COMPUTATIONAL METHODS BASED ON NONLINEAR DIFFUSION IN IMAGE PROCESSING

Olga Drblíková

1. Introduction

One kind of meaningful information is given by images. We obtain them by converting continuos signals into digital format. Image processing is an interesting topic of study because of the diversity of applications and used mathematical methods. The most important of them are radar, seismology, meterology, astronomy, ultrasonicimaging, microscopy, and many others.

Moreover, due to acquisition, storage, transmission and display, the images can degrade with respect to completness of information. Mathematicians and computer scientists try to find mathematical models and numerical schemes to produce images of higher quality, with enhanced sharpness, filtered out noise and reconstructed edges.

This paper consists of three basic parts. First we present important models used for image processing. It provides basic information on properties and development of models like the linear heat equation, anisotropic Perona-Malik diffusion and modified Perona-Malik model in the sense of Catté et al. and Weickert model. In the second part we derive semi-implicit and explicit finite volume schemes for Weickert and both Perona-Malik models. Finally in last part we present the results of few numerical experiments.

2. Review of some diffusion models in image processing

Image processing involves operations like image filtration, edge detection, deblurring, image enhancement, restoration, shape analysis, etc. In the last decades mathematicians tried to find well understandable models and relialable numerical schemes for image processing and to improve schemes of their forerunners to get a processed image more quickly and not so computionally expensive.

The goal of this section is to present some partial differential equation, based methods for image restoration. The important examples of numerical schemes based on partial differential equations for image filtering and enhancement are: the linear heat equation, original Perona-Malik scheme and modified Perona-Malik schemes. The image processing consists of two opposite requirements — filtering (smoothing, diffusion), where the noise has been removed and enhancement, where the structures like edges, corners and T-junctions have been preserved. The mathematicians try to determine constant regions of an image and regions which contain edges and then make filtering dependent on this observation.

The oldest and most investigated equation in image processing is probably the parabolic linear heat equation (see [10] and [24]). The most simple way to improve image by smoothing was made by convolution with Gaussian of increasing variance (corresponding to scale). It is well-known that convolution of the signal with Gaussians at each scale is equivalent to solving of the linear heat (diffusion) equation with the signal as initial datum (see [10] and [29]). If we embed the original data u_0 into a family of gradually simplified versions of u_0, which depend on scale (or time parameter $t, t \geq 0$), then the linear multiscale analysis associated with u_0

consists in solving the system

$$u_t(x,t) = \Delta u(x,t) \quad (1)$$
$$u(x,0) = u_0(x). \quad (2)$$

This equation has the solution for the initial datum with bounded quadratic norm

$$u(x,\tilde{t}) = \begin{cases} u_0(x), & t=0, \\ (G_{\tilde{t}} * u_0)(x) & t>0, \end{cases} \quad (3)$$

where

$$G_{\tilde{t}}(x) = \frac{1}{(4\pi\tilde{t})^{\frac{d}{2}}} exp\left(\frac{-|x|^2}{4\tilde{t}}\right) \quad (4)$$

is the Gauss function and the convolution of the functions $f(x)$ and $g(x)$ is defined by

$$f(x) * g(x) = \int_{R^d} f(s)g(x-s)ds. \quad (5)$$

Though the heat equation has been (and is) successfully applied in image processing, it has some drawbacks. It is too smoothing (see [2]). Due to this diffusion model is not convenient for images, where we require good visual impression or where a precise location of edges is necessary because the Gaussian smoothing blurs edges and moves their positions. Since these linear diffusion shortcomings were too restrictive, the mathematicians have switched to nonlinear diffusion models.

If the partial differential equation for image processing is of nonlinear type, then the associated scale space is called nonlinear. One can study the axioms and fundamental properties of scale spaces in [1], where the notion of image multiscale analysis has been introduced.

First application of the nonlinear diffusion equations is connected with the end of the 80s. Perona and Malik (see [20]) introduced equation

$$u_t - \nabla \cdot (g(|\nabla u|)\nabla u) = 0 \quad (6)$$
$$u(0,x) = u_0(x), \quad (7)$$

where g is a smooth nonincreasing function, $g(0) = 1$, $g(s) > 0$, $g(s) \to 0$ for $s \to \infty$. In the computer vision community this equation is well-known as original Perona-Malik model or also as so-called anisotropic diffusion. The most significant advantage of this model is a better edge enhancement in comparing with the convolution with Gaussians.

Diffusion in this model is governed by the shape of the diffusion coefficient given by the function g and by its dependence on the edge indicator ∇u. Smoothing with the help of anisotropic diffusion is different in the regions where the signal has only small variance of intensity and in the regions of strong change of signal.

This scheme requires a suitable choice for function g. In practice, e.g.,

$$g(s) = \frac{1}{1+Ks^2}, \quad (8)$$

$$g(s) = e^{-s} \quad (9)$$

or

$$g(s) \;=\; \frac{1}{1+\frac{s}{K}}, \tag{10}$$

with a constant $K > 0$ are used (see [16] and [2]). If $s = |\nabla u(x)|$ is small then $g(|\nabla u(x|)$ tends to 1 and diffusion will be strong in a neighbourhood of a point x. In the contrary, if $s = |\nabla u(x)|$ is large then $g(|\nabla u(x)|$ is small and diffusion is low. This idea ensures accurate detection of edges.

