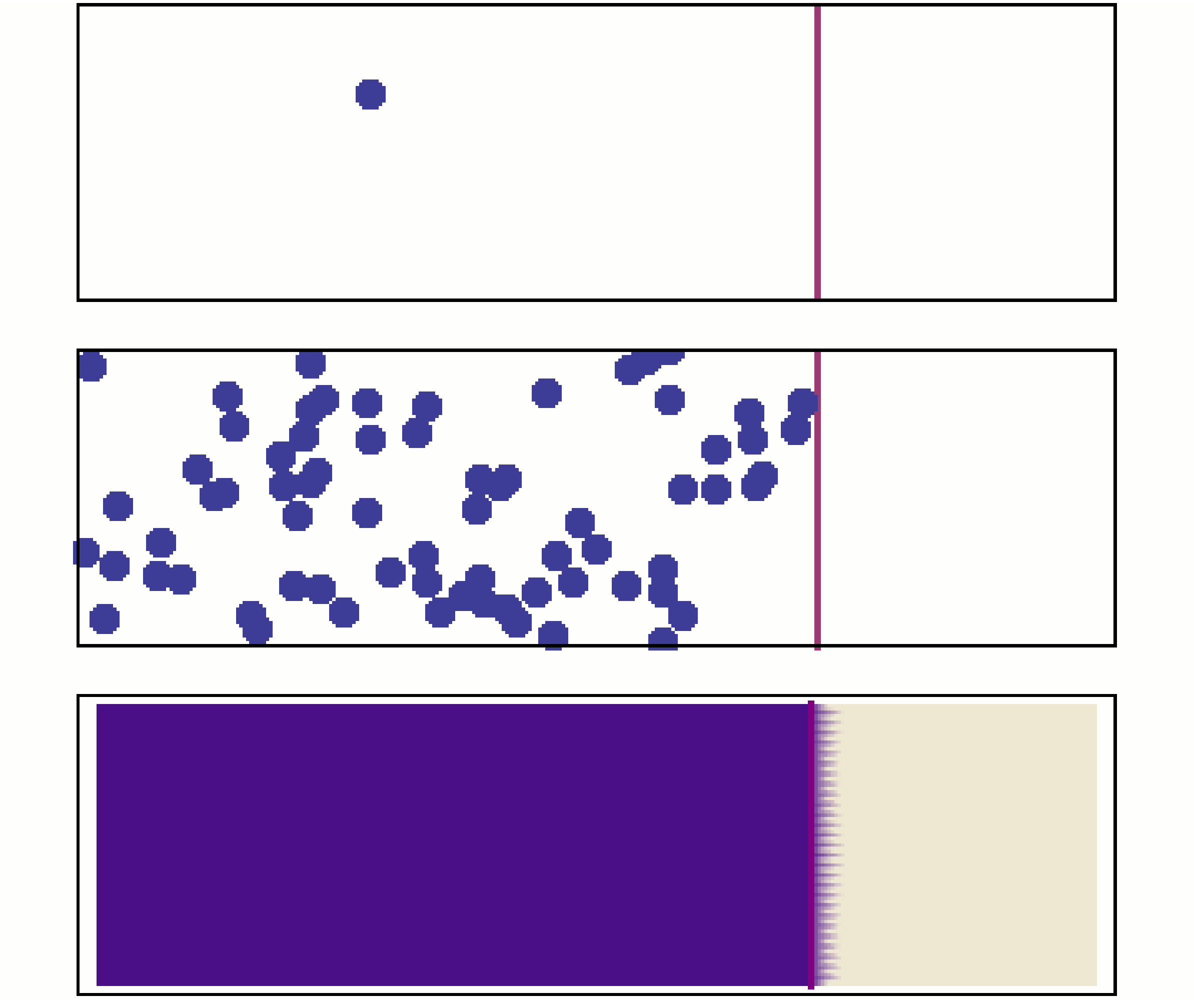
Image Processing

Diffusion Filters

Diffusion – Background

Motivation: a physical process – concentration balancing



Diffusion – Background

Concentration is a real-valued function in space, i.e. $u : \mathbb{R}^n \to \mathbb{R}$
For instance in physic it is often $u : \mathbb{R}^3 \to \mathbb{R}$

