Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's

David Tschumperlé
CNRS UMR 6072 (GREYC/ENSICAEN) - Image Team

MIA'2006, Paris/France, Sept. 2006

Presentation Layout

- Diffusion PDE's for Multi-Valued Images : A Quick Review
- Introducing Curvature-Preserving PDE's for Image Regularization
- Connections with Line Integral Convolutions and Implementation
- Application Results

Presentation Layout

⇒ **Diffusion PDE's for Multi-Valued Images : A Quick Review**

- Introducing Curvature-Preserving PDE's for Image Regularization
- Connections with Line Integral Convolutions and Implementation
- Application Results

PDE's and Image Regularization

- Convolution and Linear PDE (Koenderink[84], Alvarez-Guichard-etal[92], ...) :

$$I_{(t)} = I_{(t=0)} * G_\sigma \quad \text{where} \quad G_\sigma = \frac{1}{4\pi t}\, e^{-\frac{x^2+y^2}{4t}} \quad \Longleftrightarrow \quad \frac{\partial I}{\partial t} = \Delta I = \mathrm{div}\,(\nabla I)$$

- Nonlinear PDE (Perona-Malik[90], Alvarez [92], ...) :

$$\frac{\partial I}{\partial t} = \mathrm{div}\,(c(\|\nabla I\|)\;\nabla I) \qquad \text{with} \quad c : \mathbb{R} \longrightarrow \mathbb{R}$$

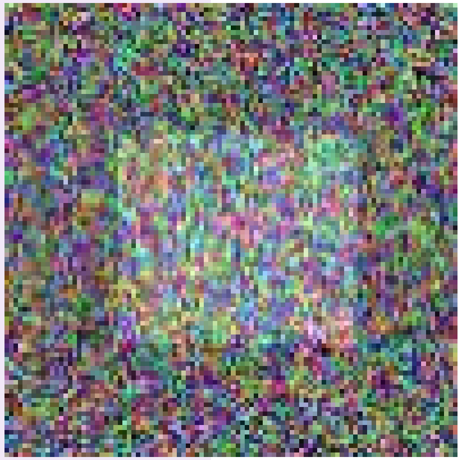
Noisy image

Heat flow

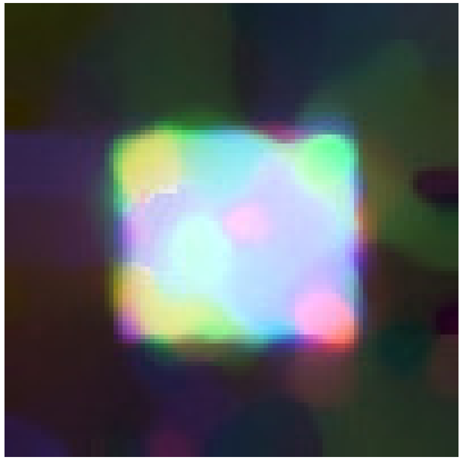
Perona-Malik

Regularization PDE's and Multi-Valued Images

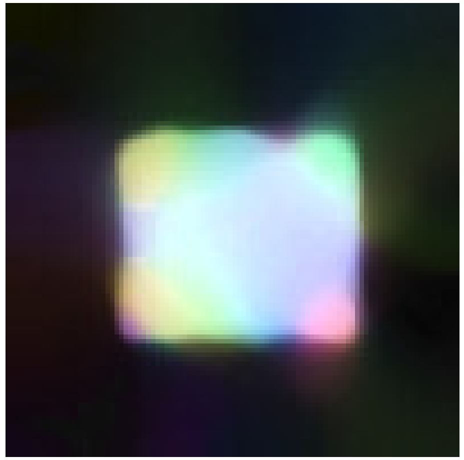
- Image $\mathbf{I} : \Omega \to \mathcal{N}$ of multi-valued points : vectors ($\mathcal{N} = \mathbb{R}^n$), matrices ($\mathcal{N} = \mathcal{M}_n$).
  (Alvarez, Aubert, Barlaud, Blanc-Feraud, **Blomgren**, Charbonnier, **Chan**, Cohen, **Deriche**, Kornprobst, **Kimmel**, **Malladi**, Munford, Morel, Nordström, Osher, Perona, Malik, Rudin, **Sapiro**, **Sochen**, **Tschumperlé**, **Weickert**,...)



Color image ($\mathcal{N} = \mathbb{R}^3$)

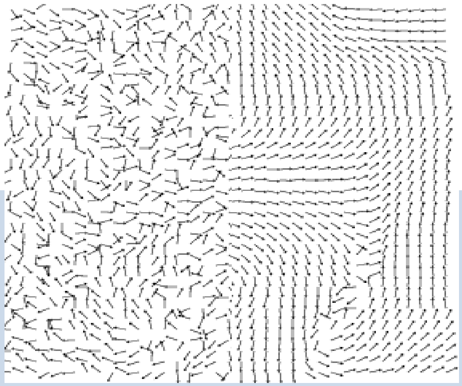


Scalar PDE applied on each channel

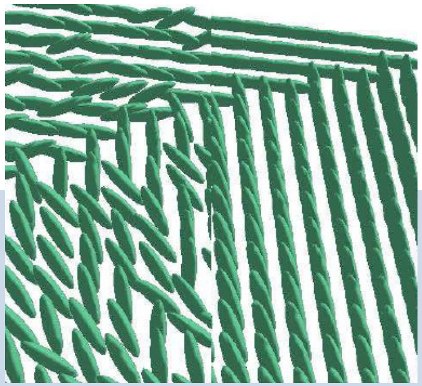


Multi-valued PDE

Color image



Direction field



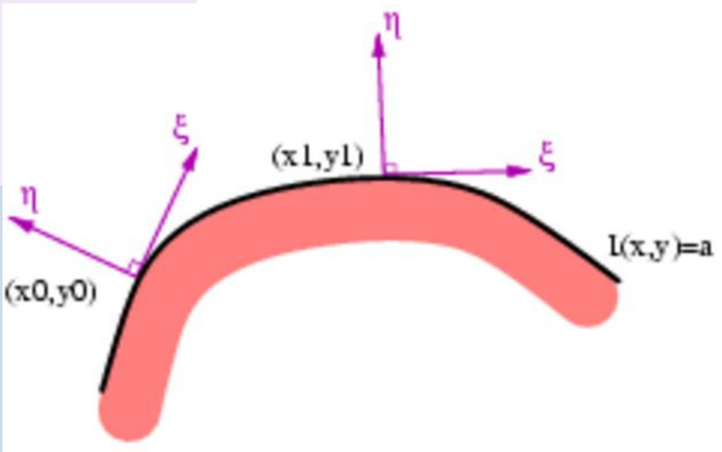
Tensor field

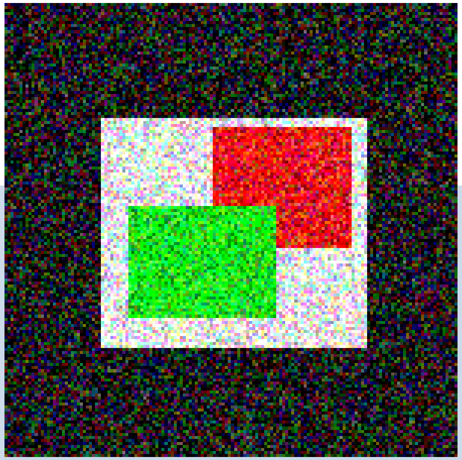
Basic Regularization Principle

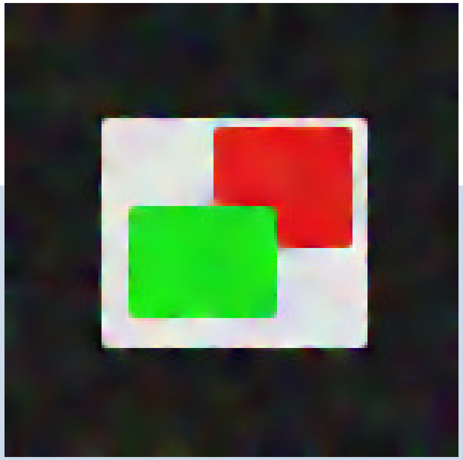
- PDE regularization is mainly based on local image smoothing.

- Local smoothing rules :
  - On a edge, smoothing performed only along the edge direction *(anisotropic smoothing)*.
  - On homogeneous regions, smoothing performed equally in all directions *(isotropic smoothing)*.

- **Generalization :** Separating the definition of a smoothing geometry, and the smoothing process itself.







Definition of a smoothing geometry

- $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$, a multi-valued image.

