Image processing with partial differential equations

Karol MIKULA

Department of Mathematics and Descriptive Geometry
Slovak University of Technology
813 68 Bratislava
Slovakia

Abstract

In many applications computers analyse images or image sequences which are often contaminated by noise, and their quality can be poor (e.g. in medical imaging). We discuss how nonlinear partial differential equations (PDEs) can be used to automatically produce an image of much higher quality, enhance its sharpness, filter out the noise, extract shapes, etc. The models are based on the well-known Perona-Malik image selective smoothing equation and on geometrical equations of mean curvature flow type. Since the images are given on discrete grids, PDEs are discretized by variational techniques, namely by the semi-implicit finite element, finite volume and complementary volume methods in order to get fast and stable solutions. Convergence of the schemes to variational solutions of these strongly nonlinear problems and the extension of the methods to adaptive scheme strategies improving computational efficiency are discussed. Computational results with artificial and real 2D, 3D images and image sequences are presented.

1 Introduction

The aim of this paper is to present mathematical models, numerical methods and computational results in the processing of two-dimensional (2D), three-dimensional (3D) images and image sequences. The models which we use, are based on a partial differential equations (PDEs) approach. Namely, PDEs of nonlinear (degenerate) diffusion type are applied to initially given images. From the mathematical point of view, the input processed image can be modelled by a real function $u^0(x)$, $u^0 : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ represents a spatial domain. Typically Ω is rectangular and $d = 2$ or 3. In the case of an input image sequence $u^0(x, \theta)$, $u^0 : \Omega \times [0, T_A] \to \mathbb{R}$, it depends on the additional parameter θ representing a point in a real time interval of acquisition $[0, T_A]$.

Image processing operations involving PDEs are nonlinear image filtration, edge detection, deblurring and image enhancement, restoration, image inpainting, shape extraction and analysis, image segmentation, motion analysis, motion based filtering etc. [1, 59, 61, 8, 46, 55]. Typical 2D examples are given by a large variety of medical images, satellite or camera system images, old archive documents, texts pre-processed for automatic reading, old corrupted photographs or any other digital images of poor quality. 3D examples arise in bioengineering, medicine or in material quality control, where 3D volumetric aquisition methods are widely used nowadays. The processing of image sequences can be found in the restoration of movies, video sequence analysis, visual recording of growth (of human organs, leaves of plants, etc.) or in improvement of the quality of medical image sequences. For example, the ultrasound acquisition of a beating heart in 3D echocardiography gives one interesting application. This article is a review of models and methods which can be used in computational image processing. The details can be found in the original papers [26, 5, 42, 22, 56, 41, 27, 6, 33, 34, 23, 24, 45].

2 Nonlinear diffusion models in image processing

The first step to use PDEs for image processing was done in the beginning of eighties [32, 65]. By the simple observation that the Gauss function

$$G_\sigma(x) = \frac{1}{(4\pi\sigma)^{d/2}} e^{-|x|^2/4\sigma}$$

is a fundamental solution of the linear heat (diffusion) equation, it has been possible to replace the classical image processing operation—convolution of an image with G_σ with a given variance $v = \sqrt{2\sigma}$ (Gaussian smoothing)—by solving the linear heat equation for a corresponding time $t = \sigma$ with initial condition given by the processed image. It is well known that Gaussian smoothing (linear diffusion) blurs edges in the images and moves their positions. Although such a phenomenon can cause no problems in some examples of data analysis, in image processing, where the visual impression is important and a precise localization of edges is also necessary (e.g. to compute volumes of segmented objects), the linear image smoothing is generally not the best choice. A way has been found to overcome these shortcomings, namely to switch to nonlinear diffusion models.

Due to the evolutionary character of the process which controls the processing using diffusion equations, application of any PDE to an initially given image is understood as its embedding in the so-called *scale space*. In the case of nonlinear PDEs one speaks about *nonlinear scale space*. The axioms and fundamental properties of such embeddings have been given and studied in [1], where the notion of *image multiscale analysis* has been introduced. The image multiscale analysis associates with a given image $u^0(x)$ a family $u(t,x)$ of smoothed-simplified images depending on an abstract parameter $t \in [0,T]$, the *scale*. As has been proved in [1], if such a family fulfills certain basic assumptions—pyramidal structure, regularity and local comparison principle—then $u(t,x)$, $u : [0,T] \times \Omega \to \mathbb{R}$, can be represented as the unique viscosity solution (in the sense of [12]) of a general second order (degenerate) parabolic partial differential equation. This theoretical result has also an important practical counterpart. The equations of (degenerate) parabolic type have a smoothing property, so they are a natural tool for filtering (image simplification) by removing spurious structures, e.g. noise. Moreover, the simplification should be "image oriented", e.g. it should respect edges and not blur them. Or, it should recognize motion of a structure in an image sequence, and consequently the smoothing (diffusion) should respect the motion coherence in consecutive frames. Such requirements, or even more sophisticated ones related to the geometrical characteristics of the image, bring strong nonlinearity into the parabolic PDEs, and make this field interesting not only because of the applications but also from a mathematical and numerical point of view.

2.1 Anisotropic diffusion of Perona-Malik type

Since the end of the 80s, the nonlinear diffusion equations have been used for processing of 2D and 3D images. After the pioneering work of Perona and Malik [51] who modified the linear heat equation to nonlinear diffusion preserving edge positions, there has been a great deal of interest in the application and analysis of such equations. At present, the following nonlinear PDE [9] is widely used

$$u_t - \nabla.(g(|\nabla G_\sigma * u|)\nabla u) = 0, \tag{2.1}$$

where $u(t,x)$ is an unknown function defined in $Q_T \equiv [0,T]\times\Omega$. The equation is accompanied by zero Neumann boundary conditions and the initial condition

$$\frac{\partial u}{\partial \nu} = 0 \quad \text{on} \quad I \times \partial\Omega, \tag{2.2}$$

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

where ν is the unit normal vector to the boundary of Ω. We assume that $\Omega \subset \mathbb{R}^d$ is a bounded rectangular domain, $I = [0,T]$ is a scaling interval,

$$g : \mathbb{R}_0^+ \to \mathbb{R}^+ \text{ is a nonincreasing function,} \quad g(\sqrt{s}) \text{ is smooth,} \tag{2.4}$$

$$g(0) = 1, \quad \text{and we admit } g(s) \to 0 \text{ for } s \to \infty,$$

$$G_\sigma \in C^\infty(\mathbb{R}^d) \text{ is a smoothing kernel (e.g. the Gauss function),} \tag{2.5}$$

$$\int_{\mathbb{R}^d} G_\sigma(x)\,dx = 1, \qquad \int_{\mathbb{R}^d} |\nabla G_\sigma|\,dx \leq C_\sigma,$$

$$G_\sigma(x) \to \delta_x \text{ for } \sigma \to 0, \quad \delta_x \text{ is the Dirac measure at the point } x,$$

$$u^0 \in L_\infty(\Omega), \tag{2.6}$$

and

$$\nabla G_\sigma * u = \int_{\mathbb{R}^d} \nabla G_\sigma(x-\xi)\tilde{u}(\xi)\,d\xi, \tag{2.7}$$

where $\tilde{u}$ is an extension of u to $\mathbb{R}^d$. One can consider the extension of u by 0 outside Ω or the reflective periodic extension of the image [9].

The equation (2.1) represents a modification of the original Perona-Malik model [51, 29, 47]

$$u_t - \nabla.(g(|\nabla u|)\nabla u) = 0, \tag{2.8}$$

called also *anisotropic diffusion* in the computer vision community. Perona and Malik introduced (2.8) in the context of edge enhancement. The equation selectively diffuses the image in the regions where the signal has small variance in intensity in contrast with those regions where the signal changes its tendency. Such a diffusion process is governed by the shape of the diffusion coefficient given by the function g in (2.4) and by its dependence on ∇u, which is understood as an edge indicator [51]. Since $g \to 0$ for large gradients, the diffusion is strongly slowed down on edges, while outside them it provides averaging of pixel intensities as in the linear case. From a mathematical point of view, for practical choices of g (e.g. $g(s) = 1/(1+s^2)$, $g(s) = e^{-s^2}$), the original Perona-Malik equation can behave locally like the backward heat equation. It is, in general, an ill-posed problem which suffers from non-uniqueness and whose solvability is a difficult problem [29]. One way to overcome this mathematical disadvantage has been proposed by Catté, Lions, Morel and Coll in [9]. They introduced the convolution with the Gaussian kernel G_σ into the decision process for the value of the diffusion coefficient. Since convolution with the Gaussian is equivalent to linear diffusion, their model combines ideas of linear and nonlinear scale space equations. Such a slight modification made it possible to prove the existence and uniqueness of solutions for the modified equation, and to keep the practical advantages of the original formulation. Moreover, usage of the *Gaussian gradient* $\nabla G_\sigma * u$ combines the theoretical and implementation aspects of the model. The convolution (with prescribed σ) gives a unique way to compute gradients of a piecewise constant image. It also bounds (depending on σ) the gradient of the solution as input of the function g in the continuous model—which corresponds to the situation in numerical implementations where gradients evaluated on a discrete grid are finite. Also, the local edge enhancement is more understandable in the presence of noise.

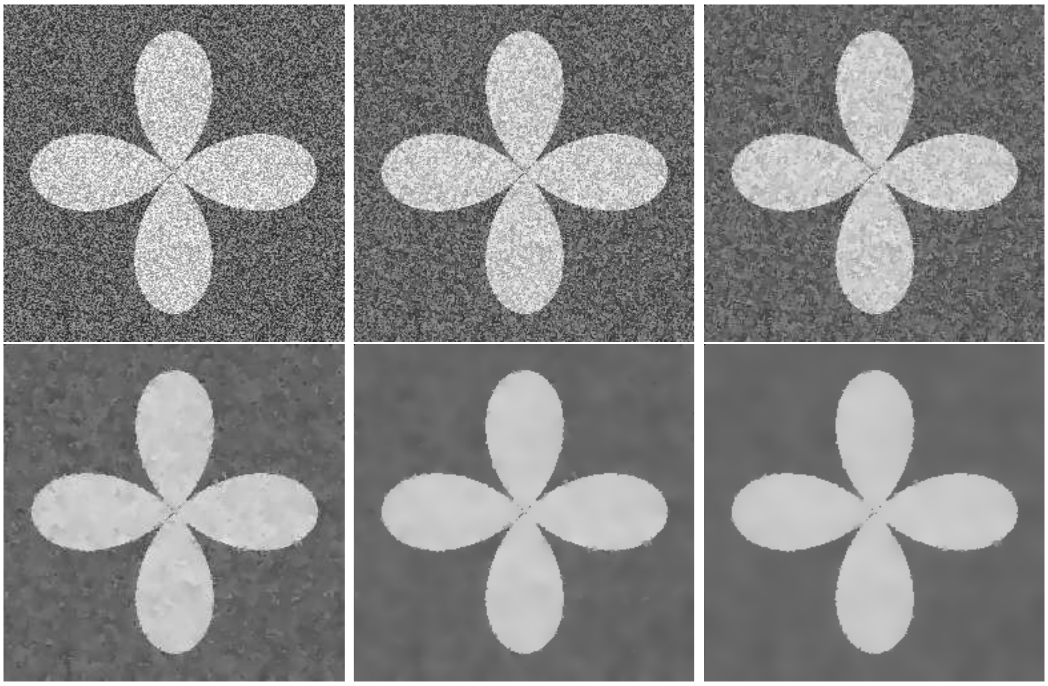

Figure 1: Smoothing of the noisy image keeping the edges using anisotropic diffusion. Shown are the 0th, 10th, 20th, 30th, 40th and 50th discrete steps of a semi-implicit finite volume algorithm [41].

We present applications of the regularized Perona-Malik equation (2.1) in the following three examples. First, the artificial image (Fig. 1, 256×256 pixels) with additive noise is processed by our finite volume scheme [41]; then nonlinear multiscale analysis of the medical image (Fig. 2, 463×397 pixels) computed by the co-volume discretization [24] is given; and finally, there is an application of the 3D adaptive finite element method [5, 6] to 3D echocardiographic image of one moment of the cardiac cycle with the left ventricle in open

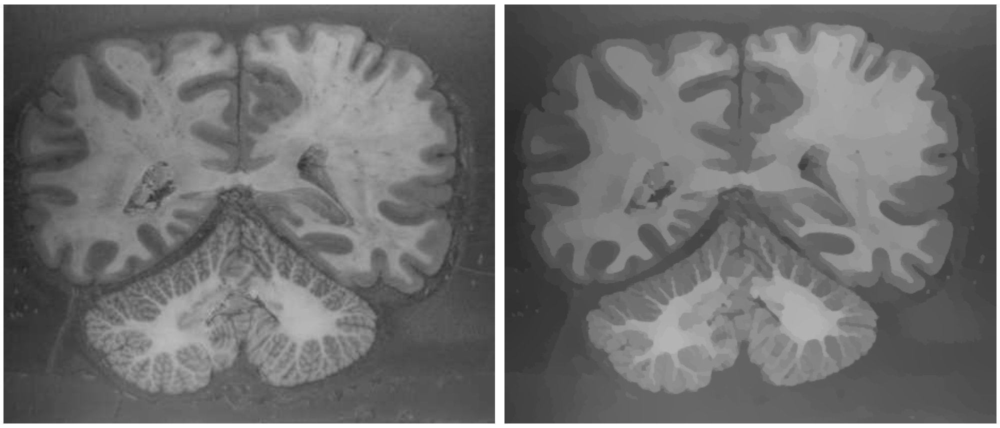

Figure 2: Initial image (left); result after 20 steps of regularized Perona-Malik filtering (right) using semi-implicit complementary volume discretization [24].

phase. In Fig. 3, one can see a visualization of the isosurface representing the boundary between blood and muscle forming an edge in 3D image intensity.

2.2 Curvature-driven equations in image processing

In rather general situations, the edges are related to level lines (level surfaces in 3D) of the image intensity function. For example, as one can see in Fig. 4, the edge representing the blood-cardiac muscle interface corresponds also to an isosurface of the greylevel image intensity function. In the three subfigures, the corresponding isolines are visible in 2D cutting slices. The isoline gives a curve in 2D plane representing the boundary of the left ventricle in the slice. The curve is non-smooth because of the errors in the acquisition. One would like to smooth it in order to remove the large acquisition errors. One way how to proceed is to move the curve (surface in 3D) in the direction of its inner normal with the velocity proportional to its (mean) curvature. The motion of convex and concave pieces is opposite due to the sign of the curvature, and the large fingers shrink much faster than the smoother parts, due to the curvature dependence of the flow. The motion by (mean) curvature is governed by a heat equation (Gaussian smoothing), but applied in the intrinsic curve (surface) geometry. In Fig. 5 we present the smoothing effect of such *geometrical diffusion*. On the left, the cuttings of unfiltered iso-surfaces are plotted, on the right the filtered ones (all after binarization with the same treshold). We can see an immediate extinction of small structures (noise) due to their high curvature, and a smoothing of the larger structures in the image.

The motion by curvature is used in numerous of applications related to the so-called free-boundary problems and interphase dynamics [59]. There exists a mathematical theory for such problems [16, 11]; also, numerical methods are available and in further development. From the computational point of view there are two main approaches for solving curve or surface evolution governed by curvature. In the so-called *Lagrangean approach* [14, 15, 43, 44, 45, 13], the curve, respectively surface, itself is discretized. Then a system of algebraic equations is derived for the new position of discrete points [14, 15, 43, 44, 31], or the equations are given for their intrinsic characteristics like curvature and tangential angle [40, 39, 45]. This system is solved to get the new curve (surface) position in the next discrete time step.

