Lecture 14:
Image Enhancement V:
Nonlinear Diffusion Filtering

Contents

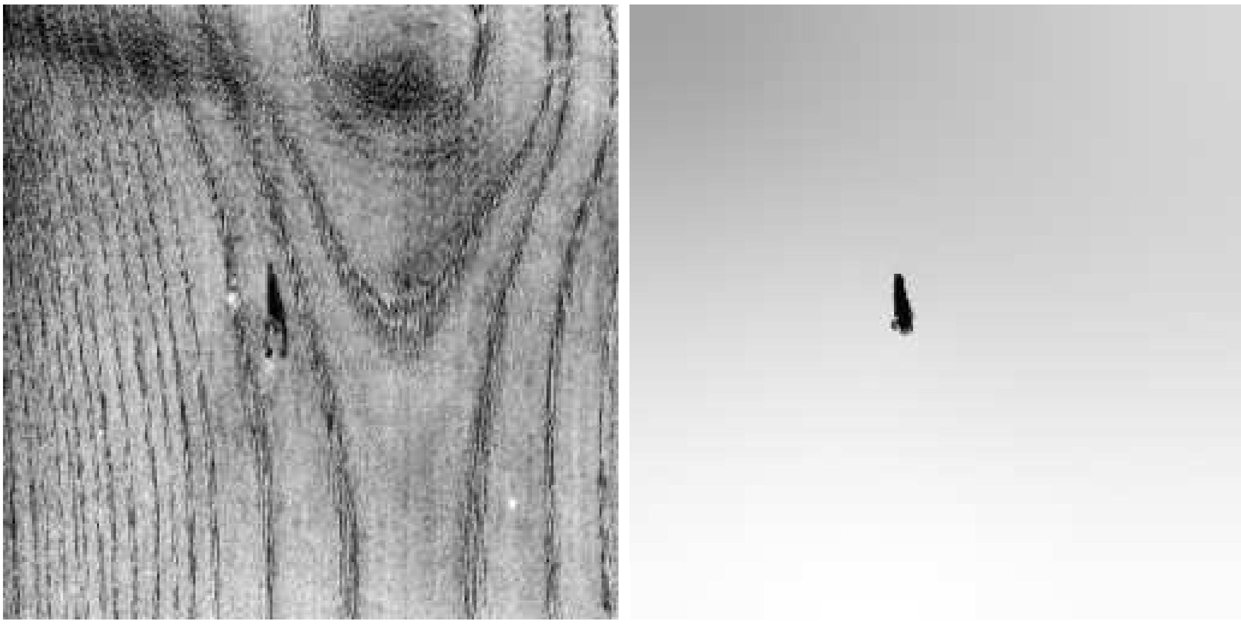
Motivation

Nonlinear Diffusion Filters

- offer a number of advantages:
  - smooth within a region and preserve edges
    (in contrast to shift-invariant linear filters)
  - preserve the average grey value
    (in contrast to morphological methods and most M-smoothers)
  - are shift-invariant and do not suffer from over- and undershoots
    (in contrast to classical wavelet shrinkage)
  - can be designed in a flexible way and often give impressive results
- do also have some less pleasant properties:
  - somewhat slower than linear and morphological filters
  - mathematically a bit more demanding

Therefore, we have to refresh our maths first.

Motivation (2)