Although Perona-Malik model behaves better than the heat equation, it still has some disadvantages. The noise induce too large oscilations of the ∇u and diffusion introduced by the model will not help, since all these noisy edges will be kept. Further, if the product $g(s)s$ is decreasing, the Perona-Malik equation can behave locally like the backward heat equation, which is ill-posed, and we may obtain unstable process. So, for g used in practice (see (10) and (9)) both the existence and the uniqueness of a solution can not be obtained. This problem has been studied in [9].

One way how to overcome these mathematical difficulties has been proposed by Catté, Lions, Morel and Coll (see [3] and [8]). They introduced the convolution with the Gaussian kernel $G_{\tilde{t}}$ into the decision process for the value of the diffusion coefficient. The form of this model is given as

$$u_t - \nabla \cdot (g(|\nabla G_{\tilde{t}} * u|)\nabla u) \;=\; 0 \quad \text{in } Q_T \equiv I \times \Omega \tag{11}$$

$$\frac{\partial u}{\partial n} \;=\; 0 \quad \text{on } I \times \partial\Omega \tag{12}$$

$$u(x,0) \;=\; u_0 \quad \text{in } \Omega, \tag{13}$$

where $\Omega \subset R^d$ is a rectangular domain, n is the normal unit vector to $\sigma = K|L$ outward to K, $I = [0,T]$ is a scaling interval, g is a decreasing function, $g(0) = 1$, $0 < g(s) \to 0$ for $s \to \infty$, $g(\sqrt{t})$ is smooth. $G_{\tilde{t}} \in C^\infty(R^d)$ is a smoothing kernel with $\int\limits_{R^d} G_{\tilde{t}}(x)dx = 1$, $G_{\tilde{t}}(x) \to \delta_x$ for $\tilde{t} \to 0$, where δ_x is the Dirac measure at point x and $u_0 \in L^2(\Omega)$ (see [16] and [17]).

This model is very convenient because it connects ideas of linear and nonlinear scale space equations. This modification allowed to prove the existence and the uniqueness of solution for the modified Perona-Malik equation. Other important advantage is that this model also keep the practical favourable properties of the original Perona-Malik model. The convolution with the prescribed $\tilde{t}$ offers a unique way to compute gradients of piecewise constant image.

The other generalization of the Perona-Malik equation has been introduced by Weickert (see [25], [26], [27] and [28]). This equation has a form

$$u_t - \nabla \cdot (D\nabla u) \;=\; 0, \tag{14}$$

where D is a matrix depending on the eigenvalues and on the eigenvectors of the (regularised) structure tensor $J = \nabla u(\nabla u)^T$ (for details see next sections). This modification is useful in any situation, where is diserable strong smoothing in one direction and low smoothing in the perpendicular direction. Moreover, Preusser and Rumpf also introduced this idea in multiscale flow field visualization in the computional fluid dynamics (see [22]) and Mikula, Preusser and Rumpf consider a similar idea in processing of image sequences (see [18] and [21]).

In image processing we use a structure descriptor. A very simple structure descriptor is given by $\nabla u_{\tilde{t}}$, where

$$u_{\tilde{t}}(x,t) \;=\; (G_{\tilde{t}} * u(\cdot,t))(x), \qquad (\tilde{t} > 0). \tag{15}$$

We can use $\nabla u_{\tilde{t}}$ for detecting edges in some images (see [3] and (11)) but for images with parallel structures this structure descriptor is unuseful. We know that for small $\tilde{t}$ high fluctuations remain, while larger $\tilde{t}$ leads to entirely useless results. This is due to the fact that for larger $\tilde{t}$ neighbouring gradients with same orientation, but opposite sign cancel each other. We need the structure descriptor invariant under sign changes, so we replace $\nabla u_{\tilde{t}}$ by its tensor product

$$J_0(\nabla u_{\tilde{t}}) = \nabla u_{\tilde{t}} \otimes \nabla u_{\tilde{t}} = \nabla u_{\tilde{t}} \nabla u_{\tilde{t}}^T \tag{16}$$

The matrix J_0 is symmetric and positive semidefinite and its eigenvectors are parallel and orthogonal to $\nabla u_{\tilde{t}}$, respectively. We can average J by applying other convolution with Gaussian G_ρ

$$J_\rho(\nabla u_{\tilde{t}}) = G_\rho * (\nabla u_{\tilde{t}} \otimes \nabla u_{\tilde{t}}), \qquad (\rho \geq 0). \tag{17}$$

In computer vision community the matrix

$$J_\rho = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

is well-known as structure tensor or interest operator or second moment matrix. Its exploitation is possible to find in many tasks, for example in analysis of flow-like textures (see [23]), corners and T-junctions (see [5] and [19]), shape cues (see [15]) and also spatio-temporal image sequences (see [6]).

This matrix J_ρ is symmetric and positive semidefinite and its eigenvalues are given as follows

$$\mu_{1,2} = \frac{1}{2}\left((a+c \pm \sqrt{(a-c)^2 + 4b^2})\right), \qquad \mu_1 \geq \mu_2. \tag{18}$$

Since the eigenvalues integrate the variation of the grey values within a neighbourhood of size $O(\rho)$, they describe the avarage contrast in the eigendirections v and w. The integration scale ρ reflects the characteristic size of the texture and in the most cases, it is large in comparison to the noise scale $\tilde{t}$.

