Image Processing and Pattern Recognition

Assignment 3 - Coherence Enhancing diffusion

November 14, 2016

Deadline: November 28, 2016 at 23:59h.
Submission: Upload report and implementation zipped (*[surname].zip*) to the TeachCenter.

1 Goal

In this exercise you will implement the so-called *coherence enhancing diffusion*, a variant of the celebrated[1] Perona-Malik nonlinear diffusion [2]. In their paper, Perona & Malik proposed a diffusivity function that stops at image edges to guide the diffusion process. The resulting denoising performance was very good, and some of todays state-of-the-art image denoising algorithms build on this idea.
The coherence enhancing diffusion guides the diffusion process along regions of similar gradient, and can be used to create artistic effects (see fig. 1).

(a) Original image (b) Perona-Malik diffusion (c) Coherence-enhancing diffusion

Figure 1: Perona-Malik and coherence-enhancing diffusion at stop time 100

2 Method

Let $u : \Omega \to \mathbb{R}$ be the image function, with $\Omega \subset \mathbb{R}^2$ the image domain. The basic diffusion equation is given as

$$\frac{\partial u}{\partial t} = \operatorname{div}(\nabla u) \tag{1}$$

This is a partial differential equation motivated by the physical behaviour of *e.g.* the diffusion of heat in a material. To increase the modeling accuracy, a so-called *diffusion tensor* is introduced into eq. (1)

$$\partial_t u = \operatorname{div}(D \nabla u), \tag{2}$$

where ∂_t is an abbreviation for the derivative *w.r.t.* time and D is the diffusion tensor. It models the rate of diffusion in different directions according to material properties. In our context, we can

[1] 10000+ citations!

use the diffusion tensor to guide the diffusion process along image edges. In the simplest case where D is the identity matrix, we speak of *isotropic* diffusion, if D is densely populated the method is called *anisotropic* diffusion.
The idea of coherence-enhancing diffusion is to guide the diffusion process along coherent regions, *i.e.* regions of similar structure (*e.g.* edge direction). To this end, we use the *structure tensor*. The structure tensor is a 2×2 matrix, that captures for every pixel information about the surrounding image structure. It has been used in the context of interest point detection [1], and is given by

$$S = G_\sigma * \begin{bmatrix} \tilde{u}_x^2 & \tilde{u}_x \tilde{u}_y \\ \tilde{u}_x \tilde{u}_y & \tilde{u}_y^2 \end{bmatrix}, \tag{3}$$

where subscripts denote directional derivatives. G_σ is a Gaussian smoothing kernel[2], and $\tilde{u}$ is a Gauss-filtered version of u. Note that the two Gaussian filter processes G_σ and $\tilde{u}$ cannot be combined into one due to the product structure of the elements of S.
The eigenvalues μ_1, μ_2 and eigenvectors ν_1, ν_2 of the structure tensor encode information about the local structure of the image, namely the direction and speed of grayvalue variation. In particular, we have

(a) μ_1, μ_2 both small: flat region

(b) $\mu_1 \gg \mu_2$ or vice-versa: strong grayvalue variation in one direction, *i.e.* image edge

(c) μ_1, μ_2 both large: strong grayvalue variation in two directions, *i.e.* image corner

In summary, the eigenvalues determine the magnitude of the grayvalue variation, the corresponding eigenvector gives the direction of the grayvalue variation. Case (b) has eigenvector ν_1 orthogonal to the edge (it points in the direction of the largest grayvalue change), whereas ν_2 gives the *coherence direction*, *i.e.* the direction along the edge (the eigenvectors are orthogonal). We will use this information to guide the diffusion process.
The diffusion tensor is calculated as

$$D = \begin{bmatrix} \nu_1 & \nu_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \nu_1^T \\ \nu_2^T \end{bmatrix} \tag{4}$$

Using a small constant $\alpha > 0$, λ_1 and λ_2 are given as

$$\begin{cases} \lambda_1 = \alpha \\ \lambda_2 = \alpha + (1-\alpha)(1 - g(|\mu_1 - \mu_2|)) \end{cases} \tag{5}$$

and $g(\cdot)$ is a function similar to the original edge stopping function of the Perona-Malik model

$$g(s) = e^{-\frac{s^2}{2\gamma^2}} \tag{6}$$

with a parameter γ.

2.1 Solution of the diffusion equation

To solve (2), we use a semi-implicit approach. In the discrete setting, an image of size $M \times N$ is represented as a vector $U \in \mathbb{R}^{MN}$. The time-derivative ∂u_t as well as the spatial gradient ∇u are approximated by finite differences, and we write eq. (2) as

$$\frac{U^{t+1} - U^t}{\tau} = -K^T D(U^t) K U^{t+1}, \tag{7}$$

with τ the discretisation in time (timestep). The matrix K is a discrete finite difference approximation of ∇, and we use the fact that $\operatorname{div} = -\nabla^T$. Solving (7) for U^{t+1} leads to a linear system

[2] The convolution of G_σ with the matrix is understood element-wise, *i.e.* applied to each of the elements individually

of equations

$$
\begin{aligned}
U^{t+1} + \tau K^T D(U^t) K U^{t+1} &= U^t \\
\left(\mathrm{id} + \tau K^T D(U^t) K\right) U^{t+1} &= U^t \\
U^{t+1} &= \left(\mathrm{id} + \tau K^T D(U^t) K\right)^{-1} U^t, \qquad (8)
\end{aligned}
$$

where id denotes the identity matrix.

3 Framework

Your task is to implement the coherence-enhancing diffusion described in the previous section. We provide a function to compute the discrete nabla operator (`make_derivatives_2D`). As a starting point, use the following parameters:

- To compute the structure tensor S, use $\sigma = 1.5$ for the filter G_σ, and $\sigma = 0.7$ for computing $\tilde{u}$.
- $\alpha = 0.0005$
- $\gamma = 0.0001$
- $\tau = 5$
- Diffusion end time: 100

To solve eq. (8) efficiently, use the function `scipy.sparse.linalg.spsolve`.
Use your own images and experiment with the parameters! Can you describe the effects controlled by α and γ?
Try a different discretization of the nabla operator (`make_derivatives_hp_sym_2D`). What is the corresponding filter kernel that is implemented by this discretization? How does it change the result?
Include your findings in the report!

References

[1] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[2] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, July 1990.