Efficient and Reliable Schemes for Nonlinear Diffusion Filtering

Joachim Weickert, Bart M. ter Haar Romeny, Max A. Viergever

Abstract— **Nonlinear diffusion filtering is usually performed with explicit schemes. They are only stable for very small time steps, which leads to poor efficiency and limits their practical use. Based on a recent discrete nonlinear diffusion scale-space framework we present semi-implicit schemes which are stable for all time steps. These novel schemes use an additive operator splitting (AOS) which guarantees equal treatment of all coordinate axes. They can be implemented easily in arbitrary dimensions, have good rotational invariance and reveal a computational complexity and memory requirement which is linear in the number of pixels. Examples demonstrate that, under typical accuracy requirements, AOS schemes are at least ten times more efficient than the widely-used explicit schemes.**

Keywords— **Nonlinear diffusion, recursive filters, absolute stability.**

I. Introduction

Impressive results are the main reason for using nonlinear diffusion filtering in image processing: unlike linear diffusion filtering (which is equivalent to convolving with a Gaussian) edges remain well-localized and can even be enhanced. Spatial regularizations of this filter class have a solid mathematical foundation as well-posed scale-spaces [12], [44], [46], whose parameter influence is well-understood [6], [25].

Poor efficiency is the main reason for *not* using nonlinear diffusion filtering: most approaches are based on the simplest finite difference discretization by means of a so-called explicit or Euler-forward scheme. This scheme requires very small time steps in order to be stable. Hence the whole filtering procedure is rather time-consuming.

In the present paper we are going to address this problem. We present a novel type of separable schemes which do not suffer from any time step size restriction since all stability-relevant terms are discretized in an implicit manner. The backbone of these schemes is a Gaussian algorithm for solving a tridiagonal system of linear equations. It is fast, stable and requires only a few lines programming work. Its forward and backward substitution step can be regarded as a causal and anticausal filter of a recursive scheme. The presented algorithms are applicable in arbitrary dimensions and their computational and storage effort is linear in the image size. This shows their *efficiency*.

We prove the *reliability* of these schemes by verifying that they satisfy recently established criteria for nonlinear diffusion scale-spaces [45], [46]. This comes down to checking six simple criteria. If these requirements are fulfilled we can be sure that the scheme preserves the average grey value, satisfies a causality property in terms of a maximum–minimum–principle, reveals a large class of smoothing Lyapunov functionals, and converges to a constant steady-state as the time tends to infinity. It should be noted that the discrete maximum–minimum principle is a very restrictive stability criterion (more restrictive than the von Neumann stability), since it also takes into account the boundary conditions and guarantees that over- and undershoots cannot appear.

The goal of this paper is to guide the reader in a systematic way to these so-called additive operator splitting (AOS) schemes. Specific knowledge in numerical analysis is not necessary, as we shall refer to the required material in the literature whenever it is needed. However, the reader who is interested in a more detailed introduction to the matrix algebra which is useful for the present paper, may find this in Chapter 6 of Ortega's textbook [35]. As a prototype of a well-founded nonlinear diffusion filter we focus on a spatial regularization of the Perona–Malik filter [37] by Catté, Lions, Morel and Coll [12], and Whitaker and Pizer [50].

The paper is organized as follows:

Section II gives a brief survey on this diffusion model (which will call CLMC equation henceforth). In Section III we review the simplest scheme for the 1-D CLMC equation: the explicit (Euler forward) discretization in time. We analyze it by means of criteria for discrete nonlinear diffusion scale-spaces in order to explain why it requires rather prohibitive time step sizes. As a remedy we study a semi-implicit discretization for which we show that it satisfies all discrete scale-space criteria (including stability) even for arbitrary large time steps. It requires to solve a tridiagonal linear system of equations, which is easily and efficiently done by a special variant of the well-known Gaussian elimination algorithm. This so-called *Thomas algorithm* will be presented in detail, since it forms the core of the whole scheme.

In Section IV we consider the higher-dimensional case. It is argued that the simple explicit scheme leads to even more restrictive stability conditions than in the 1-D case, while the semi-implicit scheme remains absolutely stable. However, solving the m-dimensional linear system becomes

This work was done while all authors were with the Image Sciences Institute, Utrecht University Hospital, Heidelberglaan 100, E01.334, 3584 CX Utrecht, The Netherlands. It was sponsored by the Real World Computing Partnership. The corresponding author Joachim Weickert is now with the Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark. E-mail: joachim@diku.dk.

significantly less efficient for dimensions ≥ 2.

As as remedy we present an alternative scheme which is also semi-implicit, has the same approximation order, and is absolutely stable, but it can be separated into one-dimensional processes. Thus, the simple and efficient Thomas algorithm can be applied again. Unlike classical multiplicative splitting schemes from the mathematical literature, we consider an additive operator splitting (AOS). It ensures that all coordinate axes are treated equally, a very desirable symmetry property in the context of image processing. Furthermore, we shall check that the AOS schemes satisfy all criteria for discrete nonlinear scale-spaces.

The section is concluded by proposing a related method for the regularization step within the CLMC model. Since this regularization is based on a Gaussian convolution, it is natural to regard it as a linear diffusion filter for which one may also apply splitting techniques based on the Thomas algorithm.

Section V presents an m-dimensional algorithmic formulation of the AOS schemes and analyses its complexity.

In Section VI we evaluate the results by checking the performance of AOS schemes with respect to rotational invariance and accuracy. This allows us to propose reasonable time step size and to analyse the accuracy and efficiency in comparison to the unsplit semi-implicit scheme and the widely-used explicit scheme.

We conclude the paper with a summary in Section VII. A shortened preliminary version of this paper can be found in [47].

Related work. Our work has been influenced by a number of related approaches which we would like to mention here.

Implicit splitting-based approaches for linear diffusion filtering have been proposed in [20], [9], and also in [2], [3], [52] where their realization as recursive filters is suggested. Impressive results on improved efficiency by means of recursive filtering can be found in [14], [15], and the close relation between recursive filters and linear scale-space approaches has been clarified in [32]. Semidiscrete or fully discrete analogues of linear diffusion filtering have been proposed in [34], [26], [4], [38].

In the nonlinear diffusion field one can find several approaches which aim to be efficient alternatives to the conventional two-level explicit finite-difference scheme, for instance three-level methods [18], semi-implicit approaches [12], multiplicative splittings [43], multigrid methods [1], finite element techniques with adaptive mesh coarsening [5], numerical schemes with wavelets as trial functions [18], and pseudospectral methods [18]. Even hardware proposals for nonlinear diffusion filtering can be found in the literature [36], [19].

Schemes which inherit a large number of the properties of their continuous counterparts have also been proposed in the context of curvature-based nonlinear scale-spaces [8], [10], [11], [13]. Sophisticated algorithms for such processes comprise fast level set methods [40], high-order ENO schemes [41], and implicit algorithms for mean curvature motion [2], [31].

II. The continuous filter process

In the m-dimensional case the filter of Catté, Lions, Morel and Coll [12] has the following structure:

Let $\Omega := (0, a_1) \times .... \times (0, a_m)$ be our image domain and consider a (scalar) image as a bounded mapping from Ω into the real numbers $\mathbb{R}$. Then the CLMC filter calculates a filtered image $u(x,t)$ of $f(x)$ as a solution of the diffusion equation

$$\partial_t u \;=\; \mathrm{div}\left(g(|\nabla u_\sigma|^2)\,\nabla u\right) \tag{1}$$

with the original image as initial state,

$$u(x,0) \;=\; f(x), \tag{2}$$

and reflecting boundary conditions:

$$\partial_n u := 0 \quad \text{on} \quad \partial\Omega, \tag{3}$$

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

The "time" t is a scale parameter: increasing t leads to simpler image representations. The whole embedding of the original image into such a one-parameter family of simplified images is called *scale-space*. The first representative of this very general and useful image processing concept, namely linear diffusion filtering, has been derived in an axiomatic way by Taizo Iijima more than 35 years ago [23], [48].

In order to reduce smoothing at edges, the diffusivity g is chosen as a decreasing function of the edge detector $|\nabla u_\sigma|$. Here, ∇u_σ is the gradient of a smoothed version of u which is obtained by convolving u with a Gaussian of standard deviation σ:

$$\nabla u_\sigma \;:=\; \nabla(K_\sigma * u), \tag{4}$$

$$K_\sigma \;:=\; \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{|x|^2}{2\sigma^2}\right). \tag{5}$$

We use the following form for the diffusivity:

$$g(s) \;:=\; \begin{cases} 1 & (s \leq 0) \\ 1 - \exp\left(\frac{-3.315}{(s/\lambda)^4}\right) & (s > 0). \end{cases} \tag{6}$$

For such rapidly decreasing diffusivities smoothing on both sides of an edge is much stronger than smoothing across it. As a result, the gradient at edges may even be enhanced, see [37] for more details. λ plays the role of a contrast parameter: Structures with $|\nabla u_\sigma| > \lambda$ are regarded as edges, where the diffusivity is close to 0, while structures with $|\nabla u_\sigma| < \lambda$ are considered to belong to the interior of a region. Here the diffusivity is close to 1. In this sense, the CLMC model serves as a selective smoothing which prefers intraregional smoothing to interregional blurring. After some time it leads to segmentation-like results which are piecewise almost constant.