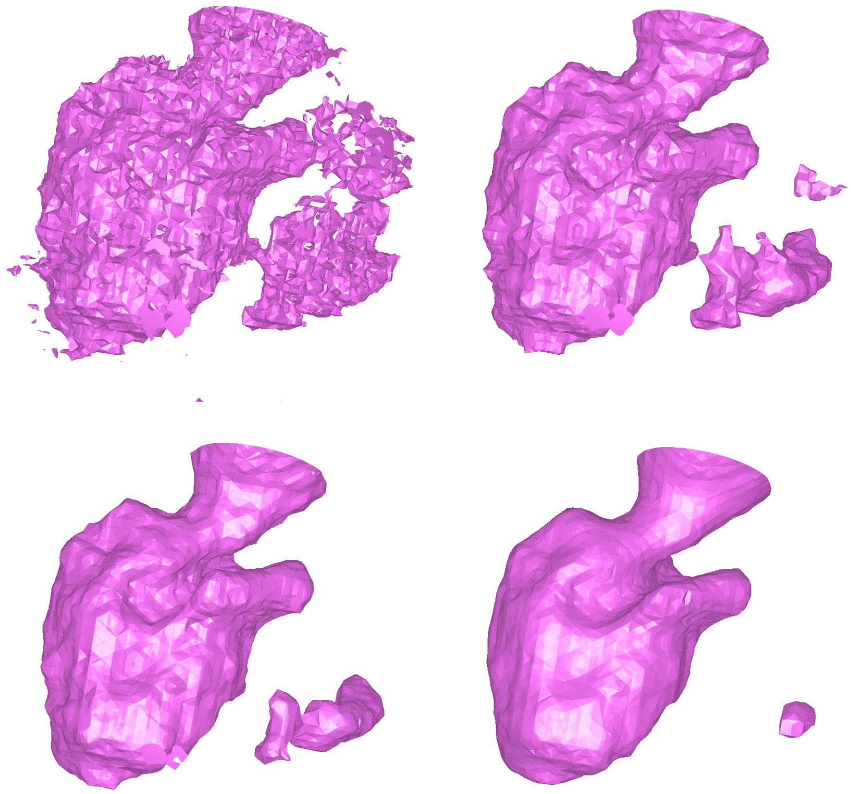

Figure 3: Smoothing of the human left ventricle by anisotropic diffusion. We visualize corresponding level surfaces in the 0th, 2nd, 4th and 8th discrete steps of the semi-implicit adaptive finite element algorithm [6].

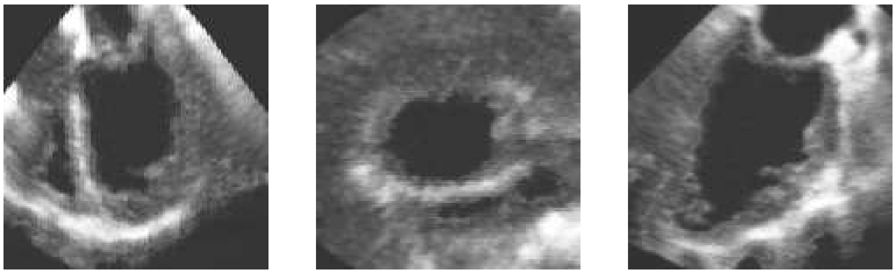

Figure 4: 2D orthogonal slices crossing in the centre of the 3D echocardiographic cube with the image of the human left ventricle.

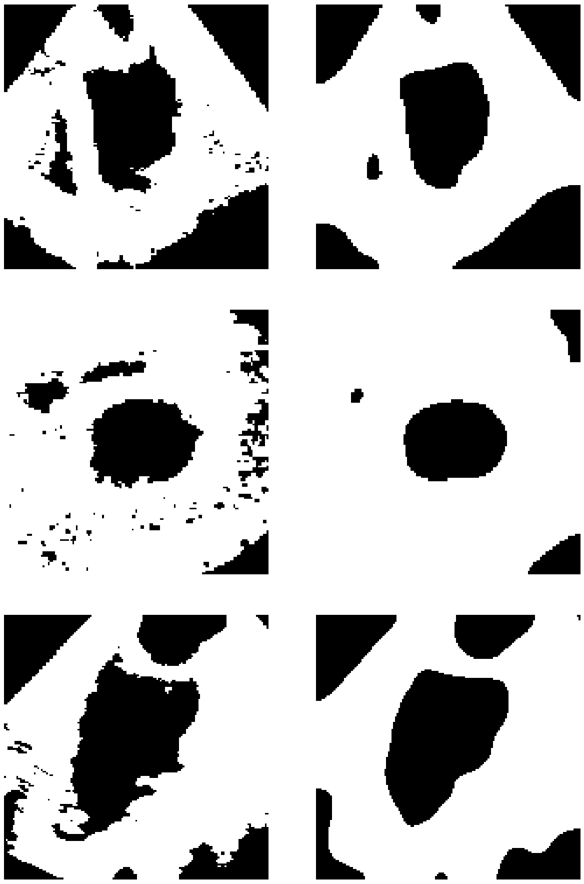

Figure 5: 2D cuts of the result of 3D processing by geometrical diffusion of mean curvature flow type.

In such a way numerically evolving curves are used e.g. in image segmentation [45]. The Lagrangean approach is an efficient and computationally fast method but, because of the parametric representation of the curves (surfaces), it can hardly handle the evolution through singularities, splitting and merging of the curves or surfaces during the evolution. In spite of that, the so-called *Eulerian approach* implicitly handles the curvature-driven motion by passing the problem to a higher dimensional space and solving there the evolution equation for a graph whose evolving level sets correspond to the evolving curve or surface. The *phase-field equations* (see e.g. [48, 7]) and *level set methods* [50, 58, 59] are approaches of that type. Especially, the *level set equation* of Osher and Sethian,

$$u_t = |\nabla u| \nabla . \left(\frac{\nabla u}{|\nabla u|} \right), \tag{2.9}$$

for curvature-dependent motion is well suited for image processing applications, since all geometrical information about image level lines or level surfaces is handled in once. As can be seen in Fig. 5 it provides denoising and silhouette smoothing at the same time. The level set equation not only moves each level set with the velocity proportional to its normal mean curvature field, but it also fulfills the so-called *morphological principle*: if u is a solution then, for any nondecreasing function φ, $\varphi(u)$ is a solution as well. This *contrast invariant* property has large significance in the axiomatic theory of image processing [1]. It also means that level sets move independently of each other; they diffuse only intrinsically, there is no diffusion across them in the normal direction. Thus it is a natural tool for directional smoothing of the image along its level lines. This idea was used in [3], where the equation

$$u_t = g(|\nabla G_\sigma * u|)|\nabla u| \nabla . \left(\frac{\nabla u}{|\nabla u|} \right) \tag{2.10}$$

has been suggested for computational image and shape analysis. It is accompanied by the same boundary and initial conditions (2.2)–(2.3) as in the case (2.1). Equation (2.10) can be used for image silhouettes smoothing ($g \equiv 1$, see e.g. [1, 2, 42, 22]), or it can be used for edge-preserving smoothing in a similar way as equation (2.1). The Perona-Malik function $g(s)$ depending on $|\nabla G_\sigma * u|$ is used to strongly slow down the motion of silhouettes which are at the same time edges. The regions between them are smoothed by the mean curvature flow.

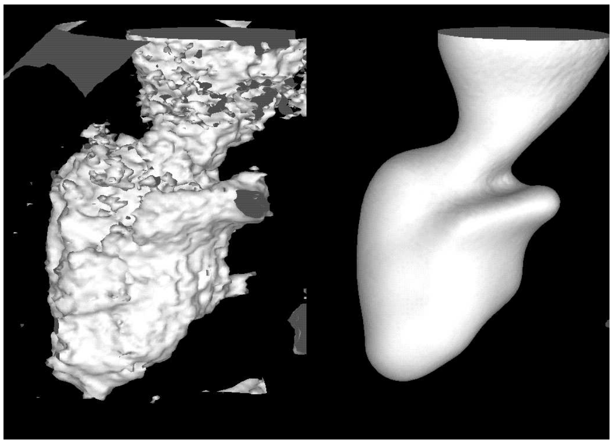

Figure 6: Ventricular shape extraction using the level set equation [42].

In Fig. 6 we present smoothing of the left ventricular iso-surface by the level set equation. In Fig. 7 we smooth an initial 321×373 pixel image (ancient coat-of-arms of the Slovak town Kremnica shown on the left), scanned from a book with neither paper nor colors of good quality. Also shown are the results after 5 and 10 discrete scale steps of the co-volume scheme for equation (2.10) with $g(s) = 1/(1+s^2)$. We also present Fig. 8, where two chromosomes are extracted from an initial noisy 3D image of a human cell by image selective smoothing (2.10) with the same g.

To end of this section, let us mention a general useful use of viewing gradient-dependent nonlinear diffusion equations. The nonlinear diffusion term $\nabla.(g(|\nabla u|)\nabla u)$ can be rewritten in 2D as

$$\nabla.(g(|\nabla u|)\nabla u) = g(|\nabla u|)u_{\xi\xi} + H'(|\nabla u|)u_{\eta\eta},$$

where $H(s) = sg(s)$ and ξ, η are tangential and orthogonal vectors to the level line, respectively. From this form one can clearly see how diffusion works along and across the image

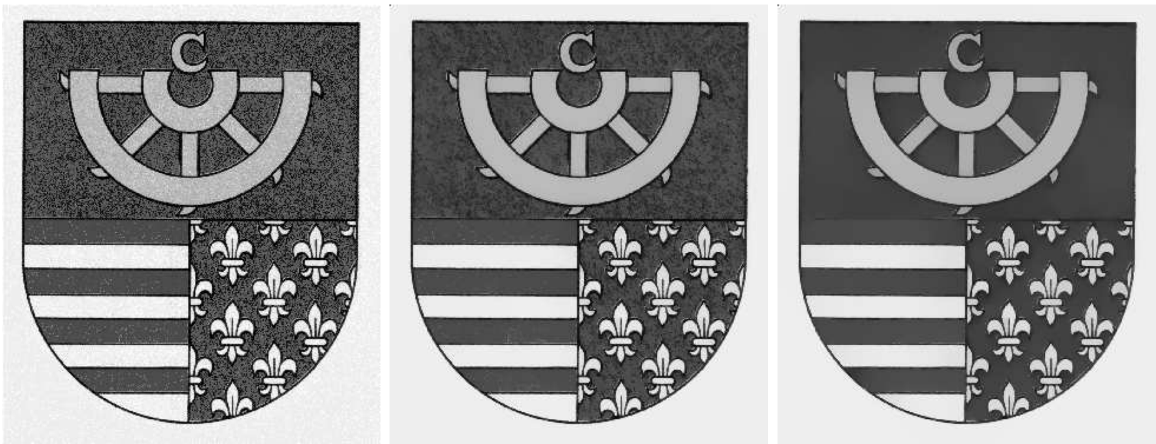

Figure 7: Initial image (left); result of smoothing after 5 (middle) and 10 (right) scale steps using equation (2.10) and co-volume discretization [23].

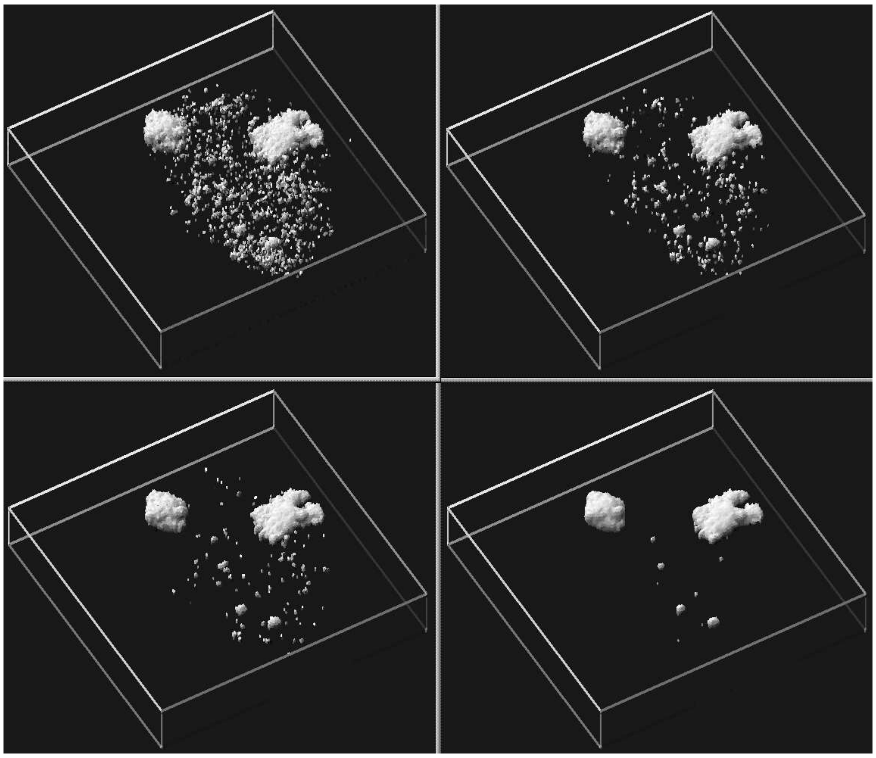

Figure 8: Extraction of two chromosomes in a human cell using geometrical diffusion (2.10) [22].

silhouettes with different choices of g. There is always positive, but possibly strongly slowed-down (depending on the shape of g) diffusion *along* level lines. *Across* level lines there can be forward diffusion (when $H'(s)$ is positive), zero diffusion (e.g. in the Rudin-Osher-Fatemi model [53] dealing with total variation denoising, and also in the mean curvature flow equation in the level set formulation), or backward diffusion (in the original Perona-Malik model [51, 47, 29]).

2.3 Some generalized models

The following generalization of the Perona-Malik equation has been introduced in [27]:

$$\partial_t b(x,u) - \nabla.(g(|\nabla G_\sigma * \beta(x,u)|)\nabla\beta(x,u)) = f(u^0 - u). \tag{2.11}$$

The functions b and β represent new nonlinearities which make the image multiscale analysis locally dependent on values of the intensity function u and on the position in the image x. Such a generalization is useful in any situation where properties of the image or requirements to the image processing operation can be expressed in dependence on x and u. For example, if a different speed of the diffusion process is desirable in different parts of the image or for different ranges of the intensity function, then equation (2.11) can be used. In the points, where the derivative β'_u is small (b'_u is large), the diffusion process is slowed down, while where β'_u is large (b'_u is small) the diffusion process is speeded up. Degenerate cases from the point of view of the theory of parabolic PDEs, when either β'_u or b'_u is equal 0 or ∞, can also be included. The degenerate cases can be interpreted as total stopping of diffusion, or as diffusion with the infinite speed in some image regions. Applying the regularized anisotropic diffusion (2.1) improves some set of edges. On the other hand, it destroys details which are under the edge threshold (given by g) or undistinguished from the noise at some scale. If such details are contained in certain ranges of greylevels, then they can be conserved by a special choice of the function β or b. As a demonstration we present Fig. 9. In that image, the colors of Flora's face are demaged only. We present the reconstruction of the original (left image) by anisotropic diffusion accompanied with the *slow diffusion effect* (image on the right). Using the proper choice of β (b is linear), which is constant for darker (lower) greylevels and linear for the upper range of u, the face is selectively smoothed and the details around it are conserved. For theory and numerical algorithm we refer to [27]. The right-hand side of (2.11) with nondecreasing function f can be used to force the solution to be close to original u^0 [49]. The Lipschitz continuous f causes no important difference in numerical analysis compared to the zero right-hand side.

