Lecture 4:

Nonlinear Isotropic Diffusion Filtering I: Modelling and Continuous Theory

Contents

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

The Perona–Malik Filter (1990)

Idea:

- avoid delocatisation and blurring of edges
- intraregional diffusion preferred to interregional smoothing
- consider isotropic nonlinear diffusion and introduce feedback into the process: adapt diffusivity g to gradient of evolving image $u(x, t)$

Nonlinear Diffusion Equation:

$$\partial_t u = \operatorname{div}\left(g(|\nabla u|^2)\, \nabla u\right)$$

- $|\nabla u|^2$ is a "fuzzy edge detector"
- diffusivity g decreasing in $|\nabla u|^2$, e.g. $g(|\nabla u|^2) := \frac{1}{1+|\nabla u|^2/\lambda^2}$.

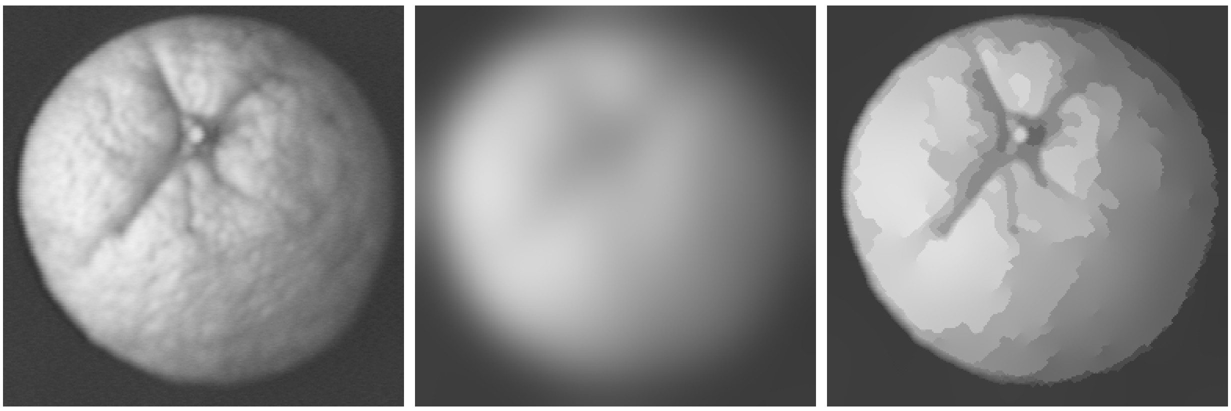
Experiments show that edges remain well-localised and may even be sharpened (!!)

The Perona–Malik Filter (2)

(a) **Left:** "View in Venice", painted by Canaletto, around 1740, National Gallery of Art, Washington, $\Omega = (0, 510) \times (0, 353)$. (b) **Right:** Perona–Malik diffusion with $\lambda = 5$ and $t = 500$. **Author:** J. Weickert, adapted from Perona/Malik (1987).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

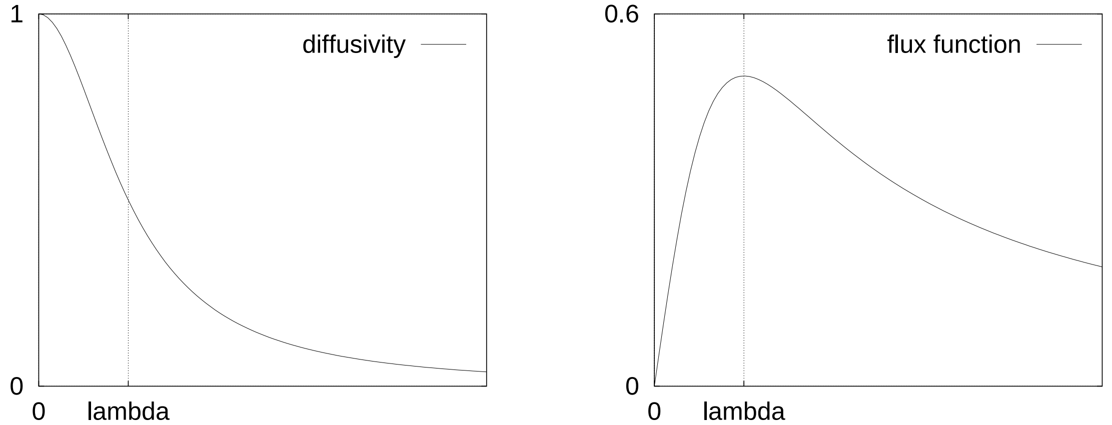
The Perona–Malik Filter (3)



Edge enhancement with the Perona-Malik filter. (a) **Left:** Original image, $\Omega = (0, 256)^2$. (b) **Middle:** Linear diffusion filtering , $t = 100$. (c) **Right:** Perona–Malik diffusion, $\lambda = 2.5$, $t = 100$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Why Do We Have Edge Enhancement?