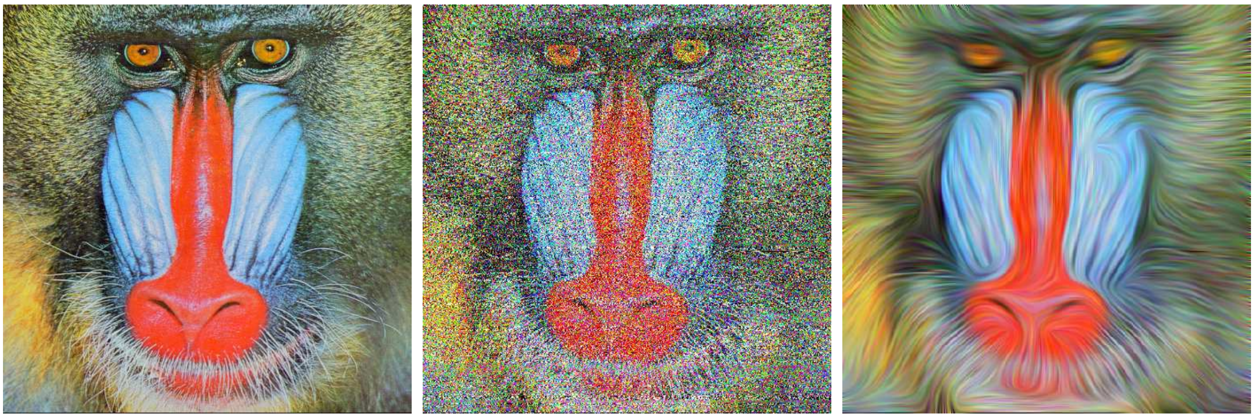


Defect detection in a wood surface. **(a) Left:** Wood surface, $\Omega = (0, 256)^2$. **(b) Right:** Isotropic nonlinear diffusion filtering (model of Catté et al., $\lambda = 4$, $\sigma = 2$, $t = 2000$). Author: J. Weickert (1996).

Motivation (3)

Fingerprint enhancement. **(a) Left:** Original image, $\Omega = (0, 256)^2$. **(b) Right:** Coherence-enhancing anisotropic diffusion, $\sigma = 0.5$, $\rho = 4$, $t = 20$. Author: J. Weickert (1995).

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 28 29



Robustness of vector-valued diffusion filtering with respect to noise. **(a) Left:** Mandrill, $\Omega = (0, 512)^2$. **(b) Middle:** With additive Gaussian noise. Noise variance per channel: four times the signal variance. **(c) Right:** Coherence enhancing anisotropic diffusion, $\sigma = 1$, $\rho = 12$, $t = 46$. Author: J. Weickert (1997).

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 28 29

Mathematical Background

Partial Derivatives

- For a sufficiently smooth function of several variables, one can compute its *partial derivatives* with respect to each of these variables. To this end, one regards it as a function of one variable and treats the other variables like constants.

- Example:

$$\begin{aligned} f(x,y) &= \sin(xy^2) + x^3 \\ \frac{\partial f}{\partial x}(x,y) &= y^2\cos(xy^2) + 3x^2 \\ \frac{\partial f}{\partial y}(x,y) &= 2xy\cos(xy^2) \end{aligned}$$

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 28 29

Mathematical Background (2)

- Equivalent notations:

$$\frac{\partial f}{\partial x} = \partial_x f = f_x$$

- Partial derivatives can be computed consecutively:

$$\frac{\partial^2 f}{\partial x \, \partial y} := \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right)$$

- Under suitable smoothness assumptions one may exchange the order of partial differentiation:

$$f_{xy} = f_{yx}$$

Mathematical Background (3)

Nabla Operator

- The column vector of the partial derivatives is called *nabla operator* or *gradient*. In 2-D:

$$\nabla := \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}$$

- Often it is possible to work with ∇ as if it were an ordinary vector. For a scalar-valued function $f(x, y)$, one gets e.g.

$$\nabla f = \begin{pmatrix} \partial_x f \\ \partial_y f \end{pmatrix}$$

∇f points in the direction of the steepest ascend of f.

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 28 29

Divergence and Laplacian

- The inner product of the nabla operator and a vector-valued function $\mathbf{j}(x,y) = (j_1(x,y),\ j_2(x,y))^\top$ is called the *divergence (Divergenz)* of $\mathbf{j}$:

$$\operatorname{div}\mathbf{j} \;:=\; \nabla^\top \mathbf{j} \;=\; (\partial_x, \partial_y) \begin{pmatrix} j_1 \\ j_2 \end{pmatrix} \;=\; \partial_x j_1 + \partial_y j_2$$

- The inner product of the divergence and the gradient is called *Laplacian (Laplace-Operator)*:

$$\Delta f \;:=\; \operatorname{div}(\nabla f) \;=\; (\partial_x, \partial_y) \begin{pmatrix} \partial_x f \\ \partial_y f \end{pmatrix} \;=\; \partial_{xx} f + \partial_{yy} f.$$

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 28 29

Partial Differential Equations (PDEs)

- *Algebraic equations* state relations between an unknown *number* and its powers.
  Example:
  $$x^2 - 8x + 15 = 0.$$
  Two solutions: $x_1 = 3$, $x_2 = 5$.