- Computing the local geometry of $\mathbf{I}$, described by the smoothed structure tensor field $\mathbf{G}_\sigma$:

$$\mathbf{G}_{\sigma_{(x,y)}} = \left(\sum_i \nabla I_i \nabla I_i^T \right) * G_\sigma$$

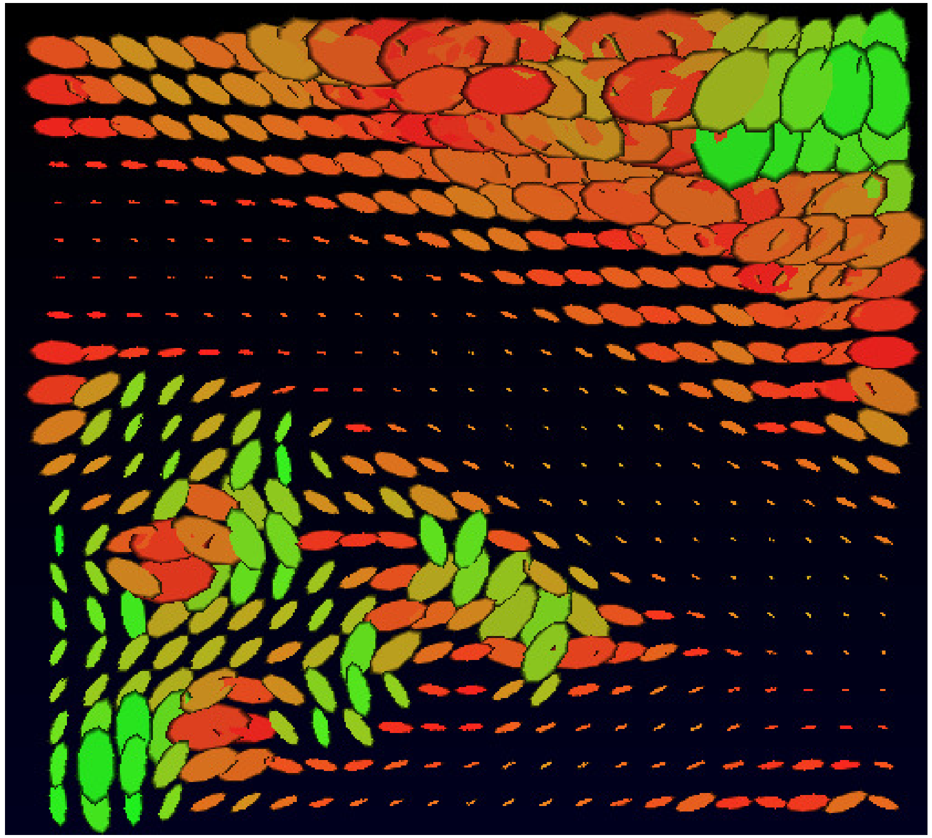
- Eigenvalues λ_+, λ_- and Eigenvectors θ_+, θ_- of $\mathbf{G}_\sigma$ are very efficient descriptors of the local configuration of $\mathbf{I}$ at (x, y).

$\Rightarrow$ The smoothing is then modelized by a tensor field $\mathbf{T}$ from $\mathbf{G}$:

$$\mathbf{T} = f_1(\lambda_+ + \lambda_-)\, \theta_-\theta_-^T + f_2(\lambda_+ + \lambda_-)\, \theta_+\theta_+^T \quad \text{with} \quad \begin{cases} f_1(s) = \frac{1}{1+s^p} \\ f_2(s) = \frac{1}{1+s^q} \end{cases}$$

Tensor field representation

Top of the Lena hat



Corresponding diffusion tensor field $\mathbb{T}$

⇒ Tensor field $\mathbb{T}$ tells about the desired smoothing directions and smoothing amplitudes.

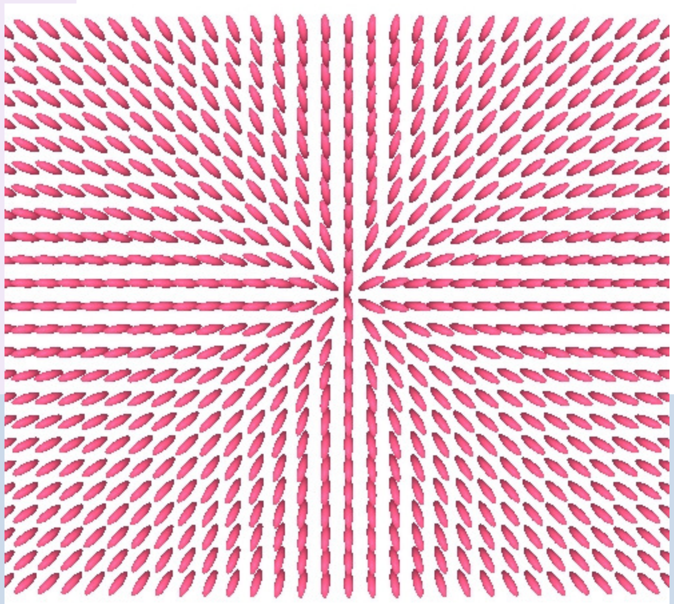
How the smoothing is then achieved ?

- The smoothing itself is performed by the application of one or several iterations of one of these generic PDE's :

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\left(\mathbf{T}\nabla I_i\right) \qquad or \qquad \frac{\partial I_i}{\partial t} = \mathrm{trace}\left(\mathbf{TH}_i\right)$$

⇒ Most of existing regularization PDE's conform with these equations.

⇒ Ideally, such PDE's comply with the diffusion tensor field $\mathbf{T}$:



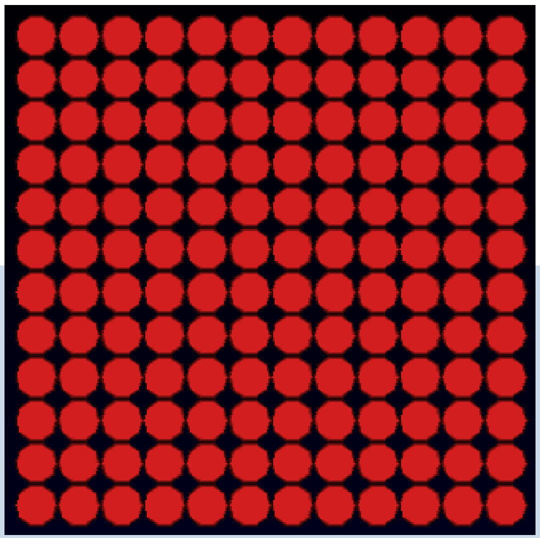
Tensor-directed PDE applied on a color image.

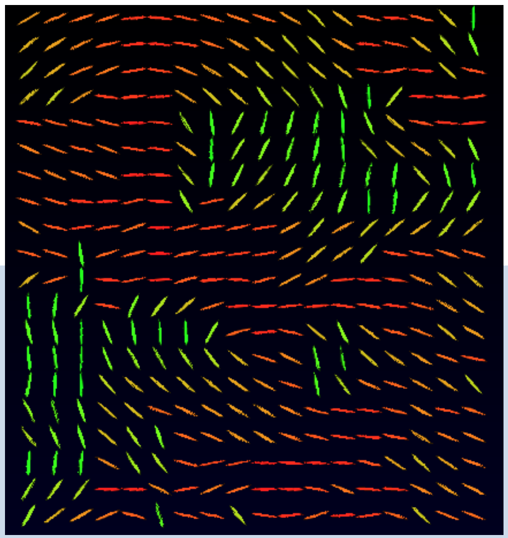
Issues encountered with above formulations

- Slow iterative processes : Several iterations are needed to get a result that is regularized enough (since $dt \rightarrow 0$).

- **Problems with Divergence formulations :**
  - Non-unicity of the tensor field : $\exists \mathbf{D}_1 \neq \mathbf{D}_2, \quad \text{div}(\mathbf{D}_1 \nabla I) = \text{div}(\mathbf{D}_2 \nabla I)$.
  - Tensor shapes not always representative of the intuitive smoothing behavior :

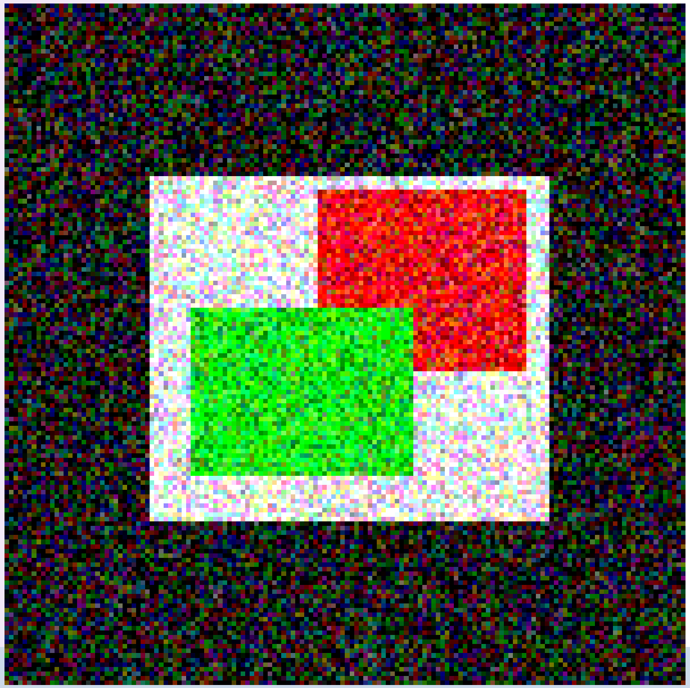
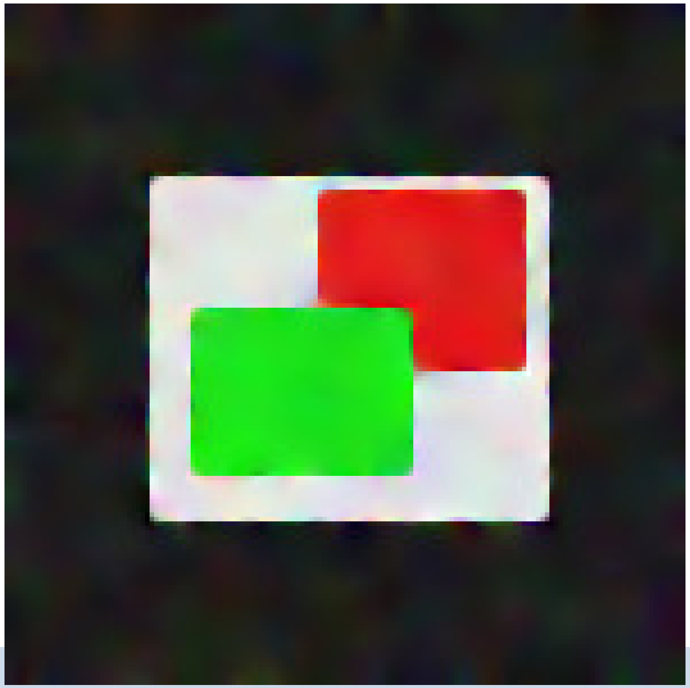
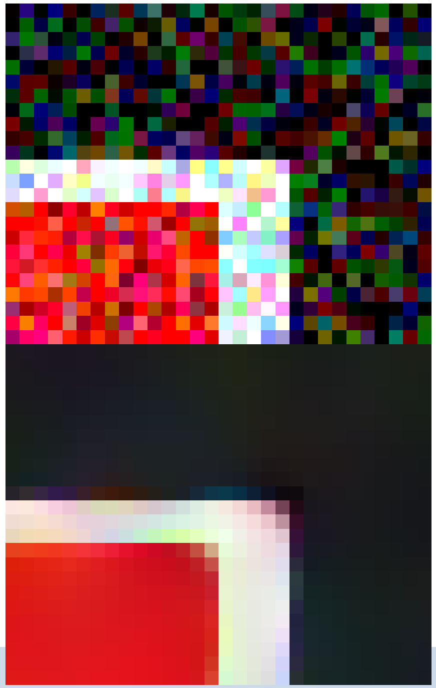
$$\mathbf{D}_1 = \mathbf{Id} \quad \text{and} \quad \mathbf{D}_2 = \frac{\nabla I \nabla I^T}{\|\nabla I\|^2} \qquad \Rightarrow \qquad \frac{\partial I}{\partial t} = \Delta I.$$