The parameter $\sigma > 0$ makes the filter insensitive to noise at scales smaller than σ. It is also a regularization

parameter which guarantees well-posedness of the process: Catté *et al.* [12] have shown that their filter has a unique solution which is infinitely times differentiable for $t > 0$. Weickert [44], [46] has proved that it depends continuously on the original image, satisfies a maximum–minimum principle and reveals a large family of smoothing Lyapunov functionals which guarantee that the solution tends to a constant image for $t \to \infty$. During the whole evolution, the average grey value remains unaltered.

Equations of this type have been successfully applied to process medical images, see e.g. [24], [50], [27]. Nevertheless, they are only one representative of a large class of nonlinear scale-spaces. Overviews of other methods can be found in [16], [21], [46].

III. One-dimensional case

A. Explicit scheme

A.1 The scheme

The one-dimensional CLMC equation is given by

$$\partial_t u \;=\; \partial_x \Big(g\big(|\partial_x u_\sigma|^2\big)\, \partial_x u \Big). \tag{7}$$

Let us now consider the simplest discrete approximation of this process. A discrete image can be regarded as a vector $f \in \mathbb{R}^N$, whose components f_i, $i \in J := \{1, ..., N\}$ display the grey values at each pixel. Pixel i represents some location x_i, and h is the grid size. We consider discrete times $t_k := k\tau$, where $k \in \mathbb{N}_0$ and τ is the time step size. By u_i^k we denote approximations to $u(x_i, t_k)$.

The simplest discretization of (7) with reflecting boundary conditions is given by

$$\frac{u_i^{k+1} - u_i^k}{\tau} \;=\; \sum_{j \in \mathcal{N}(i)} \frac{g_j^k + g_i^k}{2h^2}\,(u_j^k - u_i^k), \tag{8}$$

where $\mathcal{N}(i)$ is the set of the two neighbours of pixel i (boundary pixels have only one neighbour).

The diffusivities g_i^k approximate $g(|\nabla u(x_i, t_k)|^2)$. They can be obtained as follows:

In the spatially discrete case the convolution $u_\sigma = K_\sigma * u$ comes down to a multiplication of $u \in \mathbb{R}^N$ with a suitable matrix $H \in \mathbb{R}^{N \times N}$. In Section IV-C we shall present an efficient way to achieve this in the spatial domain. A gradient approximation by central differences gives

$$g_i^k \;:=\; g\left(\frac{1}{2} \sum_{p,q \in \mathcal{N}(i)} \left(\frac{u_p^k - u_q^k}{2h} \right)^2 \right) \tag{9}$$

for some inner pixel i. This expression remains also valid at the boundary pixels, if we extend the image by reflecting it at the boundary.

We can write the explicit scheme in matrix–vector notation as

$$\frac{u^{k+1} - u^k}{\tau} \;=\; A(u^k)\, u^k \tag{10}$$

with $A(u^k) = (a_{ij}(u^k))$ and

$$a_{ij}(u^k) \;:=\; \begin{cases} \frac{g_i^k + g_j^k}{2h^2} & (j \in \mathcal{N}(i)), \\ -\sum\limits_{n \in \mathcal{N}(i)} \frac{g_i^k + g_n^k}{2h^2} & (j = i), \\ 0 & (\text{else}). \end{cases} \tag{11}$$

This comes down to the iteration scheme

$$u^{k+1} = \Big(I + \tau A(u^k) \Big)\, u^k, \tag{12}$$

where $I \in \mathbb{R}^N$ is the unit matrix. This scheme is called *explicit*, since u^{k+1} can be directly calculated from u^k without solving a system of equations.

Such an explicit iteration step is computationally very cheap: its requires mainly to calculate the three nonvanishing matrix extries per row and to perform a matrix–vector multiplication. The computational and storage effort is linear in the pixel number N. But does this explicit scheme also create a good discrete scale-space and how far can we come with one step? We can find an answer to these question by applying a framework for discrete nonlinear diffusion scale-spaces which we shall review next.

A.2 Criteria for discrete nonlinear diffusion scale-spaces

Recently a scale-space interpretation for the continuous CLMC equation and its anisotropic generalizations has been established [44], [46]. In addition to invariances such as the preservation of the average grey value, it has been shown that – it spite of its contrast-enhancing potential – these equations create smoothing scale-spaces: the obey a maximum–minimum principle, have a large class of smoothing Lyapunov functionals, and converge to a constant steady-state.

It would be desirable to ensure that discrete approximations do also reveal these qualities *exactly*. Criteria have been identified under which one can guarantee that a discrete scheme of type

$$u^0 \;=\; f, \tag{13}$$

$$u^{k+1} \;=\; Q(u^k)\, u^k, \qquad \forall\, k \in \mathbb{N}_0, \tag{14}$$

possesses such properties [45], [46]. All one has to check are the following criteria for $Q(u^k) = (q_{ij}(u^k))$:

(D1) *Continuity in its argument:*

$$Q \in \mathrm{C}(\mathbb{R}^N, \mathbb{R}^{N \times N}) \tag{15}$$

(D2) *Symmetry:*

$$q_{ij} = q_{ji} \qquad \forall\, i, j \in J. \tag{16}$$

(D3) *Unit row sum:*

$$\sum_{j \in J} q_{ij} = 1 \qquad \forall\, i \in J. \tag{17}$$

(D4) *Nonnegativity:*

$$q_{ij} \geq 0 \qquad \forall\, i, j \in J. \tag{18}$$

(D5) *Positive diagonal:*

$$q_{ii} > 0 \qquad \forall\, i \in J. \tag{19}$$

(D6) *Irreducibility:*
We can connect any two pixels by a path with nonvanishing diffusivities. Formally:
For any $i, j \in J$ there exist $k_0,...,k_r \in J$ with $k_0 = i$ and $k_r = j$ such that $q_{k_p k_{p+1}} \neq 0$ for $p = 0,...,r-1$.

Under these prerequisites the filtering process is well-posed and satisfies the following discrete scale-space properties [45], [46]:

(a) Average grey level invariance:
The average grey level $\mu := \frac{1}{N}\sum_{j\in J} f_j$ is not affected by the discrete diffusion filter:

$$\frac{1}{N}\sum_{j\in J} u_j^k = \mu \qquad \forall\, k \in \mathbb{N}_0. \tag{20}$$

This invariance is required in scale-space based segmentation algorithms such as the hyperstack [33].

(b) Extremum principle:

$$\min_{j\in J} f_j \leq u_i^k \leq \max_{j\in J} f_j \qquad \forall\, i \in J, \quad \forall\, k \in \mathbb{N}_0. \tag{21}$$

This property is much more than a stability result which forbids under- and overshoots. It also ensures that iso-intensity linking towards the original image is possible. Hence, it states an important causality property, cf. [22].

(c) Smoothing Lyapunov sequences:
The process is a simplifying, information-reducing transform with respect to many aspects:

(i) The p-norms

$$\|u^k\|_p \;:=\; \Big(\sum_{i=1}^{N} |u_i^k|^p\Big)^{1/p} \tag{22}$$

are decreasing in k for all $p \geq 1$.

(ii) All even central moments

$$M_{2n}[u^k] \;:=\; \frac{1}{N}\sum_{j=1}^{N}(u_j^k - \mu)^{2n} \quad (n \in \mathbb{N}) \tag{23}$$

are decreasing in k.

(iii) The entropy

$$S[u^k] \;:=\; -\sum_{j=1}^{N} u_j^k \ln u_j^k, \tag{24}$$

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

(d) Convergence to a constant steady-state:

$$\lim_{k\to\infty} u_i^k = \mu \quad \forall\, i \in J. \tag{25}$$

Thus, the discrete scale-space evolution tends to the most global image representation that is possible: a constant image with the same average grey level as f.

A.3 Does the explicit scheme create a discrete scale-space?

Let us now investigate if the explicit scheme (12) satisfies the criteria (D1)–(D6) for discrete nonlinear scale-spaces. Let

$$Q(u^k) \;:=\; (q_{ij}(u^k)) := I + \tau A(u^k). \tag{26}$$

By virtue of (11) we observe that the continuity of Q with respect to its argument follows directly from the continuity of the diffusivity g.

The symmetry of Q follows from (11) and the symmetry of the neighbourhood relation ($i \in \mathcal{N}(j) \iff j \in \mathcal{N}(i)$).

By the construction of A it is also evident that the row sums of A vanish. Hence, all row sums of Q are 1, which proves (D3).