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 28 29

Mathematical Background (6)

- *Differential equations* state relations between an unknown *function* and its derivatives. Example:

  $$\frac{du(t)}{dt} = 5u(t).$$

  Infinitely many solutions:

  $$u(t) = ae^{5t} \qquad \text{(for every arbitrary } a \in \mathbb{R}\text{)}.$$

  Specifying an additional *initial condition* such as

  $$u(t=0) = 2,$$

  can lead to a unique solution:

  $$u(t) = 2e^{5t}.$$

- If u depends only on a single variable (t), we have ordinary derivatives and the differential equation is an *ordinary ordinary differential equation (ODE, gewöhnliche Differentialgleichung)*.

- If u is a function of multiple variables (e.g. x,t), partial derivatives may appear. Then we have a *partial differential equation (PDE, partielle Differentialgleichung)*.

Mathematical Background (7)

Example

- Consider the *linear one-dimensional diffusion equation*

  $$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

  with initial condition $f(x)$:

  $$u(x, t=0) = f(x).$$

- It can be shown that this problem has a unique solution that is given by convolution with a Gaussian K_σ with standard deviation $\sigma = \sqrt{2t}$:

  $$u(x,t) = (K_{\sqrt{2t}} * f)(x) := \int_{-\infty}^{\infty} K_{\sqrt{2t}}(x')\, f(x-x')\, dx'$$

- Such nice analytical solutions only exist for a few simple PDEs.
  Usually numerical approximations are required.

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 28 29

Physical Background: What is Diffusion?

- Diffusion equilibrates concentration differences by redistributing mass.
  It preserves the total mass.

- (Isotropic) Diffusion processes are described by the PDE

$$\partial_t u \;=\; \mathrm{div}\,(g \cdot \nabla u)$$

  where the divergence and the nabla operator involve the spatial derivatives only, not the temporal ones.

- The longer the diffusion time t, the more the concentration differences are equilibrated.

- The diffusivity g may depend on the location.
  At locations where g is larger, the diffusion proceeds faster.

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 28 29

A Continuous Diffusion Filter

Diffusion in Image Processing

grey values:	are regarded as concentrations
image domain:	rectangular domain $\Omega := (0, a_1) \times (0, a_2)$
image:	bounded function $f : \Omega \to \mathbb{R}$

Diffusion Filter:

Computes a filtered version $u(x, y, t)$ of $f(x, y)$ as solution of the diffusion equation

$$\partial_t u = \mathrm{div}\,(g \; \nabla u)$$

with the original image as initial condition,

$$u(x, y, 0) = f(x, y),$$

and reflecting boundary conditions
($\mathbf{n}$: unit normal vector at image bounday $\partial\Omega$, and $\partial_{\mathbf{n}} u := \mathbf{n}^\top \nabla u$):

$$\partial_{\mathbf{n}} u = 0.$$

A Continuous Diffusion Filter (2)

How is the Diffusivity g Chosen ?

- We want to reduce the diffusion at edges in order to preserve them.

- $|\nabla u| = \sqrt{u_x^2 + u_y^2}$ is a reasonable indicator for an edge:
  Edges are locations where $|\nabla u|$ is large (more details in Lecture 18).

- Choose a diffusvity that is decreasing in $|\nabla u|$, e.g. the *Charbonnier diffusivity*

