Anisotropic Diffusion Partial Differential Equations in Multi-Channel Image Processing: Framework and Applications

David Tschumperlé, Rachid Deriche

Anisotropic Diffusion PDE's for Multi-Channel Image Regularization : Framework and Applications

David Tschumperlé* Rachid Deriche*

* *Image Team, GREYC / ENSICAEN - UMR CNRS 6072, 6 Bd du Maréchal Juin, 14050 Caen Cedex, France.*

* *Odyssée Project Team, INRIA/ENPC/ENS - INRIA, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis, France.*

Abstract

We review recent methods based on diffusion PDE's (Partial Differential Equations) for the purpose of multi-channel image regularization. Such methods have the ability to smooth multi-channel images anisotropically and can preserve then image contours while removing noise or other undesired local artifacts. We point out the pros and cons of the existing equations, providing at each time a local geometric interpretation of the corresponding processes. We focus then on an alternate and generic tensor-driven formulation, able to regularize images while specifically taking the curvatures of local image structures into account. This particular diffusion PDE variant is actually well suited for the preservation of thin structures and gives regularization results where important image features can be particularly well preserved compared to its competitors. A direct link between this curvature-preserving equation and a continuous formulation of the Line Integral Convolution technique (Cabral and Leedom, 1993) is demonstrated. It allows the design of a very fast and stable numerical scheme which implements the multi-valued regularization method by successive integrations of the pixel values along curved integral lines. Besides, the proposed implementation, based on a fourth-order Runge Kutta numerical integration, can be applied with a subpixel accuracy and preserves then thin image structures much better than classical finite-differences discretizations, usually chosen to implement PDE-based diffusions. We finally illustrate the efficiency of this diffusion PDE's for multi-channel image regularization - in terms of speed and visual quality - with various applications and results on color images, including image denoising, inpainting and edge-preserving interpolation.

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

Preliminary Notations

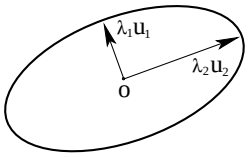
- Throughout this chapter, we will represent a *multi-channel* or *multi-valued image* by a continuous function $\mathbf{I} : \Omega \to \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^2$ is the definition domain of the image (basically a $2D$ rectangle $W \times H$) and $n \in \mathbb{N}^+$ is the dimension of each vector-valued image pixel $\mathbf{I}(\mathbf{X})$ located at $\mathbf{X} = (x \;\; y)^T \in \Omega$. The notation I_i stands for the i^{th} *channel* of the image $\mathbf{I}$. Note that I_i can be considered itself as a scalar-valued image $I_i : \Omega \to \mathbb{R}$. Thus, we have

$$\forall \mathbf{X} = (x, y) \in \Omega, \qquad \mathbf{I}_{(\mathbf{X})} = \left(I_{1(\mathbf{X})} \;\; I_{2(\mathbf{X})} \;\; \ldots \;\; I_{n(\mathbf{X})}\right)^T$$

For the common case of color images, we naturally get $n = 3$, i.e. three vector components (R,G,B) per pixel, retrieved respectively from the red (I_1), green (I_2) and blue (I_3) channels of a color image $\mathbf{I}$.

- We will also intensely use 2^{nd}-order *diffusion tensors* in equations described in this chapter. A diffusion tensor $\mathbf{D}$ is assimilated to a 2×2 *symmetric* and *positive-definite* matrix, having then two positive eigenvalues λ_1, λ_2 and two associated orthonormal eigenvectors $\mathbf{u}_1 \perp \mathbf{u}_2$. The *shape* of a tensor $\mathbf{D}$ may be seen as an *ellipse*, oriented by the vector basis $\mathbf{u}_1 \perp \mathbf{u}_2$ and elongated by λ_1 and λ_2, as illustrated below.

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \lambda_1 \, \mathbf{u}_1\mathbf{u}_1^T + \lambda_2 \, \mathbf{u}_2\mathbf{u}_2^T$$



When $\lambda_2 >> \lambda_1$ (lenghtened ellipse), the tensor $\mathbf{D}$ is said to be *anisotropic* and has $\mathbf{u}_2$ has its principal orientation. When $\lambda_1 = \lambda_2 = \beta$, the tensor $\mathbf{D}$ is *isotropic* and thus equal to a weighted version of the 2x2 identity matrix $\mathbb{I}_{\mathrm{d}}$

$$\lambda_1 = \lambda_2 = \beta \quad \Longrightarrow \quad \mathbf{D} = \beta \, \mathbb{I}_{\mathrm{d}} = \begin{pmatrix} \beta & 0 \\ 0 & \beta \end{pmatrix}$$

An *isotropic* tensor $\mathbf{D}$ have no privileged orientations, all vectors of $\mathbb{R}^2$ being possible eigenvectors of $\mathbf{D}$.

- Finally, we will denote G_σ, a normalized $2D$ Gaussian function with a standard deviation of σ :

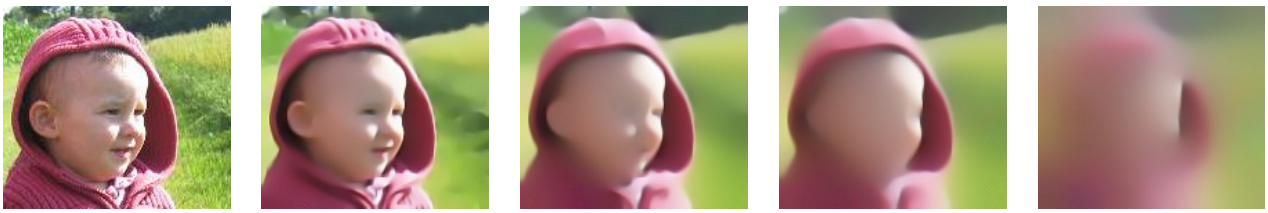
$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

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. Removing noise or scratches from degraded images is indeed a fundamental pre-processing step that can possibly ease the further analysis of the image data by higher-level algorithms such as detectors of important image features (edges, corners, objects, motion,...). The ability to create simplified versions of the image data is very interesting as well, when considering the analysis of the images at multiple scales. In a more general manner, image regularization is one of the key stage of most computer vision algorithms since it plays a fundamental role for solving *ill-posed* computer vision problems [53], including restoration, segmentation, registration, surface reconstruction, etc. This explains why a lot of image regularization formalisms have been already proposed and studied in the literature.
Perona & Malik in their pioneering work [80] in the early 90's were the first to imagine image regularization in terms of anisotropic diffusion PDE's (Partial Differential Equations). Their method, applied on scalar-valued images (one value by pixel), has particularly raised a strong interest for PDE-based formulations, since it succeeded in smoothing image data in a nonlinear way, removing the noise quite well while allowing the preservation of significant image features, such as contours and corners (discontinuities of the signal), despite an initial formulation that has been proved later to be unstable [122]. Firstly created to describe physical laws and natural motions of mechanic objects and fluids (strings, water, wind [126]), diffusion PDE's had been already widely studied and interesting theoretical results coming from the fields of physics and mathematics have found interesting implications for the purpose of data regularization. Actually, 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. This is not restrictive : Gaussian noise, scratches or compression artifacts are, for instance, local degradations usually encountered in digital (original or digitized) images.
Following the way opened by Perona & Malik, many authors have proposed variants of diffusion PDE's for image regularization since then, mostly for the restoration of scalar-valued datasets. Important theoretical contributions in this field concern the way the classical isotropic diffusion equation (heat flow) has been extended to deal with anisotropic smoothing [67, 80, 88, 118], how diffusion PDE's may be seen as gradient descents of various energy functionals [9, 25, 32, 59, 85], and the link between regularization PDE's and the concept of nonlinear scale spaces [4, 68, 72]. Extensions of these techniques to deal with color images and more generally multi-channel datasets have been more recently tackled in [21, 59, 78, 88, 89, 101, 109, 110, 118, 119] (among others), leading to more elaborated expressions : a coupling term between image channels generally appears in the equations. Diffusion equations dealing with constrained multi-dimensional datasets have been also proposed, allowing to regularize images of unit vectors [37, 61, 79, 96], orthonormal matrices [33, 105], positive-definite matrices [33, 104], or image data defined on implicit surfaces [16, 29, 97]. Usually, this kind of constrained PDE's simply get an extra constraint term added to the corresponding unconstrained equation and will not be discussed here.

Despite this wide range of existing constrained and unconstrained PDE formalisms for scalar and multi-channel images, all proposed methods have something in common : a nonlinear regularization PDE such as $\frac{\partial I}{\partial t} = \mathcal{R}$ locally *smoothes* the

image I along one or several directions of the plane that are different at each image point, depending on the local image configuration. Typically, the principal smoothing direction is always chosen to be parallel to the image contours, resulting in an *anisotropic* regularization that does not destroy the edges. This has an interesting interpretation in terms of *scale-space* : as the image data are gently regularized step-by-step, a continuous sequence of smoother images $I_{(t)}$ is generated whereas the evolution time t of the PDE goes by. Obviously, anisotropic regularization algorithms must let the less significant data features disappear first (preferably noise), while the interesting image details (edges) are preserved as long as they become unimportant themselves within the image [4, 68, 72, 80, 127]. Roughly speaking, regularization PDE's may be seen as iterative and nonlinear filters that simplify the image little by little and minimize then the image variations (Fig.1).



(a) Initial image $I_{(t=0)}$ (b) $t = 50$ (c) $t = 250$ (d) $t = 1000$ (e) $t = 3000$

Figure 1: Nonlinear regularization PDE's and the notion of anisotropic scale-space.

Note therefore that such equations generally do not converge towards a very interesting solution. Basically, the image obtained at convergence ($t \to \infty$) is constant everywhere, corresponding to an image without any variations. This is indeed the most simplified image we can obtain. To avoid this undesired over-simplification, regularization algorithms are usually based on a modified PDE velocity $\mathcal{R}' = \mathcal{R} + \alpha\,(I_{\text{noisy}} - I)$ including a so-called *data fidelity term* weighted by a user-defined parameter $\alpha \in \mathbb{R}^+$. It avoids the expected solution (regularized image) at convergence to be too different from the original noisy image (not constant, by the way). Another classical restoration technique is done by stopping the pure regularization flow $\frac{\partial I}{\partial t} = \mathcal{R}$ after a finite number of iterations (which becomes thus a parameter of the method). Here, we are mainly interested in the regularization term $\mathcal{R}$ itself rather than the one containing the fidelity term $\mathcal{R}'$. For a vast mathematical study about linear or nonlinear fidelity terms, please refer to [71, 74, 75].
As it is clear that local and oriented image smoothing is one of the key idea used by most PDE-based regularization methods, it naturally leads to the problem of defining a coherent geometry from a multi-channel image. It must be the first aim of a good regularization algorithm. Following this simple and general principle, recent contributions [109, 110, 118] proposed two different and generic PDE-based frameworks able to design regularization processes from any given 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). Then, a local geometry of the desired smoothing is defined by the mean of a second field of *diffusion tensors*, depending on the first one. Finally, one step of the smoothing process itself is performed through one or several iterations of a specific diffusion PDE. This procedure is repeated until the image is regularized enough.

In this chapter, we will first discuss the definition of a local geometry for multi-channel images, by reviewing and comparing proposed solutions in the literature [20, 21, 41, 86, 118] (section 1). Then, we will review important works already proposed for scalar and multi-channel image regularization within a diffusion PDE framework. These methods may be classified into three different approaches which are (1) variational formulations, (2) divergence expressions and (3) oriented Laplacians. We will mainly try to focus on the interpretation of the algorithms in terms of local smoothing (section 2). We particularly point out the advantages and drawbacks of each equation in real cases. Then, we focus on a very recent alternative, formulated as a tensor-driven diffusion that regularizes multi-channel images while taking specific curvature constraints into account (section 3). This formulation is mathematically positioned between previous existing equations, in a way that it solves most issues encountered with classical regularization methods. Moreover, we show that a theoretical interpretation of the curvature-constrained formalism exists in terms of Line Integral Convolutions which is a simple filtering technique originally proposed by Cabral and Leedom in [23]. This direct analogy allows the design of an explicit numerical scheme that implements the regularization PDE by successive integrations of pixel values along integral lines (section 4). This iterative scheme has two

main advantages compared to classical PDE implementations : on one hand, it preserves thin image structures remarkably well, since it naturally works at a sub-pixel accuracy, thanks to the use of a fourth-order Runge Kutta integration. On the other hand, the algorithm is able to run up to three times faster than classical explicit schemes since it is unconditionally stable, even for large PDE time steps. The described method makes diffusion PDE's a generic and very efficient approach for solving image processing problems needing multi-channel image regularization.
We finally illustrate this effectiveness, 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).

1 Defining a Local Geometry for Multi-Channel Images

1.1 Local geometric features

As stated in the introduction, image regularization may be seen as a filter that reduces local pixel variations. More precisely, one wants to smooth a multi-channel image $\mathbf{I} : \Omega \to \mathbb{R}^n$ while preserving its edges (discontinuities in the image intensities), i.e. performs a local smoothing mostly along directions of the edges, avoiding a smoothing orthogonal to these edges. At a first glance, a naive idea would be to apply a scalar-valued regularization filter on each channel I_i of the multi-channel image $\mathbf{I}$, doing this independently for each $i = 1...n$. But, the correlation between image channels would be ignored in this case, and it might cause important disparities in the smoothing behavior, since local smoothing directions and amplitudes could be very different from each channel to another. Such decoupled regularization methods generally lead to undesirable over-smoothing effects destroying significant edge structures in the image.
Multi-channel image regularization is rather based on a coherent image smoothing which locally uses the same smoothing directions and amplitudes for all image channels I_i. Naturally, this means that one has first to measure the *local geometry* of a multi-channel image $\mathbf{I}$. Such a geometry consists actually in the definition of these important features at each image point $\mathbf{X} = (x, y) \in \Omega$ of $\mathbf{I}$:

- Two orthogonal directions $\theta^+_{(\mathbf{X})}$, $\theta^-_{(\mathbf{X})} \in \mathrm{S}^1$ (unit vectors of $\mathbb{R}^2$) directed respectively across and along the edges (generally the maximum and minimum variations of the image intensities at $\mathbf{X}$). The direction θ^- generally corresponds to the edge direction, when there is one, while θ^+ naturally extends the notion of *gradient direction* for multi-channel images.

- A corresponding variation norm $\mathcal{N}(\mathbf{X})$ measuring the *local strength* of an edge. This is the extension of the *vector gradient norm* for multi-channel images.

In order to construct such a vector geometry, different approaches have been considered so far and are detailed below.

1.2 Geometry from a scalar feature

One simple method consists in computing first a scalar image $f(\mathbf{I})$, using a vector to scalar function $f : \mathbb{R}^n \to \mathbb{R}$ that would ideally model the *human perception* of vector-valued edges. It is particularly conceivable for color images : one may choose for instance the lightness function (perceptual response to the luminance) coming from the $CIELAB$ color base [81] :

$$f = L^* = 116\, g(Y) - 16 \qquad \text{with} \qquad Y = 0.2125R + 0.7154G + 0.0721B$$

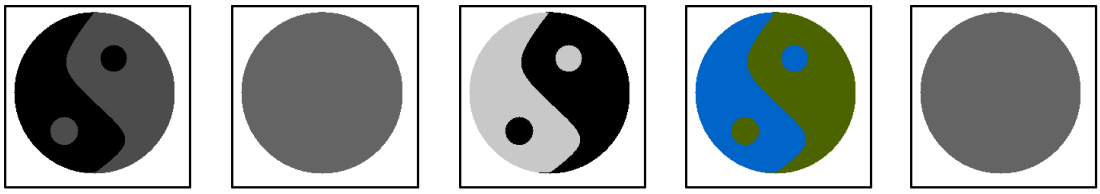
where $g : \mathbb{R} \to \mathbb{R}$ is defined by

$$\begin{cases} g(s) = \sqrt[3]{s} & \text{if} \quad s > 0.008856 \\ g(s) = 7.787s + \frac{16}{116} & \text{else} \end{cases}$$

Thus, we may define a vector-valued local vector geometry $\{\mathcal{N}, \theta_+, \theta_-\}$ of $\mathbf{I}$ by choosing

$$\begin{cases} \theta_+ = \dfrac{\nabla f(\mathbf{I})}{\|\nabla f(\mathbf{I})\|} \\ \\ \theta_- \perp \theta_+ \end{cases} \qquad \text{and} \qquad \mathcal{N} = \|\nabla f(\mathbf{I})\|$$

However, this method has two major drawbacks. First, this is not always possible to easily define a significant function f for multi-channel images (particularly when the number of channel is $n > 3$). Second, there are mathematically no functions f that can detect all possible vector-valued variations. For instance, the lightness function defined above will not be able to detect *iso-lightness* vector contours in a color image. It is the case for the image illustrated on Fig.2 : the contours inside the colored yin-yang symbol will not be detected by $\mathcal{N} = \|\nabla f(\mathbf{I})\|$, since $f(\mathbf{I})$ is constant therein. As a consequence, the smoothing performed here will be either isotropic or oriented in a wrong direction : the existing color edges inside the yin-yang symbol will be probably blurred.



(a) Red channel R (b) Green channel G (c) Blue channel B (d) Color image (R, G, B) (e) Lightness (scalar) image L^*

Figure 2: Using lightness L^* to detect geometry of a color image fail for iso-lightness contours.

