Les cahiers du GREYC

David Tschumperlé

Fast Anisotropic Smoothing of Multi-Valued Images

using Curvature-Preserving PDE's

(revised version, May 2006)

Université de Caen - Campus II
Bd du Maréchal Juin, 14032 Caen Cedex - FRANCE
Tél : 02 31 56 73 31 - Fax : 02 31 56 73 30
E-mail : greyc@info.unicaen.fr

Ecole Nationale Supérieure d'Ingénieur de Caen
Bd du Maréchal Juin, 14050 Caen Cedex - FRANCE
Tél : 02 31 45 25 04 - Fax : 02 31 45 26 98
E-mail : greyc@ensicaen.fr

Abstract

We are interested in PDE's (Partial Differential Equations) in order to smooth multi-valued images in an anisotropic manner. Starting from a review of existing anisotropic regularization schemes based on diffusion PDE's, we point out the pros and cons of the different equations proposed in the literature. Then, we introduce a new tensor-driven PDE, regularizing images while taking the curvatures of specific integral curves into account. We show that this constraint is particularly well suited for the preservation of thin structures in an image restoration process. A direct link is made between our proposed equation and a continuous formulation of the LIC's (Line Integral Convolutions by Cabral and Leedom [11]). It leads to the design of a very fast and stable algorithm that implements our regularization method, by successive integrations of pixel values along curved integral lines. Besides, the scheme numerically performs with a sub-pixel accuracy and preserves then thin image structures better than classical finite-differences discretizations. Finally, we illustrate the efficiency of our generic curvature-preserving approach - in terms of speed and visual quality - with different comparisons and various applications requiring image smoothing : color images denoising, inpainting and image resizing by nonlinear interpolation.

Keywords : Multi-valued Images, Data Regularization, Anisotropic Smoothing, Diffusion PDE's, Tensor-valued Geometry, Denoising, Inpainting, Nonlinear Interpolation.

1   Introduction

Obtaining regularized versions of noisy or corrupted image data has always been a desirable goal in the fields of computer vision and image processing. It is useful, either to restore degraded images (which is the most direct application of image regularization methods) or - more indirectly - as a pre-processing step that eases further analysis of the considered data. Regularization is actually one of the key operations needed by many image analysis algorithms. A lot of image regularization formalisms have been then already proposed in the literature for this purpose.

Since the pioneering work of Perona-Malik [33] in the early 90's, the framework of anisotropic diffusion PDE's (Partial Differential Equations) has particularly raised a strong interest for data regularization : such equations have the ability to smooth data in a nonlinear way, allowing the preservation of significant image discontinuities. PDE's are local formulations and thus, they are well adapted to deal with degraded images where sources of data corruption are local or semi-local too : gaussian noise, scratches or compression artefacts are local degradations usually encountered in digital (original or digitized) images. Therefore, many variants of diffusion PDE's have been proposed so far for the restoration of image datasets. In particular, important contributions in this field concern the way the classical isotropic diffusion equation (heat flow) has been extended to deal with anisotropic smoothing [33, 27, 37, 52], how diffusion PDE's may be seen as gradient descents of various energy functionals [4, 13, 16, 23, 36], and the link between regularization PDE's and the concept of non-linear scale spaces [1, 28, 30]. Extensions of these techniques to color images and more generally multi-valued datasets have been also tackled in [38, 44, 48, 53]. More recently, regularization PDE's under constraints have been proposed in order to deal with more specific datasets, as fields of unit vectors [18, 24, 32, 41], orthonormal matrices [17, 45], positive-definite matrices [17, 46], or image data defined on implicit surfaces [7, 14, 42].

Despite this wide range of existing constrained and unconstrained PDE formalisms, all regularization methods have something in common : they locally *smooth* the image along one or several directions of the plane that are different at each image point. Typically, the principal smoothing directions are chosen to be parallel to the image contours, resulting in an *anisotropic* regularization that does not destroy edges. As a requirement, defining a correct *smoothing behavior* is one of the first aim of a good regularization algorithm, the second being the precision of the smoothing process itself : it must respect the defined smoothing geometry as much as possible.

Following this general principle, authors of [48, 52] recently proposed two different PDE-based frameworks able to design specific regularization processes from a given (user-defined) underlying local smoothing geometry. These methods have two main interests : on one hand, they unify a lot of previously proposed equations into generic diffusion PDE's and provide a local *geometric interpretation* of the corresponding regularizations. On the other hand, they clearly separate the design of the smoothing geometry from the smoothing process itself : in a first step, one retrieves the geometry of the structures inside the image (generally by the computation of the so-called *structure tensor field* **G**). Then, a local geometry of the desired smoothing is defined by the mean of a second field **T** of *diffusion tensors* (depending on **G**). Finally, one step of the smoothing process itself (driven by **T**) is performed through one or several iterations of a specific diffusion PDE. This procedure is repeated until the image is regularized enough.

In this article, we first review these two efficient and unifying regularization methods acting on unconstrained multi-valued

images, following our interpretation of separating the smoothing from the geometry (section 2). We particularly point out the advantages and drawbacks of each equation in real cases. We propose then a comparable tensor-driven diffusion PDE that regularizes multi-valued images while respecting specific *curvature constraints* (section 3). Actually, our equation is mathematically positioned between the two previous formulations, in a way that it solves the issues inherent to both methods. Moreover, we propose a theoretical interpretation of our curvature-constrained formalism in terms of LIC's (Line Integral Convolutions [11]). This analogy leads to the proposal of a novel numerical scheme that implements our PDE (section 4), by successive integrations of pixel values along integral lines. This iterative scheme has two main advantages compared to classical PDE implementations : on one hand, it preserves the orientations of thin image structures, since it naturally works at a sub-pixel accuracy. On the other hand, the algorithm is able to run up to three times faster than classical explicit scheme since it is unconditionally stable, even for large PDE time steps. Finally, we illustrate the effectiveness of our curvature-preserving method, in terms of computational speed and visual quality, with results on color image restoration, color image inpainting and non-linear resizing, among all possible applications in the area of image regularization (section 5).

2   Anisotropic Smoothing of Images with PDE's : A Review

Let us consider a multi-valued image $\mathbf{I} : \Omega \to \mathbb{R}^n$ ($n = 3$ for color images) corrupted by noise and defined on a domain $\Omega \subset \mathbb{R}^2$. We denote by $I_i : \Omega \to \mathbb{R}$, the scalar channel i of $\mathbf{I}$: $\forall \mathbf{X} = (x, y) \in \Omega$, $\mathbf{I}_{(\mathbf{X})} = \begin{pmatrix} I_{1(\mathbf{X})} & I_{2(\mathbf{X})} & ... & I_{n(\mathbf{X})} \end{pmatrix}^T$. Regularizing $\mathbf{I}$ can be done by one among the large variety of existing diffusion PDE's. We will focus anyway on the recent works in [48, 52], which are unifying approaches.

2.1   Local Geometry and Diffusion Tensors

Basically, PDE-based regularization may be seen as the local smoothing of an image $\mathbf{I}$ along defined directions depending themselves on the local configuration of the pixel intensities. One wants to smooth $\mathbf{I}$ while preserving its edges (discontinuities in image intensities), i.e. performs a local smoothing mostly along directions of the edges, avoiding smoothing orthogonally to these edges. Naturally, this means that one has first to retrieve the *local geometry* of the image $\mathbf{I}$. It consists in the definition of these important features at each image point $\mathbf{X} = (x, y) \in \Omega$:

- Two orthogonal directions $\theta^+_{(\mathbf{X})}$, $\theta^-_{(\mathbf{X})} \in S^1$ (unit vectors of $\mathbb{R}^2$) directed along the local maximum and minimum variations of image intensities at $\mathbf{X}$. The direction θ^- generally corresponds to the edge direction, when there is one.

- Two corresponding positive values $\lambda^+_{(\mathbf{X})}$, $\lambda^-_{(\mathbf{X})}$ measuring effective variations of the image intensities along $\theta^+_{(\mathbf{X})}$ and $\theta^-_{(\mathbf{X})}$ respectively. λ^-, λ^+ are related to the local *strength* of an edge.

For scalar images $I : \Omega \to \mathbb{R}$, this local geometry $\{ \lambda^{+/-}, \theta^{+/-} \mid \mathbf{X} \in \Omega \}$ is usually retrieved by the computation of the gradient field ∇I, or smoothed gradient field $\nabla I_\sigma = \nabla I * G_\sigma$ where G_σ is a 2D gaussian kernel, with a variance σ. Thus, $\lambda^+ = \|\nabla I_\sigma\|^2$ is a possible measure of the local strength of the contours, while $\theta^- = \nabla I_\sigma^\perp / \|\nabla I_\sigma\|$ gives the contours direction. It is worth to notice that $\{ \lambda^{+/-}, \theta^{+/-} \mid \mathbf{X} \in \Omega \}$ can be represented in a more convenient form by a field $\mathbf{G}$: $\Omega \to P(2)$ of 2×2 symmetric and semi-positive matrices, named *tensors* : $\forall \mathbf{X} \in \Omega$, $\mathbf{G}_{(\mathbf{X})} = \lambda^- \; \theta^- \theta^{-T} + \lambda^+ \; \theta^+ \theta^{+T}$. Eigenvalues of $\mathbf{G}$ are indeed λ^- and λ^+ and corresponding eigenvectors are θ^- and θ^+. For instance, the local geometry of scalar-valued images I can be expressed with the tensor $\mathbf{G}_{(\mathbf{X})} = \nabla I_{(\mathbf{X})} \nabla I^T_{(\mathbf{X})}$.

For multi-valued images $\mathbf{I} : \Omega \to \mathbb{R}^n$, the local geometry can be retrieved in a similar way, by the computation of the field $\mathbf{G}$ of *structure tensors*. As noticed in [21, 52], this extends naturally the gradient for multi-valued images :

$$\forall \mathbf{X} \in \Omega, \quad \mathbf{G}_{(\mathbf{X})} = \sum_{i=1}^{n} \nabla I_{i(\mathbf{X})} \nabla I^T_{i(\mathbf{X})} \quad \text{where} \quad \nabla I_i = \begin{pmatrix} \frac{\partial I_i}{\partial x} \\ \\ \frac{\partial I_i}{\partial y} \end{pmatrix} \tag{1}$$

A gaussian-smoothed version $\mathbf{G}_\sigma = \mathbf{G} * G_\sigma$ is usually computed to retrieve a more coherent geometry. $\mathbf{G}_{\sigma(\mathbf{X})}$ is a good estimator of the local multi-valued geometry of $\mathbf{I}$ at $\mathbf{X}$: its spectral elements give at the same time the vector-valued variations (by the eigenvalues λ^-, λ^+ of $\mathbf{G}_\sigma$) and the orientations (edges) of the local image structures (by the eigenvectors $\theta^- \perp \theta^+$ of $\mathbf{G}_\sigma$), σ being proportional to the so-called noise scale.

