Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim
D-68131 Mannheim, Germany

Reihe Informatik
5/1999

Fast Parallel Algorithms for a Broad Class of Nonlinear Variational Diffusion Approaches

J. Weickert, J. Heers, C. Schnörr, K.J. Zuiderveld,
O. Scherzer, H.S. Stiehl

Technical Report 5/1999
Computer Science Series
March 1999

Fast Parallel Algorithms for a Broad Class of Nonlinear Variational Diffusion Approaches

J. Weickert[1], J. Heers[2], C. Schnörr[1], K.J. Zuiderveld[3], O. Scherzer[4], H.S. Stiehl[2]

[1]Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim
D-68131 Mannheim, Germany
E-mail: {Joachim.Weickert, Christoph.Schnoerr}@ti.uni-mannheim.de

[2]Department of Computer Science
University of Hamburg
Vogt-Kölln-Str. 30
D-22527 Hamburg, Germany
E-mail: {heers, stiehl}@informatik.uni-hamburg.de

[3]Image Sciences Institute
Utrecht University Hospital
HP E01.334, P.O. Box 85500
NL-3508 GA Utrecht, The Netherlands
E-mail: karel@cv.ruu.nl

[4]Industrial Mathematics Institute
University of Linz
Altenberger Str. 69
A-4040 Linz, Austria
E-mail: scherzer@indmath.uni-linz.ac.at

Abstract

Variational segmentation and nonlinear diffusion approaches have been very active research areas in the fields of image processing and computer vision during the last years. In the present paper, we review recent advances in the development of efficient numerical algorithms for these approaches. The performance of parallel implementations of these algorithms on general–purpose hardware is assessed. A mathematically clear connection between variational models and nonlinear diffusion filters is presented that allows to interpret one approach as an approximation of the other, and vice versa. Numerical results confirm that, depending on the parametrization, this approximation can be made quite accurate. Our results provide a perspective for uniform implementations of both nonlinear variational models and diffusion filters on parallel architectures.

Keywords: energy minimization, variational segmentation, nonlinear diffusion, numerical linear algebra, parallel computing, medical imaging

Sub–topics: 1.2 fast numerical procedures, 1.4 parallel implementation

1 Introduction

Variational approaches for image processing and computer vision have been the subject of considerable interest during the last years. A growing community has focused on various research problems including the mathematical foundations of image segmentation [27], nonlinear regularization approaches to image restoration [30, 17], stochastic modeling of spatial context [24, 42], and in a wide range of corresponding applications [28]. In each case, the approach is formulated in terms of an energy functional which precisely specifies the goal of the criteria being used for judging the output of a processing stage. Often results from various branches of mathematics like functional analysis, convex optimization, or numerical mathematics, for example, can successfully be applied to clarify the properties of a variational approach. As a result, variational modeling has contributed much to the fields of image processing and computer vision during the last years.

Parallel to this development, the application of partial differential equations (PDE's) to image processing has been a very active field of research [18, 39, 9]. Research problems that motivated corresponding work include nonlinear extensions to the linear scale–space paradigm [1], invariant scale–spaces [31, 32], active contours and surfaces [25, 8], and sound mathematical models for image enhancement through local backward diffusion. For a review, we refer to [38, 39].

In many cases, these approaches are superior to conventional existing image processing methods and thus provide new perspectives for various application areas like, for example, medical image analysis. Consequently, there is an increasing interest in efficient numerical algorithms enabling implementations of these computationally demanding approaches that work at acceptable computing speed. For example, a challenging task concerns the processing of 3D medical image data with a PDE–based approach such that interactivity becomes feasible.

In the present paper, we review recent advances in the development of efficient numerical algorithms for both variational modeling approaches [20, 21] and PDE–based approaches [41, 40] to nonlinear adaptive image processing. Numerical experiments with parallel implementations of these algorithms on general–purpose hardware are reported. Although the underlying mathematical models appear to be quite different (non–quadratic minimization problems vs. nonlinear evolution equations), we show that they can lead to very similar algorithms. One important tool to understand these close relations is a connection which has recently been exploited in [33] to establish scale-space properties for regularization methods. Our paper may also shed light on how the algorithmic advances reported may interrelate in future work.

The remainder of this paper is organized as follows. Section 2 reviews non-quadratic variational models for adaptive image processings and discusses a connection to approaches based on nonlinear parabolic evolution equations. Section 3 is devoted to efficient numerical approximations by means of finite elements and finite difference methods. It describes a linearization technique as well as corresponding algorithms suited to compute minimizing functions efficiently. Furthermore it is shown how similar ideas can be used for diffusion filtering, and how they can be modified towards an even more efficient additive splitting scheme. Numerical experiments with parallel implementations of our algorithms using general–purpose hardware are reported in Section 4. We conclude with pointing out possible directions of further research in Section 5.

2 Continuous formulations

2.1 Variational approaches

In this section, we sketch the mathematical formulation of a variational approach for adaptive image processing. For a more detailed account we refer to [35].

Let $g : x \in \Omega \to \mathbb{R}$ denote the grey-value function of a given image defined over an m-dimensional image domain $\Omega \subset \mathbb{R}^m$. The goal is to find a filtered version u of the original image g such that u minimizes a suitable energy functional which precisely specifies the goal of our processing step. We consider minimization problems of the following form:

$$u = \inf_{v \in \mathcal{H}} J(v) \,, \quad J(v) = \frac{1}{2} \int_\Omega \Big\{ (v-g)^2 + \alpha \cdot \lambda(|\nabla v|) \Big\} dx \tag{1}$$

where $\mathcal{H}$ is some Hilbert space and $\alpha > 0$ serves as a weight factor. The function λ depends on the gradient of admissible functions v and is piecewise defined:

$$\lambda(s) = \begin{cases} \lambda_{low}(s) & 0 \le s \le c \\ \lambda_{high}(s) & s > c. \end{cases}$$

Typical definitions include $\lambda_{low}(s) \sim s^2$ and $\lambda_{high}(s) \sim s$ or $\lambda_{high}(s) \sim const.$ (cf., e.g., [7, 15, 6]). Accordingly, the functional (1) can be written in a form which is easier to interpret:

$$J(v) = \frac{1}{2} \Big[\int\limits_\Omega (v-g)^2 \, dx + \alpha \int\limits_{\{x:|\nabla v| \le c\}} \lambda_{low}(|\nabla v|) \, dx + \alpha \int\limits_{\{x:|\nabla v| > c\}} \lambda_{high}(|\nabla v|) \, dx \Big]. \tag{2}$$

The first term of the right hand side of (2) measures the similarity between functions v and given image data g. The second term measures the smoothness of functions v within regions with a low gradient of v. The third term measures properties of functions v where v refers to local features in terms of significant variations. For example, using $\lambda_{high}(s) \sim s$, this term measures the length of iso–contour lines of v summed up over the local contrast (cf. [34]). As a result, algorithms minimizing the functional (1) lead to an approximation of the given image function g with a piecewise smooth function u. Since the domains of integration in (2) depend on the solution u itself, the corresponding local smoothing process generating u from g is nonlinear and adaptive. Figures 1 and 2 show examples.