1.3 Di Zenzo multi-valued geometry

In order to overcome this limitation, a very elegant solution has been proposed by Di Zenzo in [41]. He considers a multi-channel image $\mathbf{I} : \Omega \to \mathbb{R}^n$ as a vector field, and looks for the local variations of the vector norm $\|d\mathbf{I}\|^2$, mainly given by a variation matrix $\mathbf{G} = (g_{i,j})$. We get :

$$d\mathbf{I} = \mathbf{I}_x \, dx + \mathbf{I}_y \, dy \qquad \text{where} \quad \mathbf{I}_x = \frac{\partial \mathbf{I}}{\partial x} \quad \text{and} \quad \mathbf{I}_y = \frac{\partial \mathbf{I}}{\partial y} \qquad (\in \mathbb{R}^n)$$

then

$$\|d\mathbf{I}\|^2 = d\mathbf{I}^T \, d\mathbf{I} = \|\mathbf{I}_x\|^2 \, dx^2 + 2 \, \mathbf{I}_x^T \mathbf{I}_y \, dxdy + \|\mathbf{I}_y\|^2 \, dy^2$$

i.e.

$$\|d\mathbf{I}\|^2 = d\mathbf{X}^T \, \mathbf{G} \, d\mathbf{X} \qquad \text{where} \qquad \mathbf{G} = \sum_{i=1}^{n} \nabla I_i \, \nabla I_i^T \quad \text{and} \quad d\mathbf{X} = \begin{pmatrix} dx \\ dy \end{pmatrix}$$

$\mathbf{G}$ is denoted as the *structure tensor*. It sums variation contributions from each image channel I_i. It is easy to see that $\mathbf{G}$ is a 2×2 symmetric and semi positive-definite matrix. Its coefficients $(g_{i,j})$ are simply :

$$\begin{cases} g_{11} & = & \sum_{i=1}^{n} I_{i_x}^2 \\ g_{12} = g_{21} & = & \sum_{i=1}^{n} I_{i_x} I_{i_y} \\ g_{22} & = & \sum_{i=1}^{n} I_{i_y}^2 \end{cases}$$

In the common case of color images $\mathbf{I} = (R, G, B)$, $\mathbf{G}$ is defined as :

$$\mathbf{G} = \begin{pmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{pmatrix} \tag{1}$$

The interesting point about $\mathbf{G}$ is that its positive eigenvalues $\lambda_{+/-}$ give *the maximum and the minimum values of* $\|d\mathbf{I}\|^2$ while the orthogonal eigenvectors θ_+ and θ_- are the corresponding *orientations* of these extrema, and are formally given by :

$$\lambda_{+/-} = \frac{g_{11} + g_{22} \pm \sqrt{\Delta}}{2} \qquad \text{and} \qquad \theta_{+/-} /\!\!/ \begin{pmatrix} 2 \, g_{12} \\ g_{22} - g_{11} \pm \sqrt{\Delta} \end{pmatrix} \tag{2}$$

where $\Delta = (g_{11} - g_{22})^2 + 4\, g_{12}^2$. The vectors $\theta_\pm$ are normalized to the unit vector afterward.
With this simple and efficient approach, Di Zenzo opened a natural way to deal with the local vector geometry of multi-channel images, through the use of the *oriented orthogonal basis* $(\theta_+\ ,\ \theta_-)$ and the *variations measures* (λ_+, λ_-). A slight variant has been proposed by Weickert in [118]. He rather proposed to study the eigenvalues and eigenvectors of a Gaussian-smoothed version $\mathbf{G}_\sigma$ of the structure tensor $\mathbf{G}$:

$$\mathbf{G}_\sigma = \sum_{i=1}^{n} \left[\left(\nabla I_{i_\alpha} \nabla I_{i_\alpha}^T \right) * G_\sigma \right] \qquad \text{where} \qquad \nabla I_{i_\alpha} = \nabla (I_i * G_\alpha) \tag{3}$$

where G_α and G_σ are 2D Gaussian kernels with variances respectively equal to α and σ. User-defined parameters α and σ have an influence on the smoothness of the obtained structure tensor field, and by extension, on the regularity of the retrieved vector-valued image geometry. It is worth to notice that eigenvalues of $\mathbf{G}_\sigma$ are well adapted to discriminate different geometric cases :

- When $\lambda_+ \simeq \lambda_- \simeq 0$, there are very few vector variations around the current point $\mathbf{X} = (x, y)$: the region is *almost flat* and does not contain any edges or corners (it is the case for the inside of the strips in Fig.3a). For this configuration, the variation norm $\mathcal{N}$ we have to define should be low.

- When $\lambda_+ \gg \lambda_-$, there are a lot of vector variations. The current point may be located on a *vector edge* (it is the case for the edges of the strips in Fig.3a). For this configuration, the variation norm $\mathcal{N}$ should be high.

- When $\lambda_+ \simeq \lambda_- \gg 0$, we are located on a *saddle point of the vector surface*, which can be possibly a *corner structure* in the image (for instance, the intersections of the strips in Fig.3a). In this case $\mathcal{N}$ should be even higher than for the previous configuration. Regularization algorithms have indeed a tendency to smooth corners fastly. A very high variation measure estimated on corner points would attenuate the smoothing there, which is often a desired effect.

Actually, a lot of proposed regularization algorithms acting on multi-channel images have implicitly or explicitly based their smoothing behavior from these Di Zenzo's attributes. In particular, three different choices of vector gradient norms $\mathcal{N}$ have been proposed so far in the literature to measure vector-valued variations :

- $\mathcal{N} = \sqrt{\lambda_+}$, as a natural extension of the scalar gradient norm viewed as *the value of maximum variations* [20, 86, 87] (Fig.3b and Fig.4b). This norm will not particularly give importance to corners compared to straight edges.

- $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$, also called *coherence norm*, have been chosen in [89, 114, 116]. Note that this norm fails to detect discontinuities that are saddle points of the vector-valued surface. This is illustrated on the intersections of the strips (Fig.3c), as well as in the center and left-right parts of the child's eye (Fig.4c). This will mainly perturb any regularization process that uses this norm since some colored sharp corners, considered as homogeneous regions, will be probably over-smoothed.

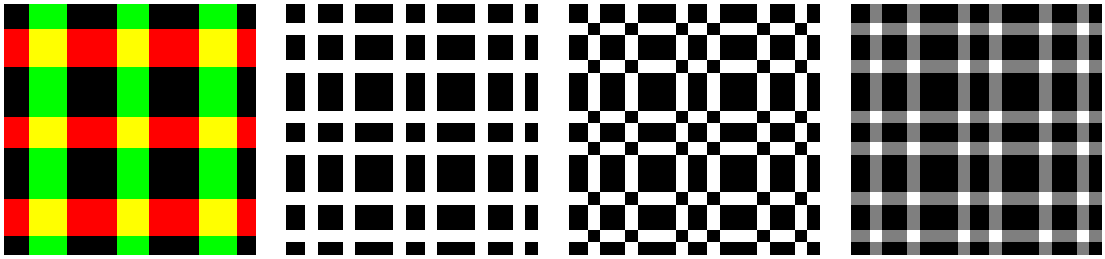
- $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$, also denoted by $\|\nabla \mathbf{I}\|$ is often chosen [14, 21, 78, 96, 104, 105] since it detects edges and corners in a good way, and is easy to compute. Indeed, it does not require an eigenvalue decomposition of $\mathbf{G}$ as the other norms did, because

$$\mathcal{N}_+ = \|\nabla \mathbf{I}\| = \sqrt{\text{trace}(\mathbf{G})} = \sqrt{\sum_{i=1}^{n} \|\nabla I_i\|^2} \tag{4}$$

  Moreover, the norm $\mathcal{N}_+$ has the interesting property of giving preferences to certain corners (Fig.3d). This is very valuable for image restoration purposes, since the smoothing can be attenuated on high-curvature structures which are classically hard to preserve.

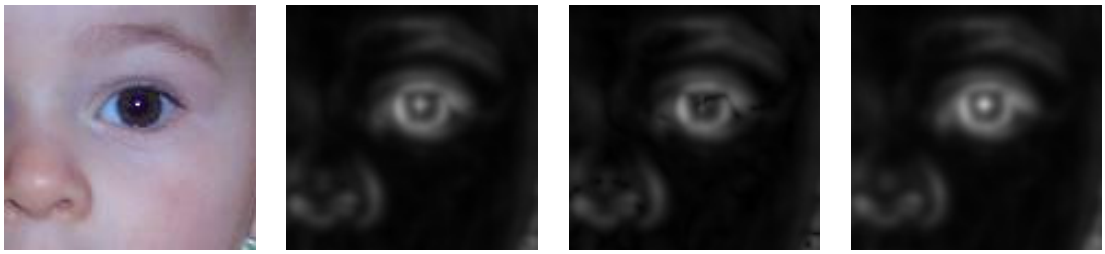
Note that for the scalar case ($n = 1$), the structure tensor calculus reduces to :

$$\text{when } n = 1\ , \quad \|dI\|^2 = d\mathbf{X}\ \mathbf{G}^1\ d\mathbf{X} \quad \text{where} \quad \mathbf{G}^1 = \nabla I \nabla I^T = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$



(a) Color checkerboard (size 40×40) (b) $\mathcal{N} = \sqrt{\lambda_+}$ (c) $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ (d) $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$

Figure 3: Comparing possible vector variation norms $\mathcal{N}, \mathcal{N}_-$ and $\mathcal{N}_+$ for a synthetic color image.



(a) Color photograph (small portion 60×40) (b) $\mathcal{N} = \sqrt{\lambda_+}$ (c) $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ (d) $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$

Figure 4: Comparing possible vector variation norms $\mathcal{N}, \mathcal{N}_-$ and $\mathcal{N}_+$ for a real color image.

In this case, the eigenvectors $\theta^1_{+/-}$ and the eigenvalues $\lambda^1_{+/-}$ of $\mathbf{G}^1$ are :

$$\begin{cases} \theta^1_- = \xi = \dfrac{\nabla I^\perp}{\|\nabla I\|} \\ \theta^1_+ = \eta = \dfrac{\nabla I}{\|\nabla I\|} \end{cases} \quad \text{associated to} \quad \begin{cases} \lambda^1_- = 0 \\ \lambda^1_+ = \|\nabla I\|^2 \end{cases}$$

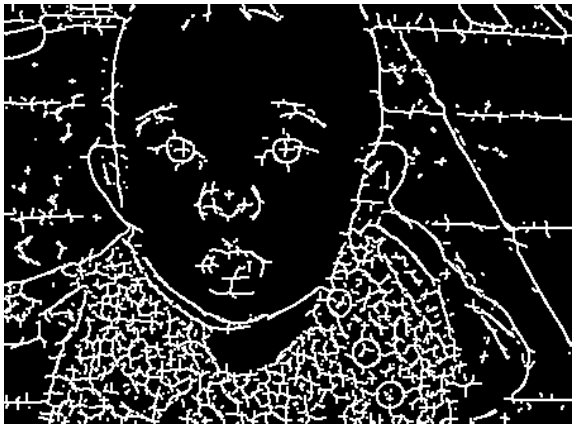
Basically, it means that the three above defined norms $\mathcal{N}_+$, $\mathcal{N}_-$ and $\mathcal{N}$ all reduce to $\|\nabla I\|$ in the case of scalar-valued images, which is a desired property.

Once a local vector geometry is defined, we can use it as a measure in many image analysis processes involving multi-channel images (not only for regularization algorithms). For instance, color edge detection may be performed by finding thresholded local maxima of the $\mathcal{N}_+$ norm (Fig.5 and [66, 103, 107]). This vector geometry computation has also been integrated as a measure of contours in some multi-channel image segmentation methods [86, 87].
For all reasons given above, the norm $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$ associated to the Di Zenzo geometry is probably one of the best measure for detecting local variations in multi-channel images and will be considered as it in the next parts of this chapter.

2 PDE-based Smoothing of Multi-Valued Images : A Review

We review and propose a classification of classical smoothing methods based on diffusion PDE's into three different approaches, related to different interpretation levels of the regularization processes, from the most global to the most local ones. For each section, we will start describing the original idea for scalar-valued images, then extending it for multi-channel datasets.

(a) Color image



(b) Detecting color edges with the norm $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$

Figure 5: Using a vector variation norm for color edge detection.

2.1 Variational methods

Contrary to the formulation of the original Perona-Malik equation, several methods have been proposed to apprehend the problem of image regularization as a global minimization procedure, within a variational framework. Formalisms described in [9, 25, 32, 88, 118] among numerous references, contributed to define generic energy functionals measuring global image variations. The idea is that minimizing adapted variation functionals will flatten low image variations (then gradually remove the noise), while preserving the high ones (avoiding the smoothing of image contours). The formulation of the ϕ-*functionals* gathers some of these approaches in a general framework and gives a very unifying way to proceed :
A noisy *scalar image* I_{noisy} can be regularized by minimizing the following ϕ-functional :

$$\min_{I:\Omega\to\mathbb{R}} E(I) = \int_\Omega \phi(\|\nabla I\|)\, d\Omega \tag{5}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is *an increasing function*, directing the regularization behavior and penalizing high gradient norms. The minimization is performed via the corresponding *diffusion PDE evolution*, coming from the Euler-Lagrange equations of $E(I)$:

$$\left\{ \begin{array}{l} I_{(t=0)} = I_{\text{noisy}} \\ \\ \dfrac{\partial I}{\partial t} = \text{div}\left(\dfrac{\phi'(\|\nabla I\|)}{\|\nabla I\|}\, \nabla I \right) \end{array} \right. \tag{6}$$

Different choices of functions ϕ lead to different proposed regularization methods. One especially finds the simple isotropic smoothing (equivalent to a Gaussian convolution), as introduced by Tikhonov [99], as well as the well-known Perona-Malik [80] and Total Variation (TV) anisotropic flows [85]. Lot of regularization methods acting on scalar-valued images have been unified by the ϕ-function formalism (Fig.6).

Function name	$\phi(s)$	Reference
Tikhonov	s^2	[99]
Perona-Malik	$1 - \exp(-s^2/K^2)$	[80]
Minimal surfaces	$2\sqrt{1+s^2} - 2$	[31]
Geman-McClure	$s^2/(1+s^2)$	[46]
Total Variation	s	[85]
Green	$2\log(\cosh(s))$	[52]

Figure 6: List of different ϕ-functions and corresponding references.

A natural extension of the ϕ-functionals for the regularization of *multi-channel* images $\mathbf{I}$ could consist in minimizing the following cost functional $E(\mathbf{I})$ measuring a global multi-channel image variation :

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} E(\mathbf{I}) = \int_\Omega \phi(\mathcal{N}(\mathbf{I}))\ d\Omega \tag{7}$$

where $\mathcal{N}(\mathbf{I})$ is one of the three local variation norms defined in section (1).
But more generally, as vector-valued images possess two distinct variation measures λ_+ and λ_- (eigenvalues of the structure tensor $\mathbf{G}$) contrary to a single measure $\|\nabla I\|$ for scalar images, it seems rather quite natural to minimize a functional defined by a function $\psi : \mathbb{R}^2 \to \mathbb{R}$ of two variables instead of a single one. The ψ-functional below is thus a more complete extension of the ϕ-function formulation for multi-channel images.

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} E(\mathbf{I}) = \int_\Omega \psi(\lambda_+, \lambda_-)\ d\Omega \tag{8}$$

The Euler-Lagrange equations of (8) can be derived, and reduce to a simple form of divergence-based PDE (see Appendix A for details about this Euler-Lagrange derivation) :

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\left(\left[\frac{\partial \psi}{\partial \lambda_+}\ \theta_+\theta_+^T + \frac{\partial \psi}{\partial \lambda_-}\ \theta_-\theta_-^T\right]\nabla I_i\right) \qquad (i = 1..n) \tag{9}$$

The choice of specific cases of ψ-functions leads to previous vector-valued regularization approaches defined as variational methods, such as the whole range of vector-valued ϕ-functionals [21, 78, 97] :

$$\psi(\lambda_+, \lambda_-) = \phi(\sqrt{\lambda_+ + \lambda_-})$$

or the Beltrami flow framework [58, 59, 60, 92, 94, 93] :

$$\psi(\lambda_+, \lambda_-) = \sqrt{(1+\lambda_+)(1+\lambda_-)}$$

Note that this last approach is also equivalent to define the minimizing functional $E(\mathbf{I})$ as a Polyakov action which is actually a physical measure of the area of the image $\mathbf{I}$ seen as a $2D$ surface embedded in a $(n+2)$D space. This geometric interpretation helps in understanding how functional minimization can play a role in smoothing images by forcing them to be more regular, here by finding a minimal surface (Fig.7).

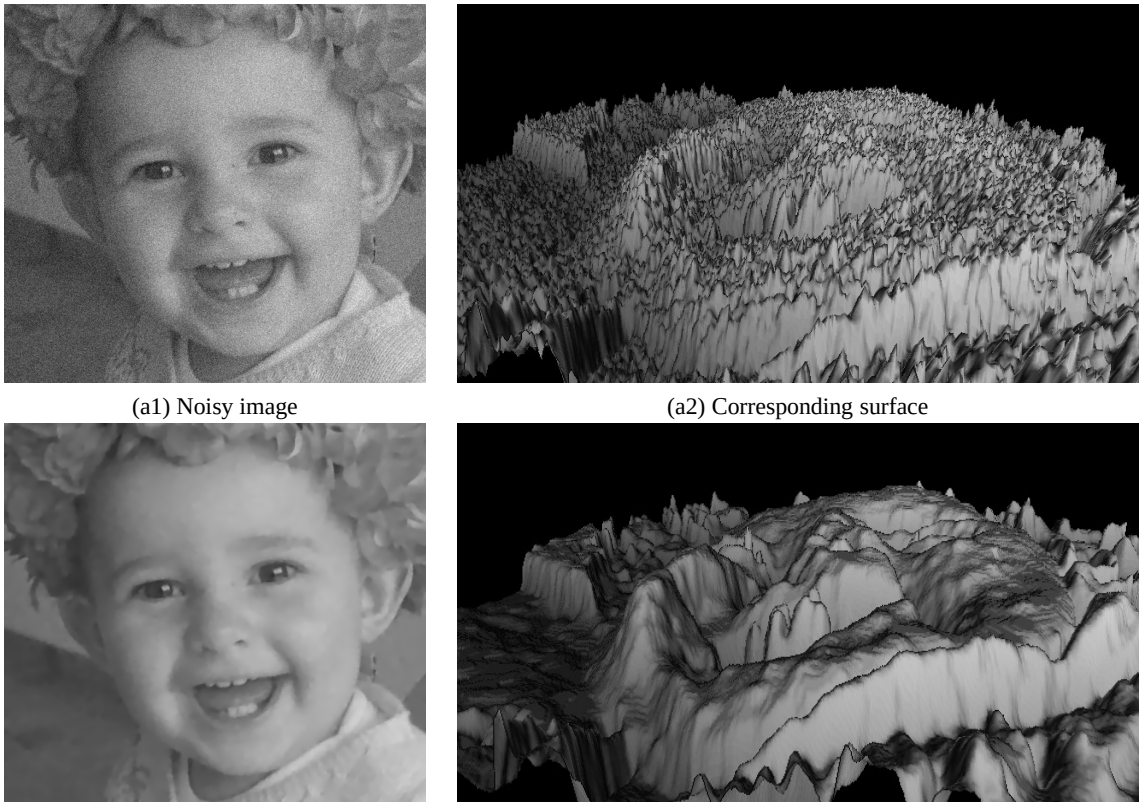
Despite the interesting global geometric interpretation of variational formulations, such methods clearly lacks in flexibility. Indeed, they are formulated as global minimizations processes, despite the local geometric smoothing properties that are intrinsically desired for regularization purposes. Such PDE's are obtained by the Euler-Lagrange derivation of a functional and cannot thus be finely tuned to adapt themselves to local geometric cases (contours, corners, etc.). Unfortunately, this adaptability is primordial in many situations especially when the level of noise is high.

