Acta Univ. Palacki. Olomuc., Fac. rer. nat.,
Mathematica **54**, 2 (2015) 41–61

Quad-tree Based Finite Volume Method for Diffusion Equations with Application to SAR Imaged Filtering

Zuzana KRIVÁ [a], Juraj PAPČO [b], Jakub VANKO [c]

Faculty of Civil Engineering STU, Radlinského 11, Bratislava, Slovakia
[a] *e-mail: kriva@math.sk*
[b] *e-mail: juraj.papco@stuba.sk*
[c] *e-mail: vanko.jakub@gmail.com*

(Received June 22, 2015)

Abstract

In this paper we present a method to remove the noise by applying the Perona Malik algorithm working on an irregular computational grid. This grid is obtained with a quad-tree technique and is adapted to the image intensities—pixels with similar intensities can form large elements. We apply this algorithm to remove the speckle noise present in SAR images, i.e., images obtained by radars with a synthetic aperture enabling to increase their resolution in an electronic way. The presence of the speckle in an image degrades the quality of the image and makes interpretation of features more difficult. Our purpose is to remove this noise to such a degree that the edge detection or landscape elements detection can be performed with relatively simple tools. The progress of smoothing leads to grids with significantly less number of elements than the original number of pixels. The results are compared with measurements performed on an inspected area of interest. At the end we show the possibility to modify the scheme to the adaptive mean curvature flow filter which can be used to smooth the boundaries.

Key words: Image processing, linear heat equation, finite volume method, adaptivity, SAR image, speckle noise.

2010 Mathematics Subject Classification: 65M50, 65M60, 68U10

1 Introduction

When we apply diffusion equation to noisy images, it often happens that with a progress of smoothing large areas of homogeneous intensity arise. By an

adaptive algorithm we will understand an algorithm using an adaptive grid, i.e., the grid where the larger grid elements are used in homogeneous regions (see e.g., Fig.1). In this paper, our method to remove the noise is based on the regularized Perona–Malik model representing an evolutionary parabolic PDE [3]. Adaptive algorithms for this model have been explored, e.g., in [1] for triangular finite elements grids or in [7], [8] for quad-tree finite volume grids. The criterion which decides if smaller elements can be merged into larger element is *called coarsening criterion* and in the above mentioned papers it was based on a small intensity difference of pixels to be merged. In the case of a success the new representative value of a larger element is set to an arithmetic mean.

In [7] and [8] we have noticed that the adaptive algorithm is able to flatten the image better than a nonadaptive algorithm though it can sometimes introduce weak artificial vertical and horizontal edges which can be removed in post processing. This effect can be explained by two types of diffusivity: the Perona–Malik type which represent a weighted averaging between a pixel and its neighbors and using the arithmetic mean for setting a representative value of large elements of a quad-tree grid. The quad-tree grid also simplifies the representation of the images, thus making it faster especially in the last steps of smoothing process. Moreover this representation can be used in post processing and can be possibly used by other algorithms. In this paper w e have decided to explore the use of such an algorithm in detection of landscape elements in SAR images, i.e. the images obtained by radars with a synthetic aperture enabling their better resolution. Such images are degraded by the speckle noise: we want to remove this noise and to explore performance of edge detection and simple segmentation algorithms on smoothed versions of the images. The results are included in the section Numerical experiments. For the space discretization we use the finite volume method (FVM) [4]: the finite volumes correspond to quad-tree square elements.

In [7] and partially in [8], during the evaluation of the gradients we had to access to neighbors of a processed finite volume sharing only the vertices. Because the grid is not stored explicitly [7], finding such neighbors requires a lot of tests. In FVM each finite volume has a representative point in its center.

In the presented algorithm we use a technique which employs also midpoints of finite volume edges—to get the gradients and their norms *locally*. Afterwards the solution values in these points are obtained using the conservation principle: all the time for every FV we need only its neighbors sharing a common edge (i.e. [5], [6]). To achieve a finer work on edges, for every finite volume we evaluate four gradients ([6]) which are used later to form edge diffusion coefficients. We derive the schemes semi-implicit in time, leading to solving linear systems. The decrease of grid elements leads to a decrease of a number of unknowns in this linear system and can speed up computations significantly.

First, in Section 2 we explain the principles of the adaptive method for the linear diffusion model which can be understood as a special case of the Perona–Malik equation. There, no gradients and their norms are needed—they are explained afterward in Section 3 and Section 4. The section with numerical experiments and results follow.

2 Solving the linear heat equation on an adaptive grid

2.1 The basic adaptive grid

Now let our task be exploring the solution of the linear heat equation on an adaptive grid, i.e., the grid which is adapted to the processed data. We use a grid which is based on quad-trees with the prescribed ratio of adjacent elements sides: 1 : 1 or 1 : 2. The methodology of building and traversing the grid is described e.g. in [7]. There, the following coarsening criterion has been used: the cells can be merged to a larger square element if an intensity difference of all pixels within this candidate square is below a prescribed tolerance ε.

In our method to form the adaptive grid we store also values of in midpoints of the edges (denoted later as u_σ), so now we use three parameters to steer the grid creation:

- ε_1 – the value for a maximal intensity difference of pixels in a grid element. If it is satisfied, the subregions are merged into a larger one with the representative value set to their average.
- ε_2 – the value for a maximal intensity difference of u_{σ_1} and u_{σ_2} (the solution values in edge representative points, see Fig. 2),
- ε_3 – the value for a maximal intensity difference between a volume value and edge values. This criterion enables to keep small elements in the vicinity of edges (see e.g. Fig. 1).

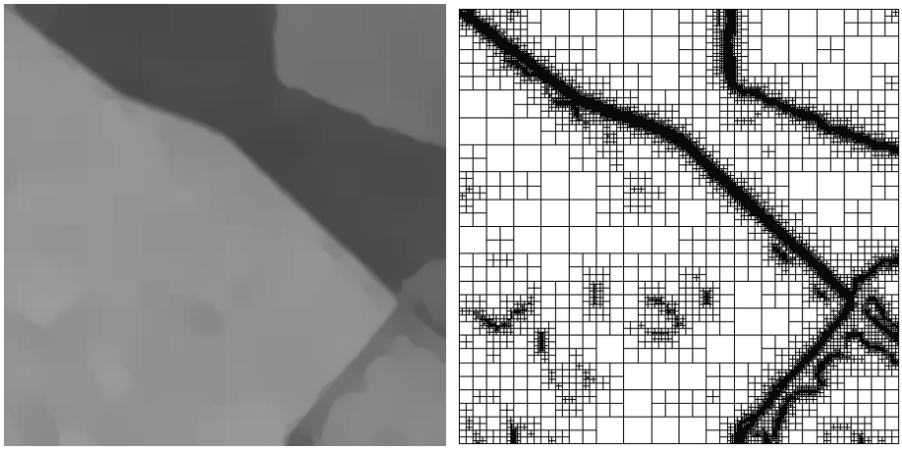

Figure 1: Example of an adaptive grid. In flat regions the larger elements are used, in the vicinity of edges the elements are small.