With the help of the eigenvalues of J_ρ we can obtain useful information on the coherence of a structure. The expression $(\mu_1 - \mu_2)^2$ is large for anisotropic structures and tends to zero for isotropic structures. We can also identify kind of the image structures. Constant areas are characterized by $\mu_1 = \mu_2 = 0$, straight edges by $\mu_1 \gg \mu_2 = 0$ and corners by $\mu_1 \geq \mu_2 \gg 0$.

The corresponding orthonormal set of eigenvectors (v, w) to eigenvalues (μ_1, μ_2) is given by

$$v = (v_1, v_2), \qquad w = (w_1, w_2), \tag{19}$$

$$v_1 = 2b, \qquad v_2 = c - a + \sqrt{(a-c)^2 + 4b^2},$$

$$w \perp v, \qquad w_1 = -v_2, \qquad w_2 = v_1,$$

The orientation of the eigenvector w, which corresponds to the smaller eigenvalue μ_2 is called coherence orientation. This orientation has the lowest fluctuations.

Since we have a tool for analysing coherence, we draw our goals to enhance of image coherence. One of possibilities, how to do it, can be done by embedding the structure tensor analysis into a nonlinear diffusion filter.

The idea of nonlinear diffusion filtering is as follows. We get a processed version $u(x,t)$ of an image $u_0(x)$ with a scale parameter $t \geq 0$ as the solution of diffusion equation

$$u_t = div(D\nabla u) \tag{20}$$

$$u(x,0) = u_0(x) \tag{21}$$

$$< D\nabla u, n >= 0, \tag{22}$$

where n is the outer normal unit vector and $< \cdot, \cdot >$ the usual Euclidean scalar product. D is a positive definite 2×2 matrix. It is called diffusion tensor because it steers the diffusion process and its eigenvalues determine the diffusivities in the directions of the eigenvectors. For enhancing coherence, D must steers a filtering process such that diffusion is strong mainly along the coherence direction w and it increases with the coherence $(\mu_1 - \mu_2)^2$. To obtain it, we require that D must possess the same eigenvectors v and w as the structure tensor $J_\rho(\nabla u_{\tilde{t}})$ and we choose the eigenvalues of D as

$$\begin{aligned} \lambda_1 &= \alpha, \quad \alpha \in (0,1), \alpha \ll 1, \\ \lambda_2 &= \begin{cases} \alpha, & \text{if} \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{(\mu_1-\mu_2)^2}\right), C > 0 & \text{else.} \end{cases} \end{aligned}$$

The matrix D has a form

$$D = ABA^{-1}, \tag{23}$$

where

$$A = \begin{pmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{pmatrix}$$

$$B = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

We use the exponential function because it ensures that the smoothness of the structure tensor carries over to the diffusion tensor and that λ_2 does not exceed 1. The positive parameter α guarantees that the process never stops. Even if $(\mu_1 - \mu_2)^2$ tends to zero so the structure becomes isotropic, there still remains some small linear diffusion with diffusivity $\alpha > 0$. Such α is a regularization parameter, which keeps the diffusion tensor uniformly positive definite. C has a role of a treshold parameter. Since if $(\mu_1 - \mu_2)^2 \gg C$ then $\lambda_2 \approx 1$ and if $(\mu_1 - \mu_2)^2 \ll C$ then $\lambda_2 \approx \alpha$.

3. Semi-implicit finite volume scheme for anisotropic diffusion

The aim of this section is the derivation of numerical schemes for the following equation

$$\frac{\partial u}{\partial t} - \nabla \cdot (D\nabla u) = 0, \tag{24}$$

where $D \equiv D(u)$ denotes the diffusion tensor and u denotes an unknown function $u(x,t)$. In our application D may depend on this solution u, however, we will consider also the case that it is a given matrix. Let us consider a rectangular image domain $\Omega = (0, n_1) \times (0, n_2)$ and let an image $u(x)$ be represented by a bounded mapping $u : \Omega \to R$. Our image is represented by $n_1 \times n_2$ pixels (finite volumes) such that it looks as mesh with n_1 rows and n_2 columns (see Fig.1).

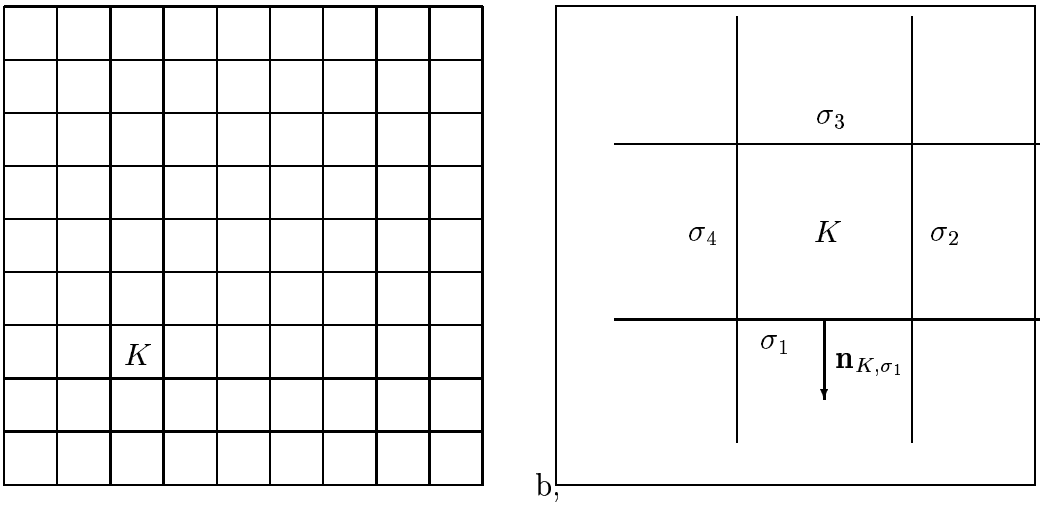

Fig.1 a) A mesh consists of pixels (finite volumes) K. b) A detail of this mesh - a finite volume K and its boundary $\sigma = \cup\sigma_i, i = 1, 2, 3, 4$.