(a) **Left:** Diffusivity $g(s^2) = \frac{1}{1+s^2/\lambda^2}$. (b) **Right:** Corresponding flux function $\Phi(s) = \frac{s}{1+s^2/\lambda^2}$.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Consider 1-D Perona–Malik equation

$$\partial_t u \;=\; \partial_x \Big(g(u_x^2)\, u_x \Big)$$

with the diffusivity

$$g(u_x^2) \;=\; \frac{1}{1+u_x^2/\lambda^2}.$$

The flux function

$$\Phi(u_x) \;:=\; g(u_x^2) \cdot u_x$$

gives

$$\partial_t u \;=\; (\Phi(u_x))_x \;=\; \Phi'(u_x)\, u_{xx},$$

with

$$\begin{array}{llll} \Phi'(u_x) > 0 & \text{for} & |u_x| < \lambda & \text{(forward diffusion)} \\ \Phi'(u_x) < 0 & \text{for} & |u_x| > \lambda & \text{(backward diffusion)} \end{array}$$

Thus we have (smoothing) *forward diffusion* for $|u_x| < \lambda$
and (edge-enhancing) *backward diffusion* for $|u_x| > \lambda$.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

The Perona–Malik Filter (6)

Diffusivities Leading to Forward–Backward Diffusion

$$g(s^2) = \frac{1}{1+s^2/\lambda^2}$$

$$g(s^2) = \exp\left(\frac{-s^2}{2\lambda^2}\right)$$

$$g(s^2) = \begin{cases} 1 & (s^2 = 0) \\ 1 - \exp\left(\frac{-3.31488}{(s/\lambda)^8}\right) & (s^2 > 0). \end{cases}$$

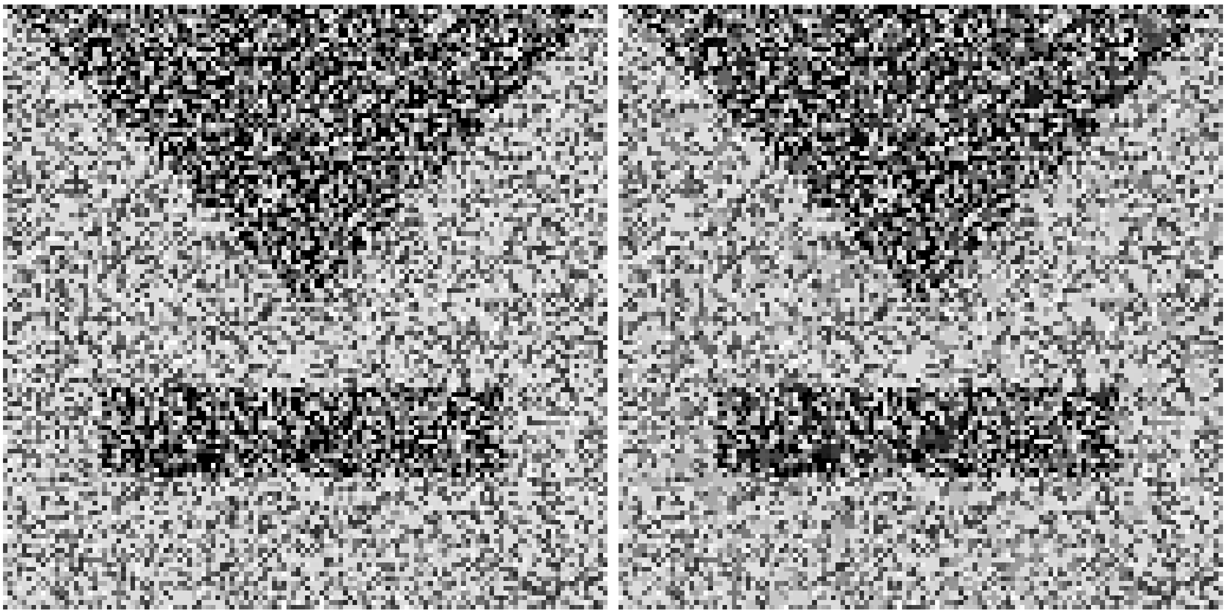
Properties:

- $g > 0$ and $g \in C^\infty$.
- $g(0) = 1$, g decreasing on $[0, \infty)$, and $\lim_{s^2 \to \infty} g(s^2) = 0$.
- The contrast parameter $\lambda > 0$ separates forward and backward diffusion.
- The latter diffusivities decrease more rapidly.
  Often they give more 'segmentation-like' results.