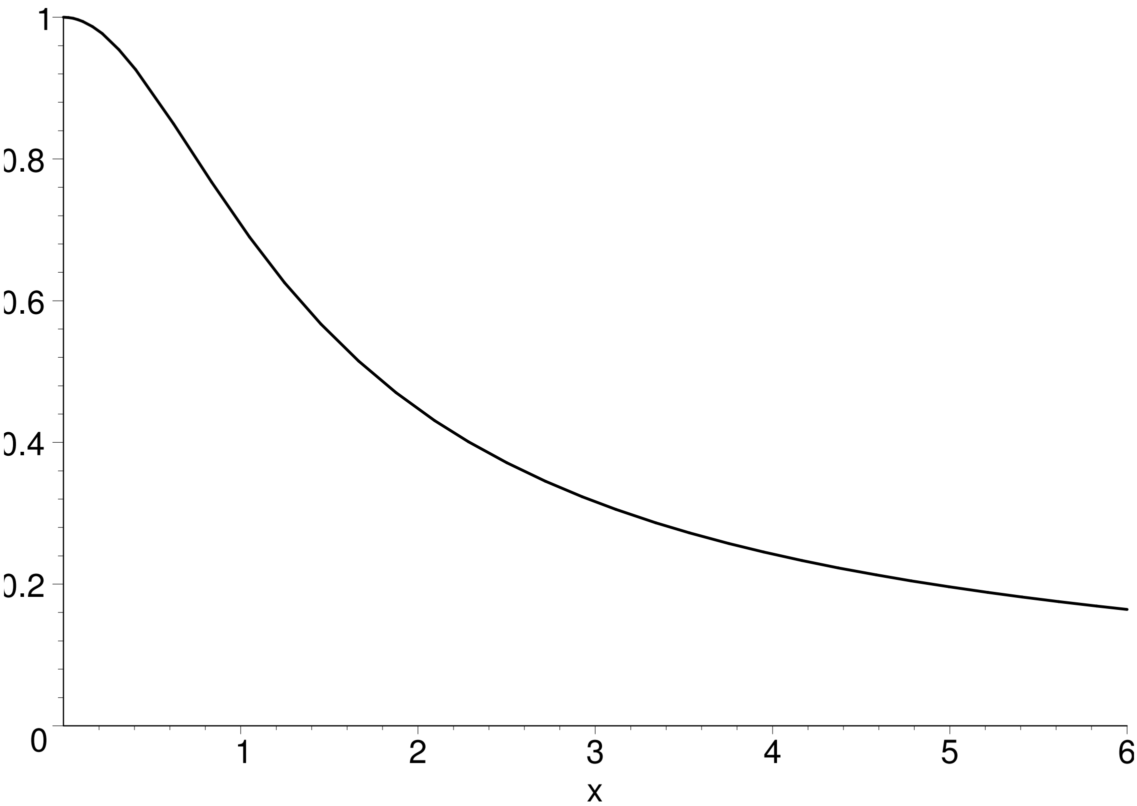
$$g(|\nabla u|) \;:=\; \frac{1}{\sqrt{1 + |\nabla u|^2/\lambda^2}}$$

  or the faster decreasing *Perona–Malik diffusivity* (allowing edge enhancement)

$$g(|\nabla u|) \;:=\; \frac{1}{1 + |\nabla u|^2/\lambda^2}$$

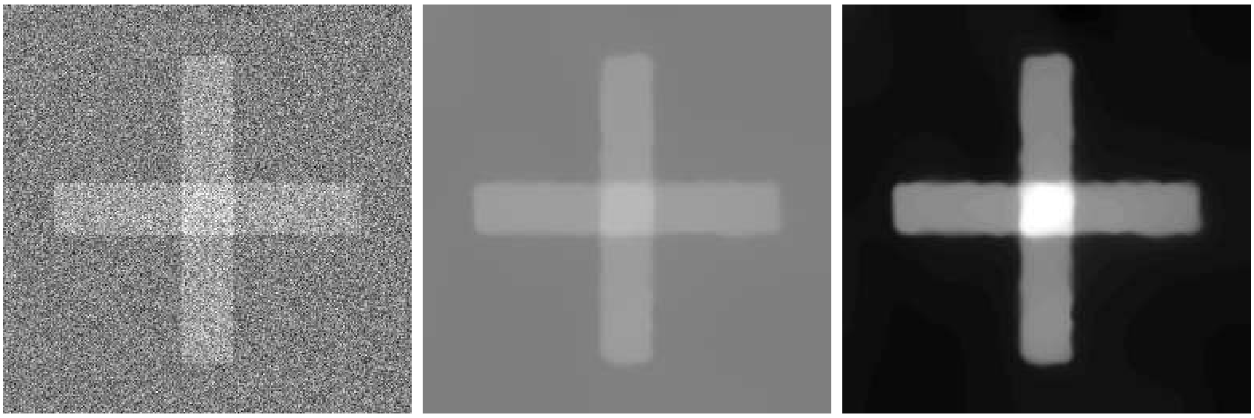
- The positive parameter λ serves as *contrast parameter:*
  Locations with $|\nabla u| > \lambda$ are regarded as edges, where the diffusivity is reduced significantly.