2.2 Divergence-based diffusion PDE's

One level of flexibility for designing regularization PDE's has been reached with the introduction of more generic divergence expressions [9, 5, 65, 88, 118]. Basically, the idea was to replace the function $\phi'(\|\nabla I\|)/\|\nabla I\|$ in the divergence of the scalar-valued PDE (6) by expressions depending on more appropriate image features. In one hand, this gives more freedom to design regularization PDE's that better fit local constraints. On the other hand, one often loses the *global interpretation* of the regularization process : generally, such designed equations do not correspond to a functional minimization anymore.
Historically, authors of [5] first proposed to use a diffusivity function $g(\|\nabla I * G_\sigma\|)$ depending on the convolved gradient norm $\|\nabla I * G_\sigma\|$, rather than simply considering $\|\nabla I\|$ as a measure of image variations, for the regularization of scalar-valued images :

$$\frac{\partial I}{\partial t} = \mathrm{div}\left(g(\|\nabla I * G_\sigma\|)\ \nabla I\right)$$

where G_σ is a $2D$ normalized Gaussian function.
This has initially been done to ensure the well-posedness of the regularization formulations. But, it also appeared that it allowed to respect a more coherent local diffusion geometry by involving a *larger neighborhood* in the computation of the local image variations that influence the smoothing process.



(a1) Noisy image (a2) Corresponding surface

(b1) Restored image (b2) Corresponding surface

Figure 7: Example of image denoising by surface area minimization.

A major generalization of divergence-based equations for scalar and multi-channel images has been more recently proposed by Weickert in [115, 116, 117, 118]. Basically, the idea consists in considering image pixels as chemical concentrations or temperatures which diffuse with respect to some physical laws (Fick Law and continuity equations). He proposed this very generic divergence-based equation, parametrized by a field $\mathbf{D} : \Omega \to \mathcal{P}(2)$ of 2×2 diffusion tensors :

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\,(\mathbf{D}\nabla I_i) \qquad (i = 1..n) \tag{10}$$

The tensor field $\mathbf{D}$ defines a *gradient flux* and controls then the local diffusion behavior of the smoothing process (10). Note that the ψ-functional formalism described in previous section (2.1) is just a particular case of the PDE (10) with $\mathbf{D}$ defined as :

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

More specifically, Weickert proposed to design the diffusion tensor $\mathbf{D}$ for each image point $\mathbf{X} = (x, y)$, by selecting its two eigenvectors $\mathbf{u}, \mathbf{v}$ and eigenvalues λ_1, λ_2 as functions of the spectral elements of the smoothed structure tensor $\mathbf{G}_\sigma$ (3) such that :

$$\left\{ \begin{array}{l} \mathbf{u} = \theta^+ \\ \\ \mathbf{v} = \theta^- \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \lambda_1 = \beta \\ \\ \lambda_2 = \left\{ \begin{array}{ll} \beta & (\text{if } \lambda_+ = \lambda_-) \\ \beta + (1-\beta)\exp\left(\frac{-C}{(\lambda_+ - \lambda_-)^2}\right) & \text{else} \end{array} \right. \end{array} \right. \tag{11}$$

($C > 0$ and $\beta \in [0, 1]$ are user-fixed parameters of the method).

The tensor $\mathbf{D}$ is computed at each image point as : $\mathbf{D} = \lambda_1 \mathbf{u}\mathbf{u}^T + \lambda_2 \mathbf{v}\mathbf{v}^T$.

It is worth to notice that the tensor field $\mathbf{D}$ is the same for all image channels I_i, ensuring that all I_i are smoothed by a *common multi-channel geometry*, which takes the correlation between image channels into account (since $\mathbf{D}$ depends on $\mathbf{G}_\sigma$), contrary to a uncorrelated channel-by-channel approach.

Weickert assumed that the tensor shape at each point $\mathbf{X} = (x, y)$ of the field $\mathbf{D}$ give the preferred smoothing geometry at this point. The idea behind the choice (11) was then :

- On almost constant regions, we have $\lambda_+ \simeq \lambda_- \simeq 0$ and then we will get $\lambda_1 \simeq \lambda_2 \simeq \beta$, i.e $\mathbf{D} \simeq \alpha\, \mathbb{I}_\mathrm{d}$. The tensor $\mathbf{D}$ is then defined to be *isotropic* in flat regions.
- Along image contours, we have

  $$\lambda_+ \gg \lambda_- \gg 0 \qquad \text{and as a result,} \qquad \lambda_2 > \lambda_1 > 0$$

  Here, the diffusion tensor $\mathbf{D}$ will be then *anisotropic*, mainly directed by the smoothed direction θ^- of the image contours.

However, it is important to notice that the amplitudes and directions of the local smoothing performed by the divergence-based PDE (10) are actually not precisely defined by the eigen characteristics (shapes) of the diffusion tensor $\mathbf{D}$ at $\mathbf{X}$. This may lead to a smoothing behavior that is not expected, as illustrated by the simple following example. 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 of 1. This is of course for illustration purposes, since all image discontinuities would be destroyed by choosing such a smoothing geometry. Intuitively, we would define $\mathbf{D}$ at each point $\mathbf{X} \in \Omega$ as :

$$\forall \mathbf{X} \in \Omega, \quad \mathbf{D}_{(\mathbf{X})} = \left(\frac{\nabla I}{\|\nabla I\|}\right)\left(\frac{\nabla I}{\|\nabla I\|}\right)^T$$

leading to the simplification of (10) as

$$\frac{\partial I}{\partial t} = \mathrm{div}\left(\frac{1}{\|\nabla I\|^2}\, \nabla I \nabla I^T \nabla I\right) = \mathrm{div}\,(\nabla I) = \Delta I$$

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 [62], the evolution of this well known *heat flow equation* is similar to the convolution of the image I by a normalized Gaussian kernel G_σ with a variance $\sigma = \sqrt{2\,t}$. So, this particular choice of *anisotropic* tensors $\mathbf{D}$ leads to an *isotropic* smoothing behavior, without preferred smoothing orientations. Note that choosing $\mathbf{D} = \mathbb{I}_\mathrm{d}$ (identity matrix) would give exactly the same result : different tensors fields $\mathbf{D}$ with very different shapes (isotropic or anisotropic) may define the same regularization behavior. Actually, the divergence is a differential operator, so the equation (10) implicitly depends on the *spatial variations* of the tensor field $\mathbf{D}$. Clearly, the divergence equation (10) hampers the design of a significant *pointwise* smoothing behavior.

2.3 Oriented heat flows

Oriented heat flows, also named oriented Laplacians formulations consider that a local smoothing process can be decomposed into two orthogonal and uni-dimensional heat flows respectively oriented along two directions $\mathbf{u}_1$ and $\mathbf{u}_2$ (these vectors forming an orthonormal basis) associated with two smoothing amplitudes c_1 and c_2. The smoothing amplitudes and orientations are naturally different for each image point, since they adapt themselves to the local configuration of the image (Fig.8). The resulting equation is written as the sum of these two heat flows :

$$\frac{\partial \mathbf{I}}{\partial t} = c_1 \mathbf{I}_{\mathbf{u}_1\mathbf{u}_1} + c_2 \mathbf{I}_{\mathbf{u}_2\mathbf{u}_2} \tag{12}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are unit orthogonal vectors and $c_1, c_2 \geq 0$.
$\mathbf{I}_{\mathbf{u}_1\mathbf{u}_1}$ and $\mathbf{I}_{\mathbf{u}_2\mathbf{u}_2}$ denote the second derivatives of $\mathbf{I}$ in the directions $\mathbf{u}_1$ and $\mathbf{u}_2$ and their vector components are formally defined as :

$$\forall i = 1..n, \quad I_{i_{\mathbf{u}_1\mathbf{u}_1}} = \mathbf{u}_1^T \mathbf{H}_i \mathbf{u}_1 \qquad \text{and} \qquad I_{i_{\mathbf{u}_2\mathbf{u}_2}} = \mathbf{u}_2^T \mathbf{H}_i \mathbf{u}_2$$

where $\mathbf{H}_i$ is the Hessian of I_i, defined on each point $\mathbf{X} \in \Omega$ by

$$\mathbf{H}_i = \begin{pmatrix} I_{i_{xx}} & I_{i_{xy}} \\ I_{i_{xy}} & I_{i_{yy}} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 I_i}{\partial x^2} & \frac{\partial^2 I_i}{\partial x \partial y} \\ \frac{\partial^2 I_i}{\partial x \partial y} & \frac{\partial^2 I_i}{\partial y^2} \end{pmatrix} \tag{13}$$

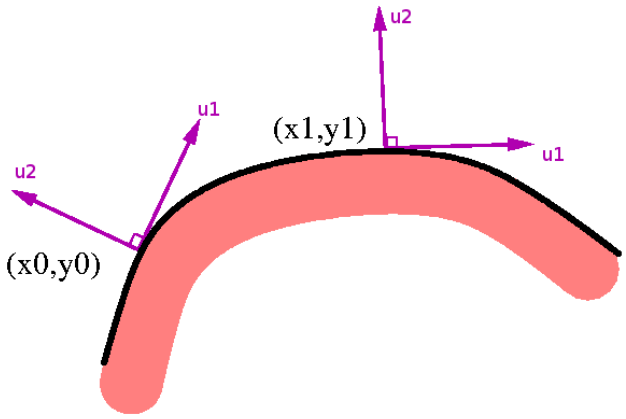


Figure 8: Principle of oriented Laplacians : Two 1D smoothing are done along adapted directions.

Here, the diffusion behavior is entirely defined by the knowledge of the smoothing directions $\mathbf{u}_1, \mathbf{u}_2$ and the corresponding weights c_1 and c_2.

This formulation has been first proposed in [63, 64] for the regularization of scalar-valued images I, with the following choices for c_1, c_2 and $\mathbf{u}_1, \mathbf{u}_2$:

$$\begin{cases} \mathbf{u}_1 = \xi = \frac{\nabla I^{\perp}}{\|\nabla I\|} \\ \mathbf{u}_2 = \eta = \frac{\nabla I}{\|\nabla I\|} \end{cases} \quad \text{and} \quad \begin{cases} c_1 = 1 \\ c_2 = g(\|\nabla I * G_\sigma\|) \end{cases}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a function decreasing to 0 (the pixel diffusion must vanish on high gradients). It allows a *permanent anisotropic smoothing along the edges* ξ, even on very high gradients since $c_1 = 1$ everywhere in the image. The general formalism of oriented Laplacians (12) allows also to find other well-known equations, such as *the mean curvature flow* $\frac{\partial I}{\partial t} = I_{\xi\xi}$, obtained with $c_1 = 1$, $c_2 = 0$, $\mathbf{u}_1 = \xi$ and $\mathbf{v}_2 = \eta$ [39]. Note that this was *not possible* to get with divergence-based expressions (10). Other works for scalar image regularization has used similar variants of this technique [24, 67].

Ringach and Sapiro [89] proposed an extension of the mean curvature flow $I_t = I_{\xi\xi}$ for multi-channel images, using an oriented Laplacian formulation (12). They naturally used the Di Zenzo attributes to incorporate informations on the multi-channel geometry in their proposed equation :

$$\frac{\partial \mathbf{I}}{\partial t} = g(\lambda_+ - \lambda_-)\, \mathbf{I}_{\theta_- \theta_-} \tag{14}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a positive decreasing function, avoiding the smoothing of high gradients regions. It was one of the first attempts to construct an oriented Laplacian PDE directly from a local vector-valued geometry. Indeed, all channels I_i are smoothed along a *common vector edge direction* with a *common intensity*. Despite this great idea, some drawbacks subsist :

- The coherence norm $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ is used here as a measure of vector-valued variations, in order to reduce diffusion on image contours. This may not be a good choice since some corner structures will not respond highly to the norm $\mathcal{N}_-$, as explained in section (1), and will be then over-smoothed.

- In flat regions ($\mathcal{N}_- \to 0$), the diffusion is made along a single direction θ_-, which is mainly directed by the noise since no coherent structures exist in these regions. Undesired texture effects result from this mono-directional smoothing. This is particularly true here, since contrary to decoupled regularizations, vector components are *not blended* with this method (the diffusions in all image channels I_i follow a common direction). Isotropic smoothing would be more adapted in order to remove noise in such flat regions.

2.4 Trace-based diffusion PDE's

A simple generalization of oriented Laplacians have been proposed in [101, 109, 110]. The idea relies on the use of a generic diffusion tensor field $\mathbf{T} : \Omega \to \mathcal{P}(2)$ to describe the diffusion geometry of the equation (12), instead of separately describing local directions θ_+, θ_- and amplitudes c_1, c_2 of smoothing. Actually, the proposed equation was just a rewrite of the previous PDE (12), using a *trace* operator :

$$\forall i = 1, .., n, \qquad \frac{\partial I_i}{\partial t} = \text{trace}\,(\mathbf{T}\mathbf{H}_i) \tag{15}$$

where $\mathbf{H}_i$ stands for the Hessian of I_i (13) and the tensor field $\mathbf{T}$ is computed as :

$$\forall \mathbf{X} \in \Omega, \quad \mathbf{T}_{(\mathbf{X})} = c_1\ \mathbf{u}_1\mathbf{u}_1^T + c_2\ \mathbf{v}_2\mathbf{v}_2^T$$

Note that in this case, each channel I_i of $\mathbf{I}$ is also smoothed with a common tensor field $\mathbf{T}$.
Actually, equations (12) and (15) are strictly equivalent, but this last one makes clearly appear the separation of the smoothing geometry (defined by the tensor field $\mathbf{T}$) from the smoothing itself. This is actually close to the idea of the Weickert's method that led to the divergence PDE (10) : the regularization problem simplifies now to the design of a tensor field $\mathbf{T}$ adapted to the considered application. But in the case of trace-based PDE's, the tensor field that defines the local smoothing behavior has the interesting property of *unicity* : two *different* tensor fields will necessarily lead to *different* smoothing behaviors. Indeed, equation (15) has a simple geometric interpretation in terms of *local filtering* with oriented Gaussian kernels.
Indeed, let consider first that $\mathbf{T}$ is a constant tensor field. Then, it can be demonstrated that the formal solution of the PDE (15) is (see Appendix B for details) :

$$I_{i_{(t)}} = I_{i_{(t=0)}}\ *\ G^{(\mathbf{T},t)} \qquad (i = 1..n) \tag{16}$$

where $*$ stands for the convolution operator and $G^{(\mathbf{T},t)}$ is an *oriented Gaussian kernel*, defined by :

$$G^{(\mathbf{T},t)}(\mathbf{X}) = \frac{1}{4\pi t}\exp\left(-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right) \quad \text{with} \quad \mathbf{X} = (x \quad y)^T \tag{17}$$

This is in fact a generalization of the Koenderink's idea [62], who proved this property in the field of computer vision for the simpler case of the *isotropic diffusion tensor* $\mathbf{T} = \mathbb{I}_{\mathrm{d}}$, resulting in the well-known *heat flow* equation : $\frac{\partial I_i}{\partial t} = \Delta I_i$.
Fig.9 illustrates three Gaussian kernels $G^{(\mathbf{T},t)}(x, y)$ respectively obtained with isotropic and anisotropic tensors $\mathbf{T}$ and the corresponding evolutions of the diffusion PDE (15) on a color image. It is worth to notice that the Gaussian kernels $\mathbf{G}^{(\mathbf{T},t)}$ give the classical representations of the tensors $\mathbf{T}$ with ellipses. Conversely, it is clear that the tensors $\mathbf{T}$ represent the exact geometry of the smoothing performed by the PDE (15).
When $\mathbf{T}$ is not constant (which is generally the case), i.e. represents a field $\Omega \to \mathrm{P}(2)$ of variable diffusion tensors, the PDE (15) becomes *nonlinear* but can be viewed as the application of temporally and spatially varying *local masks* $G^{\mathbf{T},t}(\mathbf{X})$ over the image $\mathbf{I}$. Fig.10 illustrates three examples of spatially varying tensor fields $\mathbf{T}$, represented with fields of ellipsoids, and the corresponding evolutions of (15) on a color image. As before, *the shape of each tensor* $\mathbf{T}$ *gives the exact geometry of the local smoothing process* performed by the trace-based PDE (15) point by point. As the trace is not a differential operator, the local interpretation of the smoothing process as a convolution with an oriented Gaussian mask is valid here.

