Vector Valued Image Regularization with PDE's: A Common Framework for Different Applications

CVPR 2003 Best Student Paper Award

Definitions

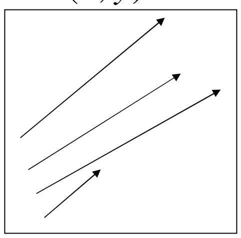
- **Scalar images:** Intensity images
- **Vector valued images:** RGB, HSV, YIQ…
- **Regularization:** Finding approximate solution of ill-posed problems.
- Let $\mathbf{G}$ be 2x2 matrix $\mathbf{G} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$$trace(\mathbf{G}) = \sum_{i=1}^{2} g_{i,i}$$

Divergence and Curl

vector function

$\vec{A}(x, y)$



Divergence defines expansion or contraction per unit volume

$$div(\vec{A}) = \nabla \cdot \vec{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y}$$

Vectors are obtained from image gradients ∇I

$$\Delta I = div(\nabla I) = div\begin{pmatrix} I_x \\ I_y \end{pmatrix} = I_{xx} + I_{yy}$$

Curl of a vector field is defined by cross product between vectors

Structure Tensor

$$\mathbf{G} = \sum_{i=1}^{n} \nabla I_i \nabla I_i^T$$

One dimensional image (gray level)

$$\mathbf{G}_{gray} = \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

$$I_x = I(x, y) - I(x-1, y)$$
$$I_y = I(x, y) - I(x, y-1)$$

Structure Tensor

$$\mathbf{G}_{color} = \nabla R \nabla R^T + \nabla G \nabla G^T + \nabla B \nabla B^T$$

$$\mathbf{G}_{color} = \begin{bmatrix} R_x R_x & R_x R_y \\ R_x R_y & R_y R_y \end{bmatrix} + \begin{bmatrix} G_x G_x & G_x G_y \\ G_x G_y & G_y G_y \end{bmatrix} + \begin{bmatrix} B_x B_x & B_x B_y \\ B_x B_y & B_y B_y \end{bmatrix}$$

Eigenvalue and eigenvectors of $\mathbf{G}$ (spectral elements)

$$\mathbf{G}_{color} \begin{bmatrix} \boldsymbol{\varphi}_+^1 \\ \boldsymbol{\varphi}_+^2 \end{bmatrix} = \underbrace{\boldsymbol{\lambda}_+}_{eigenvalue} \underbrace{\begin{bmatrix} \boldsymbol{\varphi}_+^1 \\ \boldsymbol{\varphi}_+^2 \end{bmatrix}}_{eigenvector}$$

Can be computed using Matlab functions

More Insight (hessian&tensor)

- Hessian of image I

$$\mathbf{H} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

- Laplacian of I

$$\Delta I = \mathrm{div}(\nabla I) = trace(\mathbf{H}) = I_{xx} + I_{yy}$$

- **Tensor:** Just a matrix. They use symmetric semi-positive definite matrix

Image Regularization

- **Functional minimization:** Euler-Lagrange equations

$$\underbrace{\min}_{I:\Omega\to R^n} \int_\Omega \phi(N(I))d\Omega$$

- **Divergence expression:** Diffusion of pixel values from high to low concentration

$$\tfrac{\partial I}{\partial t} = div(D\nabla I_i)$$

- **Oriented Laplacians:** Image smoothing in eigenvector directions weighted by corresponding eigenvalue.

$$\tfrac{\partial I}{\partial t} = c_1 I_{uu} + c_2 I_{vv}$$

Variational Problem

- Define variational problem:

$$E(I) = \int_\Omega F(I_x, I_y, \dot{I}_x, \dot{I}_y) d\Omega$$

$$\frac{\partial F}{\partial I} - \frac{d}{dx}\frac{\partial F}{\partial I_x} - \frac{d}{dy}\frac{\partial F}{\partial I_y} = 0$$

Euler-Lagrange equation

- Solution using classic iterative method:

$$\begin{cases} I_{t=0} = I_0 \\ \dfrac{\partial I(t)}{\partial t} = I(t) - I(t-1) = -\left(\dfrac{\partial F}{\partial I}(t-1) - \dfrac{d}{dx}\dfrac{\partial F}{\partial I_x}(t-1) - \dfrac{d}{dy}\dfrac{\partial F}{\partial I_y}(t-1)\right) \end{cases}$$