A Continuous Diffusion Filter (3)



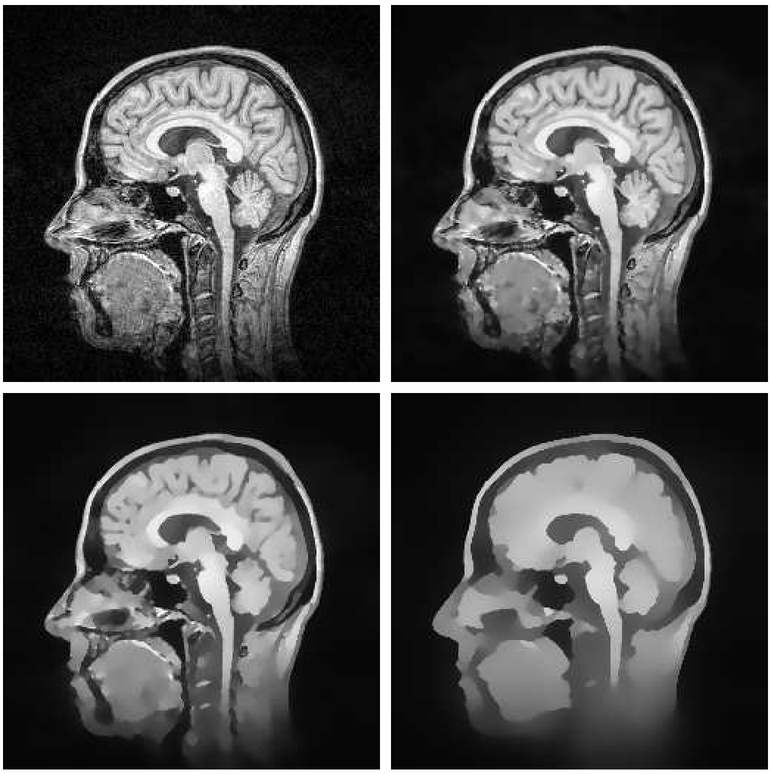
Plot of the Charbonnier diffusivity for $\lambda = 1$. Author: B. Burgeth (2002).

A Continuous Diffusion Filter (4)



Left: Noisy original image, 256×256 pixels. **Middle:** Nonlinear diffusion with Charbonnier diffusivity, $\lambda = 0.1$, $t = 500$. **Right:** Affine rescaling of the range of (b) to the interval $[0, 255]$. Author: J. Weickert (2002).

A Continuous Diffusion Filter (5)



Influence of the diffusion time t. **(a) Top left:** Original image, $\Omega = (0, 256)^2$. **(b) Top right:** Filtered with the Perona-Malik diffusivity with $\lambda = 4$ and diffusion time $t = 5$. **(c) Bottom left:** $t = 20$. **(d) Bottom right:** $t = 80$. Author: J. Weickert (2007).

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 28 29

Influence of the contrast parameter λ. **(a) Top left:** Filtered with the Perona-Malik diffusivity with diffusion time $t = 20$ and contrast parameter $\lambda = 2$. **(b) Top right:** $\lambda = 4$. **(c) Bottom left:** $\lambda = 6$. **(d) Bottom right:** $\lambda = 10$. Author: J. Weickert (2007).

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 28 29

A Simple Algorithm

Finite Difference Approximation

- rewrite the continuous diffusion equation as

$$\partial_t u \;=\; \partial_x\Big(g(|\nabla u|)\,\partial_x u\Big) \;+\; \partial_y\Big(g(|\nabla u|)\,\partial_y u\Big)$$

- $u_{i,j}^k$ approximates $u(ih_1, jh_2, k\tau)$ with spatial grid sizes h_1, h_2, and time step size τ.
- approximations in $(ih_1, jh_2, k\tau)$:

$$\begin{aligned}
\partial_t u &\approx \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\tau} \\
\partial_x(g\,\partial_x u) &\approx \frac{1}{h_1}\left((g\,\partial_x u)_{i+1/2,j}^k - (g\,\partial_x u)_{i-1/2,j}^k\right) \\
&\approx \frac{1}{h_1}\left(g_{i+1/2,j}^k\,\frac{u_{i+1,j}^k - u_{i,j}^k}{h_1} - g_{i-1/2,j}^k\,\frac{u_{i,j}^k - u_{i-1,j}^k}{h_1}\right).
\end{aligned}$$