Vanishing of the first variation of the functional J (1) yields a variational equation determining minimizing functions u:

$$L(u,v) := \int_\Omega \Big\{ (u-g)v + \alpha \, \rho(|\nabla u|) \nabla u \cdot \nabla v \Big\} dx = 0 \qquad \forall v \in \mathcal{H} \tag{3}$$

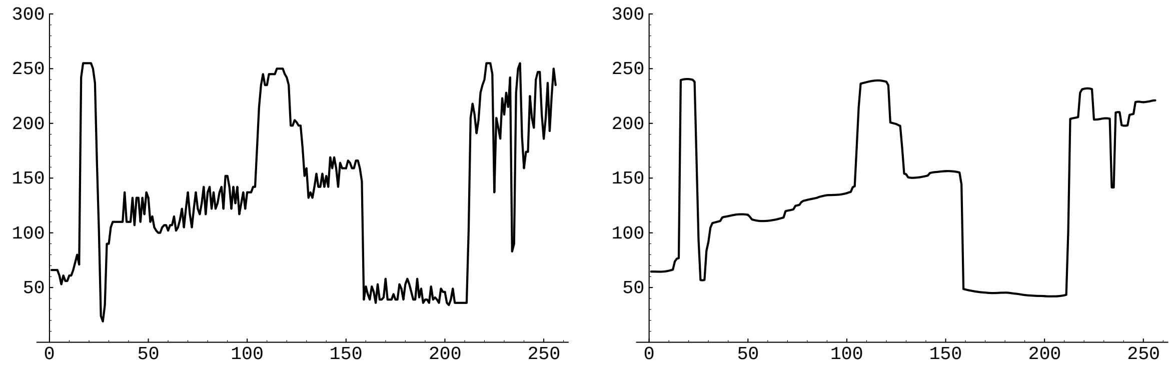

Figure 1: Nonlinear adaptive smoothing of a 1D-signal

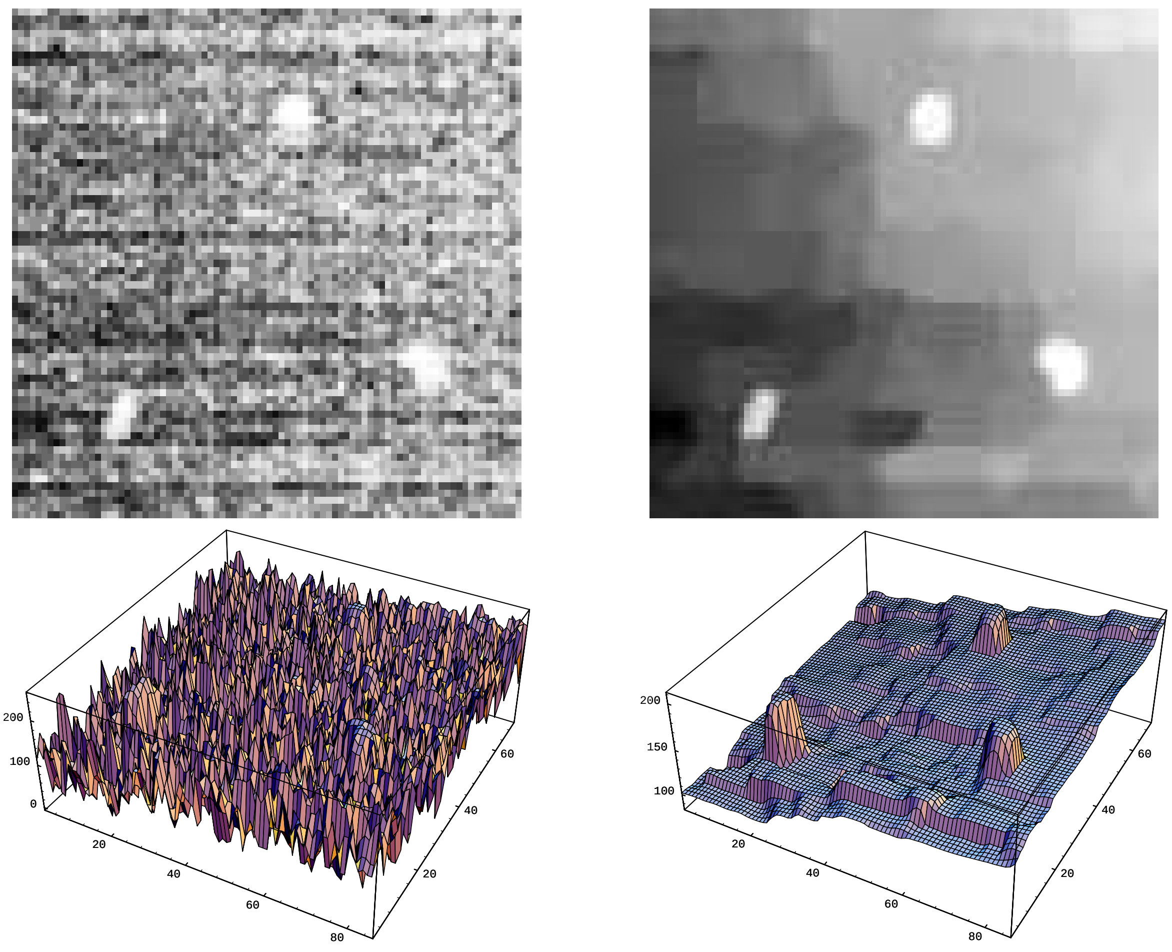

Figure 2: Top: Variational restoration of a mammography. Bottom: The images from above as 3D–plots.

where the so–called diffusivity function $\rho(\cdot)$ is derived from $\lambda(\cdot)$ as

$$\rho(s) = \frac{\lambda'(s)}{2s}. \tag{4}$$

Partial integration in (3) shows that u is also a (weak) solution to the Euler equation

$$\alpha \operatorname{div}\bigl(\rho(|\nabla u|)\nabla u\bigr) - (u - g) = 0 \tag{5}$$

with reflecting (homogeneous Neumann) boundary conditions:

$$\frac{\partial u}{\partial n} \;=\; 0 \quad \text{on } \partial\Omega, \tag{6}$$

where n denotes the normal to the image boundary $\partial\Omega$.

2.2 Related diffusion filters

2.2.1 Diffusion filters with monotone fluxes

Interestingly, the preceding variational approach has strong connections to nonlinear diffusion filtering. This shall be explained in the sequel.

A simple nonlinear diffusion filter can be constructed as follows. A processed version $u(x,t)$ of the original image $g(x)$ is obtained by solving the diffusion equation

$$\partial_t u \;=\; \operatorname{div}\Bigl(\rho(|\nabla u|)\;\nabla u\Bigr) \tag{7}$$

with the original image as initial state,

$$u(x,0) \;=\; g(x), \tag{8}$$

and reflecting boundary conditions:

$$\frac{\partial u}{\partial n} \;=\; 0 \quad \text{on } \partial\Omega. \tag{9}$$

The "time" t is a scale parameter: larger values lead to simpler image representations. One can assure well-posedness of the diffusion filter if the diffusivity $\rho(s)$ creates a flux $\rho(s)\cdot s$, which is monotonically increasing for $s \geq 0$. Such diffusivities ρ arise from equation (4) when the smoothness potential λ is convex.

By writing the Euler equation (5) as

$$\frac{u-g}{\alpha} = \operatorname{div}\bigl(\rho(|\nabla u|)\nabla u\bigr) \tag{10}$$

it becomes clear that variational image restoration with regularization parameter α approximates a diffusion filter at time α: indeed, (10) is nothing more than an implicit time discretization of the diffusion equation (7). This relation has been exploited in [33] for establishing various relations between variational image restoration and nonlinear diffusion filtering.

One would expect that this discretization is of higher accuracy for smaller α. However, Figure 3 shows that also for rather large α, the approximation remains surprisingly good. In this example we have used a diffusivity of type

$$\rho(s) = \frac{1}{\sqrt{1 + s^2/\nu^2}} \tag{11}$$

with $\nu = 2$. It can be derived from the convex potential

$$\lambda(s) = 2\nu^2 \sqrt{1 + s^2/\nu^2}. \tag{12}$$

This example illustrates that *variational image restoration and nonlinear diffusion filtering are basically equivalent.* As a consequence, efficient numerical methods for one of these paradigms can also be used for the other.

We note that the diffusivity (11) is a decreasing function. As a consequence diffusion is small at those locations where the gradient is large which in turn reduces blurring at edges.

