Medical Image Analysis and Processing

Image Noise Filtering

Anisotropic Diffusion Filter

Emad Fatemizadeh

Distance/online Course: Session 13

Date: 11 April 2021, 22th Farvardin 1400

Contents

Mathematics of Diffusion Filter

- Definition:
- $\boldsymbol{x} = [x_1 \ \ x_2]^t$: pixel coordination vector, t: transpose operator
- $u(x_1, x_2) = u(\boldsymbol{x})$: Gray Level image
- Partial derivative:

$$u_{x_1} = \frac{\partial u}{\partial x_1},\ u_{x_2} = \frac{\partial u}{\partial x_2},\ u_{x_1x_1} = \frac{\partial^2 u}{\partial x_1^2},\ u_{x_2x_2} = \frac{\partial^2 u}{\partial x_2^2},\ u_{x_1x_2} = \frac{\partial^2 u}{\partial x_1 \partial x_2}$$

- Gradient vector:

$$\nabla u(\boldsymbol{x}) = \left[u_{x_1} \ \ u_{x_2}\right]^t$$

Mathematics of Diffusion Filter

› Definition:

› Unit normal vector:

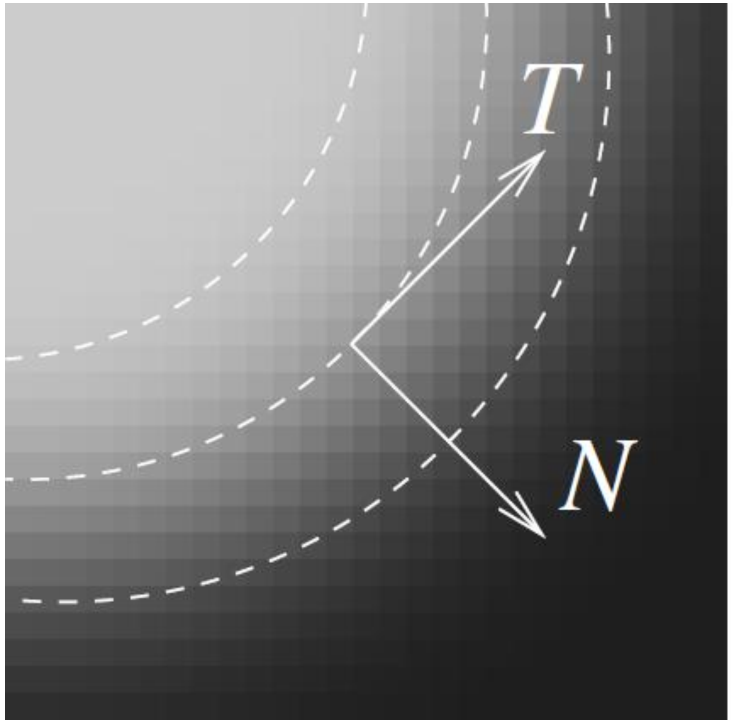
$$\vec{N}(\boldsymbol{x}) = \frac{\nabla u(\boldsymbol{x})}{|\nabla u(\boldsymbol{x})|} = \frac{1}{|\nabla u(\boldsymbol{x})|}\begin{bmatrix} u_{x_1} & u_{x_2} \end{bmatrix}^t$$

› Unit tangent vector:

$$\vec{T}(\boldsymbol{x}) = \frac{1}{|\nabla u(\boldsymbol{x})|}\begin{bmatrix} u_{x_2} & -u_{x_1} \end{bmatrix}^t \text{ or } \frac{1}{|\nabla u(\boldsymbol{x})|}\begin{bmatrix} -u_{x_2} & u_{x_1} \end{bmatrix}^t$$

Mathematics of Diffusion Filter

- Definition:
- *Isophote* curve (Level Set): Curves along which the intensity is constant.
- Unit normal vector: $\vec{N}(\boldsymbol{x})$
- Unit tangent vector: $\vec{T}(\boldsymbol{x})$



Mathematics of Diffusion Filter

- Definition:
- It can be shown:

$$u_{TT} = T^{t}H(u)T = \frac{1}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_2x_2} - 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2\, u_{x_1x_1}\right)$$

$$u_{NN} = N^{t}H(u)N = \frac{1}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$