2.2 The linear heat equation

$$\partial_t u(x,t) - \Delta u(x,t) = 0 \quad \text{in } Q_T \equiv \Omega \times I, \tag{2.1}$$

$$\partial_{\vec{n}} u(x,t) = 0, \quad \forall x \in \partial\Omega \times I, \tag{2.2}$$

$$u(0,x) = u_x^0, \quad \forall x \in \Omega. \tag{2.3}$$

Here, $u(x,t)$ is an unknown function representing smoothed (filtered) image intensity defined defined in $\Omega \subset R^2$, $I = [0,T]$ is the time interval and $u^0(x)$ is the image we start with.

2.3 The numerical scheme

First we introduce the necessary notation for the finite volume discretization. Let $\mathcal{T}$ be an adaptive grid with finite volumes p of measure $|p|$ and let $N(p)$ be the set of neighbors $q \in \mathcal{T}$ for which common interface of p and q is a line segment σ with a nonzero measure in 1D. Let every finite volume p have a representative point x_p lying in its center.

Let u_p denote the solution value constant over p. Let ε_p be a set of all edges σ of a finite volume p, with their measures denoted by $|\sigma|$. As we have mentioned in the introduction we use also the midpoints of the edges with values obtained by the conservation principle. Let x_σ be the midpoint of an edge σ belonging to p and u_σ be the value in this midpoint. Then $d_{p\sigma} = |x_\sigma - x_p|$ and $n_{p\sigma}$ is a normal vector to σ outward to p. The derivative in the direction $n_{p\sigma}$ is approximated by

$$\nabla u^n \cdot n_{p\sigma} \approx \frac{u_\sigma^n - u_p^n}{d_{p\sigma}}.$$

Having the grid, we can integrate the diffusion equation over a finite volume p and we use the divergence theorem to obtain

$$\int_p \partial_t u \, dx - \int_{\partial p} \nabla u . n_p \, ds = 0, \tag{2.4}$$

where $n_p = (n_x, n_y)$ is the outward unit normal vector to ∂p.

We replace the time derivative by a finite difference using a uniform time step $\tau = t^n - t^{n-1}$, where t^{n-1}, t^n are previous and current time steps, respectively. Let u^n be the solution in the n^{th} time step and u_p^n denotes the solution over the finite volume p in the n^{th} time step. Having the integral form (2.4) for (2.1), let us denote by

$$f_{p\sigma}^n = \int_\sigma \nabla u^n . n_{p\sigma} \, ds \tag{2.5}$$

the implicit flux through boundary σ of p. Then the implicit scheme can be rewritten in the following general form

$$(u_p^n - u_p^{n-1}) \, |p| = \tau \sum_{\sigma \in \varepsilon_p} f_{p\sigma}^n, \tag{2.6}$$

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

The flux $f_{p\sigma}^n$ contains a normal derivative of a solution at the time step t^n evaluated on the boundary σ of p. Because this normal derivative is constant over σ, the flux can be approximated numerically by:

$$f_{p\sigma}^n = \int_\sigma \nabla u^n . n_{p\sigma}\, ds \approx F_{p\sigma}^n = \frac{|\sigma|}{|d_{p\sigma}|} \frac{\left(u_\sigma^n - u_p^n\right)}{d_{p\sigma}}. \tag{2.7}$$

After rearranging the terms we come to the following implicit numerical scheme in the flux form:

$$(u_p^n - u_p^{n-1}) \frac{|p|}{\tau} = \sum_{\sigma \in \varepsilon_p} F_{p\sigma}^n = \sum_{\sigma \in \varepsilon_p} \frac{|\sigma|}{|d_{p\sigma}|} \left(u_\sigma^n - u_p^n\right). \tag{2.8}$$

The nonadaptive grid. Elements are squares of the same size, so $\frac{|\sigma|}{d_{p\sigma}} = 2$. For every p we have the equation:

$$(u_p^n - u_p^{n-1}) \frac{|p|}{\tau} = \sum_{\sigma \in \varepsilon_p} 2 \left(u_\sigma^n - u_p^n\right). \tag{2.9}$$

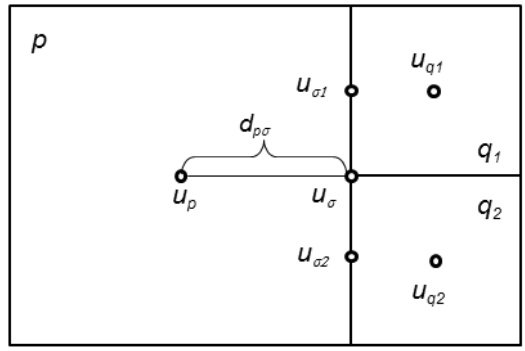

Figure 2: Notation for a nonconformal situation, i.e., adjacent finite volumes with nonequal size.

The adaptive grid. In a nonconformal situation, i.e., for non equal adjacent elements, we use the notation displayed in Fig. 2. The value in u_σ will not be a degree of freedom, but we set it to

$$u_\sigma = \frac{u_{\sigma_1} + u_{\sigma_2}}{2}. \tag{2.10}$$

To balance fluxes, we use

$$2(u_\sigma - u_p) = 2(u_{q_1} - u_{\sigma_1}) + 2(u_{q_2} - u_{\sigma_2}). \tag{2.11}$$

After putting together (2.10) and (2.11) we get:

$$3\frac{u_{\sigma_1}+u_{\sigma_2}}{2} = u_{q_1}+u_{q_2}+u_p,$$
$$u_\sigma = \frac{1}{3}(u_{q_1}+u_{q_2}+u_p). \tag{2.12}$$

Eliminating u_σ from the equation (2.9). If p has got two *smaller* neighbors q_1 and q_2 over an edge, then using (2.12) we have

$$2(u_\sigma^n - u_p^n) = \frac{2}{3}u_{q_1}^n + \frac{2}{3}u_{q_2}^n - \frac{4}{3}u_p^n = \frac{2}{3}(u_{q_1}^n - u_p^n) + \frac{2}{3}(u_{q_2}^n - u_p^n). \tag{2.13}$$

Finally let us set u_{σ_1} and u_{σ_2} for the element q_1 and the element q_2 having a large neighbor p. We set

$$u_{\sigma_1} = \frac{1}{3}u_p + \frac{2}{3}u_{q_1}, u_{\sigma_2} = \frac{1}{3}u_p + \frac{2}{3}u_{q_2}, \tag{2.14}$$

(notice that (2.12) holds).