Weickert (see e.g. [61, 63]) introduced a generalization of the Perona-Malik equation of the form

$$u_t - \nabla.(D\nabla u) = 0, \tag{2.12}$$

where D is a matrix depending on the eigenvalues and eigenvectors of the so-called *structure tensor* $\nabla u(\nabla u)^T$. The dependence is such that diffusion strongly prefers direction of line structures in the image. That idea has also been used by Preusser and Rumpf in multiscale flow field visualization in computational fluid dynamics [52].

There exist generalizations of the basic equations from the previous two subsections to the processing of color images. An RGB image can be viewed as a composition of three greyscale

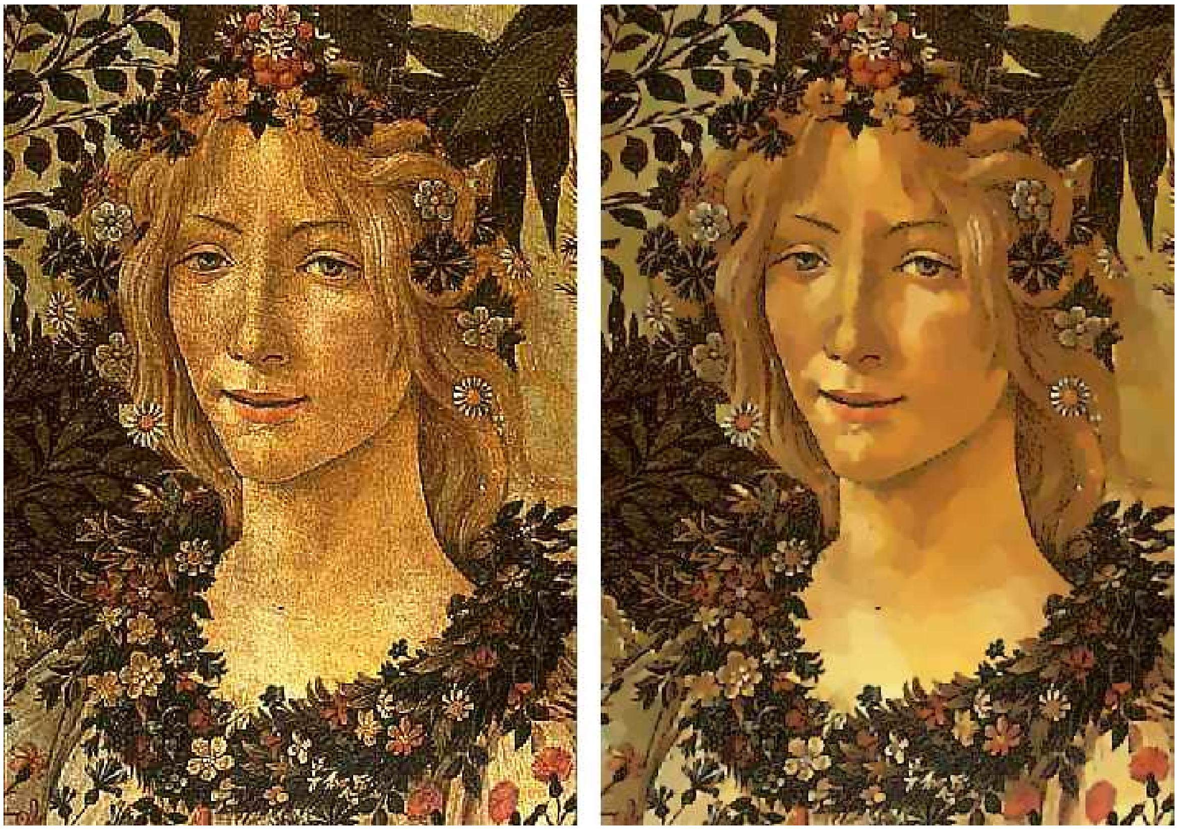

Figure 9: Processing of a color image using slowed anisotropic diffusion [27].

images representing the levels of intensity for red, green and blue colors. Then it is natural to consider a Perona-Malik-type system of equations adapted to the RGB image. The idea is not to apply the Perona-Malik anisotropic diffusion equation to each channel independently (which would be the simplest possibility), but to synchronize the diffusion in each channel by computing a common diffusion coefficient depending on the information coming from all three colors. In [64, 63, 34], dealing with color images and vector valued diffusion, the following system of nonlinear partial differential equations has been considered:

$$\partial_t u_i - \nabla.(d\nabla u_i) = 0, \quad i = 1, 2, 3, \tag{2.13}$$

$$d = g\left(\sum_{j=1}^{3} |\nabla G_\sigma * u_j|\right). \tag{2.14}$$

The equations are accompanied by initial and zero Neumann boundary conditions for each color. In the case (2.13)–(2.14), the edges of a highly destroyed channel can be recovered by information coming from the remaining channels [34]. A geometrical model based on minimal surfaces and curvature-driven motion for processing of color images is given in [30].

2.4 Space-time filtering algorithm

A given 3D space-time image sequence (e.g. in 3D echocardiography) $u^0(x,\theta)$ is a special 4D image where a motion coherence of subsequent frames is present. Usually, the aim is to extract relevant motion information from the sequence, filter out the noise, and enhance moving structures. To that end, it seems reasonable to use additional information (in comparison with still image processing) given by the motion correspondence in the image sequence.

We assume that certain objects acquired at different times, and thus being in different frames of the sequence, are formed by points that preserve their intensity along the motion trajectory. Such objects are called *Lambertian structures.* Moreover, we assume that motion is smooth in time, and thus the motion trajectories are close to straight lines locally. Designing the model we consider the following quantity [1, 2] proposed by Guichard [19]:

$$\begin{aligned} clt_u(t,x,\theta) = \min_{w_1,w_2} \frac{1}{(\Delta\theta)^2}(\Delta\theta\ |\langle \nabla u(t,x,\theta), w_1 - w_2\rangle| \\ + |u(t,x-w_1,\theta-\Delta\theta) - u(t,x,\theta)| \\ + |u(t,x+w_2,\theta+\Delta\theta) - u(t,x,\theta)|) \end{aligned} \tag{2.15}$$

where w_1, w_2 are arbitrary vectors in $\mathbb{R}^N$ and $\Delta\theta$ is a time increment. The scalar function clt_u will introduce a measure of coherence in time for the moving structures. It consists of the sum of three positive parts and we want to find the minimum in all possible directions w_1, w_2. The last two terms in the sum on the right-hand side of (2.15) are related to the differences in the intensities of end-points of the candidate Lambertian velocity vectors w_1, w_2. To find the directions of such vectors, we look at the points whose intensity value is closest to the intensity $u(t,x,\theta)$ in the previous frame (term $|u(t,x-w_1,\theta-\Delta\theta) - u(t,x,\theta)|$) and in the next frame (term $|u(t,x+w_2,\theta+\Delta\theta) - u(t,x,\theta)|$). Those differences are scaled by the factor $1/(\Delta\theta)^2$. Note that if we find corresponding Lambertian points, then both terms vanish. The first term in the sum, namely $|\langle \nabla u(t,x,\theta), w_1 - w_2\rangle|/(\Delta\theta)$, corresponds to the so-called *apparent acceleration*, i.e. to the difference between the candidate Lambertian velocity vectors w_1 and w_2 in the direction of ∇u. For details and some more background from the optical flow point of view we refer to [1, 2, 19]. The quantity clt_u is thus related to the curvature of the space-time level curve passing through the space-time point (x,θ) in the scale t (curvature of the Lambertian trajectory). The value of clt_u vanishes for the Lambertian points that are in uniform motion. This is consistent with the purpose not to alter such trajectories. On the other hand, for the noisy points there is no motion coherence and thus clt_u will be large there.

Concerning the space coherence, we assume that distinguished structures are located in the regions with a certain mean value of the image intensity, and that the object boundary forms an edge in the image. In order to construct a spatial diffusion process we thus require specific behavior on the edges as in the Perona-Malik anisotropic diffusion equation.

To combine time coherence of moving objects with their spatial localization we consider the following equation for the processing of image sequences [56]:

$$u_t = clt_u \nabla.(g(|\nabla G_\sigma * u|)\nabla u). \tag{2.16}$$

As an application, we are dealing with a phantom-like image sequence consisting of expanding, slightly deforming and moving ellipses with the inner structure in the form of quatrefoils. We add impulsive (salt & pepper), Gaussian noise and blurring to the frames of

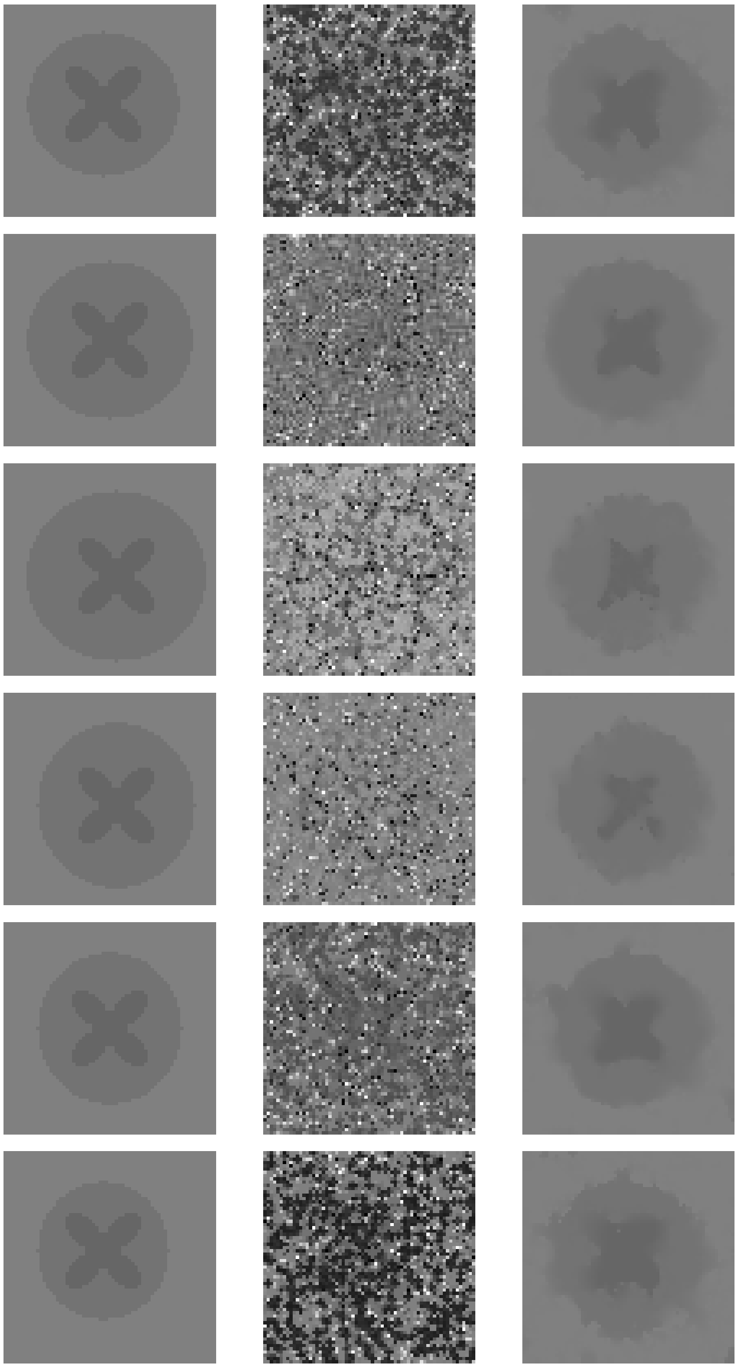

Figure 10: 2D phantom-original (left), noisy (middle) and processed (right) images.

the image sequence. The original six-frame sequence and its destroyed version are shown in the first two columns of Fig. 10. The reconstruction of any noisy frame of this sequence by a standard (still image) filtering algorithm is very difficult task (by our experience impossible). The right-hand column of Fig. 10 represents the results of (2.16) applied to the noisy sequence after 10 discrete scale steps of the numerical implementation from [56].

Next we have applied the method to 3D echocardiographic sequence. In Figs. 11–12 the iso-surfaces corresponding to the blood-muscle interface have been visualized. Figs. 11–12 consist of three sub-figures. For each row, on the left we show the echo-volume visualized using the original noisy data, in the middle the result after three discrete scale steps, and on the right after nine discrete scale steps of the model (2.16).

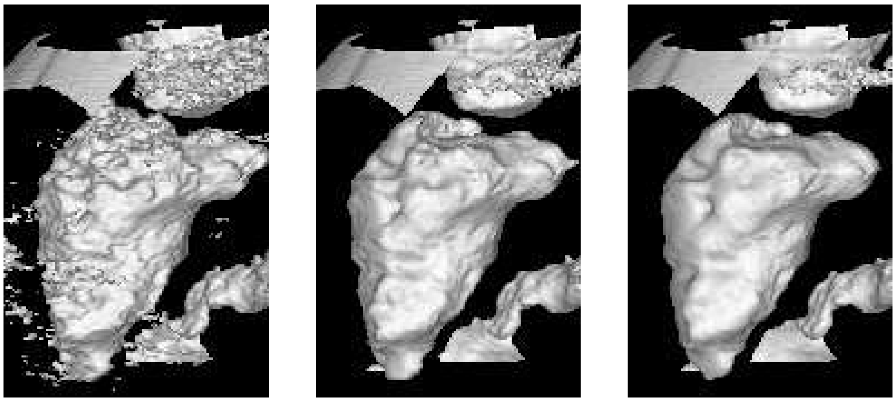

Figure 11: The multiscale analysis of the 1st frame of a 3D-echocardiographic image sequence by the equation (2.16) [56].

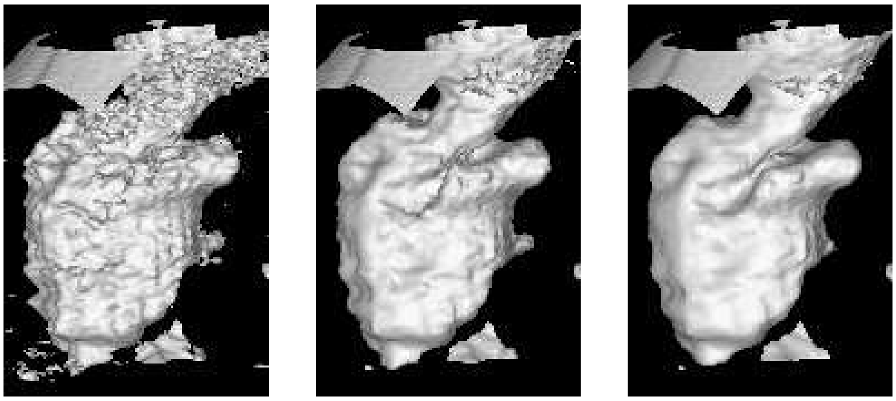

Figure 12: The multiscale analysis of the 7th frame of a 3D-echocardiographic image sequence by the equation (2.16) [56].

