Course 049064: Variational Methods in Image Processing

Notes 2: Diffusion Processes - Part II

Guy Gilboa

1 Stability of PDE's

Some important characteristics of PDE's (partial differential equations).

1.1 Well posedness

A PDE is well posed if we have

1. Existence of a solution.

2. Uniqueness.

3. Continuous dependence on the initial data.

Well posed problems are often related to models of physical problems. Typical examples are the diffusion equation and Laplace equation.

If one of the conditions above is not fulfilled, the PDE is *ill-posed*. A typical example is inverse diffusion.

If the solution is highly perturbed by small changes in the initial conditions, the equation is *ill-conditioned* (can be understood as somewhere between well- and ill-posed).

Perona-Malik, with the given original diffusion coefficients, is ill-posed, there is no uniqueness for the analytic equation. It can be shown there are many possible stair-type steady-state solutions, see [3, 6]. Solutions to this is to convolve the gradient estimation for the diffusion coefficient with a Gaussian $c(|\nabla u * g_\sigma|)$ [1] or implicitly by discrete schemes [5].

1.2 Extremum principle

Also called maximum principle. In general this principle means that the maximum of a function in a domain is on the boundary of the domain. Same for minimum.

In evolutions, we check the behavior of the evolution in time. An evolution which holds an extremum principle means that both extrema are in the initial condition. That is:

$$\max_x u(t;x) \leq \max_x (t=0;x),\ \forall t \in (0,\infty),$$

and for the minimum

$$\min_x u(t;x) \geq \min_x (t=0;x),\ \forall t \in (0,\infty).$$

Forward diffusion equations hold the maximum principle.

1.3 Steady-state solution

The solution at $t \to \infty$ is called the steady state solution, after the system reaches an equilibrium state. For steady states we have

$$u_t = 0.$$

Therefore, for the linear diffusion equation $u_t = \Delta u$ we have

$$\Delta u = 0,$$

which is the Laplace equation. The solution depends on the boundary conditions. In the 1D case, $u_{xx} = 0$ which means the solution is of the form $ax + b$.

Example: Assume we have a function $u(t;x)$ defined on the interval $[0,1]$ with initial conditions $u(t=0;x) = f(x)$, then a steady state solution can be computed analytically, depending on the boundary conditions:

1. For Dirichlet boundary conditions $u(t;x=0) = 0$, $u(t;x=1) = 2$ the steady state is $u(t=\infty;x) = 2x$.

2. For zero Neumann boundary conditions, $u_x(t;x=0) = u_x(t;x=1) = 0$, we have $u(t=\infty;x) = \text{const} = \int_0^1 f(x)dx$.

2 Inverse diffusion

Linear inverse diffusion equation ($c(x) \equiv -1$):

$$u_t = -\Delta u.$$

This is an unstable ill-posed process. By changing the sign of the time differentiation $dt \to -dt$ one can view this also as reversing in time the forward diffusion process. That is going from a blurred signal to a sharp one (not a physical process).

In the linear case with constant diffusion coefficient, one can view this as an extreme high-pass-filter: Reversing the diffusion equation means deconvolving Gaussian convolution. Mathematically, this can be done by inverse filtering.

In Fourier space - spatial Gaussian transforms to a Gaussian $G(f)$ in the frequency domain which decays very fast as frequency grows. The inverse filter is $1/G(f)$ grows exponentially large for high frequencies yielding unstable process (in practice, very sensitive to noise).

2.1 Perona-Malik has inverse diffusion elements

Let us assume the P-M diffusion coefficient $c(s) = \frac{1}{1+(s/k)^2}$. In 1D we thus have

$$u_t = \partial_x \left(\frac{u_x}{1+(u_x/k)^2} \right)$$

which can be written also as

$$u_t = u_{xx} \left(1 - (u_x/k)^2\right)/A,$$

where $A = (1 + (u_x/k)^2)^2 > 0$.

Therefore, the process can be interpreted as two different processes operating on the image simultaneously in different regions: for x such that $|u_x| < k$ we have forward diffusion, whereas for locations where $|u_x| > k$ we have inverse diffusion.

3 Numerical Discretization of PDE's

3.1 Discretized differential operators

There are many ways to discretize PDE's (e.g. finite elements, spectral methods). We will use the more simple *finite difference* schemes which are easy to implement, understand and analyze.