When we eliminate u_σ for a finite volume p having a *larger* neighbor q, we have

$$u_\sigma = \frac{1}{3}u_q + \frac{2}{3}u_p \tag{2.15}$$

and then

$$2\left(u_\sigma^n - u_p^n\right) = 2\left(\frac{1}{3}u_q^n + \frac{2}{3}u_p^n - u_p^n\right) = \frac{2}{3}\left(u_q^n - u_p^n\right). \tag{2.16}$$

Now the semi-implicit adaptive numerical scheme for (2.1)–(2.3) where all σ are eliminated can be written in the form:

$$(u_p^n - u_p^{n-1})\frac{m(p)}{\tau} = \sum_{q\in N(p)} T_{pq}\left(u_q^n - u_p^n\right). \tag{2.17}$$

where

- for q of the different size as p, $T_{pq} = \frac{2}{3}$.
- for q of the same size as p, $T_{pq} = 1$.

The scheme leads to the following linear system:
Let $0 = t_0 \le t_1 \le \ldots \le t_{N_{max}} = T$ denote the time stepping with $t_n = t_{n-1} + \tau$, where τ is the time step. For $n = 1, \ldots, N$ we look for u_p^n, $p \in \mathcal{T}_h$ satisfying (2.17) [14].

2.4 Alternative solution

The linear system can be solved in such a way that for nonconformal situations u_σ is not eliminated from the fluxes. In such a case for p we have the flux like in (2.9) and we add the equations (2.18)–(2.20) for updating σ to the system

(only once, i.e., in the situation smaller p versus bigger q). In our method we eliminate u_σ only in the case that over some edge we have a neighbor of the same size.

$$u_\sigma = \frac{u_{\sigma_1} + u_{\sigma_2}}{2}, \tag{2.18}$$

$$u_{\sigma_1} = \frac{1}{3}u_p + \frac{2}{3}u_{q_1}, \tag{2.19}$$

$$u_{\sigma_2} = \frac{1}{3}u_p + \frac{2}{3}u_{q_2}. \tag{2.20}$$

The advantage of such an approach is that equations of the linear system has a fixed number of elements and is easy to manipulated with.

3 The regularized Perona–Malik equation

In this section, we deal with the initial boundary value problem [3]:

$$\partial_t u - \nabla.(g(|\nabla G_s * u|)\nabla u) = 0, \quad \text{in } Q_T \equiv \Omega \times I, \tag{3.1}$$

$$\partial_{\vec{n}} u(x,t) = 0, \quad \forall x \in \partial\Omega \times I, \tag{3.2}$$

$$u(0,x) = u_x^0, \quad \forall x \in \Omega. \tag{3.3}$$

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

$$g(v) \text{ is a decreasing smooth function,}$$
$$g(0) = 1,\ 0 < g(v) \to 0 \text{ for } v \to \infty, \tag{3.4}$$

$$G_s \in C^\infty(\mathbb{R}^2) \text{ is a smoothing kernel with } \int_{\mathbb{R}^2} G_s(x)\,dx = 1 \tag{3.5}$$

$$\text{and } G_s(x) \to \delta_x \text{ for } s \to 0,\ \delta_x \text{ – Dirac function at point } x,$$
$$u^0 \in L_2(\Omega). \tag{3.6}$$

In our experiments we use the function

$$g(v) = \frac{1}{1 + Kv^2}.$$

This equation is nonlinear: the diffusion is slowed down on edges characterized by higher value of presmoothed (Gaussian) gradients $\nabla G_s * u$. Their norms will be inputs for the function g. The outputs are called the diffusion coefficients and for an edge σ of the finite volume p they are denoted by $g_{p\sigma}$. The evaluation of the gradients will be shown later. Now it is important that the diffusion coefficient is constant over σ, it depends on the solution from the previous time step, so by similar argument like in (2.7) we can derive the local form of the semi-implicit numerical scheme:

$$(u_p^n - u_p^{n-1})|p| = \tau \sum_{\sigma \in \varepsilon_p} g_{p\sigma}^{n-1,s} \frac{|\sigma|}{d_{p\sigma}} \left(u_\sigma^{n+1} - u_p^{n+1}\right). \tag{3.7}$$

3.1 Evaluation of the gradient for (3.1)

Let us denote by V_p the set of all vertices of a finite volume p, e.g., in 2-dimensional case $V_p = \{y_1, y_2, y_3, y_4\}$ (see Fig. 3)). For $y \in V_p$, we denote by (p, y) the rectangle whose faces are parallel to those of p, and whose set of vertices contains x_p and y. Let V_σ denote the endpoints of the edge σ. We denote by $\varepsilon_{p,y}$ the set of all $\sigma \in \varepsilon_p$ such that $y \in V_\sigma$. In our case it has two elements, for example, in Fig. 3 $\varepsilon_{p,y_1} = \{\sigma_1, \sigma_2\}$. The norm of the gradient evaluated on the part (p, y) is denoted by $|\nabla_{p,y} u|$ and holds [5]

$$|\nabla_{p,y} u| = \sqrt{\frac{4}{h^2} \sum_{\sigma \in \varepsilon_{p,y}} (u_\sigma - u_p)^2}. \tag{3.8}$$

Then in the semi-implicit scheme (3.7) the diffusion coefficient $g_{p\sigma}^{n,s}$ is based on the relationship

$$g_{p\sigma}^{n,s} = \sum_{y \in V_\sigma} g(|\nabla_{p,y}(G_s * u^n|). \tag{3.9}$$

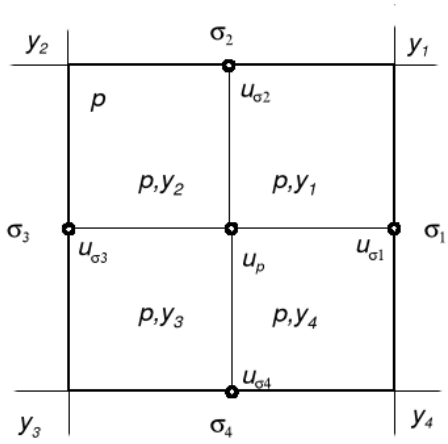

Figure 3: Notation for evaluation of the gradient. x_{σ_i} – the midpoints of σ_i are denoted by circles on the edges, the solution value in x_{σ_i} is denoted by u_{σ_i}.

Remark 1 If $s = 0$ we have the basic Perona–Malik equation and we apply (3.8) directly. For $s > 0$, to get the Gaussian gradient $\nabla(G_s * u)$, we can replace performing the convolution by solving the linear heat equation (2.1) for time t corresponding to s ($t = \sqrt{\frac{s}{2}}$). We apply (2.1) to u^n and using (2.11) for u_σ we evaluate the norm of this presmoothed gradient applying (3.8). In the following notation we omit s and write only the time index to stress that the diffusion coefficient depends on the solution from the previous time step.