- where $H(u)$ is Hessian of u:

$$H(u) = \begin{bmatrix} u_{x_1x_1} & u_{x_1x_2} \\ u_{x_1x_2} & u_{x_2x_2} \end{bmatrix}$$

Mathematics of Diffusion Filter

› Let's start with the following problem:

› $\frac{\partial u}{\partial t} = div\,(g(|\nabla u|^2)\nabla u), \quad on\ \Omega \times]0, T]$

› $u(x_1, x_2, t = 0) = u_0(x_1, x_2), \qquad \left.\frac{\partial u}{\partial \vec{n}}\right|_{\partial\Omega} = 0$

$$div(g(|\nabla u|^2)\nabla u) = \frac{\partial}{\partial x_1}\left(g(|\nabla u|^2)u_{x_1}\right) + \frac{\partial}{\partial x_2}\left(g(|\nabla u|^2)u_{x_2}\right)$$

$$= 2g'(|\nabla u|^2)\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$

$$+g(|\nabla u|^2)\left(u_{x_1x_1} + u_{x_2x_2}\right)$$

Mathematics of Diffusion Filter

› A simple manipulation:

$$div(g(|\nabla u|^2)\nabla u) = \frac{\partial}{\partial x_1}\left(g(|\nabla u|^2)u_{x_1}\right) + \frac{\partial}{\partial x_2}\left(g(|\nabla u|^2)u_{x_2}\right)$$

$$= 2g'(|\nabla u|^2)\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$

$$+ \frac{g(|\nabla u|^2)}{\left(u_{x_1}^2 + u_{x_2}^2\right)}\left(u_{x_1}^2 + u_{x_2}^2\right)\left(u_{x_1x_1} + u_{x_2x_2}\right)$$

$$= 2g'(|\nabla u|^2)\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$

$$+ \frac{g(|\nabla u|^2)}{\left(u_{x_1}^2 + u_{x_2}^2\right)}\left(u_{x_1}^2 u_{x_1x_1} + u_{x_1}^2 u_{x_2x_2} + u_{x_2}^2 u_{x_1x_1} + u_{x_2}^2 u_{x_2x_2}\right)$$

Mathematics of Diffusion Filter

› Define: $b(s) = 2sg'(s) + g(s) \Rightarrow 2g'(s) = \frac{b(s)-g(s)}{s}$

$$div(g(|\nabla u|^2)\nabla u) =$$
$$\frac{b(|\nabla u|^2) - g(|\nabla u|^2)}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$
$$+ \frac{g(|\nabla u|^2)}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_1x_1} + u_{x_1}^2 u_{x_2x_2} + u_{x_2}^2 u_{x_1x_1} + u_{x_2}^2 u_{x_2x_2}\right)$$

Mathematics of Diffusion Filter

› Some simplification and recall from slide #30 (u_{NN} and u_{TT})

$$div(g(|\nabla u|^2)\nabla u) =$$

$$\frac{g(|\nabla u|^2)}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_2x_2} - 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_1x_1}\right)$$

$$+\frac{b(|\nabla u|^2)}{|\nabla u|^2}\left(u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2}\right)$$

$$\frac{\partial u}{\partial t} = div(g(|\nabla u|^2)\nabla u) = g(|\nabla u|^2)u_{TT} + b(|\nabla u|^2)u_{NN}$$

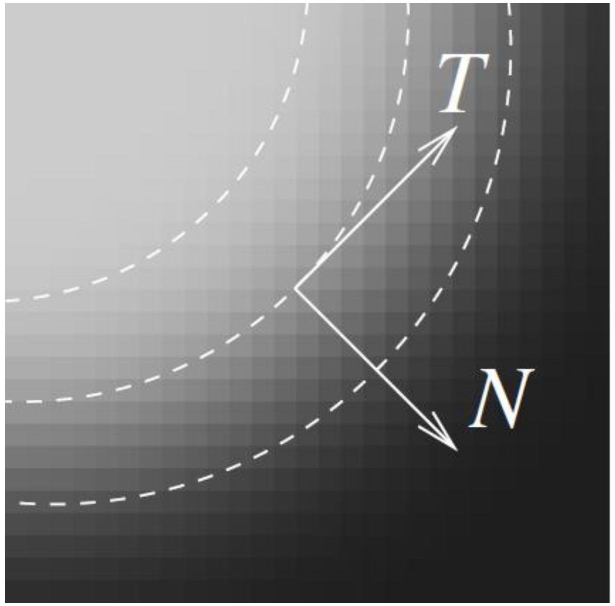
Mathematics of Diffusion Filter

- How to select diffusion coefficient, $g(s)$:

$$div(g(|\nabla u|^2)\nabla u) = g(|\nabla u|^2)u_{TT} + b(|\nabla u|^2)u_{NN}$$

- Unfortunately, $g(s)$ and $b(s)$ are coupled together:

$$b(s) = 2sg'(s) + g(s)$$



Mathematics of Diffusion Filter

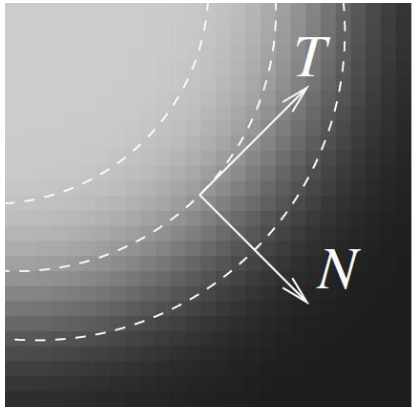
› How to select diffusion coefficient, $g(s)$:

$$\frac{\partial u}{\partial t} = div(g(|\nabla u|^2)\nabla u) = g(|\nabla u|^2)u_{TT} + b(|\nabla u|^2)u_{NN}$$

› $b(s) = 2sg'(s) + g(s)$

› Weak variation condition (low gradient regions):

smoothing the same in all directions



› Thus:

$$\lim_{s\to 0} b(s) = \lim_{s\to 0} g(s) > 0$$

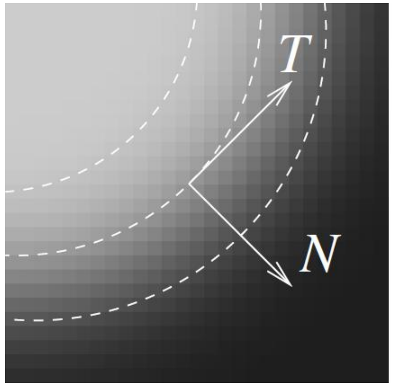
Mathematics of Diffusion Filter

- How to select diffusion coefficient, $g(s)$:

$$\frac{\partial u}{\partial t} = div(g(|\nabla u|^2)\nabla u) = g(|\nabla u|^2)u_{TT} + b(|\nabla u|^2)u_{NN}$$

- $b(s) = 2sg'(s) + g(s)$
- Strong variation condition (high gradient regions):
- Thus:

$$\begin{cases} \lim_{s\to\infty} b(s) = 0 \\ \lim_{s\to\infty} g(s) = \beta \end{cases} \Rightarrow \lim_{s\to\infty} \frac{b(s)}{g(s)} = 0 \Rightarrow \lim_{s\to\infty} \frac{sg'(s)}{g(s)} = -\frac{1}{2}$$



Mathematics of Diffusion Filter

› How to select diffusion coefficient, $g(s)$:

› Design Criteria:

- $b(s) > 0$, positive diffusion
- $g(0) = 1$, "weak condition"
- $g(s) \approx \frac{1}{\sqrt{s}},\ \ s \to \infty$, "strong condition"
- $b(s) = 2sg'(s) + g(s)$

› Example:

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

Mathematics of Diffusion Filter

- Conclusion:

  We need to de-couple $g(s)$ and $b(s)$

- Solution:

  Anisotropic diffusion, replace $g(s)$ by diffusion tensor, D

Nonlinear Anisotropic Diffusion Filtering

› Idea: Anisotropic diffusion (direction-dependent diffusion)

› Consider anisotropic diffusion equation:

$$\frac{\partial u}{\partial t} = div(D\nabla u)$$

› If D depends on (∇u) (or ∇u_σ) then it can be designed such that fulfilled desired properties, (preserving edges while reducing noise)

Nonlinear Anisotropic Diffusion Filtering

- Basics: Recall that D is symmetric *pdm*, its eigen-decomposition is:

$$D = \lambda_1 v_1 v_1^t + \lambda_2 v_2 v_2^t$$

- where $\{\lambda_i\}_{i=1}^2$ are positive eigenvalues, $\{v_i\}_{i=1}^2$ are orthogonal eigenvectors

$$\frac{\partial u}{\partial t} = div(D\nabla u)$$

$$\frac{\partial u}{\partial t} = div\Big((\lambda_1 v_1 v_1^t + \lambda_2 v_2 v_2^t)\nabla u\Big)$$

$$\frac{\partial u}{\partial t} = div(\lambda_1 v_1 [v_1^t \nabla u] + \lambda_2 v_2 [v_2^t \nabla u])$$

- $[v_1^t \nabla u]$ and $[v_2^t \nabla u]$ are scalar
- Two directions ($\{v_i\}_{i=1}^2$) with independent weights $\{\lambda_i\}_{i=1}^2$ for diffusion!