Once the local geometry $\mathbf{G}_\sigma$ of $\mathbf{I}$ has been determined this way, authors of [48, 52] proposed to design a particular field $\mathbf{T} : \Omega \to \mathrm{P}(2)$ of *diffusion tensors* which specifies the local smoothing geometry that should drive the regularization process. Of couse, $\mathbf{T}$ depends on the local geometry of $\mathbf{I}$, and is thus defined from the spectral elements λ^-, λ^+ and θ^-, θ^+ of $\mathbf{G}_\sigma$:

$$\forall \mathbf{X} \in \Omega, \quad \mathbf{T}_{(\mathbf{X})} = f^-_{(\lambda^+, \lambda^-)} \, \theta^- \theta^- + f^+_{(\lambda^+, \lambda^-)} \, \theta^+ \theta^{+T} \tag{2}$$

Basically, $f^{+/-} : \mathbb{R}^2 \to \mathbb{R}$ designates two functions which set the strengths of the desired smoothing along the respective directions θ^-, θ^+. Several choices for f^-, f^+ are possible, depending on the considered application. For image denoising, a possible choice is (proposed in [16, 44, 48]) :

$$f^-_{(\lambda_+, \lambda_-)} = \frac{1}{(1 + \lambda^+ + \lambda^-)^{p_1}} \quad \text{and} \quad f^+_{(\lambda_+, \lambda_-)} = \frac{1}{(1 + \lambda^+ + \lambda^-)^{p_2}} \quad \text{with} \quad p_1 < p_2$$

At this point, the desired smoothing behavior is intended to be :

- If a pixel $\mathbf{X}$ is located on an image contour ($\lambda^+_{(\mathbf{X})}$ is high), the smoothing on $\mathbf{X}$ would be performed mostly along the contour direction $\theta^-_{(\mathbf{X})}$ (since $f^+_{(.,.)} << f^-_{(.,.)}$), with a smoothing strength inversely proportional to the contour strength.

- If a pixel $\mathbf{X}$ is located on a homogeneous region ($\lambda^+_{(\mathbf{X})}$ is low), the smoothing on $\mathbf{X}$ would be performed in all possible directions (isotropic smoothing), since $f^+_{(.,.)} \simeq f^-_{(.,.)}$ and then $\mathbf{T} \simeq \mathbb{I}_d$ (identity matrix).

This is one possible choice for f^-, f^+ in order to satisfy basic image denoising requirements. In [52], the same kind of considerations leads to similar diffusion functions. Actually, this is quite natural to design a smoothing behavior from the image structure *before* applying the regularization process itself.

Pre-defining the smoothing geometry $\mathbf{T}$ for each PDE iteration is the first stage of regularization algorithms proposed in [48, 52]. The corresponding smoothing must be applied then. The important differences between all existing regularization methods lie first on the definition of $\mathbf{T}$, but also on the form of the diffusion PDE that will be used to perform the smoothing. Choosing different smoothing functions f^-, f^+ and diffusion PDE's detailed below leads to the unification of most unconstrained image regularization methods proposed in the literature [1, 4, 7, 8, 13, 15, 16, 23, 28, 30, 33, 36, 37, 38].

2.2   The divergence-based PDE

Considering a corrupted multi-valued image $\mathbf{I} : \Omega \to \mathbb{R}^n$ and a local smoothing geometry $\mathbf{T} : \Omega \to \mathrm{P}(2)$ defined as a field of diffusion tensors (2), the following divergence PDE can be used to anisotropically smooth $\mathbf{I}$ "along" $\mathbf{T}$:

$$\forall i = 1, .., n, \qquad \frac{\partial I_i}{\partial t} = \mathrm{div}\left(\mathbf{T}\nabla I_i\right) \tag{3}$$

This classical equation in PDE-based regularization has been introduced by Weickert in [52], and adapted for color/multivalued images in [53]. Note that the tensor field $\mathbf{T}$ is the same for all image channels I_i, ensuring that all I_i are smoothed by a *common multi-valued geometry* which takes the correlation between image channels into account (since $\mathbf{T}$ depends on $\mathbf{G}$), contrary to a uncorrelated channel-by-channel approach. The notable characteristics of (3) are :

(a) Pros : It unifies a lot of existing scalar or multi-valued regularization approaches and proposes at the same time two interpretation levels of the regularization process :

- *local interpretation* : (3) may be seen as the physical law describing local diffusion processes of the pixels individually regarded as temperatures or chemical concentrations in an anisotropic environment which is locally described by $\mathbf{T}$.

- *global interpretation* : the problem of image regularization is often expressed as the minimization of a specific energy functional $E(\mathbf{I})$, depending on the spatial variations of $\mathbf{I}$ [4, 7, 13, 14, 16, 17, 23]. Finding $\mathbf{I}$ that minimizes $E(\mathbf{I})$ is usually done by a gradient descent (i.e. a PDE), coming from the Euler-Lagrange equations of $E(\mathbf{I})$, resulting in a particular case of (3). In [44, 48], we demonstrated that the minimization of the general multi-valued ψ-functional

$$E(\mathbf{I}) = \int_\Omega \psi(\lambda^+, \lambda^-) \, d\Omega \qquad \text{where} \quad \psi : \mathbb{R}^2 \to \mathbb{R} \tag{4}$$

is done by the divergence PDE (3) with $\mathbf{T} = \frac{\partial \Psi}{\partial \lambda^-} \, \theta^- \theta^{-T} + \frac{\partial \Psi}{\partial \lambda^+} \, \theta^+ \theta^{+T}$. In this case, the λ_+, λ_- are the two positive eigenvalues of the *non-smoothed* structure tensor field $\mathbf{G} = \sum_i \nabla I_i \nabla I_i^T$, while the θ_+, θ_- are the two corresponding orthonormal eigenvectors of $\mathbf{G}$. Similar results have been demonstrated for scalar-valued images [4, 16, 26] (and references therein).

(b) Cons : Strictly speaking, the PDE (3) *does not fully respect the geometry* $\mathbf{T}$. The smoothing performed is not always the one that could be expected. We illustrate this fact by considering the simple case of single direction smoothing. Suppose we want to anisotropically smooth a scalar image $I : \Omega \to \mathbb{R}$ everywhere along the gradient direction $\frac{\nabla I}{\|\nabla I\|}$ with a constant strength 1. This is of course for illustration purposes, since all image discontinuities would be destroyed with such a smoothing geometry. Intuitively, we should define $\mathbf{T}$ as : $\forall \mathbf{X} \in \Omega, \quad \mathbf{T}_{(\mathbf{X})} = \left(\frac{\nabla I}{\|\nabla I\|} \right) \left(\frac{\nabla I}{\|\nabla I\|} \right)^T$, leading to the simplification of (3) as $\frac{\partial I}{\partial t} = \mathrm{div} \left(\frac{1}{\|\nabla I\|^2} \nabla I \nabla I^T \nabla I \right) = \mathrm{div} \left(\nabla I \right) = \Delta I$, where $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ stands for the Laplacian of I. As noticed in [25], the evolution of this so-called *heat flow equation* is similar to the convolution of the image I by a normalized gaussian kernel G_σ with a variance $\sigma = \sqrt{2\,t}$. This choice of *anisotropic* tensors $\mathbf{T}$ leads to an *isotropic* smoothing, without preferred directions. Note that choosing $\mathbf{T} = \mathbb{I}_d$ (identity matrix) would give exactly the same result : different tensors fields $\mathbf{T}$ with very different shapes (isotropic or anisotropic) define the same regularization behavior. Indeed, the divergence is a differential operator, so (3) implicitly depends on the *spatial variations* of $\mathbf{T}$. Thus, the divergence equation (3) hampers the design of a pointwise smoothing behavior (see [44, 48] for more details on this particular point).

2.3 The trace-based PDE

In order to respect the local smoothing geometry $\mathbf{T}$, we have proposed in [44, 48] a regularization PDE, very similar to the divergence equation (3), but based on a *trace* operator :

$$\forall i = 1, .., n, \qquad \frac{\partial I_i}{\partial t} = \mathrm{trace}\left(\mathbf{T} \mathbf{H}_i \right) \qquad \text{with} \qquad \mathbf{H}_i = \begin{pmatrix} \frac{\partial^2 I_i}{\partial x^2} & \frac{\partial^2 I_i}{\partial x \partial y} \\[2mm] \frac{\partial^2 I_i}{\partial x \partial y} & \frac{\partial^2 I_i}{\partial y^2} \end{pmatrix} \tag{5}$$

$\mathbf{H}_i$ stands for the Hessian of I_i. The equation (5) is a tensor-based expression of the following PDE, expressed with simultaneous oriented and weighted $1D$ Laplacians :

$$\frac{\partial \mathbf{I}}{\partial t} = f^-_{(\lambda^-, \lambda^+)} \, \mathbf{I}_{\theta^- \theta^-} + f^+_{(\lambda^-, \lambda^+)} \, \mathbf{I}_{\theta^+ \theta^+}$$

where $\mathbf{I}_{\theta^- \theta^-} = \frac{\partial^2 \mathbf{I}}{\partial \theta^{-2}}$ represents the second directional derivative of $\mathbf{I}$ along θ^- (the same for θ^+). Particular cases of (5) have been proposed in [4, 26, 27, 12, 37, 38, 44, 48] for scalar or multi-valued images. Note that each channel I_i of $\mathbf{I}$ is also smoothed with a common tensor field $\mathbf{T}$.

(a) Pros : As demonstrated in [44, 48], the evolution of (5) has an interesting geometric interpretation in terms of local filtering with oriented and normalized gaussian kernels. It may be seen locally as the application of a very small convolution around each $\mathbf{X}$ with a gaussian mask $G_t^{\mathbf{T}}$ *oriented* by the tensor $\mathbf{T}_{(\mathbf{X})}$:

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

This ensures that the smoothing performed by (5) is truly oriented along the pre-defined smoothing geometry $\mathbf{T}$. As the trace is not a differential operator, the spatial variation of $\mathbf{T}$ does not trouble the diffusion directions here and two different tensor fields will necessarily lead to different smoothing behaviors. Note that under certain conditions, the divergence PDE (3) may be also developed as a trace formulation (5). In this case, the tensors inside the trace and the divergence *are not the same* [44, 48].

(b) Cons : Contrary to the divergence formulations (3), trace-based equations (5) are very local formulations and thus, are rarely connected to global formulations expressed with energy functionals such as (4). This is particularly true when considering multi-valued images, despite recent papers tried to explore such links [44, 48]. For scalar-valued images ($n = 1$),

some correspondences are known anyway [4, 16, 20, 26]. In the sequels, we will mainly focus on the local behavior of regularization PDE's.