2.2.2 Diffusion filters with nonmonotone fluxes

Intuitively one would expect that edges are better preserved if the diffusivity decreases more rapidly. For instance, a diffusivity of type

$$\rho(s) \ := \begin{cases} 1 & (s = 0) \\ 1 - \exp\left(\frac{-3.315}{(s/\nu)^4}\right) & (s > 0) \end{cases} \tag{13}$$

decreases more rapidely than (11).

However, the corresponding potential λ is only convex for gradient magnitudes which do not exceed ν: the factor 3.315 ensures that the flux $= s\rho(s)$ is increasing for $|s| \leq \nu$ and decreasing for $|s| > \nu$. Thus, ν is a contrast parameter separating low-contrast regions with (smoothing) forward diffusion from high-contrast locations where backward diffusion may enhance edges [29]. In this case, we should not expect the existence of a unique solution which is stable.

However, it is possible to regularize diffusion filters with nonmonotone fluxes in such a way that they become well-posed [10]. This can be achieved by replacing the edge detector $|\nabla u|$ in $\rho(|\nabla u|)$ by a Gaussian derivative $|\nabla u_\sigma|$:

$$\nabla u_\sigma \ := \ \nabla(K_\sigma * u), \tag{14}$$

$$K_\sigma \ := \ \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{|x|^2}{2\sigma^2}\right). \tag{15}$$

After some time this filter creates segmentation-like results which are piecewise almost constant. This is illustrated in Figure 4. For $t \to \infty$, however, the image converges to a constant grey-value function [39]. Well-posedness results for this filter can be found in [10, 39] and a scale-space interpretation is given in [39].

It should be noted that such regularized diffusion filters cannot be written as a variational problem. We have presented them here because they offer additional perspectives in terms of contrast enhancement. Moreover, they can be treated with the same numerical techniques that are used for unregularized diffusion filters with monotone fluxes.

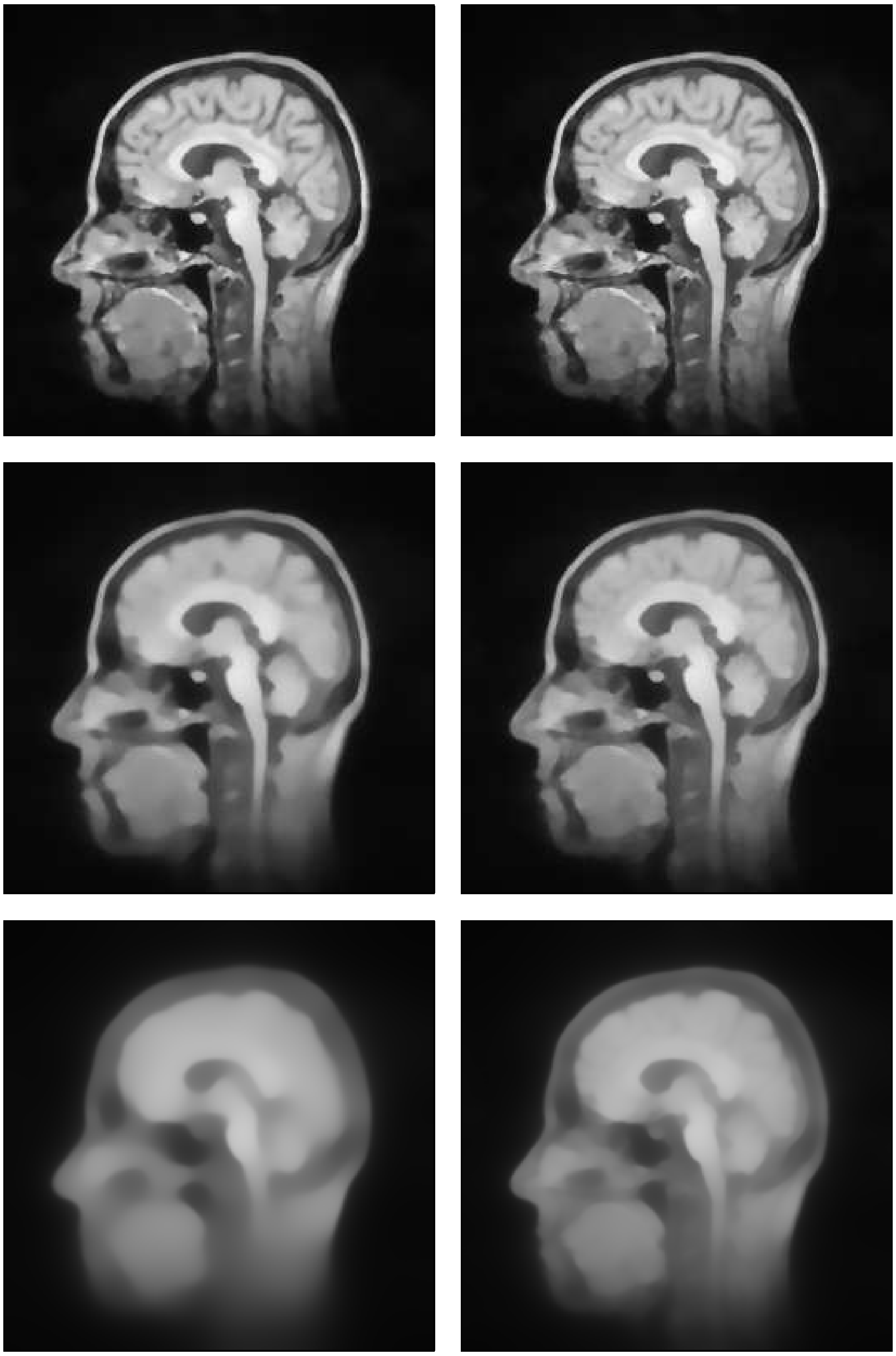

Figure 3: Comparison between the nonlinear diffusion filter (7) and the variational image restoration (10). (A) LEFT COLUMN: Temporal evolution of the diffusion filter, $t = 10$, 30, 100. (B) RIGHT COLUMN: Variational restoration with regularization parameters $\alpha = 10$, 30, 100.

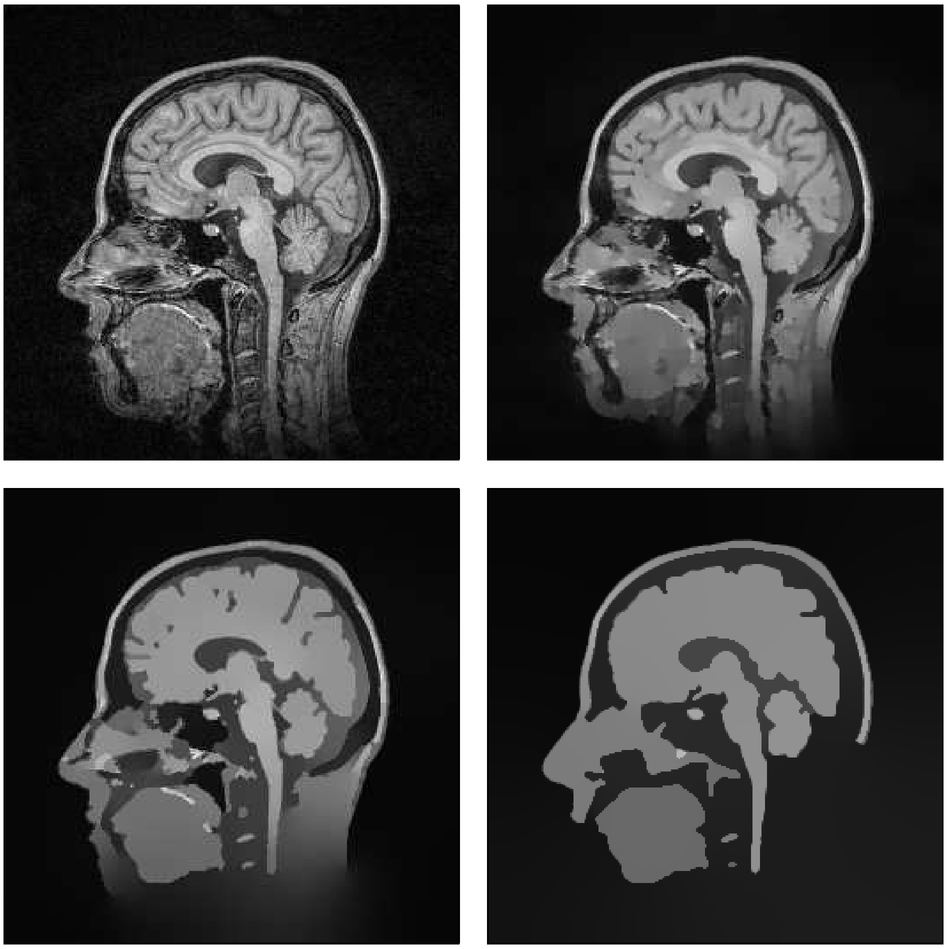

Figure 4: Temporal evolution of the regularized nonlinear diffusion filter with nonmonotone flux function.