We consider it in a scaling (time) interval $I = [0, T]$. Let $0 = t_0 \leq t_1 \leq \cdots \leq t_{N_{max}} = T$ denote the time discretization with $t_n = t_{n-1} + k$, where k is the time (scale) step. For $n = 0, \ldots, N_{max}$ we will look for u^n an approximation of a solution u at time t_n. By integrating equation (24) over a control volume K, we obtain

$$\int_K \frac{\partial u}{\partial t} dx - \int_K \nabla \cdot (D \nabla u) dx \quad = \quad 0. \tag{25}$$

We provide a semi-implicit in time discretization and use a divergence theorem to get

$$\frac{u_K^n - u_K^{n-1}}{k} m(K) - \sum_{\sigma \in \varepsilon_K} \int_\sigma D_K^{n-1} \nabla u^n \cdot \mathbf{n}_{K,\sigma} ds \quad = \quad 0, \tag{26}$$

where u_K^n is a value of K in time t_n, $m(K)$ is the measure of the control volume K with boundary ∂K, σ is an edge of the control volume K, ε_K is a subset of ε such that $\partial K = \bigcup_{\sigma \in \varepsilon_K} \sigma$, $\varepsilon = \bigcup_{K \in \tau} \varepsilon_K$, where τ_h is admissible finite volume mesh (see [4]), D_K^{n-1} is a mean value of $D^{n-1} \equiv D(u^{n-1})$ on K, that is $D_K^{n-1} = \frac{1}{m(K)} \int_K D^{n-1} dx$ and $\mathbf{n}_{K,\sigma}$ is the normal unit vector to σ outward to K. We denote an approximation of $\int_\sigma -D_K^{n-1} \nabla u^n \cdot \mathbf{n}_{K,\sigma} ds$ by $F_{K,\sigma}$. Then we follow the derivation of the scheme given in [4] and get subsequently

$$\frac{u_K^n - u_K^{n-1}}{k} m(K) + \sum_{\sigma \in \varepsilon_K} F_{K,\sigma} \quad = \quad 0, \tag{27}$$

where $F_{K,\sigma}$ is an auxiliary unknown, namely the flux for control volume K and for $\sigma \in \varepsilon_K$. We have choosen expresion for $F_{K,\sigma}$ in the following form (see [4])

$$F_{K,\sigma} \quad = \quad -m(\sigma) \lambda_{K,\sigma} \frac{u_\sigma - u_K}{d_{K,\sigma}}, \tag{28}$$

$$\lambda_{K,\sigma} = |D_K^{n-1}\mathbf{n}_{K,\sigma}|, \tag{29}$$

where $m(\sigma)$ is the measure of edge σ, u_σ is an auxiliary unknown, namely some expected approximation of u in σ, for all $\sigma \in \varepsilon_K$ and $d_{K,\sigma}$ is the Euclidean distance from x_K to $y_0 = D_{K,\sigma} \bigcap \sigma$. $x_K \in \overline{K} = K \bigcup \partial K$ and if $\sigma = K|L = K \bigcap L \neq \emptyset$, then it is assumed that $x_K \neq x_L$ and that the straight line $D_{K,L}$ going through x_K and x_L is orthogonal to $K|L$. $L \in \tau_h$ is the control volume, such that $\exists\ \sigma \in \varepsilon_K$, $\sigma = K|L$. $D_{K,\sigma}$ is a straight line, going through x_K and orthogonal to σ such that $D_{K,\sigma} \bigcap \sigma \neq \emptyset$. Then we have

$$\frac{u_K^n - u_K^{n-1}}{k} m(K) + \sum_{\sigma \in \varepsilon_K} -m(\sigma)\lambda_{K,\sigma} \frac{u_\sigma - u_K}{d_{K,\sigma}} \quad = \quad 0. \tag{30}$$

In order to have a conservativity of the scheme, we set $F_{K,\sigma} = -F_{L,\sigma}$. Using this condition we can express u_σ from (28) to get

$$u_\sigma = \frac{1}{\frac{\lambda_{K,\sigma}}{d_{K,\sigma}} + \frac{\lambda_{L,\sigma}}{d_{L,\sigma}}} \left(\frac{\lambda_{K,\sigma}}{d_{K,\sigma}} u_K + \frac{\lambda_{L,\sigma}}{d_{L,\sigma}} u_L \right). \tag{31}$$

Applying (31) in equation (28), we get

$$F_{K,\sigma} \quad = \quad -m(\sigma)\lambda_{K,\sigma} \frac{\frac{1}{\frac{\lambda_{K,\sigma}}{d_{K,\sigma}} + \frac{\lambda_{L,\sigma}}{d_{L,\sigma}}} \left(\frac{\lambda_{K,\sigma}}{d_{K,\sigma}} u_K^n + \frac{\lambda_{L,\sigma}}{d_{L,\sigma}} u_L^n \right) - u_K^n}{d_{K,\sigma}}. \tag{32}$$