Issues encountered with above formulations

- **Problems with Trace formulations :**
  - Better respect of the considered tensor-valued geometry.
  - But tends to over-smooth high-curvature structures (corners).

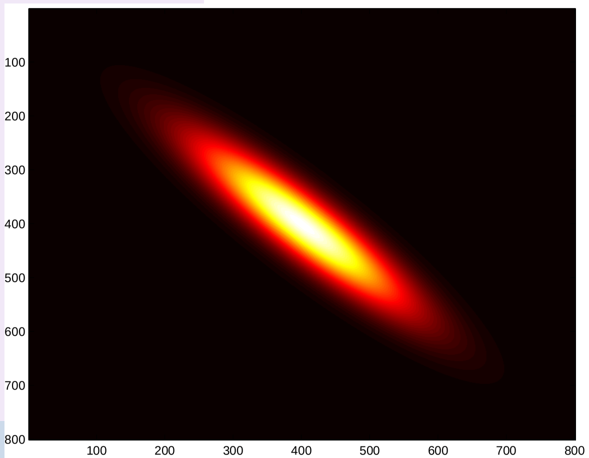




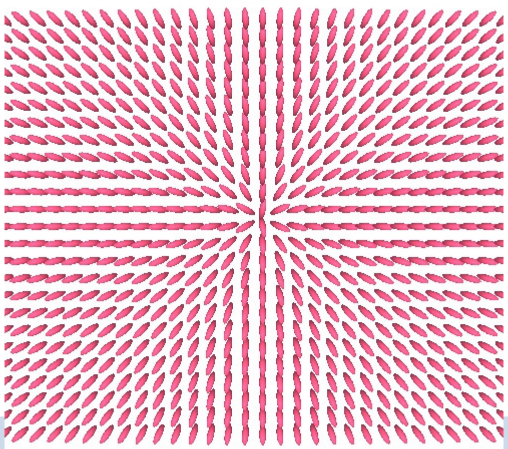
Issues encountered with above formulations

- Can be understood from the mono-directional smoothing case :

$$\frac{\partial I}{\partial t} = \text{Trace}(\mathbf{w}\mathbf{w}^T\mathbf{H})$$

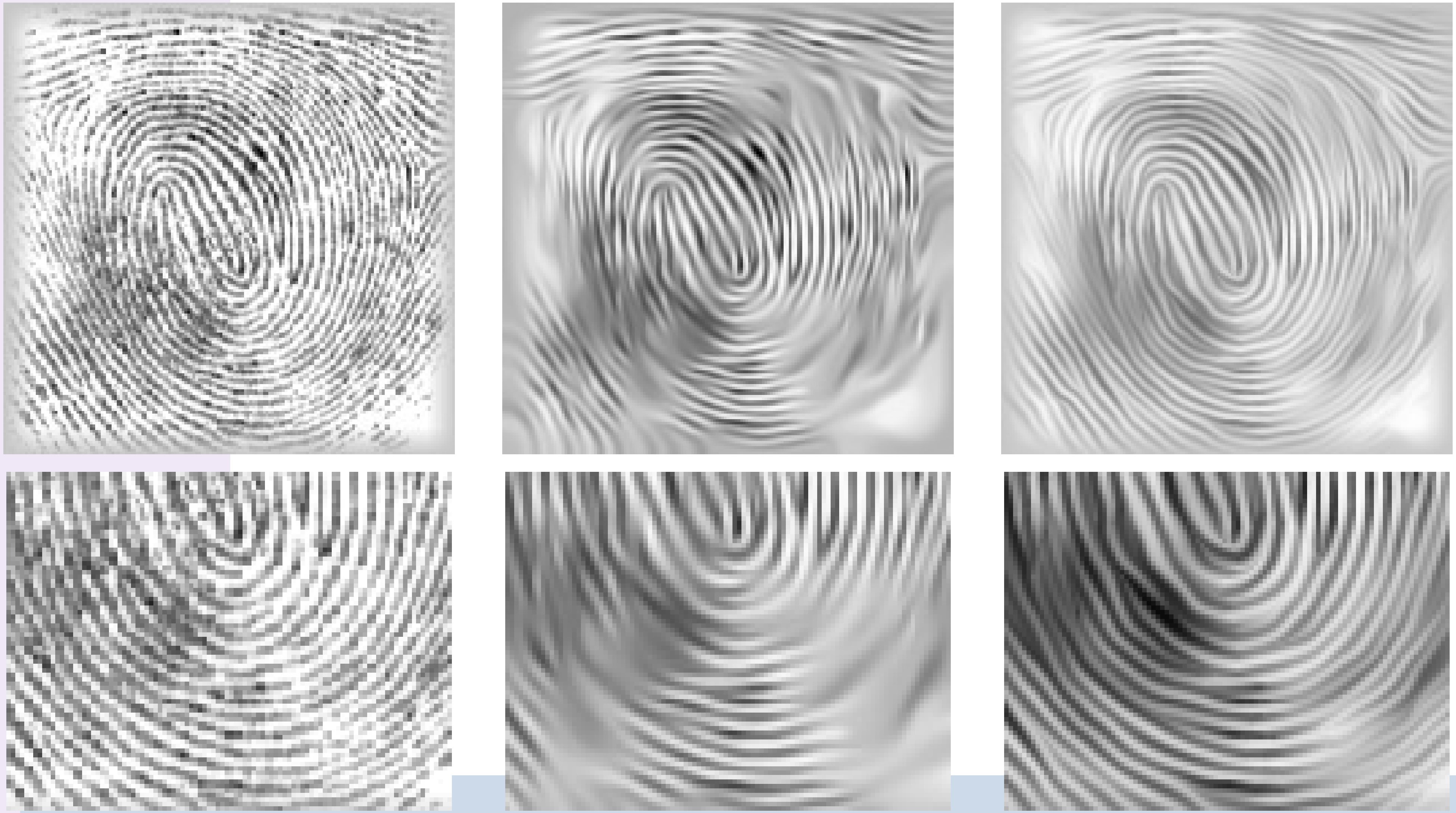
- Geometrical interpretation in terms of local smoothing by gaussian kernels oriented by $\mathbf{w}_{(x,y)}$.





- Ideally, we would be able to locally smooth with a kernel that curves itself regarding to the local image structure.

Coming Next...



$\Rightarrow$ Need for specific PDE's avoiding smoothing of structures having high curvatures.

Presentation Layout

- Diffusion PDE's for Multi-Valued Images : A Quick Review

⇒ **Introducing Curvature-Preserving PDE's for Image Regularization**

- Connections with Line Integral Convolutions and Implementation

- Application Results

Curvature-preserving PDE's

- For the mono-directional case, we propose the following PDE :

$$\frac{\partial I_i}{\partial t} = \text{trace}\left(\mathbf{w}\mathbf{w}^T\,\mathbf{H}_i\right) + \nabla I_i^T \mathbf{J}_{\mathbf{w}}\mathbf{w}$$

where $\mathbf{J}_{\mathbf{w}} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}$ and $\mathbf{H}_i = \begin{pmatrix} \frac{\partial^2 I_i}{\partial x^2} & \frac{\partial^2 I_i}{\partial x \partial y} \\ \frac{\partial^2 I_i}{\partial x \partial y} & \frac{\partial^2 I_i}{\partial y^2} \end{pmatrix}$.

⇒ Classical “Trace” formulation + Constraint term depending on the variations of $\mathbf{w}$.

How did the constraint term appear ?