In the same way that structure tensors code for each image pixel $\mathbf{X}$ the main directions of the edges θ_- as well as the edge strength $\lambda_+ + \lambda_-$, the diffusion tensor field $\mathbf{T}$ will code similarly the preferred local smoothing directions as well as the desired smoothing amplitudes along these directions, for each image pixel $\mathbf{X}$. Of course, $\mathbf{T}_{(\mathbf{X})}$ must depend on the local geometry of $\mathbf{I}$, and is thus defined from the spectral elements λ^-, λ^+ and θ^-, θ^+ of the smoothed structure tensor $\mathbf{G}_\sigma$. For image denoising purposes, the choice proposed in [101, 109, 110] is :

$$c_1 = f^-_{(\lambda_+,\lambda_-)} = \frac{1}{(1+\lambda^+ + \lambda^-)^{p_1}} \quad \text{and} \quad c_2 = f^+_{(\lambda_+,\lambda_-)} = \frac{1}{(1+\lambda^+ + \lambda^-)^{p_2}} \quad \text{with} \quad p_1 < p_2 \tag{18}$$

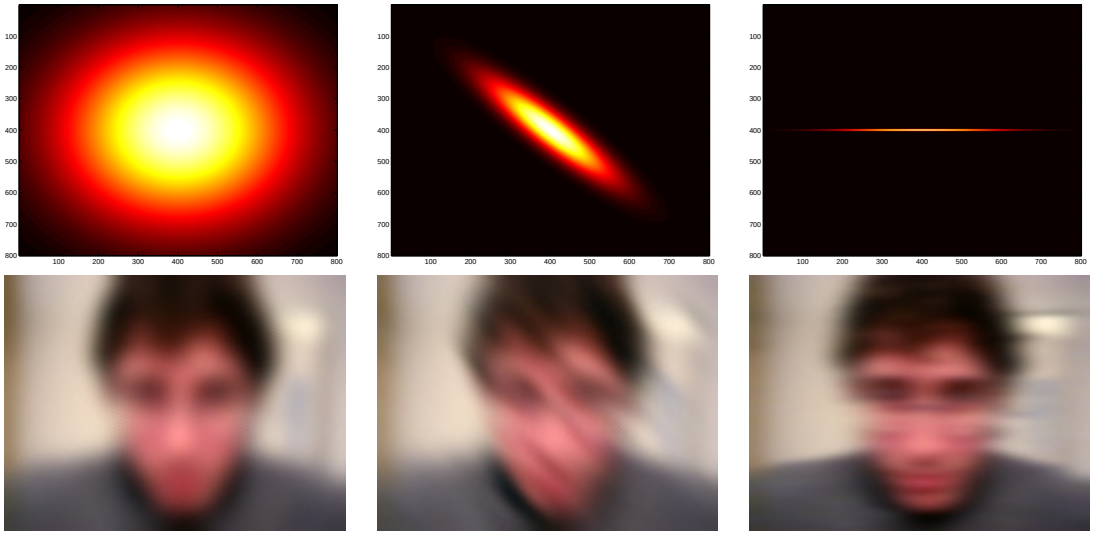


Figure 9: Trace-based PDE's (15) viewed as convolutions by oriented 2D Gaussian kernels.

Figure 10: Trace-based PDE's (15) with non-constant diffusion tensor fields $\mathbf{T}$.

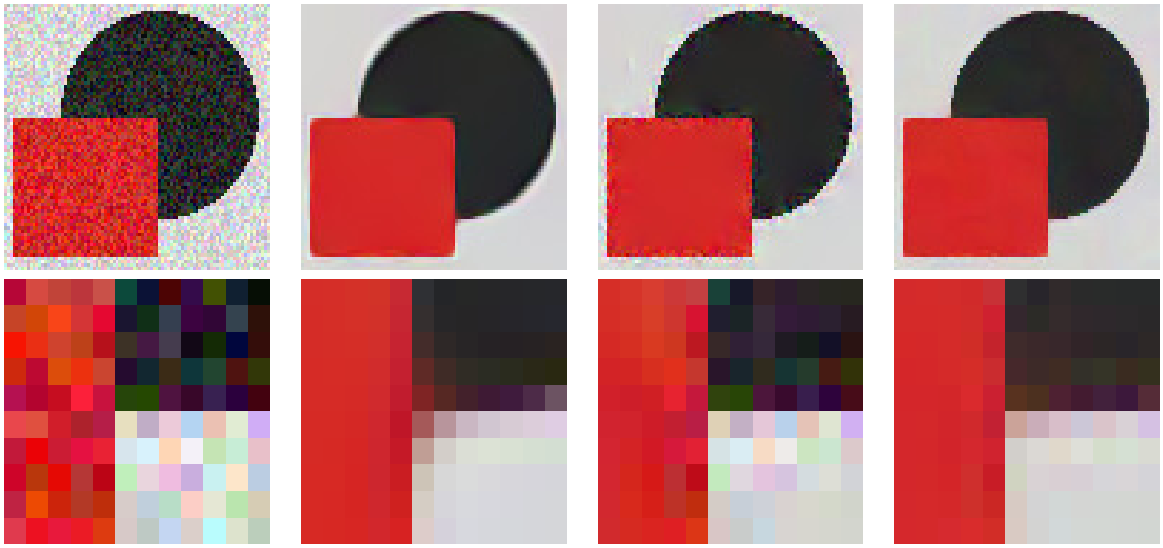
where $p_1, p_2 \in \mathbb{R}$ are parameters of the proposed method.

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. Actually, this is quite natural

to design a smoothing behavior from the image structure *before* applying the regularization process itself.
The trace-based equation (15) has been a great attempt to separate the smoothing geometry from the smoothing process itself, while providing a geometrical interpretation on how the smoothing is performed. It proved some natural links between PDE's and other local filtering techniques, as the Bilateral Filtering [11, 100]. Another similar approach based on non-Gaussian convolution kernels has been also proposed for the specific case of the Beltrami Flow in [92].
But the fact that the trace equation (15) behaves locally as an oriented Gaussian smoothing whose strength and orientation is directly related to the tensor $\mathbf{T}_{(\mathbf{X})}$ has a major drawback. Indeed, 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 the smoothing 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.11b and Fig.12b where (15) has been applied on synthetic and real color image and $\mathbf{T}$ has been defined as (18) (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.11b, or in blending parallel thin curved structures in Fig.12b.
To avoid this over-smoothing effect, one may try to stop the action of the diffusion PDE 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 imprecise in the presence of noise, even when using the Di Zenzo geometry. Conversely, image under-smoothing on edges may occur when limiting the diffusion too much on regions with high intensity variations (Fig.11c). There is a difficult trade-off between complete noise removal and preservation of curved structures, when using trace-based PDE's (15).



(a) Noisy synthetic color image

(b) Applying trace-based PDE (15), with $p_1 = 0.5, p_2 = 1.2$.

(c) Applying trace-based PDE (15), with $p_1 = 0.9, p_2 = 1.2$.

(d) Applying our constrained PDE (29), with $p1 = 0.5, p_2 = 1.2$.

Figure 11: Problems encountered when using trace-based PDE's (15) on curved image structures.

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 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, the method will locally filter the image with *curved Gaussian kernels* when necessary, in order to better preserve image structures. For comparison purposes, results of the curvature-preserving equation is shown on Fig.11d and Fig.12c.

2.5 Links between existing regularization methods

The link between these three formulations is generally not trivial, especially for vector-valued images. Actually, it is well known for the classical case of ϕ-functional regularization of *scalar* images ($n = 1$) : One can start from a regularizing

(a) Image of a fingerprint

(b) Applying trace-based PDE (15), with $p_1 = 0.5, p_2 = 1.2$.

(c) Applying our constrained PDE (29), with $p1 = 0.5, p_2 = 1.2$.

Figure 12: Comparisons between trace-based PDE's (15) and our new curvature-preserving PDE's (29) on a real image.

functional minimization (A) and find the corresponding divergence-based (B) and oriented Laplacians (C) formulations :

$$\begin{aligned} & (A) \ : \quad \min_{I:\Omega\to\mathbb{R}} \int_\Omega \phi(\|\nabla I\|)\, d\Omega \\ \Rightarrow \quad & (B) \ : \quad \frac{\partial I}{\partial t} = \mathrm{div}\left(\frac{\phi^{'}(\|\nabla I\|)}{\|\nabla I\|}\,\nabla I\right) \\ \Rightarrow \quad & (C) \ : \quad \frac{\partial I}{\partial t} = \frac{\phi^{'}(\|\nabla I\|)}{\|\nabla I\|}\, I_{\xi\xi} + \phi^{''}(\|\nabla I\|)\, I_{\eta\eta} \end{aligned} \tag{19}$$

where $\eta = \nabla I/\|\nabla I\|$ and $\xi \perp \eta$. Note that this regularization generally leads to *anisotropic smoothing* (in the sense that it is performed in privileged spatial directions with different weights), despite the *isotropic shape* of the underlying tensor $\mathbf{D} = \frac{\phi^{'}(\|\nabla I\|)}{\|\nabla I\|}\,\mathbb{I}_{\mathrm{d}}$ in the divergence expression. It is also worth to mention that this global-to-local path (from variational to trace-based equations) can be rarely be followed in the inverse order.

For multi-channel images, this link can be also found :

$$\begin{aligned} & (A) \quad \min_{\mathbf{I}:\Omega\to\mathbb{R}^n} \int_\Omega \psi(\lambda_+, \lambda_-)\, d\Omega \\ \Rightarrow \quad & (B) \quad \frac{\partial I_i}{\partial t} = \mathrm{div}\,(\mathbf{D}\,\nabla I_i) \qquad \text{where} \quad \mathbf{D} = \frac{\partial \psi}{\partial \lambda_+}\,\theta_+\theta_+^T + \frac{\partial \psi}{\partial \lambda_-}\,\theta_-\theta_-^T \\ \Rightarrow \quad & (C) \quad \frac{\partial I_i}{\partial t} = \sum_{j=1}^{n} \mathrm{trace}\left((\delta_{ij}\mathbf{D} + \mathbf{Q}^{ij})\mathbf{H}_j\right) \end{aligned} \tag{20}$$

where the δ_{ij} is the Kronecker symbol ($\delta_{ij} = 1$ when $i = j$, and 0 elsewhere) , $\mathbf{Q}^{ij}$ designates a *family of* n^2 *tensors*

$(i,j=1..n)$, defined as the symmetric parts of the following matrices $\mathbf{P}^{ij}$ (i.e, $\mathbf{Q}^{ij}=(\mathbf{P}^{ij}+\mathbf{P}^{ij^T})/2$) :

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

with

$$\alpha=\tfrac{f_1(\lambda_+,\lambda_-)-f_2(\lambda_+,\lambda_-)}{\lambda_+-\lambda_-}\ \text{ and }\ \beta=\tfrac{\lambda_+f_2(\lambda_+,\lambda_-)-\lambda_-f_1(\lambda_+,\lambda_-)}{\lambda_+-\lambda_-}$$

and

$$f_1=\frac{\partial\psi}{\partial\lambda_+}\quad\text{and}\quad f_2=\frac{\partial\psi}{\partial\lambda_-}$$

The development (A)⇒(B) from the functional to the divergence formulation is detailed in Appendix A. The development (B)⇒(C) from the divergence to the trace-based equation is detailed in Appendix C. This last development initially proposed in [109, 110] unifies a whole range of previously proposed vector-valued regularization algorithms (variational and divergence based PDE's) into an extended trace-based equation, composed of *several channel-diffusion contributions* that have direct geometric interpretations in terms of local filtering by Gaussian kernels. Though the geometric interpretation of the overall sum of trace equations is not direct, it is interesting to see that *additional diffusion tensors* $\mathbf{Q}^{ij}$ clearly appear in the trace expressions, and contribute to modify the inner tensor $\mathbf{D}$, which is finally not representative of the smoothing behavior. More generally, tensors appearing in traces and divergences generally lead to different smoothing behaviors.

3 Curvature-Preserving PDE's

The framework of *curvature-preserving PDE's*, first introduced in [102] defines a specific variant of multi-channel diffusion PDE's. Its goal is to provide a generic tensor-driven regularization method as the divergence-based PDE (10) and trace-based PDE (15), but also focuses on the preservation of thin curved structures. We review this very efficient formalism and show how it can be understood from a local smoothing geometry viewpoint.

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 $\mathbf{w}_{(\mathbf{X})}=(u_{(\mathbf{X})}\ \ v_{(\mathbf{X})})^T$.
The *curvature-preserving* regularization PDE that smoothes $\mathbf{I}$ along $\mathbf{w}$ is defined by :

$$\forall i=1,\ldots,n,\qquad \frac{\partial I_i}{\partial t}=\text{trace}\left(\mathbf{w}\mathbf{w}^T\,\mathbf{H}_i\right)+\nabla I_i^T\mathbf{J}_{\mathbf{w}}\mathbf{w}\tag{21}$$

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

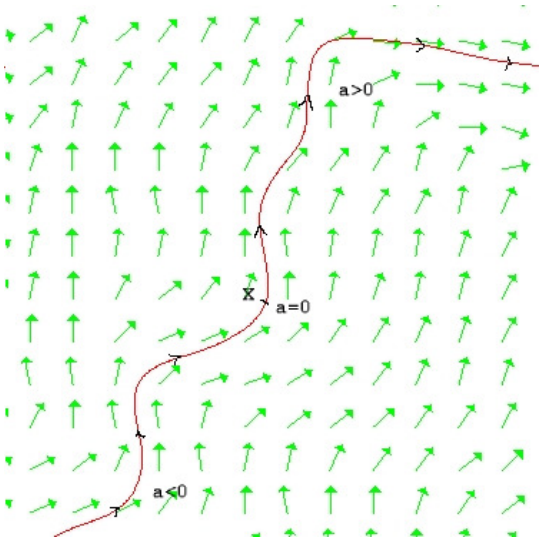
$$\mathbf{J}_{\mathbf{w}}=\begin{pmatrix}\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y}\\ \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}\end{pmatrix}\quad\text{and}\quad\mathbf{H}_i=\begin{pmatrix}\frac{\partial^2 I_i}{\partial x^2} & \frac{\partial^2 I_i}{\partial x\partial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

The PDE (21) adds a term $\nabla I_i^T\mathbf{J}_{\mathbf{w}}\mathbf{w}$ to the trace-based equation (15) that smoothes $\mathbf{I}$ along $\mathbf{w}$ with locally oriented Gaussian kernels (see section 2.4). This extra term naturally depends on the variation of the vector field $\mathbf{w}$. Let us explain how (21) is related to $\mathbf{w}$.

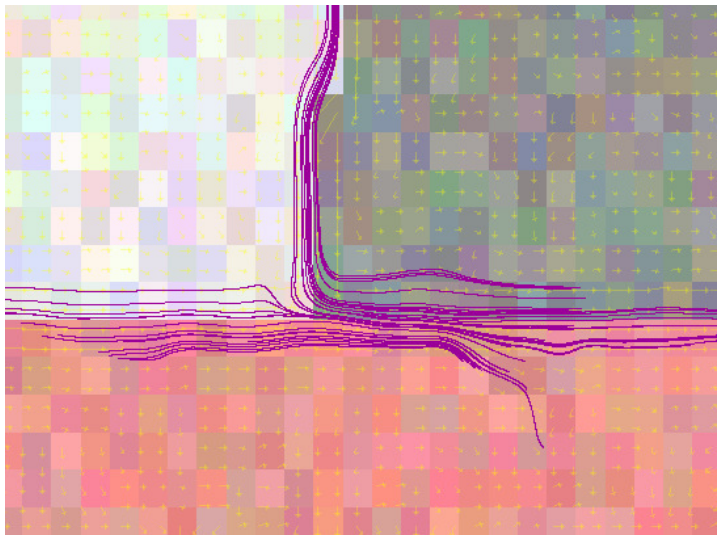
Let $\mathcal{C}_{(a)}^{\mathbf{X}}$ be the curve defining the *integral curve* of $\mathbf{w}$, starting from $\mathbf{X}$ and parametrized by $a\in\mathbb{R}$:

$$\left\{\begin{array}{rcl}\mathcal{C}_{(0)}^{\mathbf{X}} &=& \mathbf{X}\\ \\ \frac{\partial\mathcal{C}_{(a)}^{\mathbf{X}}}{\partial a} &=& \mathbf{w}(\mathcal{C}_{(a)}^{\mathbf{X}})\end{array}\right.\tag{22}$$

When $a \to +\infty$ the integral curve $\mathcal{C}^{\mathbf{X}}_{(a)}$ is tracked *forward*, and *backward* when $a \to -\infty$ (Fig.13). We denote by $\mathcal{F}$ the family of integral curves of $\mathbf{w}$.



(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 13: Integral curve $\mathcal{C}^{\mathbf{X}}$ of a vector field $\mathbf{w} : \Omega \to \mathbb{R}^2$.

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

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

with $h \to 0$, 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 $\mathcal{C}^{\mathbf{X}}$:

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

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 $a \to I_i(\mathcal{C}^{\mathbf{X}}_{(a)})$ at $a = 0$ is then :

$$\begin{aligned} \frac{\partial^2 I_i(\mathcal{C}^{\mathbf{X}}_{(a)})}{\partial a^2}_{|a=0} &= \lim_{h\to 0}\ \frac{1}{h^2}\left[I_i(\mathcal{C}^{\mathbf{X}}_{(h)}) + I_i(\mathcal{C}^{\mathbf{X}}_{(-h)}) - 2I_i(\mathcal{C}^{\mathbf{X}}_{(0)})\right] \\ &= \lim_{h\to 0}\ \frac{1}{h^2}\left[h^2\ \nabla I_i^T\mathbf{J}_{\mathbf{w}_{(\mathbf{X})}}\mathbf{w}_{(\mathbf{X})} + h^2\ \text{trace}\left(\mathbf{w}_{(\mathbf{X})}\mathbf{w}_{(\mathbf{X})}^T\mathbf{H}_{i(\mathbf{X})}\right) + O(h^3)\right] \\ &= \text{trace}\left(\mathbf{w}_{(\mathbf{X})}\mathbf{w}_{(\mathbf{X})}^T\mathbf{H}_{i(\mathbf{X})}\right) + \nabla I_i^T\mathbf{J}_{\mathbf{w}_{(\mathbf{X})}}\mathbf{w}_{(\mathbf{X})} \end{aligned} \tag{23}$$

Note that this is exactly the right term in the proposed curvature-preserving PDE (21).