3 Discrete formulations

3.1 Variational approach

3.1.1 Linearization

The goal of this section is to replace the nonlinear equation (3) by a sequence of linear equations, the solution methods of which are well understood and suited for parallel implementations. In the following, we describe a particular linearization technique introduced to image processing by Geman and Reynolds [16], and further studied in [12, 21]. For alternative linearization techniques and a comparison within a more general framework we refer to [21].

The key idea is to replace the non-quadratic functional (1) – from which equation (3) is derived – by a sequence of quadratic functionals. Following Geman and Reynolds [16], we introduce an auxiliary function w and consider the problem of minimizing

$$\tilde{J}(v,w) = \frac{1}{2}\int_\Omega \Big\{(v-g)^2 + \alpha\cdot\big(w|\nabla v|^2 + \psi(w)\big)\Big\}dx \tag{16}$$

where the function $\psi(\cdot)$ depends on $\lambda(\cdot)$ in (1) and is chosen such that $\tilde{J}(v,w)$ is convex in w and [16]:

$$J(v) = \inf_w \tilde{J}(v,w).$$

The minimization of (16) is accomplished by the following two–step procedure (k denotes the iteration index):

$$w^k = \arg\min_w \tilde{J}(u^k, w), \tag{17}$$

$$u^{k+1} = \arg\min_u \tilde{J}(u, w^k). \tag{18}$$

Variational calculus shows that equation (18) is equivalent to

$$\int_\Omega \Big\{(u^{k+1}-g)v + \alpha\, w^k \nabla u^{k+1}\cdot\nabla v dx\Big\}dx = 0. \tag{19}$$

After discretization, u^{k+1} can be computed as solution to a *linear* system of equations.

Making equation (17) more explicit, however, is not as straightforward as with equation (18), in the general case. Nevertheless, the computation of w^k is not difficult due to the convexity of $\tilde{J}(v,w)$ with respect to w. In the particular case of a *convex* original functional (1) equation (17) explicitly reads:

$$w^k = \rho(|\nabla u^k|)$$

with $\rho(\cdot)$ from (4). Note that in this case equation (19) amounts to "freeze" the nonlinear part of equation (3) for one iteration step. To our knowledge, this so-called *Kačanov method* has been introduced more than 30 years ago [22, 14]. Nevertheless, it turned out to be both efficient and competitive with respect to other techniques (see Section 4 and [21]).

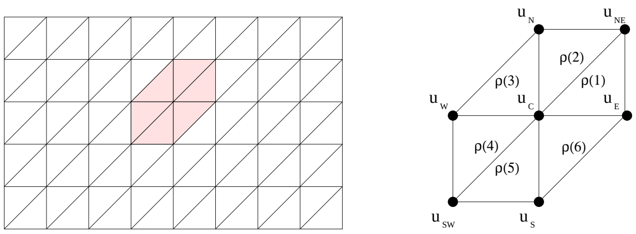

Figure 5: **Left:** Triangulation of the rectangular image domain Ω with mesh width 1. The nodes correspond to pixel positions. **Right:** An interior node u_C with adjacent triangles d_i and the corresponding ρ-values (for piecewise linear basis functions $\rho(k) = \rho(|\nabla u|) = const.$ for each triangle d_k).

3.1.2 Finite element discretization

In this section we explain briefly how a discrete version of the variational approach (3) can be obtained using the Finite Element Method. Note that the linear system corresponding to (19) can easily be computed by replacing $\rho(|\nabla u|)$ in (3) with w^k. For a sound introduction to the Finite Element Method we refer to, e.g., [13]. The Finite Element Method can be extended easily to adaptive algorithms with a coarse resolution in homogeneous regions and a fine one around edges. For more details on such speed-up techniques we refer to [34] in the context of variational image restoration, and to [3] in the context of nonlinear diffusion filtering.

The basic idea behind the Finite Element Method is the restriction of optimization problems to finite–dimensional subspaces. Let $\{\phi_1, ..., \phi_N\}$ be basis functions of a finite–dimensional subspace $\mathcal{H}_h \subset \mathcal{H}$. Then, the restriction of (3) to $\mathcal{H}_h$ reads:

$$L(u_h, \phi_i) = 0 \;, \quad \forall i = 1, ..., N \;, \tag{20}$$

with minimizer $u_h \in \text{span}\{\phi_1, ..., \phi_N\}$. If we define the mappings

$$I : \mathbb{R}^N \to \mathcal{H}_h; \;\mathbf{u} \to \sum_j u_j \phi_j \;, \tag{21}$$

and

$$\mathbf{L}_i(\mathbf{u}) := L(I(\mathbf{u}), \phi_i) \;, \quad \mathbf{u} = (u_1, ..., u_N)^T \;, \tag{22}$$

then the solution of (20) is equivalent to the solution of the nonlinear system:

$$\mathbf{L}(\mathbf{u}) = \mathbf{0} \;, \quad \mathbf{L} = (\mathbf{L}_1, ..., \mathbf{L}_N)^T \;. \tag{23}$$

For the case of two–dimensional (2D) grey-value images we use piecewise linear basis functions as follows. The first step is to triangulate the underlying image domain, in this case the rectangular area $\Omega = [0, N_1 - 1] \times [0, N_2 - 1]$, as illustrated in Figure 5. Next, we assign to each mesh node $P_{i,j}$ a basis function $\phi_{i,j}$ which is uniquely defined by the following conditions:

$$\begin{array}{ll} \phi_{i,j}(x) & \text{is linear within each triangle } d_k \ , \\ \phi_{i,j}(x) = 1 & \text{at node } P_{i,j} \ , \\ \phi_{i,j}(x) = 0 & \text{at every node } P_{k,l} \neq P_{i,j} \ . \end{array}$$

Given discrete grey-value images, and elements of the subspace $\mathcal{H}_h$ in general, images are represented by simply interpolating the values of corresponding nodal variables $v_{i,j}$ in a piecewise linear fashion:

$$I : \mathbb{R}^{N_1 \times N_2} \to \mathcal{H}_h \ , \quad \mathbf{v} \to \sum_{i,j} v_{i,j} \phi_{i,j}$$

with grey-value $v_{i,j}$. From (21) and (22) we thus obtain:

$$\sum_{k,l} (u_{k,l} - g_{k,l}) \int_\Omega \phi_{k,l} \phi_{i,j} \, dx + \sum_{k,l} u_{k,l} \int_\Omega \rho(|\nabla u_h|) \nabla \phi_{k,l} \cdot \nabla \phi_{i,j} \, dx = 0. \tag{24}$$

These integrals vanish for all pairs of nodes (i,j) and (k,l) which have no triangle in common. The remaining integrals can be computed analytically to obtain a sparse system of nonlinear equations in terms of the nodal variables of the solution image $\mathbf{u}$. Additional details and applications to different variational problems can be found in [35].