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 28 29

A Simple Algorithm (2)

- The diffusivity approximation $g^k_{i+1/2,j}$ is the arithmetic mean of $g^k_{i,j}$ and $g^k_{i+1,j}$.
- $g^k_{i,j}$ can be computed as follows:

$$
\begin{aligned}
g^k_{i,j} &= g\left(\sqrt{(\partial_x u)^2 + (\partial_y u)^2}\right)\Big|^k_{i,j} \\
&\approx g\left(\sqrt{\left(\frac{u^k_{i+1,j} - u^k_{i-1,j}}{2h_1}\right)^2 + \left(\frac{u^k_{i,j+1} - u^k_{i,j-1}}{2h_2}\right)^2}\right)
\end{aligned}
$$

A Simple Algorithm (3)

Resulting Numerical Scheme:

$$
\begin{aligned}
\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\tau} &= \frac{1}{h_1}\left(g^k_{i+1/2,j}\,\frac{u^k_{i+1,j} - u^k_{i,j}}{h_1} - g^k_{i-1/2,j}\,\frac{u^k_{i,j} - u^k_{i-1,j}}{h_1}\right) \\
&+ \frac{1}{h_2}\left(g^k_{i,j+1/2}\,\frac{u^k_{i,j+1} - u^k_{i,j}}{h_2} - g^k_{i,j-1/2}\,\frac{u^k_{i,j} - u^k_{i,j-1}}{h_2}\right)
\end{aligned}
$$

u^{k+1}_{ij} can be *explicitly* computed from known values at the time step k:

$$
\begin{aligned}
u^{k+1}_{i,j} = u^k_{i,j} &+ \frac{\tau}{h_1}\left(g^k_{i+1/2,j}\,\frac{u^k_{i+1,j} - u^k_{i,j}}{h_1} - g^k_{i-1/2,j}\,\frac{u^k_{i,j} - u^k_{i-1,j}}{h_1}\right) \\
&+ \frac{\tau}{h_2}\left(g^k_{i,j+1/2}\,\frac{u^k_{i,j+1} - u^k_{i,j}}{h_2} - g^k_{i,j-1/2}\,\frac{u^k_{i,j} - u^k_{i,j-1}}{h_2}\right).
\end{aligned}
$$

This scheme is called an *explicit scheme*.

A Simple Algorithm (4)

- The explicit scheme can be rewritten as weighted averaging:

$$
\begin{aligned}
u_{i,j}^{k+1} \;=\;& \frac{\tau g_{i+1/2,j}^k}{h_1^2}\, u_{i+1,j}^k + \frac{\tau g_{i-1/2,j}^k}{h_1^2}\, u_{i-1,j}^k + \frac{\tau g_{i,j+1/2}^k}{h_2^2}\, u_{i,j+1}^k + \frac{\tau g_{i,j-1/2}^k}{h_2^2}\, u_{i,j-1}^k \\
&+ \left(1 - \frac{\tau g_{i+1/2,j}^k}{h_1^2} - \frac{\tau g_{i-1/2,j}^k}{h_1^2} - \frac{\tau g_{i,j+1/2}^k}{h_2^2} - \frac{\tau g_{i,j-1/2}^k}{h_2^2}\right) u_{i,j}^k
\end{aligned}
$$

- At the image borders, introduce an additional one-pixel layer by mirroring. This ensures reflecting boundary conditions.

A Simple Algorithm (5)

Stencil Notation for $h_1 := h_2 := 1$:

0	$\tau g_{i,j+1/2}^k$	0
$\tau g_{i-1/2,j}^k$	$-\tau g_{i,j+1/2}^k$ $-\tau g_{i-1/2,j}^k + 1 - \tau g_{i+1/2,j}^k$ $-\tau g_{i,j-1/2}^k$	$\tau g_{i+1/2,j}^k$
0	$\tau g_{i,j-1/2}^k$	0

- space- and time-dependent adaptive averaging where the weights sum up to 1
- central weight becomes negative if τ is too large $\Longrightarrow$ instability
- For $|g| \leq 1$ we get stable convex combinations if $\tau \leq 0.25$.
  In this case no over- and undershoots are possible.