Note that the trace equation (5) behaves locally as an oriented gaussian smoothing whose strength and orientation is directly related to the tensor $\mathbf{T}_{(\mathbf{X})}$. But on curved structures (like corners), this gaussian behavior is not desirable : when the local variation of the edge orientation θ^- is high, a gaussian filter tends to *round* corners, even by conducting it only along θ^-. This is due to the fact that an oriented gaussian mask is not curved itself. This classical behavior is also best known as the "mean curvature flow" effect, characterized by the PDE $\frac{\partial \mathbf{I}}{\partial t} = \frac{\partial^2 \mathbf{I}}{\partial \theta^{-2}}$. This problem is illustrated on Fig.1b and Fig.2b where (5) has been applied on synthetic and real color image and $\mathbf{T}$ has been defined as (2) (then $f^- \neq 0$). One can easily see how image structures are subject to the mean curvature flow effect, resulting in rounding the corners of the square in Fig.1b, or in blending parallel thin curved structures in Fig.2b.

To avoid this over-smoothing effect, most regularization PDE's try to stop their action on corners (by vanishing tensors $\mathbf{T}_{(\mathbf{X})}$ there, i.e $f^- = f^+ = 0$). But this implies the detection of curved structures on noisy or corrupted images, which is generally a hard task. Conversely, image under-smoothing on edges may occur when limiting the diffusion too much on regions with high intensity variations (Fig.1c). There is a difficult trade-off between complete noise removal and preservation of curved structures, when using trace-based PDE's (5).

(a) Noisy synthetic color image

(b) Applying trace-based PDE (5), with $p_1 = 0.5$, $p_2 = 1.2$.

(c) Applying trace-based PDE (5), with $p_1 = 0.9$, $p_2 = 1.2$.

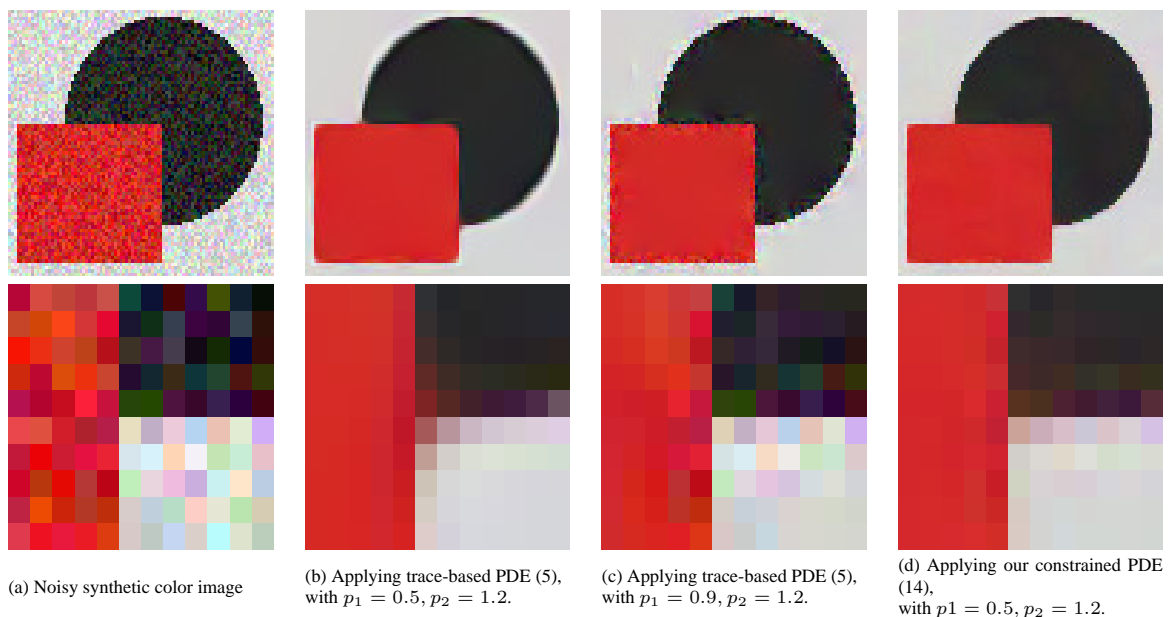(d) Applying our constrained PDE (14), with $p1 = 0.5$, $p_2 = 1.2$.

Figure 1: Problems encountered when using trace-based PDE's (5) on curved image structures (details are shown on the second row).

Actually, this kind of regularization processes does not care about the *curvature* of the smoothing directions, and by extension, of the curvature of the image contours. Taking this curvature into account is a very desirable goal and has motivated the work presented in the sequels : in section 3, we propose a new class of trace-based regularization PDE's that smooth an image $\mathbf{I}$ along a tensor field $\mathbf{T}$, *while implicitly taking curvatures of specific integral curves of $\mathbf{T}$ into account*. Roughly speaking, we want to locally filter the image with *curved gaussian kernels* when necessary, in order to better preserve image structures. For illustration purposes, results of our curvature-preserving equation is shown on Fig.1d and Fig.2c.*

3 Curvature-Preserving PDE's

3.1 The single direction case

To illustrate the general idea of curvature-preserving PDE's, we first focus on image regularization along a *vector field* $\mathbf{w} : \Omega \to \mathbb{R}^2$ instead of a tensor field $\mathbf{T}$. We consider then a local smoothing everywhere along a single direction $\frac{\mathbf{w}}{\|\mathbf{w}\|}$, with a smoothing strength $\|\mathbf{w}\|$. We denote the two spatial components of $\mathbf{w}$ by $\mathbf{w}_{(\mathbf{X})} = (u_{(\mathbf{X})} \; v_{(\mathbf{X})})^T$.

(a) Image of a fingerprint

(b) Applying trace-based PDE (5), with $p_1 = 0.5$, $p_2 = 1.2$.

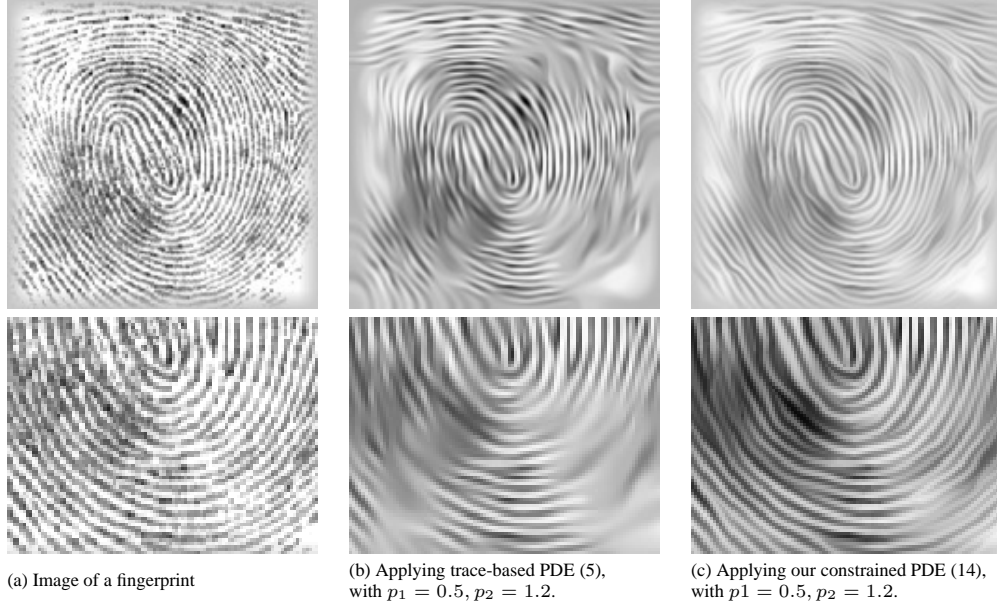(c) Applying our constrained PDE (14), with $p1 = 0.5$, $p_2 = 1.2$.

Figure 2: Comparisons between trace-based PDE's (5) and our new curvature-preserving PDE's (14) on a real image.

We propose to define the following *curvature-preserving* regularization PDE that smoothes $\mathbf{I}$ along $\mathbf{w}$ by :

$$\forall i = 1, \ldots, n, \qquad \frac{\partial I_i}{\partial t} = \text{trace}\left(\mathbf{w}\mathbf{w}^T \, \mathbf{H}_i\right) + \nabla I_i^T \mathbf{J}_\mathbf{w} \mathbf{w} \tag{6}$$

where $\mathbf{J}_\mathbf{w}$ stands for the Jacobian of $\mathbf{w}$, and $\mathbf{H}_i$ is the Hessian of I_i.

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

The PDE (6) adds a term $\nabla I_i^T \mathbf{J}_\mathbf{w} \mathbf{w}$ to the trace-based equation (5) that smoothes $\mathbf{I}$ along $\mathbf{w}$ with locally oriented gaussian kernels (see section 2.3). This extra term naturally depends on the variation of the vector field $\mathbf{w}$. Let us explain how (6) is related to $\mathbf{w}$.

Let $\mathcal{C}_{(a)}^\mathbf{X}$ be the curve defining the *integral curve* of $\mathbf{w}$, starting from $\mathbf{X}$ and parameterized by $a \in \mathbb{R}$:

$$\begin{cases} \mathcal{C}_{(0)}^\mathbf{X} & = \quad \mathbf{X} \\ \\ \frac{\partial \mathcal{C}_{(a)}^\mathbf{X}}{\partial a} & = \quad \mathbf{w}(\mathcal{C}_{(a)}^\mathbf{X}) \end{cases} \tag{7}$$

When $a \to +\infty$ the integral curve $\mathcal{C}_{(a)}^\mathbf{X}$ is tracked *forward*, and *backward* when $a \to -\infty$ (Fig.3). We denote by $\mathcal{F}$ the family of integral curves of $\mathbf{w}$.

A second-order Taylor development of $\mathcal{C}_{(a)}^\mathbf{X}$ around $a = 0$ is :

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

(a) Integral curve of a general field $\mathbf{w}$.