The Perona–Malik Filter (7)

What About Well-Posedness ?

- Pure backward diffusion processes are ill-posed:
  Solution exists only for C^∞ initial data and is highly sensitive to perturbations.
- Kichenassamy (1997), Esedoglu (2001), March et al. (2008), ... :
  Established complicated *continuous* theory for forward-backward diffusion equations of Perona-Malik type:
  - existence of a generalised solution (with discontinuities)
  - instability w.r.t. perturbations of the initial data
- However, there is mainly one experimentally observable artifact for *discrete* implementations of the Perona–Malik filter: staircasing
- Weickert and Benhamouda (1997):
  - spatial finite difference discretisation creates well-posed problem:
    unique solution for $t \in [0, \infty)$, stable, extremum principle, constant steady-state
  - 1-D explicit scheme is monotonicity preserving:
    monotone initial data remain monotone after filtering



Besides some well-posedness problems, the Perona-Malik filter also suffers from a practical shortcoming: It misinterprets noise as high-gradient edges and tries to preserve it for a long time. (a) **Left:** Test image, $\Omega = (0, 128)^2$. (b) **Right:** Perona–Malik diffusion, $\lambda = 3.5$, $t = 80$.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Regularised Nonlinear Diffusion

Goals:

- have full well-posedness already in *continuous* model
- become more independent of the choice of the discretisation
- reduce staircasing artifacts and problems with noise

First Regularisation Attempt in the Context of Oceanography:

- Posmentier (1977) avoided small-scale staircasing by averaging ∇u in $g(|\nabla u|^2)$ over some neighbourhood.

Regularised Nonlinear Diffusion (2)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Sound Mathematical Formulation in Image Processing:

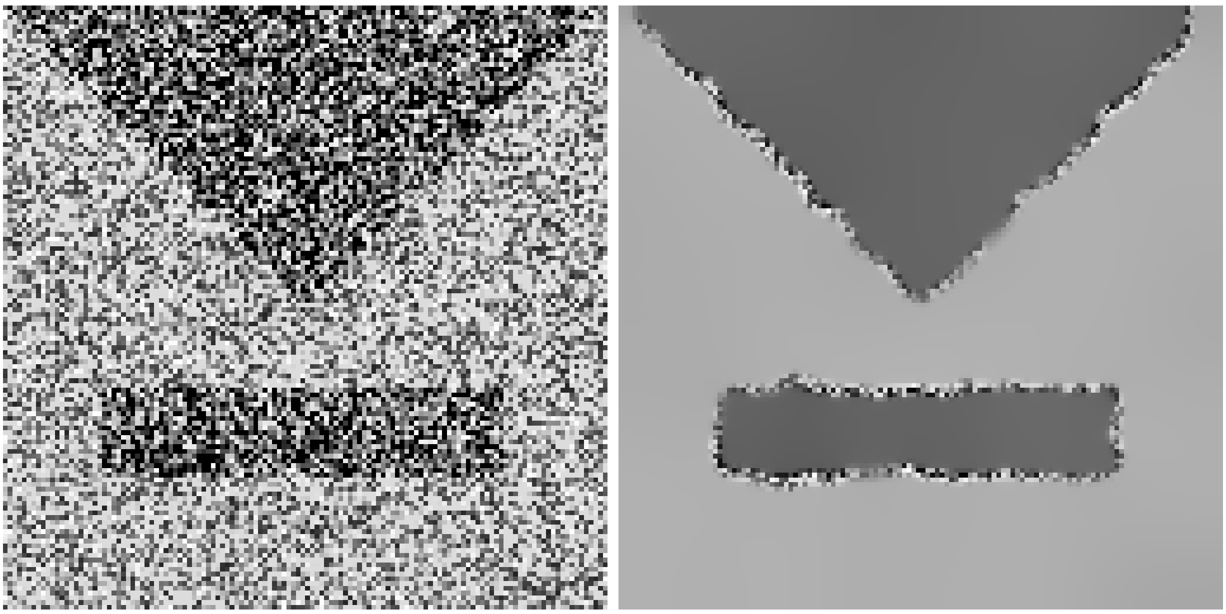
- Catté/Lions/Morel/Coll (1992) used Gaussian convolution for averaging
- proved existence of unique smooth solution
- continuous dependence on the initial data (Weickert 1998)

Numerical Experiments:

- more robust under noise (Catté et al. 1992)
- can still enhance edges (Benhamouda 1994)
- can avoid staircasing (Nitzberg/Shiota 1992)
- less sensitive to discretisation (Fröhlich/Weickert 1994)