Like before, the values in u_σ are set using the conservation principle. In a regular grid we have:

$$u_\sigma^n = \frac{g_{p\sigma}^{n-1} u_p^n + g_{q\sigma}^{n-1} u_q^n}{g_{p\sigma}^{n-1} + g_{q\sigma}^{n-1}} \tag{3.10}$$

Let $g_{p\sigma} g_{q_1\sigma}$ ($g_{p\sigma} g_{q_2\sigma}$) be diffusion coefficient for q_1 (q_2) evaluated on a portion of σ common to a finite volume p and q_1 (q_2). In the adaptive scheme, in a nonconformal situation, in a similar manner like for the heat equation, we can derive:

$$u_{\sigma_1}^n = \frac{g_{p\sigma}^{n-1} u_p^{n+1} + 2 g_{q_1\sigma}^{n-1} u_{q_1}^{n+1}}{g_{p\sigma}^{n-1} + 2 g_{q_1\sigma}^{n-1}} \tag{3.11}$$

$$u_{\sigma_2}^n = \frac{g_{p\sigma}^{n-1} u_p^{n+1} + 2 g_{q_2\sigma}^{n-1} u_{q_2}^{n+1}}{g_{p\sigma}^{n-1} + 2 g_{q_2\sigma}^{n-1}} \tag{3.12}$$

and

$$u_\sigma = \frac{u_{\sigma_1} + u_{\sigma_2}}{2}. \tag{3.13}$$

If $g \equiv 1$, we have the linear heat equation and (3.11) and (3.12) correspond to (2.19) and (2.20).

3.2 Construction of the linear system

Now we describe the construction of the fluxes in the basic equation (3.7). Let us write it in a form:

$$(u_p^n - u_p^{n-1})|p| = \tau \sum_{\sigma \in \varepsilon_p} F_{p\sigma}^n. \tag{3.14}$$

Like before we distinguish two cases: the conformal situation with a neighbor of the same size and the nonconformal one between two elements of non equal size.

1. **The conformal situation.** On the edge σ between p and q, we eliminate u_σ and get:
$$F_{p\sigma}^n = \frac{2 g_{p\sigma}^{n-1} g_{q\sigma}^{n-1}}{g_{p\sigma}^{n-1} + g_{q\sigma}^{n-1}} (u_q^n - u_p^n). \tag{3.15}$$

2. **The nonconformal situation.** We use the local flux
$$F_{p\sigma}^n = g_{p\sigma}^{n-1} (u_q^n - u_p^n) \tag{3.16}$$
and add equations (3.11), (3.12) and (3.13) to the system(once).

The advantage of this approach is that every equation of the linear system has the same number of entries and it is easy to organize. The matrix of the linear system is the M-matrix, so it can be solved iteratively, e.g., using the Gauss-Seidel method or its modification—the SOR method, both with a good convergence.

Now u_σ can be expressed as a weighted mean of u_p,u_{q_1} and u_{q_2}: omitting the time indices we have

$$u_\sigma = w_1 \cdot u_p + w_2 \cdot u_{q_1} + w_3 \cdot u_{q_2}, \tag{3.17}$$

where

$$\begin{aligned} w_1 &= \frac{g_{p\sigma}^2 + g_{p\sigma}g_{q_1\sigma} + g_{p\sigma}g_{q_2\sigma}}{g_{p\sigma}^2 + 2g_{p\sigma}g_{q_1\sigma} + 2g_{p\sigma}g_{q_2\sigma} + 4g_{q_1\sigma}g_{q_2\sigma}}, \\ w_2 &= \frac{g_{p\sigma}g_{q_1\sigma} + 2g_{q_1\sigma}g_{q_2\sigma}}{g_{p\sigma}^2 + 2g_{p\sigma}g_{q_1\sigma} + 2g_{p\sigma}g_{q_2\sigma} + 4g_{q_1\sigma}g_{q_2\sigma}}, \\ w_3 &= \frac{g_{p\sigma}g_{q_2\sigma} + 2g_{q_1\sigma}g_{q_2\sigma}}{g_{p\sigma}^2 + 2g_{p\sigma}g_{q_1\sigma} + 2g_{p\sigma}g_{q_2\sigma} + 4g_{q_1\sigma}g_{q_2\sigma}}, \end{aligned}$$

where $g_{q_i\sigma}$ is a diffusion coefficient evaluated on a portion of σ common to a finite volume p and q_i.

3.3 The stability condition

We want to show that the scheme (3.14) is unconditionally stable, i.e., for any choice of a time step τ it holds that if $u_{\min} \le u_p^n \le u_{\max}$, then it also holds $u_{\min} \le u_p^{n+1} \le u_{\max}$.

Proof Let us write the scheme (3.14) in the following form:

$$u_p^{n+1} - \frac{\tau}{|p|} \sum_{\sigma \in \varepsilon_p} F_{p\sigma}^{n+1} = u_p^n. \tag{3.18}$$

Let us show (3.18) for the maximum. Let $u_{\max}^n$ denote the maximum in the n^{th} time step. We want to show that

$$\forall p \; u_p^n \le u_{max}^n \Rightarrow \forall p \; u_p^{n+1} \le u_{max}^n. \tag{3.19}$$

Let the maximum in the n^{th} time step be achieved in a finite volume pM with a value u_{pM}^n (there can be more such finite volumes). Substituting pM for p into (3.18) we get

$$u_{pM}^{n+1} - \frac{\tau}{|p|} \sum_{\sigma \in \varepsilon_{pM}} F_{pM,\sigma}^{n+1} = u_{pM}^n. \tag{3.20}$$

Now, let us explore the fluxes $F_{pM,\sigma}$ (3.15) and (3.16). The diffusion coefficients and their combinations we use are positive, $u_q^{n+1} - u_{pM}^{n+1}$ is always negative or equal to zero, and such is also $u_\sigma^{n+1} - u_{pM}^{n+1}$, due to (3.17), so $-\sum_{\sigma \in \varepsilon_p} F_{p\sigma}^{n+1}$ is nonnegative and it it holds

$$u_{pM}^{n+1} \le u_{pM}^n.$$

Then (3.19) and the fact that in pM we have the maximum in the n^{th} time step imply that

$$\forall p \; u_p^{n+1} \le u_{pM}^{n+1} = \max_r u_r^{n+1} \le u_{pM}^n \le \max_r u_r^n = u_{max}^n$$

and thus

$$\forall p \; u_p^{n+1} \leq u_{max}^n.$$

The left inequality of (3.18) can be shown in a similar manner. Also the stability condition for a linear heat equation and the mean curvature flow equation, mentioned in the following paragraph can be shown likewise. □

Example 1 In this example we take an image of the size 128×128 degraded by the *additive* noise (see Fig. 4). We have performed 20 time steps with $\tau = 1$, for the Perona–Malik function we set $K = 500$ and to create the adaptive grid $\varepsilon_1 = 0.015$, $\varepsilon_2 = 0.02$ and $\varepsilon_3 = 0.005$. The initial number of finite volumes was 16 384, after 5 time steps we had 10 147 finite volumes and at the end the grid had 5176 elements.

Figure 4: Left: the original noisy image. Middle: the filtered image. Right: the final adaptive grid.

