A level set method for anisotropic geometric diffusion in 3D image processing. SIAM J. Appl.Math. **62**(5), 1772–1793 (2002)
25. Samet, H.: Application of spatial data structures: Computer Graphics, Image Processing and GIS. Adison Wesley, New York (1990)
26. Samet, H.: The design and analysis of spatial data structures. Adison Wesley, New York (1989)
27. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Science. Cambridge University Press, New York (1999)
28. Vemuri, B., Chen, Y., Wang, Z.: Registration assisted image smoothing and segmentation. ECCV **4**, 546–559 (2002)