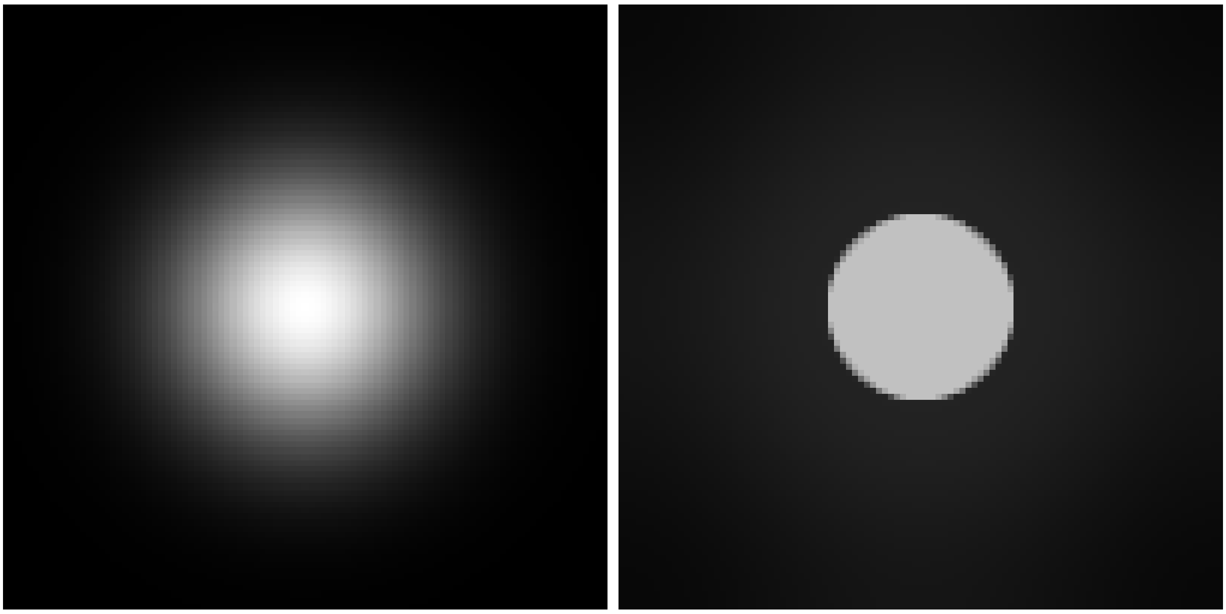
Regularised Nonlinear Diffusion (3)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



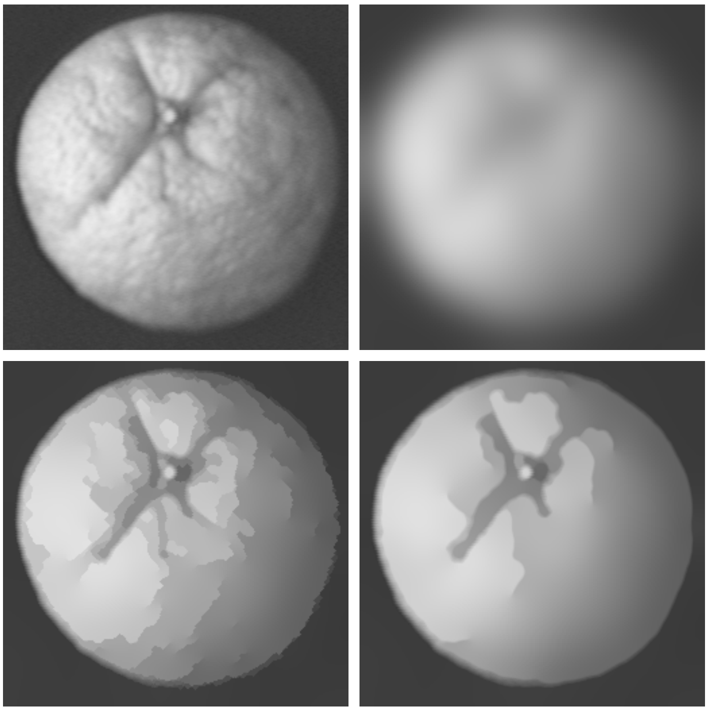
Regularisation renders the Perona-Malik filter more robust under noise. (a) **Left:** Test image, $\Omega = (0, 128)^2$. (b) **Right:** Regularised isotropic nonlinear diffusion, $\lambda = 3.5$, $\sigma = 3$, $t = 80$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



Regularisation of the Perona-Malik filter does still allow edge enhancement. (a) **Left:** Gaussian-like image, $\Omega = (0, 101)^2$. (b) **Right:** Regularised isotropic nonlinear diffusion, $\lambda = 9$, $\sigma = 0.7$, $t = 250$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



Regularisation of the Perona-Malik filter reduces staircasing artifacts. (a) **Top left:** Original image, $\Omega = (0, 256)^2$. (b) **Top right:** Linear diffusion, $t = 100$. (c) **Bottom left:** Perona–Malik diffusion, $\lambda = 2.5$, $t = 100$. (d) **Bottom right:** Regularised isotropic nonlinear diffusion, $\lambda = 2.5$, $\sigma = 1.5$, $t = 100$.