The expressions in (24) are weighted sums. Applying the linearization technique described in section 3.1.1, that is "freezing" the nonlinear part as in equation (19), the nonlinear system (24) reduces to a linear system of type

$$\mathbf{B}^k \mathbf{u}^{k+1} = \mathbf{b} \ , \tag{25}$$

where

$$\mathbf{B}^k = \mathbf{D} + \mathbf{R}(\mathbf{u}^k) \ , \qquad \mathbf{b} = \mathbf{D}\mathbf{g} \ .$$

The matrices $\mathbf{D}$ and $\mathbf{R}$ are sparse, since they describe convolutions within a 3×3 neighbourhood. For some inner node, the corresponding stencil notations are given by

$$\begin{array}{|c|c|c|} \hline 0 & \frac{1}{12} & \frac{1}{12} \\ \hline \frac{1}{12} & \frac{1}{2} & \frac{1}{12} \\ \hline \frac{1}{12} & \frac{1}{12} & 0 \\ \hline \end{array} \tag{26}$$

for $\mathbf{D}$, and by

$$\begin{array}{|c|c|c|} \hline 0 & \frac{1}{12} - \alpha \frac{\rho(2)+\rho(3)}{2} & \frac{1}{12} \\ \hline \frac{1}{12} - \alpha \frac{\rho(3)+\rho(4)}{2} & \begin{array}{c} \frac{1}{2} + \alpha \frac{\rho(2)+\rho(3)}{2} + \alpha \frac{\rho(3)+\rho(4)}{2} \\ +\alpha \frac{\rho(1)+\rho(6)}{2} + \alpha \frac{\rho(5)+\rho(6)}{2} \end{array} & \frac{1}{12} - \alpha \frac{\rho(1)+\rho(6)}{2} \\ \hline \frac{1}{12} & \frac{1}{12} - \alpha \frac{\rho(5)+\rho(6)}{2} & 0 \\ \hline \end{array} \tag{27}$$

for the system matrix $\mathbf{B}^k$. It should be noted that the entries $\rho(.)$ are functions of $\mathbf{u}^k$; see also Figure 5. The necessary modifications of these stencils at boundary points are automatically obtained by taking into consideration the correct domain of integration in (24). It is not hard to see that $\mathbf{B}^k$ is symmetric and positive definite.

3.2 Nonlinear diffusion filtering

3.2.1 Diffusion interpretation of the Kačanov method

It is instructive to study also a finite difference discretization of the Kačanov method for variational image restoration: it follows that such a discretization can be regarded as a fixed point iteration for solving the implicitly time-discretized diffusion equation (10). This can be seen in the following way.

A discrete m-dimensional image can be regarded as a vector $\mathbf{g} \in \mathbb{R}^N$, whose components g_i, $i \in \{1, ..., N\}$, denote the grey-values at the pixels. Pixel i represents the location x_i. By u_i and ρ_i we denote approximations to $u(x_i, t)$ and $\rho(|\nabla u(x_i, t)|)$, respectively, where the gradient is replaced by central differences.

A spatial finite difference discretization of (10) is given by

$$\frac{u_i - g_i}{\alpha} \;=\; \sum_{l=1}^{m} \sum_{j \in \mathcal{N}_l(i)} \frac{\rho_j + \rho_i}{2}\,(u_j - u_i), \tag{28}$$

where $\mathcal{N}_l(i)$ consists of the two neighbours of pixel i along the l direction (boundary pixels may have only one neighbour). In vector–matrix notation this becomes

$$\frac{\mathbf{u} - \mathbf{g}}{\alpha} \;=\; \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u})\,\mathbf{u}, \tag{29}$$

where $\mathbf{A}_l$ describes the diffusive interaction in l direction.
This equation can be rewritten in fixed point structure as

$$\mathbf{u} \;=\; \Big(\mathbf{I} - \alpha \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u})\Big)^{-1} \mathbf{g}. \tag{30}$$

A corresponding fixed point iteration is given by

$$\mathbf{u}^{k+1} \;=\; \Big(\mathbf{I} - \alpha \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\Big)^{-1} \mathbf{g}. \tag{31}$$

This is nothing else but the finite difference approximation

$$\frac{\mathbf{u}^{k+1} - \mathbf{g}}{\alpha} \;=\; \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\,\mathbf{u}^{k+1} \tag{32}$$

to the Kačanov method (19), whose Euler equation is given by

$$\frac{u^{k+1} - g}{\alpha} \;=\; \operatorname{div}\Big(\rho(|\nabla u^k|)\,\nabla u^{k+1}\Big). \tag{33}$$

The fixed point iteration (31) requires to solve the sequence of linear systems

$$\Big(\mathbf{I} - \alpha \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\Big)\, \mathbf{u}^{k+1} \;=\; \mathbf{g} \tag{34}$$

with $k = 1, ..., \infty$. Since this method is globally convergent, we may use an arbitrary initial vector $\mathbf{u}^0$. A natural choice is $\mathbf{u}^0 := \mathbf{g}$. In practice, one stops the iterations when $\mathbf{u}^k$ and $\mathbf{u}^{k+1}$ are sufficiently close, or when the residue obtained from plugging $\mathbf{u}^k$ into (29) is close to 0.

The system matrix for the finite difference Kačanov scheme (34) can be represented by a convolution stencil of type

0	$-\alpha\,\frac{\rho(N)+\rho(C)}{2}$	0
$-\,\alpha\,\frac{\rho(W)+\rho(C)}{2}$	$1+\alpha\,\frac{\rho(N)+\rho(C)}{2}+\alpha\,\frac{\rho(W)+\rho(C)}{2}$ $+\alpha\,\frac{\rho(E)+\rho(C)}{2}+\alpha\,\frac{\rho(S)+\rho(C)}{2}$	$-\alpha\,\frac{\rho(E)+\rho(C)}{2}$
0	$-\alpha\,\frac{\rho(S)+\rho(C)}{2}$	0

(35)

where the diffusivity $\rho(C)$ refers to some inner pixel, and $\rho(N)$, $\rho(W)$, $\rho(E)$, and $\rho(S)$ represent the northern, western, eastern, and southern neighbours, respectively. They are all functions of $\mathbf{u}^k$. We observe large structural similarities to the finite element Kačanov method. Both the finite difference and the finite element method boil down to the solution of a sequence of linear systems of equations with a sparse and symmetric positive definite system matrix.

3.2.2 A semi-implicit scheme for nonlinear diffusion filtering

If one is interested in solving the nonlinear diffusion equation (7) for some time α, one may proceed iteratively in n steps with step size $\tau = \alpha/n$. Let us denote by $\mathbf{u}^k$ the finite difference approximation at time $k\tau$. One possibility for a finite difference discretization to (7) is the scheme

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} \;=\; \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\, \mathbf{u}^{k+1} \tag{36}$$

with $\mathbf{u}^{(0)} := \mathbf{g}$. We observe that we have to solve n linear systems of type

$$\Big(\mathbf{I} - \tau \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\Big)\, \mathbf{u}^{k+1} \;=\; \mathbf{u}^k. \tag{37}$$

Since this scheme does not give the solution $\mathbf{u}^{k+1}$ directly (explicitly), but requires to solve a linear system first, it is called a *linear-implicit (semi-implicit)* method. Its system matrix has the same structure as the one used in the finite difference Kačanov method.

In the 1-D case the system matrix is tridiagonal and diagonally dominant. For such a system a Gaussian algorithm for tridiagonal systems (also called *Thomas algorithm*) is stable and solves the problem with linear complexity, both with respect to memory and computational time.

For dimensions $m \geq 2$, however, it is not possible to order the pixels in such a way that in the i-th row all nonvanishing elements of the system matrix can be found within the positions $[i, i-m]$ to $[i, i+m]$: Usually, the matrix possesses a much larger bandwidth. Applying direct algorithms such as Gaussian elimination would destroy the zeros within the band and would lead to an immense storage and computation effort. Classical iterative algorithms like the Jacobi or Gauß–Seidel method become slow for large τ, since this increases the condition number of the system matrix. Just like in the finite element case, a preconditioned conjugate gradient technique will be quite efficient.