Variational Problem Horn&Schunck Example

$$f_x u + f_y v + f_t = 0 \qquad u = \tfrac{dx}{dt} \qquad v = \tfrac{dy}{dt}$$

$$E(u,v,\dot{u},\dot{v}) = \iint \left(f_x u + f_y v + f_t\right)^2 + \lambda\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dxdy$$

$$\frac{\partial E}{\partial u} - \frac{\partial}{\partial x}\frac{\partial E}{\partial u_x} - -\frac{\partial}{\partial y}\frac{\partial E}{\partial u_y} \Rightarrow \iint 2\left(f_x u + f_y v + f_t\right) f_x + 2\lambda \underbrace{\left(u_{xx} + u_{yy}\right)}_{u - u_{avg}} dxdy = 0$$

$$\frac{\partial E}{\partial v} - \frac{\partial}{\partial x}\frac{\partial E}{\partial v_x} - -\frac{\partial}{\partial y}\frac{\partial E}{\partial v_y} \Rightarrow \iint 2\left(f_x u + f_y v + f_t\right) f_y + 2\lambda \underbrace{\left(v_{xx} + v_{yy}\right)}_{v - v_{avg}} dxdy = 0$$

$$u^t = u_{avg}^{t-1} - \frac{f_x}{\lambda}\left(f_x u + f_y v + f_t\right) \qquad v^t = v_{avg}^{t-1} - \frac{f_y}{\lambda}\left(f_x u + f_y v + f_t\right)$$

Defining Energy Functional

- Functional: (minimization based)

$$E(I) = \underbrace{\min}_{I:\Omega \to R^3} \int_\Omega \psi(\lambda_+, \lambda_-) d\Omega$$

increasing function
$\forall b > a \ , \ f(b) > f(a)$

- Euler-Lagrange is given by (relation to divergence based methods)

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\begin{pmatrix} \frac{\partial \psi}{\partial I_{i_x}} \\ \frac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} = \operatorname{div}(D \nabla I_i)$$

$$D = 2\frac{\partial \psi}{\partial \lambda_+}\theta_+\theta_+^T + 2\frac{\partial \psi}{\partial \lambda_+}\theta_-\theta_-^T$$

D Matrix

$$D = 2\frac{\partial\psi}{\partial\lambda_+}\theta_+\theta_+^T + 2\frac{\partial\psi}{\partial\lambda_+}\theta_-\theta_-^T$$ is 2x2.

Eigenvalues and eigenvectors of D are

$$\lambda_1 = 2\frac{\partial\psi}{\partial\lambda_+} \qquad \lambda_2 = 2\frac{\partial\psi}{\partial\lambda_-}$$

$$\mathbf{u}_1 = \theta_+ \qquad \mathbf{u}_2 = \theta_-$$

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\left(\left[2\frac{\partial\psi}{\partial\lambda_+}\theta_+\theta_+^T + 2\frac{\partial\psi}{\partial\lambda_+}\theta_-\theta_-^T\right]\nabla I_i\right)$$

Old School Laplacian Approach

$$\frac{\partial I}{\partial t} = c_1 I_{uu} + c_2 I_{vv}$$

$I^t - I^{t-1}$

Second order image derivatives in directions of eigenvectors of $\mathbf{G}$ at that point (structure tensor) (Edge preserving smoothing)

$$\frac{\partial I_i}{\partial t} = trace(T\mathbf{H}_i)$$

$$T = c_1\mathbf{u}_1\mathbf{u}_1^T + c_2\mathbf{v}_2\mathbf{v}_2^T$$

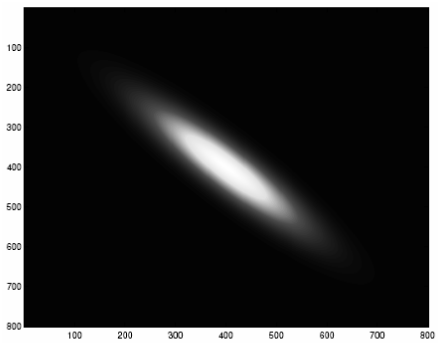
Solution of this PDE can be given by:

$$I_{i_{(t)}} = I_{i_{(t=0)}} * G^{(\mathbf{T},t)}$$

$$G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t}\exp\left(-\frac{\mathbf{x}\mathbf{T}^{-1}\mathbf{x}}{4t}\right)$$