3 Variational computational methods

The aim of this chapter is to present numerical schemes for solving the nonlinear diffusion equations presented in the previous sections. Since images are given on a discrete grid (pixel/voxel structure), we discretize the PDEs to get a numerical scheme implemented on the computer. One can use a wide range of methods devoted to the numerical solution of PDEs. Semi-implicit schemes [9, 26, 5, 6, 62, 63, 41, 33, 34, 52, 42, 22, 23, 24, 13], where the nonlinear terms of the equation are treated from the previous discrete scale step, and linear terms are considered at the current scale level, have favorable stability and efficiency properties. For space discretization either finite element method [26, 5, 6, 52, 13], finite/complementary volume method [41, 33, 34, 23] or finite difference methods [9, 62, 63] can be used. In this paper we discuss discretization in space by variational techniques, i.e. using finite element and finite/complementary volume methods. They are based on integral (weak, variational) formulations of the initial-boundary value problems for PDEs. Variational methods have a strong physical background since they are based on principles like minimization of energy (finite element method) or conservation laws (finite and complemetary volume methods). They allow a clear and physically meaningful derivation of difference equations which are local and easy to implement. Convergence of the schemes to solutions of the PDEs can be proved.

For our presentation we have chosen two representative geometry-driven diffusion models: the regularized Perona-Malik anisotropic diffusion equation (2.1), and the nonlinear degenerate diffusion equation of mean curvature flow type (2.10) due to Alvarez, Lions and Morel. They are accompanied by the boundary and initial conditions (2.2)–(2.3), and for both equations we assume the hypotheses (2.4)–(2.6). Both models represent a similar view to image selective smoothing, in the sense that edge positions are preserved in the image multiscale analysis.

Remark In the next paragraphs, we will use the standard notations for the functional spaces $L_p(\Omega)$, $p \geq 1$, and $C^k(\overline{\Omega})$. By V we denote the Sobolev space $W^{1,2}(\Omega)$ of $L_2(\Omega)$-functions with square integrable weak derivates. The absolute value will be denoted by $|\cdot|$ and the norm by $\|\cdot\|$ with a subscript given by the corresponding functional space. By C we denote a general (large) constant.

3.1 Semi-discretizations in scale

We start by the semi-discretizations in scale of the problems given by (2.1) and (2.10), respectively. Choosing $N \in \mathbb{N}$ we obtain the length of the uniform discrete scale step $k = T/N$. We replace the scale derivative in (2.1), (2.10) by backward difference. The nonlinear terms of the equations are treated from the previous scale step while the linear terms are considered at the current scale level—this means semi-implicitness of the method. Let us start with approximation in scale of equation (2.1) (see [9, 26, 41]).

Semi-discrete linear scheme for solving equation (2.1) *Let $N \in \mathbb{N}$, $k = T/N$ and $\sigma > 0$ be fixed numbers, and let u^0 be given by* (2.3). *For every $n = 1, \ldots, N$, we look for a function u^n which is a solution of the equation*

$$\frac{u^n - u^{n-1}}{k} - \nabla.(g(|\nabla G_\sigma * u^{n-1}|)\nabla u^n) = 0. \tag{3.1}$$

It is not difficult to see that there exist unique variational solutions u^n of (3.1) at every discrete scale step for which the stability estimates

$$\|u^n\|_{L_2(\Omega)} \le \|u^0\|_{L_2(\Omega)}, \qquad \|u^n\|_{L_\infty(\Omega)} \le \|u^0\|_{L_\infty(\Omega)}, \qquad n = 1,\dots,N,$$
$$\sum_{n=1}^{N} \|\nabla u^n\|^2_{L_2(\Omega)} k \le C, \qquad \sum_{n=1}^{N} \|u^n - u^{n-1}\|^2_{L_2(\Omega)} \le C,$$

hold [26, 6]. Moreover, if we construct the so-called Rothe's step function $\bar{u}^{(N)}(t) = u^n$, $(n-1)k < t \le nk$, one can prove its convergence in $L_2(Q_T)$ to the unique weak solution of (2.1) ([26]; see also [25] for the comprehensive theory and applications of the so-called Rothe's method in the numerical analysis of PDEs).

In spite of equation (2.1), the equation (2.10) is not in divergence form. For second order partial derivatives, as is usual in variational methods, we would like to use integration by parts or the divergence theorem to get an integral formulation (see the next sections). Thus, we first move the term in front of the divergence to the time derivative, and then we write semi-implicit discretization of (2.10) in scale [23].

Semi-discrete linear scheme for solving equation (2.10) *Let $N \in \mathbb{N}$, $k = T/N$ and $\sigma > 0$ be fixed numbers, and let u^0 be given by (2.3). For every $n = 1,\dots,N$, we look for a function u^n which is a solution of the equation*

$$\frac{1}{g(|\nabla G_\sigma * u^{n-1}|)|\nabla u^{n-1}|} \frac{u^n - u^{n-1}}{k} - \nabla . \left(\frac{\nabla u^n}{|\nabla u^{n-1}|} \right) = 0. \tag{3.2}$$

Since in the general situation there can be zero in the denominators of (3.2), we will regularize this equation in the sense of Evans and Spruck [16]. We approximate $|\nabla u^{n-1}| \approx \sqrt{\varepsilon + |\nabla u^{n-1}|^2}$ (with ε a small real number) in the implementation, and study the behavior of this regularization for ε tending to zero [23]. Let us denote

$$g^{n-1} := g(|\nabla G_\sigma * u^{n-1}|). \tag{3.3}$$

To see stability of (3.2) (formally, without regularization at this point), multiply it by $u^n - u^{n-1}$, then integrate over Ω. Using

$$a(a-b) = \frac{1}{2}a^2 - \frac{1}{2}b^2 + \frac{1}{2}(a-b)^2, \tag{3.4}$$

we get

$$\int_\Omega \frac{(u^n - u^{n-1})^2}{kg^{n-1}|\nabla u^{n-1}|} dx + \frac{1}{2} \int_\Omega \frac{|\nabla u^n|^2 - |\nabla u^{n-1}|^2 + |\nabla u^n - \nabla u^{n-1}|^2}{|\nabla u^{n-1}|} dx = 0.$$

Since

$$|\nabla u^n - \nabla u^{n-1}|^2 = (|\nabla u^n| - |\nabla u^{n-1}|)^2 + \left(\frac{\nabla u^n}{|\nabla u^n|} - \frac{\nabla u^{n-1}}{|\nabla u^{n-1}|} \right)^2 |\nabla u^n| \, |\nabla u^{n-1}|,$$

we get

$$\int_\Omega \frac{(u^n - u^{n-1})^2}{kg^{n-1}|\nabla u^{n-1}|}\,dx + \frac{1}{2}\int_\Omega \frac{|\nabla u^n|^2 - |\nabla u^{n-1}|^2 - (|\nabla u^n| - |\nabla u^{n-1}|)^2}{|\nabla u^{n-1}|}\,dx$$
$$+ \int_\Omega \frac{(|\nabla u^n| - |\nabla u^{n-1}|)^2}{|\nabla u^{n-1}|}\,dx + \frac{1}{2}\int_\Omega \left(\frac{\nabla u^n}{|\nabla u^n|} - \frac{\nabla u^{n-1}}{|\nabla u^{n-1}|}\right)^2 |\nabla u^n|\,dx = 0.$$

Due to the positivity of the other terms we get for the second term

$$\int_\Omega \frac{|\nabla u^n|\,|\nabla u^{n-1}| - |\nabla u^{n-1}|^2}{|\nabla u^{n-1}|}\,dx \le 0,$$

which gives

$$\|\nabla u^n\|_{L_1(\Omega)} \le \|\nabla u^{n-1}\|_{L_1(\Omega)}, \tag{3.5}$$

and thus

$$\|\nabla u^n\|_{L_1(\Omega)} \le \|\nabla u^0\|_{L_1(\Omega)}, \quad 1 \le n \le N, \tag{3.6}$$

which is a $W^{1,1}$ stability estimate for the linear semi-implicit scheme (3.2), i.e. an estimate of the decay of total variation of the semi-discrete solutions. This estimate is a basic property of the flow by mean curvature and of a solution of the level set equation as well [60], and can also be interpreted as a curve shortening property [14, 40] of level sets.

3.2 Space discretizations

A discrete image is given on structure of pixels/voxels with rectangular shape in general (but it is not necessary for the methods presented in this paper). We will use this image structure to create a computational grid for the spatial discretization methods. Concerning the relation of the computational grid to the pixel structure, there will be a difference between the finite element and the complementary volume methods on the one hand, and the finite volume method on the other. The difference is related to the type of approximation of the solution of the partial differential equations assumed in those methods. While in the finite volume method the approximation of the solution is assumed to be piecewise constant [17], in the complementary volume and finite element methods it is assumed to be continuous piecewise linear. Thus, in the finite volume method, the computational grid will directly be taken to be the pixel structure of the image. The initially given and subsequently computed values of discrete intensity are considered as approximations of average of continuous intensity function on pixels. On the other hand, in the finite element and complementary volume methods, the initially given values of discrete intensity, and also the computed values, are considered as approximations of the continuous intensity function in the centers of pixels. The centers of pixels then correspond to the nodes of the finite element or the complementary volume triangulation. We can get such a triangulation by connecting the centers of pixels by a new rectangular mesh and then dividing every rectangle into two triangles (or six tetrahedra in 3D). It also means that in these two methods the computational domain Ω is given as the union of all triangles constructed in this way (Ω thus corresponds to the image domain minus the outer half of every boundary pixel)—see Fig. 13. Let us note that the splitting of every pixel into two triangles, as depicted in Fig. 13, is not the only possibility. The orientation of triangles can change locally, e.g. following an edge direction, or it can be given by a refinement procedure in a bisection algorithm [5]. We will assume that the constructed triangulation has no interior angle larger than $\pi/2$. Then, for the complementary volume method, we will construct a dual mesh. This dual mesh will again, in a sense, copy the pixel structure of the image. Let us note that in the finite element method we will use just the triangulation, in the complementary volume method we use both the triangulation and the dual mesh, and in the finite volume method we use just the pixel structure of the image corresponding to the dual mesh. Let us note also the possibility to use the center of pixels (and the corresponding rectangular mesh) to construct a bilinear finite element solution of the problem [52].

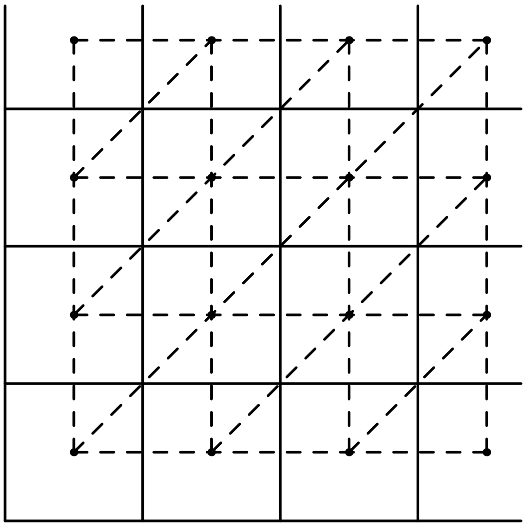

Figure 13: The image pixels (solid lines) corresponding to the finite volume mesh and to the dual mesh for the complementary volume method. Triangulation (dashed lines) for the finite element and complementary volume methods, with nodes (round points) corresponding to the centers of pixels.

We now define quantities which will be used in descriptions of fully discrete schemes (see also [60, 17, 10]). Let us assume that we are given a triangulation $\mathcal{T}$ (e.g. given by a previous construction). The dual mesh will consist of cells p (also called complementary or finite volumes) associated with the pth node of the triangulation $\mathcal{T}$, $p = 1, \ldots, M$. The dual mesh will be denoted by τ. The co-volume p is bounded by the lines (planes in 3D) that bisect and are perpendicular to the edges emanating from the node. By $m(p)$ we denote measure in $\mathbb{R}^d$ of p. We will denote the edge of $\mathcal{T}$ connecting the pth node to the qth by σ_{pq}, and its length by d_{pq}. We denote by E_{pq} the set of simplices having σ_{pq} as an edge, i.e., $E_{pq} = \{T \in \mathcal{T} | \sigma_{pq} \subset T\}$. Let e_{pq} denote the co-edge that is the perpendicular bisector of σ_{pq}, and let x_{pq} be a point of intersection of e_{pq} and σ_{pq}. By $m(e_{pq})$ we denote the measure of e_{pq} in $\mathbb{R}^{d-1}$. For each $T \in E_{pq}$, let c_{pq}^T be the length of the portion of e_{pq} that is in T, i.e., $c_{pq}^T = m(e_{pq} \cap T)$. Let $S(p)$ be the set of simplices that have the pth node as a vertex, and for each node of $\mathcal{T}$ let $N(p)$ denote the set of nodes q, $q \neq p$, which are connected to the pth node by an edge and for which $m(e_{pq}) \neq 0$. In the situation depicted in Fig. 13, $N(p)$ consists of four neighbouring nodes q in the direction of the coordinate axes, for every inner node p of the triangulation. We denote by $\mathcal{E}$ the set of pairs of adjacent finite volumes,

$$\mathcal{E} = \{(p,q) \in \tau^2,\ p \neq q,\ m(e_{pq}) \neq 0\}.$$

Given a triangulation $\mathcal{T}$, we define the set $V_h \subset V$ of piecewise linear finite elements, i.e.,

$$V_h = V_h(\mathcal{T}) := \{v \in C^0(\overline{\Omega})|\ v|_T \in \mathcal{P}_1 \text{ for all } T \in \mathcal{T}\}.$$

For any $v_h \in V_h$ we will use the notation $v_p := v_h(x_p)$, where x_p is the coordinate vector of the pth node of the triangulation. Let $u_h^0 = I_h(u^0) \in V_h(\mathcal{T})$ be the nodal interpolant of u^0. This will be the initial function for the finite element and complementary volume methods. In the finite volume method we denote by $\overline{u}_p$ the representative value for the cell (the bar indicates that we mean the average value in the cell and not a nodal value). The initial condition for the finite volume scheme is then assumed to be

$$\overline{u}_p^0 = \frac{1}{m(p)} \int_p u^0(x)\, dx, \quad p \in \tau. \tag{3.7}$$

Before going to spatial discretizations, let us make the following remark on the realization of the convolution included in the evaluation of the Perona-Malik function g in either (3.1) or (3.2). We use two strategies. The first is the following: Using the Gauss function G_σ as the smoothing kernel, one can replace the term $G_\sigma * u^{n-1}$ by solving the linear heat equation for time σ with the initial condition given by u^{n-1}. This linear equation can be solved numerically at the same grid by just one implicit step with length σ. Thus, as a realization of the convolution, we look for a function u^c which is a solution of the heat equation discretized in time by the backward Euler method with step σ:

$$\frac{u^c - u^{n-1}}{\sigma} = \Delta u^c, \tag{3.8}$$

where Δ denotes the Laplace operator. This strategy is very suitable for the finite element and complementary volume methods, since in this case, the numerical solution u^c of (3.8) is piecewise linear on triangles, its gradients are constant, and thus we can simply evaluate the Perona-Malik function on every triangle.