(b) Example of integral curves when $\mathbf{w}$ is the lowest eigenvector of the structure tensor $\mathbf{G}$ of a color image $\mathbf{I}$ (one block is one color pixel).
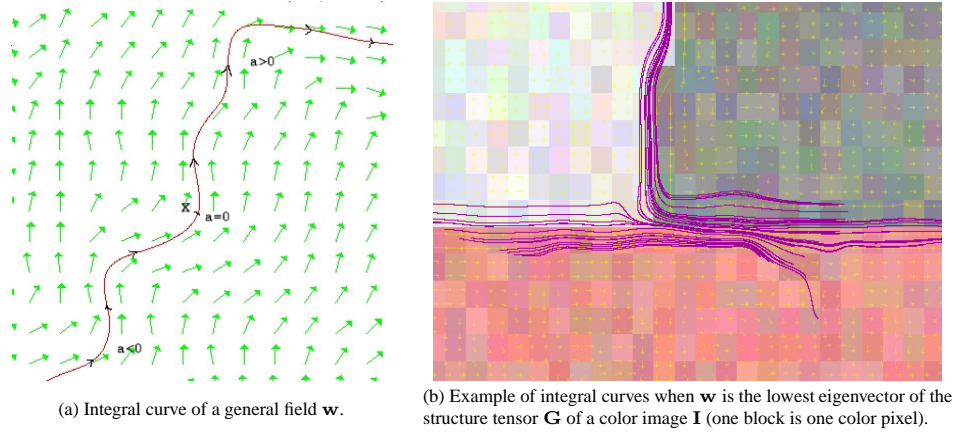
Figure 3: Integral curve $\mathcal{C}^{\mathbf{X}}$ of vector fields $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$.

with $h \rightarrow 0$, and $O(h^n) = h^n \, \epsilon_n$. Then, we can compute a second-order Taylor development of $I_i(\mathcal{C}^{\mathbf{X}}_{(a)})$ around $a = 0$, which corresponds to the variations of the image intensity near $\mathbf{X}$ when following the integral curve $\mathcal{C}^{\mathbf{X}}$:

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

The term $\text{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) = \frac{\partial^2 I_i}{\partial \mathbf{w}^2}$ corresponds to the second directional derivative of I_i along $\mathbf{w}$.

The second derivative of the function $a \rightarrow I_i(\mathcal{C}^{\mathbf{X}}_{(a)})$ at $a = 0$ is then :

$$
\begin{aligned}
\frac{\partial^2 I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial a^2}_{|a=0} &= \lim_{h \rightarrow 0} \frac{1}{h^2} \left[I_i(\mathcal{C}^{\mathbf{X}}_{(h)}) + I_i(\mathcal{C}^{\mathbf{X}}_{(-h)}) - 2I_i(\mathcal{C}^{\mathbf{X}}_{(0)}) \right] \\
&= \lim_{h \rightarrow 0} \frac{1}{h^2} \left[h^2 \, \nabla I_i^T \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})} + h^2 \, \text{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) + O(h^3) \right] \\
&= \text{trace} \left(\mathbf{w}_{(\mathbf{X})} \mathbf{w}_{(\mathbf{X})}^T \mathbf{H}_{i(\mathbf{X})} \right) + \nabla I_i^T \mathbf{J}_{\mathbf{w}_{(\mathbf{X})}} \mathbf{w}_{(\mathbf{X})}
\end{aligned}
\tag{8}
$$

Note that this is exactly the right term in our curvature-preserving PDE (6).

Actually, (6) can be seen individually for all integral curves of $\mathcal{F}$ instead of each point $\mathbf{X} \in \Omega$: consider another point $\mathbf{Y} \in \mathcal{C}^{\mathbf{X}}$. Then, there exist $\epsilon \in \mathbb{R}$ such that $\mathbf{Y} = \mathcal{C}^{\mathbf{X}}_{(\epsilon)}$. Indeed, $\mathcal{C}^{\mathbf{X}}$ and $\mathcal{C}^{\mathbf{Y}}$ describe the same curve (7) with different parameterizations : $\forall a \in \mathbb{R}, \ \mathcal{C}^{\mathbf{Y}}_{(a)} = \mathcal{C}^{\mathbf{X}}_{(\epsilon+a)}$. As (6) is verified on $\mathbf{Y}$, then $\frac{\partial I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial t}_{|a=\epsilon} = \frac{\partial^2 I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial a^2}_{|a=\epsilon}$. This is obviously true for $\epsilon \in \mathbb{R}$ since (6) is verified for all points $\mathbf{Y}$ lying on the integral curve $\mathcal{C}^{\mathbf{X}}$. Then, the PDE (6) may be also written as :

$$
\forall \mathcal{C} \in \mathcal{F}, \ \forall a \in \mathbb{R}, \qquad \frac{\partial I_i(\mathcal{C}_{(a)})}{\partial t} = \frac{\partial^2 I_i(\mathcal{C}_{(a)})}{\partial a^2}
\tag{9}
$$

We recognize in (9) a *one-dimensional heat flow constrained on $\mathcal{C}$*. This is actually very different from a heat-flow *oriented* by $\mathbf{w}$, as in the formulation $\frac{\partial I_i}{\partial t} = \frac{\partial^2 I_i}{\partial \mathbf{w}^2}$ since the curvatures of integral curves of $\mathbf{w}$ are now implicitly taken into account. In particular, our constrained equation has the interesting property to vanish when image intensities are perfectly constant on the integral curve $\mathcal{C}$, whatever the curvature of $\mathcal{C}$ is. In this context, defining a field $\mathbf{w}$ that is tangent everywhere to the image structures will allow the preservation of these structures, even if they are curved (such as corners). This is not the case with divergence or trace-based PDE's (3),(5) classically used in image regularization. This curvature-preserving property of (6) is illustrated on Fig.1d and Fig.2b.

Our constrained equation (6) is an *elliptic* PDE since the matrix $\mathbf{w}\mathbf{w}^T$ is positive definite. The existence and unicity of the solutions of (6) are not directly approached in this article. Anyway, in next section 3.2, we show that its solution can be approximated by the technique of line integral convolutions, which is a well-posed analytical approach.

3.2 Curvature-Preserving PDE's and Line Integral Convolutions

Line Integral Convolutions (LIC) have been first introduced in [11] as a technique to render a textured image $\mathbf{I}^{LIC}$ that represents a vector field $\mathbf{w} : \Omega \to \mathbb{R}^2$. The idea, originally expressed as a discrete form, consists in smoothing an image $\mathbf{I}^{noise}$ - containing only noise - by averaging its pixel values along the integral curves of $\mathbf{w}$. Actually, a continuous formulation of a LIC is then :

$$\forall \mathbf{X} \in \Omega, \qquad \mathbf{I}^{LIC}_{(\mathbf{X})} = \frac{1}{N} \int_{-\infty}^{+\infty} f(p) \, \mathbf{I}^{noise}(\mathcal{C}^{\mathbf{X}}_{(p)}) \, dp \tag{10}$$

where $f : \mathbb{R} \to \mathbb{R}$ is an even function (strictly decreasing to 0 on $\mathbb{R}^+$) and $\mathcal{C}^{\mathbf{X}}$ is defined as the *integral curve* (7) of $\mathbf{w}$ through $\mathbf{X}$. The normalization factor N allows the preservation of the average pixel value along $\mathcal{C}^{\mathbf{X}}$ and is equal to $N = \int_{-\infty}^{+\infty} f(p) \, dp$.

As noticed in section 3.1, our curvature-preserving PDE (6) can be seen as the one-dimensional heat flow (9) constrained on the integral curve $\mathcal{C}^{\mathbf{X}} \in \mathcal{F}$. Using the variable substitution $\mathbf{L}_{(a)} = \mathbf{I}(\mathcal{C}^{\mathbf{X}}_{(a)})$, (9) can be also written as $\frac{\partial \mathbf{L}}{\partial t}(a) = \mathbf{L}''_{(a)}$. The solution $\mathbf{L}^{[t]}$ at time t is known to be the convolution of $\mathbf{L}^{[t=0]}$ by a normalized gaussian kernel G_t (see [20, 25]) :

$$\mathbf{L}^{[t]}_{(a)} = \int_{-\infty}^{+\infty} \mathbf{L}^{[t=0]}_{(p)} \, G_{t(a-p)} \, dp \quad \text{with} \quad G_{t(p)} = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{p^2}{4t}\right) \tag{11}$$

Substituting $\mathbf{L}$ in (11) with $a = 0$, and remembering that $\mathcal{C}^{\mathbf{X}}_{(0)} = \mathbf{X}$ and $G_{t(-p)} = G_{t(p)}$:

$$\forall \mathbf{X} \in \Omega, \qquad \mathbf{I}^{[t]}_{(\mathbf{X})} = \int_{-\infty}^{+\infty} \mathbf{I}^{[t=0]}(\mathcal{C}^{\mathbf{X}}_{(p)}) \, G_{t(p)} \, dp \tag{12}$$

The equation (12) is a particular form of the continuous LIC-based formulation (10) with a gaussian weighting function $f = G_t$. Here, the normalization factor is $N = \int_{-\infty}^{+\infty} G_{t(p)} \, dp = 1$. Intuitively, the evolution of our curvature-preserving PDE (6) may be seen as the application of local convolutions by normalized one-dimensional gaussian kernels *along integral curves* $\mathcal{C}$ of $\mathbf{w}$. This kind of anisotropic image smoothing considers then a *curved* filtering, instead of just an oriented one.

Applying this setting on a multi-valued image $\mathbf{I}$, with $\mathbf{w}$ being the lowest eigenvector of the structure tensor field $\mathbf{G}$ (i.e. the contour direction) allows the anisotropic smoothing of $\mathbf{I}$ with edge preservation, even if these edges are curved. This is illustrated on Fig.3b, where few integral lines $\mathcal{C}^{\mathbf{X}}$ are computed, around a typical T-junction structure. Note how the streamlines rotate when arriving at the junction, with a sub-pixel precision. The streamlines have been computed with a 2nd-order Runge-Kutta scheme.

Note that (12) is an analytical solution of (6) when $\mathbf{w}$ *does not evolve over time*. This property is generally not verified when dealing with general nonlinear regularization PDE's, where the smoothing geometry is re-evaluated at each time step (this defines a temporal non-linearity). In order to get this kind of non-linearity, we will then to perform several successive iterations of our LIC scheme (12), where the vector field $\mathbf{w}$ is updated at each iteration. This is actually a good way of approximating (6). Classical explicit schemes usually consider the smoothing geometry $\mathbf{w}$ as constant between two successive PDE iterations $\mathbf{I}^{[t]}$ and $\mathbf{I}^{[t+dt]}$. Thus, our curvature-preserving equation (6) will be efficiently discretized by several iterations of our LIC formulation (12) (section 4).

Note also that PDE-based algorithms performing vector flow visualization with textures have been already proposed in [6, 34], mainly inspired by the popular LIC technique, but no theoretical links between PDE-based formulations and LIC's have been done. Moreover, the use of divergence-based equations proposed in these paper does not ensure the correctness of the smoothing directions, as pointed out in section 2.2.

3.3 Between Traces and Divergences

We illustrate here how our curvature-preserving PDE (6) may be regarded compared to trace and divergence expressions (3), (5), for the case of single direction smoothing $\mathbf{T} = \mathbf{w}\mathbf{w}^T$.