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 28 29

Summary (1)

Summary

- Nonlinear diffusion filters regard the original image as initial state of a diffusion process.
- Its diffusivity is reduced at edges.
- A simple discretisation leads to an iterated adaptive averaging.
- The weights are determined from the time step size and the diffusivity.
- For suitable discretisations, nonlinear diffusion filtering preserves the average grey value, and it does not create over- and undershoots (Pseudo–Gibbs phenomena).

Summary (2)

Literature

- P. Perona, J. Malik: Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 12, 629–639, 1990.
  (the first journal paper on nonlinear diffusion filtering)
- J. Weickert: Nonlinear diffusion filtering. In B. Jähne, H. Haußecker, P. Geißler (Eds.): *Handbook of Computer Vision and Applications, Vol. 2: Signal Processing and Pattern Recognition*. Academic Press, San Diego, pp. 423–450, 1999.
  (survey paper dealing also with implementational aspects)
- J. Weickert: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.
  (monograph on continuous and discrete foundations, as well as on modelling aspects)
  Earlier version: Ph.D. thesis, available under
  `http://www.mia.uni-saarland.de/weickert/Papers/diss.ps.gz`
- G. Aubert, P. Kornprobst: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Second Edition, Springer, New York, 2006.
  (excellent book for the mathematically inclined reader)

Assignment P4 – Programming Work

Please download the required files from the webpage

`http://www.mia.uni-saarland.de/Teaching/ipcv07.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex04.tgz`.

Problem 1 (Morphological Operations) (10 + 4 points)

The programme `morphologyTemplate.c` has subroutines for erosion and dilation with a square as structuring element.

(a) Use these subroutines for completing the subroutines `closing`, `opening`, `white_top_hat`, `black_top_hat` and `selfdual_top_hat`, such that they perform the corresponding operations. Note that the index range of the image `u[i][j]` is given by `i=1,...,nx` and `j=1,...,ny`.

(b) With the completed programme, try to solve the following problems:
- remove windows and doors (`house.pgm`).
- create a university owl at night (`owl.pgm`).
- separate fabric structures and vessels from their background (`fabric.pgm, angiogram.pgm`).

If a greyscale image has a poor dynamics, one can normalise it with `xv`: Click on the `Windows` button, go to `Color Editor` and click on `Norm`.

Problem 2 (Morphological Enhancement) (3 + 3 points)

(a) Complete the subroutine `contrast_enhancement` such that it adds `factor` times the white top hat to the original image and subtracts `factor` times the black top hat. Thus, the variable `factor` serves as a parameters for contrast enhancement.

(b) Use this programme for enhancing the contrast in the images `mammogram.pgm`. Such X-ray images of the female breast have a poor contrast. From a medical point of view, it is important to recognise small bright calcifications and star-shaped structures. Apart from the contrast enhancement, use also a suitable top hat.

Assignment P4 (3)

Submission

Please remember that up to three people from the same tutorial group can work and submit their results together. For submitting the files rename the main directory Ex04 to Ex04_<your_name> and use the command

```
tar czvf Ex04_<your_name>.tgz Ex04_<your_name>
```

to pack the data. The directory that you pack and submit should at least contain the following files:

- the source code for morphologyTemplace.c including the code for the code for the five subroutines from Problem 1(a) and the subroutine from Problem 2(a).
- the four processed images from Problem 1(b) - to show that you solved the tasks
- the two processed images from Problem 2(b) - to compare contrast enhancement with a top hat.
- a text file README that contains the operators that you have used for the tasks in the problems 1(b) and 2(b) as well as information on all people working together for this assignment.

Please make sure that only your final version of the programmes and images are included.
Submit the file via e-mail to your tutor via the address:

```
ipcv-xx@mia.uni-saarland.de
```

where xx is either t1, t2, t3, t4, w1 or w2 depending on your tutorial group.

Deadline for submission: Tuesday, December 18, 10 am (before the lecture)