In the finite volume method we can use another approach. Since we will consider piecewise constant spatial approximations $\overline{u}_p^n$ in every discrete scale step n, we can replace the convolution (integral) by a sum over pixels. For the gradient of the convolution term at any point x we get

$$\nabla G_\sigma * u^{n-1}(x) = \sum_r \overline{u}_r^{n-1} \int_r \nabla G_\sigma(x-s)\, ds, \tag{3.9}$$

where we used the convolution derivative property

$$\frac{\partial}{\partial x_i}(G_\sigma * u^{n-1}(x)) = \frac{\partial G_\sigma}{\partial x_i} * u^{n-1}(x)$$

and

$$\frac{\partial G_\sigma}{\partial x_i} * u^{n-1}(x) = \int_{\mathbb{R}^d} \frac{\partial G_\sigma}{\partial x_i}(x-s) u^{n-1}(s)\, ds = \sum_r \overline{u}_r^{n-1} \int_r \frac{\partial G_\sigma}{\partial z_i}(x-s)\, ds.$$

The sum in (3.9) is evaluated over control volumes r which surround the point x. If we choose a compactly supported smoothing kernel with support in a ball $B_{\sigma^*}(0)$ with radius σ^*, e.g. the function

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

where $\sigma^* = \sigma$ and the constant Z is chosen so that G_σ has unit mass, then the sum is restricted only to the control volumes contained in $B_{\sigma^*}(x)$, the ball centered at x. If the point x is close to the boundary of the image domain, we use an extension of a discrete solution. The coefficients $\int_r \nabla G_\sigma(x-s)\,ds$ in (3.9) can be computed in advance using a computer algebra system, e.g. Mathematica [66]. The same situation arises when we use the Gauss function and consider as the ball $B_{\sigma^*}(0)$ a "numerical support" of the Gauss function (i.e., we consider a domain in which the values of the Gauss function are above some treshold). Then, again, just a finite sum is evaluated in (3.9).

3.3 Finite element method in image processing

To describe the ideas of finite element space discretization, let us consider the equation (2.1) [26, 5]. Let k, σ be given numbers. Before the discretization we use approach (3.8) for the realization of the convolution, and then we write (3.1) as a couple of integral identities (weak formulations):

$$\int_\Omega u^n v\,dx + k \int_\Omega g(|\nabla u^c|)\nabla u^n \nabla v\,dx = \int_\Omega u^{n-1} v\,dx, \tag{3.10}$$

$$\int_\Omega u^c v\,dx + \sigma \int_\Omega \nabla u^c \nabla v\,dx = \int_\Omega u^{n-1} v\,dx \tag{3.11}$$

that hold for all $v \in V$. Then, at each scale level n, we look for a continuous piecewise linear function $u_h^n \in V_h(\mathcal{T})$ satisfying

$$\int_\Omega u_h^n v_h\,dx + k \int_\Omega g(|\nabla u_h^c|)\nabla u_h^n \nabla v_h\,dx = \int_\Omega u_h^{n-1} v_h\,dx \tag{3.12}$$

for all $v_h \in V_h(\mathcal{T})$, with $u_h^c \in V_h(\mathcal{T})$ being the solution of

$$\int_\Omega u_h^c v_h\,dx + \sigma \int_\Omega \nabla u_h^c \nabla v_h\,dx = \int_\Omega u_h^{n-1} v_h\,dx, \quad \forall v_h \in V_h(\mathcal{T}). \tag{3.13}$$

Considering the standard Lagrangian base functions $\varphi_q \in V_h(\mathcal{T})$, $q = 1, \dots, M$, given by $\varphi_q(x_p) = \delta_{qp}$ (Kronecker delta) for all nodes of $\mathcal{T}$, the functions u_h^n, u_h^c are given by

$$u_h^n = \sum_{p=1}^{M} u_p^n \varphi_p, \qquad u_h^c = \sum_{p=1}^{M} u_p^c \varphi_p. \tag{3.14}$$

Then using (3.14)) in (3.12)–(3.13) and taking $v_h = \varphi_q$, $q = 1, \dots, M$, as test functions, we get two Ritz-Galerkin systems of linear equations for the nodal values u_p^n, u_p^c, $p = 1, \dots, M$, of the piecewise linear functions u_h^n, u_h^c, respectively:

$$\sum_{p=1}^{M}\left(\int_\Omega \varphi_p\varphi_q\,dx + k\int_\Omega g(|\nabla u_h^c|)\nabla\varphi_p\nabla\varphi_q\,dx\right)u_p^n = \int_\Omega u_h^{n-1}\varphi_q\,dx, \quad q=1,\dots,M, \tag{3.15}$$

$$\sum_{p=1}^{M}\left(\int_\Omega \varphi_p\varphi_q\,dx + \sigma\int_\Omega \nabla\varphi_p\nabla\varphi_q\,dx\right)u_p^c = \int_\Omega u_h^{n-1}\varphi_q\,dx, \quad q=1,\dots,M. \tag{3.16}$$

Thus, in every discrete scale step we need to solve two linear systems with the matrices $\mathbf{M}+k\mathbf{A}(g(|\nabla u_h^c|))$, $\mathbf{M}+\sigma\mathbf{A}(1)$, respectively, where $\mathbf{M}_{q,p} = \int_\Omega \varphi_p\varphi_q\,dx$ is the so-called mass matrix and $\mathbf{A}(w)_{q,p} = \int_\Omega w\nabla\varphi_p\nabla\varphi_q\,dx$ is the stiffness matrix, which are symmetric and positive definite. The discrete solutions can be found efficiently by the preconditioned conjugate gradient method. It is also customary to use the so-called lumped (diagonalized) mass matrix $\mathbf{M}_{q,p} = \sum_{k=1}^{M}\varphi_p(x_k)\varphi_q(x_k)\int_\Omega\varphi_k\,dx$ in the systems (3.15)–(3.16). In order to improve the efficiency of the finite element method, a choice of different (coarsened) triangulations $\mathcal{T}_n$ in subsequent scale steps $n=1,\dots,N$, is possible (see [5, 6, 52] and the next subsection).

The same ideas as above can be used for the finite element discretization of equation (2.10). The only difference will be in the facts that the mass matrix $\mathbf{M}(w)$ will depend on $w = 1/(g(|\nabla u_h^c|)|\nabla u_h^{n-1}|)$, the stiffness matrix $\mathbf{A}(w)$ will depend on $w = 1/|\nabla u_h^{n-1}|$, and that Evans-Spruck regularization is used in the denominators.

3.4 Adaptivity in the finite element method

We can improve efficiency of the finite element method by the use of adaptively chosen grids at each scale step. Usually, for time-dependent problems, a modification consisting of refinement and coarsening steps is necessary to adjust the grid at a given time step [4]. However, for the problem (2.1) it is sufficient to coarsen the initial grid successively. There is no spatial *movement* of edges, hence no refinements of the grids are needed. This access may reduce the computational effort considerably, since with the increasing scale the solution tends to be more flat in large sub-regions of the image. The coarsening of the computational grids rapidly reduces the number of unknowns in the linear systems to be solved at the discrete scale steps of the method.

Concerning the adaptive algorithm, first we generate a triangulation $\mathcal{T}_0$ corresponding to a pixel/voxel structure of the image by (globally) refining a coarse grid $\mathcal{T}^0$, the so-called *macro triangulation*. The refinement procedure generates a sequence $\mathcal{T}^0, \mathcal{T}^1, \mathcal{T}^2, \dots$ of finer and finer meshes until the desired structure is reached. Then $\mathcal{T}_0 := \mathcal{T}^{k_0}$, where k_0 is the last refinement step. For the refinement we have chosen the so called *bisection method*, which allows easily for coarsening [4].

Refinement of the grid by bisection [4] Before starting the refinement process, one edge of every triangle of the triangulation at kth refinement level is marked (Fig. 14). This edge is called the *refinement edge* (a good choice is the longest one). To divide a single triangle, it is cut through the midpoint of the refinement edge and the vertex opposite to the refinement edge. The new refinement edges are chosen opposite to the new vertex (Fig. 15).

Let us start with $\mathcal{T}^0$. Then for every bisection level k let Σ^+ be the set of those triangles which have to be divided ($\Sigma^+ = \mathcal{T}^k$ in case of uniform refinement). Then *one bisection step* (see also Fig. 16) at level k is given by:

while $\Sigma^+ \neq \emptyset$ **do**

- `bisect all` $T \in \Sigma^+$ `as described above, obtain an intermediate triangulation` $\hat{\mathcal{T}}^k$ `(possibly non-conforming)`
- `let now` Σ^+ `be the set of those tetrahedra with a non-conforming node.`

endwhile
$\mathcal{T}^{k+1} := \hat{\mathcal{T}}^k$

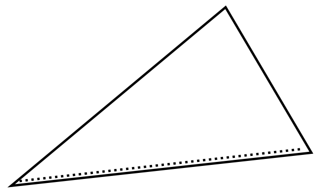

Figure 14: Triangle with refinement edge

Figure 15: Bisection of a single triangle

To provide a *local coarsening*, we make the following definitions:

(i) A simplex $T \in \mathcal{T}$ has level l if T was obtained after l refinement steps.

(ii) A simplex T is said to have *locally finest level* if the levels of all neighbours are less than or equal to the level of T.

(iii) Let $T \in \mathcal{T}$, and let T' be the father of T. A vertex P of T which was inserted while bisecting T' is called the *coarsening node* of T.

(iv) Let K be an edge of the triangulation $\mathcal{T}$ and K' the "father"-edge of K with midpoint Q. Set $M := \{T \in \mathcal{T} | T \cap K' \neq \emptyset\}$. If Q is the coarsening node for all $T \in M$, then M is called a *resolvable patch* (Fig. 17).

If M is a resolvable patch, then all $T \in M$ can be coarsened without interfering with $T' \in \mathcal{T}$ outside M. Therefore resolvable patches are the configurations which we allow to be coarsened. This guarantees that the coarsening process stays local.

Coarsening of the grid [4] Let $\mathcal{T}_n$ be a triangulation obtained by refinement and coarsening steps. Let $\Sigma^- \subset \mathcal{T}_n$ be the set of triangles to be derefined. Then *one coarsening step* consists of:

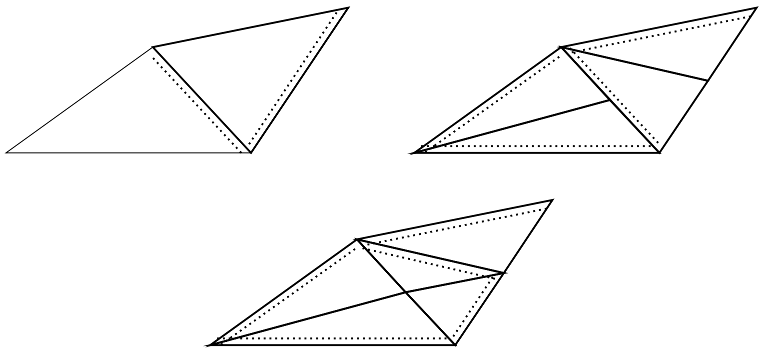

Figure 16: $\mathcal{T}^0$, $\hat{\mathcal{T}}^0$ and $\mathcal{T}^1$

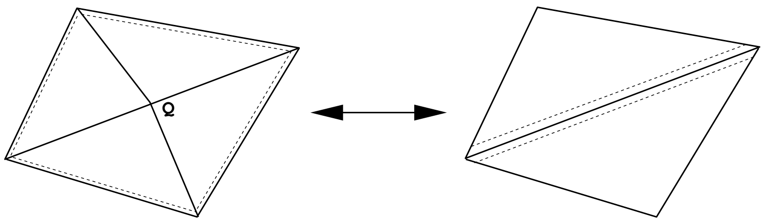

Figure 17: Resolvable patch M with coarsening node Q and coarsened patch

for each $T \in \Sigma^-$ **do**
    if T `belongs to a resolvable patch` M **then**
        if $T' \in \Sigma^-$ `for all` $T' \in M$ **then**
            `derefine` M
        endif
    endif
enddo

Coarsening criterion and our adaptive method [5] As the local behavior of $\nabla G_\sigma * u$ determines the evolution process and is an indicator for edges, the *coarsening criterion* is based on this value. Let $\varepsilon > 0$ be a given tolerance. For the nth scale step and u_h^n, the corresponding numerical solution on the grid $\mathcal{T}_n$, we allow all triangles $T \in \mathcal{T}_n$ to be coarsened, if

$$h_T |\nabla u_h^n| \leq \varepsilon \quad \text{on} \quad T,$$

where h_T is the diameter of triangle T. Thus we have the following *adaptive scheme*:

```
Let T_0, u_h^0 be given.
for n = 1,2,...N do
      define Σ^- := {T ∈ T_{n-1}; h_T|∇u_h^{n-1}| ≤ ε on T}
      derefine T_{n-1} according to Σ^- to obtain T_n
      set up the matrix M + σA(1)
      compute u_h^c ∈ V_h(T_n)
      set up the matrix M + kA(g(|∇u_h^c|))
      compute u_h^n ∈ V_h(T_n)
enddo
```

The same adaptive algorithm can be used in the 3D case, just changing triangles to tetrahedra. For 3D implementation and computational results we refer to [6]. There exist finite element software packages (see e.g. [57]) based on the bisection refinement and coarsening method, thus the adaptivity described above can be implemented also in such an environement.

3.5 Complementary volume spatial discretization

In this subsection we will discretize equation (2.10) or, more precisely, the semi-discrete approximation (3.2) by means of the complementary volume method [23]. In order to derive the complementary volume spatial discretization [60], we integrate (3.2) over a co-volume p

$$\int_p \frac{u^n - u^{n-1}}{g^{n-1}|\nabla u^{n-1}|k}\, dx = \int_p \nabla.\left(\frac{\nabla u^n}{|\nabla u^{n-1}|}\right) dx. \tag{3.17}$$

Using the divergence theorem on the right-hand side we get

$$\int_p \nabla\left(\frac{\nabla u^n}{|\nabla u^{n-1}|}\right) dx = \int_{\partial p} \frac{1}{|\nabla u^{n-1}|}\frac{\partial u^n}{\partial \nu}\, ds = \sum_{q\in N(p)} \int_{e_{pq}} \frac{1}{|\nabla u^{n-1}|}\frac{\partial u^n}{\partial \nu}\, ds. \tag{3.18}$$

If $u_h^n \in V_h(\mathcal{T})$ is a continuous piecewise linear function on a triangulation $\mathcal{T}$ and if its nodal values are denoted by $u_p^n = u_h^n(x_p)$, then

$$\sum_{q\in N(p)} \int_{e_{pq}} \frac{1}{|\nabla u_h^{n-1}|}\frac{\partial u_h^n}{\partial \nu}\, ds = \sum_{q\in N(p)} \left(\sum_{T\in E_{pq}} \frac{c_{pq}^T}{|\nabla u_T^{n-1}|}\right) \frac{u_q^n - u_p^n}{d_{pq}}, \tag{3.19}$$

where $|\nabla u_T^{n-1}|$ denotes the constant value of the gradient of u_h^{n-1} in the simplex T. The complementary volume method approximates the left-hand side of (3.17) by

$$\frac{m(p)(u_p^n - u_p^{n-1})}{kg(|\nabla u_p^c|)|\nabla u_p^{n-1}|}, \tag{3.20}$$

where $|\nabla u_p^{n-1}|$, $|\nabla u_p^c|$ denote an approximation of the gradient in co-volume p. For that purpose we have chosen the average value of gradients in the co-volume [60]

$$|\nabla u_p| \approx \sum_{T\in S(p)} \frac{m(T\cap p)}{m(p)}|\nabla u_T|. \tag{3.21}$$