This shows that there is also a large amount of structural similarities between discrete variational approaches and discrete nonlinear diffusion filters. In contrast to variational image restoration, however, the number of linear systems to be solved is *a-priori* fixed to n and no *a posteriori* stopping criterion is required. Clearly, choosing a larger n improves the approximation quality.

A possible speed advantage of semi-implicit nonlinear diffusion filtering as compared to the Kačanov method for variational image restoration may result from the fact that the time step size τ is usually smaller than the regularization parameter α. Therefore, nonlinear diffusion filtering has a better-conditioned system matrix than variational image restoration. Thus, one may expect that iterative solvers for linear systems whose convergence depends on the condition number will converge faster for nonlinear diffusion filters. This also applies to preconditioned conjugate methods, although to a much smaller amount than in classical iterative solvers.

3.2.3 AOS schemes for nonlinear diffusion filtering

Next we discuss a modification of the semi-implicit scheme

$$\mathbf{u}^{k+1} = \Big(\mathbf{I} - \tau \sum_{l=1}^{m} \mathbf{A}_l(\mathbf{u}^k)\Big)^{-1} \mathbf{u}^k \tag{38}$$

which leads to a further speed improvement. It takes advantage from two observations that we made in the last section:

1. It is computationally easier to solve linear equations stemming from 1-D diffusions than those from higher-dimensional diffusion processes.

2. For accuracy reasons, diffusion filtering often uses not too large time step sizes τ.

Now the idea is to replace (38) by the *additive operator splitting (AOS) scheme* [40]

$$\mathbf{u}^{k+1} = \frac{1}{m} \sum_{l=1}^{m} \Big(\mathbf{I} - m\tau \mathbf{A}_l(\mathbf{u}^k)\Big)^{-1} \mathbf{u}^k. \tag{39}$$

Such a scheme has several interesting properties:
The operators $\mathbf{I} - m\tau \mathbf{A}_l(\mathbf{u}^k)$ describe one-dimensional diffusion processes along the x_l axes. Under a consecutive pixel numbering along the direction l they come down to strictly diagonally dominant tridiagonal matrices. The corresponding linear systems of equations can be solved directly in an efficient and stable way by the Thomas algorithm. Its forward elimination and backward substitution step can be regarded as a causal and an anticausal recursive filter, respectively.

Moreover, (39) has the same first-order Taylor expansion in τ as the semi-implicit scheme: although both methods are algebraically different, they are $O(\tau + h_1^2 + ... + h_m^2)$ approximations to the continuous equation. Thus, the approximation quality improves with decreasing τ, and the solutions produced by the AOS scheme and the semi-implicit one become more and more similar.

Since AOS is an *additive* splitting, we are assured that all coordinate axes are treated in exactly the same manner. This is in contrast to the finite element discretization from Section 3.1.2 and to conventional splitting techniques from the literature, which are *multiplicative* [11]. They may produce different results if the image is rotated by 90 degrees.

Recently a general framework for discrete nonlinear diffusion scale-spaces has been established, which guarantees that the discretization reveals the same scale-space properties as its continuous counterpart [37, 39]. One can verify [40] that the AOS scheme creates such a discrete nonlinear diffusion scale-space for every step size τ. As a consequence, it preserves the average grey level μ, satisfies a causality property in terms of a maximum–minimum principle, and converges to a constant steady state. Moreover, the process is a simplifying, information-reducing transform with respect to many aspects: The p-norms

$$\|\mathbf{u}^k\|_p := (\sum_{i=1}^{N} |u_i^k|^p)^{1/p} \qquad (p \geq 2) \tag{40}$$

and all even central moments

$$M_{2n}[\mathbf{u}^k] := \frac{1}{N} \sum_{j=1}^{N} (u_j^k - \mu)^{2n} \tag{41}$$

are decreasing in k, and the entropy

$$S[\mathbf{u}^k] := -\sum_{j=1}^{N} u_j^k \, \ln u_j^k, \tag{42}$$

a measure of uncertainty and missing information, is increasing in k (if f_j is positive for all j).

For further algorithmic details and a performance analysis of AOS schemes we refer to [40]. There it is demonstrated that, under typical accuracy requirements, AOS schemes are one magnitude more efficient than the commonly used schemes for nonlinear diffusion filtering. In Section 4 we will see that speed increase by another order of magnitude is posssible by parallel implementations.

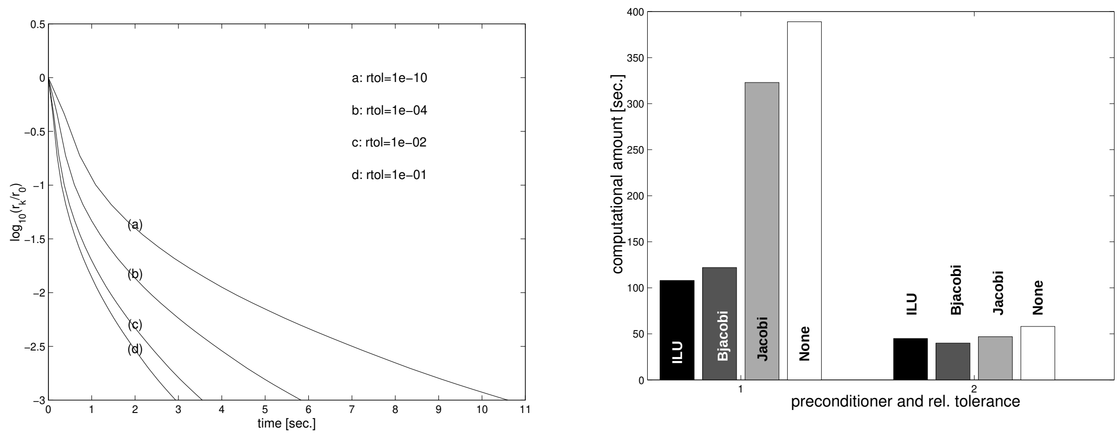

Figure 6: **Left:** Convergence of the variational method for various accuracies $rtol$ of the linear solver for a 256×256 pixel sized image (computed with 16 processing units). **Right:** Computational costs of different preconditioners for exact (left: $rtol = 10^{-6}$) and inexact linear solvers (right: $rtol = 10^{-1}$) (computed with 1 processing unit).

4 Parallel implementations

4.1 Variational approaches

To solve the linear system (25) successively as part of the two–step iteration (17)–(18), we implemented an inexact version of the well–known *conjugate gradient (CG)* method along with several preconditioners on a multi–processor SGI Power–Challenge machine at the Computer Center of the University of Hamburg. We used the software package PETSc (Portable Extensible Toolkit for Scientific Computing) [4, 5], which is based on the message passing standard MPI [26].

Among the class of iterative Krylov subspace solvers (cf., e.g., [23]), the CG–method is nearly optimal for the class of matrices $\mathbf{B}^k$ considered here (sparse, symmetric, positive definite). "Inexact" refers to the stopping criterion $rtol \cdot \|\mathbf{r}^k\|_2$, that is comparing the relative reduction of the initial *linear* residuals $\mathbf{r}^k = \mathbf{B}^k\mathbf{u}^k - \mathbf{b}$ to a threshold. It turned out in our experiments that the rather crude criterion $rtol = 0.1$ leads to a fast inner loop of the iteration (17)–(18) (Fig. 6, left). The stopping criterion of the overall iteration (17)–(18) was $rtol = 0.001$ with respect to the *nonlinear* residuals of the system (24). We note that convergence of the Kačanoc method using an "inexact" inner loop has not been proven yet. For hints how this may be accomplished we refer to the work of Axelson [2].