In this case, the divergence PDE (3) may be developed as :

$$
\begin{aligned}
\mathrm{div}\left(\mathbf{w}\mathbf{w}^T\,\nabla I_i\right) &= \mathrm{div}\left(\begin{array}{c} u^2\,\frac{\partial I_i}{\partial x} + uv\,\frac{\partial I_i}{\partial y} \\[2mm] uv\,\frac{\partial I_i}{\partial x} + v^2\,\frac{\partial I_i}{\partial y} \end{array}\right) \\[4mm]
&= \left(u^2\,\frac{\partial^2 I_i}{\partial x^2} + 2uv\,\frac{\partial^2 I_i}{\partial x \partial y} + v^2\,\frac{\partial^2 I_i}{\partial y^2}\right) + \nabla I_i^T\left(\begin{array}{c} 2u\,\frac{\partial u}{\partial x} + u\,\frac{\partial v}{\partial y} + v\,\frac{\partial u}{\partial y} \\[2mm] 2v\,\frac{\partial v}{\partial y} + u\,\frac{\partial v}{\partial x} + v\,\frac{\partial u}{\partial x} \end{array}\right) \\[4mm]
&= \mathrm{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T\left[\left(\begin{array}{c} u\,\frac{\partial u}{\partial x} + v\,\frac{\partial u}{\partial y} \\[2mm] u\,\frac{\partial v}{\partial x} + v\,\frac{\partial v}{\partial y} \end{array}\right) + \left(\begin{array}{c} u\,\frac{\partial u}{\partial x} + u\,\frac{\partial v}{\partial y} \\[2mm] v\,\frac{\partial u}{\partial x} + v\,\frac{\partial v}{\partial y} \end{array}\right)\right] \\[4mm]
&= \mathrm{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T\mathbf{J_w}\mathbf{w} + \mathrm{div}(\mathbf{w})\nabla I_i^T\mathbf{w}
\end{aligned}
$$

Thus, we recognize in these three different terms :

- The first term corresponds to the trace PDE (5), that smoothes locally $\mathbf{I}$ along $\mathbf{w}$.

- The two first terms correspond to our *curvature-constrained* regularization PDE (6), that smoothes locally $\mathbf{I}$ along $\mathbf{w}$ while taking the curvature of integral curves $\mathcal{C}$ of $\mathbf{w}$ into account.

- The three terms together correspond to the classical divergence PDE (3) that performs local diffusions of $\mathbf{I}$ along $\mathbf{w}$. This last term $\mathrm{div}(\mathbf{w})\nabla I_i^T\mathbf{w}$ is mainly responsible for the perturbations of the effective smoothing direction, as described in section 2.2. It is not desirable for image regularization purposes.

It is interesting to observe that our curvature-constrained PDE (6) is then "mathematically" positioned between the trace (5) and divergence formulations (3), and allows at the same time the full respect of the pre-defined smoothing directions $\mathbf{w}$, while preserving curved images structures.
Note that we can also write our curvature-preserving PDE (6) as a divergence-based PDE minus a constraint term :

$$
\mathrm{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T\mathbf{J_w}\mathbf{w} = \mathrm{div}\left(\mathbf{w}\mathbf{w}^T\,\nabla I_i\right) - \mathrm{div}(\mathbf{w})\nabla I_i^T\mathbf{w}
$$

Two particular cases of directions $\mathbf{w}$ are worth studying, in the case of scalar-valued images ($n = 1$) :

- When $\mathbf{w} = \frac{\nabla I^\perp}{\|\nabla I\|}$ (isophote direction), then $\quad \nabla I^T\mathbf{J_w}\mathbf{w} = -I_{\mathbf{w}\mathbf{w}}$, vanishing then the velocity of our curvature-preserving evolution equation (6), by counterbalancing the trace-based term (which is nothing more than the *mean curvature motion* in this case). No smoothing will be then performed. This is quite natural since pixel along the isophotes have constant values, so averaging those values will not change the image. Note by comparison that the velocity of the corresponding divergence-based expression $\mathrm{div}\left(\mathbf{w}\mathbf{w}^T\,\nabla I_i\right)$ also vanishes here.

- When $\mathbf{w} = \frac{\nabla I}{\|\nabla I\|}$ (gradient direction), then $\quad \nabla I^T\mathbf{J_w}\mathbf{w} = 0$, and the velocity of our curvature-preserving PDE (6) becomes simply $I_{\mathbf{w}\mathbf{w}}$, which really corresponds to a smoothing of the image along the gradient direction (the same as the unconstrained trace-based PDE (5)). Note by comparison that the velocity of the corresponding divergence-based expression is ΔI in this case, which corresponds to an isotropic smoothing of the image, instead of an anisotropic one.

These two particular cases allows to better understand the difference of regularization behaviors between the trace, divergence and curvature-preserving formulations.
Note also that in case where $\mathbf{w}$ is a divergence free field (i.e $\div(\mathbf{w}) = 0$), the divergence-based PDE (3) and our curvature-preserving formulation (6) are strictly equivalent.

3.4 Extension to multi-directional smoothing

We extend our single-direction smoothing PDE (6) so that it can deal with a tensor-valued geometry $\mathbf{T} : \Omega \to \mathrm{P}(2)$, instead of a vector-valued geometry $\mathbf{w}$. As pointed out in section 2.1, a diffusion tensor describes much more complex smoothing behaviors than single directions. In particular, it may represents both *anisotropic* or *isotropic* regularization behaviors. The extension of our curvature-preserving PDE (6) is not straightforward : the notions of curvature and integral curves of tensors-valued fields $\mathbf{T}$ are not as natural as with direction fields $\mathbf{w}$.

To tackle this problem, we propose to locally decompose a tensor-driven smoothing process into several vector-driven smoothing processes along different orientations. We first notice that

$$\int_{\alpha=0}^{\pi} a_\alpha a_\alpha^T \, d\alpha = \frac{\pi}{2} \, \mathbb{I}_d \quad \text{where} \quad a_\alpha = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}$$

Then, any 2×2 tensor $\mathbf{T}$ may be written as :

$$\mathbf{T} = \frac{2}{\pi} \, \sqrt{\mathbf{T}} \left(\int_{\alpha=0}^{\pi} a_\alpha a_\alpha^T \, d\alpha \right) \sqrt{\mathbf{T}}$$

where $\sqrt{\mathbf{T}} = \sqrt{f^+}\mathbf{u}\mathbf{u}^T + \sqrt{f^-}\mathbf{v}\mathbf{v}^T$ stands for the square root of $\mathbf{T} = f^+\mathbf{u}\mathbf{u}^T + f^-\mathbf{v}\mathbf{v}^T$. One can easily verify that $(\sqrt{\mathbf{T}})^2 = \mathbf{T}$ and $(\sqrt{\mathbf{T}})^T = \sqrt{\mathbf{T}}$. Thus, the tensor $\mathbf{T}$ may be decomposed as :

$$\begin{aligned} \mathbf{T} &= \frac{2}{\pi} \int_{\alpha=0}^{\pi} \sqrt{\mathbf{T}} a_\alpha a_\alpha^T \sqrt{\mathbf{T}}^T \, d\alpha \\ &= \frac{2}{\pi} \int_{\alpha=0}^{\pi} (\sqrt{\mathbf{T}} a_\alpha)(\sqrt{\mathbf{T}} a_\alpha)^T \, d\alpha \end{aligned} \tag{13}$$

We have split the tensor $\mathbf{T}$ into a sum of *atomic* tensors $(\sqrt{\mathbf{T}} a_\alpha)(\sqrt{\mathbf{T}} a_\alpha)^T$, each being purely anisotropic and directed only along the direction of the vector $\sqrt{\mathbf{T}} a_\alpha \in \mathbb{R}^2$. The equation (13) naturally suggests to decompose any tensor-driven regularization PDE into a sum of single direction smoothing processes, each of them respecting the overall geometry $\mathbf{T}$. For instance :

- If $\mathbf{T} = \mathbb{I}_d$ (identity matrix), the tensor is isotropic and : $\forall\alpha \in [0,\pi]$, $\sqrt{\mathbf{T}} a_\alpha = a_\alpha$. The resulting smoothing will be then performed in all directions a_α of the plane with the same strength.

- If $\mathbf{T} = \mathbf{u}\mathbf{u}^T$ (where $\mathbf{u} \in \mathrm{S}^1$), the tensor is purely anisotropic and : $\forall\alpha \in [0,\pi]$, $\sqrt{\mathbf{T}} a_\alpha = (\mathbf{u}^T a_\alpha)\mathbf{u}$. The resulting smoothing will be then performed only along the direction $\mathbf{u}$ of the tensor $\mathbf{T}$.

Then, using (13) and considering that each single direction smoothing must be done with a curvature-preserving approach (6), we propose the following constrained regularization PDE, acting on a multi-valued image $\mathbf{I} : \Omega \to \mathbb{R}^n$ and driven by a tensor-valued smoothing geometry $\mathbf{T}$:

$$\forall i = 1, \ldots, n, \quad \frac{\partial I_i}{\partial t} = \frac{2}{\pi} \int_{\alpha=0}^{\pi} \text{trace}\left((\sqrt{\mathbf{T}} a_\alpha)(\sqrt{\mathbf{T}} a_\alpha)^T \mathbf{H}_i \right) + \nabla I_i^T \, \mathbf{J}_{\sqrt{\mathbf{T}} a_\alpha} \sqrt{\mathbf{T}} a_\alpha \, d\alpha$$

which can be simplified as :

$$\forall i = 1, \ldots, n, \quad \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) + \frac{2}{\pi} \nabla I_i^T \int_{\alpha=0}^{\pi} \mathbf{J}_{\sqrt{\mathbf{T}} a_\alpha} \sqrt{\mathbf{T}} a_\alpha \, d\alpha \tag{14}$$

where $a_\alpha = (\cos\alpha \quad \sin\alpha)^T$, and $\mathbf{J}_{\sqrt{\mathbf{T}} a_\alpha}$ stands for the Jacobian of the vector field $\Omega \to \sqrt{\mathbf{T}} a_\alpha$. Note that this kind of smoothing decomposition along all orientations of the plane can be also found in [51]. As in the single direction smoothing case, (14) may be seen as a trace-based equation (5), where an extra term has been added in order to respect the curvature of all integral lines passing through the tensor-valued geometry $\mathbf{T}$.

