ADDITIVE OPERATOR SPLITTING

Seminar Numerical Algorithms for Image Analysis

Martin Grochulla

May 16, 2007

Outline

- Diffusion
- Explicit and semi-implicit schemes
- Discrete nonlinear diffusion scale-spaces
- Schemes in higher dimensions
- AOS scheme
- Results and summary

Diffusion

physical process, which

- equilibrates concentration differences
  Fick's law (isotropic case):

$$\underbrace{j}_{flux} = -\underbrace{g}_{diffusivity} \cdot \nabla \underbrace{u}_{concentration}$$

- preserves masses
  continuity equation:

$$\partial_t u = -\mathbf{div}\, j$$

$\Rightarrow$ yields *diffusion equation*

$$\partial_t u = \mathbf{div}\ (g \cdot \nabla u)$$

diffusion equation

$$\partial_t u = \mathbf{div}\ (g \cdot \nabla u)$$

- image domain $\Omega := (0, a_1) \times \ldots \times (0, a_m)$
- image $f(x)$: bounded mapping from $f : \Omega \mapsto \mathbb{R}$
- filtered image $u(x, t)$: solution of the (nonlinear) diffusion equation

$$\partial_t u = \mathbf{div}\ \left(g(|\nabla u_\sigma|^2) \cdot \nabla u\right)$$

with $u(x, 0) = f(x)$ and $\partial_n u := 0$ on $\partial\Omega$ (n denotes the normal to the image boundary $\partial\Omega$)

filtered image computed as solution of nonlinear diffusion equation

$$\partial_t u = \mathbf{div}\ \left(g(|\nabla u_\sigma|^2) \cdot \nabla u \right),$$

(Catté et al.) where

- ∇u_σ: gradient of a Gaussian smoothed version of u:

$$\nabla u_\sigma := \nabla \left(K_\sigma * u \right)$$
$$K_\sigma := \frac{1}{(2 \cdot \pi \cdot \sigma^2)^{\frac{m}{2}}} \cdot \exp\left(-\frac{|x|^2}{2 \cdot \sigma^2} \right)$$

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

in the one-dimensional case the diffusion equation

$$\partial_t u = \textbf{div}\ \left(g(|\nabla u_\sigma|^2) \cdot \nabla u\right)$$

turns into

$$\partial_t u = \partial_x \left(g(|\partial_x u_\sigma|^2) \cdot \partial_x u\right)$$

with a discrete approximation we have a vector $f \in \mathbb{R}^N$ with components $f_i, i \in J = 1, \ldots, N$ as image, where

- pixel i represents some location x_i
- h represents the grid size
- $t_k := k \cdot \tau, k \in \mathbb{N}_0$ are discrete time points
- τ is the time step size

the simplest approximation is given by

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

where

- u_i^k denotes the approximation of $u(x_i, t_k)$
- $\mathcal{N}(i)$ is the neighborhood of pixel i
- g_i^k approximates the term $g(|\nabla u(x_i, t_k)|^2)$:

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

- using matrix-vector notation, we can rewrite the equation from above, yielding

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

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

$$a_{ij}(u^k) := \begin{cases} \frac{g_i^k + g_j^k}{2 \cdot h^2}, & j \in \mathcal{N}(i) \\ -\sum_{n \in \mathcal{N}(i)} \frac{g_i^k + g_n^k}{2 \cdot h^2}, & j = i \\ 0, & j \notin \mathcal{N}(i) \wedge i \neq j \end{cases}$$

- rearranging the terms gives us the (explicit) iteration scheme

$$u^{k+1} = \Big(I + \tau \cdot A(u^k)\Big) \cdot u^k.$$

- in the explicit scheme we have

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k) \cdot u^k$$

- modifying the explicit scheme gives us

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

  which leads to

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

- in order to compute the solution of this equation we have to solve a linear system of equations
- fortunately this system is tridiagonal

- Gaussian elimination algorithm for tridiagonal matrices
- highly efficient (linear complexity)
- easy to implement
- stable for every strictly diagonally dominant system
- given the linear system $B \cdot u = d$ the algorithm computes the solution in three steps:
  1. LR decomposition
  2. Forward substitution
  3. Backward substitution

for discrete scheme type with

$$u^0 = f \qquad \text{and} \qquad u^{k+1} = A(u^k) \cdot u^k \quad \forall k \in \mathbb{N}_0$$

the following criteria have to be fulfilled:

1. continuity of the argument: $A \in C(\mathbb{R}^N, \mathbb{R}^{N \times N})$
2. symmetry: $a_{ij} = a_{ji} \qquad \forall i, j \in J$
3. unit row sum: $\sum_{j \in J} a_{ij} = 1 \qquad \forall i \in J$
4. nonnegativity: $a_{ij} \geqslant 0 \qquad \forall i, j \in J$
5. positive diagonal: $a_{ii} > 0 \qquad \forall i \in J$
6. irreducibility: $\forall i, j \in J \, \exists k_0, \ldots, k_r \in J :$

$$k_0 = i \wedge k_r = j \wedge \forall p = 0, \ldots, r-1 \colon a_{k_p k_{p+1}} \neq 0$$

1. average grey level invariance:

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

   with $\mu := \frac{1}{N} \cdot \sum_{j \in J} f_j$
2. extremum principle:

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

3. smoothing Lyapunov sequences:
   - the p-norms

$$||u^k||_p := \left(\sum_{i=1}^{N} |u_i^k|^p\right)^{1/p}$$

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

3.
- all even central moments

$$M_{2n}[u^k] := \frac{1}{N} \cdot \sum_{j=1}^{N} (u_j^k - \mu)^{2n}, \qquad n \in \mathbb{N}$$

  are decreasing in k
- the entropy

$$S[u^k] := -\sum_{j=1}^{N} u_j^k \cdot \ln u_j^k$$

  is increasing in k (if f_j is positive for all j)

4. convergence to a constant steady-state:

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

- in higher dimensions the diffusion equation is

$$\partial_t u = \sum_{l=1}^{m} \partial_{x_l} \left(g(|\nabla u_\sigma|^2) \cdot \partial_{x_l} u \right).$$

- in matrix-vector notation we get

$$u^{k+1} = \left(I + \tau \cdot \sum_{l=1}^{m} A_l(u^k) \right) \cdot u^k$$

for the explicit scheme

- and

$$u^{k+1} = \left(I - \tau \cdot \sum_{l=1}^{m} A_l(u^k) \right)^{-1} \cdot u^k$$

for the semi-implicit scheme, with $A_l = (a_{ijl})_{ij}$ corresponding to the derivatives along the l-th coordinate axis.

- time step size in explicit schemes gets smaller for higher dimensions
- matrix in semi-explicit scheme not tridiagonal and thus not solvable by Thomas algorithm
- other algorithms for solving the system of equations are rather slow or need significantly more storage

$\Rightarrow$ modifying the semi-implicit scheme

$$u^{k+1} = \left(I - \tau \cdot \sum_{l=1}^{m} A_l(u^k)\right)^{-1} \cdot u^k$$

we get

$$u^{k+1} = \frac{1}{m} \cdot \sum_{l=1}^{m} \left(I - m \cdot \tau \cdot A_l(u^k)\right)^{-1} \cdot u^k$$

- approximates the same continuous diffusion process and has the same approximation order
- creates a discrete scale-space for all step sizes
- the operators

$$B_l(u^k) := I - m \cdot \tau \cdot A_l(u^k)$$

  describe one-dimensional diffusion processes
- in contrast to multiplicative splittings, as for example the locally one-dimensional scheme, all coordinate axes are treated in the same manner

- presmoothing $u_\sigma = K_\sigma * u$
- Gaussian convolution with standard deviation σ is equivalent to linear diffusion filtering for some time $T = \sigma^2/2$
- linear diffusion process is separable, therefore multiplicative splitting can be used
- can be computed efficiently with Thomas algorithm

one AOS step in m dimensions:

`input:` $u = u^n$

- `regularization:` $v := K_\sigma * u$
- `calculate diffusivity` $g(|\nabla v|^2)$
- `create copy:` $f := u$
- `initialize sum:` $u := 0$
- `for` $l = 1, \dots, m$:
  `calculate` $v := (m \cdot \mathrm{I} - m^2 \cdot \tau \cdot A_l)^{-1} \cdot f$:
    `solve` N/N_l `tridiagonal systems of size` N_l
    `with Thomas algorithm`
  `update` $u := u + v$