Actually, (21) 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 (22) with different parametrization : $\forall a \in \mathbb{R},\ \ \mathcal{C}^{\mathbf{Y}}_{(a)} = \mathcal{C}^{\mathbf{X}}_{(\epsilon+a)}$. As (21) 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 (21) is verified for all points $\mathbf{Y}$ lying on the integral curve $\mathcal{C}^{\mathbf{X}}$. Then, the PDE (21) may be also written as :

$$\forall \mathcal{C} \in \mathcal{F},\ \ \forall a \in \mathbb{R}, \qquad \frac{\partial I_i(\mathcal{C}_{(a)})}{\partial t} = \frac{\partial^2 I_i(\mathcal{C}_{(a)})}{\partial a^2} \tag{24}$$

We may recognize in (24) 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, the constrained equation (21) 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 (10),(15). This curvature-preserving property of (21) is illustrated on Fig.11d and Fig.12b.
The constrained equation (21) is an *elliptic* PDE since the matrix $\mathbf{w}\mathbf{w}^T$ is positive definite. The existence and unicity of the solutions of (21) are not directly approached here. Anyway, in 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 [23] 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 under a discrete formula, 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 :

$$\forall \mathbf{X} \in \Omega, \qquad \mathbf{I}^{LIC}_{(\mathbf{X})} = \frac{1}{N} \int_{-\infty}^{+\infty} f(p)\ \mathbf{I}^{noise}(\mathcal{C}^{\mathbf{X}}_{(p)})\ dp \tag{25}$$

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* (22) 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, the curvature-preserving PDE (21) can be seen as the one-dimensional heat flow (24) constrained on the integral curve $\mathcal{C}^{\mathbf{X}} \in \mathcal{F}$. Using the variable substitution $\mathbf{L}_{(a)} = \mathbf{I}(\mathcal{C}^{\mathbf{X}}_{(a)})$, (24) 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 [39, 62]) :

$$\mathbf{L}^{[t]}_{(a)} = \int_{-\infty}^{+\infty} \mathbf{L}^{[t=0]}_{(p)}\ G_{t(a-p)}\ dp \quad \text{ with } \quad G_{t(p)} = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{p^2}{4t}\right) \tag{26}$$

Substituting $\mathbf{L}$ in (5.4) with $a = 0$, and remembering that $\mathcal{C}^{\mathbf{X}}_{(0)} = \mathbf{X}$ and $G_{t(-p)} = G_{t(p)}$:

$$\forall \mathbf{X} \in \Omega, \qquad \mathbf{I}^{[t]}_{(\mathbf{X})} = \int_{-\infty}^{+\infty} \mathbf{I}^{[t=0]}(\mathcal{C}^{\mathbf{X}}_{(p)})\ G_{t(p)}\ dp \tag{27}$$

The equation (27) is a particular form of the continuous LIC-based formulation (25) 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 the curvature-preserving PDE (21) 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-channel 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.13b, 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 4^{th}-order Runge-Kutta scheme.

Note that (27) is an analytical solution of (21) 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 (thus defining a temporal non-linearity). In order to get rid of this kind of non-linearity, we will then to perform several successive iterations of the LIC scheme (27), where the vector field $\mathbf{w}$ is updated at each iteration. This is actually a good way of approximating (21). 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, the curvature-preserving equation (21) will be efficiently discretized by several iterations of the LIC formulation (27). This will be detailed in section 4.

3.3 Between Traces and Divergences

We illustrate here how the curvature-preserving PDE (21) may be regarded compared to trace and divergence expressions (10), (15), for the case of single direction smoothing $\mathbf{T} = \mathbf{w}\mathbf{w}^T$.
In this case, the divergence PDE (10) may be developed as :

$$\begin{aligned}
\text{div}\left(\mathbf{w}\mathbf{w}^T\ \nabla I_i\right) &= \text{div}\begin{pmatrix} u^2\ \frac{\partial I_i}{\partial x} + uv\ \frac{\partial I_i}{\partial y} \\ \\ uv\ \frac{\partial I_i}{\partial x} + v^2\ \frac{\partial I_i}{\partial y} \end{pmatrix} \\
&= \left(u^2\ \frac{\partial^2 I_i}{\partial x^2} + 2uv\ \frac{\partial^2 I_i}{\partial x \partial y} + v^2\ \frac{\partial^2 I_i}{\partial y^2}\right) + \nabla I_i^T \begin{pmatrix} 2u\ \frac{\partial u}{\partial x} + u\ \frac{\partial v}{\partial y} + v\ \frac{\partial u}{\partial y} \\ \\ 2v\ \frac{\partial v}{\partial y} + u\ \frac{\partial v}{\partial x} + v\ \frac{\partial u}{\partial x} \end{pmatrix} \\
&= \text{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T \left[\begin{pmatrix} u\ \frac{\partial u}{\partial x} + v\ \frac{\partial u}{\partial y} \\ \\ u\ \frac{\partial v}{\partial x} + v\ \frac{\partial v}{\partial y} \end{pmatrix} + \begin{pmatrix} u\ \frac{\partial u}{\partial x} + u\ \frac{\partial v}{\partial y} \\ \\ v\ \frac{\partial u}{\partial x} + v\ \frac{\partial v}{\partial y} \end{pmatrix}\right] \\
&= \text{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T \mathbf{J}_{\mathbf{w}}\mathbf{w} + \text{div}(\mathbf{w})\nabla I_i^T\mathbf{w}
\end{aligned}$$

Thus, we recognize in these three different terms :

- The first term corresponds to the trace PDE (15), that smoothes locally $\mathbf{I}$ along $\mathbf{w}$, using oriented Gaussian kernels.
- The two first terms correspond to the *curvature-constrained* regularization PDE (21), 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 (10) that performs local diffusions of $\mathbf{I}$ along $\mathbf{w}$. This last term $\text{div}(\mathbf{w})\nabla I_i^T\mathbf{w}$ 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 the curvature-constrained PDE (21) is then "mathematically" positioned between the trace (15) and divergence formulations (10), 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 the curvature-preserving PDE (21) as a divergence-based PDE minus a constraint term :

$$\text{trace}\left(\mathbf{w}\mathbf{w}^T\mathbf{H}_i\right) + \nabla I_i^T\mathbf{J}_{\mathbf{w}}\mathbf{w} = \text{div}\left(\mathbf{w}\mathbf{w}^T\ \nabla I_i\right) - \text{div}(\mathbf{w})\nabla I_i^T\mathbf{w}$$

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 $\nabla I^T\mathbf{J}_{\mathbf{w}}\mathbf{w} = -I_{\mathbf{ww}}$, vanishing then the velocity of the curvature-preserving evolution equation (21), 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 should not modify the image. Note by comparison that the velocity of the corresponding divergence-based expression $\text{div}\left(\mathbf{w}\mathbf{w}^T\ \nabla I_i\right)$ also vanishes here.

- When $\mathbf{w} = \frac{\nabla I}{\|\nabla I\|}$ (gradient direction), then $\nabla I^T \mathbf{J}_{\mathbf{w}} \mathbf{w} = 0$, and the velocity of the curvature-preserving PDE (21) becomes simply $I_{\mathbf{ww}}$, which really corresponds to a smoothing of the image along the gradient direction (the same as the unconstrained trace-based PDE (15)). 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 $\text{div}(\mathbf{w}) = 0$), the divergence-based PDE (10) and the curvature-preserving formulation (21) are strictly equivalent. This is very rarely the case anyway.

3.4 Extension to multi-directional smoothing

In [102], the single-direction smoothing PDE (21) has been extended so that it can deal with a tensor-valued geometry $\mathbf{T} : \Omega \to \mathrm{P}(2)$, instead of a vector-valued geometry $\mathbf{w}$. This is important, since 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 the curvature-preserving PDE (21) 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 proposed to locally decompose a tensor-driven smoothing process into several vector-driven smoothing processes along different orientations. We first notice that

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

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 :

$$\begin{aligned} \mathbf{T} &= \frac{2}{\pi} \int_{\alpha=0}^{\pi} \sqrt{\mathbf{T}} a_\alpha a_\alpha^T \sqrt{\mathbf{T}}^T \, d\alpha \\ &= \frac{2}{\pi} \int_{\alpha=0}^{\pi} (\sqrt{\mathbf{T}} a_\alpha)(\sqrt{\mathbf{T}} a_\alpha)^T \, d\alpha \end{aligned} \tag{28}$$

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 (28) 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 (28) and considering that each single direction smoothing must be done with a curvature-preserving approach (21), we end up with the following curvature-constrained regularization PDE, acting on a multi-channel image $\mathbf{I} : \Omega \to \mathbb{R}^n$ and driven by a tensor-valued smoothing geometry $\mathbf{T}$:

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

which can be simplified as :

$$\forall i = 1, \ldots, n, \qquad \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) + \frac{2}{\pi} \nabla I_i^T \int_{\alpha=0}^{\pi} \mathbf{J}_{\sqrt{\mathbf{T}} a_\alpha} \sqrt{\mathbf{T}} a_\alpha \, d\alpha \tag{29}$$

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 [113]. As in the single direction smoothing case, (29) may be seen as a trace-based equation (15), 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 the regularization method (29), one can benefit from the LIC-based interpretation of curvature-preserving PDE's presented in section 3.2. Indeed, we can explicitly discretize (29) 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 (21) 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-pondered 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 [83] integration scheme. Faster LIC implementations have been proposed in [95] but do not deal with Gaussian pondering functions, as needed here.
This simple observation leads then to the following fast algorithm for the implementation of one iteration of the curvature-preserving PDE (29) :

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

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

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

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

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

- 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.

- Average all LIC's computed in step 4.

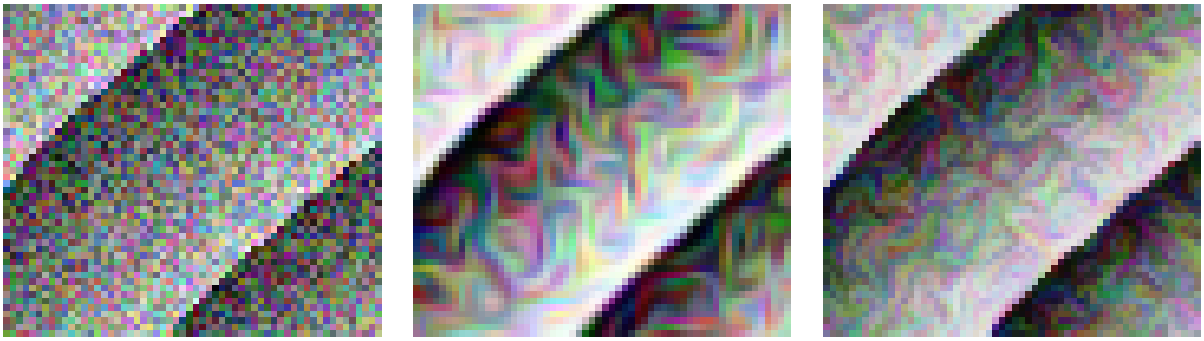
The main parameters of the algorithm are p_1, p_2, σ, dt and the number of PDE iterations nb that are applied. The characteristics of this scheme, compared to the classical finite-difference one is :

- 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 4^{th} Runge-Kutta interpolation [83] is used to track the integral curves $\mathcal{C}$ in the image.

- It allows to choose very large time steps dt, since the scheme we proposed is unconditionally stable. Indeed, dt is simply proportional to the overall smoothing variance of the Gaussian-pondering convolutions done 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 only $nb = 1$ iteration with very large time steps dt. In fact, this leads to a rough approximation of (29), since we lost the temporal non-linearity property of the PDE. But for images with few noise, this gave surprisingly good results. Actually, the spatial non-linearity seems to play a more important role than the temporal non-linearity in the PDE evolution.

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.14, we illustrate the efficiency of the scheme, compared to the classical finite-difference one. A synthetic noisy image is anisotropically smoothed with the PDE (29), with $p_1 = 0.01$ and $p_2 = 100$ (smoothing mostly along isophotes θ_-, with a strength of 1). The LIC-based scheme (Fig.14c) clearly 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 the LIC-based scheme (stopped at $t = 100$).

Figure 14: Comparisons between classical explicit PDE schemes, and LIC-based implementation of the PDE (29).

5 Applications

We illustrate the wide variety of problems that can be tackled by the regularization PDE's proposed throughout this chapter. Particularly, we use the curvature-preserving regularization approach described in section 3 in the following examples, dealing with color images. We show results of color image denoising, inpainting and resizing by nonlinear interpolation. Given processing time have been obtained on a 2.8Ghz i686 Intel processor.

5.1 Color image denoising and artifacts removal

Image denoising is of course a direct application of regularization methods. Sensor inaccuracies, digital quantifications or compression artifacts are indeed some of the various noise sources that can affect a digital image, and suppressing them is a desirable goal. It generally leads to small random variations that affect pixels of the image and that must be removed.

In Fig.15-19, we illustrate how the curvature-preserving PDE framework (29) can be successfully applied to remove such artifacts while preserving the essential structures of the processed images.

- Fig.15 shows an application of the regularization method on the famous 512×512 "Baboon" color image, artificially degraded by adding uncorrelated Gaussian noise on (R, G, B). This color image has been then denoised with equation (29). Thanks to the proposed LIC-based numerical implementation, only one PDE iteration has been necessary to denoise the image, with parameters $p_1 = 0.5$, $p_2 = 0.7$, $\sigma = 1.5$ and $dt = 50$. Processing time is 19.3 seconds for the entire image.

- Fig.16 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. Using the PDE-based regularization method (29) allows to clearly remove the grains 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 (size 586×367).

- Fig.17 shows the restoration of a digital photograph shot under low luminosity conditions by a cellular phone. Such devices have usually poor quality cameras, leading to the apparition of important digital noise (more precisely, Poisson noise) on the acquired color images. Processed color image has a size of 262×280 and has been restored in 4 seconds (one PDE iteration), with parameters $p_1 = 0.2$, $p_2 = 0.5$, $\sigma = 2$, $dt = 120$. Note how the curvature-preserving PDE (29) is able to adapt itself locally to the multi-channel image geometry, in order to preserve thin structures while removing the noise quite well.

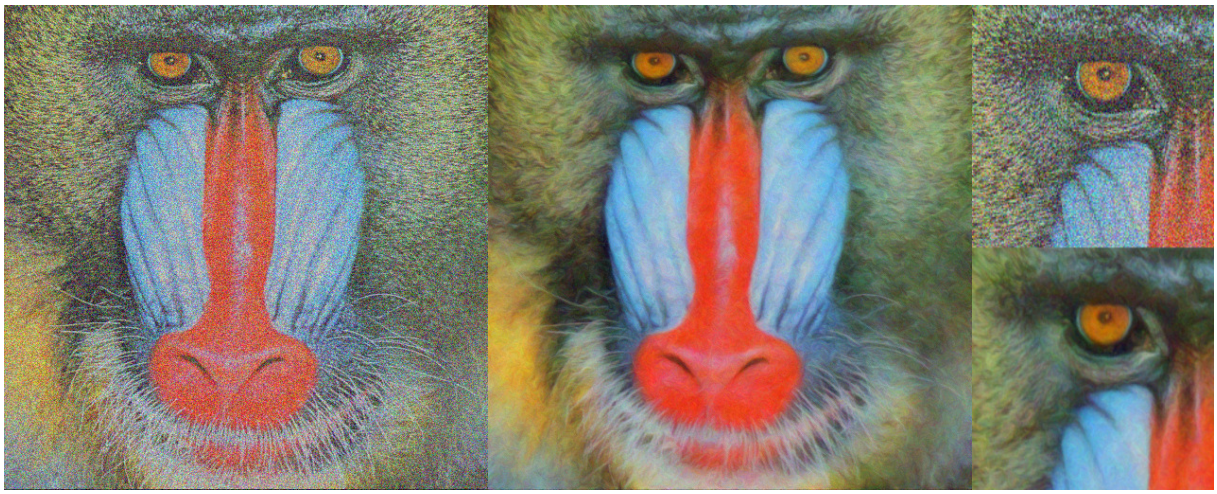
- Image regularization can also be useful when dealing with other types of noise. The enhancement of JPEG-compressed image is such an example of interest. Fig.18 illustrate the suppression of compression artifacts in color images. A JPEG compressed color image of size 283×249, where the JPEG quality ratio has been set to 10% is processed by the multi-channel image regularization algorithm. Usual block effects inherent to the DCT compression are visible on the compressed image. 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). Processing time is 5.4 seconds for the entire image.

- Fig.19 illustrates another regularization with a different type of noise. This time, the regularization method is used to improve a digital true-color image quantified in 256 colors by the Floyd-Steinberg algorithm (size=355×287) which introduces some dithering effects in the image. One PDE iteration has been applied here, 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. This kind of reconstruction may be interesting for image compression algorithms, allowing them to retrieve a true color images, even by storing them with a quantified palette.