4 Implementation considerations

In order to implement our regularization method (14), we benefit from the LIC-based intepretation of curvature-preserving PDE's presented in section 3.2. Indeed, we can explicitely discretize (14) by the following Euler scheme :

$$\mathbf{I}^{[t+dt]} = \mathbf{I}^{[t]} + \frac{2dt}{N} \left(\sum_{k=0}^{N-1} \mathcal{R}(\sqrt{\mathbf{T}}a_\alpha) \right)$$

where $\alpha = k\pi/N$ (in the interval $[0, \pi]$), dt is the usual temporal discretization step and $\mathcal{R}(\mathbf{w})$ represents a discretization of the mono-directional smoothing PDE velocity (6) that preserve curvatures along a vector field $\mathbf{w}$. If we write this expression as : $\mathbf{I}^{[t+dt]} = \frac{1}{N} \left(\sum_{k=0}^{N-1} \mathbf{I}^{[t]} + 2dt \, \mathcal{R}(\sqrt{\mathbf{T}}a_\alpha) \right)$, we may express it as the averaging of different gaussian-ponderated LIC's along vector fields $\sqrt{\mathbf{T}}a_\alpha$:

$$\mathbf{I}^{[t+dt]} = \frac{1}{N} \left(\sum_{k=0}^{N-1} \mathbf{I}^{[t]}_{LIC(\sqrt{\mathbf{T}}a_\alpha)} \right) \quad ,$$

where each gaussian variance has a standard deviation dt.

Basically, the difficulty here is the LIC computation, which needs the tracking of integral curves of a vector field. Here, we used a very simple method based on the classical Runge-Kutta [35] integration scheme. Faster LIC implementations have been proposed in [40] but do not deal with gaussian ponderation functions, as needed here.

This simple observation leads then to the following fast algorithm for the implementation of one iteration of our curvature-preserving PDE (14) :

1. Compute the smoothed structure tensor field $\mathbf{G}_\sigma$ from $\mathbf{I}^{[t]}$:

$$\mathbf{G}_\sigma = G_\sigma * \sum_{i=1}^{n} \left(\begin{array}{cc} \left(\frac{\partial I_i^{[t]}}{\partial x}\right)^2 & \left(\frac{\partial I_i^{[t]}}{\partial x}\right)\left(\frac{\partial I_i^{[t]}}{\partial y}\right) \\ \left(\frac{\partial I_i^{[t]}}{\partial x}\right)\left(\frac{\partial I_i^{[t]}}{\partial y}\right) & \left(\frac{\partial I_i^{[t]}}{\partial y}\right)^2 \end{array} \right)$$

   σ will depend on the noise scale. We used relatively low values (between 0 and 1.5) for our experiments in section 5.

2. Compute the eigenvalues λ^+, λ^- and eigenvectors θ^+, θ^- of $\mathbf{G}_\sigma$.

3. Compute the smoothing geometry tensor field $\mathbf{T}$ from $\mathbf{G}_\sigma$: $\mathbf{T} = \frac{1}{(1+\lambda^++\lambda^-)^{p_1}} \, \theta^-\theta^{-T} + \frac{1}{(1+\lambda^++\lambda^-)^{p_2}} \, \theta^+\theta^{+T}$

4. For all α in $[0, \pi]$ (discretized with a user-fixed step d_α) :

   - Compute the vector field $\mathbf{w} = \sqrt{\mathbf{T}} \, a_\alpha$.
   - Perform a Line Integral Convolution of $\mathbf{I}^{[t]}$ along $\mathcal{C}^{\mathbf{X}}$ in the forward and backward directions.

5. Average all LIC's computed in step 4.

The main parameters of our algorithm are p_1, p_2, σ, dt and the number of PDE iterations nb that are applied. The characteristics of this scheme, compared to the classical finite-difference one is :

- It allows the preservation of thin image structures from a numerical point of view : the smoothing is performed along integral curves of $\mathbf{w}$, with a sub-pixel accuracy. Precise Runger-Kutta interpolation is used to track the integral curves $\mathcal{C}$.

- It allows to choose very large time steps dt, since the scheme we proposed is unconditionally stable. Indeed, dt simply corresponds to a smoothing variance of the gaussian-ponderated convolution along $\mathcal{C} \in \mathcal{F}$.

- As a result, the regularization algorithm performs very fast. Very few iterations are necessary to get the result, even if each iteration is more time-consuming. For our applications, presented in section 5, we were even able to choose $nb = 1$ iteration with very large time steps dt. In fact, this leads to a rough approximation of (14), since we lost the temporal non-linearity property of the PDE. But for images with few noise, this gave suprisingly good results. Actually, the spatial non-linearity seems to play a more important role than the temporal non-linearity in our scheme.

The smoothing is done as an averaging of multiple LIC's in different directions α. The choice of the discretization step d_α is important in this context. Actually, in regions where the smoothing needs to be mostly anisotropic, only few values of α are necessary since in all cases, the smoothing will be done along the same single direction. But in homogeneous regions needing isotropic smoothing, a smaller d_α will give much better results. Practically speaking, we chose $d_\alpha = 45^o$ which is enough to get a good precision for isotropic smoothing.

On Fig.4, we illustrate the efficiency of our new scheme, compared to the classical finite-difference one. A synthetic noisy image is anisotropically smoothed with our PDE (14), with $p_1 = 0.01$ and $p_2 = 100$ (smoothing mostly along isophotes θ_-, with a strength of 1). The LIC-based scheme (Fig.4c) better preserves the structure along time t. This is due to the important role played by the sub-pixel accuracy property of the underlying LIC computation.

(a) Noisy color image.

(b) Regularization using a finite-difference scheme (stopped at $t = 100$).

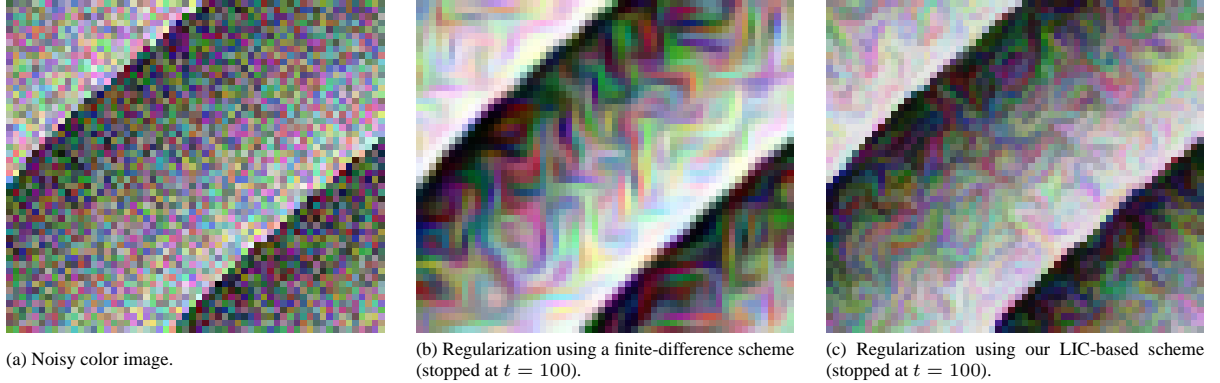(c) Regularization using our LIC-based scheme (stopped at $t = 100$).

Figure 4: Comparisons between classical explicit PDE schemes, and LIC-based implementation of our PDE (14).

5 Application Results

We present different application results of our curvature-preserving PDE (14), implemented by the LIC-based scheme and applied on 24bits color images $\mathbf{I} : \Omega \to [0, 255]^3$. The (R, G, B) color base has been considered for the PDE evolutions. All experiments have been performed on a PC 2.8 Ghz running Linux (single CPU). The implementation has been done in C++, thanks to *the CImg Library* [49], a very simple-to-use and powerful image processing library. For each result presented below, we detail the used parameters and the processing time.

5.1 Color Image Denoising and Regularization

Image denoising is a direct application of regularization methods. Sensor inaccuracies, digital quantifications or compression artefacts are indeed some of the various noise sources that can affect a digital image, and suppressing them is a desirable goal. In Fig.5, we illustrate how our curvature-preserving PDE (14) can be successfully applied to remove such artefacts while preserving the essential structures of the processed images.

- Fig.5a shows a restoration of the "baboon" color image, artificially degraded by adding uncorrelated gaussian noise on (R, G, B). This 512×512 color image has been regularized with (14) and a 140×111 portion of the image is shown. Only one PDE iteration has been necessary, with $p_1 = 0.5$, $p_2 = 0.7$, $\sigma = 1.5$ and $dt = 50$. Processing time is 19.3 seconds for the entire image.

- Fig.5b illustrates a real case where a color photograph has been digitized from a grainy paper, leading to the apparition of watered effects on the digital picture (size=586×367). Using our regularization method allows to clearly remove the grain while preserving quite fine structures (palm tree leafs). Shown image is a 152×133 portion of the original one. Only one PDE iteration has been necessary, with $p_1 = 0.5$, $p_2 = 0.7$, $\sigma = 1$ and $dt = 10$. Processing time is 11 seconds for the entire image.

- Fig.5c deals with the suppression of compression artefacts in color images. A JPEG version of the "Lena" color image (size=256×256, where the JPEG quality ratio has been set to 10%) is processed by our regularization algorithm. Usual

block effects inherent to the DCT compression are visible on the compressed image (left). One PDE iteration is applied then, with $p_1 = 0.5$, $p_2 = 0.9$, $\sigma = 2$, $dt = 200$, in order to get the regularized result (right). Only a 100×73 portion of the original image is shown. Processing time is 6.4 seconds for the entire image.

- Fig.5d illustrates how our regularization method is used to improve a digital color image quantified in 256 colors by the Floyd-Steinberg algorithm (size=355×287). One PDE iteration has been applied, with $p_1 = 0.5$, $p_2 = 0.8$, $\sigma = 1$, $dt = 30$. A 136×118 portion of the image is shown. Processing time is 12.8 seconds for the entire image.

- Fig.5e shows a digital photograph shot under low luminosity conditions, leading to the apparition of real digital noise (poisson noise). Processed color image has size=293×306 and has been restored in 5.6 seconds (one PDE iteration), with parameters $p_1 = 0.2$, $p_2 = 0.5$, $\sigma = 2$, $dt = 120$.

- Fig.5f illustrates how exaggerating the smoothing geometry can create interesting painting effects. One PDE iteration of (14) has been applied, with $p_1 = 0.5$, $p_2 = 1.2$, $\sigma = 4$ (which leads to an exaggeratedly smooth geometry $\mathbf{T}$) and $dt = 20$. Processing time is 26 seconds for this 460×365 color image.

Note that our equation (14) is acting a as intelligent image smoother. It is actually not able to perform edge enhancement, as divergence-based PDE's (3) may do. This would be possible anyway by adding a classical shock-filter term (such as proposed in [2, 31] for scalar images and extended in [44, 55] for multi-valued ones) to our curvature-preserving PDE formulation (14). Generally, this enhancement is not necessary for noisy images, particularly since we preserve the edges very well with our curvature-preserving method.