Previous relation can be simplified to

$$F_{K,\sigma} = \tau_\sigma (u_K - u_L), \tag{33}$$

where

$$\tau_\sigma = m(\sigma) \frac{\lambda_{K,\sigma}\lambda_{L,\sigma}}{\lambda_{K,\sigma} d_{L,\sigma} + \lambda_{L,\sigma} d_{K,\sigma}}. \tag{34}$$

Applying (33) in equation (27) we have

$$\frac{u_K^n - u_K^{n-1}}{k} m(K) + \sum_{\sigma \in \varepsilon_K} \tau_\sigma (u_K^n - u_L^n) \quad = \quad 0, \tag{35}$$

Rearranging the previous equation, we obtain the semi-implicit scheme for equation (24)

$$\left[\frac{m(K)}{k} + \sum_{\sigma \in \varepsilon_K} \tau_\sigma \right] u_K^n - \sum_{\sigma \in \varepsilon_K} \tau_\sigma u_L^n \quad = \quad \frac{m(K)}{k} u_K^{n-1} \tag{36}$$

Since in 2D image processing we work with regular square mesh, we can simplify τ_σ by using following relations: $m(K) = h^2$, $m(\sigma) = h$, $d_{K,\sigma} = d_{L,\sigma} = \frac{h}{2}$, where h is a space step. Then we get

$$\tau_\sigma = 2 \frac{\lambda_{K,\sigma}\lambda_{L,\sigma}}{\lambda_{K,\sigma} + \lambda_{L,\sigma}} \tag{37}$$

In a similar way we can also derive an explicit scheme for solving equation (24). In this case instead of equation (35) we have

$$\frac{u_K^n - u_K^{n-1}}{k} m(K) + \sum_{\sigma \in \mathcal{E}_K} \tau_\sigma (u_K^{n-1} - u_L^{n-1}) \quad = \quad 0, \tag{38}$$

So we get

$$u_K^n \quad = \quad \left[1 - \frac{k}{m(K)} \sum_{\sigma \in \mathcal{E}_K} \tau_\sigma \right] u_K^{n-1} + \frac{k}{m(K)} \sum_{\sigma \in \mathcal{E}_K} \tau_\sigma u_L^{n-1}. \tag{39}$$

In these numerical schemes we need to compute the diffusion tensor D at every discrete time step t_n. We obtain it in the following way. First we calculate u_x and u_y for each pixel of the image u^{n-1} except of pixels on the boundary in the following form

$$u_x[i][j] = \frac{u^{n-1}[i][j+1] - u^{n-1}[i][j-1]}{2h} \tag{40}$$

$$u_y[i][j] = \frac{u^{n-1}[i+1][j] - u^{n-1}[i-1][j]}{2h}, \tag{41}$$

where i is a number of the row and j is a number of the column, in which the pixel is situated in the image and $u^{n-1}[i][j]$ is a value of grey level intensity of this pixel at $(n-1)th$ discrete time level. Then we construct a structure tensor J by outer product of these vectors

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} \begin{pmatrix} u_x & u_y \end{pmatrix},$$

so we get

$$J = \begin{pmatrix} (u_x)^2 & u_x u_y \\ u_x u_y & (u_y)^2 \end{pmatrix}.$$

Then the matrix D^{n-1} is constructed by (18), (19) and (23). Let us note that here as well as in our implementation of this scheme we omit convolutions mainly due to simplicity of presentation.

Since the tensor anisotropic diffusion is as well as the modified Perona-Malik model in sense Catté, Lions, Morel and Coll some generalization of the original Perona-Malik model, this semi-implicit finite volume schemes can be also used for both Perona-Malik models. The only difference is changing D into g.

Perona-Malik model has the following form

$$\frac{\partial u}{\partial t} - \nabla \cdot (g \nabla u) \quad = \quad 0, \tag{42}$$

where $g \equiv g(|\nabla u|)$ for the original Perona-Malik model
and $g \equiv g(|\nabla G_{\tilde{t}} * u|)$ for the modified Perona-Malik model by Catté et col.

In our numerical schemes we need to compute a value of the function $g(s)$ at every discrete time step t_n. There are several possible choices for $g(s)$ there. We

have choosen $g(s) = \frac{1}{1+Ks^2}$ with a constant $K > 0$ for our experiments.
The function g for an original Perona-Malik problem has a form

$$g(s) = \begin{pmatrix} \frac{1}{1+K(u_x)^2+K(u_y)^2} & 0 \\ 0 & \frac{1}{1+K(u_x)^2+K(u_y)^2} \end{pmatrix}.$$

We compute u_x and u_y by (40) and (41).
In case of the modified Perona-Malik model the convolution is realised with kernel $G_{\tilde{t}}(x) = \frac{1}{Z} exp\left(\frac{|x|^2}{|x|^2-(\tilde{t})^2}\right)$, where the constant Z is choosen so that $G_{\tilde{t}}$ has unit mass.
The function g has a form

$$g(s) = \begin{pmatrix} \frac{1}{1+K(sx)^2+K(sy)^2} & 0 \\ 0 & \frac{1}{1+K(sx)^2+K(sy)^2} \end{pmatrix}.$$

