ANISOTROPIC DIFFUSION FOR DENOISING AND EDGE DETECTION

Overview of some theory and results

Simon Lebastard

January 26, 2016

École des Ponts Paristech

SCALE-SPACE AND ISOTROPIC DIFFUSION

- Represent of scales of frequencies in the image
- Useful to find edges, without knowing the corresponding intensity gradient

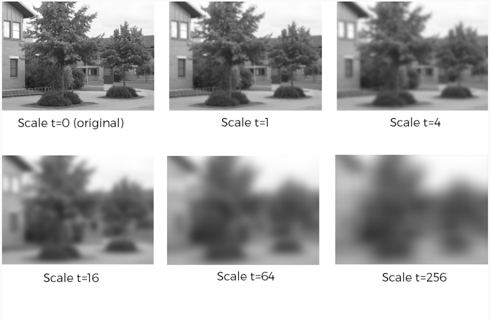


Figure: Scale-space at different scales

Diffusion problem (general):

$$I_t = \mathrm{div}(c(x, y, t)\nabla I) \tag{1}$$

$$I(x, y, t) = \frac{1}{\sqrt{4\pi ct}} \int I(x', y', 0) \exp^{-\frac{(x-x')^2+(y-y')^2}{4ct}} dx'dy' \tag{2}$$

which is equivalent to convolution of the image with the kernel:

$$G(x, y, t) = \frac{1}{\sqrt{4\pi ct}} \exp^{-\frac{x^2+y^2}{4ct}} \tag{3}$$

ISOTROPIC DIFFUSION: SCRIPT & RESULTS

Script is available at: `https://github.com/slebastard/TIVA_anisotropic_diffusion`

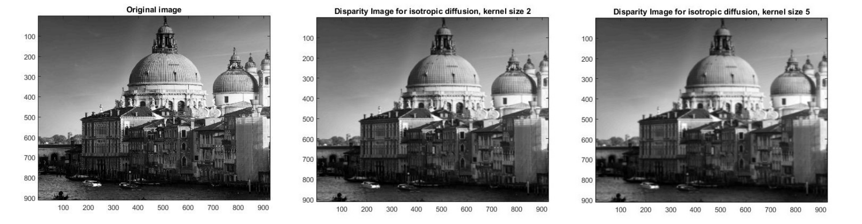


Figure: Result of isotropic diffusion for a constant diffusion function, for various kernel width

ISOTROPIC DIFFUSION: DRAWBACKS

- Edges disappear with noise
- Edges are shifted away from their original location, we would like to keep them where they are

ANISOTROPIC DIFFUSION: PERONA & MALIK

- Work on diffusion so that edges can be preserved and detected

$$I_t = \mathrm{div}(c(x, y, t)\nabla I) \tag{4}$$

- Work on diffusion so that edges can be preserved and detected
- The rest of the image should be blurred

$$I_t = \operatorname{div}(c(x, y, t)\nabla I) \tag{4}$$

Perona & Malik introduced two main classes of diffusion functions, respectively named quadratic and exponential:

$$c_\kappa(||\nabla I||) = \frac{1}{1 + (\frac{||\nabla I||}{\kappa})^2} \tag{5}$$

$$c_\kappa(||\nabla I||) = \exp((-\frac{||\nabla I||}{\kappa})^2) \tag{6}$$

INFLUENCE OF κ PARAMETER

$$c_\kappa(||\nabla I||) = \frac{1}{1 + (\frac{||\nabla I||}{\kappa})^2} \tag{7}$$

$$c_\kappa(||\nabla I||) = \exp((-\frac{||\nabla I||}{\kappa})^2) \tag{8}$$

Figure: Anisotropic filtering with the exponential diffusion function with a low κparameter($= 0.02$)