5.2   Color Image Inpainting

Image inpainting is a very new and challenging application, which consists in filling-in missing (user-defined) image regions by guessing pixel values such that the reconstructed image still looks natural. Basically, the user provides one color image $\mathbf{I} : \Omega \to \mathbb{R}^3$, and one *mask* image $M : \Omega \to \{0, 1\}$. The inpainting algorithm must fill-in the regions where $M(\mathbf{X}) = 1$, by the mean of some intelligent interpolations. Inpainting algorithms can be used for instance to remove various structures in images (scratches, logos or real objects). Pioneering work on image inpainting has been first proposed as a variational formulation by Masnou and Morel [29], followed by many PDE-based solutions [8, 9, 15, 48]. It is also worth to cite some papers related to inpainting without use of PDE's [19, 22], among others.
In this article, we see the inpainting process as a direct application of our proposed curvature-preserving PDE (14). Applying the diffusion equation only on the regions to inpaint allows the neighbor pixels to diffuse inside these regions : a nonlinear completion of the image data along isophotes directions is thus naturally done, reconstructing the missing parts of the image. This kind of PDE-based inpainting technique has been also proposed in [8, 15, 48]. Note that it is not able to perform texture reconstructions, and texture synthesis steps will be sometimes necessary [3, 9, 56].

- Fig.6a shows how our PDE-based inpainting technique can be used to remove real objects from digital photographs. A 500×500 color image (left) is inpainted with a user-defined mask (middle). The inpainted image (right) is obtained in 4 minutes 11 seconds, after 200 iterations of our PDE (14) with parameters $p_1 = 0.001$, $p_2 = 100$, $\sigma = 4$, $dt = 150$. Note that $p_1 << p_2$ encourages smoothing only along the isophote directions with a strength of 1 everywhere.

- Fig.6b shows an application of subtitles removing in a movie frame. Image size is 300×162 and the inpainted image has been obtained after 20 iterations of (14), with $p_1 = 0.001$, $p_2 = 100$, $\sigma = 4$, $dt = 50$, for a total processing time of 11 seconds.

- Fig.6c illustrates the reconstruction capabilities of our inpainting technique. Half of the pixels of a 300×246 color image have been suppressed by masking them with a 16×16 checkerboard-shaped mask. Then, the image is reconstructed using 10 iterations of our PDE (14) inside the inpainting mask, with parameters $p_1 = 0.001$, $p_2 = 100$, $\sigma = 4$ and $dt = 50$ (processing time is 1 minute 01 seconds).

For each inpainting result shown in this article, the initialization of the pixel values inside the inpainting masks at $t = 0$ has been done by white noise. Actually, the inpainting algorithm is not much dependent of the initialization step : the equation (14) diffuses neighborhood values inside the inpainting mask until convergence, and there is then a strong border condition. We didn't see much difference between different types of initialization (noise, zero-filling or linear interpolation).
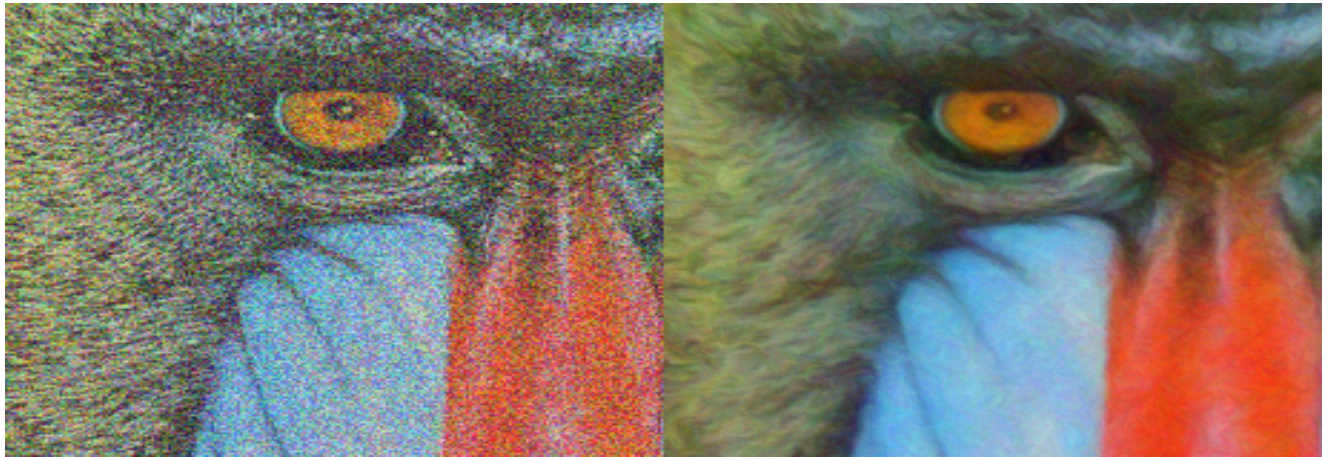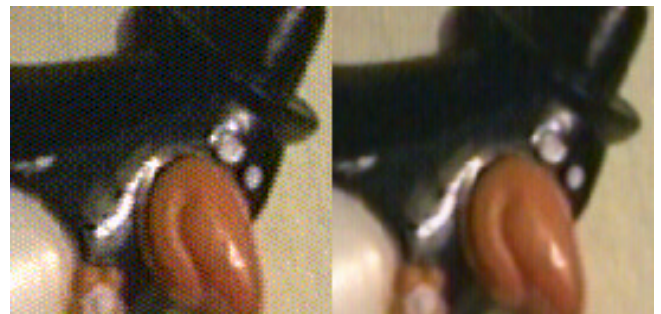
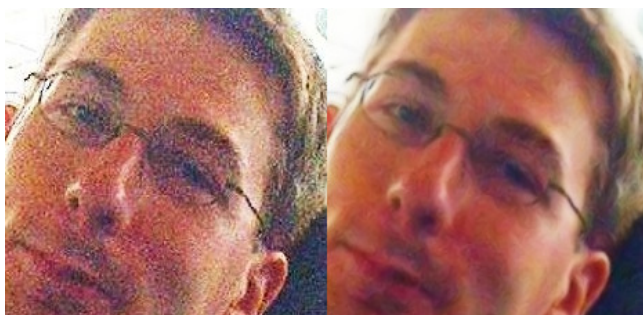(a) Denoising of the "baboon" color image (19.3s).

(b) Watered effect suppression in a color image (11s).
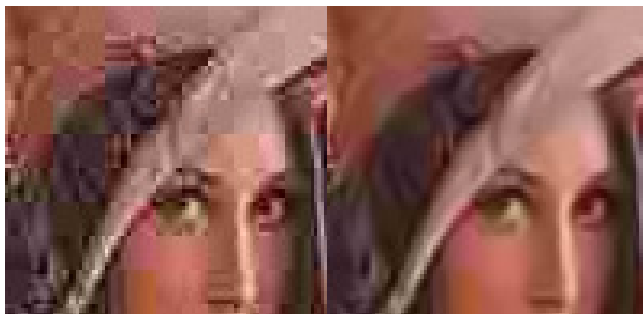
(c) Suppression of JPEG compression artefacts in the "lena" image (6.4s).

(d) Suppression of quantification artefacts in a 8bits color image (12.8s).
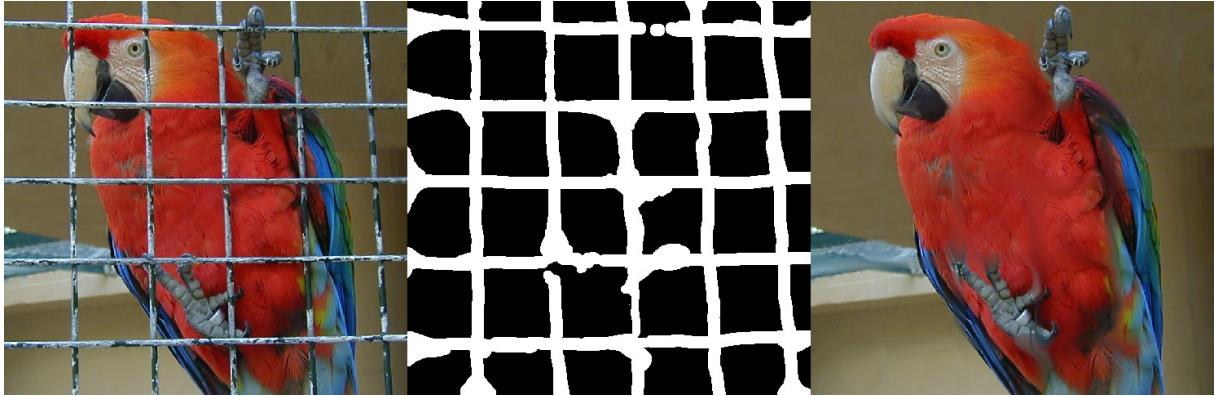

(e) Denoising of a digital photograph with digital noise (5.6s).

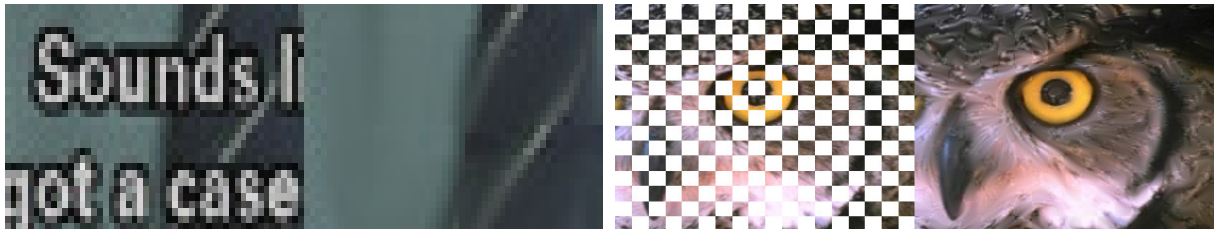(f) Creating painting effects with over-smoothing procedures (26s).

Figure 5: Results of color image regularization using our curvature-preserving PDE's (14).

(a) Inpainting a cage (middle) in a color image (left). Inpainted in 4m11s (right).

(b) Removing subtitles from a movie frame (11s).

(c) *Left :* Zoom of (b).   *Right :*   Reconstruction of a color image where 50% of the pixel values have been suppressed (1m01s).

Figure 6: Results of color image inpainting using our curvature-preserving PDE (14).

(a) Original color image

Figure 7: Comparisons of image resizing, using Nearest-neighbor (first row), Linear (second row), Bicubic (third row) and PDE-based (last row) interpolations.

5.3 Color Image Resizing

The inpainting technique naturally suggests that the nonlinear interpolation implicitly performed by our regularization PDE (14) can be also applied to magnify images, just as a replacement to classical linear or bicubic interpolations. This is done as follows : starting from a linear or bicubic magnification of a color image as an initialization, we apply our curvature-preserving PDE (14) everywhere except on the pixels having "known" intensity values (pixels created from the original image points). This is actually very similar to image inpainting with a very sparse grid for the mask.