Spatial concentration gradient $\nabla u = (\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \ldots)$ leads to the
Flux $j : \mathbb{R}^n \to \mathbb{R}^n$ (vector field)

Fick's law: $j = -D \cdot \nabla u$

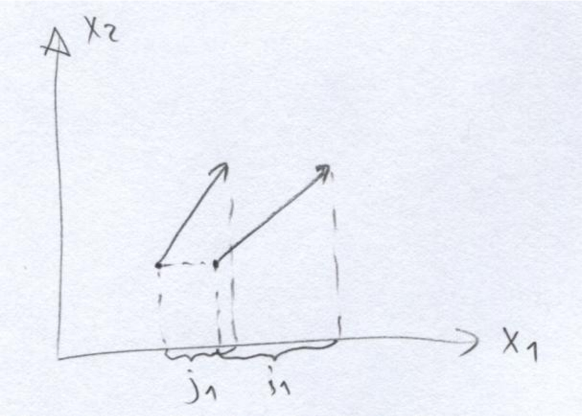
D is a positive definite symmetric matrix – **Diffusion Tensor**

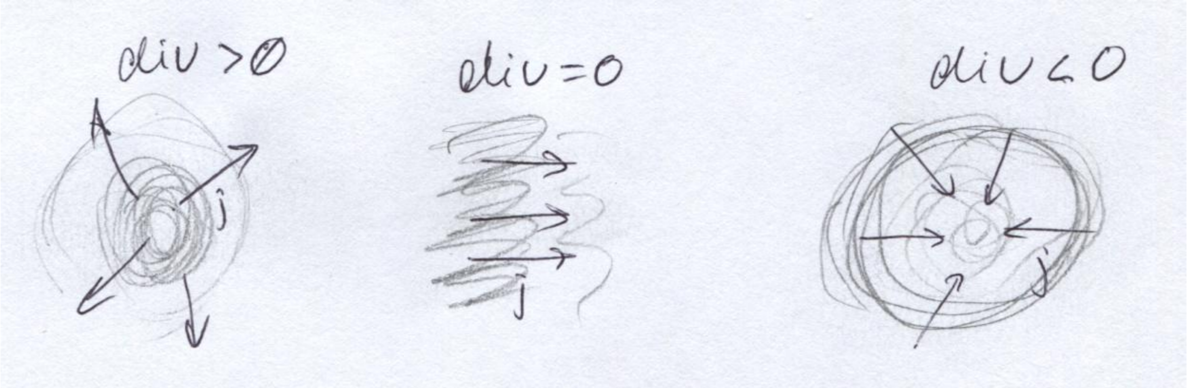
Diffusion – Background

It follows from the mass conservation (second Fick's law)

$$\frac{\partial u}{\partial t} = -\mathrm{div}\; j = \mathrm{div}(D \cdot \nabla u)$$

with **divergence** $j(x) = \frac{\partial j_1(x)}{\partial x_1} + \frac{\partial j_2(x)}{\partial x_2} + \dots$
(a real-valued function)





Diffusion for images – Idea

The image is interpreted as the initial concentration distribution

$$u(x, y, t = 0) = I(x, y)$$

The "image" is changed (in time) according to $\partial u / \partial t = \operatorname{div}(D \nabla u)$

The diffusion tensor D controls the process

Cases with respect do D :

is a scalar	→ isotropic
is a general tensor	→ anisotropic
independent on u	→ linear
dependent on u	→ non-linear

(all four combinations are possible)

Homogenous diffusion

Special case of the linear isotropic diffusion.