To improve the condition number and, in turn, the speed of convergence, preconditioners $\mathbf{L}, \mathbf{R}$ were applied to the linear system (25):

$$(\mathbf{L}^{-1}\mathbf{B}^k\mathbf{R}^{-1})\mathbf{R}\mathbf{u}^{k+1} = \mathbf{L}^{-\mathbf{1}}\mathbf{b}$$

Classical preconditioners are obtained through either an additive matrix–splitting (Jacobi or S(S)OR–preconditioning) or a multiplicative matrix–splitting (ILU or ICC

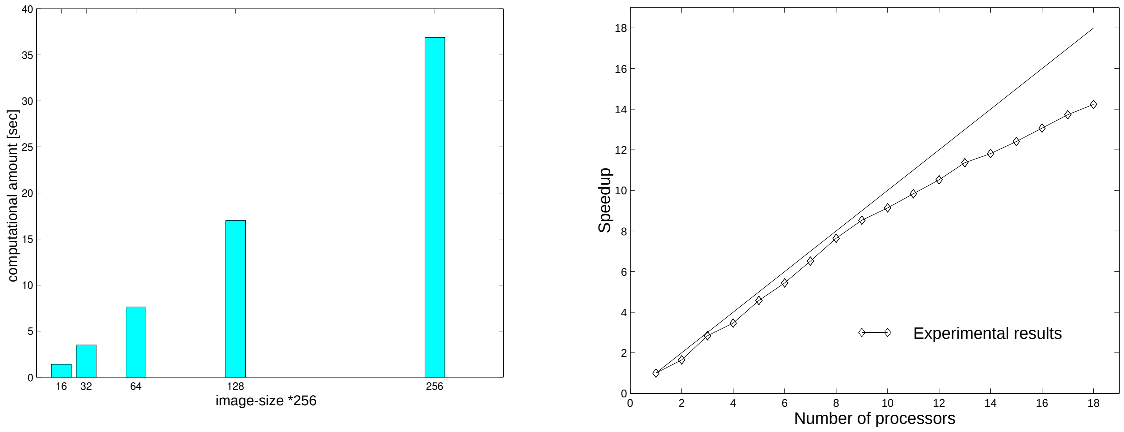

Figure 7: **Left:** Computational costs as a function of the image size (1 processing unit, $rtol = 0.1$, Block–Jacobi preconditioning). **Right:** Speed–up factor as a function of the number of processing units used.

factorizations). The approach of *domain decomposition* is an alternative way of preconditioning (Block–Jacobi or Block–Gauss–Seidel, for example), which is more suited for parallel implementations [36]. Figure 6, right, shows the influence of various preconditioners on the convergence speed for both exact and inexact linear solvers. The less expensive Block–Jacobi preconditioner combined with the inexact solver turned out to be best and reduced the computational amount by one order of magnitude. Figure 7, left, depicts the dependency of the computation time on the image size. Fig. 7, right, finally shows a nearly optimal linear increase of the speed–up factor as a function of the number of processing units (18 units were available to us). This proves the efficiency of the parallel implementation.

These results show that the combination of appropriate problem–specific numerical concepts (Kacanov linearization, inexact linear CG solver, Block–Jacobi preconditioning) with an efficient parallel implementation yields a reduction of the overall computational costs of two orders of magnitude. Using more massive parallelism a further reduction of one order of magnitude should be feasible, enabling the processing of 2D images in quasi–realtime (2–3 frames per second). Furthermore, application of the AOS scheme to the variational approach, based on the relationship to nonlinear diffusion filters described in sections 2.2.1 and 3.2.1, will be considered in future work.

4.2 Nonlinear diffusion filtering

For nonlinear diffusion filtering, the AOS schemes described in Section 3.2.3 work efficiently on serial computer architectures. However, they also offer two intrinsic levels of parallelism: *coarse grain parallelism* and *mid grain parallelism.*

The coarse grain parallelism can be described as follows. The result $\mathbf{u}^k$ of an m-dimensional AOS scheme can be regarded as the average of m filters of type

$$\mathbf{v_l}^{k+1} := \Big(I - m\tau \mathbf{A_l}(\mathbf{u}^k)\Big)^{-1} \mathbf{u}^k \qquad (l = 1, ..., m).$$

Since $\mathbf{v_1}^{k+1}, \ldots, \mathbf{v_m}^{k+1}$ can be calculated independently from each other, it is possible to distribute their computation to different processors of a parallel machine.

Often parallel computers reveal more processors than the dimensionality m of the problem. In this case it is interesting to exploit the mid grain parallelism which is described next. Let N_j denote the number of pixels in the j direction. Then $(\mathbf{I} - m\tau\mathbf{A}_l(u^k))^{-1}$ creates $\prod_{j \neq l} N_j$ one-dimensional diffusion processes along the l direction. These processes are completely independent from each other and can be computed in parallel.

As a demonstrator for a mid grain parallelization we have implemented a three-dimensional AOS scheme on an SGI Power Challenge XL with eight 195 MHz R10000 processors with 1 MByte secondary cache [41]. The test image is a 3-D ultrasound data set of size $138 \times 208 \times 138$ which depicts a 10-weeks old human fetus.

We used the AOS scheme for the regularized nonlinear diffusion filter from Section 2.2.2. The contrast parameter ν was set to the 40% quantile of the cumulative histogram for $\nabla(K_\sigma * f)$, and the standard deviation σ of the Gaussian was chosen as the unit length of the cubic voxel. For our test image 8 iterations with $\tau = 10$ were sufficient to give satisfactory denoising, as is seen in Fig. 8. Note the significantly improved visibility of the skull and the hands.

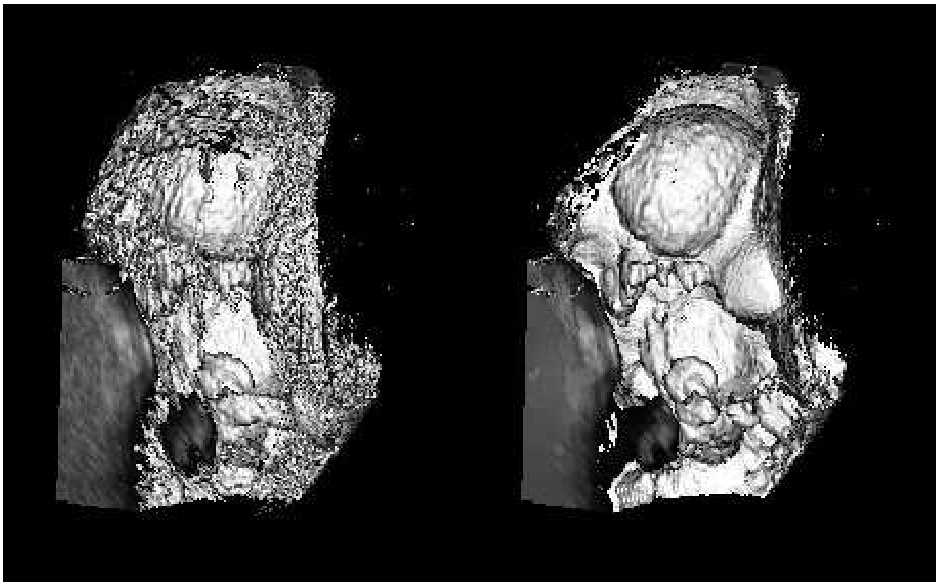

Figure 8: Rendering of a 3-D ultrasound image of a 10-week old fetus. LEFT: Original. RIGHT: Filtered.

Implementing the parallel AOS scheme was done in two steps: First we have further optimized our existing serial AOS code by using the C++-based dedicated TULIP library[1]. This feature has led to a speed-up by a factor of three. In a second step the

[1] TULIP has been developed by Karel Zuiderveld and Fred Appelman (Image Sciences Institute, Utrecht) using cache memory addressing in a sophisticated way in order to speed up the way of accessing neighbouring voxels.

Table 1: Execution time for eight AOS iterations on an SGI Power Challenge for an image of size $138 \times 208 \times 138$ using a varying number of processors.