Thus, let us investigate the nonnegativity. From $a_{ij} \geq 0$ for $i \neq j$, we also have $q_{ij} \geq 0$ for $i \neq j$. Thus, we can focus on the diagonal entries. If they are all positive, both (D4) and (D5) are satisfied. Since

$$q_{ii} = 1 - \tau \sum_{j\neq i} a_{ij} \tag{27}$$

and $\sum_{j\neq i} a_{ij} > 0$, positive diagonal entries require that

$$\tau \;<\; \frac{1}{\max_i \sum_{j\neq i} a_{ij}(u^k)}. \tag{28}$$

In order to show that Q is irreducible, let us assume that τ satisfies this restriction and consider two arbitrary pixels s_1 and s_2. If $s_1 \leq s_2$ then the positivity of g implies that

$$q_{s_1,s_1+1} > 0, \quad q_{s_1+1,s_1+2} > 0, \quad \cdots, \quad q_{s_2-1,s_2} > 0. \tag{29}$$

If $s_1 > s_2$ then

$$q_{s_1,s_1-1} > 0, \quad q_{s_1-1,s_1-2} > 0, \quad \cdots, \quad q_{s_2+1,s_2} > 0. \tag{30}$$

This establishes (D6).

From these considerations we conclude that the explicit scheme creates a discrete scale-space provided that the time step size satisfies the restriction (28). In image processing one usually sets $h := 1$. Since the diffusivity g is bounded from above by 1, definition (11) allows us to guarantee (28) for $\tau < 1/2$.

In practice, this is often a very severe step size restriction. It means that the use of an explicit scheme is limited rather by its stability than its accuracy. For this reason it would be interesting to look for schemes with better stability properties. This shall be done next.

B. Semi-implicit scheme

B.1 The scheme

We consider a slightly more complicated discretization of (7), namely

$$\frac{u^{k+1} - u^k}{\tau} \;=\; A(u^k)\, u^{k+1}, \tag{31}$$

which leads to the scheme

$$\left(I - \tau A(u^k)\right) u^{k+1} = u^k. \tag{32}$$

We observe that this scheme does not give the solution u^{k+1} directly (explicitly): It requires to solve a linear system first. For this reason it is called a *linear-implicit (semi-implicit)* scheme.

Remark. One may also be interested in studying the *(fully) implicit* scheme

$$\frac{u^{k+1} - u^k}{\tau} \;=\; A(u^{k+1})\, u^{k+1}, \tag{33}$$

leading to a nonlinear system of equations. This is more complicated to solve. Below we shall see, however, that such a high effort is not necessary, since already semi-implicitness is sufficient to guarantee absolute stability.

B.2 Does the semi-implicit scheme create a discrete scale-space?

In order to establish the semi-implicit scheme (32) as a discrete scale-space we have to check (D1)–(D6) again.

First we have to show that

$$B(u^k) \;:=\; (b_{ij}(u^k)) \;:=\; (I - \tau A(u^k)) \tag{34}$$

is invertible. This is easily seen, because B is strictly diagonally dominant:

$$|b_{ii}| > \sum_{j \neq i} |b_{ij}| \qquad \forall\, i \in J. \tag{35}$$

It is well-known from linear algebra that strictly diagonally dominant matrices are invertible, see e.g. [35, p. 226]. Thus,

$$Q(u^k) \;:=\; (q_{ij}(u^k)) \;:=\; B^{-1}(u^k) \tag{36}$$

exists and the continuity of Q in its argument follows from the continuity of g. Moreover, the symmetry of A carries also over to B and Q, which establishes (D2).

In order to prove (D3), consider $w := (1, ..., 1)^T \in \mathbb{R}^N$. Since B has unit row sum, we have $Bw = w$. This implies that

$$w = B^{-1}w = Qw. \tag{37}$$

Reading this componentwise shows that Q has also unit row sum.

(D4)–(D6) can be verified in one step. We already know that B is strictly diagonally dominant. It is also immediately seen that B is irreducible, $b_{ij} \leq 0$ for $i \neq j$, and $b_{ii} > 0$ for all i. Then a theorem by Varga [42, p. 85] tells us that $Q = B^{-1}$ satisfies

$$q_{ij} > 0 \qquad \forall\, i, j \in J. \tag{38}$$

Thus, Q is nonnegative, has positive diagonal and is irreducible.

From these considerations we observe that the semi-implicit scheme creates a discrete nonlinear diffusion scale-space for arbitrarily large time steps. In particular, it is unconditionally stable and does not suffer from any time step size restriction. Unlike the explicit scheme, it can be fully adapted to the desired accuracy without the need to choose small time steps for stability reasons.

B.3 Solving the tridiagonal linear system: The Thomas algorithm

The semi-implicit scheme requires to solve a linear system, where the system matrix is tridiagonal and diagonally dominant. The most efficient way to do this is the so-called *Thomas algorithm*, a Gaussian elimination algorithm for tridiagonal systems. It can be found in many textbooks on numerical analysis, e.g. [39, pp. 43–45]. However, since it builds the backbone of our algorithms and since we want to keep this paper selfcontained, we survey its algorithmic features here.

The principle is as follows. Suppose we want to solve a tridiagonal linear system $Bu = d$ with

$$B = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{N-2} & \alpha_{N-1} & \beta_{N-1} \\ & & & \gamma_{N-1} & \alpha_N \end{pmatrix}. \tag{39}$$

Then the Thomas algorithm consists of three steps.

Step 1: LR decomposition

We decompose B into the product of a lower bidiagonal matrix

$$L = \begin{pmatrix} 1 & & & \\ l_1 & 1 & & \\ & \ddots & \ddots & \\ & & l_{N-1} & 1 \end{pmatrix} \tag{40}$$

and an upper bidiagonal matrix

$$R = \begin{pmatrix} m_1 & r_1 & & \\ & \ddots & \ddots & \\ & & m_{N-1} & r_{N-1} \\ & & & m_N \end{pmatrix} \tag{41}$$

Comparing the coefficients shows that $r_i = \beta_i$ for all i, and m_i and l_i can be obtained as follows:

$m_1 := \alpha_1$
`for` $i = 1, 2, ..., N-1$:
  $l_i := \gamma_i / m_i$
  $m_{i+1} := \alpha_{i+1} - l_i \beta_i$

Solving $LRu = d$ for u is done in two steps:

Step 2: Forward substitution

We solve $Ly = d$ for y. This gives

$y_1 := d_1$
`for` $i = 2, 3, ..., N$:
  $y_i := d_i - l_{i-1} y_{i-1}$

Step 3: Backward substitution

We solve $Ru = y$ for u. This leads to

$$\boxed{\begin{array}{l} u_N := y_N/m_N \\ \texttt{for } i = N-1, N-2, ..., 1: \\ \quad u_i := (y_i - \beta_i u_{i+1})/m_i \end{array}}$$

This completes the Thomas algorithm. It is stable for every strictly diagonally dominant system matrix. One may also regard it as a recursive filtering: The LR decomposition determines the filter coefficients, Step 2 is a causal filter and Step 3 an anticausal one. The whole scheme is very efficient: it requires only

$$2(N-1) + (N-1) + 1 + 2(N-1) \;=\; 5N - 4 \tag{42}$$

multiplications/divisions, and

$$(N-1) + (N-1) + (N-1) \;=\; 3N-3 \tag{43}$$

subtractions. Hence the CPU effort is *linear* in N. The same is true for the storage effort.

Applying the Thomas algorithm to the semi-implicit scheme takes almost twice as long as one iteration of the explicit scheme, but we may use much larger time steps, since the scheme is absolutely stable.

IV. Higher-dimensional case

A. Explicit and semi-implicit schemes

The m-dimensional CLMC equation is given by

$$\partial_t u \;=\; \sum_{l=1}^{m} \partial_{x_l} \Big(g(|\nabla u_\sigma|^2)\; \partial_{x_l} u \Big). \tag{44}$$

We can discretize the m summands of the right hand side in the same manner as in the 1-D case. Using only one index for pixel numbering, we may represent the whole image of size $N_1 \times ... \times N_m$ as a vector of size $N := N_1 \cdots N_m$. In this vector–matrix notation we can write the m-dimensional explicit scheme as

$$u^{k+1} \;=\; \Big(I + \tau \sum_{l=1}^{m} A_l(u^k) \Big)\, u^k \tag{45}$$

and its semi-implicit counterpart as

$$u^{k+1} \;=\; \Big(I - \tau \sum_{l=1}^{m} A_l(u^k) \Big)^{-1} u^k. \tag{46}$$

In both cases the matrix $A_l = (a_{ijl})_{ij}$ corresponds to derivatives along the l-th coordinate axis. Let us also introduce

$$A(u^k) \;:=\; (a_{ij}(u^k)) \;:=\; \sum_{l=1}^{m} A_l(u^k). \tag{47}$$

What about the reliability of both schemes? Checking the discrete scale-space requirements (D1)–(D6) can be done in a similar way as in the 1-D case, see [46, Theorem 8] for more details. As in the 1-D case one obtains that the explicit scheme creates a discrete scale-space for

$$\tau \;\le\; \frac{1}{\max\limits_i \sum\limits_{j \neq i} a_{ij}(u^k)}, \tag{48}$$

and that the semi-implicit scheme satisfies all requirements unconditionally.