Diffusion tensor is a “constant”, i.e. $D = c \cdot \mathbb{1}$ ($\mathbb{1}$ is a unit matrix)

$$u(x,0) = I(x), \qquad \frac{\partial u}{\partial t} = \operatorname{div}(c \cdot \nabla u) = c \cdot \Delta u$$

with the **Laplace Operator** $\Delta u = \operatorname{div}(\nabla u) = \frac{\partial u^2}{\partial x \partial x} + \frac{\partial u^2}{\partial y \partial y}$

There exists the analytical solution (for $c = 1$):

$$u(x,t) = (G_{\sqrt{2t}} * I)(x)$$

i.e. the convolution of the image I
with the Gaussian of variance $\sigma = \sqrt{2t}$ → basically smoothing

Numerical Schemes

For homogenous diffusion as the example:

$$\frac{\partial u(x,u,t)}{\partial t} = \frac{\partial u(x,y,t)^2}{\partial x \partial x} + \frac{\partial u(x,y,t)^2}{\partial x \partial x}$$

Approximate derivatives (continuous) by **finite differences** (discrete)

$$\frac{\partial u(x,u,t)}{\partial t} = \frac{u(x,y,t+\tau) - u(x,y,t)}{\tau} + O(\tau)$$

$$\frac{\partial u(x,y,t)^2}{\partial x \partial x} = \frac{u(x+h,y,t) - 2u(x,y,t) + u(x-h,y,t)}{h^2} + O(h^2)$$

$$\frac{\partial u(x,y,t)^2}{\partial y \partial y} = \frac{u(x,y+h,t) - 2u(x,y,t) + u(x,y-h,t)}{h^2} + O(h^2)$$

τ is the step in time, h – spatial resolution.

$O(\cdot)$ are left out → approximation.

Numerical Schemes

All together:

$$u(x,y,t+\tau) = \left(1 - \frac{4\tau}{h^2}\right) u(x,y,t) +$$
$$+ \frac{\tau}{h^2}\Big(u(x+1,y,t) + u(x-1,y,t) + u(x,y+1,t) + u(x,y-1,t)\Big)$$

This was an **explicit** schema:

the new values are computed from the old ones directly

It is **stable** (converges) if all “weights” are non-negative, i.e. $\frac{\tau}{h^2} \leq \frac{1}{4}$

Numerical Schemes

Implicit Schema: Divergences in the **next** time step are used:

$$u(x,y,t+\tau) = \left(1 - \frac{4\tau}{h^2}\right) u(x,y,t) +$$

$$+ \frac{\tau}{h^2}\Big(u(x+1,y,t) + u(x-1,y,t) + u(x,y+1,t) + u(x,y-1,t)\Big)$$

becomes

$$u(x,y,t+\tau) = u(x,y,t) +$$

$$\frac{\tau}{h^2}\Big(u(x+1,y,t+\tau) + u(x-1,y,t+\tau) + u(x,y+1,t) + u(x,y-1,t+\tau) -$$