4 Diffusion of multichannel images

First let us consider a RGB image. It can be viewed as a composition of three grayscale images, representing levels of intensity for red, green and blue colors. In [9] we propose a model which synchronizes the diffusion in each channel by computing a common diffusion coefficient depending on information coming from all three channels. This can often lead to a better preservation of edges. If we have multiple SAR images, i.e., images of the same area taken in different time periods, we can do the same (see Fig. 5). Of course such images can have a different set of edges because of different land cover: we can expect an improvement for dubbed edges in the same position. We introduce formulas (4.1) and (4.2) showing two possible ways how to obtain common diffusion coefficients. We consider two images, for more images the generalization is straightforward [14].

$$g_{p\sigma}^n = 2g\Big(\sum_{\sigma \in \varepsilon_{p,y}} |\nabla_{p,y}^1 u^n| + |\nabla_{p,y}^2 u^n|\Big) \tag{4.1}$$

$$g_{p\sigma}^n = \frac{1}{2} \sum_{\sigma \in \varepsilon_{p,y}} g(|\nabla_{p,y}^1 u^n|) + g(|\nabla_{p,y}^2 u^n|), \tag{4.2}$$

where the upper indices of the nabla operator are related to $image1$ and $image2$. Results are displayed in Fig. 6.

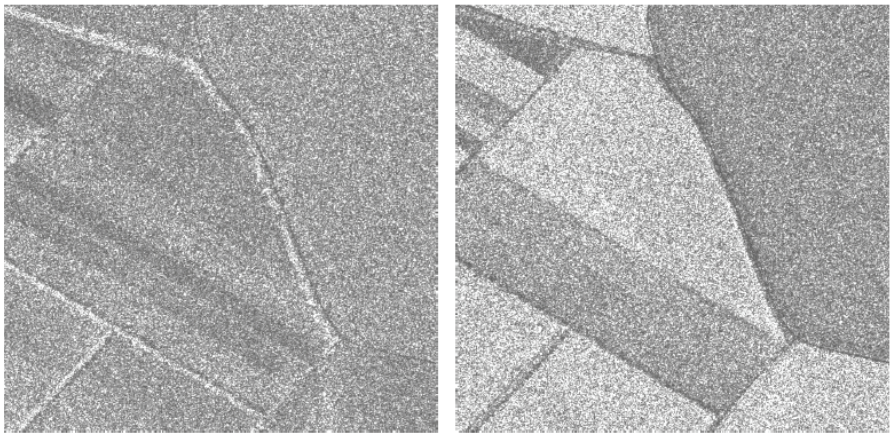

Figure 5: The images taken in different seasons of the same year (© 2012.

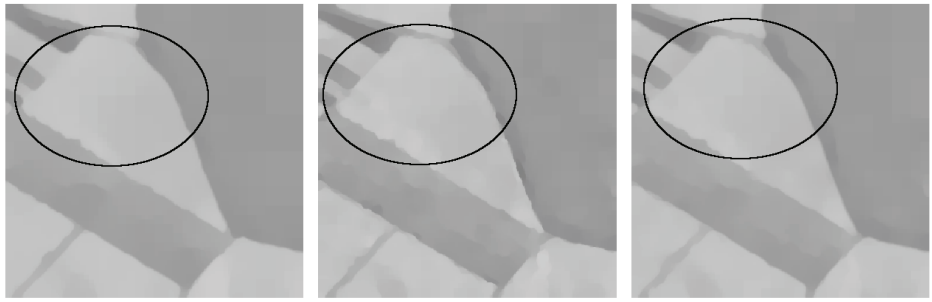

Figure 6: The right image from Fig. 5 is processed as a single image (on the left) and as a multichannel image by (4.1) and by (4.2). The black ellipse marks the area when we can observe the positive effect of multichannel diffusion.

5 The mean curvature flow (MCF) filter in a level set formulation

This filter can be used to solve problems of unsmooth boundaries and to remove the noise with a high curvature, like e.g. salt and pepper noise, see Fig. 7. The performance of this filter is similar to the median filter. Both averaging and median filters have been used since the beginning of SAR imaging. Though the equation of MCF is based on different principles, its scheme can be obtained by a simple modification of the Perona–Malik scheme. It has been used in

Experiment 2 as a fast post processing step, because the quad-tree structure of the image created by the Perona–Malik filter can be used.

Figure 7: MCF filter smooths the boundaries and removes noise with a high curvature.

The equation for this filter is

$$\partial_t u - |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0. \tag{5.1}$$

Because in flat regions the norm of a gradient can be equal to zero, we use *Evans–Spruck regularization*

$$|\nabla u| \approx |\nabla u|_\varepsilon = \sqrt{|\nabla u|^2 + \varepsilon^2} \tag{5.2}$$

For this equation we use a simpler evaluation of gradients which is sufficient for this equation (see Remark 2). Let us denote:

$$|\nabla u_p|^2 = \frac{2}{|p|} \sum_{\sigma \in \varepsilon_p} (u_\sigma - u_p)^2 \tag{5.3}$$

and

$$f_p = \sqrt{|\nabla u_p|^2 + \varepsilon^2}. \tag{5.4}$$

For our adaptive grid we use the following numerical scheme

$$\frac{(u_p^{n+1} - u_p^n)}{f_p} |p| = \tau \sum_{\sigma \in \varepsilon_p} \frac{2}{f_p} \left(u_\sigma^{n+1} - u_p^{n+1} \right). \tag{5.5}$$

If two elements share the same edge, to balance the fluxes, in a conformal situation we have

$$u_\sigma^{n+1} = \frac{f_q^n u_p^{n+1} + f_q^n u_p^{n+1}}{f_q^n + f_p^n}. \tag{5.6}$$

The mathematical and numerical properties of this scheme working on a regular square grid have been studied in [6]. For examples with a known analytical solution, EOC for the semi-implicit scheme has been reported as 2.

Like before, we can write the adaptive scheme as follows

$$\frac{(u_p^{n+1} - u_p^n)}{f_p}|p| = \tau \sum_{\sigma \in \varepsilon_p} F_{p\sigma}^{n+1}. \tag{5.7}$$

with a following setting of fluxes $F_{p\sigma}^{n+1}$:

1. **The conformal situation.** We eliminate u_σ and get:

$$F_{p\sigma}^{n+1} = \frac{2}{f_p^n + f_q^n}(u_q^{n+1} - u_p^{n+1})$$

2. **The nonconformal situation.** We use the local flux $F_{p\sigma}^{n+1} = \frac{2}{f_p^n}(u_q^{n+1} - u_p^{n+1})$ and add equations (5.8),(5.9) and (5.10)to the system(once):

$$u_{\sigma_1}^{n+1} = \frac{f_{q_1}^n u_p^{n+1} + 2 f_p^n u_{q_1}^{n+1}}{f_{q_1}^n + 2 f_p^n} \tag{5.8}$$

$$u_{\sigma_2}^{n+1} = \frac{f_{q_2}^n u_p^{n+1} + 2 f_p^n u_{q_2}^{n+1}}{f_{q_2}^n + 2 f_p^n} \tag{5.9}$$

and

$$u_\sigma = \frac{u_{\sigma_1} + u_{\sigma_2}}{2}. \tag{5.10}$$

Remark 2 For the Perona–Malik model, if we use only one volume gradient and evaluate its norm by (5.3), we get a faster algorithm, but the noise on edges is not removed so well (see Fig. 8).