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

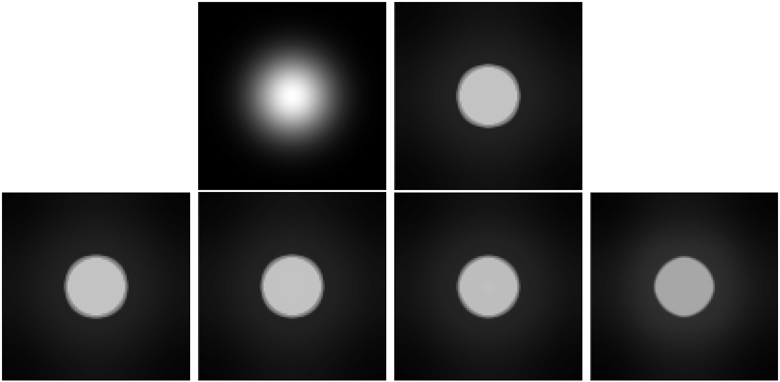

Figure: Nonlinear diffusion filtering of a Gaussian-like test image ($\lambda = 8, \sigma = 1.5$). Top left: Original image, $\Omega = (0, 101)^2$.
Top right: Explicit scheme, 800 iterations, $\tau = 0.25$.
Bottom: AOS scheme, $800/200/40/10$ iterations, $\tau = 0.25/1/5/20$.

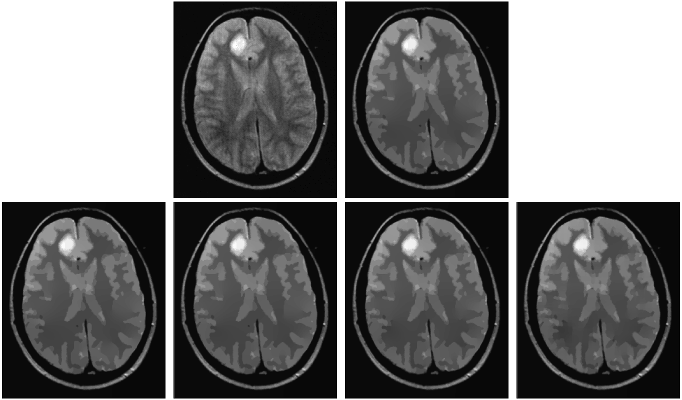

Figure: Nonlinear diffusion filtering of a medical image ($\lambda = 2, \sigma = 1$). Top left: Original image, $\Omega = (0, 255) \times (0, 308)$. Top right: Explicit scheme, 800 iterations, $\tau = 0.25$. Bottom: AOS scheme, $800/200/40/10$ iterations, $\tau = 0.25/1/5/20$.

Summary

additive operator scheme for nonlinear diffusion filter:

- satisfies criteria for nonlinear diffusion scale-spaces
- easy to implement, efficient, fast and unconditionally stable

Literature

Further reading ...

[1] *Efficient and reliable schemes for nonlinear diffusion filtering*, J. Weickert, B. M. ter Haar Romeny, M. A. Viergever, 1998

[2] *Image selective smoothing and edge detection by nonlinear diffusion*, F. Catté, P.-L. Lions, J.-M. Morel, T. Coll, 1992

[3] *A parallel splitting up method and its application to navier-stokes equations*, T. Lu, P. Neittaanmäki, X.-C. Tai, 1991

Thank you

Thank you!

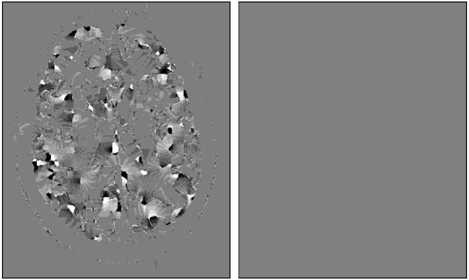

Figure: (Non-)Commutation of nonlinear diffusion operators: difference between filtering prior to rotation by 90 degrees, and rotation prior to filtering. Test image: Brain ($\lambda = 2, \sigma = 1, \tau =$ 20, 10 iterations). Multiplicative splitting (left) treats x- and y-axes differently. Additive operator splitting (right) treats all axes equally.

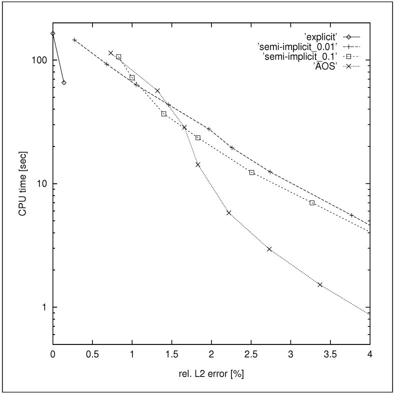

Figure: Tradeoff between efficiency and accuracy of nonlinear diffusion solvers. Test image: Brain ($\lambda = 2, \sigma = 1$, stopping time $T = 200$). Hardware: one R10000 processor on an SGI Challenge XL.