5.2 Color image inpainting

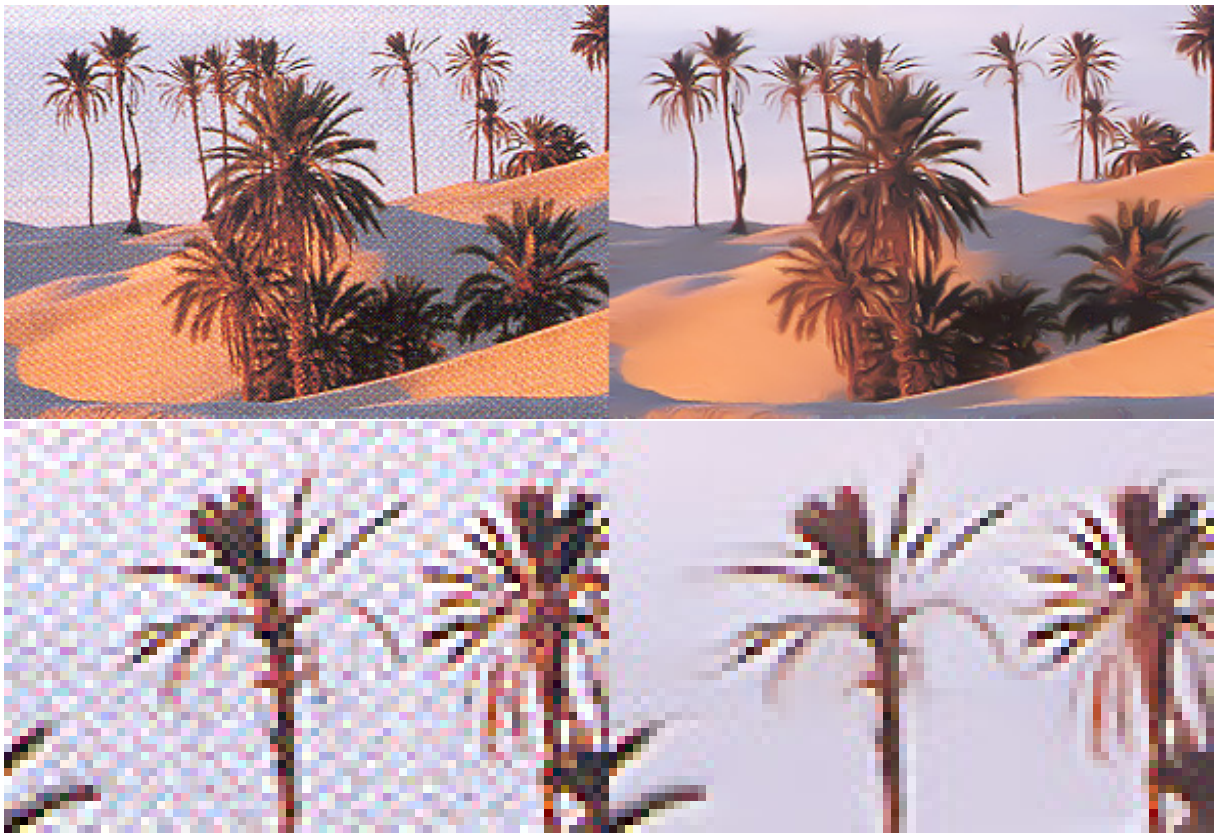
Image inpainting is a very new and challenging application, which consists in filling-in missing image regions (defined by the user) by *guessing pixel values* such that the reconstructed image still looks natural. Basically, the user provides one color image $\mathbf{I} : \Omega \to \mathbb{R}^3$, and 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) that are usually bigger than other image artifacts. Image inpainting has been first proposed as a method based on a variational formulation by Masnou and Morel [70], followed by many solutions based on diffusion or transport PDE's [15, 17, 30, 109, 110]. It is also worth to cite some papers related to inpainting without use of PDE's [38, 55], among others.
In this article, we see the inpainting process as a direct application of the proposed curvature-preserving PDE (29). We apply the diffusion equation only on the regions to inpaint, allowing the neighbor pixels of these regions to diffuse inside : a nonlinear completion of the image data along isophotes directions is thus naturally done, reconstructing the missing parts of the image, since the performed smoothing tries to follow a coherent multi-valued image geometry computed from the known parts of the image. We illustrate the concept of inpainting with PDE's in Fig.20-24.

- Fig.20 illustrates a simple case where one wants to remove text from a color image, by guessing the pixel colors behind the text. The mask used here is easily detected from the degraded image by considering only green pixels. Result of



Noisy color image (left), denoised image (middle), zoom on the eye (right).

Figure 15: Denoising of the color image “Baboon” corrupted with artificial Gaussian noise.



Noisy color image (left), denoised image (right), details are shown on the bottom row.

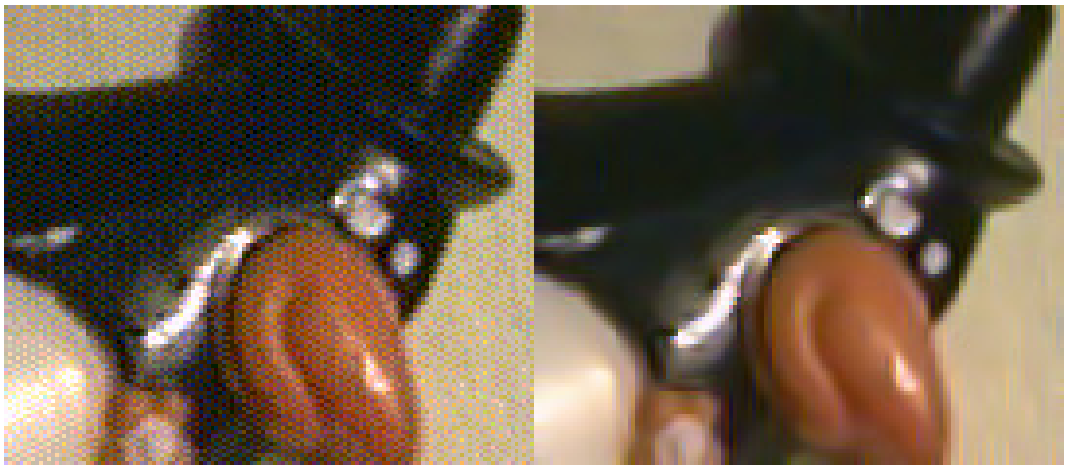
Figure 16: Denoising of the color image “Tunisia Desert”, containing watered effects.

Noisy color image (left), denoised image (right), details are shown on the bottom row.

Figure 17: Denoising of the color image “Lolotte”, real color photograph.

JPEG color image (left), regularized image (right), details are shown on the bottom row.

Figure 18: Removing blocs artifacts from the JPEG compressed color image “Baby”.



Quantized color image (left), regularized image (right) (details).

Figure 19: Removing quantization noise in the quantized color image “Penguin”.

the image inpainting is shown on the right. Note how structures have been naturally completed in a coherent way. This example also shows the limitation of the reconstruction when it comes to reconstruct textured regions. For instance, the algorithm has not been able to reconstruct one eye of the woman on the left. This is anyway a very hard task, and it is not surprising that diffusion equations that perform only local smoothing fail in such complex cases.

- Fig.21 shows how the PDE-based inpainting technique can be also used to remove real objects from digital photographs. A 500×500 color image (left) is inpainted with a user-defined mask (middle), corresponding to the region overlapping the initial man's glasses. The inpainted image (right) is obtained in 4 minutes 11 seconds, after 200 iterations of the PDE (29) 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. Another similar example of real object removal can be found in Fig.24, where a cage automatically disappear in a color photograph of a parrot.
- Fig.22-23 illustrate the reconstruction capabilities of our inpainting technique. Here, half of the pixels of two color images have been suppressed by masking them with checkerboard-shaped mask with squares of respective sizes 16×16 and 32×32. The images are then reconstructed using several iterations of the PDE (29) applied only inside the inpainting masks, with parameters $p_1 = 0.001$, $p_2 = 100$, $\sigma = 4$ and $dt = 50$ This kind of application is very interesting for image transmission within a network. It can be used to generate coherent reconstructed images even if corrupted network packets have been received.

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 (29) 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).

5.3 Color image interpolation

With the same kind of techniques, one can easily perform image magnification by edge-preserving interpolation. Starting from a linear or bicubic interpolation of a small image, and applying the PDE (29) on the image (excepted on the original known pixels that form a sparse inpainting mask), we can compute magnified images that have been regularized while taking their local geometry into account. It allows to remove usual bloc or jagging effects inherent to classical linear interpolation techniques. This is a technique which is actually very similar to image inpainting with a very sparse grid for the inpainting mask.

- Fig.25 illustrates one example of image resizing. An original 195×173 color image is resized by a factor $\times 4$ with classical nearest-neighbor, linear and bicubic interpolations, then by the PDE-based technique (29). It is quite clear that the non-linear regularization filter, driven by the image geometry, allows to remove the aliasing effects usually encountered with simple interpolation methods, while correctly preserving the small structures of the image.

Notice that the original known points of the images are always preserved during the regularizing PDE flow, ensuring that resizing back the image to its original dimension (sub-sampling) always results in the original input data.

5.4 Flow visualization

We present here a last application of regularization PDE's for visualization purposes. Considering a 2D vector field $\mathcal{F} : \Omega \to \mathbb{R}^2$, we have several ways to visualize it. We can first use vector graphics (Fig.26) (left), but we need to subsample the field since this kind of representation is not adapted to represent very dense flows. A better solution is as follows. We smooth a completely noisy (color) image $\mathbf{I}$, with a regularizing flow equivalent to (29) but where $\mathbf{T}$ is directed by the directions of $\mathcal{F}$, instead of the local geometry of $\mathbf{I}$. It means that we have to define the tensor field $\mathbf{D}$ in (29) as :

$$\forall \mathbf{X} \in \Omega, \qquad \mathbf{D}_{(\mathbf{X})} = \frac{1}{\|\mathcal{F}\|} \mathcal{F}\mathcal{F}^T \tag{30}$$

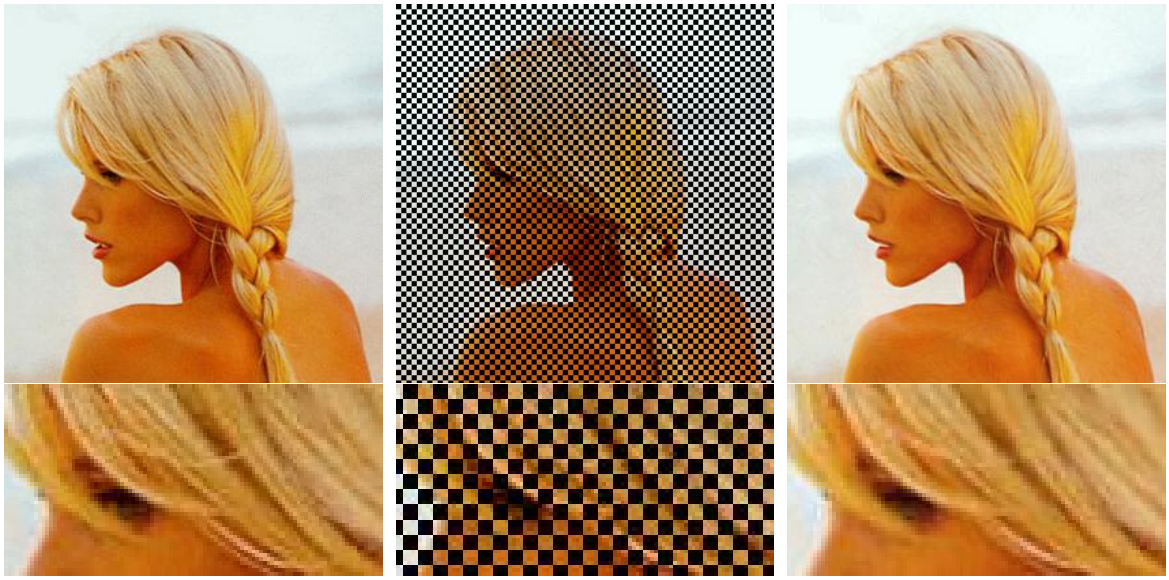
This is a field of fully anisotropic tensors, each time oriented along the flow $\mathcal{F}$.
This technique is in fact equivalent to using a single LIC filter [23]. But it is interesting to see that when the PDE evolution time t goes by, a visualization *scale-space* of $\mathcal{F}$ is explicitly constructed (Fig.27). Here, the used regularization equation (30) *ensures that the smoothing of the pixels is done exactly in the direction of the flow* $\mathcal{F}$. This is not the case in [13, 22, 40], where the authors proposed a similar idea using a divergence-based expression. Using similar divergence-based techniques would raise a risk of smoothing the image in false directions, as this has been pointed out in section 2.2.

Corrupted color image (left), inpainted image (right). Details are shown on the bottom row.

Figure 20: Removing undesirable text in color image by an inpainting method.

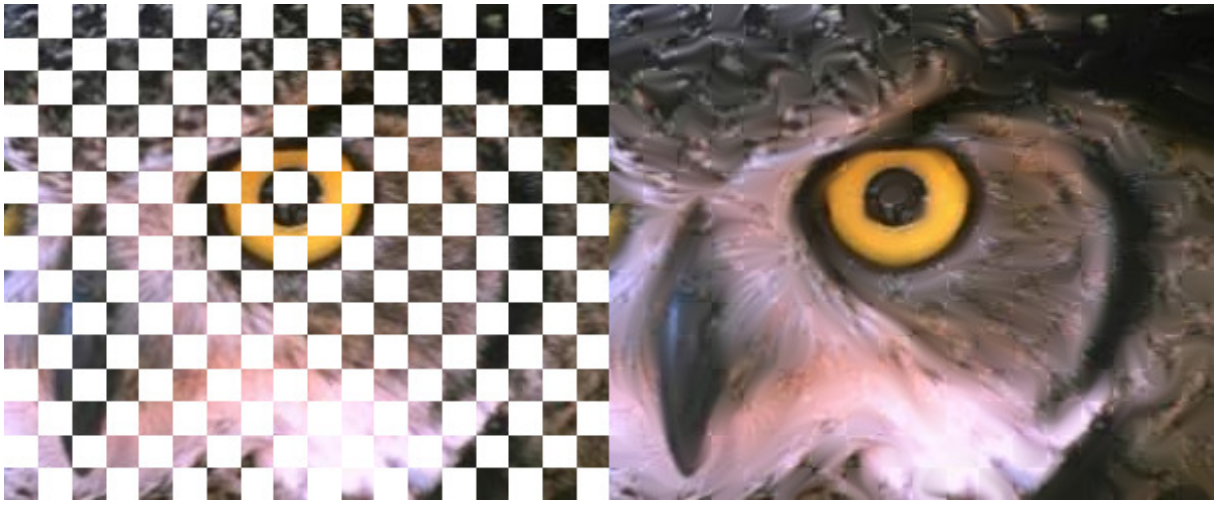
Original color image (left), inpainting mask (middle), inpainted image (right).

Figure 21: Removing a real object in a color image by an inpainting method.



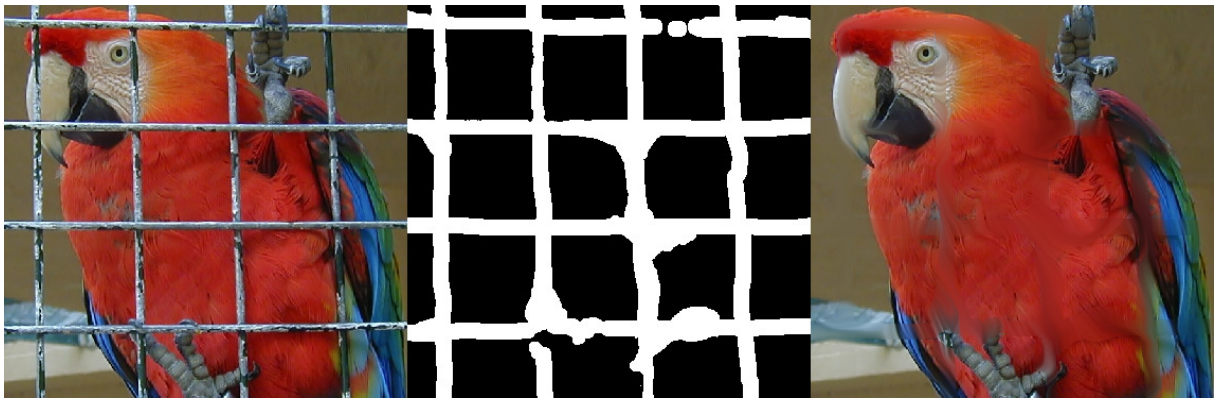
Original color image (left), masked color image (middle), reconstruction (right).

Figure 22: Reconstruction of 50% of a color image by an inpainting method.



Masked color image (left), reconstruction (right).

Figure 23: Reconstruction of 50% of a color image by an inpainting method.



Original color image (left), inpainting mask (middle), inpainted image (right).

Figure 24: Removing a real object in a color image by an inpainting method.

Conclusion

Multi-channel image regularization is a fundamental process in many image processing and computer vision algorithms and it is then primordial to get a full control of this process, as well as to understand what the used equations are exactly doing from a geometric point of view. In this chapter, we have described the most classical PDE-based methods proposed for the regularization of multi-channel images and introduced a very efficient curvature-preserving framework that generally outperforms its competitors. This is not only due to the particular aim of preserving fine and curved structures, but also thanks to the proposed numerical scheme that is especially efficient since it works at a subpixel level. Clearly, this kind of multi-channel image regularization technique can play a role in a lot of image processing applications. The processing time, which was one of the famous drawback of PDE-based methods, is not a problem anymore. All these reasons makes the framework of multi-valued diffusion PDE's a very good choice for image regularization purposes. This has been illustrated in this chapter with results on color image denoising, inpainting and resizing. But many other applications may benefit from the proposed curvature-preserving framework.

It is worth to notice that other application results of the curvature-preserving algorithm can be found at the following web page :

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

The binaries of the algorithm can be also downloaded and tested on different architectures, as well as the source code (C++) which are available as a part of the open source image processing library : *The CImg Library* [111].

(a) Thumbnail image.

(b) Details from the image resized by nearest-neighbor interpolation.

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

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

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

Figure 25: Comparisons of image resizing methods, Nearest-neighbor (first row), Linear (second row), Bicubic (third row) and PDE-based (last row) interpolations.