Figure 8: Comparing two ways of the gradient evaluation for the Perona-Malik equation, mentioned in Remark 2. Left: the noisy image. Middle: one volume gradient with a norm evaluated by (5.3) is used. Right: the four gradients evaluation and the diffusion coefficients (Fig. 3) and (3.9) method is used.

6 Numerical experiments—filtering of SAR images

In this section we explore the use of the presented algorithm in filtering of SAR images with the aim to detect landscape elements. First, let us note that SAR images are images of the Earth obtained by spaceborn or airborn radar systems. Because the resolution of the resulting image depends on the antenna's length which has its physical limits, the length of the antenna is increased synthetically with a help of electronic systems (SAR is an acronym for Synthetic Aperture Radar). These images are degraded by grainy multiplicative noise called speckle that alters the real intensity values of features: the lighter is the image the greater is the variance of the noise. Due to the speckle we can never rely on a single pixel value; one way how to solve this problem is spatial filtering (i.e. [10]). Our images contain measurements of the amplitude of the backscattered radiation. The details about the images are given in Remark 3.

Due to speckle we can never rely on a single pixel value: one way how to solve this problem is spatial filtering. Many filters to remove the speckle have been designed. However, these mostly statistical filters try to preserve features of the image with respect to backscattering coefficient or polarimetric properties and cannot be always successfully used in detection features like edges or boundaries of areas of interest etc. We explore the ability of the presented algorithm to remove noise in such a way that the edges in images representing boundaries are preserved as much as possible and the interior is smooth sufficiently for automatic edge detection and segmentation algorithms. To detect edges we use either the Canny edge detector based on the second derivative or we use the Fuzzy select tool for a selection based on a region growing method which is incorporated in Gimp (Gnu Image Manipulation Program)—well available software for work with images.

Because in the areas with stronger reflected intensity speckle is stronger, we logarithm the image, thus transforming the multiplicative noise into additive one, for which the algorithms mentioned above can be applied. The use of logarithm suppresses high returns much more than low returns. Averaging the logarithmic values, and then taking inverse logarithm is not identical to averaging of pixel values directly, so we do not preserve the radiometric properties of the image, but for edge detection, clusterization, and segmentation this algorithm appears to be useful.

To estimate the *stopping time* we used Gaussian Blur filter in Gimp to find the radius of the Gaussian to achieve a sufficient smoothness for selected subimages. The radius was transformed to the time like in Remark 1. At this time also the decrease of elements was slowed down significantly. Because the method is unconditionally stable we have no restriction on the time step. Though the larger time steps increase the number of iterations, they are faster. In Experiment 3, $\tau = 20$ leads to smoother edges than for $\tau = 10$, but the sharpening effect is smaller. The parameter ε_1, ε_2 and ε_3 depend on intensity differences we want to tolerate. The values introduced in Experiment 1 could be used in all experiments. The algorithms produce a family of images: at the end the selected time—the size and the number of time steps were tuned by a visual inspection.

Experiment 1 This experiment works on a relatively small detail of a SAR image (Fig. 9) to which the Perona–Malik diffusion by (3.14)–(3.16) is applied. First of all we can observe the poor quality of edges and then the smooth quality of the result. We can also see that in the grid the resulting edge is characterized by a relatively thin band of small elements which depends on setting ε_1, ε_2 and ε_3. Below the original image we see the region selected by the fuzzy select tool based on similarity: we determine the seed by clicking in a region we want to select and set the threshold for admissible intensity difference. The resulting edge is unsmoothed, but such is also the SAR intensity information. This problem is discussed in the next experiment. The best results were achieved for the following setting of parameters:$K = 100$ after 15 time steps it was increased to 2000,$\tau = 10$ and to create the adaptive grid $\varepsilon_1 = 0.015$, $\varepsilon_2 = 0.02$ and $\varepsilon_3 = 0.005$.

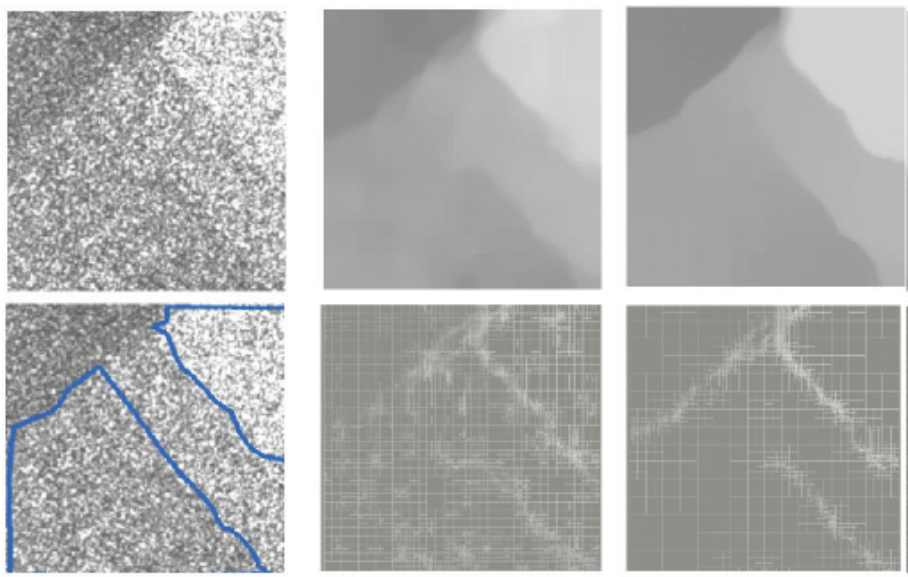

Figure 9: Left: the original data with the speckle, below border of detected regions are displayed as well. Middle: the data after 10 time steps, with the adaptive grid below. Right: the data after 30 time steps, with the adaptive grid below. The high value of K caused the sharpening effect.

Experiment 2 In this experiment we use the same algorithm to filter a similar detail of the same image, but this time we apply the edge detector based on the second derivative—the Canny edge detector [2], which looks for edge points in maxima of the norm of the gradient and produces thin edges. On the left of Fig. 10 and Fig. 11 we see the result of filtration and edge detection. The unsmoothed boundary is caused by the speckle and also by the fact that in the Perona Malik model the diffusion is slowed down on edges. However we expect

that the real boundaries are smooth so we apply the adaptive numerical scheme for the MCF (5.7)–(5.10) and let it work on the adaptive grid created in the last step of the diffusion. We see the results on Fig. 10 and Fig. 11 on the right. The boundaries are smoothed but because the gradient can be weakened some small parts of the edges can be lost.