$$- 4u(x,y,t+\tau)\Big)$$

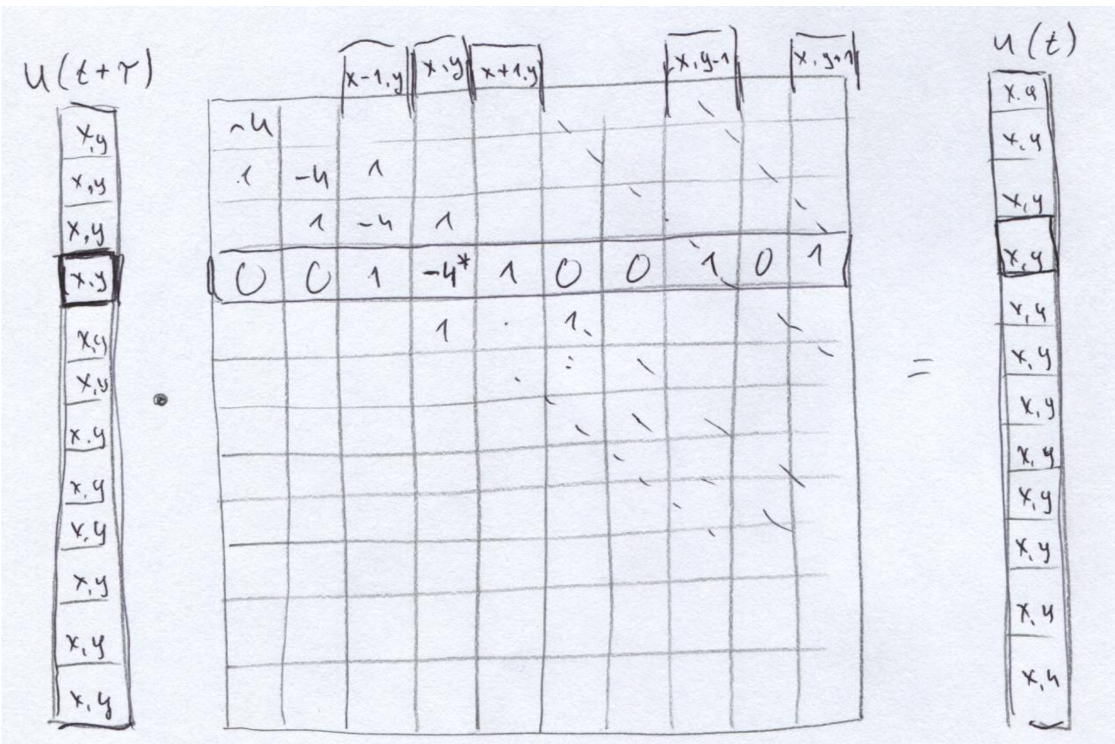
New values can not be computed from the old ones directly, because they depend on each other. However, they depend **linearly**

→ system of linear equations.

Numerical Schemes

$$u(x,y,t{+}\tau) = u(x,y,t) +$$
$$\frac{\tau}{h^2}\Big(u(x{+}1,y,t{+}\tau) + u(x{-}1,y,t{+}\tau) + u(x,y{+}1,t) + u(x,y{-}1,t{+}\tau) -$$
$$-\,4u(x,y,t{+}\tau)\Big)$$

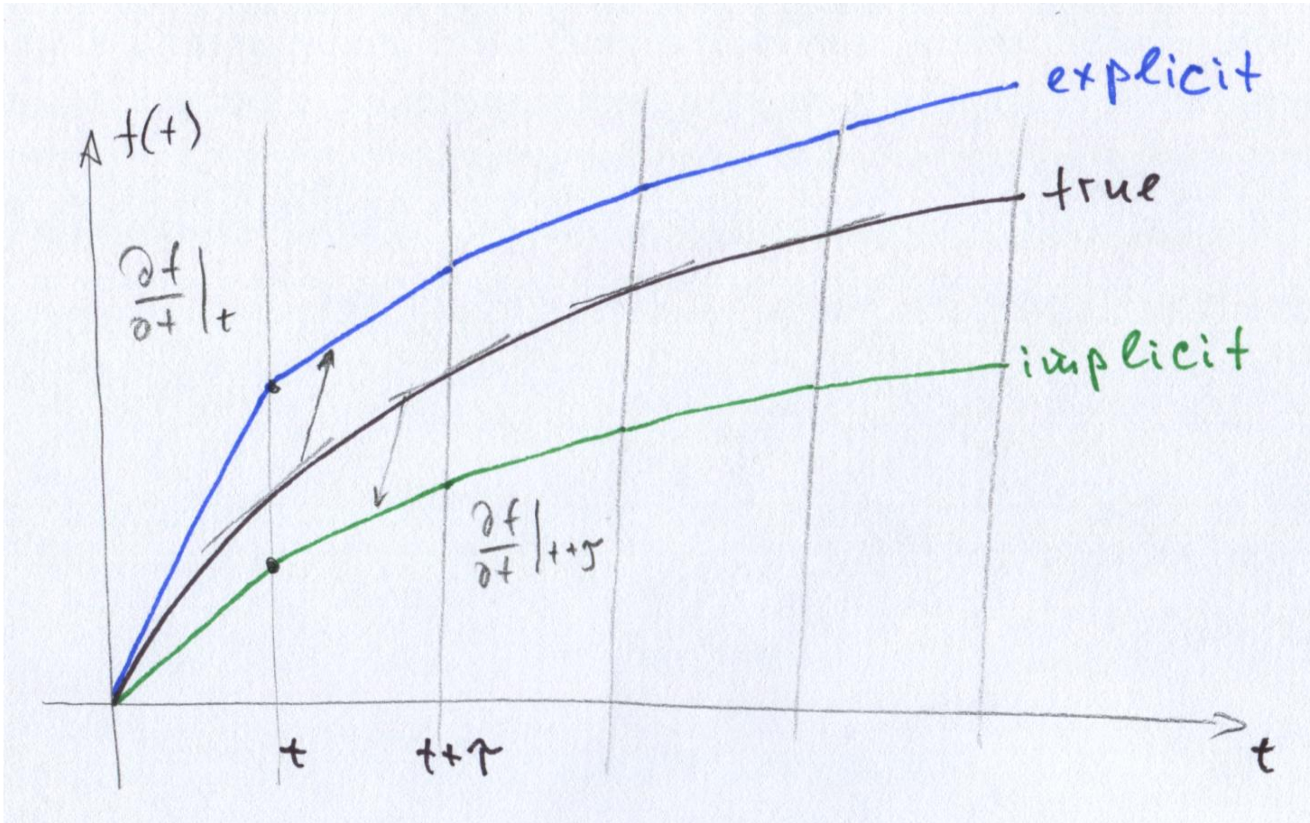
System of linear equation:



Huge, but **sparse**:
Special iterative methods
(Jakobi …)

Numerical Schemes – explicit vs. implicit

Stability (an oversimplified example):



Explicit: less stable, fast

Implicit: more stable, slow (solve a system at each time)

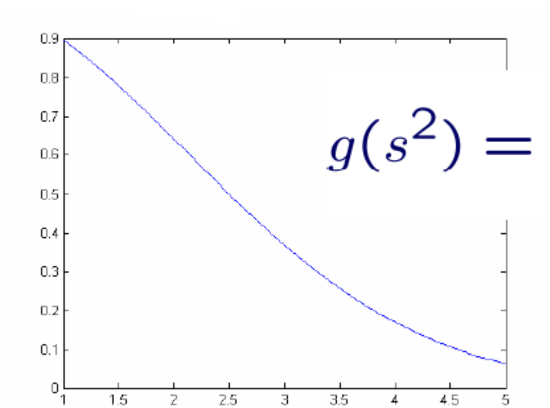
Linear isotropic diffusion

The Idea – smooth dependent on edge information

$$c \cdot \triangle u \equiv c(x, y, I) \cdot \triangle u$$

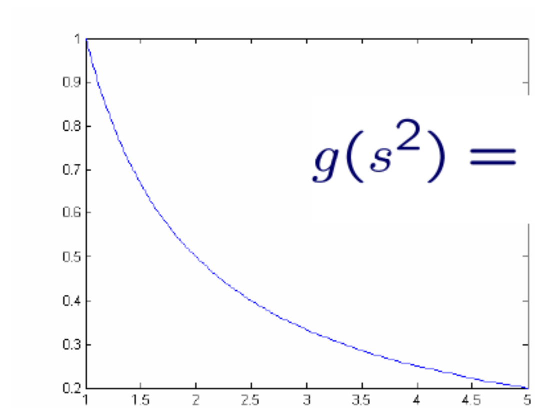
With a pre-computed $c(x, y, I)$

Very often $c(x, y, I) \;=\; g(|\nabla I(x, y)|^2)$ is a positive decreasing function (Diffusivity) of the squared length of image gradients



