Nonlinear complex and cross diffusion for texture preserving denoising

Dirk A. Lorenz
dirk@ee.technion.ac.il

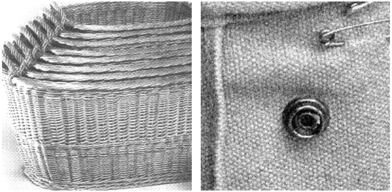
joint work with Yehoshua Y. Zeevi

HASSIP 06, September 14, 2006

1. Going complex with diffusion
   - Why and how
   - Existence of solutions
2. Preserve textures while diffuse
3. Implementation

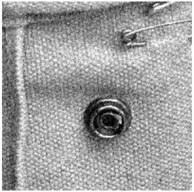
Diffusion for denoising

Images are noisy.



Diffusion for denoising

Images are noisy.

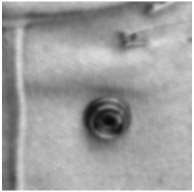


Smoothing PDEs for denoising

Heat equation (1980s)

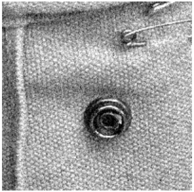
$$I_t = \operatorname{div}(c\nabla I)$$
$$I(0,x) = I_0(x)$$

$I(3,x) =$



Diffusion for denoising

Images are noisy.



Smoothing PDEs for denoising

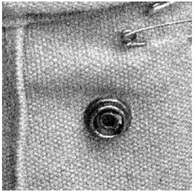
Heat equation (1980s)
Perona-Malik equation (1990)

$$I_t = \operatorname{div}(c_\lambda(|\nabla I|)\nabla I)$$
$$I(0,x) = I_0(x)$$

$I(3,x) =$

Diffusion for denoising

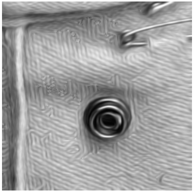
Images are noisy.



Smoothing PDEs for denoising

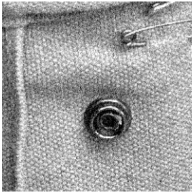
Heat equation (1980s)
Perona-Malik equation (1990)
coherence enhancing diffusion (1996)

$$I_t = \operatorname{div}(C(|\nabla I_\sigma|)\nabla I)$$
$$I(0,x) = I_0(x)$$

$I(3,x) =$ 

Diffusion for denoising

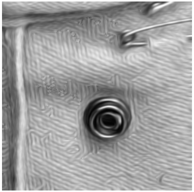
Images are noisy.



Smoothing PDEs for denoising

Heat equation (1980s)
Perona-Malik equation (1990)
coherence enhancing diffusion (1996)

$$I_t = \operatorname{div}(C(|\nabla I_\sigma|)\nabla I)$$
$$I(0,x) = I_0(x)$$

$I(3,x) =$ 

(Use edge detector in the diffusion coefficient)

Store the second derivative

Gilboa, Sochen, Zeevi (2002): Why not $c \in \mathbb{C}$?

$$I_t = e^{i\theta} \Delta I$$

Observation imag$I \approx \Delta I$

The imaginary part is a storage for second derivative information ($I = u + iv$):

$$u_t = \cos\theta \Delta u - \sin\theta \Delta v$$
$$v_t = \cos\theta \Delta v + \sin\theta \Delta u$$

This motivates:

$$I_t = e^{i\theta} \operatorname{div}(c(\text{imag} I) \nabla I)$$

From complex to cross diffusion

$$I_t = e^{i\theta}\,\mathrm{div}(c(\mathrm{imag}I)\nabla I)$$

$$I = u + iv$$

Coupled equations for real and imaginary part:

$$u_t = \mathrm{div}(c(v)(\cos\theta\nabla u - \sin\theta\nabla v)) \qquad u(0,x) = I_0(x)$$
$$v_t = \mathrm{div}(c(v)(\cos\theta\nabla v + \sin\theta\nabla u)) \qquad v(0,x) = 0$$

From complex to cross diffusion

$$I_t = e^{i\theta}\,\mathrm{div}(c(\mathrm{imag}I)\nabla I)$$

$$I = u + iv$$

Coupled equations for real and imaginary part:

$$u_t = \mathrm{div}(c(v)(\cos\theta\nabla u - \sin\theta\nabla v)) \qquad u(0,x) = I_0(x)$$
$$v_t = \mathrm{div}(c(v)(\cos\theta\nabla v + \sin\theta\nabla u)) \qquad v(0,x) = 0$$

Cross diffusion system

From complex to cross diffusion

$$I_t = e^{i\theta}\,\mathrm{div}(c(\mathrm{imag}I)\nabla I)$$

$$I = u + iv$$

Coupled equations for real and imaginary part:

$$\begin{aligned} u_t &= \mathrm{div}(c(v)(\cos\theta\nabla u)) & u(0,x) &= I_0(x) \\ v_t &= \mathrm{div}(c(v)(\cos\theta\nabla v + \sin\theta\nabla u)) & v(0,x) &= 0 \end{aligned}$$

Reduce to triangular system

Cross diffusion systems are complex

General form of quasilinear parabolic cross diffusion system:

$$u_t = \operatorname{div}(A(u,v)\nabla u + B(u,v)\nabla v) + f(u,v)$$
$$v_t = \operatorname{div}(C(u,v)\nabla v + D(u,v)\nabla u) + g(u,v)$$

Example (What is so complicated?)

$$u_t = \operatorname{div}(c(u,v)\nabla u)$$
$$v_t = \Delta v + \Delta u$$

Δu not bounded $\rightsquigarrow$ solution for v?