Figure 10: Left: the image filtered by the Perona–Malik filter. Right: Applying of the mean curvature flow filter.

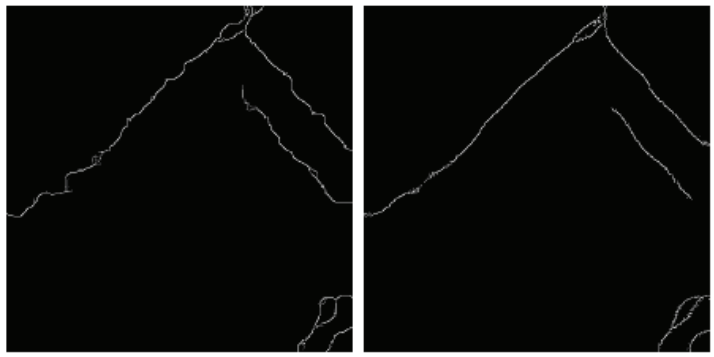

Figure 11: Canny edge detector applied to the image in Fig. 10 on the left and Canny edge detector applied to the same image smoothed by the mean curvature flow model in Fig. 10 on the right.

In the following experiments we work with two subsets of the SAR image displayed in Fig. 12 denoted as 1 and 2. The red dots frame the area of interest (AOI) and are results of in situ measurements performed in 2015. The field in bounded in the northeast (NE) by an asphalt road framed by trees, in the northwest (NW) by an alley of trees with grass below and the southeast (SE) side is bounded by various border types as trees, field roads and more agricultural fields. The different borders are essential for ability of our algorithms to detect the edges of landscape elements.

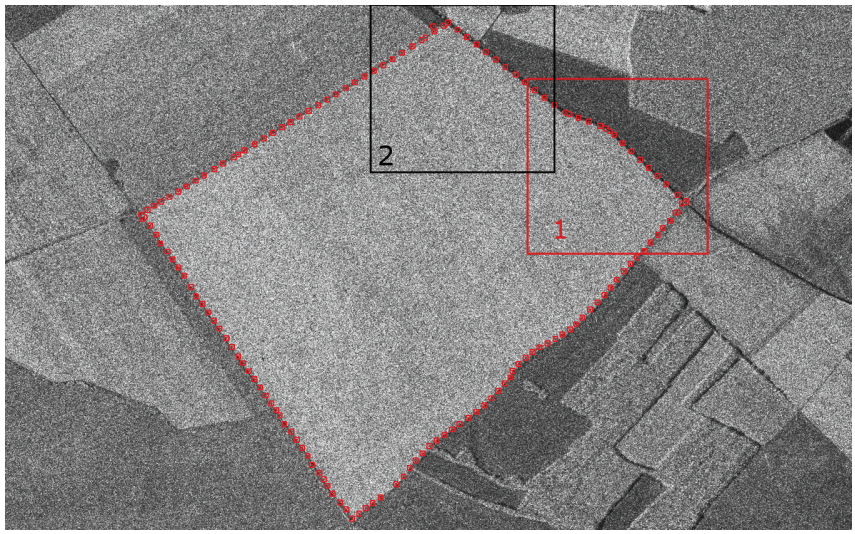

Figure 12: The data for Experiments 3 and 4 (the data is described in Remark 3). The area of interest (AOI)—the field described by red circles. They denote the positions obtained by insitu measurements. The marked subregion 1 denotes the sub-image for Experiment 4 and the subregion 2 for Experiment 3.

Experiment 3 In this experiment we work with a sub-image 2 (black square in Fig. 12) of the size 1024×1024. This sub-image 2 is filtered by the adaptive Perona–Malik filter (3.14)–(3.16) with the following setting of parameters: $T = 40$, $\tau = 20$, $K = 200$ and after 15 time steps it is set to 3000. The grid parameters are same as in the previous example. The results are depicted in Fig. 13. First, using the fuzzy select tool, we detect the area of AOI (the dotted region from Fig. 12.) We can see, that the NW border matches well with the measurements despite the low contrast of the neighboring regions. We consider this ability to be the strong side of our algorithm. The NE border if framed except of the asphalt road with trees. In the upper part the road is preserved though the intensity and width are not, in the second part of the border we see the shadow caused by trees and this is what is detected by the fuzzy select tool (see also Fig. 15).

Time step	1	3	5	8	10
No. of elements	1047367	299548	150658	100501	84148
% of elements	99.9	28.6	14.14	9.6	8.8
Time step	15	20	30	35	40
No. of elements	63952	54622	46069	43126	40762
% of elements	6.1	5.2	4.4	4.1	3.9

Table 1: The number of elements in the adaptive grid in particular time step for Experiment 3.

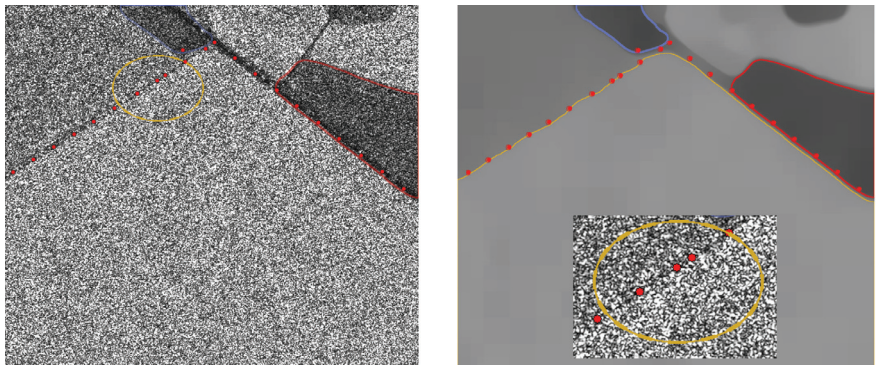

Figure 13: Left: The original image with red circles marking the measured points. Right: The detected AOI framed by a yellow line. The detail of the heavy noise on the northwest border, which is detected very well.

The detection of the road is better described in Experiment 4. The above thin line separating the regions of the same quality is not preserved at all—this algorithm in its basic version is not able to catch thin lines. In Table 1 we show number of elements in selected time steps. `http://www.math.sk/wiki/`

Experiment 4 In this experiment we filter the sub-image 1 with (3.14)–(3.16) (red square in Fig. 12). The result is depicted in Fig. 14. First, the detected boundaries of AOI are found by the fuzzy select tool (green line, on the right of Fig. 14). It confirms observations from the previous experiment. Then we tried to detect the road (yellow line, on the left of Fig. 14). It was detected by the fuzzy select tool—the seed was selected as a point on the road, the similarity threshold was low. We see that the road fits well with measured points.