A continuous function $u(t;x)$ is discretized to u_i^n, where $t = n\Delta t$ and $x = ih$, (n and i are usually integers). We assume to have a discrete regular grid with a step size h (in principle a different h can be used for each dimension, we will assume the spatial discretization is the same in all directions). Δt is the time step.

Remark: For images we usually simply assume a unit step $h = 1$ between two adjacent pixels.

3.1.1 Derivative operators

We will use the 1D notations (generalization to any dimension is straightforward). Those are the common first order derivative operators:

Forward difference: $D^+u_i = \frac{u_{i+1}-u_i}{h}$

Backward difference: $D^-u_i = \frac{u_i-u_{i-1}}{h}$

Central difference: $D^0u_i = \frac{u_{i+1}-u_{i-1}}{2h}$

Remark: For smooth functions, the central difference has a second order accuracy (proportional to h^2), whereas the forward and backward operators are first order (proportional to h). However, as images are not smooth and the fact that derivatives of high oscillations are not estimated well in the central scheme, the forward or backward schemes are usually used for our PDE's.

Time: Can be discretized similarly. Usually a backward differnce is used (since we cannot predict the future): $D_t^+u_i^n = \frac{u_i^n-u_i^{n-1}}{\Delta t}$.

Partial derivative: As in the continuous case, the derivative is taken only with respect to one variable. For instance in the two dimensional case, for $u_{i,j}$ which is the discretized version of $u(x,y)$, the forward difference in the x direction is: $D_x^+u_{i,j} = (u_{i+1,j} - u_{i,j})/h$.

Higher order derivatives: For higher order one can use a composition of the first derivatives operators. For instance, the standard second derivative operator (central) is:

$$D^2u_i = D^+(D^-u_i) = \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2}.$$

This can also be seen as twice the central derivative at half pixel resolution: $D_{1/2}^0u_i = \frac{u_{i+1/2}-u_{i-1/2}}{h}$. And therefore $D^2u_i = D_{1/2}^0(D_{1/2}^0u_i)$.

3.2 Evolutions

We use the backward difference for the time evolution: $u_t \approx \frac{u(t+\Delta t)-u(t)}{\Delta t}$. For an evolution of the form $u_t = F(u)$, the general iteration scheme is:

$$u^{n+1} = u + \Delta t F(u).$$

There are several ways to model the flow, depending on the time taken for u on the right hand side.

Explicit scheme: The right side depends solely on the previous iteration,

$$u_i^{n+1} = u_i^n + \Delta t F(u^n).$$

Very convenient to solve but requires small time steps (see CFL condition below).

Implicit scheme: $F()$ on the right side depends on the current iteration,

$$u_i^{n+1} = u_i^n + \Delta t F(u^{n+1}).$$

A system of linear equations are needed to solve this. Harder, good for large time steps.

Semi-implicit scheme: A combination of the two above schemes. For instance Weickert gives the following $\alpha-$ semi-implicit model (with $\alpha \in [0,1]$) [4] p. 102:

$$u_i^{n+1} = u_i^n + \Delta t A(u^n)(\alpha u^n + (1-\alpha)u^n),$$

where A depends nonlinearly on u. This enjoys the unconditional time-step stability afforded by implicit schemes without the need to solve large systems of equations.

3.3 CFL condition

For explicit diffusion schemes (linear and nonlinear) with 4-neighbor connectivity, the time step is bounded by:

$$\Delta t \leq \frac{0.25h^2}{\max_x c(x)}$$

This is a consequence of the CFL condition, which is a general criterion for evolution of PDE's, called after Courant, Friedrichs and Lewy [2].

For example, in a standard implementation of Perona-Malik where $h = 1$, $c() \leq 1$ we have $\Delta t \leq 0.25$.

In the more general case, with M neighbors connectivity, p order PDE we have $\Delta t \leq \frac{h^p}{M \max_x c(x)}$.

References

[1] F. Catte, P. L. Lions, J. M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Num. Anal.*, 29(1):182–193, 1992.

[2] Richard Courant, Kurt Friedrichs, and Hans Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[3] Satyanad Kichenassamy. The perona–malik paradox. *SIAM Journal on Applied Mathematics*, 57(5):1328–1342, 1997.

[4] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner-Verlag, Stuttgart, Germany, 1998.

[5] Joachim Weickert and Brahim Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the peronamalik paradox. In *Advances in Computer Vision*, pages 1–10. Springer, 1997.

[6] Kewei Zhang. Existence of infinitely many solutions for the one-dimensional perona-malik model. *Calculus of Variations and Partial Differential Equations*, 26(2):171–199, 2006.