- If $\mathcal{C}^{\mathbf{X}}$ stands for the integral curve of $\mathbf{w}$ starting from $\mathbf{X} = (x, y)$, and parameterized by a s.a : $\mathcal{C}^{\mathbf{X}}_{(0)} = \mathbf{X}$ et $\frac{\partial \mathcal{C}^{\mathbf{X}}_{(a)}}{\partial a} = \mathbf{w}(\mathcal{C}^{\mathbf{X}}_{(a)})$, then :

$$\mathcal{C}^{\mathbf{X}}_{(h)} = \mathcal{C}^{\mathbf{X}}_{(0)} + h \frac{\partial \mathcal{C}^{\mathbf{X}}_{(a)}}{\partial a}_{|a=0} + \frac{h^2}{2} \frac{\partial^2 \mathcal{C}^{\mathbf{X}}_{(a)}}{\partial a^2}_{|a=0} + O(h^3) = \mathbf{X} + h \mathbf{w}_{(\mathbf{X})} + \frac{h^2}{2} \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} + O(h^3)$$

with $h \to 0$, and $O(h^n) = h^n \, \epsilon_n$. Thus, we get :

$$\begin{aligned} I_i(\mathcal{C}^{\mathbf{X}}_{(h)}) &= I_i \left(\mathbf{X} + h \mathbf{w}_{(\mathbf{X})} + \frac{h^2}{2} \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} + O(h^3) \right) \\ &= I_i(\mathbf{X}) + h \nabla I_{i(\mathbf{X})}^T \left(\mathbf{w}_{(\mathbf{X})} + \frac{h}{2} \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} \right) + \frac{h^2}{2} \mathrm{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) + O(h^3) \end{aligned}$$

- and then...

$$\begin{aligned} \frac{\partial^2 I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial a^2}_{|a=0} &= \lim_{h \to 0} \frac{1}{h^2} \left[I_i(\mathcal{C}^{\mathbf{X}}_{(h)}) + I_i(\mathcal{C}^{\mathbf{X}}_{(-h)}) - 2 I_i(\mathcal{C}^{\mathbf{X}}_{(0)}) \right] \\ &= \lim_{h \to 0} \frac{1}{h^2} \left[h^2 \, \nabla I_i^T \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} + h^2 \, \mathrm{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) + O(h^3) \right] \\ &= \mathrm{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) + \nabla I_i^T \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} \end{aligned}$$

Interpretation of the constraint term

- Our curvature-preserving PDE can be written as :

$$\frac{\partial I_i}{\partial t} = \frac{\partial^2 I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial a^2}_{|a=0} = \Delta^{\mathbf{X}}_{\mathcal{C}} I_i$$

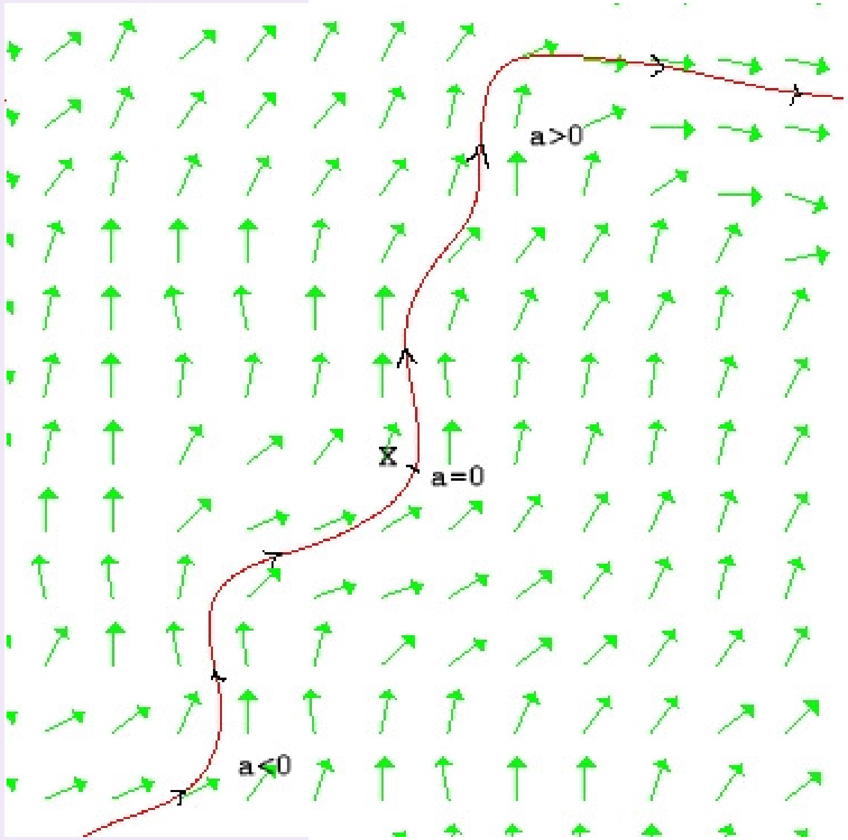
where $\mathcal{C}^{\mathbf{X}}$ is the integral line of $\mathbf{w}$ starting from $\mathbf{X}$, and parameterized as :

$$\mathcal{C}^{\mathbf{X}}_{(0)} = \mathbf{X} \qquad \text{and} \qquad \frac{\partial \mathcal{C}^{\mathbf{X}}_{(a)}}{\partial a} = \mathbf{w}(\mathcal{C}^{\mathbf{X}}_{(a)})$$

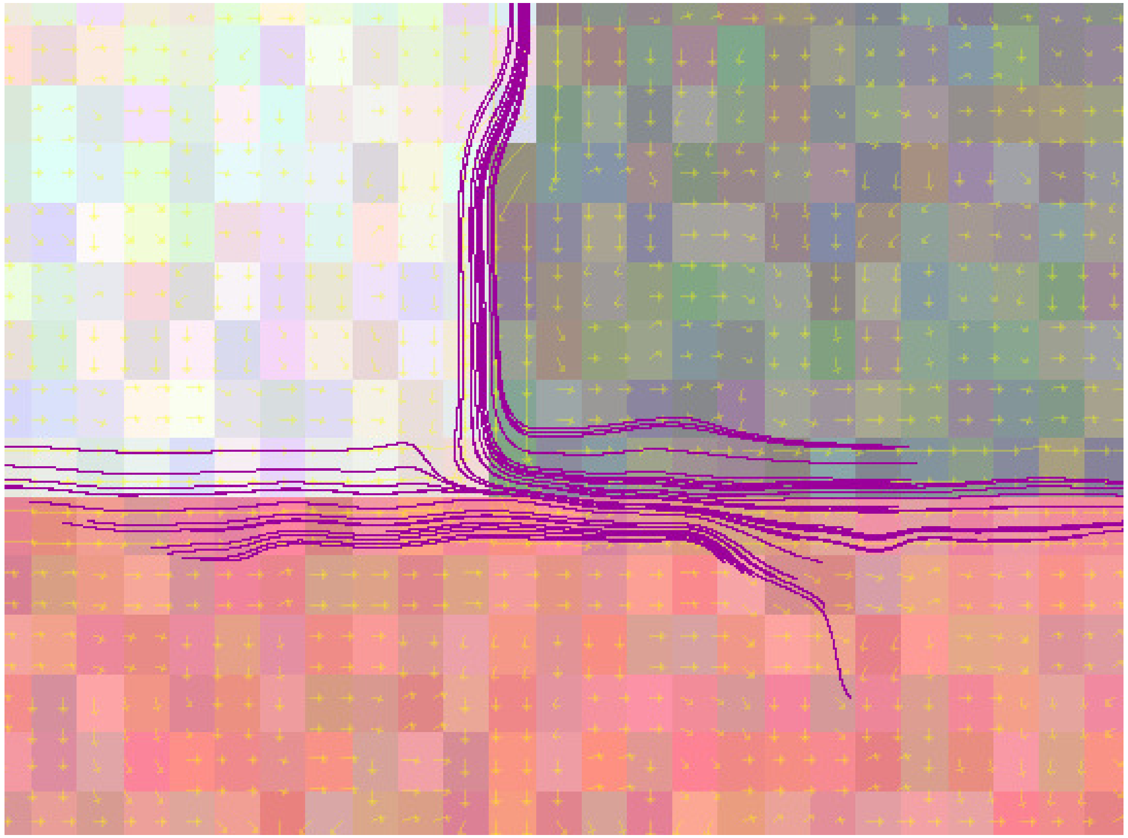
- PDE equivalent to a heat flow on the integral lines of $\mathbf{w}$.

- If $\mathbf{w}$ is chosen to be the directions of the image contours (eigenvector θ_- of $\mathbf{G}_\sigma$), the smoothing will respect the shape of the contour, whatever its curvature is.

Smoothing along integral lines



(a) An integral line $\mathcal{C}^{\mathrm{X}}$



(b) Some integral lines around a triple-junction.

⇒ The performed smoothing will preserve the curved structure.

Extension to a tensor-based geometry

- More generaly, we are more interested to a tensor-valued smoothing geometry $\mathbf{T}$ than a vectorial one $\mathbf{w}$.

- We propose to decompose the field $\mathbf{T}$ along all orientations of the plane :

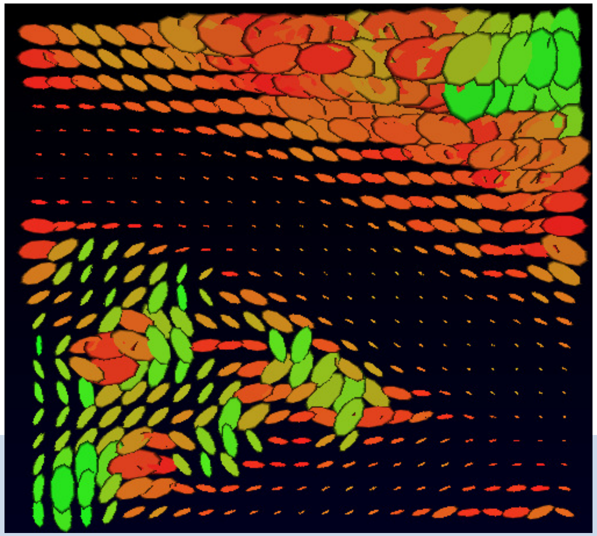
$$\mathbf{T} = \frac{2}{\pi}\sqrt{\mathbf{T}}\left(\int_{\alpha=0}^{\pi} a_\alpha a_\alpha^T \, d\alpha\right)\sqrt{\mathbf{T}} \quad \text{where } a_\alpha = \begin{pmatrix} \cos\alpha & \sin\alpha \end{pmatrix}^T.$$

- This suggests to extend naturally our original PDE to this tensor-directed one :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) + \frac{2}{\pi}\nabla I_i^T \int_{\alpha=0}^{\pi} \mathbf{J}_{\sqrt{\mathbf{T}}a_\alpha}\sqrt{\mathbf{T}}a_\alpha \, d\alpha$$

Extension to a tensor-based geometry

- Local behavior of our equation :
  - When the tensor $\mathbf{T}$ is isotropic, we are on an homogeneous region : the smoothing is performed with the same strength in all directions a_α.
  - When the tensor $\mathbf{T}$ is anisotropic, we are on an image contour : the smoothing is performed only along this contour (also respecting its curvature).