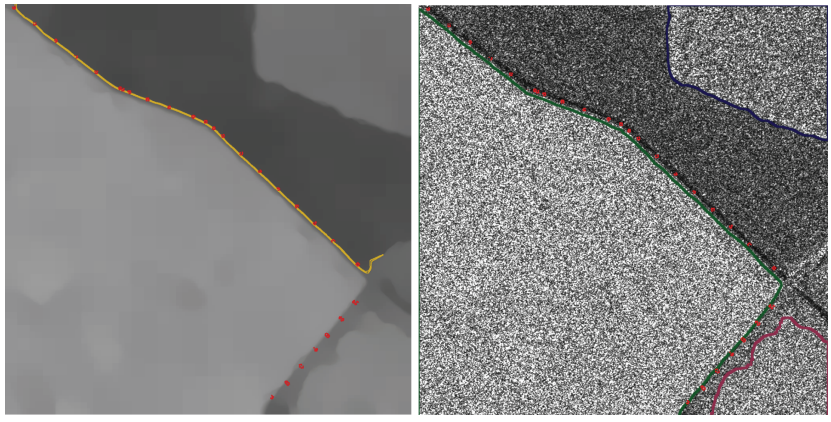

Figure 14: Left: The filtered image with detection of the road obtained by the fuzzy select tool (yellow line). Right: Detection of selected regions regions overlaying the original data.

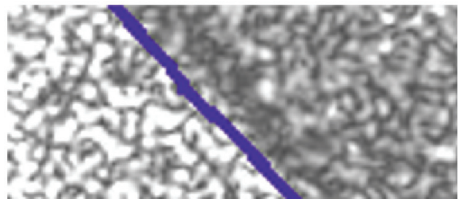

Figure 15: The detail of the road with trees (Experiment 4). The detected line shows what is detected if AOI is detected automatically.

Remark 3 The test image is a 4518×2793 pixel subset from a detected SAR image. Subset was separated from the dataset of the project ID LAN1583: Objekt Recognition Based on High-Resolution Radar Imagery, kindly supported by the German Aerospace Center (DLR) [12]. The image was acquired by the satellite TerraSAR-X (TSX) [13], which was launched in June 2007. The satellites carry a high frequency X-band SAR sensor (31 mm wavelength), which can be operated in flexible imaging modes in order to meet the requirements of versatile applications. Selected image was acquired on December 3, 2008 within Imaging Spotlight acquisition mode and single VV polarization in ascending pass direction. Spotlight imaging modes use phased array beam steering in azimuth direction to increase the illumination time, i.e. the size of the synthetic aperture. Spatial resolution is 1.2 m in slant range direction and 1.7 m in azimuth direction. The images correspond to the CEOS Level 1b quality and it was delivered in the Enhanced Ellipsoid Corrected (EEC) version which offer the highest level of geometric correction available for TerraSAR-X Basic Image Products (TerraSAR-X Image Product Guide, 2014). The performance of standard despeckling algorithms and adaptive Perona–Malik algorithm on this data has been studied in [15].

Conclusion We see the advantage of our algorithm in several aspects: we obtain the quad-tree representation of the image which is fine on edges and in details which have been preserved. The post-processing can be done on this structure The last steps of the algorithm are fast. From the last steps we can select that step which suits best the edge detector we want to used. Thanks to the fast diffusion on large regions we are able to preserve also edges between regions with a relatively small difference of a mean intensity, like e.g. the NW border of AOI. In such cases, also edge http://www.math.sk/wiki/detectors based on the first derivative were successful and produced thin edges. To mention other commonly used edge detectors, good localization of edges (and also detection of the road) was achieved with applying Difference of Gaussians.

Acknowledgement This work was supported by APVV-0072-11, VEGA 1/0714/15 and VEGA 1/0642/13. TerraSAR-X data for this work were provided by the Deutsches Zentrum für Luft- und Raumfahrt within the TSX-Archive-2012 project ID LAN1583: Object Recognition Based on High-Resolution Radar Imagery.

References

[1] Bänsch, E., Mikula, K.: *A coarsening finite element strategy in image selective smoothing*. Computing and Visualization in Science **1** (1997), 53–61.

[2] Canny, J.: *A computational approach to edge detection, in pattern analysis and machine intelligence*. IEEE Transactions on Pattern Analysis and Machine Intelligence **ol.PAMI-8**, 6 (1986), 679–698.

[3] Catté, F., Lions, P. L., Morel, J. M., Coll, T.: *Image selective smoothing and edge detectizon by nonlinear diffusion*. SIAM J. Numer. Anal. **29** (1992), 182–193.

[4] Eymard, R., Gallouët, T., Herbin, R.: *Finite volume method*. In: Handbook for Numerical Analysis **7**, *Elsevier*, Amsterdam, 2000, 713–1020.

[5] Eymard, R., Handlovičová, A., Mikula, K.: *Approximation of nonlinear parabolic equations using a family of conformal and non-conformal schemes*. Communications in Pure and Applied Analysis **11** (2012), 147–172.

[6] Eymard, R., Handlovičová, A., Mikula, K.: *Study of a finite volume scheme for the regularized mean curvature ow level set equation*. IMA Journal of Numerical Analysis **31** (2011), 813–846.

[7] Krivá, Z., Mikula, K.: *An adaptive finite volume scheme for solving non-linear diffusion equations in image processing*. J. Visual Communication and Image Representation **13** (2002), 22–35.

[8] Krivá, Z., Mikula, K.: *Adaptive diamond cell finite volume method in image processing*. In: Proc. Algoritmy 2009, Conf. on Scientific Computing (2009), 174–188.

[9] Krivá, Z., Mikula, K.: *A model and numerical scheme for processing of color images*. Journal of Electrical Engineering FEI STU **12** (2000), 21–25.

[10] Lee, J.-S., Pottier, L.: Polarimetric Radar Imaging: from Basics to Applications. *CRC Press*, Boca Raton, 2009.

[11] Osher, S., Sethian, J. A.: *Front propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation*. J. Comput. Phys. **29** (1988), 12–49.

[12] Papčo, J., Bakoň, M.: *Object Recognition Based on High Resolution Radar Imagery. German Aerospace Center (DLR)*, TSX-Archive-2012 project ID LAN1583, 2012.

[13] Airbus, Defence and Space: *TerraSAR-X Image Product Guide, Basic and Enhanced Radar Satellite Imagery*. (http://www2.geo-airbusds.com/files/pmedia/public/r459 9 201408 tsxx-itd-ma-0009 tsx-productguide i2.00.pdf), 2014.

[14] Trubač, M.: Adaptívne algoritmy metódy konečných objemov aplikované na analýzu viackanálového obrazu. Diploma thesis, *STU*, Bratislava, 2014.

[15] Vanko, J.: Monitoring Landscape Changes Using Satellite Radar Imagery. Diploma thesis, *STU*, Bratislava, 2015.