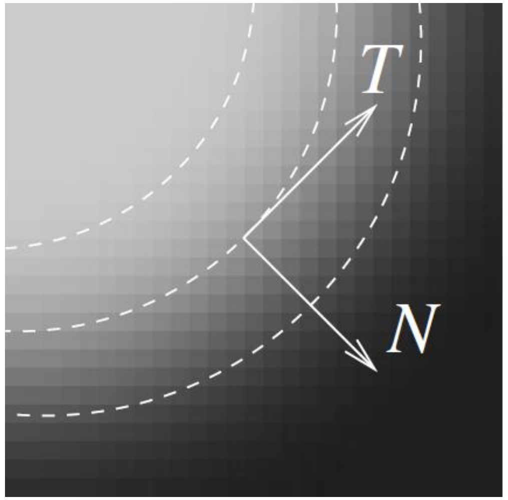
Nonlinear Anisotropic Diffusion Filtering

› Design idea for D:

$$\frac{\partial u}{\partial t} = div(\lambda_1 v_1 [v_1^t \nabla u] + \lambda_2 v_2 [v_2^t \nabla u])$$

› Design Criteria:

- $v_1 \parallel \nabla u$ or $v_1 \parallel \vec{N}$
- $v_2 \perp \nabla u$ or $v_2 \parallel \vec{T}$
- $\lambda_1 < \lambda_2$



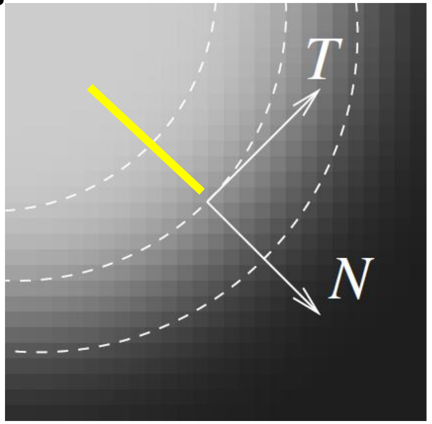
Nonlinear Anisotropic Diffusion Filtering

- Lets check $J_0 = (\nabla u)(\nabla u)^t$ which is positive semi-definite:

$$J_0 = (\nabla u)(\nabla u)^t = (\nabla u \otimes \nabla u) = \begin{pmatrix} u_{x_1}^2 & u_{x_1} u_{x_2} \\ u_{x_1} u_{x_2} & u_{x_2}^2 \end{pmatrix}$$

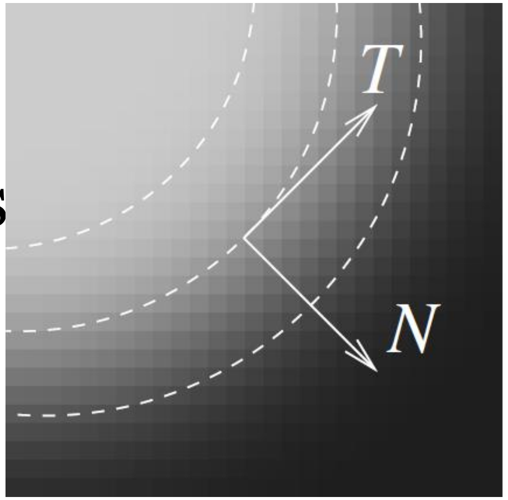
where $\otimes$ is (tensor/outer/matrix) product operator

- This is an orientation (not direction) descriptor.
- Identify gradients which differ only by their sign and share the same orientation, but point in opposite directions.