Laplacian Example (constant T)

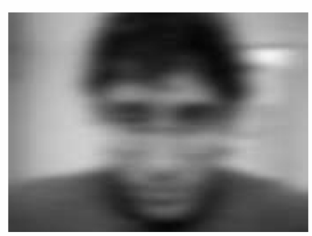
Constant T, such that direction (eigenvectors are same)

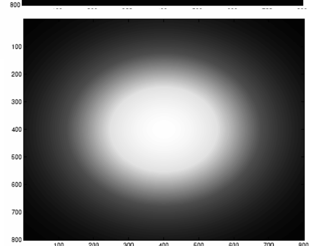


(b1) Gaussian kernel $\mathbf{G}^{(\mathbf{T}_2,t)}$ with $t = 1$, $\lambda_1 = 1, \lambda_2 = 0.05$ and $\theta = -\frac{\pi}{4}$.

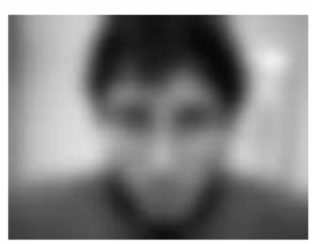
(b2) Corresponding PDE flow $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}_2\mathbf{H}_i)$, at $t = 1$

Laplacian Example





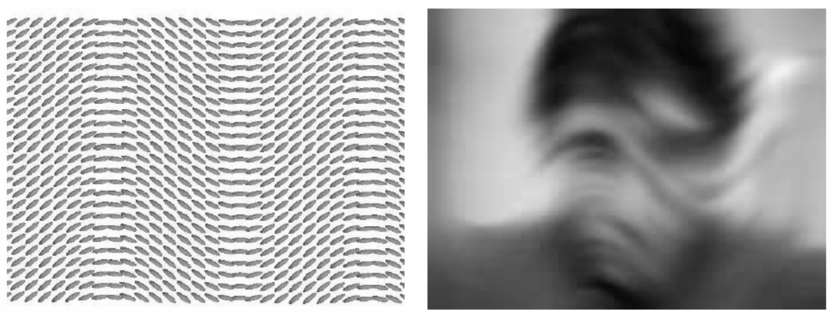
(a1) Gaussian kernel $\mathbf{G}^{(\mathbf{T}_1,t)}$ with $t = 1$, $\lambda_1 = \lambda_2 = 1$ and $\theta \in \mathbb{R}$.



(a2) Corresponding PDE flow $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}_1\mathbf{H}_i)$, at $t = 1$

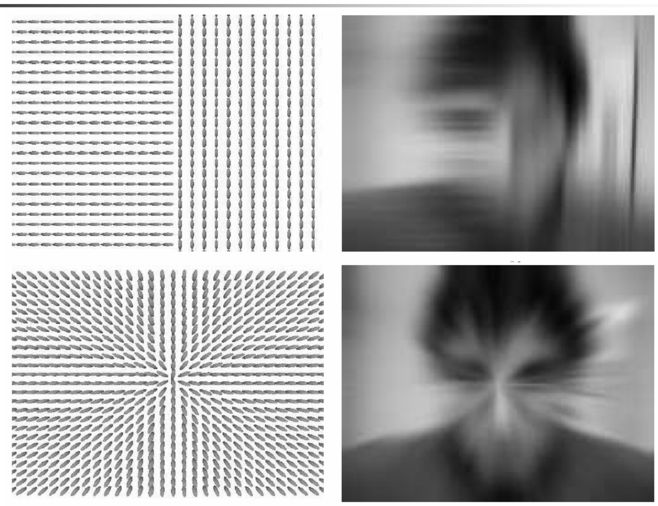
Laplacian Example (varying T)

T is not constant and but independent of image content. There are similarities with bilateral filtering.