Continuous Well-Posedness and Scale-Space Theory

Assumptions on the Isotropic Regularised Filter Class

Let f be bounded, $\sigma > 0$, $\nabla u_\sigma := \nabla(K_\sigma * u)$, and consider

$$\begin{aligned} \partial_t u = \operatorname{div}\left(g(|\nabla u_\sigma|^2)\, \nabla u\right) \quad & \text{on} \quad \Omega \times (0,\infty), \\ u(\mathbf{x},0) = f(\mathbf{x}) \quad & \text{on} \quad \Omega, \\ \partial_{\mathbf{n}} u = 0 \quad & \text{on} \quad \partial\Omega \times (0,\infty), \end{aligned}$$

where the diffusivity g satisfies

- (C1') Smoothness:
  $g \in \mathrm{C}^\infty[0,\infty)$.

- (C3') Positivity:
  $g > 0$.

Theoretical Results

(a) **Well-Posedness and Regularity**

- existence of a unique (distributional) solution $u(\mathbf{x},t)$
- smoothness: $u \in \mathrm{C}^\infty(\bar{\Omega} \times (0,\infty))$
- $u(t)$ depends continuously on f w.r.t. $\|\,.\,\|_{\mathrm{L}^2(\Omega)}$.

Practical Importance:
Stability under perturbations of initial image:
useful for stereo images, image sequences, CT/MRI data, ...

(b) **Preservation of Average Grey Level**

$$\frac{1}{|\Omega|} \int_\Omega u(\mathbf{x},t)\, d\mathbf{x} \;=\; \frac{1}{|\Omega|} \int_\Omega f(\mathbf{x})\, d\mathbf{x} \;=:\; \mu.$$

Practical Importance:
medical imaging and some segmentation algorithms (hyperstack)

Continuous Well-Posedness and Scale-Space Theory (3)

(c) **Maximum–Minimum Principle**

$$\inf f \ \le \ u(\mathbf{x},t) \ \le \ \sup f$$

Practical Importance:
closely related to smoothing properties (causality);
e.g. Hummel 1986: maximum principle for parabolic operators
$\iff$ scale-space creates no new level crossings for $t > 0$.

(d) **Lyapunov Functionals**

$$V(t) \ := \int_{\Omega} r(u(\mathbf{x},t))\, d\mathbf{x}$$

is a Lyapunov function for all convex $r \in \mathrm{C}^2$:
$V(t)$ is decreasing and bounded from below.

Continuous Well-Posedness and Scale-Space Theory (4)

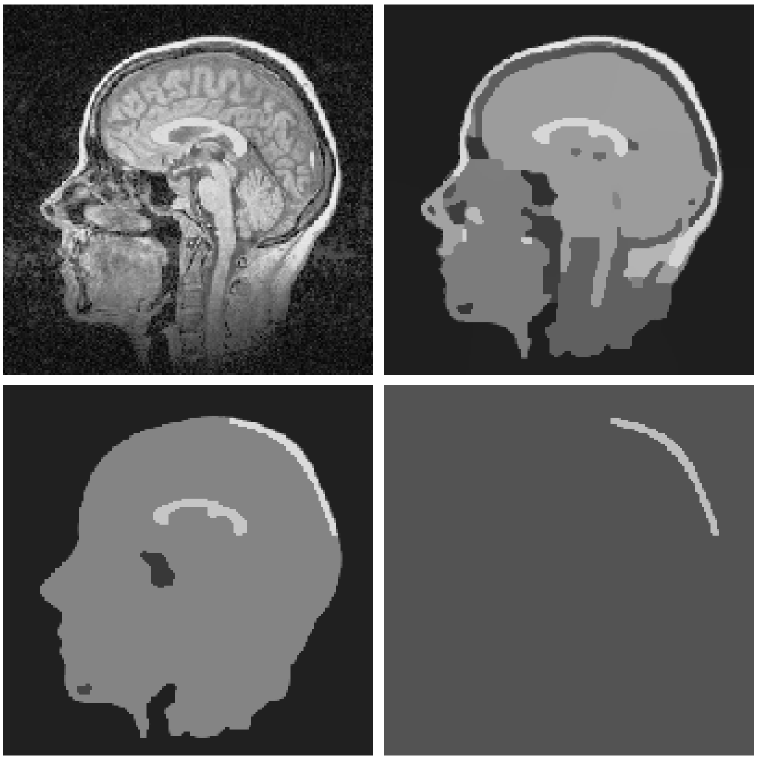
Practical Importance:
Transformation is simplifying, information-reducing:
Interesting results for specific r:

- **Interpretation in Fourier and Wavelet Domain:**
  $r(s) := |s|^p$ with $2 \le p \le \infty$:
  L^p norms decrease in t.
- **Probabilistic Interpretation:**
  $r(s) := (s-\mu)^{2n}$:
  All even central moments decrease.
- **Information Theoretic Interpretation:**
  $r(s) := s \ln s$:
  Entropy $-\int u \ln u \, d\mathbf{x}$ increases (if f is positive).

(e) **Constant Image for $t \to \infty$**
$u(x,t)$ converges to average grey level μ.
(Convergence in $L^p(\Omega)$, $1 \le p < \infty$.)

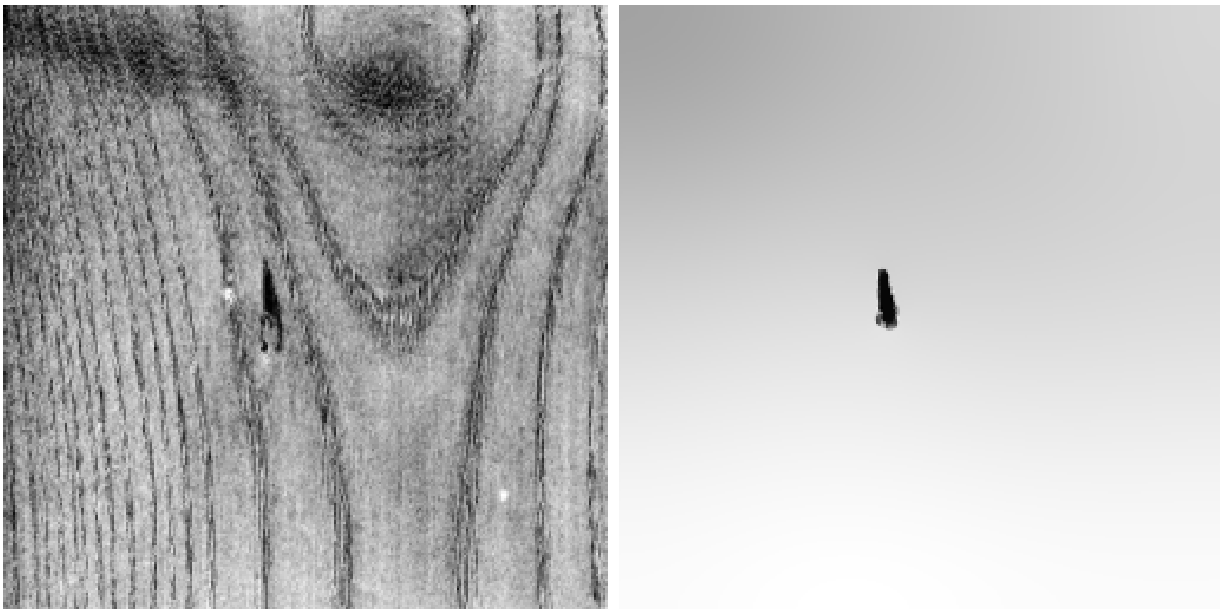
Practical Importance:
Most global image representation within a scale-space.

Continuous Well-Posedness and Scale-Space Theory (5)