- Fig.7 illustrates one example of image resizing. An original 220×210 color image is resized by a factor $\times 2$ with classical nearest-neighbor, linear and bicubic interpolations, then by our PDE-based technique (14). Our non-linear regularization filter allows to remove the aliasing effects usually encountered with simple interpolation methods, while correctly preserving the edges of the image.

Notice that we always preserve the values of the points corresponding to the original thumbnail image, so that resizing back the image to its original dimension (sub-sampling) results in the original input data.

5.4 Other Results & Availability

Many application results of our algorithm can be found at the following web page :

<div align="center">

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

</div>

You can also download and test the algorithm on different architectures. Finally, the source of the algorithm (in C++) is available as a part of the open source *CImg Library* [49].

Conclusion

We proposed a generic constrained regularization formalism able to anisotropically smooth multi-valued images with PDE's while preserving natural curvature constraints. It can be used in a wide range of image processing applications, including image denoising, inpainting, interpolation among others. This formalism makes the link between general diffusion PDE's and Line Integral Convolutions, and leads to the design of a very fast and efficient numerical scheme that implements the method. This may ease the introduction of nonlinear diffusion methods for time-critical applications in the area of multimedia, medical imaging and image processing in general.

References

[1] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. *Axioms and fundamental equations of image processing.* Archive for Rational Mechanics and Analysis, Vol.123, No.3, pp.199–257, 1993.

[2] L. Alvarez and L. Mazorra. *Signal and Image Restoration using Shock Filters and Anisotropic Diffusion.* SIAM Journal of Numerical Analysis, Vol.31, No.2, pp.590–605, 1994.

[3] M. Ashikhmin. *Synthesizing Natural Textures*. ACM Symposium on Interactive 3D Graphics, Research Triangle Park, NorthCarolina, pp.217–226, March 2001.

[4] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Applied Mathematical Sciences, Vol.147, Springer-Verlag, January 2002.

[5] D. Barash. *A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing and the Nonlinear Diffusion Equation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.6, p.844.

[6] J. Becker, T. Preusser, and M. Rumpf. *PDE methods in flow simulation post processing.* Computing and Visualization in Science, Vol.3, No.3, pp.159–167, 2000.

[7] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro. *Variational Problems and Partial Differential Equations on Implicit Surfaces.* Computing and Visualization in Science, Vol.174, No.2, pp.759–780, 2001.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. *Image inpainting.* ACM SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques, pp.417–424, 2000.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. *Simultaneous Structure and Texture Image Inpainting*. IEEE Transactions on Image Processing, Vol.12, No.8, pp.882–889, August 2003.

[10] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. *Robust anisotropic diffusion.* IEEE Transaction on Image Processing, Vol.7, No.3, pp.421–432, 1998.

[11] B. Cabral and L.C. Leedom. *Imaging vector fields using line integral convolution.* SIGGRAPH'93, in Computer Graphics Vol.27, pp.263–272, 1993.

[12] R. Carmona and S. Zhong. *Adaptive Smoothing Respecting Feature Directions.* IEEE Transactions on Image Processing, Vol.7, No.3, pp.353–358, 1998.

[13] A. Chambolle and P.L. Lions. *Image recovery via total variation minimization and related problems.* Nümerische Mathematik, Vol.76, No.2, pp.167–188, 1997.

[14] T. Chan and J. Shen. *Variational restoration of non-flat image features : Models and algorithms.* SIAM Journal of Applied Mathematics, Vol.61, No.4, pp.1338–1361, 2000.

[15] T. Chan and J. Shen. *Non-texture inpaintings by curvature-driven diffusions.* Journal of Visual Communication and Image Representation, Vol.12, No.4, pp.436–449, 2001.

[16] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. *Deterministic edge-preserving regularization in computed imaging.* IEEE Transactions on Image Processing, Vol.6, No.2, pp.298–311, 1997.

[17] C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. *Regularizing Flows for Constrained Matrix-Valued Images.* Journal of Mathematical Imaging and Vision, Vol.20, No.2, pp.147-162, January 2004.

[18] O. Coulon, D.C. Alexander, and S.R. Arridge. *A regularization scheme for diffusion tensor magnetic resonance images.* 17th International Conference on Information Processing in Medical Imaging, LNCS, Vol.2082, pp.92–105, 2001.

[19] A. Criminisi, P. Perez, and K. Toyama. *Object Removal by Exemplar-based Inpainting* IEEE Conference on Computer Vision and Pattern Recognition, Vol.2, pp.721–728, June 2003.

[20] R. Deriche and O. Faugeras. *Les EDP en traitement des images et vision par ordinateur.* Traitement du Signal, Vol.13, No.6, 1997.

[21] S. Di Zenzo *A note on the gradient of a multi-image.* Computer Vision, Graphics and Image Processing, Vol.33, pp.116-125, 1986.

[22] J. Jia and C.K. Tang *Image Repairing : Robust Image Synthesis by Adaptive ND Tensor Voting* IEEE Conference on Computer Vision and Pattern Recognition, Vol.1, pp.643–650, June 2003.

[23] R. Kimmel, R. Malladi, and N. Sochen. *Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images.* International Journal of Computer Vision, Vol.39, No.2, pp.111–129, September 2000.

[24] R. Kimmel and N. Sochen. *Orientation diffusion or how to comb a porcupine.* Journal of Visual Communication and Image Representation, Vol.13, pp.238–248, 2002.

[25] J.J. Koenderink. *The structure of images.* Biological Cybernetics, Vol.50, pp.363–370, 1984.

[26] P. Kornprobst, R. Deriche, and G. Aubert. *Nonlinear operators in image restoration* IEEE Conference on Computer Vision and Pattern Recognition, pp 325-331, June 1997.

[27] K. Krissian. *Multiscale Analysis : Application to Medical Imaging and 3D Vessel Detection.* Ph.D. Thesis, INRIA-Sophia Antipolis/France, 2000.

[28] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[29] S. Masnou and J-M. Morel. *Level Lines Based Disocclusion.* IEEE International Conference on Image Processing, Vol.3, pp.259-263, October 1998.

[30] M. Nielsen, L. Florack, and R. Deriche. *Regularization, scale-space and edge detection filters.* Journal of Mathematical Imaging and Vision, Vol.7, No.4, pp.291–308, 1997.

[31] S. Osher and L.I. Rudin. *Feature-oriented image enhancement using shock filters.* SIAM Journal of Numerical Analysis, Vol.27, No.4, pp.919–940, August 1990.

[32] P. Perona. *Orientation diffusions.* IEEE Transactions on Image Processing, Vol.7, No.3, pp.457–467, March 1998.

[33] P. Perona and J. Malik. *Scale-space and edge detection using anisotropic diffusion.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, No.7, pp.629–639, July 1990.

[34] T. Preusser and M. Rumpf. *Anisotropic nonlinear diffusion in flow visualization.* IEEE Visualization Conference, 1999.

[35] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *"Runge-Kutta Method"* in Numerical Recipes in FORTRAN: The Art of Scientific Computing, Cambridge University Press, pp. 704-716, 1992.

[36] L. Rudin, S. Osher, and E. Fatemi. *Nonlinear total variation based noise removal algorithms.* Physica D, Vol.60, pp.259–268, 1992.

[37] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis.* Cambridge University Press, 2001.

[38] G. Sapiro and D.L. Ringach. *Anisotropic diffusion of multi-valued images with applications to color filtering.* IEEE Transactions on Image Processing, Vol.5, No.11, pp.1582–1585, 1996.

[39] N. Sochen, R. Kimmel, and A.M. Bruckstein. *Diffusions and confusions in signal and image processing.* Journal of Mathematical Imaging and Vision, Vol.14, No.3, pp.195–209, 2001.

[40] D. Stalling and H.C. Hege. *Fast and Resolution Independent Line Integral Convolution.* ACM SIGGRAPH, 22nd Annual Conference on Computer Graphics and Interactive Technique, pp.249–256, 1995.

[41] B. Tang, G. Sapiro, and V. Caselles. *Direction diffusion.* IEEE International Conference on Computer Vision, pp.1245, 1999.

[42] B. Tang, G. Sapiro, and V. Caselles. *Diffusion of general data on non-flat manifolds via harmonic maps theory : The direction diffusion case.* International Journal of Computer Vision, Vol.36, No.2, pp.149–161, February 2000.

[43] C. Tomasi and R. Manduchi. *Bilateral filtering for gray and color images.* IEEE International Conference on Computer Vision, pp.839–846, January 1998.

[44] D. Tschumperlé *PDE's Based Regularization of Multi-valued Images and Applications.* PhD Thesis, Université de Nice-Sophia Antipolis/France, December 2002.

[45] D. Tschumperlé and R. Deriche. *Orthonormal Vector Sets Regularization with PDE's and Applications.* International Journal of Computer Vision, Vol.50, pp. 237–252, 2002.

[46] D. Tschumperlé and R. Deriche. *Diffusion tensor regularization with constraints preservation.* IEEE Conference on Computer Vision and Pattern Recognition, Vol.1, pp.948-953, December 2001.

[47] D. Tschumperlé and R. Deriche *Diffusion PDE's on Vector-Valued Images : Local Approach and Geometric Viewpoint.* IEEE Signal Processing Magazine, Vol.19, No.5, pp.16–25, September 2002.

[48] D. Tschumperlé and R. Deriche *Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.4, April 2005.

[49] D. Tschumperlé. *The CImg Library :* `http://cimg.sourceforge.net`. The C++ Template Image Processing Library.

[50] B. Vemuri, Y. Chen, M. Rao, T. McGraw, T. Mareci, and Z. Wang. *Fiber tract mapping from diffusion tensor MRI.* IEEE Workshop on Variational and Level Set Methods in Computer Vision, July 2001.

[51] J. Weickert. *Anisotropic Diffusion Filters for Image Processing Based Quality Control.* 7th European Conference on Mathematics in Industry, pp.355–362, 1994.

[52] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner-Verlag, Stuttgart, 1998.

[53] J. Weickert. *Coherence-Enhancing Diffusion of Colour Images.* Image and Vision Computing, Vol.17, pp.199–210, 1999.

[54] J. Weickert and T. Brox. *Diffusion and Regularization of Vector and Matrix-valued Images.* Inverse Problems, Image Analysis, and Medical Imaging, Vol.313 of Contemporary Mathematics, pp.251–268, 2002.

[55] J. Weickert *Coherence-Enhancing Shock Filters.* Pattern Recognition, 25th DAGM Symposium, LNCS, Vol.2781, pp.1–8, 2003.

[56] L.Y Wei and M. Levoy *Fast Texture Synthesis using Tree-structured Vector Quantization.* ACM SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques, pp.479–488, 2000.