(c) Direction field $\mathbf{T}_2 : \Omega \to \mathrm{P}(2)$ (d) Smoothed image with $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}_2\mathbf{H}_i)$

Laplacian Example (varying T)



(a) Tensor field $\mathbf{T}_1 : \Omega \to \mathrm{P}(2)$ (b) Smoothed image with $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}_1\mathbf{H}_i)$

Relation Between T and D

$$D = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial I_i}{\partial t} &= div(D\nabla I_i) \\
&= \mathrm{div}\begin{pmatrix} aI_{i_x} + bI_{i_y} \\ bI_{i_x} + cI_{i_y} \end{pmatrix} \\
&= \frac{\partial}{\partial x}(aI_{i_x} + bI_{i_y}) + \frac{\partial}{\partial y}(bI_{i_x} + cI_{i_y}) \\
&= aI_{i_{xx}} + 2bI_{i_{xy}} + cI_{i_{yy}} + I_{i_x}\left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y}\right) + I_{i_y}\left(\frac{\partial a}{\partial x} + \frac{\partial c}{\partial y}\right) \\
&= trace(D\mathbf{H}_i) + \nabla I_i^T \overrightarrow{div}(D)
\end{aligned}$$

image Hessian

divergence of matrix

$$D = \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T$$

Relation Between T and D

$$\frac{\partial I_i}{\partial t} = trace(T\mathbf{H}_i)$$

$$\frac{\partial I_i}{\partial t} = trace(D\mathbf{H}_i) + \boxed{\nabla I_i^T \overrightarrow{div}(D)}$$

$$\alpha = \frac{2}{\lambda_+ - \lambda_-}\left(\frac{\partial \psi}{\partial \lambda_+} - \frac{\partial \psi}{\partial \lambda_-}\right)$$

$$\begin{aligned}
\nabla I_i^T \overrightarrow{div}(D) &= trace\left(\overrightarrow{div}(D)\nabla I_i^T\right) \\
&= \alpha\, trace\left(div(G)\nabla I_i^T\right) \\
&+ trace\left(G\nabla \alpha \nabla I_i^T\right) \\
&+ trace\left(\nabla \beta \nabla I_i^T\right)
\end{aligned}$$

$$\beta = \frac{2}{\lambda_+ - \lambda_-}\left(\lambda_+ \frac{\partial \psi}{\partial \lambda_-} - \lambda_- \frac{\partial \psi}{\partial \lambda_+}\right)$$

Relation Between T and D

Let's just solve for $div(G)$. Rest can be solved similarly:

$$
\begin{aligned}
\overrightarrow{div}(G) &= \sum_{j=1}^{3} \overrightarrow{div}\begin{pmatrix} I_{j_x}^2 & I_{j_x} I_{j_y} \\ I_{j_x} I_{j_y} & I_{j_y}^2 \end{pmatrix} \\
&= \sum_{j=1}^{3} \begin{pmatrix} 2I_{j_x} I_{j_{xx}} + I_{j_x} I_{j_{yy}} + I_{j_y} I_{j_{xy}} \\ 2I_{j_y} I_{j_{yy}} + I_{j_{xx}} I_{j_y} + I_{j_x} I_{j_{xy}} \end{pmatrix} \\
&= \sum_{j=1}^{3} \begin{pmatrix} I_{j_x}\left(I_{j_{xx}} + I_{j_{yy}}\right) \\ I_{j_y}\left(I_{j_{xx}} + I_{j_{yy}}\right) \end{pmatrix} + \begin{pmatrix} I_{j_x} I_{j_{xx}} + I_{j_y} I_{j_{xy}} \\ I_{j_x} I_{j_{xy}} + I_{j_y} I_{j_{yy}} \end{pmatrix} \\
&= \sum_{j=1}^{3} \Delta I_j \nabla I_j + \mathbf{H}_j \nabla I_j
\end{aligned}
$$

Relation Between T and D

Using *trace* property: $trace(\mathbf{AB}) = trace\left(\frac{\mathbf{A}+\mathbf{A}^T}{2}\mathbf{B}\right)$

$$
div(D\nabla I_i) = \sum_{j=1}^{3} trace\left((\delta_{ij} D + Q^{ij})\mathbf{H}_j\right)
$$

Hessian

Kronecker func.