What does this mean regarding efficiency? In the m-dimensional case each inner pixel i_0 has $2m$ neighbours with which it is connected via nonvanishing entries in the i_0-th row of A. From (11) we see that we can estimate

$$\sum_{j \neq i} a_{ij}(u^k) \;\le\; \left(\frac{2}{h_1^2} + ... + \frac{2}{h_m^2} \right) \sup_s g(s) \tag{49}$$

where h_1,...,h_m denote the dimensions of an m-dimensional pixel. With $h_1 = ... = h_m = 1$ and $\sup g = 1$, restriction (48) may be replaced by

$$\tau < \frac{1}{2m}. \tag{50}$$

Thus, the allowed step size of the explicit scheme becomes even smaller for higher dimensions.

However, this does not necessarily imply that the semi-implicit scheme becomes superior. There appears a new problem as well: Although the actual structure of the matrix A depends on the pixel numbering, it is not possible anymore to order the pixels in such a way that in the i-th row all nonvanishing matix elements can be found within the positions $[i, i-m]$ to $[i, i+m]$: Usually, the matrix reveals 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. Hence, iterative algorithms have to be applied. Classical methods like Gauß–Seidel or SOR do not need additional storage and convergence can be guaranteed for the special structure of A. This convergence, however, is rather slow. Faster iterative methods such as the SSOR-CG algorithm [30, pp. 154–161] need significantly more storage, which can become prohibitive for large images. A typical problem of iterative methods is also that their convergence becomes slower for larger τ, since this increases the condition number of the system matrix. Multigrid methods [7] appear to be one possibility to circumvent many of these problems, but their implementation is more complicated.

Recapitulating, we see that for dimensions ≥ 2 the semi-implicit scheme remains absolutely stable, but it is difficult to take full advantage of this because of the problems to solve the arising linear system as efficiently as it was possible in the 1-D case with the Thomas algorithm.

B. AOS schemes

In order to address the abovementioned problem let us consider a modification of the semi-implicit scheme (46), namely the *additive operator splitting (AOS) scheme*

$$u^{k+1} \;=\; \frac{1}{m} \sum_{l=1}^{m} \Big(I - m\tau A_l(u^k) \Big)^{-1} u^k. \tag{51}$$

Several points should be noted:

- The explicit scheme (45), the semi-implicit scheme (46) and the AOS scheme (51) have the same first-order Taylor expansions in τ. It is easy to see that all schemes are $\mathcal{O}(\tau + h_1^2 + ... + h_m^2)$ approximations to the continuous equation. From this viewpoint, all schemes are consistent to the original equation. One should not make the mistake to regard the AOS scheme as an algebraically incorrect reformulation of the semi-implicit scheme: The explicit scheme is also different from the semi-implicit one, but it approximates the same continuous diffusion process.
- The operators

$$B_l(u^k) \;:=\; I - m\tau A_l(u^k) \tag{52}$$

describe one-dimensional diffusion processes along the x_l axes. Thus, under a suitable pixel numbering they come down to strictly diagonally dominant tridiagonal matrices which can be inverted in an efficient and stable way by the Thomas algorithm from Section III-B.3.

- Since it is an *additive* splitting, all coordinate axes are treated in exactly the same manner. This is in contrast to conventional splitting techniques from the literature [17], [28], [30], [51]. They are *multiplicative* splittings such as the *locally one-dimensional (LOD) scheme*

$$u^{k+1} \;=\; \prod_{l=1}^{m} \Big(I - \tau A_l(u^k)\Big)^{-1} u^k. \tag{53}$$

Since in the general nonlinear case the split operators do not commute, the result of multiplicative splittings will depend on the order of the one-dimensional operators. This disadvantage will be discussed in Section VI in more detail.

B.1 Does the AOS scheme create a discrete scale-space?

The discussed properties suggest that the AOS scheme is an interesting candidate for an efficient discrete diffusion scale-space. Thus, let us now assess its reliability by checking the criteria (D1)–(D6).

Many reasonings carry over from the 1-D semi-implicit scheme: First we observe that $Q_l := (q_{ijl})_{ij} := B_l^{-1}$ exist, since B_l is strictly diagonally dominant. Also the continuity of

$$Q := \frac{1}{m} \sum_{l=1}^{m} Q_l \tag{54}$$

in its argument is a direct consequence of the continuous diffusivity g and the construction of A_l.

In the same way the symmetry of Q goes back to the symmetry of A_l. Note that the symmetry of A_l is independent of the pixel numbering: a permutation of their numbering transforms A_l into PA_lP^{-1} for some permutation matrix P. Since $P^{-1} = P^T$ and there exists a pixel numbering such that A_l is transformed into a symmetric tridiagonal matrix just as in the 1-D case, it is clear that A_l has to be symmetric.

With the same reasoning as in III-B.2, we know that not only B_l, but also Q_l has row sum 1. Thus, Q has also unit row sum.

To verify (D4), we observe that $B_l = (b_{ijl})_{ij}$ is strictly diagonally dominant, $b_{iil} > 0$ for all i, and $b_{ijl} \leq 0$ for $i \neq j$. Under these circumstances we may conclude from [29, p. 192] that $Q_l = B_l^{-1}$ is nonnegative in all components. This implies the nonnegativity of Q.

Let us now check (D5) and (D6) in one step. Since B_l, $l = 1, ..., m$ represent one-dimensional diffusion operators, it follows that there exist permutation matrices P_l, $l = 1, ..., m$ such that $P_l B_l P_l^T$ is not only diagonally dominant, but also tridiagonal and block irreducible[1]. Within each irreducible matrix block, we have a positive diagonal and nonnegative off-diagonals. Applying again Varga's theorem [42, p. 85], we conclude that the inverse of each block contains only positive elements. From this it follows that $a_{i_0 j_0 l} \neq 0$ for some $i_0, j_0 \in J$ implies that $q_{i_0 j_0 l} > 0$. Thus, the irreducibility of $A = \sum_{l=1}^{m} A_l$ carries over to $Q = \frac{1}{m} \sum_{l=1}^{m} Q_l$, and (D6) is satisfied. In particular, since A_l is constructed such that $a_{iil} \neq 0$ for all $i \in J$, it is clear that Q contains only positive diagonal elements. Therefore, (D5) is verified as well.

These discussions show that the AOS scheme creates a discrete nonlinear diffusion scale-space for all time step sizes.

C. Regularization

This section describes a simple method for calculating the presmoothing $u_\sigma = K_\sigma * u$ in a way which is consistent with the ideas presented above.

It is well-known that Gaussian convolution with standard deviation σ is equivalent to linear diffusion filtering ($g \equiv 1$) for some time $T = \sigma^2/2$. Thus we may use the (semi-)implicit[2] scheme again in order to obtain a stable algorithm. Several things make the situation even easier than in the nonlinear setting:

- Frequently, σ is in the order of the pixel size. In this case we may regularize in a single step by filtering once with a time step size $T = \sigma^2/2$.
- The linear diffusion process is separable. Therefore, the order of the one-dimensional approximations is not of importance and we may also use a multiplicative splitting:

$$Pu^k := \prod_{l=1}^{m} S_l u^k. \tag{55}$$

- The system in step l can be decomposed into N/N_l tridiagonal systems with the same system matrix. Thus, the LR decomposition needs to be done only

[1] Each of the N/N_l blocks represents the pixels where all components except for the $l-$th are identical.

[2] Semi-implicit and implicit are identical in the linear case.

once for an $N_l \times N_l$-matrix of type

$$\begin{pmatrix} 1+r_l & -r_l & & & \\ -r_l & 1+2r_l & -r_l & & \\ & \ddots & \ddots & \ddots & \\ & & -r_l & 1+2r_l & -r_l \\ & & & -r_l & 1+r_l \end{pmatrix} \tag{56}$$

with $r_l := \tau / h_l^2$. Therefore, the main effort boils down to performing N/N_l times the same forward and backward substitution step from the Thomas algorithm. This requires only $3N-2$ multiplications/divisions and $2N-2$ subtractions. Such an effort is comparable with the recursive filters presented in [3], [2], [52], but unlike those Fourier-based methods, the algorithm presented here allows an adequate treatment of the reflecting boundary conditions and preserves the average grey value.