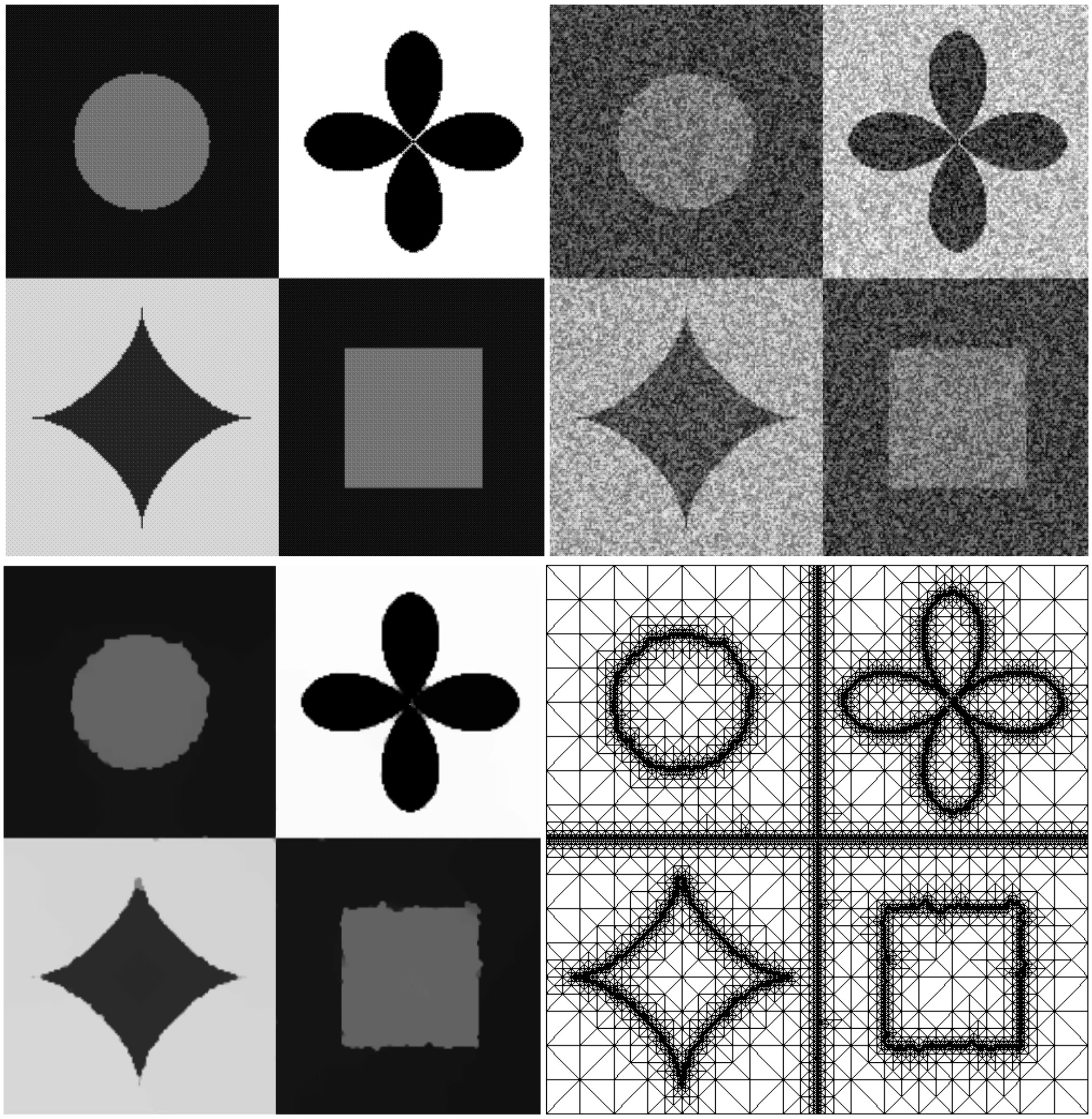

Figure 18: Processing of a noisy image by the adaptive finite element method for the anisotropic diffusion equation (2.1). Upper part: Original image and its noise-corrupted version; lower part: smoothed image after 8 discrete scale steps and corresponding 2D triangulation which is fine only along edges [5].

If we denote

$$b_p^{n-1} = \frac{m(p)}{g(|\nabla u_p^c|)|\nabla u_p^{n-1}|}, \tag{3.22}$$

$$a_{pq}^{n-1} = \frac{1}{d_{pq}} \sum_{T \in E_{pq}} \frac{c_{pq}^T}{|\nabla u_T^{n-1}|}, \tag{3.23}$$

we can write the

Linear fully discrete complementary volume scheme for solving equation (2.10) *For $n = 1, \dots, N$ we look for u_p^n, $p = 1, \dots, M$, satisfying the equation*

$$b_p^{n-1} u_p^n + k \sum_{q \in N(p)} a_{pq}^{n-1}(u_p^n - u_q^n) = b_p^{n-1} u_p^{n-1}. \tag{3.24}$$

Of course, before solving (3.24), we have to put $|\nabla u_p^c|$ into (3.22). The function $u_h^c \in V_h(\mathcal{T})$ is found by the same idea as given in (3.17)–(3.20) applied to (3.8); i.e., we solve (3.24) with u_p^n replaced by u_p^c, and with $b_p^{n-1} \equiv b_p = m(p)$, $a_{pq}^{n-1} \equiv a_{pq} = m(e_{pq})/d_{pq}$. Then $|\nabla u_p^c|$ is computed by (3.21) and put into the Perona-Malik function g. The system (3.24) gives a symmetric positive definite M-matrix with diagonal dominance, so standard or preconditioned linear solvers can be used.

Using the nonnegativeness of b_p^{n-1}, a_{pq}^{n-1}, we can get a uniform L_∞-stability estimate for the fully discrete scheme (3.24) in the form

$$\min_{p \in \mathcal{T}} u_p^0 \le \min_{p \in \mathcal{T}} u_p^n \le \max_{p \in \mathcal{T}} u_p^n \le \max_{p \in \mathcal{T}} u_p^0, \quad 1 \le n \le N, \tag{3.25}$$

which means

$$\|u_h^n\|_{L_\infty(\Omega)} \le \|u_h^0\|_{L_\infty(\Omega)}, \quad 1 \le n \le N. \tag{3.26}$$

To see (3.25), let us rewrite (3.24) in the form

$$u_p^n + \frac{k}{b_p^{n-1}} \sum_{q \in N(p)} a_{pq}^{n-1}(u_p^n - u_q^n) = u_p^{n-1}, \tag{3.27}$$

and let $\max u_h^n = \max_{r \in \mathcal{T}} u_r^n$ be achieved in the node p. Then the second term is nonnegative and thus $u_p^n \le u_p^{n-1} \le \max_{r \in \mathcal{T}} u_r^{n-1}$, which gives the result for max. The relation for min is derived in a similar way.

Already in the description of semidiscrete schemes we mentioned the use of the Evans-Spruck regularization [16], $|\nabla u|_\varepsilon = \sqrt{\varepsilon + |\nabla u|^2}$, instead of $|\nabla u|$ everywhere in the schemes to prevent possible zero gradients in the denominators. One can observe that the Evans-Spruck regularization is well suited for image smoothing purposes. For zero gradients such regularized discretization (3.24) reduces to a discretization of the linear heat equation. This is quite desirable at points with small change in intensity. On the other hand, the influence of ε can be neglected, and thus we have a curvature-driven flow of level lines, for large gradients. All results which were (formally) derived so far for fully discrete or semi-discrete schemes, namely the L_∞ and $W^{1,1}$ estimates (3.25) and (3.6), are valid also for the ε-regularization.

Moreover, they do not depend on the regularization parameter ε. Thus, we can pass to the limit and understand the solution of (3.24) in a generalized sense (see [23] for details). Concerning the computational point of view, if ε is very small the diagonal dominance of the system can be very weak. In [23] efficient linear solvers also for such a situation are studied. Fig. 19 gives an example of an application of the complementary volume scheme to a mammogram image (171×192 pixels).

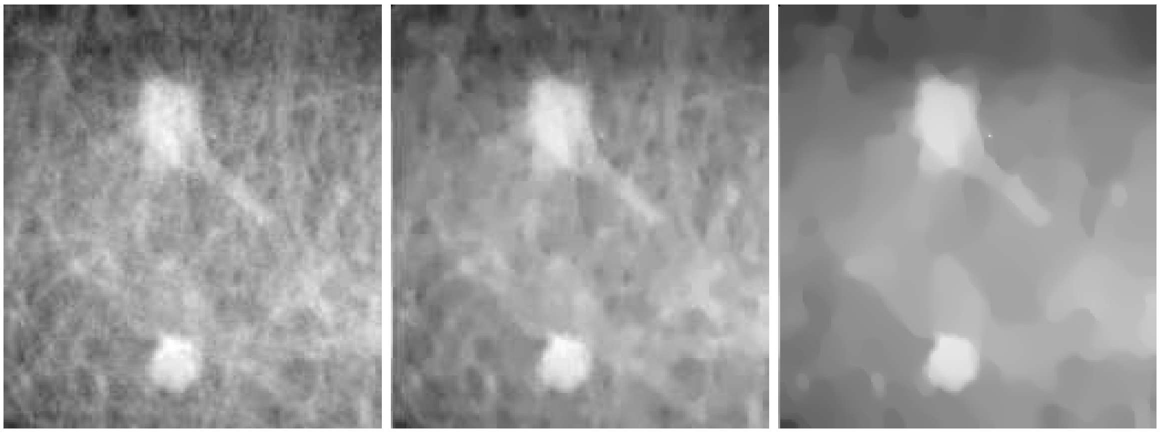

Figure 19: Initial image (left), result of multiscale analysis by equation (2.10) after 30 (middle) and 100 (right) discrete scale steps [23].

Using the integration (3.17) and the relation (3.19), one can also derive a complementary volume discretization of equation (2.1) or, more precisely, for the semi-discrete coupling (3.1), (3.8). In such a way, we get the system (3.24) with

$$b_p^{n-1} = m(p), \qquad a_{pq}^{n-1} = \frac{1}{d_{pq}} \sum_{T \in E_{pq}} c_{pq}^T g(|\nabla u_T^c|), \tag{3.28}$$

where u_h^c is computed in the same way as above. Let us note that such a discretization of (2.1) can be considered as a special mass lumping approximation in the finite element method.

3.6 Finite volume discretization

In this subsection we will consider the finite volume discretization of equation (2.1) only. In our approach, the finite volume grid corresponds to the dual mesh τ, and finite volumes correspond to co-volumes as they are identical with the pixel/voxel structure of the image. In general, the finite volume can be either a simplex of the triangulation or a co-volume of the dual mesh [17]; we use the second strategy. The main difference as compared to the complementary volume technique, is that the approximating functions in the finite volume method are not in $V_h(\mathcal{T})$, but they are just piecewise constant on finite volumes. Thus, we cannot work directly with gradients or normal derivatives since they are either zero (inside finite volumes) or infinite (on their boundaries). The derivation of the finite volume scheme follows the ideas of the last paragraph in the previous subsection. We integrate the equation (2.1) in every finite volume p. Then, by means of $\overline{u}_p$ representing an approximate value of the solution inside the finite volume $p, p \in \tau$, we approximate fluxes through the boundary of p (compare (3.19)). The value of diffusion coefficient along e_{pq} is approximated by its value at the point x_{pq}. We denote by $T_{pq} = m(e_{pq})/d_{pq}$ the transmissivity coefficient, and get the

Linear fully discrete finite volume scheme for solving equation (2.1) [41] *Let $N \in \mathbb{N}$, $k = T/N$ and $\sigma > 0$ be fixed numbers, $t_n = nk$, $n = 0, \ldots, N$. For $n = 1, \ldots, N$ we look for $\overline{u}_p^n, p \in \tau$, satisfying*

$$m(p)\overline{u}_p^n + k \sum_{q \in N(p)} g_{pq}^{\sigma,n-1} T_{pq} \left(\overline{u}_p^n - \overline{u}_q^n\right) = m(p)\overline{u}_p^{n-1}, \tag{3.29}$$

starting with

$$\overline{u}_p^0 = \frac{1}{m(p)} \int_p u^0\,(x)\; dx, \quad p \in \tau, \tag{3.30}$$

where

$$g_{pq}^{\sigma,n-1} = g(|\nabla G_\sigma * \tilde{u}_{h,k}(x_{pq}, t_{n-1})|) \tag{3.31}$$

and $\tilde{u}_{h,k}$ is an extension of the piecewise constant function $\overline{u}_{h,k}$ ($h = \max_{p\in\tau} \operatorname{diam}(p)$) defined as follows

$$\overline{u}_{h,k}(x,t) = \sum_{n=0}^{N} \sum_{p \in \tau} \overline{u}_p^n \chi_{\{x \in p\}} \chi_{\{t_{n-1} < t \leq t_n\}} \tag{3.32}$$

with the boolean function $\chi_{\{A\}} = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{elsewhere.} \end{cases}$

The function $\overline{u}_{h,k}$, constructed using discrete values given by the scheme (3.29), is considered as the approximation of the solution and its convergence to a unique weak solution of (2.1) as $h, k \to 0$ can be proved ([41], see also the next subsection for the main ideas). It is clear that (3.29) gives a linear system with a symmetric, strictly diagonally dominant M-matrix, so there exists a unique discrete solution at every discrete scale step. Moreover, using the same trick as in (3.27), we get L_∞-stability of the scheme

$$\min_{p\in\tau} \overline{u}_p^0 \leq \min_{p\in\tau} \overline{u}_p^n \leq \max_{p\in\tau} \overline{u}_p^n \leq \max_{p\in\tau} \overline{u}_p^0, \quad 1 \leq n \leq N. \tag{3.33}$$

3.7 Convergence of the finite volume scheme

3.1 Definition A *weak solution* of the regularized Perona-Malik problem (2.1)–(2.3) is a function $u \in L_2(I, V)$ satisfying the identity

$$\int_0^T \int_\Omega u \frac{\partial \varphi}{\partial t}(x,t)\, dx\, dt + \int_\Omega u_0(x)\varphi(x,0)\, dx - \int_0^T \int_\Omega g(|\nabla G_\sigma * u|)\nabla u \nabla \varphi\, dx\, dt = 0 \tag{3.34}$$

for all $\varphi \in \Psi$, where Ψ is the space of smooth test functions

$$\Psi = \{\varphi \in C^{2,1}(\overline{\Omega} \times [0,T]), \nabla\varphi.\overrightarrow{n} = 0 \text{ on } \partial\Omega \times (0,T), \varphi(\cdot,T) = 0\}. \tag{3.35}$$

In [9] Catté, Lions, Morel and Coll proved that there exists a unique weak solution of (2.1)–(2.3) which is also the classical solution of the problem at the same time. To get the existence they used Schauder's fixed point theorem with iterations in entire parabolic equations. In [41] we look for that solution in a computationally natural and efficient way using a semi-implicit finite volume scheme. In this subsection we give the main ideas of the proof of the convergence of the finite volume solution to the weak solution of (2.1)–(2.3). The strategy is to prove some *a priori* estimates of the numerical solution, using the scheme (3.29). These estimates will lead to the so-called *space* and *time translate estimates* for the approximate solution $\overline{u}_{h,k}$. The space and time translate estimates are specific formulations of the equicontinuity in the following well-known criterion (see e.g. [35]):

Kolmogorov's relative compactness criterion in $L_2(Q_T)$ *The set $K \subset L_2(Q_T)$ is relatively compact if and only if*

(i) *K is bounded, i.e., there exists $C > 0$ such that $\|f\| \le C$ for every $f \in K$;*

(ii) *K is mean equicontinous, i.e., for every $\varepsilon > 0$ there exists $\delta > 0$ such that*

$$\int_{Q_T} (f(x+\gamma) - f(x))^2 \, dx < \varepsilon^2$$

for each $f \in K$ and γ with $|\gamma| < \delta$.