Presentation Layout

- Diffusion PDE's for Multi-Valued Images : A Quick Review
- Introducing Curvature-Preserving PDE's for Image Regularization

⇒ **Connections with Line Integral Convolutions and Implementation**

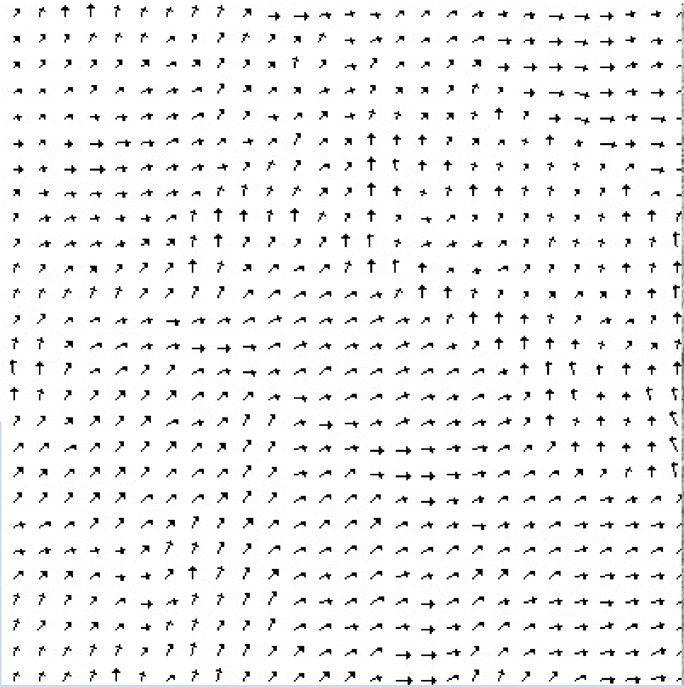
- Application Results

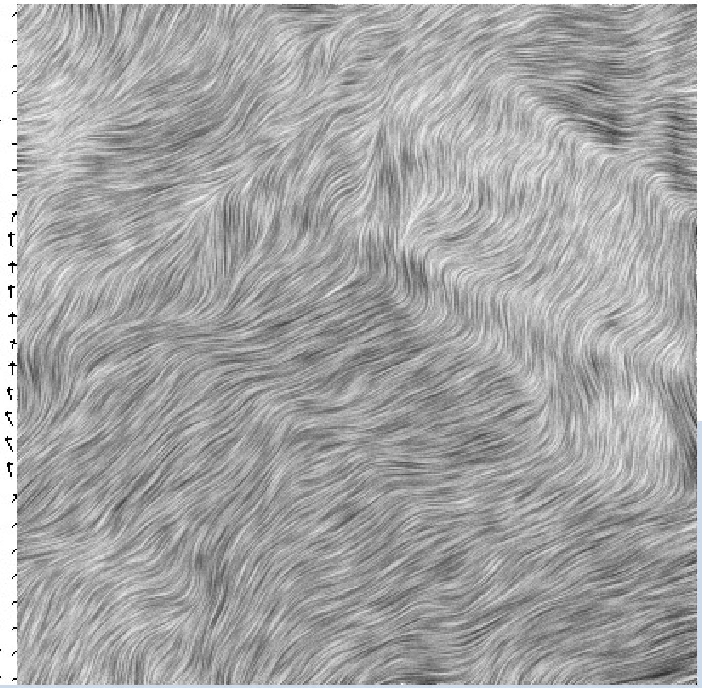
Line Integral Convolutions (LIC's)

- [Cabral & Leedom, 93] : Way to create textured versions of 2D vector fields $\mathcal{F}$.

⇒ From a pure noisy image $\mathbf{I}^{\text{noise}}$, one computes for each pixel $\mathbf{X} = (x, y)$

$$\mathbf{I}^{LIC}_{(x,y)} = \frac{1}{N} \int_{-\infty}^{+\infty} f(p)\, \mathbf{I}^{noise}(\mathcal{C}^{\mathbf{X}}_{(p)})\, dp \qquad \text{where} \qquad \begin{cases} \mathcal{C}^{\mathbf{X}}_{(0)} & = & \mathbf{X} \\ \frac{\partial \mathcal{C}^{\mathbf{X}}_{(a)}}{\partial a} & = & \mathcal{F}(\mathcal{C}^{\mathbf{X}}_{(a)}) \end{cases}$$





Link between our PDE and LIC's

- Our mono-directional PDE is a $1D$ heat flow on an integral line $\mathcal{C}^{\mathbf{X}}$.

- The solution of a classical heat flow is a convolution of the data by a Gaussian kernel (Koenderink[84], Alvarez-Guichard-etal[92]).

⇒ We can then implement all sub-smoothings of our PDE along $\mathbf{w} = \sqrt{\mathbf{T}} a_\alpha$, as local convolutions of the pixel data by a 1D Gaussian, along the integral lines $\mathcal{C}^{\mathbf{X}}$ of $\mathbf{w}$.

⇒ Possible and direct implementation of our PDE by iterative short LIC computations.

$$\mathbf{I}^{regul}_{(\mathbf{X})} = \frac{1}{N} \int_0^{\pi} \int_{-dt}^{dt} f(a)\, \mathbf{I}^{noisy}(\mathcal{C}^{\theta}_{(\mathbf{X},a)})\, da\, d\theta$$

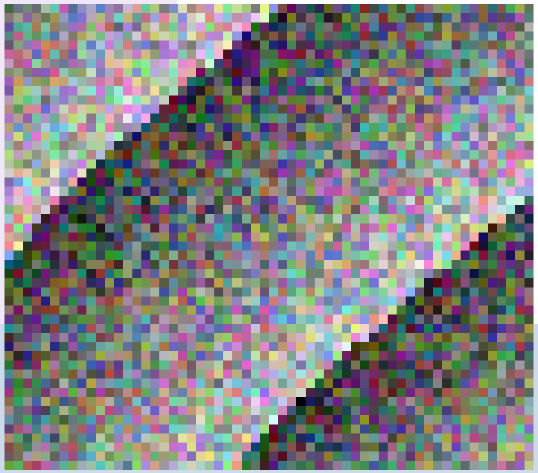
where $f()$ is a 1D Gaussian function, $N = \int \int f(a) da d\theta$, and dt corresponds to the time step (global smoothing strength for one iteration).

Algorithm properties

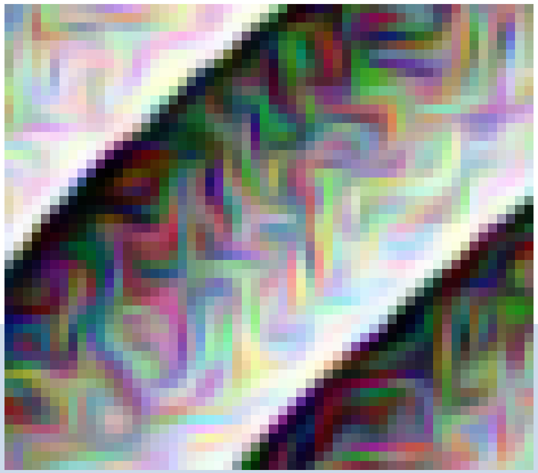
⇒ The maximum principle is verified (only local means of pixel intensities are computed).

⇒ Very stable and fast algorithm, compared to classical PDE implementations. The time step (dt) can be very large ($\simeq 50$) while process remains stable.

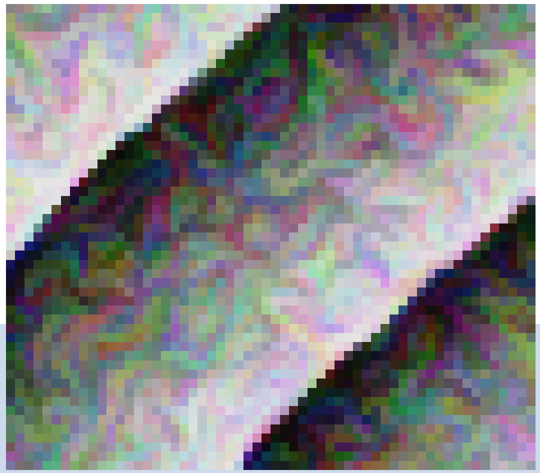
⇒ LIC-based numerical schemes allows a sub-pixel accuracy for the smoothing (4^{th}-order Runge-Kutta integration) ⇒ Better preservation of small structures.