Nonlinear Anisotropic Diffusion Filtering

- Its eigen-decomposition ($J_0 v = \mu v$):
- $v_1 \parallel \nabla u$ and $\mu_1 = |\nabla u|^2$
- $v_2 \perp \nabla u$ and $\mu_2 = 0$
- But we need $\lambda_1 < \lambda_2$, do not worry!
- eigenvalues $|\nabla u|^2$ and 0 give just the contrast (the squared gradient) in the eigen-directions



Diffusion Tensor Design

› Smooth u with Gaussian kernel $k_\sigma(\boldsymbol{x})$ to avoid false detection due to noise:

$$u_\sigma(\boldsymbol{x}) = k_\sigma(\boldsymbol{x}) * u(\boldsymbol{x})$$

› Calculate the *structure tensor*, which summarizes the predominant directions of the gradient in a specified neighborhood of a point (using Gaussian kernel $k_\rho(\boldsymbol{x})$), and the degree to which those directions are coherent:

$$J_\rho(\boldsymbol{x}) = \begin{pmatrix} k_\rho(\boldsymbol{x}) * \left(u_{\sigma x_1}^2\right) & k_\rho(\boldsymbol{x}) * \left(u_{\sigma x_1} u_{\sigma x_2}\right) \\ k_\rho(\boldsymbol{x}) * \left(u_{\sigma x_1} u_{\sigma x_2}\right) & k_\rho(\boldsymbol{x}) * \left(u_{\sigma x_2}^2\right) \end{pmatrix}$$

$$= k_\rho(\boldsymbol{x}) * \left(\nabla u_\sigma(\boldsymbol{x}) \otimes \nabla u_\sigma(\boldsymbol{x})\right)$$

› It is related to covariance matrix of ∇u_σ distribution!

Diffusion Tensor Design

- *Structure tensor* Interpretation ($J_\rho(\boldsymbol{x})v = \mu v$):
- $\mu_1 > \mu_2$: v_1 is maximally aligned with the average of gradient within the window.
- $\mu_1 > 0, \mu_2 = 0$: u within the window varies along the v_1 and constant along v_2 (linelike)
- $\mu_1 = \mu_2 > 0$: the gradient in the window has no predominant direction (isotropic)
- $\mu_1 = \mu_2 = 0$: u is constant
- Note that: $v_1 \sim \| \left(\nabla u = \vec{N}\right)$ and $v_2 \sim \| \vec{T}$
- Degree of anisotropy of the gradient (relative discrepancy): $\left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2$

Diffusion Tensor Design

- *Structure tensor* Interpretation (isotropic vs linelike orientation):

Diffusion Tensor Design

- Final Problem:
- $\frac{\partial u}{\partial t} = div\left(D\left(J_\rho(\nabla u_\sigma)\right)\nabla u\right) \quad on\ ]0,T] \times \Omega$
- $u(\boldsymbol{x}, t = 0) = f(\boldsymbol{x})$
- $\left\langle D\left(J_\rho(\nabla u_\sigma)\right)\nabla u, \vec{N}\right\rangle = 0 \qquad on\ \ ]0,T] \times \partial\Omega$
- Construct D such that:
  - Same eigenvector as $J_\rho(\boldsymbol{x}),\ J_\rho(\boldsymbol{x})v = \mu v$
  - Choose eigenvalues using orientation information such that $\lambda_1 < \lambda_2$
- $D = \lambda_1 v_1 v_1^t + \lambda_2 v_2 v_2^t$

Diffusion Tensor Design

- Two popular choices:
- Edge-enhancing anisotropic diffusion:

$$\lambda_1 = \begin{cases} 1 & \mu_1 = 0 \\ 1 - exp\left(\dfrac{-3.315}{\mu_1^4}\right) & \mu_1 \neq 0 \end{cases}$$

$$\lambda_2 = 1$$

Diffusion Tensor Design

- Two popular choices:
- Coherence-enhancing anisotropic diffusion:

$$\lambda_1 = \alpha \qquad\qquad 0 < \alpha \ll 1$$

$$\lambda_2 = \begin{cases} \alpha & \mu_1 = \mu_2 \\ \alpha + (1-\alpha) exp\left(\dfrac{-1}{(\mu_1 - \mu_2)^2}\right) & \mu_1 \neq \mu_2 \end{cases}$$

$$\lambda_1 = 0$$

$$\lambda_2 = \frac{\eta |\nabla u_\sigma|^2}{1 + (|\nabla u_\sigma| / \sigma)^2}, \qquad \eta > 0$$

The End

› AnY QuEsTiOn?