V. Algorithmic structure

A. AOS algorithm

We may summarize our considerations in the following algorithm for one AOS step in m dimensions.

```
input:   u = u^n

regularization:   v := K_σ * u
(according to IV-C)

calculate diffusivity g(|∇v|^2)
(approximate ∇v by central differences)
(use look-up table for evaluating g)

create copy:       f := u
initialize sum:    u := 0

for l = 1, ..., m:
    calculate v := (mI − m^2 τ A_l)^{-1} f:
    (solve N/N_l tridiagonal systems of
    size N_l with Thomas algorithm III-B.3)
    update:   u := u + v

output:   u = u^{n+1}
```

B. Complexity

In order to assess the complexity of AOS algorithms, let us consider dimensions $m \geq 2$ and focus on terms of order N.

From the preceding algorithm we recognize that only the four vectors u, v, g, and f are required. Thus, since all calculations may be performed in single precision, the main storage effort is $4N \times 4$ Bytes. This is independent of the dimension m.

Table I summarizes the relevant computational requirements for each step of the AOS algorithm. We observe that the effort is proportional to the number N of pixels and the dimension m. The total effort is only $11mN$ multiplications or divisions, $(10m-1)N$ additions or subtractions, and N look-ups in a table. This is less than twice the typical effort needed for an explicit scheme, a rather low price for gaining absolute stability.

TABLE I
Main operations for one m-dimensional AOS step (M/D: multiplications or divisions; A/S: additions or subtractions; LUT: look-up operations in a table).

task	M/D	A/S	LUT
regularization	3mN	2mN	
calculate $\|\nabla v\|^2$	2mN	(2m-1)N	
calculate diffusivity			N
create system matrix	mN	3mN	
Thomas algorithms	5mN	3mN	
total	11mN	(10m-1)N	N

VI. Evaluation

We have seen that AOS schemes with large time steps still reveal average grey value invariance, stability based on extremum principle, Lyapunov functionals, and convergence to a constant steady-state. Thus, they are legitimate when being considered as a pure discrete process which is not intended to approximate a continuous process.

But does this mean that it is recommendable to consider arbitrarily large time step sizes? In the extreme case: can one filter an image in one step?

In this case we should expect problems with those properties which a naturally linked to continuous ideas and which can only be satisfied approximately by discrete schemes: rotational invariance and accuracy.

A loss of rotational invariance becomes visible as a preference of certain directions, while a loss of accuracy becomes evident in those cases where filtering with time step $n\tau$ differs visually from n times filtering with τ. So let us now check these approximation effects by applying a 2-D AOS scheme to two test images.

First we check the rotational invariance. Since the AOS scheme is consistent to the original equation, we should expect good rotational invariance for small spatial and temporal steps.

Figure 1 is used as a test for rotational invariance. It depicts a Gaussian-like image and its filtered versions. For $\tau = 0.25$ both the explicit and AOS scheme are visually undistinguishable. This step size is also the stability limit for the explicit scheme, while the AOS scheme allows to increase τ further. We see that for $\tau \leq 5$ no significant changes appear. Thus, AOS may be used with 20 times larger time steps than the explicit scheme. On the other hand, even for $\tau = 20$ the deviations from a perfect circular structure are not very severe.

What about the accuracy? Figure 2 depicts the filtering of a brain image. The situation is similar as in Figure 1: For $\tau = 0.25$ the explicit and the AOS scheme are undistinguishable. The AOS scheme remains close to these results up to $\tau \approx 5$. For $\tau = 20$ we get more severe deviations: the filtering effect becomes weaker. This is a typical behaviour for implicit schemes with large time steps: implicit techniques always remain on the "safe" side (by orienting

the diffusion on the "smoother" future rather than on the "rougher" past), while their deviation from the true solution becomes larger with increasing step size. Thus, their filtering effect on the final image at a specified time decreases with increasing time step size. Again $\tau = 5$ is a good compromise between efficiency and accuracy.

After these visual inspections we shall investigate the accuracy more quantitatively. To this end we perform a comparison between the explicit scheme (45), the semi-implicit scheme (46) and the AOS scheme (51). Since no analytical solution to the CLMC equation is known, we have to use a good numerical approximation to a test example as a standard for comparison. In our case we took the explicit scheme with the small step size $\tau = 0.1$ and applied it 2000 times to the test image from Fig. 2.

The linear system of the 2-D semi-implicit scheme is solved by a *Gauß–Seidel algorithm*. Iterative methods of this type are quite popular for nonlinear PDEs in image processing [2], [31], since they are easy to implement and they do not require additional memory. Let the diffusion operator

$$B(u^k) \;:=\; I - \tau \sum_{l=1}^{m} A_l(u^k) \tag{57}$$

be decomposed into the strictly lower triangular matrix L, the diagonal matrix D, and the strictly upper triangular matrix U. Then the Gauß–Seidel method approximates the solution u^{k+1} of the semi-implicit scheme

$$B(u^k)\, u^{k+1} \;=\; u^k \tag{58}$$

by a sequence of vectors $y^{(n)}$ with

$$y^{(0)} \;=\; u^k, \tag{59}$$

$$y^{(n+1)} \;=\; (L + D)^{-1}\,(u^k - U y^{(n)}) \qquad (n \geq 0). \tag{60}$$

Every second step we calculate the residue

$$r^{(n)} := B y^{(n)} - u^k, \tag{61}$$

and we stop the iteration process if its l^2 norm $\|r^{(n)}\|_2 = (\sum_i |r_i^{(n)}|^2)^{1/2}$ satisfies

$$\|r^{(n)}\|_2 \;\leq\; \alpha \|r^{(0)}\| \tag{62}$$

with some accuracy parameter $\alpha = 0.01$ or $\alpha = 0.1$.

Table II compares the explicit scheme, the semi-implicit scheme with accuracies $\alpha = 0.01$ and $\alpha = 0.1$, respectively, and the AOS scheme.

If v denotes our reference solution (explicit scheme, $\tau = 0.1$), then we calculate the relative l^2 error of an approximation u as

$$\frac{\|u - v\|_2}{\|v\|_2}. \tag{63}$$

First we observe that the explicit scheme with $\tau = 0.25$ reveals a very small error, while the semi-implicit method with $\alpha = 0.01$ is not only less accurate, but also slower for $\tau = 0.25$ and 0.5. For $\tau \geq 1$ the semi-implicit scheme becomes faster than the explicit one. On the other hand,

TABLE II
COMPARISON OF NONLINEAR DIFFUSION SCHEMES.

scheme	τ	CPU time	rel. l^2 error
explicit	0.25	65.65 s	0.14 %
semi-implicit	0.25	145.65 s	0.27 %
($\alpha = 0.01$)	0.5	92.62 s	0.68 %
	1	62.97 s	1.05 %
	2	43.58 s	1.46 %
	5	27.62 s	1.97 %
	10	19.49 s	2.26 %
	20	12.45 s	2.74 %
	50	5.54 s	3.77 %
semi-implicit	0.25	106.43 s	0.83 %
($\alpha = 0.1$)	0.5	72.04 s	1.00 %
	1	36.70 s	1.40 %
	2	23.53 s	1.83 %
	5	12.30 s	2.51 %
	10	7.00 s	3.27 %
	20	3.53 s	4.19 %
	50	1.17 s	5.43 %
AOS	0.25	114.04 s	0.73 %
	0.5	56.55 s	1.32 %
	1	28.56 s	1.66 %
	2	14.25 s	1.83 %
	5	5.80 s	2.22 %
	10	2.95 s	2.73 %
	20	1.52 s	3.37 %
	50	0.67 s	4.29 %

the Gauß-Seidel algorithm slows down for larger τ, since this increases the condition number of the system matrix. Hence, the overall CPU time per semi-implicit step increases with increasing τ. If we relax the accuracy from $\alpha = 0.01$ to $\alpha = 0.1$, the semi-implicit scheme becomes faster, but the l^2 error also increases. For $\tau > 0.5$, the AOS scheme becomes the fastest method. Interestingly, for $\tau > 2$, it is also more accurate than the semi-implicit scheme with $\alpha = 0.1$.

It is worth noticing that there is a fundamental difference between errors in the AOS scheme and errors which are introduced by an insufficient number of Gauß–Seidel iterations: unlike AOS errors which are compatible with the discrete scale-space framework, Gauß–Seidel errors can violate these requirements. Thus, properties such as the average grey level invariance are no more satisfied in an exact manner. In order to avoid these difficulties, one would have to apply more Gauß–Seidel iterations, which will finally destroy all efficiency advantages compared to the explicit scheme; see also [31].

Figure 3 gives a graphical representation of Table II, which allows us to find the most efficient schemes for a desired accuracy. We observe that for very high accuracy requirements the explicit scheme is most appropriate[3]. This

[3]One can achieve even higher accuracy by methods which are of

is at the expense of a hight overall computational effort. On the other hand, even relaxing the accuracy requirements to a relative l^2 error of 1 % does not permit us to find a more efficient technique. For errors between 1 % and 1.7 % the semi-implicit scheme with $\alpha = 0.1$ is fastest, and for errors larger than 1.7 % AOS schemes become rapidly superior. In our previous experiments we have observed that the accuracy of AOS with $\tau = 5$ appears to be tolerable for many applications. This corresponds to an error of about 2.2 %. In this case, AOS is almost 2.5 times more efficient than the semi-implicit scheme with $\alpha = 0.1$, more than 3.5 times faster than the semi-implicit scheme with $\alpha = 0.01$, and about 11 times more efficient than the explicit scheme. Although these relations have been illustrated by one example only, additional experiments have indicated that these basic relations between explicit, semi-implicit and AOS discretizations carry over to a large class of images: The accuracy requirements of many practical problems allow an efficiency gain by one order of magnitude. All one has to do is to replace the explicit scheme by an AOS scheme with 20 times larger time step sizes.