(a) Original image



(b) PDE Regul. (explicit Euler scheme)



(c) LIC-base scheme

Presentation Layout

- Diffusion PDE's for Multi-Valued Images : A Quick Review
- Introducing Curvature-Preserving PDE's for Image Regularization
- Connections with Line Integral Convolutions and Implementation

⇒ **Application Results**

Application : Image Denoising

“Babouin” (détail) - 512x512 - (1 iter., 19s)

Application : Image Denoising

“Tunisie” - 555x367

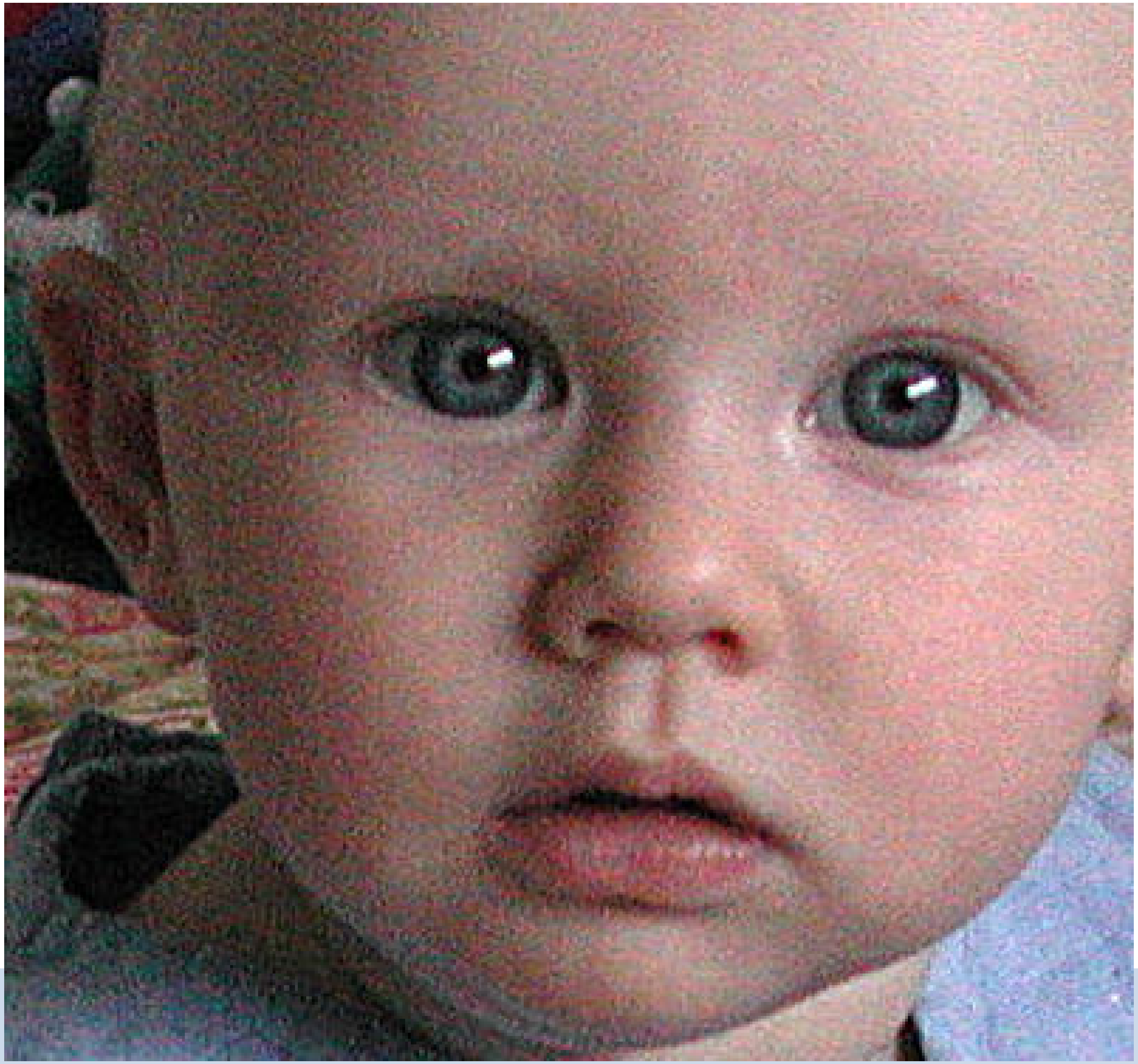
Application : Image Denoising

“Tunisie” - 555x367 - (1 iter., 11s)

Application : Image Denoising

"Tunisie" - 555x367 - (1 iter., 11s)

Application : Image Denoising

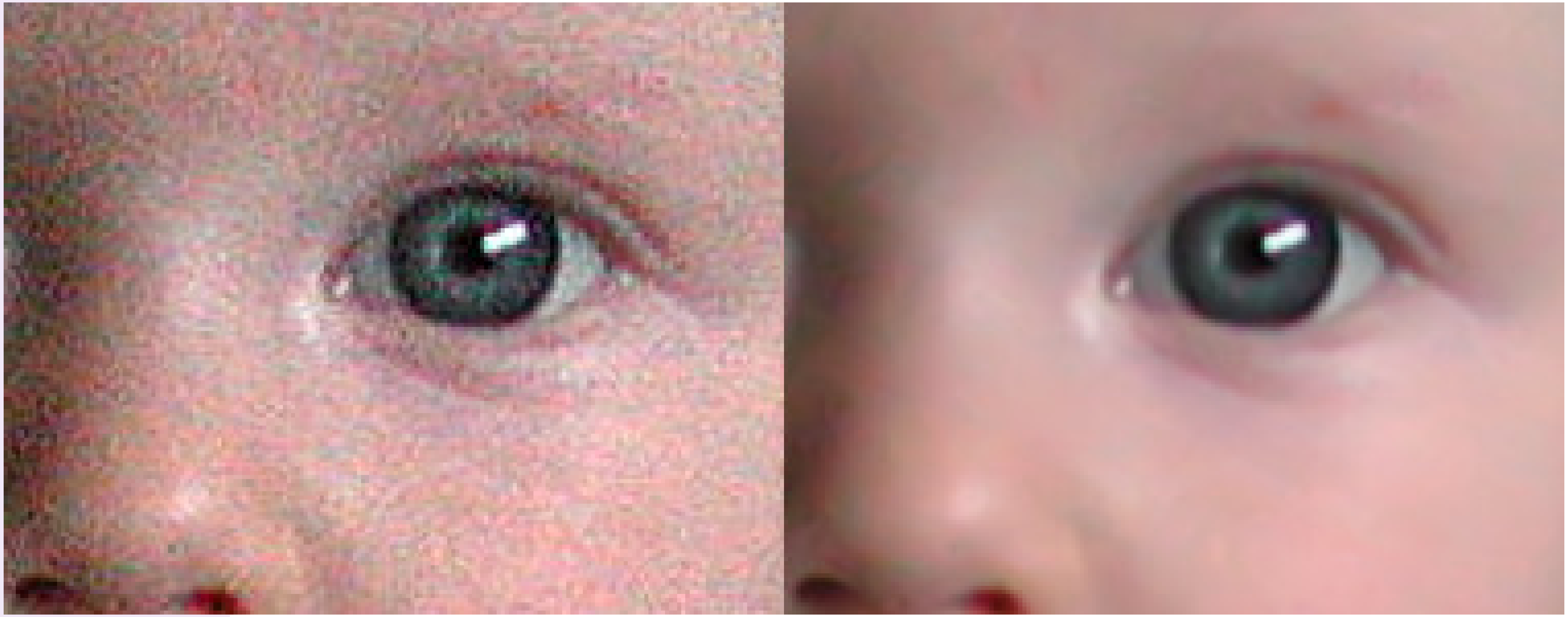


“Baby” - 400x375

Application : Image Denoising

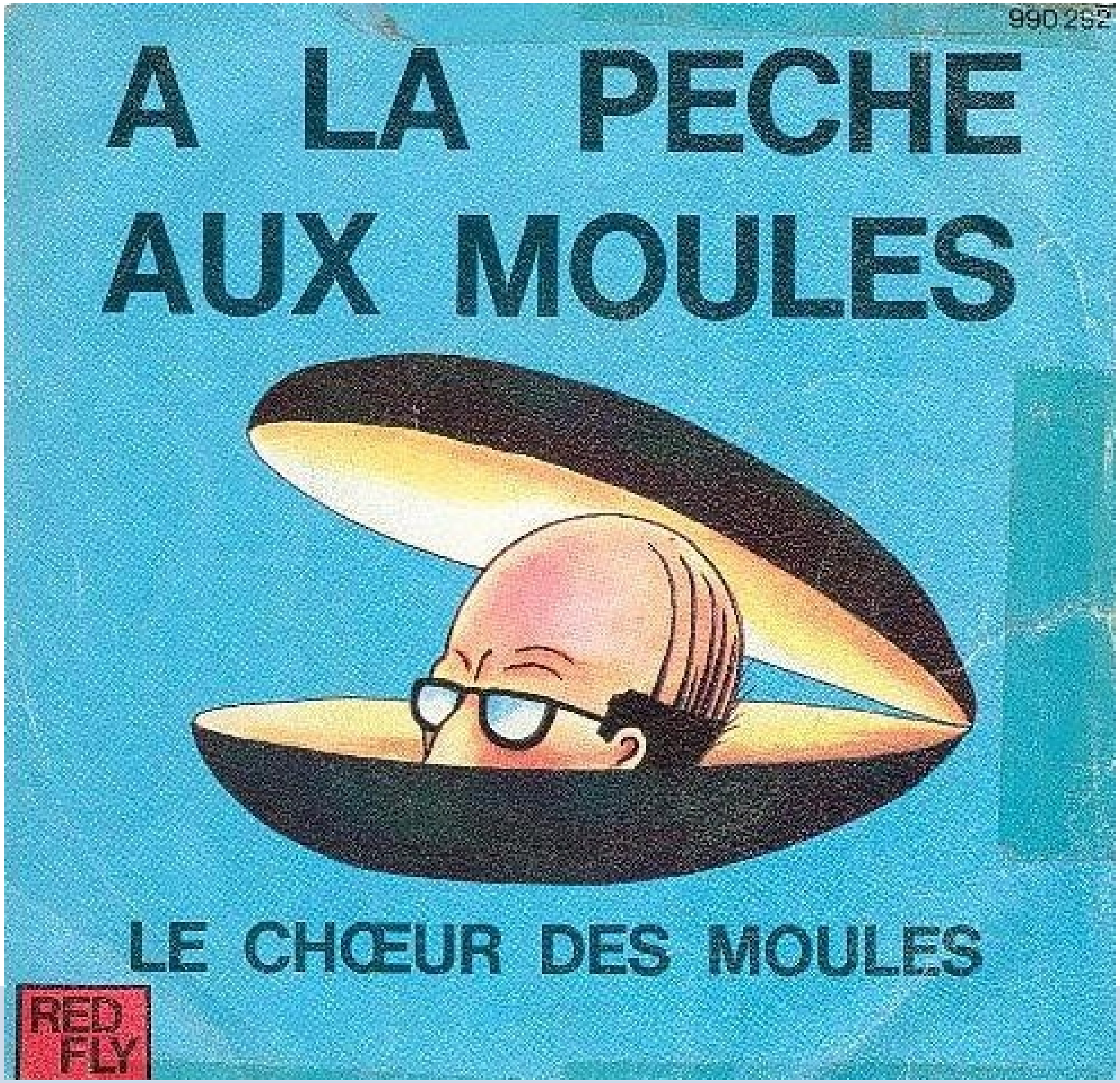
“Baby” - 400x375 - (2 iter, 5.8s)