$$
D = \underbrace{f_1(\lambda_+,\lambda_-)}_{2\frac{\partial \psi}{\partial \lambda_+}}\theta_+\theta_+^T + \underbrace{f_2(\lambda_+,\lambda_-)}_{2\frac{\partial \psi}{\partial \lambda_-}}\theta_-\theta_-^T
$$

$$
\begin{aligned}
\mathbf{P}^{ij} &= \alpha \nabla I_i^T \nabla I_j + \\
&\quad 2\left(\frac{\partial \alpha}{\partial \lambda_+}\theta_+\theta_+^T + \frac{\partial \alpha}{\partial \lambda_-}\theta_-\theta_-^T\right)\nabla I_j \nabla I_i^T \mathbf{G} + \\
&\quad 2\left(\left(\alpha + \frac{\partial \beta}{\partial \lambda_+}\right)\theta_+\theta_+^T + \left(\alpha + \frac{\partial \beta}{\partial \lambda_-}\right)\theta_-\theta_-^T\right)\nabla I_j \nabla I_i^T
\end{aligned}
$$

$$
\mathbf{Q}^{ij} = \frac{\mathbf{P}^{ij} + \mathbf{P}^{ij^T}}{2}
$$

Rewriting the Formula

$$div(D\nabla I_i) = \sum_{j=1}^{3} trace\big((\delta_{ij} D + Q^{ij})\mathbf{H}_j\big)$$

$$div(D\nabla I_i) = \sum_{j=1}^{3} trace\big(\mathbf{A}^{ij}\mathbf{H}j\big)$$

$$div(D\nabla I) = \overrightarrow{trace}(AH)$$

super matrix notation

$$A = \begin{bmatrix} A^{11} & A^{12} & A^{13} \\ A^{21} & A^{22} & A^{23} \\ A^{31} & A^{32} & A^{33} \end{bmatrix} \qquad H = \begin{bmatrix} H^1 \\ H^2 \\ H^3 \end{bmatrix}$$

Unified Expression

$$\frac{\partial I}{\partial t} = \overrightarrow{trace}(AH)$$

$$I_i^t - I_i^{t-1} = trace(\mathbf{T}\mathbf{H}_i)$$

$$I_i^t - I_i^{t-1} = trace\left(\left[\underbrace{\frac{1}{1+\lambda_+^* + \lambda_-^*}}_{f_1(\lambda_+,\lambda_-)} \theta_-^* \theta_-^{*T} + \underbrace{\frac{1}{\sqrt{1+\lambda_+^* + \lambda_-^*}}}_{f_2(\lambda_+,\lambda_-)} \theta_+^* \theta_+^{*T}\right]\mathbf{H}_i\right)$$

Numerical Implementation

- Conventional approach
  - Compute image Hessians and Gradients
- Proposed method
  - Use local filtering by Gaussians (2nd page)

$$\frac{\partial I_i}{\partial t} = trace(\mathbf{TH}_i) \Leftrightarrow I_{i_{(t)}} = I_{i_{(t=0)}} * G_\sigma^{(\mathbf{T},t)}$$

Similar to bilateral filtering $$G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t}\exp\left(-\frac{\mathbf{x}\mathbf{T}^{-1}\mathbf{x}}{4t}\right)$$

Numerical Implementation

1. Compute local convolution mask $G^{(\mathbf{T},t)}$ defining local geometry by $\mathbf{T}$.
2. Estimate $trace(\mathbf{TH}_i)$ in local neighborhood of $\mathbf{x}$.

$$[trace(\mathbf{TH}_i)](x,y) = \begin{array}{|c|c|c|}\hline i(x-1,y-1) & i(x-1,y) & i(x-1,y+1) \\ \hline i(x,y-1) & i(x,y) & i(x,y+1) \\ \hline i(x+1,y-1) & i(x+1,y) & i(x+1,y+1) \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline G(x-1,y-1) & G(x-1,y) & G(x-1,y+1) \\ \hline G(x,y-1) & G(x,y) & G(x,y+1) \\ \hline G(x+1,y-1) & G(x+1,y) & G(x+1,y+1) \\ \hline \end{array}$$

3. Apply filtering for each $trace(\mathbf{A}^{ij}\mathbf{H}_j)$ in the vector $trace(AH)$

Comparison Between Two Implementations

(a) Noisy color image

(b) Restored with the Hessian spatial discretization scheme

(c) Restored with the local filtering scheme, with 3×3 masks.