We obtain sx and sy in the following way. Using the convolution derivate property, we obtain

$$\nabla G_{\tilde{t}} * u(x_K) = \left(\frac{\partial}{\partial x_i}(G_{\tilde{t}} * u(x_K))\right)_{i=1,2} = \left(\frac{\partial G_{\tilde{t}}}{\partial x_i} * u(x_K)\right)_{i=1,2}.$$

Then we have

$$\frac{\partial G_{\tilde{t}}}{\partial x_i} * u(x_K) = \int_{R^2} \frac{\partial G_{\tilde{t}}}{\partial x_i}(s) u(x_K - s) ds$$

$$= \sum_M u_M(x_M) \int_M \frac{\partial G_{\tilde{t}}}{\partial x_i}(s) ds \approx \sum_M u_M(x_M) \frac{\partial G_{\tilde{t}}}{\partial x_i}(s) m(M),$$

where the sum is evaluated on control volumes M. These control volumes belong τ_h or they are contained in the reflexion of τ_h through boundary of Ω which are around x_K. This sum is also restricted to control volumes intersecting the circle centered at x_K with radius $\tilde{t}$. All control volumes which belong τ_h are denoted by K. Since in 2D image processing, we use uniform square mesh, we can express sx and sy by using following relation: $m(M) = h^2$ to get

$$sx = h^2 \sum_M u_M(x_M) \frac{\partial G_{\tilde{t}}}{\partial x}(s)$$

$$sy = h^2 \sum_M u_M(x_M) \frac{\partial G_{\tilde{t}}}{\partial y}(s).$$

4. Numerical experiments

This section presents some results of image filtering using the scheme from the previous section. The Figures show processing of the image with 100 $\times$ 100 pixels by the semi-implicit finite volume scheme for the Weickert equation (see Figures: 2, 3, 4) and with 50 $\times$ 50 pixels by the semi-implicit finite volume scheme for the Perona-Malik equations (see another Figures).

The original image consists of collection of parallel horizontal lines (see Fig.2a)). Figure 2b) depicts image degenerated by additive noise. Figure 3 and Figure 4 give examples of smoothed images, after 50 time steps (left) and after 100 time steps (right). Figure 3 shows the images which were obtained by linear tensor anisotropic diffusion with matrix coefficients based on original image only, i.e. $D \equiv D(u_0)$.

Figure 4 presents smoothing by nonlinear tensor anisotropic diffusion with updated coefficients during filtering i.e. by the model (20)–(23).

We observe that nonlinear model gives the image of much better quality, without blurring and very similar to the original image. In the scheme for tensor anisotropic diffusion we use $\alpha = 0.001$, $C = 1$ and the arising linear systems are solved by Gauss-Seidel iterative method.

Figure 5 presents the images before smoothing. The original image consists of a small black square in the middle of a grey square (see Fig.5a)). Figure 5b) depicts image degenerated by additive noise.

Figures 6 and 7 give examples of smoothed images obtained by diffusion with time step $t = 0.00005$ after 10 time steps (see Fig.6) and after 50 time steps (see Fig.7). Although the diffusion by the heat equation is expressive faster in comparing with the Perona-Malik equations, the images obtained by the heat equation are not so good as the images obtained by the Perona-Malik equations. The smoothing by the heat equation is isotropic, it does not depend on the image and it is the same in all directions. The edges are not preserved because of it. We can also see that a diffusion by using the modified Perona-Malik model is faster than by using the original Perona-Malik model.

In the scheme for the Perona-Malik model we use $K = 1$ and the arising linear systems are solved by Gauss-Seidel iterative method. We can also get similar results with the help of explicit finite volume schemes.

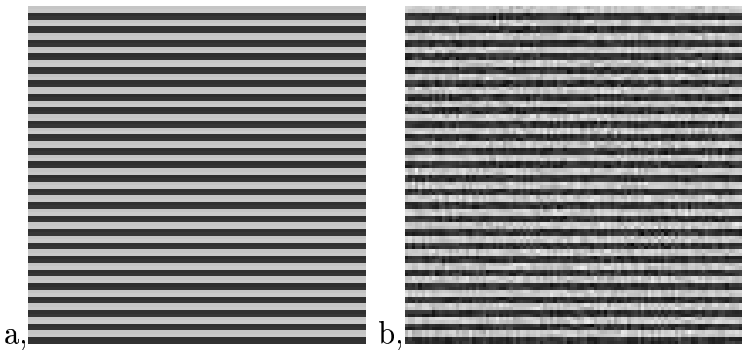

Fig.2 The images before filtering. a) an original image. b) a noisy image.

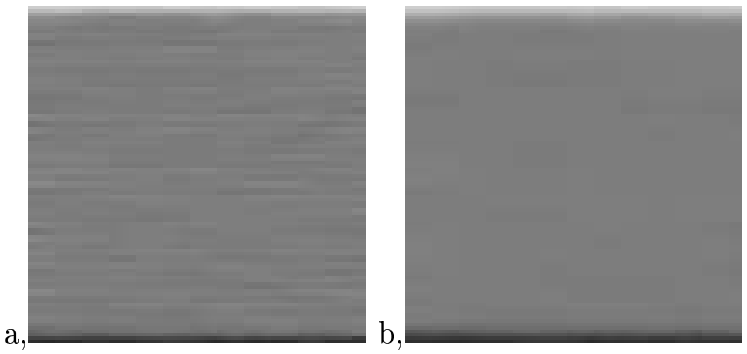

Fig.3 The images obtained by linear tensor anisotropic diffusion. a) after 50 time steps. b) after 100 time steps.