Application : Image Denoising

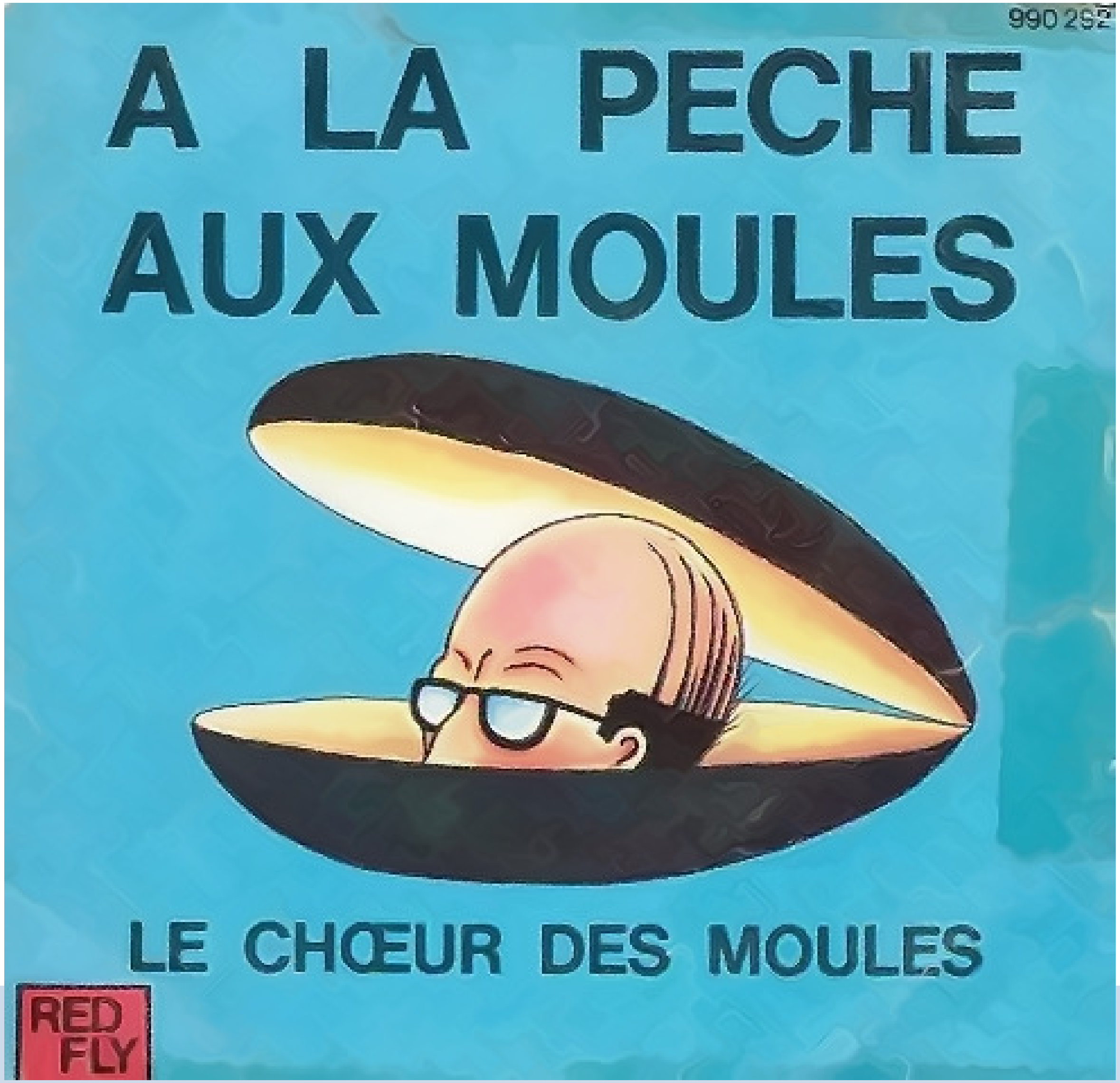


“Baby” - 400x375 - (2 iter, 5.8s)

Application : Image Denoising



“La pêche aux moules”.



“La pêche aux moules” (1 iter. 3.2s)).

Application : Image Denoising

“Chloé”

Application : Reducing JPEG artefacts

“Van Gogh”

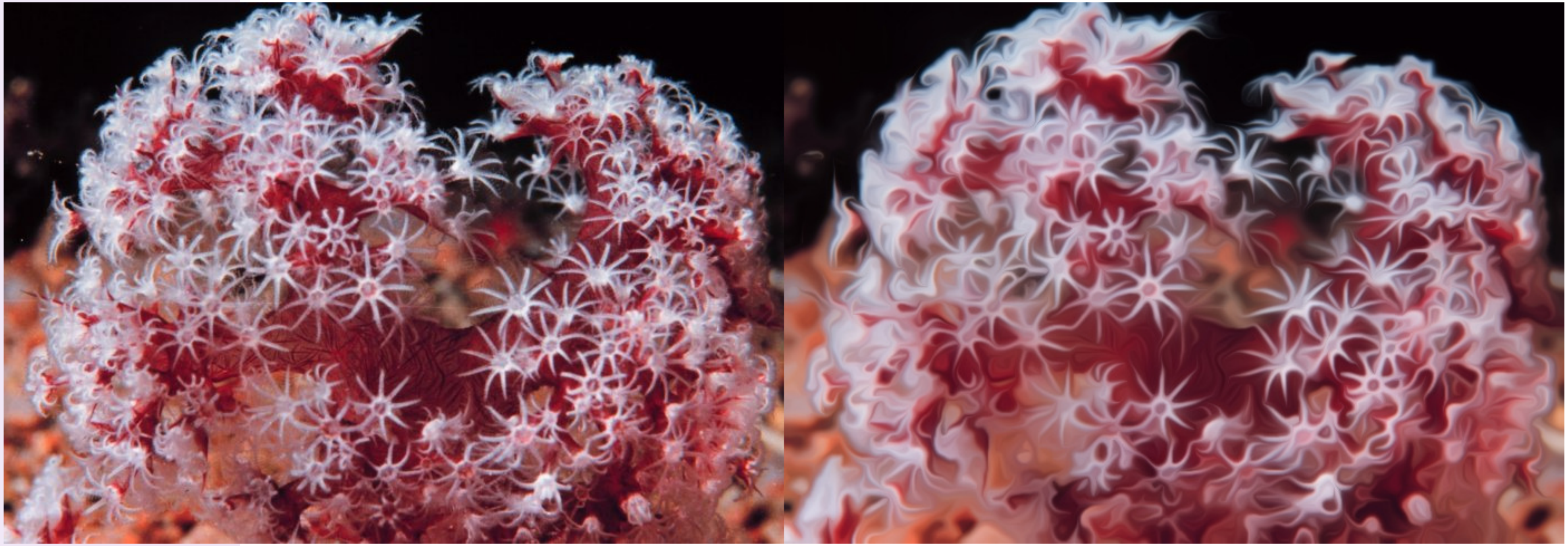
Application : Reducing JPEG artefacts

“Van Gogh” - (1 iter, 5.122s).

Application : Reducing JPEG artefacts

“Flowers” (JPEG, 10% quality).

Application : Creating Painting Effects



“Corail” (1 iter.)

Application : Image Inpainting

“Chloé au zoo”, original color image.