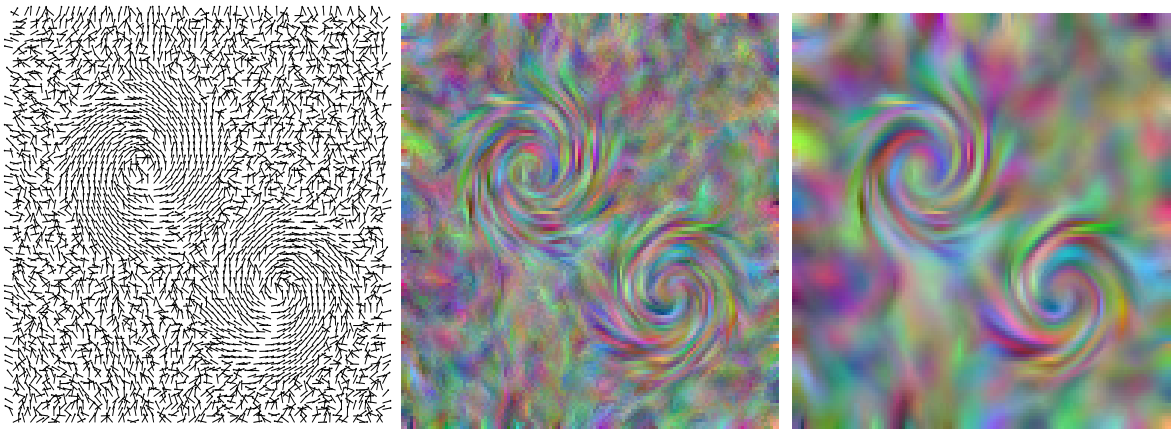


Figure 26: Visualization of a $2D$ vector field : using arrows (left), after 5 PDE iter. (middle), after 15 PDE iter. (right).

Appendix A

In this appendix, we demonstrate how the minimization of the functional

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} E(\mathbf{I}) = \int_\Omega \psi(\lambda_+,\lambda_-)\, d\Omega \tag{31}$$

can be performed by a gradient descent and its corresponding PDE flow. The Euler-Lagrange equations corresponding to the functional (31) are :

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\begin{pmatrix} \dfrac{\partial \psi}{\partial I_{i_x}} \\ \\ \dfrac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} \qquad (i = 1..n) \tag{32}$$

Actually, the vector $(\frac{\partial \psi}{\partial I_{i_x}}, \frac{\partial \psi}{\partial I_{i_y}})^T$ can be written in a more comprehensive form.
From the chain-rule property of the derivation, we have :

$$\begin{pmatrix} \dfrac{\partial \psi}{\partial I_{i_x}} \\ \\ \dfrac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \lambda_+}{\partial I_{i_x}} & \dfrac{\partial \lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_+}{\partial I_{i_y}} & \dfrac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} \begin{pmatrix} \dfrac{\partial \psi}{\partial \lambda_+} \\ \\ \dfrac{\partial \psi}{\partial \lambda_-} \end{pmatrix} \tag{33}$$

We know formally the expressions $\frac{\partial \psi}{\partial \lambda_\pm}$ since the function ψ is directly defined from the $\lambda_\pm$.
Finding the $\frac{\partial \lambda_\pm}{\partial I_{i_x}}$ and $\frac{\partial \lambda_\pm}{\partial I_{i_y}}$ is more tricky. Here is a simple way to proceed :

As the $\lambda_\pm$ are the eigenvalues of the structure tensor $\mathbf{G} = (g_{kl})$, we may decompose its derivatives (with respect to I_{i_x} and I_{i_y}), in terms of derivatives with respect to the g_{kl} :

$$\frac{\partial \lambda_\pm}{\partial I_{i_x}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_x}} \qquad \text{and} \qquad \frac{\partial \lambda_\pm}{\partial I_{i_y}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_y}} \tag{34}$$

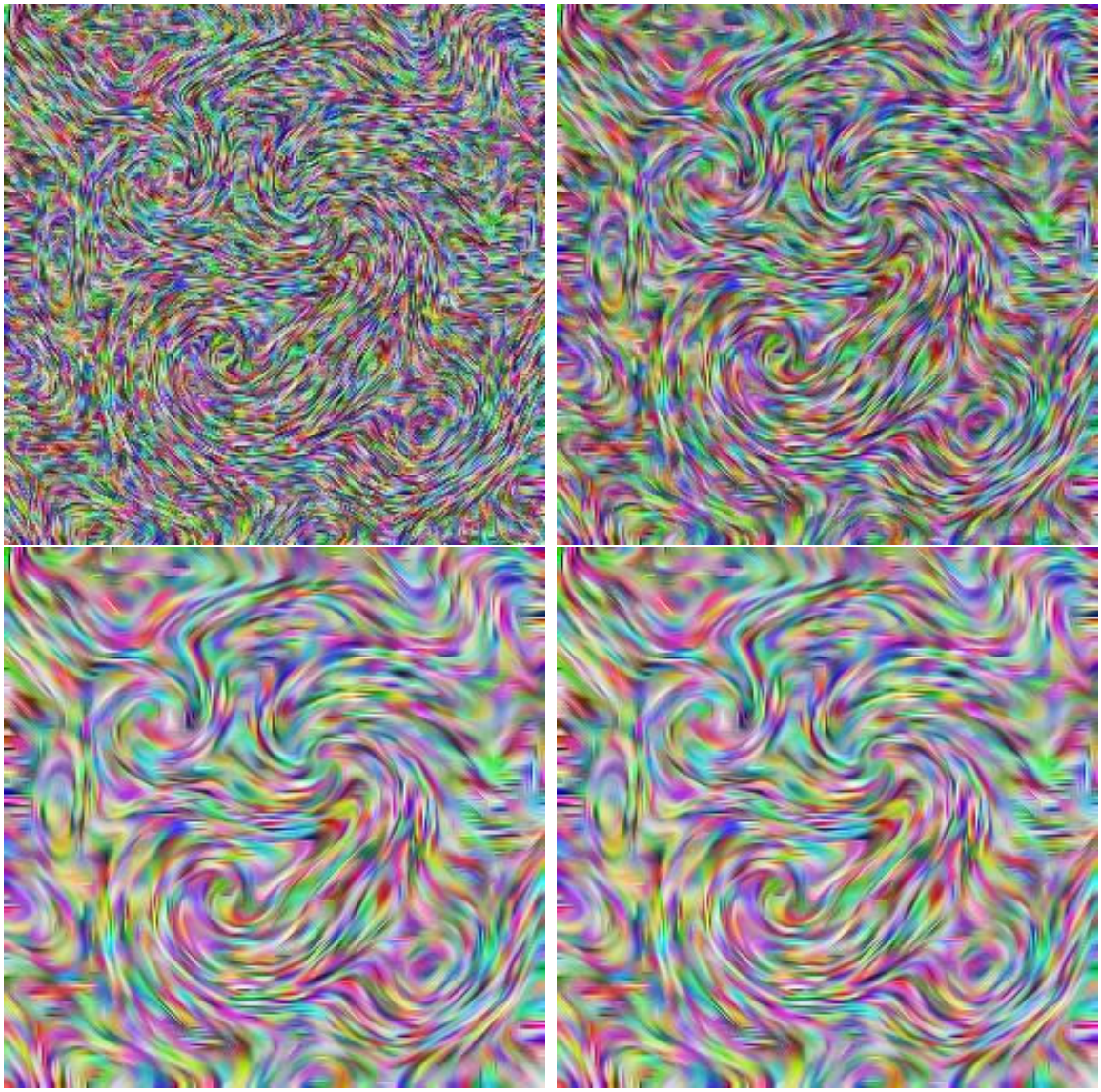


Figure 27: Visualization scale-space generated with regularization PDE's.

The expressions $\frac{\partial g_{kl}}{\partial I_{i_x}}$ and $\frac{\partial g_{kl}}{\partial I_{i_y}}$ are particularly simple :

$$\left\{ \begin{array}{l} \dfrac{\partial g_{11}}{\partial I_{i_x}} = 2I_{i_x} \\ \\ \dfrac{\partial g_{11}}{\partial I_{i_y}} = 0 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \dfrac{\partial g_{12}}{\partial I_{i_x}} = I_{i_y} \\ \\ \dfrac{\partial g_{12}}{\partial I_{i_y}} = I_{i_x} \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \dfrac{\partial g_{22}}{\partial I_{i_x}} = 0 \\ \\ \dfrac{\partial g_{22}}{\partial I_{i_y}} = 2I_{i_y} \end{array} \right.$$

i.e (34) can be written as :

$$\left(\begin{array}{c} \dfrac{\partial \lambda_\pm}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial I_{i_y}} \end{array} \right) = \left(\begin{array}{cc} 2\dfrac{\partial \lambda_\pm}{\partial g_{11}} & \dfrac{\partial \lambda_\pm}{\partial g_{12}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial g_{12}} & 2\dfrac{\partial \lambda_\pm}{\partial g_{22}} \end{array} \right) \nabla I_i \tag{35}$$

Thus, one last obstacle remains to be crossed, that is finding the formal expressions of $\frac{\partial \lambda_\pm}{\partial g_{kl}}$.
Remind that the $\lambda_\pm$ and $\theta_\pm$ are the eigenvalues and eigenvectors of the structure tensor $\mathbf{G}$:

$$\mathbf{G} = \lambda_+ \ \theta_+ \theta_+^T + \lambda_- \ \theta_- \theta_-^T$$

The derivation of this tensor, with respect to one of its coefficient g_{kl} is :

$$\begin{array}{rcl} \dfrac{\partial \mathbf{G}}{\partial g_{kl}} & = & \dfrac{\partial \lambda_+}{\partial g_{kl}} \ \theta_+ \theta_+^T + \dfrac{\partial \lambda_-}{\partial g_{kl}} \ \theta_- \theta_-^T \\ & + & \lambda_+ \ \dfrac{\partial \theta_+}{\partial g_{kl}} \theta_+^T + \lambda_- \ \dfrac{\partial \theta_-}{\partial g_{kl}} \theta_-^T \\ & + & \lambda_+ \ \theta_+ \dfrac{\partial \theta_+^T}{\partial g_{kl}} + \lambda_- \ \theta_- \dfrac{\partial \theta_-^T}{\partial g_{kl}} \end{array} \tag{36}$$

Moreover, as the $\theta_\pm$ are unitary and orthogonal eigenvectors, we have :

$$\left\{ \begin{array}{l} \theta_+^T \theta_+ = \theta_-^T \theta_- = 1 \\ \\ \theta_+^T \theta_- = \theta_-^T \theta_+ = 0 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \dfrac{\partial \theta_+^T}{\partial g_{kl}} \theta_+ = \theta_+^T \dfrac{\partial \theta_+}{\partial g_{kl}} = 0 \\ \\ \dfrac{\partial \theta_-^T}{\partial g_{kl}} \theta_- = \theta_-^T \dfrac{\partial \theta_-}{\partial g_{kl}} = 0 \end{array} \right. \tag{37}$$

We first multiply the equation (36) by $\theta_\pm^T$ at the left, by $\theta_\pm$ at the right, then use the properties (37). It allows high simplifications, and leads to these two relations :

$$\frac{\partial \lambda_+}{\partial g_{kl}} = \theta_+^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_+ \qquad \text{and} \qquad \frac{\partial \lambda_-}{\partial g_{kl}} = \theta_-^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_- \tag{38}$$

Equations (38) formally tell us how eigenvalues of a diffusion tensor $\mathbf{G}$ vary with respect to a particular coefficient g_{kl} of $\mathbf{G}$. Actually, this interesting property can be proved for any symmetric matrix. For instance, authors of [77] proposed a similar demonstration in a purely matrix form, leading to the same result. They used it to deal with general covariance matrices.

Moreover in our case, the matrices $\frac{\partial \mathbf{G}}{\partial g_{kl}}$ are very simple to write :

$$\frac{\partial \mathbf{G}}{\partial g_{11}} = \left(\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right) \ , \quad \frac{\partial \mathbf{G}}{\partial g_{12}} = \left(\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right) \quad \text{and} \quad \frac{\partial \mathbf{G}}{\partial g_{22}} = \left(\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array} \right)$$

With all these elements, we can express (35) as :

$$\left(\begin{array}{c} \dfrac{\partial \lambda_+}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_+}{\partial I_{i_y}} \end{array} \right) = 2 \ \theta_+ \theta_+^T \nabla I_i \qquad \text{and} \qquad \left(\begin{array}{c} \dfrac{\partial \lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_-}{\partial I_{i_y}} \end{array} \right) = 2 \ \theta_- \theta_-^T \nabla I_i \tag{39}$$

Finally, replacing (39) in the Euler-Lagrange equations (33) and (32), gives the vector-valued gradient descent of the functional (8) :

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n}\int_\Omega \psi(\lambda_+,\lambda_-)\;d\Omega \quad\Longrightarrow\quad \frac{\partial I_i}{\partial t}=2\;\mathrm{div}\left(\left[\frac{\partial\psi}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\psi}{\partial\lambda_-}\theta_-\theta_-^T\right]\nabla I_i\right) \tag{40}$$

(for $i=1..n$) □

Note that (40) is a divergence-based equation such that :

$$\frac{\partial I_i}{\partial t}=\mathrm{div}\left(\mathbf{D}\nabla I_i\right)\quad\text{where}\quad \mathbf{D}=2\frac{\partial\psi}{\partial\lambda_+}\;\theta_+\theta_+^T+2\frac{\partial\psi}{\partial\lambda_-}\;\theta_-\theta_-^T$$

$\mathbf{D}\in\mathrm{P}(2)$ is then a 2×2 diffusion tensor, whose eigenvalues are :

$$\lambda_1=2\frac{\partial\psi}{\partial\lambda_+}\quad\text{and}\quad\lambda_2=2\frac{\partial\psi}{\partial\lambda_-}$$

associated to these corresponding orthonormal eigenvectors :

$$\mathbf{u}_1=\theta_+\quad\text{and}\quad\mathbf{u}_2=\theta_-$$

It is also worth to mention that computing this gradient descent is done exactly in the same way, when dealing with image domains Ω defined in higher dimensional spaces ($\Omega\subset\mathbb{R}^p$ where $p>2$) More particularly, the case of 3D volume regularization ($p=3$) can be written as :

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n}\int_\Omega \psi(\lambda_1,\lambda_2,\lambda_3)\;d\Omega \quad\Longrightarrow\quad \frac{\partial I_i}{\partial t}=2\;\mathrm{div}\left(\left[\frac{\partial\psi}{\partial\lambda_1}\theta_1\theta_1^T+\frac{\partial\psi}{\partial\lambda_2}\theta_2\theta_2^T+\frac{\partial\psi}{\partial\lambda_3}\theta_3\theta_3^T\right]\nabla I_i\right)$$

In this case, the $\lambda_{1,2,3}$ are the three eigenvalues of the 3×3 structure tensor $\mathbf{G}$, and $\theta_{1,2,3}$ are the corresponding orthonormal eigenvectors.

Appendix B

In this appendix, we demonstrate that the solution of the generic trace-based PDE :

$$\forall i=1,..,n,\qquad \frac{\partial I_i}{\partial t}=\mathrm{trace}\left(\mathbf{T}\mathbf{H}_i\right)$$

is the convolution of the image $\mathbf{I}$

$$I_{i_{(t)}}=I_{i_{(t=0)}}\;*\;G^{(\mathbf{T},t)}\qquad(i=1..n)$$

by an oriented Gaussian kernel $G^{(\mathbf{T},t)}$ defined as :

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

To demonstrate this, we simply derive the kernel expression $G^{(\mathbf{T},t)}$ in time and in space :

$$\frac{\partial G^{(\mathbf{T},t)}}{\partial t}=-\frac{1}{4\pi t^2}\exp\left(-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)\left(1-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)$$

and

$$\begin{cases}\nabla G^{(\mathbf{T},t)} &=& -\dfrac{1}{8\pi t^2}\exp\left(-\dfrac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)\;\mathbf{T}^{-1}\mathbf{X}\\[2ex]\mathbf{H}_{G^{(\mathbf{T},t)}} &=& -\dfrac{1}{8\pi t^2}\exp\left(-\dfrac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)\mathbf{T}^{-1}\left(\mathbb{I}_{\mathrm{d}}-\dfrac{\mathbf{X}\mathbf{X}^T\mathbf{T}^{-1}}{2t}\right)\end{cases}$$

where $\nabla G^{(\mathbf{T},t)}$ and $\mathbf{H}_{G^{(\mathbf{T},t)}}$ are respectively the gradient and the Hessian of $G^{(\mathbf{T},t)}$.
It means that

$$\begin{aligned}\operatorname{trace}\left(\mathbf{T}\ \mathbf{H}_{G^{(\mathbf{T},t)}}\right) &= -\frac{1}{8\pi t^2}\exp\left(-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)\operatorname{trace}\left(\mathbb{I}_{\mathrm{d}}-\frac{\mathbf{X}\mathbf{X}^T\mathbf{T}^{-1}}{2t}\right)\\ &= -\frac{1}{8\pi t^2}\exp\left(-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{4t}\right)\left(2-\frac{\mathbf{X}^T\mathbf{T}^{-1}\mathbf{X}}{2t}\right)\\ &= \frac{\partial G^{(\mathbf{T},t)}}{\partial t}\end{aligned}$$

And as the convolution is a linear operation, we have

$$\begin{aligned}\frac{\partial (I_{i_0} * G^{(\mathbf{T},t)})}{\partial t} &= I_{i_0} * \frac{\partial G^{(\mathbf{T},t)}}{\partial t}\\ &= I_{i_0} * \operatorname{trace}\left(\mathbf{T}\ \mathbf{H}_{G^{(\mathbf{T},t)}}\right)\\ &= \operatorname{trace}\left(\mathbf{T}\ \mathbf{H}_{I_{i_0} * G^{(\mathbf{T},t)}}\right)\end{aligned}$$

as well as

$$\lim_{t\to 0}\ (I_{i_{(t)}} * G^{(\mathbf{T},t)}) = I_{i_0}$$

which tells us that the initial condition at $t=0$ is coherent both for the PDE and the convolution process, since the Gaussian function $G^{(\mathbf{T},t)}$ is normalized. This statement is thus true for each instant t of the PDE flow.