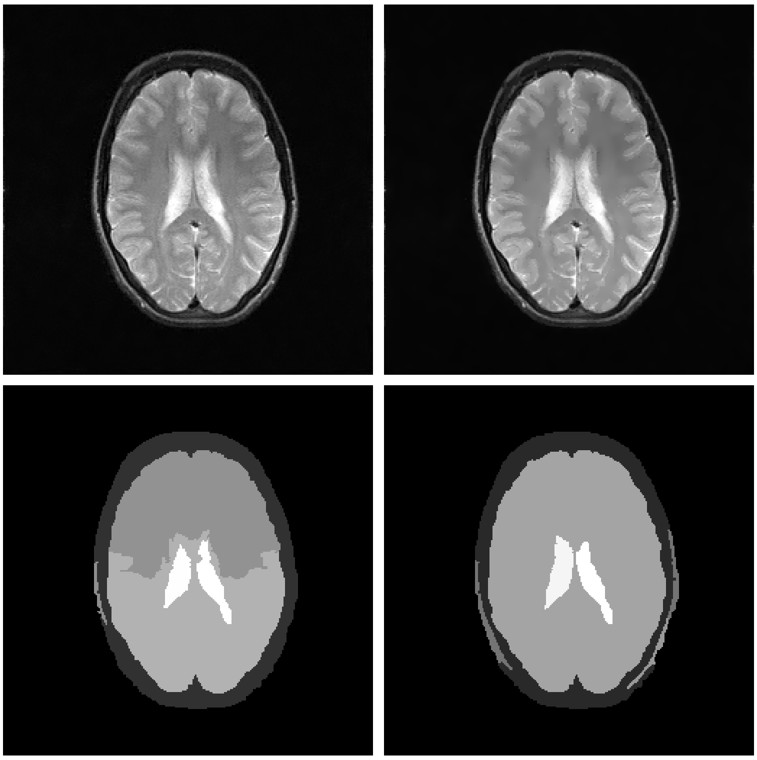
Scale-space behaviour of nonlinear isotropic diffusion filtering ($\lambda = 3$, $\sigma = 1$). (a) **Top left:** Original image, $\Omega = (0, 236)^2$. (b) **Top right:** $t = 2.5 \cdot 10^4$. (c) **Bottom left:** $t = 5 \cdot 10^5$. (d) **Bottom right:** $t = 7 \cdot 10^6$.

Continuous Well-Posedness and Scale-Space Theory (6)



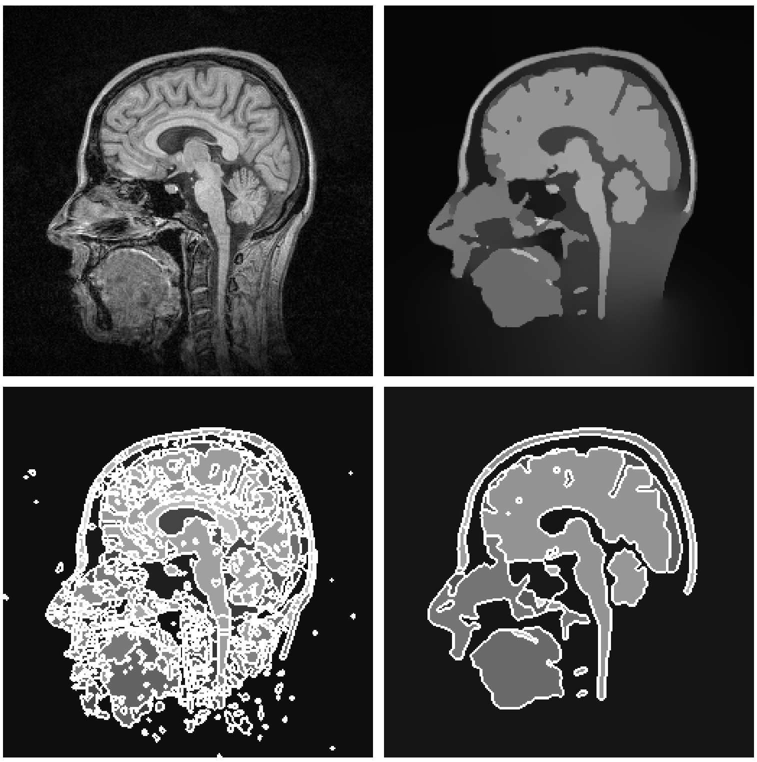
Defect detection in wood. (a) **Left:** Wood surface, $\Omega = (0, 256)^2$. (b) **Right:** Isotropic nonlinear diffusion, $\lambda = 4$, $\sigma = 2$, $t = 2000$.

Continuous Well-Posedness and Scale-Space Theory (7)



Preprocessing of an MRI slice prior to segmentation. (a) **Top left:** Head, $\Omega = (0, 256)^2$. (b) **Top right:** Diffusion-filtered, $\lambda = 5$, $\sigma = 0.1$, $t = 2.5$. (c) **Bottom left:** Segmented original image (Mumford-Shah cartoon model), $\alpha = 8192$. (d) **Bottom right:** Segmented filtered image, $\alpha = 8192$.

Continuous Well-Posedness and Scale-Space Theory (8)



Preprocessing of an MRI slice prior to segmentation. (a) **Top left:** Test image. (b) **Top right:** Filtered by isotropic nonlinear diffusion. (c) **Bottom left:** Segmented original image (watersheds with region merging). (d) **Bottom right:** Segmented diffused image.