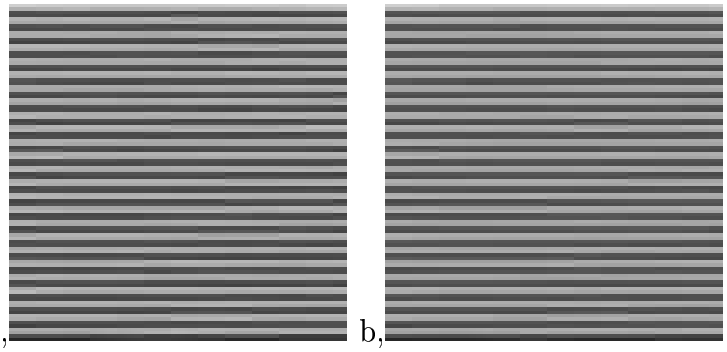

Fig.4 The images obtained by nonlinear tensor anisotropic diffusion. a) after 50 time steps. b) after 100 time steps.

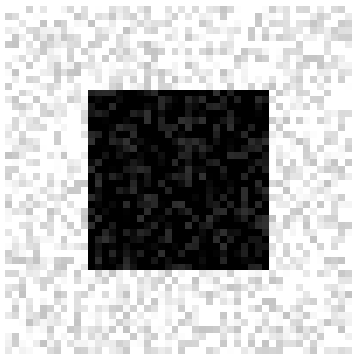

Fig.5 The image before filtering.

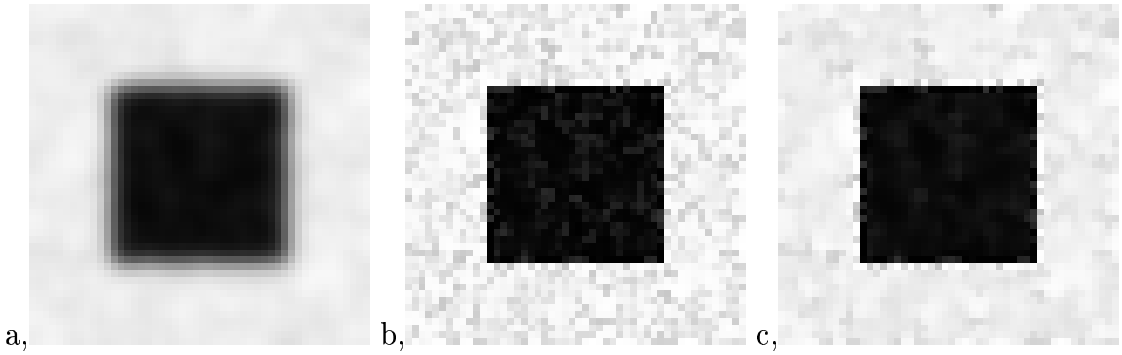

Fig.6 The filtered images after 10 time steps with time step $t = 0.00005$ obtained by a) the heat equation, b) the original Perona-Malik equation, c) the modified Perona-Malik equation with $\tilde{t} = 0.04$.

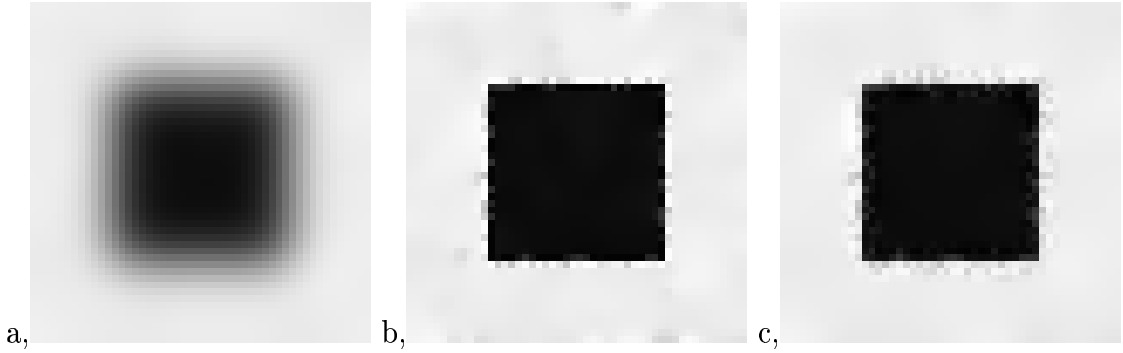

Fig.7 The filtered images after 50 time steps with time step $t = 0.00005$ obtained by a) the heat equation, b) the original Perona-Malik equation, c) the modified Perona-Malik equation with $\tilde{t} = 0.04$.

5. Conclusions

In this paper we have presented a derivation of the semi-implicit finite volume scheme for the Weickert and Perona-Malik model. We derive this scheme by semi-implicit finite volume scheme from [4]. Our scheme for the modified Perona-Malik model in sense Catté et al. is different from the semi-implicit finite volume scheme for the modified Perona-Malik model from [11], [12], [13], [14] and [16]. The authors compute the convolution in a centre of each edge and then they obtain four different $\lambda_{K,\sigma} = \tau_{K,\sigma}$. We realise the convolution in a point x_K (a centre of pixel K) and then we compute $\tau_{K,\sigma}$ as the harmonious average of $\lambda_{K,\sigma} = |g_K^{n-1} n_{K,\sigma}|$ and $\lambda_{L,\sigma}$ for each L, where L is an adjacent control volume of K.