INFLUENCE OF κ PARAMETER

$$c_\kappa(||\nabla I||) = \frac{1}{1 + (\frac{||\nabla I||}{\kappa})^2} \tag{9}$$

$$c_\kappa(||\nabla I||) = \exp((-\frac{||\nabla I||}{\kappa})^2) \tag{10}$$

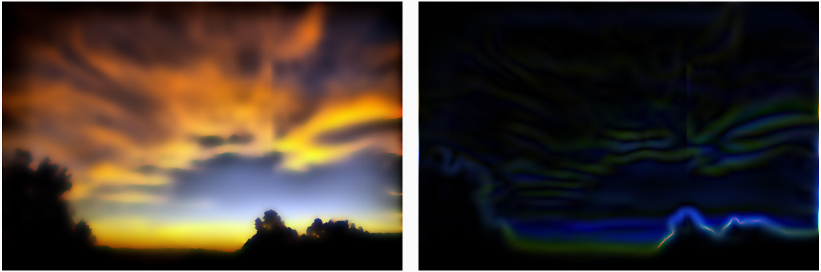


Figure: Anisotropic filtering with the exponential diffusion function with a high κparameter($= 0.2$)

Check out the script at: `https://github.com/slebastard/TIVA_anisotropic_diffusion`

Discrete version of the differential equation:

$$I_{i,j}^{t+1} = I_{i,j}^{t} + \lambda * (c_N \nabla_N I + c_S \nabla_S I + c_W \nabla_W I + c_E \nabla_E I) \tag{11}$$

where

$$c_{N,i,j}^{t} = g((||\nabla I||)_{i+\frac{1}{2},j}^{t}) \tag{12}$$

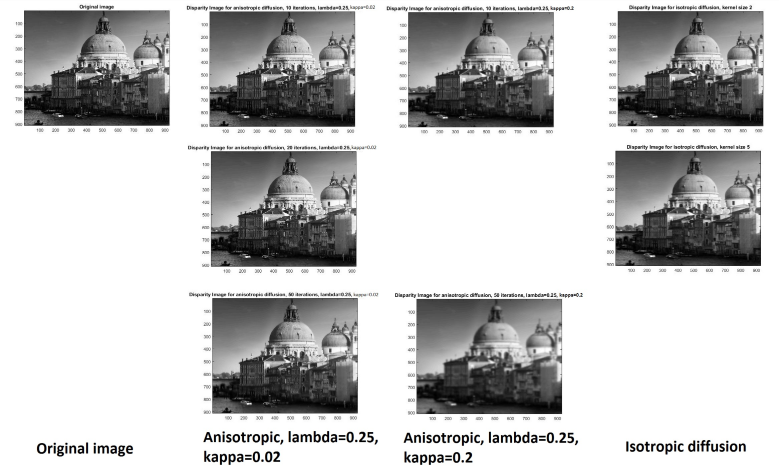


Figure: Anisotropic diffusion for various κ values and number of iterations

Former discrete scheme is equivalent to:

$$\frac{\partial I_r}{\partial t} = A \sum_{s \in N(r)} c(I_s - I_r)(I_s - I_r) \tag{13}$$

which is a gradient descent step for the following local energy:

$$E(I_r) = \sum_{s \in N(r)} c(I_s - I_r)(I_s - I_r) = \sum_{s \in N(r)} V(I_s, I_r) \tag{14}$$

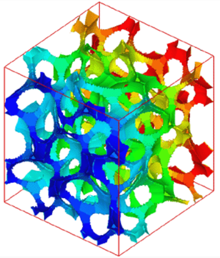
where $V(I_s, I_r) = V(I_s - I_r)$ is an increasing function of the distance between I_s and I_r

Bilateral filters explicitly compute mean values with pixels:

- at close range (for instance in a 3σ range)
- of a similar intensity

through convolution with a Gaussian kernel

Anisotropic diffusion implicitly does the same through minimizing an energy functional (only meaningful for an important number of time loops



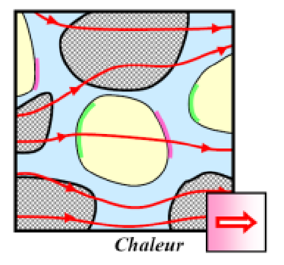


Figure: Heat diffusion inside a porous material

$c(x, y, t)$ depends on the layer, material or physical state that is at position (x, y) at time t

$$c = \frac{k}{\rho C_V} \tag{15}$$

Questions?