Proving relative compactness one gets that there exists a limit function $u \in L_2(Q_T)$ of the sequence $\overline{u}_{h,k}$ as $h, k \to 0$. Moreover, the space translates give that this limit is in $L_2(I, V)$, so it is a good candidate for the weak solution. It will be the last step of the convergence proof that u fulfills the weak identity (3.34).

3.2 Lemma (*a priori* estimates) *There exists a positive constant C, independent of h and k, such that*

(i) $\max_{0 \le n \le N} \sum_{p \in \tau} (\overline{u}_p^n)^2 m(p) \le C$;

(ii) $\sum_{n=1}^N k \sum_{(p,q) \in \mathcal{E}} T_{pq} (\overline{u}_p^n - \overline{u}_q^n)^2 \le C$;

(iii) $\sum_{n=1}^N \sum_{p \in \tau} (\overline{u}_p^n - \overline{u}_p^{n-1})^2 m(p) \le C$.

Proof Let us multiply the scheme (3.29) by $\overline{u}_p^n$ to obtain

$$(\overline{u}_p^n - \overline{u}_p^{n-1}) \overline{u}_p^n m(p) = k \sum_{q \in N(p)} g_{pq}^{\sigma,n-1} T_{pq} (\overline{u}_q^n - \overline{u}_p^n) \overline{u}_p^n. \tag{3.36}$$

Using (3.4) on the left-hand side of (3.36) and summing over $p \in \tau$, we have that

$$\begin{aligned} \frac{1}{2} \sum_{p \in \tau} (\overline{u}_p^n)^2 m(p) - \frac{1}{2} \sum_{p \in \tau} (\overline{u}_p^{n-1})^2 m(p) + \frac{1}{2} \sum_{p \in \tau} (\overline{u}_p^n - \overline{u}_p^{n-1})^2 m(p) \\ = k \sum_{p \in \tau} \sum_{q \in N(p)} g_{pq}^{\sigma,n-1} T_{pq} (\overline{u}_q^n - \overline{u}_p^n) \overline{u}_p^n. \end{aligned} \tag{3.37}$$

The following trick is often used in the finite volume technique for anti-symmetric $a_{pq} = -a_{qp}$:

$$\sum_{p\in\tau}\sum_{q\in N(p)} a_{pq}b_p = \sum_{(p,q)\in\mathcal{E}} a_{pq}b_p = \sum_{(q,p)\in\mathcal{E}} a_{qp}b_q = -\sum_{(p,q)\in\mathcal{E}} a_{pq}b_q,$$

hence

$$2\sum_{p\in\tau}\sum_{q\in N(p)} a_{pq}b_p = -\sum_{(p,q)\in\mathcal{E}} a_{pq}(b_q - b_p) \tag{3.38}$$

whence

$$\sum_{p\in\tau}\sum_{q\in N(p)} g_{pq}^{\sigma,n-1}T_{pq}(\overline{u}_q^n - \overline{u}_p^n)\overline{u}_p^n = -\frac{1}{2}\sum_{(p,q)\in\mathcal{E}} g_{pq}^{\sigma,n-1}T_{pq}(\overline{u}_p^n - \overline{u}_q^n)^2. \tag{3.39}$$

Applying (3.39) in (3.37) and summing over $n = 1,\dots,m \le N$, we have

$$\begin{aligned}\frac{1}{2}\sum_{p\in\tau}(\overline{u}_p^m)^2 m(p) + \frac{1}{2}\sum_{n=1}^{m}\sum_{p\in\tau}(\overline{u}_p^n - \overline{u}_p^{n-1})^2 m(p) + \frac{1}{2}\sum_{n=1}^{m} k \sum_{(p,q)\in\mathcal{E}} g_{pq}^{\sigma,n-1}T_{pq}(\overline{u}_p^n - \overline{u}_q^n)^2 \\ = \frac{1}{2}\sum_{p\in\tau}(\overline{u}_p^0)^2 m(p).\end{aligned}$$

Since $u^0 \in L_\infty(\Omega)$ we get the results (i)and (iii). We also have

$$\left|\frac{\partial}{\partial x_i}G_\sigma * \tilde{u}_{h,k}(x_{pq}, t_{n-1})\right| \le \int_{\mathbb{R}^d}\left|\frac{\partial}{\partial x_i}G_\sigma(x_{pq} - \xi)\tilde{u}(\xi, t_{n-1})\right| d\xi \le C_\sigma \|\tilde{u}(.,t_{n-1})\|_{L_\infty(\Omega)}.$$

Because of (3.33) we have that $|\nabla G_\sigma * \tilde{u}_{h,k}(x_{pq}, t_{n-1})| < \infty$, which in turn implies that there exists a positive constant α such that $g_{pq}^{\sigma,n-1} > \alpha > 0$, from which one can deduce the assertion (ii) of the lemma. □

In order to apply Kolmogorov's compactness criterion we have to estimate the integral

$$\begin{aligned}&\int_{Q_T}(\overline{u}_{h,k}(x+\xi, t+s) - \overline{u}_{h,k}(x,t))^2\,dx\,dt \\ &\le 2\int_{Q_T}(\overline{u}_{h,k}(x+\xi, t+s) - \overline{u}_{h,k}(x,t+s))^2\,dx\,dt + 2\int_{Q_T}(\overline{u}_{h,k}(x, t+s) - \overline{u}_{h,k}(x,t))^2\,dx\,dt.\end{aligned}$$

Using an extension of the solution it is sufficient to restrict the two integrals on the right-hand side to domains $\Omega_\xi \times (0,T)$ with $\Omega_\xi = \{x \in \Omega, [x, x+\xi] \in \Omega\}$ and $\Omega \times (0, T-s)$ with $s \in (0,T)$, respectively [17, 41].

3.3 Lemma (Space translate estimates) *For any vector $\xi \in \mathbb{R}^d$ there exists a positive constant C such that*

$$\int_{\Omega_\xi\times(0,T)}(\overline{u}_{h,k}(x+\xi, t) - \overline{u}_{h,k}(x,t))^2\,dx\,dt \le C|\xi|(|\xi| + 2h), \tag{3.40}$$

where $\Omega_\xi = \{x \in \Omega, [x, x+\xi] \in \Omega\}$.

Proof Let $\xi \in \mathbb{R}^d$ be a given vector. For all $(p,q) \in \mathcal{E}$, let us denote $\xi_{pq} = \xi/|\xi|.n_{pq}$. For all $x \in \Omega_\xi$, we denote by $E(x,p,q)$ the function defined as follows:

$$E(x,p,q) = \begin{cases} 1 & \text{if the segment } [x, x+\xi] \text{ intersects } e_{pq},\ p \text{ and } q, \text{ and } \xi_{pq} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

For any $t \in (0,T)$ there exists $n \in \mathbb{N}$ such that $(n-1)k < t \le nk$. Then for almost all $x \in \Omega_\xi$ we can see that

$$\overline{u}_{h,k}(x+\xi,t) - \overline{u}_{h,k}(x,t) = \overline{u}^n_{p(x+\xi)} - \overline{u}^n_{p(x)} = \sum_{(p,q)\in\mathcal{E}} E(x,p,q)(\overline{u}^n_q - \overline{u}^n_p),$$

where $p(x)$ is the volume of $p \in \tau$ and $x \in p$. By the Cauchy-Schwarz inequality we obtain

$$\begin{aligned}(\overline{u}_{h,k}(x+\xi,t) - \overline{u}_{h,k}(x,t))^2 \\ \le \left(\sum_{(p,q)\in\mathcal{E}} E(x,p,q)\xi_{pq}d_{pq}\right)\left(\sum_{(p,q)\in\mathcal{E}} E(x,p,q)\frac{(\overline{u}^n_q - \overline{u}^n_p)^2}{\xi_{pq}d_{pq}}\right)\end{aligned} \tag{3.41}$$

and using the fact that $\xi_{pq}d_{pq} = \xi/|\xi|.n_{pq}d_{pq} = \xi/|\xi|.(x_q - x_p)$ we have

$$\sum_{(p,q)\in\mathcal{E}} E(x,p,q)\xi_{pq}d_{pq} = \frac{\xi}{|\xi|}.(x_{p(x+\xi)} - x_{p(x)}) \le |x_{p(x+\xi)} - x_{p(x)}| \le 2h + |\xi|.$$

Now, we integrate the relation (3.41) on $\Omega_\xi \times (0,T)$:

$$\begin{aligned}\int_{\Omega_\xi\times(0,T)} (\overline{u}_{h,k}(x+\xi,t) - \overline{u}_{h,k}(x,t))^2\,dx\,dt \\ \le (2h+|\xi|)\sum_{n=1}^{N} k \sum_{(p,q)\in\mathcal{E}} \frac{(\overline{u}^n_q - \overline{u}^n_p)^2}{\xi_{pq}d_{pq}} \int_{\Omega_\xi} E(x,p,q)\,dx.\end{aligned} \tag{3.42}$$

Taking into account the area of a parallelogram we have

$$\int_{\Omega_\xi} E(x,p,q)\,dx \le m(e_{pq})\xi.n_{pq} = m(e_{pq})\frac{\xi}{|\xi|}.n_{pq}|\xi| = m(e_{pq})|\xi|\xi_{pq},$$

and applying this result in (3.42) we obtain

$$\int_{\Omega_\xi\times(0,T)} (\overline{u}_{h,k}(x+\xi,t) - \overline{u}_{h,k}(x,t))^2\,dx\,dt \le (2h+|\xi|)|\xi| \sum_{n=1}^{N} k \sum_{(p,q)\in\mathcal{E}} T_{pq}(\overline{u}^n_q - \overline{u}^n_p)^2. \tag{3.43}$$

Finally, using *a priori* estimate (ii) of Lemma 3.2 we complete the proof. □

In a more technical way, but basically using again the *a priori* estimate (ii) of Lemma 3.2 we get:

3.4 Lemma (Time translate estimate) *There exists a positive constant C such that*

$$\int_{\Omega\times(0,T-s)} (\overline{u}_{h,k}(x,t+s) - \overline{u}_{h,k}(x,t))^2\,dx\,dt \le Cs$$

for all $s \in (0,T)$.

3.5 Lemma (Convergence of $\overline{u}_{h,k}$) *There exists $u \in L_2(Q_T)$ such that for some subsequence of $\overline{u}_{h,k}$*

$$\overline{u}_{h,k} \to u \quad \text{in} \quad L_2(Q_T)$$

as $h, k \to 0$. Moreover, this limit function is in $L_2(I, V)$.

Proof From the estimate (i) of Lemma 3.2 we have that $\|\overline{u}_{h,k}\|_{L_2(Q_T)} \leq C$, so the space and time translate estimates allow us to use Kolmogorov's relative compactness criterion in $L_2(Q_T)$ and we have the first assertion of the lemma.

Let $\varphi \in C_0^\infty(Q_T)$, $\varepsilon > 0$ and $\varphi(x,t) = 0$ if $|x - \partial\Omega| < \varepsilon$. Let $0 < |\xi| < \varepsilon$. Then by the Cauchy-Schwarz inequality

$$\int_{\Omega\times(0,T)} \frac{\overline{u}_{h,k}(x+\xi,t) - \overline{u}_{h,k}(x,t)}{|\xi|}\varphi(x,t)\,dx\,dt \leq \frac{\sqrt{C|\xi|(|\xi|+h)}}{|\xi|}\|\varphi\|_{L_2(Q_T)}.$$

For the limit function u we have

$$\int_{\Omega\times(0,T)} \frac{u(x+\xi,t) - u(x,t)}{|\xi|}\varphi(x,t)\,dx\,dt \leq \sqrt{C}\|\varphi\|_{L_2(Q_T)}.$$

On the other hand, by a changing of the variables $y = x + \xi$ we get

$$\begin{aligned}
&\int_{\Omega\times(0,T)} \frac{u(x+\xi,t) - u(x,t)}{|\xi|}\varphi(x,t)\,dx\,dt \\
&\quad = \int_{\Omega\times(0,T)} \frac{u(y,t)}{|\xi|}\varphi(y-\xi,t)\,dy\,dt - \int_{\Omega\times(0,T)} \frac{u(y,t)}{|\xi|}\varphi(y,t)\,dy\,dt \\
&\quad = -\int_{\Omega\times(0,T)} \frac{\varphi(y,t) - \varphi(y-\xi,t)}{|\xi|}u(y,t)\,dy\,dt \\
&\quad \leq C\|\varphi\|_{L_2(Q_T)}.
\end{aligned}$$

Let $\xi = \omega e_i$, where e_i is i-th coordinate vector, and let $\omega \to 0$. Then

$$-\int_{\Omega\times(0,T)} \frac{\partial\varphi(x,t)}{\partial x_i}u(x,t)\,dx\,dt \leq C\|\varphi\|_{L_2(Q_T)}, \quad \forall\varphi \in C_0^\infty(\Omega).$$

Thus u has generalized spatial derivatives in $L_2(Q_T)$, so it is in $L_2(I, V)$. □

The last step is to prove that u given in Lemma 3.5 fulfills the weak identity (3.34) from Definition 3.1, and thus is a weak solution of the regularized Perona-Malik problem. Since such a solution is unique [9], not only a subsequence of $\overline{u}_{h,k}$ but the whole sequence will converge to u. Let $\varphi \in \Psi$ be given. We obtain a discrete analogy of the weak solution identity multiplying the finite volume scheme

$$(\overline{u}_p^n - \overline{u}_p^{n-1})m(p) = k \sum_{q\in N(p)} g_{pq}^{\sigma,n-1}T_{pq}(\overline{u}_q^n - \overline{u}_p^n)$$

by $\varphi(x_p, t_{n-1})$ and summing the resulting identity over all $p \in \tau$ and $n = 1, \ldots, N$:

$$\sum_{n=1}^{N} k \sum_{p\in\tau} \frac{(\overline{u}_p^n - \overline{u}_p^{n-1})}{k}\varphi(x_p, t_{n-1})m(p) = \sum_{n=1}^{N} k \sum_{p\in\tau}\sum_{q\in N(p)} g_{pq}^{\sigma,n-1}T_{pq}(\overline{u}_q^n - \overline{u}_p^n)\varphi(x_p, t_{n-1}).$$

Next we perform a discrete integration by parts

$$\sum_{n=1}^{N}(a^n - a^{n-1})b^{n-1} = a^N b^N - a^0 b^0 - \sum_{n=1}^{N} a^n (b^n - b^{n-1})$$

on the left-hand side, and a rearrangement of the sum

$$\sum_{p\in\tau}\sum_{q\in N(p)} a_{pq}b_p = -\frac{1}{2}\sum_{(p,q)\in\mathcal{E}} a_{pq}(b_q - b_p)$$

on the right-hand side, which together with $\varphi(x_p, t_N) = 0$ gives

$$\sum_{n=1}^{N} k \sum_{p\in\tau_h} \overline{u}_p^n \frac{\varphi(x_p,t_n) - \varphi(x_p,t_{n-1})}{k} m(p) + \sum_{p\in\tau_h} \overline{u}_p^0 \varphi(x_p,0) m(p)$$
$$- \frac{1}{2}\sum_{n=1}^{N} k \sum_{(p,q)\in\mathcal{E}} g_{pq}^{\sigma,n-1} T_{pq}(\overline{u}_q^n - \overline{u}_p^n)(\varphi(x_q,t_{n-1}) - \varphi(x_p,t_{n-1})) = 0.$$