We implemented our schemes to produce some filtered images. If we compare linear and nonlinear diffusion for the Weickert equation, we can see that nonlinear smoothing gives an image of expressive better quality. We can observe that although a smoothing with the help of the heat equation (a linear diffusion model) is faster in comparing with the Perona-Malik equations, the images obtained by the anisotropic diffusion are expressive better than by a smoothing in the linear diffusion model which is isotropic i.e. the same in all directions. Such filtration does not preserve edges. If we compare the smoothing by original and modified Perona-Malik model in sense Catté, Lions, Morel and Coll, we can see that the diffusion by modified Perona-Malik model is faster than by the original Perona-Malik model but the edges can be less sharper. It depends on a choice of $\tilde{t}$.

References

[**1**] Alvarez, L., Guichard, F., Lions, P. L., Morel, J. M.: Axioms and fundamental equations of image processing, Arch. Rational Mech. Anal. 123, 200–257, 1993.

[**2**] Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing (Partial Differential Equations and the Calculus of Variations) Applied mathematical sciences, Springer-Verlag New York, Inc., (2002).

[**3**] Catté, F., Lions, P. L., Morel, J. M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal. 129, 182–193, 1992.

[4] Eymard, R., Gallouët, T., Herbin, R.: Finite Volume Methods, in : Handbook for Numerical Analysis, Vol. 7 (Ph.Ciarlet, J.L.Lions, eds.), Elsevier, 2000.

[5] Förstner, M. A., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features, Proc. ISPRS Intercommission Conf. on Fast Processing of Photogrammetric Data (Interlakin, June 2–4, 1987), 281–305, 1987.

[6] Jähne, B.: Spatio-temporal image processing, Lecture Notes in Comp. Science, Vol. 751, Springer, Berlin, 1993.

[7] Kačur, J., Mikula, K.: Slow and fast diffusion effects in image processing, Computing and Visualization in Science 3(4), 185–195, 2001.

[8] Kačur, J., Mikula, K.: Solution of nonlinear diffusion appearing in image smoothing and edge detection, Appl. Numer. Math. 17, 47–59, 1995.

[9] Kichenassamy, S.: The Perona-Malik paradox, SIAM J. Appl. Math. 57, 1328–1342, 1997.

[10] Koenderink, J.J.: The structure of images, Biol. Cybern. 50, 363–370, 1984.

[11] Krivá, Z., Handlovičová, A.: Modified explicit finite volume scheme for Perona-Malik equation, Proceedings of ALGORITMY 2002 Conference on Scientific Computing, pp. 276–283.

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

[13] Krivá, Z., Mikula, K.: Adaptive finite volume schemes in processing of 3D data sets, J. Electrical Engineering 12/S, 52 (2001) pp. 53–58.

[14] Krivá, Z., Mikula, K.: A model and numerical scheme for processing of color images, Journal of Electrical Engineering 12/S, 51, (2000) pp. 21–25.

[15] Lindberg, T., Gårding, J.: Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure, Image and Vision Computing, Vol. 15, 415–434, 1997.

[16] Mikula, K., Ramarosy, N.: Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing, Numer. Math. 89 (3), 561–590, 2001.

[17] Mikula, K.: Image processing with partial differential equations, A. Bourlioux and M.J. Gander (eds.), Modern Methods in Scientific Computing and Applications, 283–321, 2002.

[18] Mikula, K., Preusser,T., Rumpf,M.: Morphological image sequence processing, to appear in Computing and Visualization, 2004.

[19] Nitzberg, M., Shiota, T.: Nonlinear image filtering with edge and corner enhancement, IEEE, Trans. Pattern. Anal. Mach. Intell., Vol. 14, 826–833, 1992.

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

[**21**] Preusser,T.,Rumpf,M.: Anisotropic nonlinear diffusion in flow visualization, In Proceedings Visualization 1999, 325–332, 1999.

[**22**] Preusser, T., Rumpf, M.: An adaptive finite element method for large scale image processing in: Proceedings of Scale Space '99, 223–243, 1999.

[**23**] Rao, A. R.; Schunck, B. G.: Computing oriented texture fields, CVGIP: Graphical Models and Image Processing, Vol. 53, 157–185, 1991.

[**24**] Romeny, B. M. t. H.: Geometry-driven diffusion in computer vision. Computational imaging and vision. Kluwer Academic Publishers, (1994).

[**25**] Weickert, J.: Coherence-enhancing diffusion of colour images. Image and Vision Computing 17, 201–212, 1999.

[**26**] Weickert, J: Coherence-enhancing diffusion of colour images, A. Sanfeliu, J.J.Villanueva, J. Vitrià (Eds.), Pattern Recognition and Image Analysis (VII NSPRIA, Barcelona, April 21–25, 1997), Vol. 1, 239–244,1997.

[**27**] Weickert, J.: Coherence-enhancing diffusion filtering, Int. J. Comput. Vision, Vol. 31, 111–127, 1999.

[**28**] Weickert, J.: Anisotropic Diffusion in Computer Vision, Teubner-Stuttgart, 1998.

[**29**] Witkin, A. P.: Scale space filtering, Proceedings of IJCAI, Karlsruhe, 1019–1021, 1983.

Author: Mgr. Olga Drblíková, Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 813 68 Bratislava,
e-mail: drblikov@math.sk