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

Perona-Malik



$$g(s^2) = \frac{1}{\sqrt{s^2}}$$

Total Variation

Non-linear isotropic diffusion

The idea – edges are more distinctive in the “de-noised” image

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|\nabla I|^2)\nabla u) \qquad \text{becomes} \qquad \frac{\partial u}{\partial t} = \operatorname{div}(g(|\nabla u|^2)\nabla u)$$

(the diffusion tensor depends on u)

A special case – TV-flow: $\partial u/\partial t = \operatorname{div}(\frac{\nabla u}{|\nabla u|})$

- No further contrast-dependent parameters
- Piecewise constant grey-value profiles (similar to segmentation)

Problem: ∞ at $|\nabla u| = 0$ → regularization $g(s^2) = \frac{1}{\sqrt{s^2+\varepsilon}}$

Implicit numerical schema leads to a system of **non-linear** equation.

Non-linear isotropic diffusion

Some examples:

Original

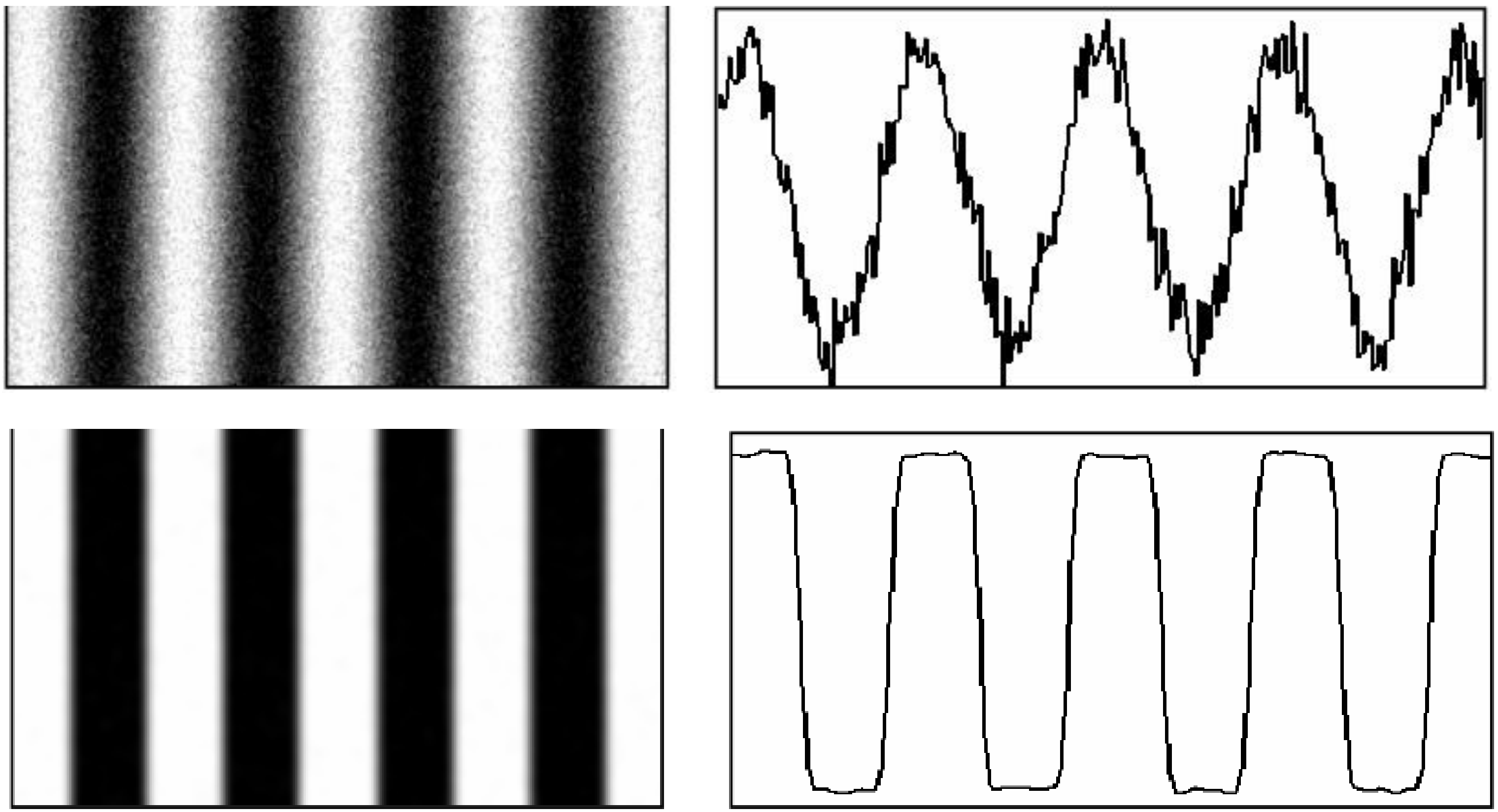
Gaussian smoothing

Non-linear diffusion

Shock-Filter

The Idea – dilation close to the local maximums and erosion close to the local minimums:

$$\frac{\partial u}{\partial t} = -\mathrm{sign}(\Delta u) \cdot |\nabla u|$$

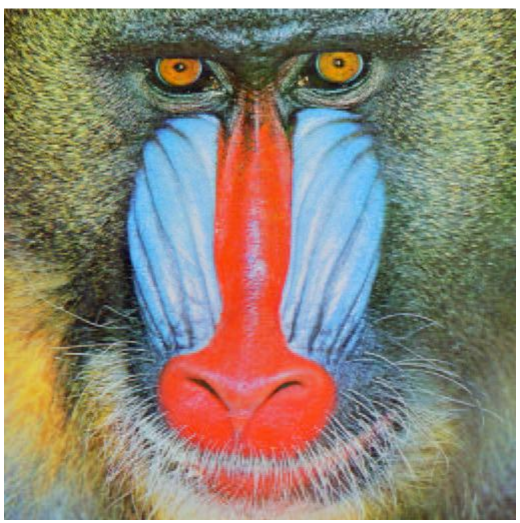
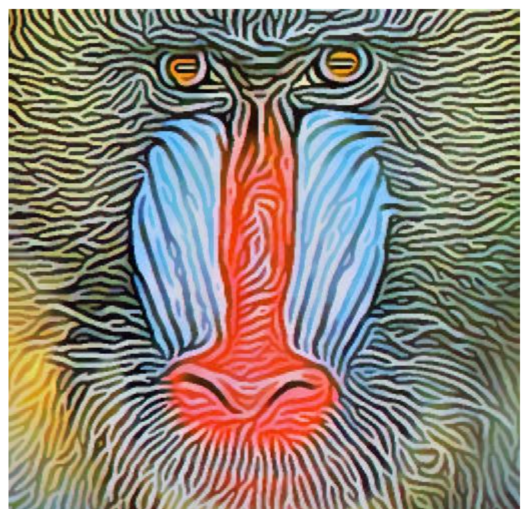


Joachim Weickert, *Coherence-Enhancing Shock Filters*, DAGM2003

Shock-Filter



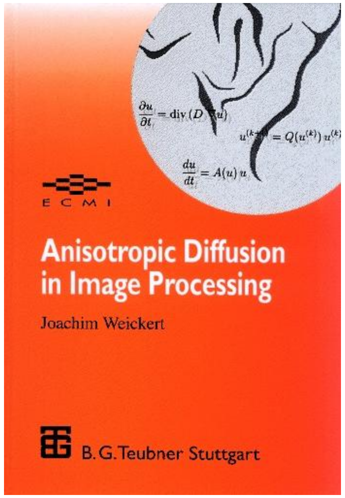
Original | Usual Shock-Filter | Coherence-Enhancing


← Different filter parameters →

Shock-Filter

Literature



Joachim Weickert: *Anisotropic Diffusion in Image Processing*
http://www.mia.uni-saarland.de/weickert/book.html

Further names:
Tomas Brox, Daniel Cremers, Andrés Bruhn …