The correspondence of the discrete terms of the previous equation and the continuous integral terms of the weak identity (3.34) can be clearly seen, and the convergence as $h, k \to 0$ can be proved (see [41] for details). Thus we get:

3.6 Theorem *The sequence $\overline{u}_{h,k}$ converges strongly in $L_2(Q_T)$ to the unique weak solution u of* (2.1)–(2.3) *as $h, k \to 0$.*

3.8 Adaptivity in the finite volume method

Similarly as in the finite element method we can accelerate the numerical solution of the finite volume scheme using adaptive (coarsened) grids. For the adaptive grid construction we choose an approach based on quadtrees, where the adaptive grid is represented by the leaves of a quadtree structure [33, 34]. We consider the coarsening criterion to merge quadruples of pixels if the difference in intensities is below a prescribed tolerance ε. The example of the adaptive grid is given in Fig. 20. One can see so-called hanging nodes in the grid, which can cause some troubles in the finite element approach, but can be handled naturally in the finite volume scheme. Our adjustment of the finite volume scheme (3.29) to adaptive grids is the following. In the definition of $g_{pq}^{\sigma,n-1}$ in (3.31), x_{pq} will be the midpoint of the common boundary of two neighboring cells (with possibly unequal areas). Then the difference will be that the sum in (3.9) can be evaluated over unequal control volumes. However, we again compute all possible coefficients of the sum in advance for every larger candidate cell on higher levels of the hierarchy. Let us mention that we allow only the so-called *balanced grids* where the ratio of the sides of neighbouring cells is $1:1$, $1:2$ or $2:1$. If the solution is piecewise constant on non-uniform cells, we can assume exchange of intensity between neighbours just in a strip of unit thickness along a cell boundary (assuming unit size of the cells in the case of a uniform rectangular grid). Then we can define

$$T_{pq} = \min\{l_p, l_q\}, \tag{3.44}$$

where l_p and l_q are the lengths of the sides of two adjacent cells p, q. With these adjustments the scheme (3.29) gives a linear system with a symmetric and strictly diagonally dominant M-matrix which guarantees existence of a unique solution. Moreover, L_∞-stability can be easily proved in every discrete scale step.

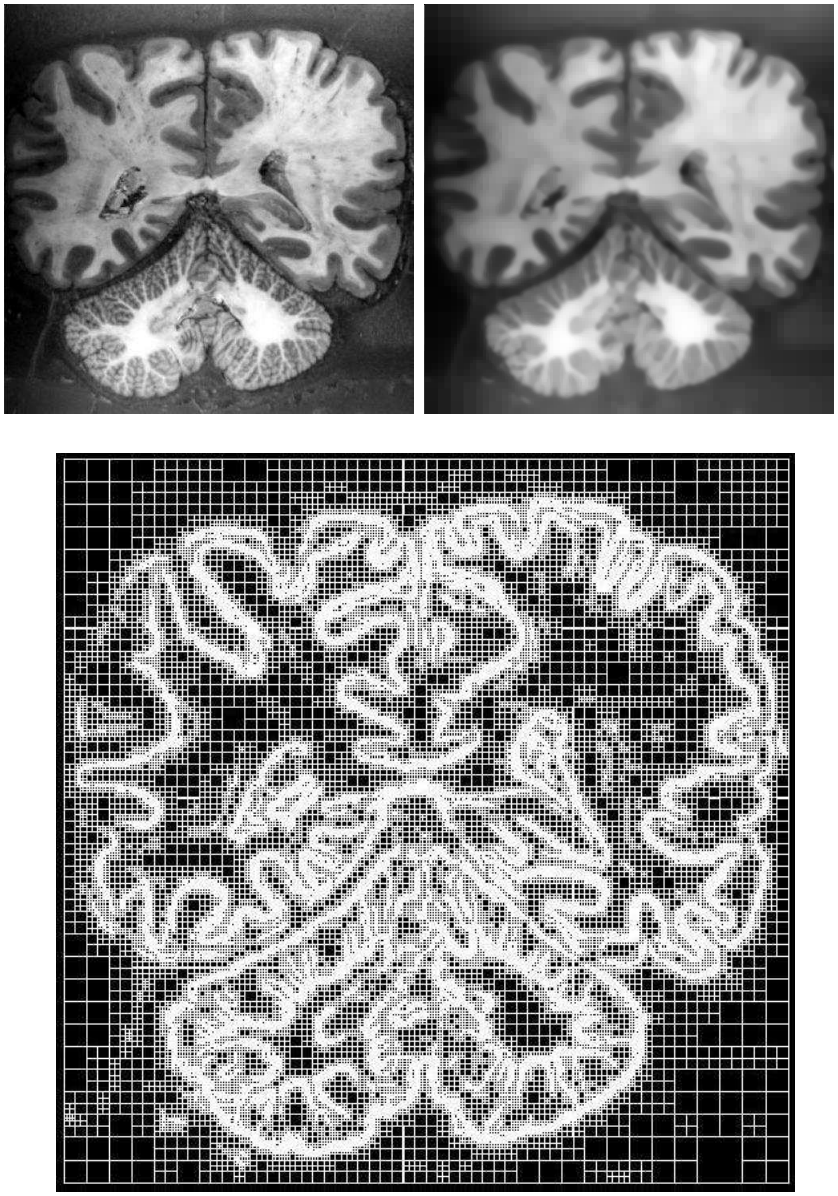

Figure 20: Processing of a medical image [33]; original (top left), multiscale analysis after 25 scale steps (top right), corresponding adaptive finite volume grid (bottom).

Acknowledgements The results presented in this paper were obtained together with Angela Handlovičová, Zuzana Krivá, Fiorella Sgallari, Eberhard Bänsch, Jozef Kačur, Claudio Lamberti, Narisoa Ramarosy, Alessandro Sarti and Daniel Ševčovič. The author is greatly indebted to all these colleagues for cooperation in computational methods for image processing and their applications. We would like to thank the TomTec company for collecting the sequence of 3D echocardiographic data used in our presentation. Our research was carried out with the support of the ICARUS and MINOS European projects at CINECA, Bologna, by SFB 256, IWR, University of Heidelberg, and with the support of the grant VEGA 1/7132/20 of the Scientific Grant Agency of the Slovak Republic. Thanks also go to A. Bourlioux, M. Gander, and G. Sabidussi from Montréal, the organizers of the NATO-ASI, where the lectures containing the material of this paper were presented.

References

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

[2] L. Alvarez and J.-M. Morel, Formalization and computational aspects of image analysis, Acta Numer.,Cambridge Univ. Press, 1994, 1–59.

[3] L. Alvarez, P. L. Lions, and J.-M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.* **29** (1992), 845–866.

[4] E. Bänsch, Local mesh refinement in 2 and 3 dimensions, *IMPACT of Computing in Science and Engineering* **3** (1991), 181–191.

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

[6] E. Bänsch and K. Mikula, Adaptivity in 3D image processing, *Computing and Visualization in Science* **4** (1) (2001), 21–30.

[7] M. Beneš, Mathematical and computational aspects of solidification of pure substances, *Acta Math. Univ. Comenian.* **70** (1) (2001), 124–152.

[8] V. Caselles, J.-M. Morel, and G. Sapiro (eds.), *IEEE Trans. Image Processing* **7** (1998).

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

[10] P. F. Ciarlet, *The Finite Element Method For Elliptic Problems*, North-Holland, Amsterdam, 1978.

[11] Y.-G. Chen, Y. Giga, and S. Goto, Uniqueness and existence of viscosity solutions of generalized mean curvature flow equation, *J. Differential Geom.* **33** (1991), 749–786.

[12] M. G. Crandall, H. Ishii, and P. L. Lions, User's guide to viscosity solutions of second order partial differential equations, *Bull. Amer. Math. Soc. (NS)* **27** (1992), 1–67.

[13] U. Diewald, T. Preusser, M. Rumpf, and R. Strzodka, Diffusion models and their accelerated solution in image and surface processing, *Acta Math. Univ. Comenian.* **70** (1) (2001), 15–34.

[14] G. Dziuk, Convergence of a semidiscrete scheme for the curve shortening flow, *Math. Models Methods Appl. Sci.* **4** (1994), 589–606.

[15] G. Dziuk, Algorithm for evolutionary surfaces, *Numer. Math.* **58** (1991), 603–611.

[16] L. C. Evans and J. Spruck, Motion of level sets by mean curvature I, *J. Differential Geom.* **33** (1991), 635–681.

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

[18] G. Golub and C. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins Univ. Press, Baltimore, MD, 1996.

[19] F. Guichard, Axiomatisation des analyses multi-échelles d'images et de films, Ph.D. Thesis, Université Paris IX Dauphine (1994).

[20] A. Greenbaum, *Iterative Methods For Solving Linear Systems*, SIAM, Philadelphia, PA, 1999.

[21] I. Gustafsson, Modified incomplete Cholesky (MIC) methods, in: *Preconditioning Methods, Theory and Applications* (D. J. Evans, e.d), Gordon and Breach, New York, 1983, 265–293.

[22] A. Handlovičová, K. Mikula, and A. Sarti, Numerical solution of parabolic equations related to level set formulation of mean curvature flow, *Computing and Visualization in Science* **1** (2) (1999), 179–182.

[23] A. Handlovičová, K. Mikula, and F. Sgallari, Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution, *Numer. Math.* DOI 10.1007/s002110100374 (2001).

[24] A. Handlovičová, K. Mikula, and F. Sgallari, Variational numerical methods for solving nonlinear diffusion equations arising in image processing, *J. Visual Communication and Image Representation* (to appear).

[25] J. Kačur, *Method of Rothe in Evolution Equations*, Teubner-Texte Math. **80**, Leipzig, 1985.

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

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

[28] M. Kass, A. Witkin, and D. Terzopulos, Snakes: active contour models, *Internat. J. Computer Vision* **1** (1987), 321–331.

[29] S. Kichenassamy, The Perona-Malik paradox, *SIAM J. Appl. Math.* **57** (5) (1997), 1328–1342.

[30] R. Kimmel, R. Malladi, and N. Sochen, Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images, *Internat. J. Computer Vision* **39** (2) (2000), 111–129.

[31] M. Kimura, Numerical analysis for moving boundary problems using the boundary tracking method, *Japan J. Indust. Appl. Math.* **14** (1997), 373–398.

[32] J. J. Koenderink, The structure of images, *Biol. Cybern.* **50** (1984), 363–370.

[33] Z. Krivá and K. Mikula, An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing, *J. Visual Communication and Image Representation* (to appear).

[34] Z. Krivá and K. Mikula, An adaptive finite volume scheme in processing of color images, in: *Proc. ALGORITMY 2000, Conf. on Scientific Computing, Podbanské, Slovakia (2000)*, 174–188.

[35] A. Kufner, O. John, and S. Fučík, *Function Spaces*, Academia, Prague, 1977.

[36] O. A. Ladyženskaja, N. N. Uralceva, *Linear and Quasilinear Elliptic Equations*, Academic Press, New York, NY, 1968.

[37] R. Malladi, J. A. Sethian, and B. Vemuri, Shape modeling with front propagation: a level set approach, *IEEE Trans. Pattern Anal. Machine Intelligence* **17** (2) (1995), 158–174.

[38] J. A. Meijerink and H. A. van der Vorst, An iterative solution method for systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comp.* **31** (1977), 148–162.

[39] K. Mikula, Solution of nonlinear curvature driven evolution of plane convex curves, *Appl. Numer. Math.* **23** (1997), 347–360.

[40] K. Mikula and J. Kačur, Evolution of convex plane curves describing anisotropic motions of phase interfaces, *SIAM J. Sci. Comput.* **17** (1996), 1302–1327.

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

[42] K. Mikula, A. Sarti, and C. Lamberti, Geometrical diffusion in 3D-echocardiography, in: *Proc. ALGORITMY'97 — Conf. on Scientific Computing* (J. Kačur and K. Mikula, eds.), Acta Math. Univ. Comenian. (N.S.) **67**, Comenius Univ. Press, Bratislava, 1998, 167–181.

[43] K. Mikula and D. Ševčovič, Solution of nonlinearly curvature driven evolution of plane curves, *Appl. Numer. Math.* **31** (1) (1999), 191–207.

[44] K. Mikula and D. Ševčovič, Evolution of plane curves driven by a nonlinear function of curvature and anisotropy, *SIAM J. Appl. Math.* **61** (5) (2001), 1473–1501.

[45] K. Mikula and D. Ševčovič, A direct method for solving an anisotropic mean curvature flow of planar curve with an external force, Preprint, Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Bratislava, 2001.

[46] M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert (eds.), *Scale-Space Theories in Computer Vision*, Lecture Notes in Comput. Sci. **1682**, Springer, Berlin, 1999.

[47] M. Nitzberg and T. Shiota, Nonlinear image filtering with edge and corner enhancement, *IEEE Trans. Pattern Analysis and Machine Intelligence* **14** (8) (1992), 826–833.

[48] R. Nochetto, M. Paolini, and C. Verdi, Sharp error analysis for curvature dependent evolving fronts, *Math. Models Methods Appl. Sci.* **3** (6) (1993), 711–723.

[49] K. N. Nordström, Biased anisotropic diffusion — A unified approach to edge detection, Preprint, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1989.

[50] S. Osher and J. Sethian, Front propagating with curvature dependent speed: algorithms based on the Hamilton-Jacobi formulation, *J. Comput. Phys.* **79** (1988), 12–49.

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

[52] T. Preusser and M. Rumpf, An adaptive finite element method for large scale image processing, in: *Proceedings of ScaleSpace'99* (1999), 223–234.

[53] L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* **60** (1992), 259–268.

[54] Y. Saad, *Iterative Methods For Sparse Linear Systems*, PWS Publ. Comp., 1996.

[55] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*, Cambridge Univ. Press, 2001.

[56] A. Sarti, K. Mikula, and F. Sgallari, Nonlinear multiscale analysis of 3D echocardiographic sequences, *IEEE Trans. Medical Imaging* **18** (6) (1999), 453–466.

[57] A. Schmidt and K. G. Siebert, ALBERT — Software for scientific computations and applications, *Acta Math. Univ. Comenian.* **70** (1) (2001), 105–122.

[58] J. A. Sethian, Numerical algotithm for propagating interfaces: Hamilton-Jacobi equations and conservation laws, *J. Differential Geom.* **31** (1990), 131–161.

[59] J. A. Sethian, *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge Univ. Press, 1999.

[60] N. J. Walkington, Algorithms for computing motion by mean curvature, *SIAM J. Numer. Anal.* **33** (6) (1996), 2215–2238.

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

[62] J. Weickert, B. M. t. H. Romeny, M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Processing* **7** (1998), 398–410.

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

[64] R Whitacker and G. Gerig, Vector-valued diffusion, *Geometry Driven Diffusion in Computer Vision* (B. M. t. H. Romemy, ed.), Kluwer, Dodrecht, 1994.

[65] A. P. Witkin, Scale-space filtering, in: *Proc. Eight Internat. Conf. on Artificial Intelligence*, vol. 2 (1983), 1019–1022.

[66] S. Wolfram, *The Mathematica Book*, 3rd ed., Cambridge Univ. Press, 1996.