Summary

- Perona-Malik filter avoids blurring and delocalisation of edges
- edge enhancement by diffusivity rapidly decreasing in $|\nabla u|^2$
- nonmonotone flux function causes forward–backward diffusion
- unique generalised solution, but in no continuous dependence on the initial data
- well-posedness in case of Gaussian regularisation: $g(|\nabla u_\sigma|^2)$ instead of $g(|\nabla u|^2)$
- practical properties of regularised Perona–Malik filter:
  - more robust under noise
  - keeps advantages of Perona–Malik filter (well-localised, enhanced edges)
  - less staircasing
- theoretical properties of regularised Perona–Malik filter:
  - extremum principle (causality) for $t \in [0, \infty)$
  - average grey level invariance
  - many smoothing Lyapunov functionals
  - convergence to constant steady-state

References

- P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, 629–639, 1990.
  (first journal paper on nonlinear diffusion in image analysis; warning: numerical algorithm presented there does not approximate the continuous equation)
- F. Catté, P.-L. Lions, J.-M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM Journal on Numerical Analysis*, Vol. 29, 182–193, 1992.
  (existence and uniqueness proof for a regularised Perona–Malik model)
- J. Weickert, *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998.
  (This lecture is based on Sections 1.3.1, 1.3.2, and Chapter 2.)

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Assignment P1 – Programming

You can download the file `Ex01.tar` from the web page

`http://www.mia.uni-saarland.de/Teaching/dic08.shtml`

To unpack the data, use `tar xvf Ex01.tar`.

(a) Implement the explicit finite difference scheme for linear diffusion filtering. The following things should make life easier:

- The main programme is available as object code `ild.o`. It contains all necessary auxiliary routines for dynamic storage allocation, image reading and writing, and for supplementing reflecting boundary pixels. It only requires a correct linear diffusion routine `lindiff`.
- Supplement the file `lindiff.c` with the missing code so that it creates a smoother output image `u` from an input image `f`.
- Images are two-dimensional arrays with the following pixel range:
  x direction: `0, ..., nx+1`
  y direction: `0, ..., ny+1`
  The first and the last pixels are the dummy boundary pixels.
- To compile your final programme, use

```
gcc -O2 -o ild ild.o lindiff.c -lm
```

  Typing `./ild` afterwards starts an interactive programme for linear diffusion filtering.
- The external image format is the PGM (P5) format (consisting of a header and grey values between 0 and 255, represented by 1 byte per pixel). To visualise an image `fabric.pgm`, just type: `xv fabric.pgm &`.

(b) Run the programme on the image `fabric.pgm`. It depicts a nonwoven fabric. One of its quality relevant parameter is the cloudiness. Try to visualise the cloudiness at different scales by choosing different iteration numbers. Is the evolution of the mean, the maximum, the minimum, and the variance in accordance with your expectation? What happens for times steps larger than 0.25?

(c) To cross-validate your programme, you may use the programme `gauss_conv`, which performs spatial convolution with truncated and sampled Gaussians. For comparing two images, you can use the programme `difference`. Don't overdo it: since the greyvalues are rounded to integer precision, a maximum difference of 1 is normal.

(d) Use the programme `./ild` to reduce the noise in the image `office-n40.pgm`. Try out different stopping times in order to achieve a visually good result. Is linear diffusion suitable for image denoising? To answer this question, you can calculate the so-called *method noise*: With the programme `difference`, you can calculate the difference between the noisy input image `office-n40.pgm` and several resulting images after applying linear diffusion. This helps to understand what the denoising method actually removes from the input.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Assignment P1 (3)

For assessment:
Use the command

```
tar -cvzf P1_yourname.tgz file 1 file 2 ...
```

to pack the following files into an archive:

- from task 1: the supplemented `lindiff.c`,
- from task 2: two representative diffusion-filtered images illustrating the cloudiness at different scales,
- from task 3: for one of the diffusion-filtered images from the previous step the corresponding Gaussian convolution counterpart and the difference image,
- from task 4: the result of linear diffusion for a suitable parameter and the corresponding method noise image.

Include in the archive also a short `README` file stating the parameters (iteration number, time step size, or sigma, respectively). for the submitted images. Depending on your tutorial group, send the archive by e-mail to one of the addresses

- `dic-g1@mia.uni-saarland.de` (group 1, Tuesday 16-18)
- `dic-g2@mia.uni-saarland.de` (group 2, Thursday 12-14)
- `dic-g3@mia.uni-saarland.de` (group 3, Thursday 16-18)

Deadline for electronic submission: Friday, May 2, 11 am.