It should be noted that the AOS schemes calculate the average of operators of type $(I - m\tau A_l)^{-1}$. They describe 1-D diffusions with a step size $m\tau$. Since multiplicative splittings such as the LOD scheme (53) use operators of type $(I-\tau A_l)^{-1}$, one can expect that they give even better accuracy. However, multiplicative splittings for nonlinear problems reveal one big disadvantage, which makes their use in many image processing applications problematic: In the general nonlinear case the split operators do not commute any longer. Thus, the result of multiplicative splittings depends on the order of the one-dimensional operators, and the grid axes are treated differently. In practice, this means that these schemes produce different results if the image is rotated by 90 degrees. Such an undesirable effect is illustrated in Fig. 4. Since AOS schemes apply the 1-D operators in parallel instead of sequentially, they do not suffer from this limitation.

Moreover, most multiplicative splittings lead to a nonsymmetric system matrix $Q(u^k)$. This violates criterion (D2) for discrete diffusion scale-spaces. For this reason, we have not considered these approaches in the present paper.

Finally we check the relation between the computational effort and the number of pixels. Table III shows the measured CPU times on a single R10000 processor of an SGI Challenge XL and on an HP 900-755, both for 2-D and 3-D images.

For small image sizes the computing times reveal good proportionality to the overall number of pixels. This is what we expect from theory. Because of cache limitations, the CPU time per pixel becomes slightly higher for larger data sets: We also observe that this deviation from the linear scaling behaviour is machine dependent. The HP remains closer to the linear scaling behaviour than the SGI. On the other hand, with its CPU memory of 1 GByte the SGI permits even to process data sets of size 8192×8192 and $512 \times 512 \times 256$.

second order in time, for instance predictor–corrector techniques [46]. Such a high accuracy, however, is rarely required in image processing.

TABLE III
MEASURED CPU TIMES FOR ONE AOS ITERATION.

image size	SGI	HP
64×64	0.0086 s	0.0168 s
128×128	0.0324 s	0.0676 s
256×256	0.134 s	0.307 s
512×512	0.711 s	1.63 s
1024×1024	6.35 s	7.38 s
2048×2048	27.8 s	34.0 s
4096×4096	145 s	
8192×8192	724 s	
$16 \times 16 \times 16$	0.0159 s	0.0331 s
$32 \times 32 \times 32$	0.116 s	0.304 s
$64 \times 64 \times 64$	1.15 s	2.45 s
$128 \times 128 \times 128$	13.4 s	20.1 s
$256 \times 256 \times 256$	237 s	
$512 \times 512 \times 256$	1340 s	

Three-dimensional data sets from medicine with typical sizes such as $256 \times 256 \times 64$ can be processed in less than one minute per AOS iteration (both on the HP and the SGI). In many practical applications less than 10 iterations are sufficient for the denoising of such data sets.

Recapping we have observed that – although the desired approximation quality is of course purpose dependent – under typical circumstances 20 times larger step sizes than the stability limit of the explicit scheme appear reasonable. They give an efficiency gain of a factor ten[4]. Especially for large data sets such as 3-D medical data this is often the difference between not applicable and applicable. We are currently testing our schemes for the filtering of 3-D ultrasound images and preprocessing 3-D MR data for segmentation. In both cases first results are encouraging.

VII. CONCLUSIONS

We have presented absolutely stable additive operator splitting (AOS) schemes for the nonlinear diffusion filter of Catté *et al.* and Whitaker and Pizer. These schemes satisfy all criteria for discrete nonlinear diffusion scale-spaces and are easy to implement in any dimension. Both computational and storage effort is linear in the number of pixels. Experiments have shown that under realistic accuracy requirements one can gain an increase of efficiency by a factor 10. This makes this type of schemes attractive for applications such as medical 3D data sets.

Implementations of AOS schemes on parallel architectures are studied in [49]. These experiments demonstrate that it is possible to gain a speed-up by another order of

[4]We have seen that an m-dimensional AOS scheme averages 1-D operators with an effective step size of $m\tau$. Thus, for higher dimensions m one should reduce the step size in order to have the same accuracy. However, since explicit schemes also have to decrease the step size for larger m in the same way, the factor 10 remains valid for every dimension.

magnitude by exploiting the intrinsic parallelism of AOS schemes. Last but not least, there are also ways to generalize AOS schemes to anisotropic diffusion filters with diffusion tensors; a first proposal in this direction can be found in [46, Section 4.4.2].

Acknowledgements

We would like to thank Robert Maas for careful proofreading and helpful hints.

References

[1] S.T. Acton, A.C. Bovik, M.M. Crawford, *Anisotropic diffusion pyramids for image segmentation*, Proc. IEEE Int. Conf. Image Processing (ICIP–94, Austin, Nov. 13–16, 1994), Vol. 3, IEEE Computer Society Press, Los Alamitos, 478–482, 1994.

[2] L. Alvarez, *Images and PDE's*, 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, 3–14, 1996.

[3] L. Alvarez, L. Mazorra, *Signal and image restoration using shock filters and anisotropic diffusion*, SIAM J. Numer. Anal., Vol. 31, 590–605, 1994.

[4] K. Åström, A. Heyden, *Stochastic analysis of image acquisition and scale-space smoothing*, J. Sporring, M. Nielsen, L. Florack, P. Johansen (Eds.), Gaussian scale-space theory, Kluwer, Dordrecht, 129–136, 1997.

[5] E. Bänsch, K. Mikula, *A coarsening finite element strategy in image selective smoothing*, Preprint No. 18/1996, Faculty of Mathematics, University of Freiburg, Hebelstr. 27, 79104 Freiburg, Germany, 1996; to appear in Computation and Visualization in Science.

[6] B. Benhamouda, *Parameter adaptation for nonlinear diffusion in image processing*, master thesis, Dept. of Mathematics, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany, 1994.

[7] W.L. Briggs, *A multigrid tutorial*, SIAM, Philadelphia, 1987.

[8] A.M. Bruckstein, G. Sapiro, D. Shaked, *Evolution of planar polygons*, Int. J. Pattern Recogn. Artif. Intell., Vol. 9, 991–1014, 1995.

[9] L.D. Cai, *Some notes on repeated averaging smoothing*, J. Kittler (Ed.), Pattern recognition, Lecture Notes in Comp. Science, Vol. 301, Springer, Berlin, 597–605, 1988.

[10] F. Catté, *Convergence of iterated affine and morphological filters by nonlinear semi-group theory*, 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, 125–133, 1996.

[11] F. Catté, F. Dibos, G. Koepfler, *A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets*, SIAM J. Numer. Anal., Vol. 32, 1895–1909, 1995.

[12] 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.

[13] T. Cohignac, F. Eve, F. Guichard, C. Lopez, J.-M. Morel, *Numerical analysis of the fundamental equation of image processing*, Preprint No. 9254, CEREMADE, Université Paris IX – Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France, 1992.

[14] R. Deriche, *Fast algorithms for low-level vision*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 12, 78–87, 1990.

[15] R. Deriche, *Recursively implementing the Gaussian and its derivatives*, V. Srinivasan, Ong S.H., Ang Y.H. (Eds.), Proc. Second Int. Singapore Conf. on Image Proc. (Singapore, Sept. 7–11, 1992), 263–267, 1992.

[16] R. Deriche, O. Faugeras, *Les EDP en traitement des images et vision par ordinateur*, Traitement du Signal, Vol. 13, No. 6, Special Issue, 1996.

[17] J. Douglas, J.E. Gunn, *A general formulation of alternating direction methods. Part I. Parabolic and hyperbolic problems*, Numer. Math., Vol. 6, 428–453, 1964.

[18] J. Fröhlich, J. Weickert, *Image processing using a wavelet algorithm for nonlinear diffusion*, Report No. 104, Laboratory of Technomathematics, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany, 1994.

[19] T. Gijbels, P. Six, L. Van Gool, F. Catthoor, H. De Man, A. Oosterlinck, *A VLSI-architecture for parallel non-linear diffusion with applications in vision*, Proc. IEEE Workshop on VLSI Signal Processing, Vol. 7, 398–707, 1994.

[20] A.R. Gourlay, *Implicit convolution*, Image Vision Comput., Vol. 3, 15–23, 1985.

[21] B.M. ter Haar Romeny (Ed.), *Geometry-driven diffusion in computer vision*, Kluwer, Dordrecht, 1994.