# proc.	exec. time	speed-up	efficiency
1	158.22 s	1.00	100.0 %
2	93.11 s	1.70	85.0 %
3	71.48 s	2.21	73.7 %
4	59.75 s	2.64	66.0 %
5	51.96 s	3.05	61.0 %

optimized code has been split up into functions acting only within 2-D slices of the 3-D image. These slices were then distributed to the different processors.

Table 1 shows the measured execution times as a function of the number of processors. We observe that an implementation of AOS schemes on a few processors is already sufficient for filtering 3-D medical data sets within their typical acquisition time.

The execution time of an algorithm on a parallel systems is given by

$$T = T_s + \frac{T_p}{p} + T_{sc}(p), \tag{43}$$

where T_s and T_p are the serial and parallel components, respectively, and $T_{sc}(p)$ is the synchronization time for the parallel processes.

If we neglect $T_{sc}(p)$, we find from Table 1 that $T_s \approx 0.16\,T$, that means about 84 % of the code is parallel. The serial part is mainly caused by I/O from and to the harddisk, and from the need to create copies of the volume data. For more iterations we may expect further improvement with respect to the scaling behaviour. The measured data indicate that due to parallel implementations, an efficiency increase by one order of magnitude is realistic.

5 Conclusion and further work

We have described recent advances in the development of efficient numerical algorithms for both nonlinear variational approaches and nonlinear diffusion approaches to adaptive image processing. The performance of parallel implementations of these algorithms on general–purpose hardware has been reported. A mathematically clear connection between both continuous and discrete formulations of variational approaches and nonlinear diffusion filters has been presented. Our work provides a basis for unifying efficient parallel implementations of these approaches and thus can be considered as an important step towards many challenging real-life applications of these theoretically well-founded methods.

Our further work will include the investigation of numerical multigrid schemes and their connections to the algorithms described in the present paper, as well as corresponding implementations on more massive parallel architectures.

Acknowledgements

This research has been partly funded by the German National Science Foundation (DFG), the Real–World Computing Partnership, the EU–TMR project VIRGO, the Fonds zur Förderung der Wissenschaftlichen Forschung (Austria), SFB F1310 (Austria), and the Fonds zur Förderung der Gewerblichen Forschung (FFF), project 200354 (Austria).

References

[1] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axioms and fundamental equations of image processing. *Arch. Rat. Mech. Anal.*, 123:199–257, 1993.

[2] O. Axelson, *Globally convergent continuation methods for non–linear equations*, 1st Workshop on Large–Scale Scientific Computations, Int. House of Scientists "F. Curie", Bulgaria, 1997

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

[4] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith, "PETSc home page," http://www.mcs.anl.gov/petsc, 1998.

[5] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith, "PETSc 2.0 users manual," Tech. Rep. ANL-95/11 - Revision 2.0.22, Argonne National Laboratory, 1998.

[6] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. of Comp. Vision*, 19(1):57–91, 1996.

[7] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[8] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(4):394–398, 1997.

[9] V. Caselles, J.M. Morel, G. Sapiro, and A. Tannenbaum, editors. *IEEE Trans. Image Proc.: Special Issue on Partial Differential Equations and Geometry–Driven Diffusion in Image Processing*, 7(3), 1998.

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

[11] G.I. Marchuk, *Splitting and alternating direction methods*, P.G. Ciarlet, J.-L. Lions (Eds.), Handbook of numerical analysis, Vol. I, 197–462, 1990.

[12] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge–preserving regularization in computed imaging. *IEEE Trans. in Image Proc.*, 6(2):298–311, 1997.

[13] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland Publ. Comp., Amsterdam, 1978.

[14] S. Fučik, A. Kratochvil, and J. Nečas. Kačanov-Galerkin method. *Comment. Math. Univ. Carolinae*, 14(4):651–659, 1973.

[15] D. Geiger and A. Yuille. A common framework for image segmentation. *Int. J. of Comp. Vision*, 6(3):227–243, 1991.

[16] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(3):367–383, 1992.

[17] D. Geman and C. Yang. Nonlinear image recovery with halfquadratic regularization. *IEEE Trans. Image Proc.*, 4(7):932–946, 1995.

[18] Bart M. ter Haar Romeny, editor. *Geometry-Driven Diffusion in Computer Vision*, Dordrecht, The Netherlands, 1994. Kluwer Academic Publishers.

[19] B. ter Haar Romeny, L. Florack, J. Koederink, and M. Viergever, editors. *Scale–Space Theory in Computer Vision*, volume 1252 of *Lect. Not. Comp. Sci.*, Berlin, 1997. Springer.

[20] J. Heers, C. Schnörr, and H.-S. Stiehl. Investigation of parallel and globally convergent iterative schemes for nonlinear variational image smoothing and segmentation. In *Proc. IEEE Int. Conf. Image Proc.*, Chicago, Oct. 4–7, 1998.

[21] J. Heers, C. Schnörr, and H.S. Stiehl. Investigating a class of iterative schemes and their parallel implementation for nonlinear variational image smoothing and segmentation. Technical report 283/99, Comp. Sci. Dept., AB KOGS, University of Hamburg, Germany, Jan. 1999.

[22] J. Kačúr, J. Nečas, J. Polák, J. Souček. Convergence of a method for solving the magnetostatic filed in nonlinear media. *Aplikace Mathematiky*, 13:456–465, 1968.

[23] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995

[24] S.Z. Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, Tokyo, 1995.

[25] R. Malladi, S.A. Sethian, and B.C. Vemuri. A fast level set based algorithm for topology-independent shape modeling. *J. of Math. Imag. Vision*, 6(2/3):269–289, 1996.

[26] Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard*, University of Tennessee, Knoxville, Tennessee, 1995.

[27] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhäuser, Boston, 1995.

[28] M. Pelillo and E.R. Hancock, editors. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 1223 of *Lect. Not. Comp. Sci.*, 1997.

[29] P. Perona, J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12:629–639, 1990.

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

[31] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *Int. J. of Comp. Vision*, 11(1):25–44, 1993.

[32] G. Sapiro and A. Tannenbaum. Area and length preserving geometric invariant scale-spaces. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(1):67–72, 1995.

[33] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. Technical Report DIKU-98/23, Dept. of Comp. Science, Univ. of Copenhagen, 1998.

[34] C. Schnörr. A study of a convex variational diffusion approach for image segmentation and feature extraction. *J. of Math. Imag. and Vision*, 8(3):271–292, 1998.

[35] C. Schnörr. Variational methods for adaptive image smoothing and segmentation. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook on Computer Vision and Applications: Signal Processing and Pattern Recognition*, volume 2, San Diego, 1999. Academic Press. In press.

[36] B. Smith, P. Bjorstad, W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for the Solution of Elliptic Partial Differential Equations*, Cambridge Univ. Press, 1997

[37] J. Weickert, *Nonlinear diffusion scale-spaces: From the continuous to the discrete setting*, M.-O. Berger, R. Deriche, I. Herlin, J. Jaffré, J.-M. Morel (Eds.), ICAOS '96: Images, wavelets and PDEs, Lecture Notes in Control and Information Sciences, Vol. 219, Springer, London, 111–118, 1996.

[38] J. Weickert. A review of nonlinear diffusion filtering. In ter Haar Romeny et al. [19], pages 3–28.

[39] J. Weickert. *Anisotropic diffusion in image processing.* Teubner, Stuttgart, 1998.

[40] J. Weickert, B. ter Haar Romeny, and M.A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Proc.*, 7(3):398–410, 1998.

[41] J. Weickert, K.J. Zuiderveld, B.M. ter Haar Romeny, and W.J. Niessen. Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering. In *Proc. IEEE Int. Conf. Image Processing (ICIP–97)*, Vol. 3, pages 396–399, Santa Barbara, Oct. 26–29, 1997.

[42] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Appl. of Mathematics.* Springer-Verlag, Heidelberg, 1995.