Boundedness is the key to existence

Theorem

Let $P, Q, R > 0$. If the solution of

$$u_t = \operatorname{div}(Q(u,v)\nabla u)$$
$$v_t = \operatorname{div}(P(u,v)\nabla v + R(u,v)\nabla u)$$

is a priori bounded, then the solution is global. *Amann, 1989.*

Theorem

Let $c > 0$, $0 < \theta < \pi/2$, $0 < k < 1$. Then

$$u_t = \operatorname{div}(c(v)\cos\theta\nabla u)$$
$$v_t = k\operatorname{div}(c(v)(\cos\theta\nabla v + \sin\theta\nabla u))$$

has a bounded global solution.

Combine best of wavelets and diffusion

	Wavelets	Diffusion
+	handle textures	denoise visually pleasing
	denoise quickly	
-	produce unpleasent artifacts	treats noise and texture equally

Combine best of wavelets and diffusion

	Wavelets	Diffusion
+	handle textures	denoise visually pleasing
	denoise quickly	
-	produce unpleasent artifacts	treats noise and texture equally

- Use wavelets for texture extraction
- Use diffusion for the denoising.
- Introduce the texture into the diffusion process

Meyer model + cross diffusion

In a world where images are in BV *and the eye measures* L^2 *distance, wavelet shrinkage would be perfect.*
(Y. Meyer)

Meyer model + cross diffusion

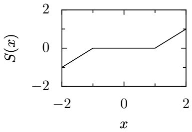
In a world where images are in BV and the eye measures L^2 distance, wavelet shrinkage would be perfect.
(Y. Meyer)

$$I_{\text{denoised}} = \operatorname{argmin} \|I - I_0\|_{L^2}^2 + t_1 |I|_{B^1_{1,1}}$$

Meyer model + cross diffusion

In a world where images are in BV and the eye measures L^2 distance, wavelet shrinkage would be perfect.
(Y. Meyer)

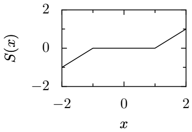
$$I_{\text{denoised}} = S_{t_1}(I_0)$$



Meyer model + cross diffusion

In a world where images are in BV and the eye measures L^2 distance, wavelet shrinkage would be perfect.
(Y. Meyer)

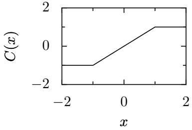
$$I_{\text{denoised}} = S_{t_1}(I_0)$$
$$I_{\text{texture}} = \operatorname{argmin} \|I - I_0\|^2_{L^2} + t_2 |I|_{B^{-1}_{\infty,\infty}}$$



Meyer model + cross diffusion

In a world where images are in BV and the eye measures L^2 distance, wavelet shrinkage would be perfect.
(Y. Meyer)

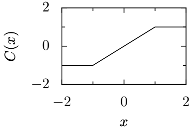
$$I_{\text{denoised}} = S_{t_1}(I_0)$$
$$I_{\text{texture}} = C_{\tilde{t}_2}(I_0)$$



Meyer model + cross diffusion

In a world where images are in BV and the eye measures L^2 distance, wavelet shrinkage would be perfect.
(Y. Meyer)

$$I_{\text{denoised}} = S_{t_1}(I_0)$$
$$I_{\text{texture}} = C_{\tilde{t}_2}(I_0)$$



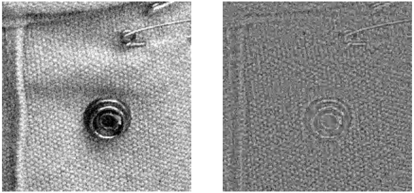
$$u_t = \operatorname{div}(g(v)\cos\theta\nabla u) + C_{\tilde{t}_2}(S_{t_1}(I_0))$$
$$v_t = k\operatorname{div}(g(v)(\cos\theta\nabla v + \sin\theta\nabla u))$$

Fast semi-implicit methods

$$\frac{u^{n+1}-u^n}{dt} = \operatorname{div}(g(v^n)\cos\theta\nabla u^{n+1})$$
$$\frac{v^{n+1}-v^n}{dt} = k\operatorname{div}(g(v^n)(\cos\theta\nabla v^{n+1}+\sin\theta\nabla u^{n+1}))$$

- Solve two linear systems of size = number of pixels
- 200×200, CG method: < 2sec
- No restriction on timestep dt

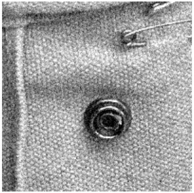
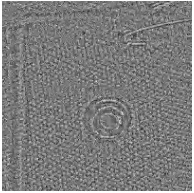
Results



$\lambda = .002$ $\lambda = .004$ $\lambda = .006$

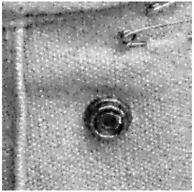
no source

Results

$\lambda = .002$ $\lambda = .004$ $\lambda = .006$

with source



Conclusion

- Cross diffusion offers a storage for differential information.
- Existence for the solution of a system
  close the nonlinear complex diffusion can be proven.
- Source terms formed by means of wavelets
  can be used to preserve texture.