[22] R.A. Hummel, *Representations based on zero-crossings in scale space*, Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition (CVPR '86, Miami Beach, June 22–26, 1986), IEEE Computer Society Press, Washington, 204–209, 1986.

[23] T. Iijima, *Basic theory of pattern normalization (for the case of a typical one-dimensional pattern)*, Bulletin of the Electrotechnical Laboratory, Vol. 26, 368–388, 1962 (in Japanese).

[24] C. Lamberti, M. Sitta, F. Sgallari, *Improvements to the anisotropic diffusion model for 2-D echo image processing*, Proc. Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, Vol. 14, 1872–1873, 1992.

[25] X. Li, T. Chen, *Nonlinear diffusion with multiple edginess thresholds*, Pattern Recognition, Vol. 27, 1029–1037, 1994.

[26] T. Lindeberg, *Scale-space for discrete signals*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 12, 234–254, 1990.

[27] D.-S. Luo, M.A. King, S. Glick, *Local geometry variable conductance diffusion for post-reconstruction filtering*, IEEE Trans. Nuclear Sci., Vol. 41, 2800–2806, 1994.

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

[29] T. Meis, U. Marcowitz, *Numerische Behandlung partieller Differentialgleichungen*, Springer, Berlin, 1978.

[30] A.R. Mitchell, D.F. Griffiths, *The finite difference method in partial differential equations*, Wiley, Chichester, 1980.

[31] E. Nicolet, M. Spühler, *Charakterisierung von unscharfen blobförmigen 3-D Strukturen*, diploma thesis, Communication Technology Laboratory, Image Science Division, ETH–Zentrum, 8092 Zürich, Switzerland, 1996.

[32] M. Nielsen, L. Florack, R. Deriche, *Regularization, scale-space and edge detection filters*, J. Math. Imag. Vision, Vol. 7, 291–307, 1997.

[33] W.J. Niessen, K.L. Vincken, J. Weickert, M.A. Viergever, *Nonlinear multiscale representations for image segmentation*, Computer Vision and Image Understanding, Vol. 66, 233–245, 1997.

[34] E. Norman, *A discrete analogue of the Weierstrass transform*, Proc. Amer. Math. Soc., Vol. 11, 596–604, 1960.

[35] J.M. Ortega, *Matrix theory*, Plenum Press, New York, 1987.

[36] P. Perona, J. Malik, *A network for multiscale image segmentation*, Proc. IEEE Int. Symp. Circuits and Systems (ISCAS–88, Espoo, June 7–9, 1988), 2565–2568, 1988.

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

[38] A.H. Salden, B.M. ter Haar Romeny, M.A. Viergever, *Linear scale-space theory from physical principles*, J. Math. Imag. Vision, to appear.

[39] H.R. Schwarz, *Numerische Mathematik*, Teubner, Stuttgart, 1988.

[40] J.A. Sethian, *Level set methods*, Cambridge University Press, Cambridge, 1996.

[41] K. Siddiqi, B.B. Kimia, C.-W. Shu, *Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution*, Graphical Models and Image Processing, Vol. 59, 278–301, 1997.

[42] R.A. Varga, *Matrix iterative analysis*, Prentice Hall, Englewood Cliffs, 1962.

[43] J. Weickert, *Zwischenbericht zum Projekt "Nichtlineare Diffusionsfilter"*, Bericht über die wissenschaftliche Tätigkeit Januar 1991 – Dezember 1991, Center for Applied Mathematics, Darmstadt – Kaiserslautern, 133–142, 1992.

[44] J. Weickert, *Theoretical foundations of anisotropic diffusion in image processing*, Computing, Suppl. 11, 221–236, 1996.

[45] 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.

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

[47] J. Weickert, *Recursive separable schemes for nonlinear diffusion filters*, B. ter Haar Romeny, L. Florack, J. Koenderink, M. Viergever (Eds.), Scale-space theory in computer vision, Lecture Notes in Computer Science, Vol. 1252, Springer, Berlin, 260–271, 1997.

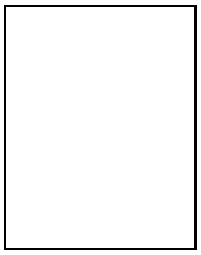
[48] J. Weickert, S. Ishikawa, A. Imiya, *On the history of Gaussian scale-space axiomatics*, J. Sporring, M. Nielsen, L. Florack, P. Johansen (Eds.), Gaussian scale-space theory, Kluwer, Dordrecht, 45–59, 1997.

[49] J. Weickert, K.J. Zuiderveld, B.M. ter Haar Romeny, W.J. Niessen, *Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering*, Proc. 1997 IEEE International Conference on Image Processing (ICIP–97, Santa Barbara, Oct. 26–29, 1997).

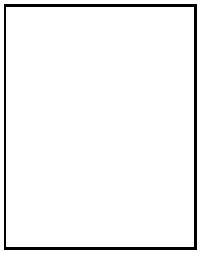
[50] R.T. Whitaker, S.M. Pizer, *A multi-scale approach to nonuniform diffusion*, CVGIP: Image Understanding, Vol. 57, 99–110, 1993.

[51] N.N. Yanenko, *The method of fractional steps: the solution of problems of mathematical physics in several variables*, Springer, New York, 1971.

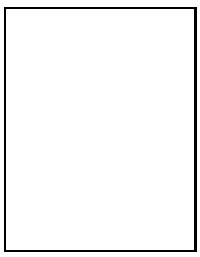
[52] D. Zhao, B. Li, *A new implementation of discrete multiscale filtering*, Proc. IEEE Int. Conf. Image Processing (ICIP–96, Lausanne, Sept. 16–19, 1996), Vol. 1, 383–386, 1996.



Joachim Weickert obtained a M.Sc. in industrial mathematics in 1991 and a Ph.D. in mathematics in 1996, both from Kaiserslautern University, Germany. After receiving the Ph.D. degree, he worked as post-doctoral researcher at the Image Sciences Institute of Utrecht University, The Netherlands. Since April 1997 he is with the computer vision group of the Department of Computer Science at Copenhagen University. His current research interests include all aspects of partial differential equations and scale-space theory in image analysis. His M.Sc. thesis was awarded the Wacker Memorial Prize for the best student thesis of the European Consortium for Mathematics in Industry. Joachim Weickert authored the book "Anisotropic Diffusion in Image Processing" (Teubner Verlag, Stuttgart), and he has given several invited lectures at conferences and international workshops.



Bart M. ter Haar Romeny received a M.Sc. in applied physics from Delft University of Technology in 1978, and Ph.D. from Utrecht University in 1983. After being the principal physicist of the Utrecht University Hospital Radiology Dept. he joined in 1989 the Image Sciences Institute of Utrecht University as an associate professor. His interests are mathematical aspects of front-end vision, in particular linear and non-linear scale-space theory, medical computer vision applications, differential geometry and perception. He authored several papers and bookchapters on these issues, edited a recent book on non-linear diffusion theory in computer vision and is involved in (resp. initiated) a number of international collaborations on these subjects. He is in the editorial board of the Journal of Math. Imaging and Vision, and was chairman of the First Intern. Conf. on Scale-Space Theory in Computer Vision, Utrecht 1997.



Max A. Viergever received the M.Sc. degree in applied mathematics in 1972 and the D.Sc. degree with a thesis on cochlear mechanics in 1980, both from Delft University of Technology. From 1972 to 1988 he was assistant/associate professor of applied mathematics at this university. Since 1988 he is professor and head of the department of Medical Imaging at Utrecht University, and as of 1996 scientific director of the newly established Image Sciences Institute of Utrecht University and the University Hospital Utrecht. He is (co)author of over 200 refereed scientific papers on biophysics and medical image processing, and (co)author/editor of 11 books. His research interests comprise all aspects of computer vision and medical imaging. Max Viergever is a board member of IPMI and IAPR, is editor of the book series Computational Imaging and Vision of Kluwer Academic Publishers, associate editor-in-chief of IEEE Transactions on Medical Imaging, editor of the Journal of Mathematical Imaging and Vision, and participates on the editorial boards of several journals.