Application : Image Inpainting

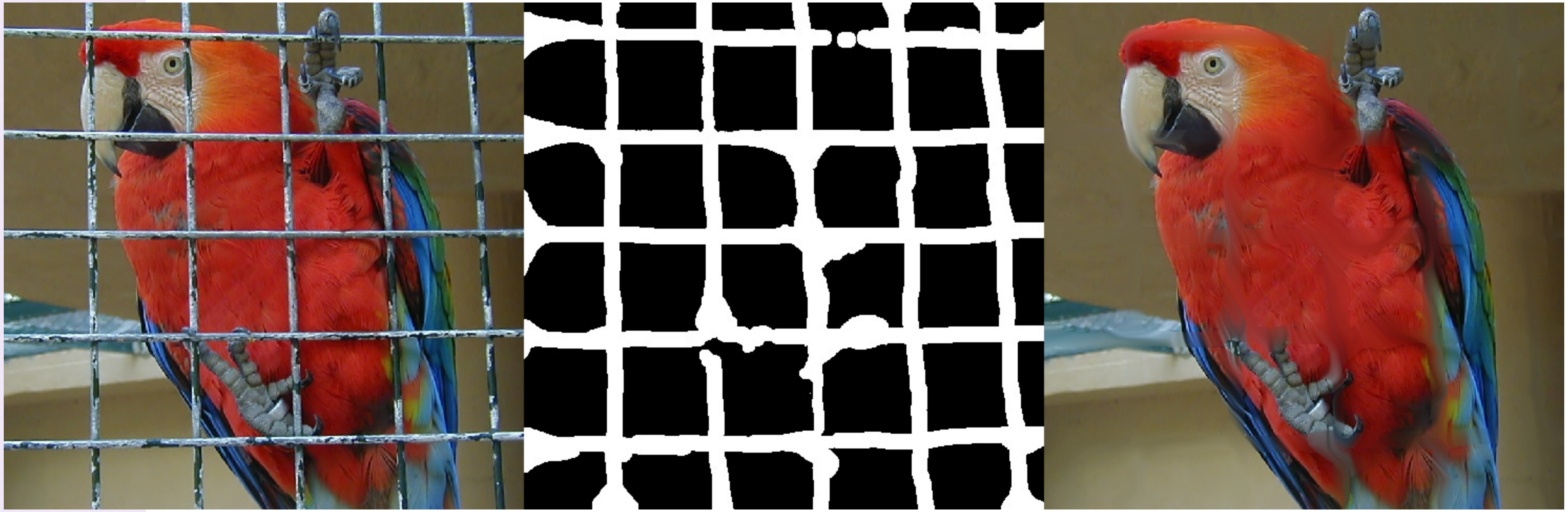
“Chloé au zoo”, inpainting mask definition.

Application : Image Inpainting

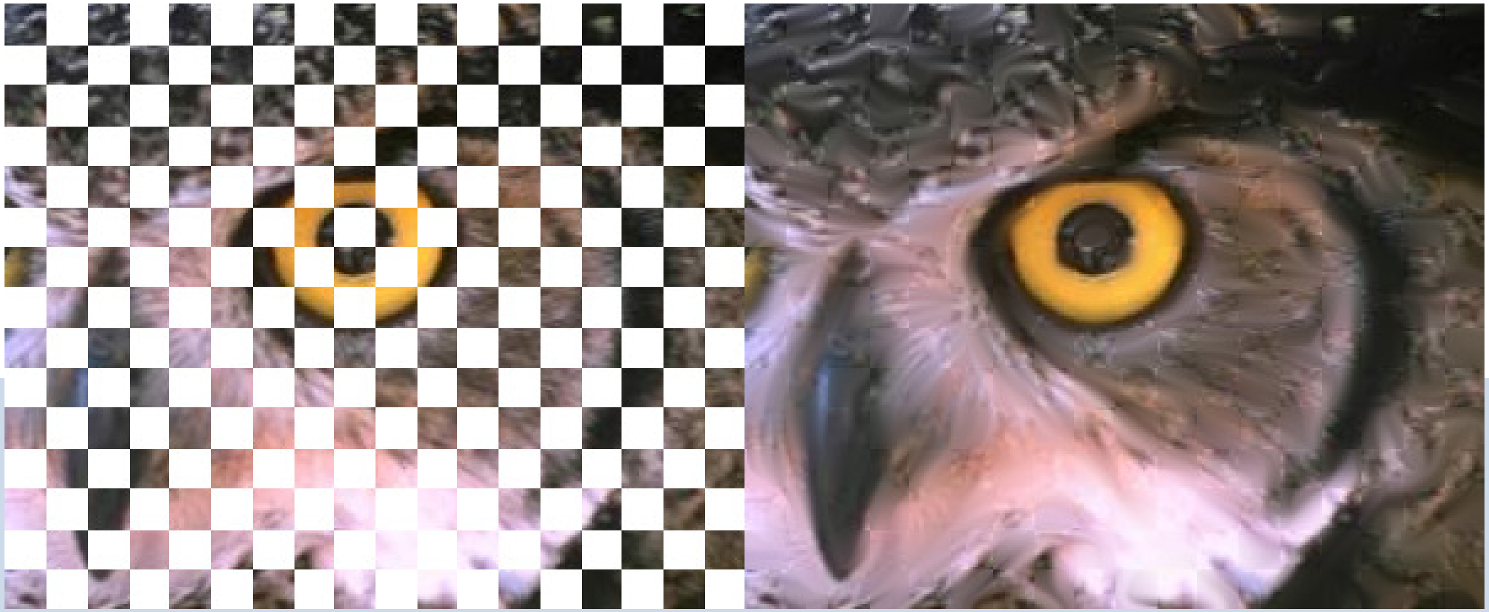
“Chloé au zoo”, inpainted with our PDE.

Application : Image Inpainting and Reconstruction

"Parrot"
500x500
(200 iter., 4m11s)



"Owl"
320x246
(10 iter., 1m01s)



Application : Image Inpainting

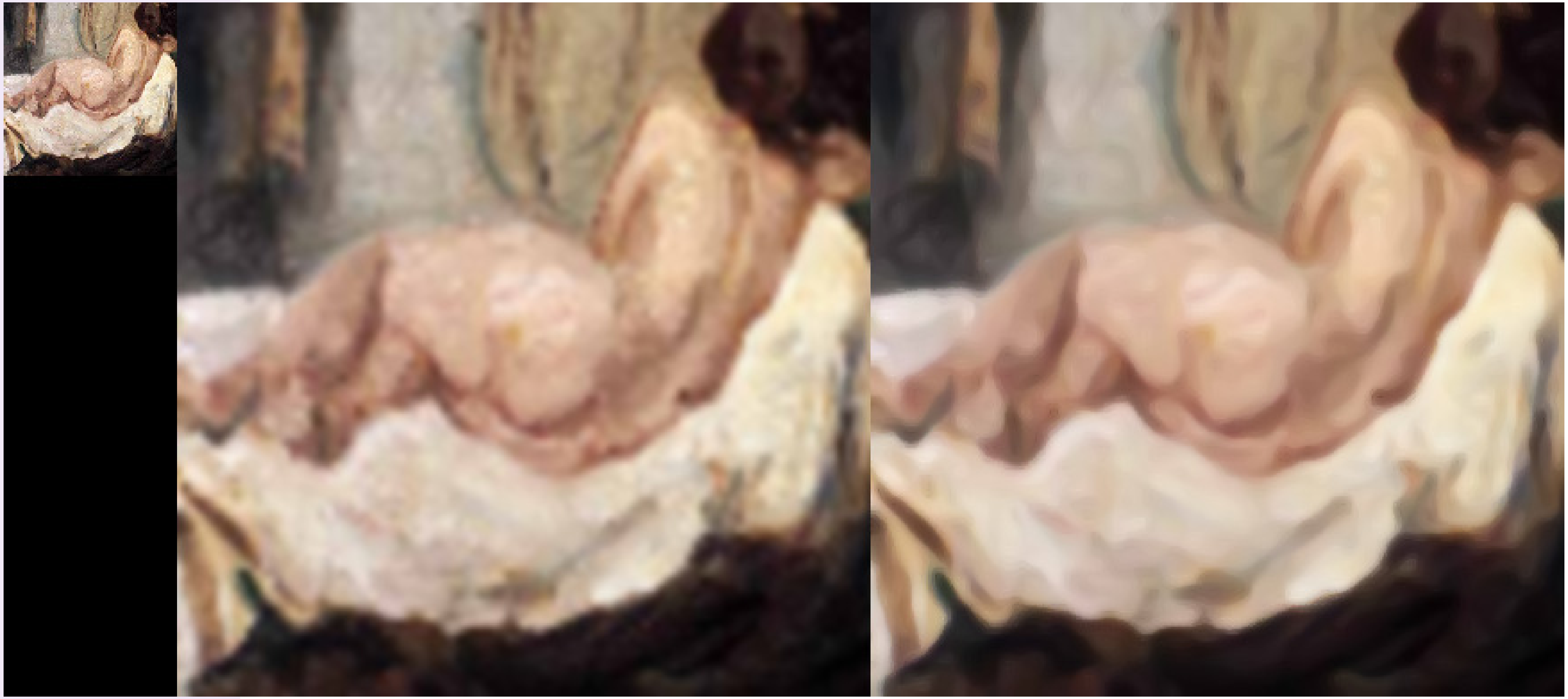
“Bird”, original color image.

“Bird”, inpainting mask definition.

Application : Image Inpainting

“Bird”, inpainted with our PDE.

Application : Image Resizing



“Nude” - (1 iter., 20s)

Application : Image Resizing

“Forest” - (1 iter., 5s)

Application : Image Resizing

(c) Details from the image resized by bicubic interpolation.

(d) Details from the image resized by a non-linear regularization PDE.

Application : Image Resizing

(a) Original color image

(b) Bloc Interpolation (c) Linear Interpolation (d) Bicubic Interpolation (e) PDE/LIC Interpolation

Summary

- Generic Tensor-driven PDE denoising : multi-channels, contour preservation.
- Preserve curvatures and small image structures very well.
- Very fast and precise implementation scheme.
- Algorithm 'GREYCSTORATION' available on the web :

http://www.greyc.ensicaen.fr/~dtschump/greycstoration/