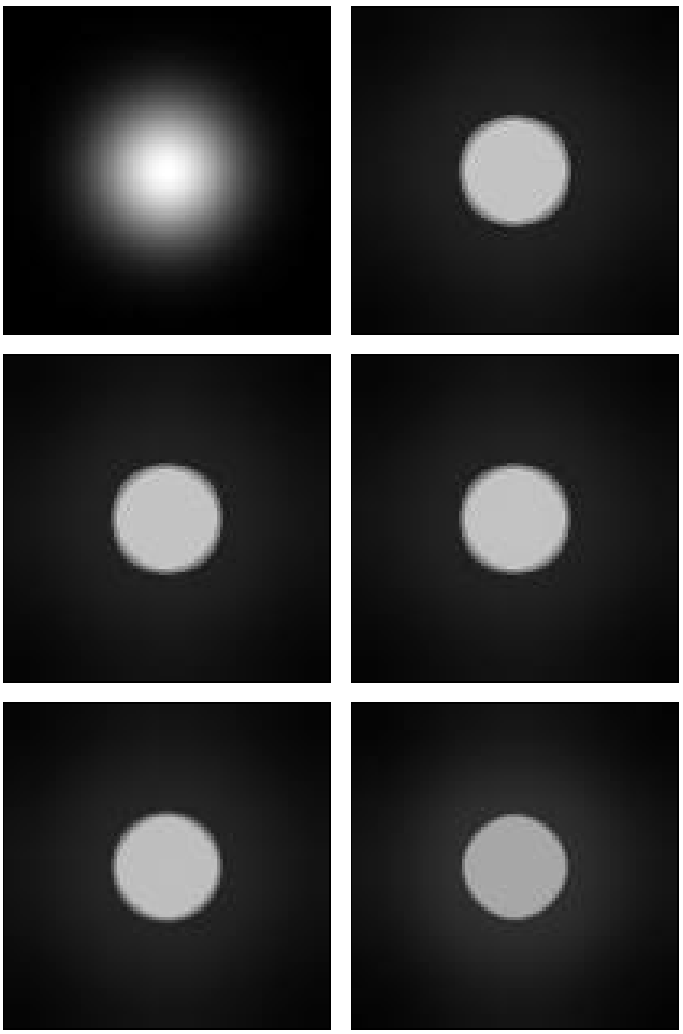


Fig. 1. Nonlinear diffusion filtering of a Gaussian-like test image ($\lambda = 8$, $\sigma = 1.5$). (a) TOP LEFT: Original image, $\Omega = (0, 101)^2$. (b) TOP RIGHT: Explicit scheme, 800 iterations, $\tau = 0.25$. (c) MIDDLE LEFT: AOS scheme, 800 iterations, $\tau = 0.25$. (d) MIDDLE RIGHT: AOS scheme, 200 iterations, $\tau = 1$. (e) BOTTOM LEFT: AOS scheme, 40 iterations, $\tau = 5$. (f) BOTTOM RIGHT: AOS scheme, 10 iterations, $\tau = 20$.

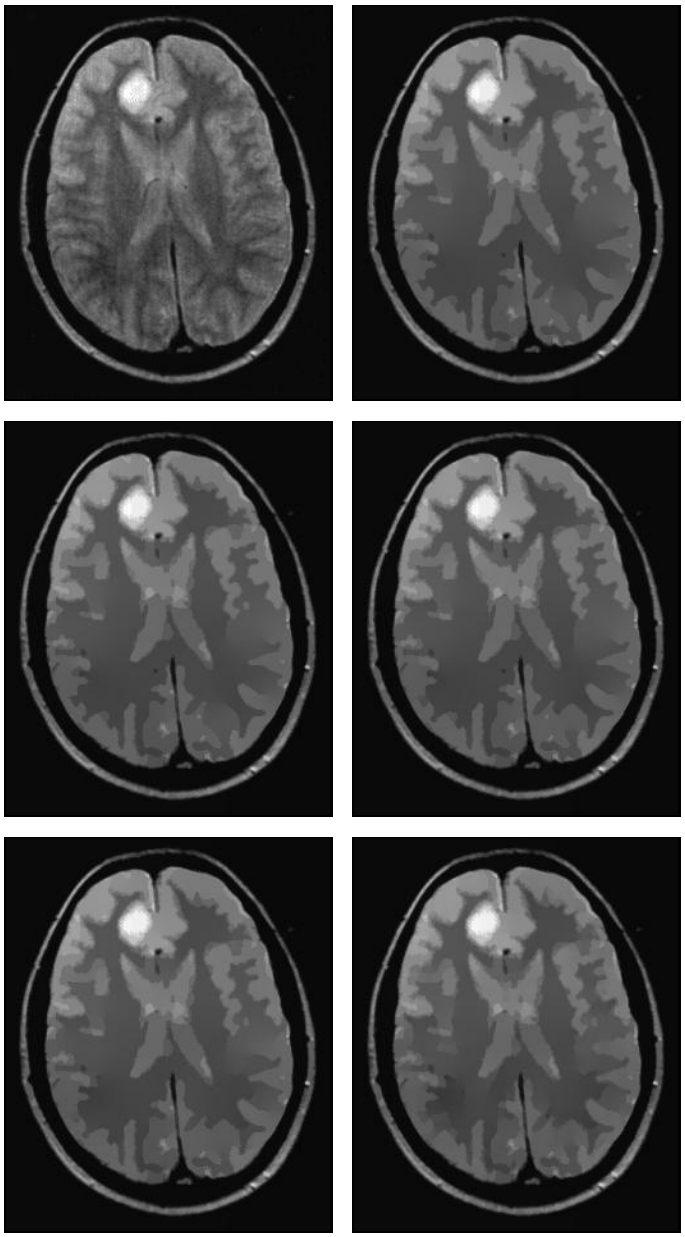


Fig. 2. Nonlinear diffusion filtering of a medical image ($\lambda = 2$, $\sigma = 1$). (a) TOP LEFT: Original image, $\Omega = (0, 255) \times (0, 308)$. (b) TOP RIGHT: Explicit scheme, 800 iterations, $\tau = 0.25$. (c) MIDDLE LEFT: AOS scheme, 800 iterations, $\tau = 0.25$. (d) MIDDLE RIGHT: AOS scheme, 200 iterations, $\tau = 1$. (e) BOTTOM LEFT: AOS scheme, 40 iterations, $\tau = 5$. (f) BOTTOM RIGHT: AOS scheme, 10 iterations, $\tau = 20$.

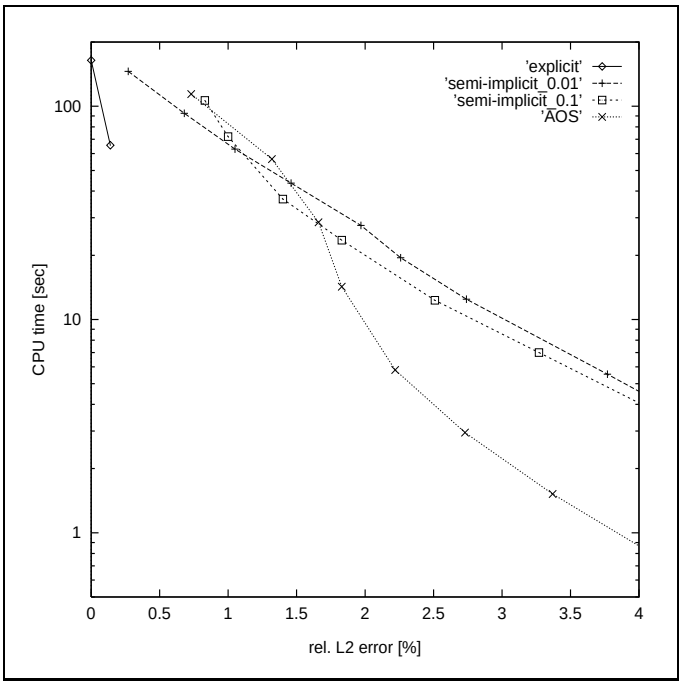


Fig. 3. Tradeoff between efficiency and accuracy of nonlinear diffusion solvers. The data were calculated on the test image from Fig. 2, size $\Omega = (0, 255) \times (0, 308)$. Filter parameters: $\lambda = 2$, $\sigma = 1$. Stopping time: $T = 200$. Hardware: one R10000 processor on an SGI Challenge XL.

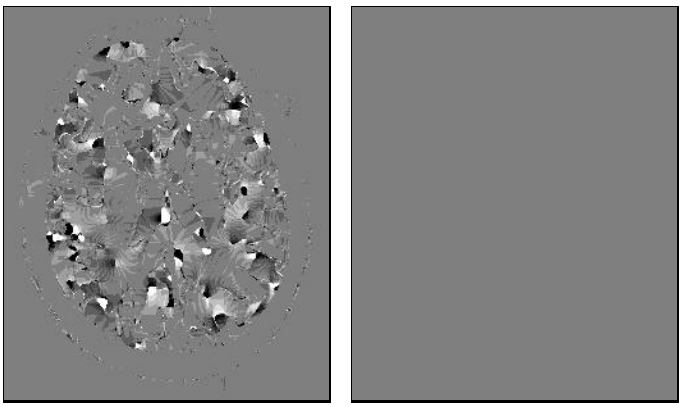


Fig. 4. (Non-)Commutation of nonlinear diffusion operators. The difference between filtering prior to rotation by 90 degrees, and rotation prior to filtering is depicted. Test image: Figure 2 ($\lambda = 2$, $\sigma = 1$, $\tau = 20$, 10 iterations). (a) LEFT: A multiplicative splitting such as LOD treats x and y axes differently. (b) RIGHT: Additive operator splitting (AOS) treats all axes equally.