Appendix C

In this appendix, we develop tensor-driven divergence PDE's into their trace-based counterpart. Most divergence-based regularization PDE's acting on multivalued images have the following form :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\left(\mathbf{D}\nabla I_i\right) \qquad\qquad (i=1..n) \tag{41}$$

where $\mathbf{D}$ is a diffusion tensor based *only on first order* operators. The fact is that $\mathbf{D}$ is often computed from the structure tensor $\mathbf{G}=\sum_{j=1}^{n}\nabla I_j \nabla I_j^T$ and depends mainly on the spatial derivatives I_{i_x} and I_{i_y}. Intuitively, as the divergence $\operatorname{div}() = \frac{\partial}{\partial x}+\frac{\partial}{\partial y}$ is itself a first order derivative operator, we should be able to write (41) only with first and second spatial derivatives I_{i_x}, I_{i_y}, $I_{i_{xx}}$, $I_{i_{xy}}$ and $I_{i_{yy}}$. Thus, it could be expressed with oriented Laplacians in each image channel I_i as well, i.e an expression based on the trace operator $\frac{\partial I_i}{\partial t} = \operatorname{trace}\left(\mathbf{D}\mathbf{H}_i\right)$.

We want to make the link between the two different diffusion tensors $\mathbf{D}$ and $\mathbf{T}$ in the divergence-based and trace-based regularization PDE's, in the case when $\mathbf{D}$ is *not constant* :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\left(\mathbf{D}\nabla I_i\right) \qquad \text{and} \qquad \frac{\partial I_i}{\partial t} = \operatorname{trace}\left(\mathbf{T}\mathbf{H}_i\right)$$

As we noticed in the previous section, these two formulations are almost equivalent, up to an additional term depending on the *variation of the tensor field* $\mathbf{D}$:

$$\operatorname{div}\left(\mathbf{D}\nabla I_i\right) = \operatorname{trace}\left(\mathbf{D}\mathbf{H}_{I_i}\right) + \nabla I_i^T\ \vec{\mathbf{div}}\left(\mathbf{D}\right) \tag{42}$$

where $\vec{\mathbf{div}}()$ is the *matrix divergence*.
A natural idea is then to decompose the additional term $\nabla I_i^T\ \vec{\mathbf{div}}\left(\mathbf{D}\right)$ into *oriented Laplacians*, expressed with additional diffusion tensors $\mathbf{Q}$ in the trace operator.

For this purpose, we will consider that the divergence tensor $\mathbf{D}$ is defined at each point $\mathbf{X}\in\Omega$ by

$$\mathbf{D} = f_1(\lambda_+,\lambda_-)\ \theta_+\theta_+^T + f_2(\lambda_+,\lambda_-)\ \theta_-\theta_-^T \quad \text{with} \quad f_{1/2}:\mathbb{R}^2\to\mathbb{R} \tag{43}$$

It means that $\mathbf{D}$ is only expressed from the eigenvalues $\lambda_\pm$ and the eigenvectors $\theta_\pm$ of the structure tensor $\mathbf{G}$:

$$\mathbf{G} = \lambda_+ \ \theta_+ \theta_+^T + \lambda_- \ \theta_- \theta_-^T$$

This is indeed a very generic hypothesis that is verified by the majority of the proposed vector-valued regularization methods, for instance the one proposed in Appendix A :

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\left(\mathbf{D} \nabla I_i\right) \qquad \text{with} \quad (43) \quad \text{and} \quad \left\{ \begin{array}{rcl} f_1(\lambda_+, \lambda_-) & = & 2\dfrac{\partial \psi}{\partial \lambda_+} \\ \\ f_2(\lambda_+, \lambda_-) & = & 2\dfrac{\partial \psi}{\partial \lambda_-} \end{array} \right.$$

In order to develop the additional diffusion term $\nabla I_i^T \vec{\mathbf{div}}\left(\mathbf{D}\right)$ in the equation (42), we propose to write $\mathbf{D}$ as a linear combination of $\mathbf{G}$ and $\mathbb{I}_\mathrm{d}$:

$$\mathbf{D} = \alpha(\lambda_+, \lambda_-)\mathbf{G} + \beta(\lambda_+, \lambda_-)\mathbb{I}_\mathrm{d} \tag{44}$$

i.e we separate the *isotropic* and *anisotropic* parts of $\mathbf{D}$, with

$$\alpha = \frac{f_1(\lambda_+, \lambda_-) - f_2(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \qquad \text{and} \qquad \beta = \frac{\lambda_+ f_2(\lambda_+, \lambda_-) - \lambda_- f_1(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \tag{45}$$

Indeed, we have

$$\begin{array}{rcl} \alpha\mathbf{G} + \beta\mathbb{I}_\mathrm{d} & = & \dfrac{f_1 - f_2}{\lambda_+ - \lambda_-} \ (\lambda_+ \ \theta_+ \theta_+^T + \lambda_- \ \theta_- \theta_-^T) + \dfrac{\lambda_+ f_2 - \lambda_- f_1}{\lambda_+ - \lambda_-} \ (\theta_+ \theta_+^T + \theta_- \theta_-^T) \\ & = & \dfrac{1}{\lambda_+ - \lambda_-} \left[\theta_+ \theta_+^T \ (\lambda_+ f_1 - \lambda_- f_1) + \theta_- \theta_-^T \ (\lambda_+ f_2 - \lambda_- f_2)\right] \\ & = & f_1 \ \theta_+ \theta_+^T + f_2 \ \theta_- \theta_-^T \\ & = & \mathbf{D} \end{array} \qquad \square$$

Here we assumed that $\lambda_+ \neq \lambda_-$ (i.e the structure tensor $\mathbf{G}$ is anisotropic). Anyway, if $\mathbf{G}$ is isotropic, one generally chooses an *isotropic* diffusion tensor $\mathbf{D}$ too, in the divergence operator of (42), i.e $f_1(\lambda_+, \lambda_-) = f_2(\lambda_+, \lambda_-)$. In this case, we choose $\alpha = 0$ and $\beta = f_1(\lambda_+, \lambda_-)$.

This decomposition is useful to rewrite the matrix divergence $\vec{\mathbf{div}}\left(\mathbf{D}\right)$ into :

$$\vec{\mathbf{div}}\left(\mathbf{D}\right) = \alpha \vec{\mathbf{div}}\left(\mathbf{G}\right) + \mathbf{G}\nabla\alpha + \nabla\beta \tag{46}$$

and the additional term of the equation (42) would be computed as :

$$\begin{array}{rcll} \nabla I^T \vec{\mathbf{div}}\left(\mathbf{D}\right) & = & \mathrm{trace}\left(\vec{\mathbf{div}}\left(\mathbf{D}\right) \nabla I_i^T\right) & \\ & = & \alpha \mathrm{trace}\left(\vec{\mathbf{div}}\left(\mathbf{G}\right) \nabla I_i^T\right) & (47) \\ & + & \mathrm{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) & (48) \\ & + & \mathrm{trace}\left(\nabla\beta\nabla I_i^T\right) & (49) \end{array}$$

In the following, we propose to find formal expressions of (47), (48) and (49).

- First, remember that the structure tensor $\mathbf{G}$ is defined as :

$$\mathbf{G} = \sum_{j=1}^{n} \nabla I_j \nabla {I_j}^T$$

We have then :

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

where ΔI_j and $\mathbf{H}_j$ are respectively the Laplacian and the Hessian of the image component I_j.
Then, we can write the expression 47 as :

$$\alpha \mathrm{trace}\left(\vec{\mathbf{div}}\left(\mathbf{G}\right) \nabla {I_i}^T\right) = \sum_{j=1}^{n} \alpha \mathrm{trace}\left(\mathbf{H}_j \left[\nabla {I_i}^T \nabla I_j \mathbb{I}_{\mathrm{d}} + \nabla I_j \nabla {I_i}^T\right]\right) \tag{50}$$

- We finally have to compute $\nabla\alpha$ and $\nabla\beta$, in the expression (48) and (49). This can be done by the decomposition :

$$\nabla\alpha = \frac{\partial\alpha}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\alpha}{\partial\lambda_-}\nabla\lambda_- \quad \text{and} \quad \nabla\beta = \frac{\partial\beta}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\beta}{\partial\lambda_-}\nabla\lambda_- \tag{51}$$

and as the $\lambda_\pm$, eigenvalues of the structure tensor $\mathbf{G}$, depends on the I_{j_x} and I_{j_y} :

$$\begin{aligned}
\nabla\lambda_\pm &= \begin{pmatrix} \lambda_{\pm_x} \\ \lambda_{\pm_y} \end{pmatrix} \\
&= \sum_{j=1}^{n} \begin{pmatrix} \frac{\partial\lambda_\pm}{\partial I_{j_x}} I_{j_{xx}} + \frac{\partial\lambda_\pm}{\partial I_{j_y}} I_{j_{xy}} \\ \frac{\partial\lambda_+}{\partial I_{j_x}} I_{j_{xy}} + \frac{\partial\lambda_\pm}{\partial I_{j_y}} I_{j_{yy}} \end{pmatrix} \\
&= \sum_{j=1}^{n} \mathbf{H}_{I_j} \begin{pmatrix} \frac{\partial\lambda_\pm}{\partial I_{x_j}} \\ \frac{\partial\lambda_\pm}{\partial I_{y_j}} \end{pmatrix}
\end{aligned}$$

In Appendix A, we derived eigenvalues of a structure tensor $\mathbf{G}$, with respect to the spatial image derivatives. We ended up with the following relation :

$$\begin{pmatrix} \frac{\partial\lambda_\pm}{\partial I_{x_j}} \\ \frac{\partial\lambda_\pm}{\partial I_{y_j}} \end{pmatrix} = 2\theta_\pm\theta_\pm^T\nabla I_j$$

Then,

$$\nabla\lambda_\pm = \sum_{j=1}^{n} 2\mathbf{H}_j\theta_\pm\theta_\pm^T\nabla I_j \tag{52}$$

We can replace (52) into the expressions of (51), in order to find the spatial gradients of α and β :

$$\begin{cases}
\nabla\alpha &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\alpha}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j \\
\nabla\beta &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\beta}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j
\end{cases} \tag{53}$$

Using (53), we finally compute the two missing parts (48) and (49) of the additional term $\nabla I_i^T \vec{\mathbf{div}}\left(\mathbf{D}\right)$:

$$\left\{\begin{array}{rcl} \mathrm{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) & = & \displaystyle\sum_{j=1}^{n}\mathrm{trace}\left(2\,\mathbf{G}\mathbf{H}_j\left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\alpha}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \\ \mathrm{trace}\left(\nabla\beta\nabla I_i^T\right) & = & \displaystyle\sum_{j=1}^{n}\mathrm{trace}\left(2\,\mathbf{H}_j\left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\beta}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \end{array}\right. \tag{54}$$

• The final step consists in putting together the equations (50) and (54), in order to express the additional term $\nabla I_i^T\vec{\mathbf{div}}\left(\mathbf{D}\right)$ in the PDE (42).

$$\nabla I_i^T\vec{\mathbf{div}}\left(\mathbf{D}\right)=\sum_{j=1}^{n}\mathrm{trace}\left(\mathbf{H}_j\mathbf{P}^{ij}\right) \tag{55}$$

where the $\mathbf{P}^{ij}$ are the following 2×2 matrices :

$$\begin{array}{rcl} \mathbf{P}^{ij} & = & \alpha\,\nabla I_i^T\nabla I_j\mathbb{I}_{\mathrm{d}} \\ & + & 2\left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\alpha}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\mathbf{G} \\ & + & 2\left((\alpha+\frac{\partial\beta}{\partial\lambda_+})\theta_+\theta_+^T+(\alpha+\frac{\partial\beta}{\partial\lambda_-})\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T \end{array} \tag{56}$$

Note that the indices i,j in the notation $\mathbf{P}^{ij}$ *do not designate* the coefficients of a matrix $\mathbf{P}$, but the parameters of the family consisting of n^2 matrices $\mathbf{P}^{ij}$ (each of them is a 2×2 matrix).
The matrices $\mathbf{P}^{ii}$ are symmetric, but generally not the $\mathbf{P}^{ij}$ (where $i\neq j$), since the gradients ∇I_i and ∇I_j are not aligned in the general case.
Yet, we want to express the equation (55) only with symmetric matrices, in order to interpret it as a sum of local smoothing processes oriented by *diffusion tensors*. Fortunately, the trace operator has this simple property :

$$\mathrm{trace}\left(\mathbf{A}\mathbf{H}\right)=\mathrm{trace}\left(\frac{\mathbf{A}+\mathbf{A}^T}{2}\mathbf{H}\right)$$

where $(\mathbf{A}+\mathbf{A}^T)/2$ is a 2×2 *symmetric* matrix (the symmetric part of $\mathbf{A}$).

Thus, we define the symmetric matrices $\mathbf{Q}^{ij}$, corresponding to the symmetric parts of the $\mathbf{P}^{ij}$:

$$\mathbf{Q}^{ij} \quad = \quad \frac{\mathbf{P}^{ij}+\mathbf{P}^{ij^T}}{2} \tag{57}$$

and we have :

$$\nabla I_i^T\vec{\mathbf{div}}\left(\mathbf{D}\right)=\sum_{j=1}^{n}\mathrm{trace}\left(\mathbf{H}_j\mathbf{Q}^{ij}\right)$$

Finally, the divergence-based PDE (42) can be written as :

$$\mathrm{div}\left(\mathbf{D}\nabla I_i\right)=\sum_{j=1}^{n}\mathrm{trace}\left((\delta_{ij}\mathbf{D}+\mathbf{Q}^{ij})\mathbf{H}_j\right) \tag{58}$$

where δ_{ij} is the Kronecker's symbol :

$$\delta_{ij}=\left\{\begin{array}{ll} 0 & \text{if } i\neq j \\ 1 & \text{if } i=j \end{array}\right.$$

This makes the link between divergence PDE's and sums of atomic trace-based PDE's. A direct geometric interpretation of (58) is not direct anyway.

References

[1] R. Abraham, J.E Marsden, and T.S. Ratiu. *Manifolds, Tensor Analysis, and Applications.* Springer-Verlag, New York, 1991.

[2] J.F. Abramatic and L.M. Silverman. *Non linear restoration of noisy images.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 4(2):141–149, 1982.

[3] L. Alvarez, R. Deriche, and F. Santana. *Recursivity and PDE's in image processing.* 15th International Conference on Pattern Recognition, volume I, pp 242–248, 2000.

[4] 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.

[5] L. Alvarez, P.L. Lions, and J.M. Morel. *Image selective smoothing and edge detection by nonlinear diffusion (II).* SIAM Journal of Numerical Analysis, 29:845–866, 1992.

[6] 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.

[7] L. Alvarez. *Images and PDE's.* Images, Wavelets and PDEs, vol.219 of Lecture Notes in Control and Information Sciences, Springer, 1996.

[8] H.C. Andrews and B.R. Hunt. *Digital Image Restoration.* Signal Processing, Prentice Hall, Englewood Cliffs, N.J., 1977.

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

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

[11] D. Barash. *Bilateral filtering and anisotropic diffusion : Towards a unified viewpoint.* Technical report, HP Laboratories Israel, 2000.

[12] F. Barbaresco. *Spatial denoising of statistical parameters estimation by Beltrami diffusion on embedding siegel space.* Physics in Signal and Image Processing (PSIP), 2003.

[13] J. Becker, T. Preusser, and M. Rumpf. *PDE methods in flow simulation post processing.* Computing and Visualization in Science, 3(3):159–167, 2000.

[14] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro. *Variational problems and partial differential equations on implicit surfaces: The framework and examples in image processing and pattern formation.* UCLA Research Report, June 2000.

[15] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. *Image inpainting.* Proceedings of the SIGGRAPH, pp.417–424. ACM Press, Addison Wesley Longman, 2000.

[16] 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.

[17] 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, 2003.

[18] 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.

[19] L. Blanc-Feraud, P. Charbonnier, G. Aubert, and M. barlaud. *Nonlinear image processing : Modeling and fast algorithm for regularization with edge detection.* Proceedings of the International Conference on Image Processing (ICIP), pp.474–477, Washington, USA, 1995.

[20] P. Blomgren. *Total Variation Methods for Restoration of Vector Valued Images.* PhD thesis, Department of Mathematics, University of California, Los Angeles, 1998.

[21] P. Blomgren and T.F. Chan. *Color tv: Total variation methods for restoration of vector-valued images.* IEEE Transactions on Image Processing, 7(3):304–309, 1998.

[22] D. Buerkle, T. Preusser, and M. Rumpf. *Transport and diffusion in timedependent flow visualization.* Proceedings IEEE Visualization, 2001.

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

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

[25] 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.

[26] T. Chan, S.H. Kang, and J. Shen. *Euler's elastica and curvature based inpainting.* SIAM Journal of Applied Mathematics, 2002.

[27] T.F. Chan, S.H. Kang, and J. Shen. *Total variation denoising and enhancement color images based on the cb and hsv color models.* Journal of Visual Communication and Image Representation, 12(4), 2000.

[28] T.F. Chan and J. Shen. *Non-texture inpainting by curvature-driven diffusions (CDD).* Journal of Visual Communication and Image Representation, 12(4):436–449, 2001.

[29] 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.

[30] T. Chan and J. Shen. *Mathematical models for local deterministic inpaintings.* Technical Report 00-11, Department of Mathematics, UCLA, Los Angeles, 2000.

[31] P. Charbonnier, G. Aubert, M. Blanc-Féraud, and M. Barlaud. *Two deterministic half-quadratic regularization algorithms for computed imaging.* Proceedings of the International Conference on Image Processing (ICIP), volume II, pages 168–172, 1994.

[32] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. *Deterministic edge-preserving regularization in computed imaging.* IEEE Transactions on Image Processing, 6(2):298–311, 1997.

[33] 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, 2004.

[34] M.T. Chu. *A list of matrix flows with applications.* Technical report, Department of Mathematics, North Carolina State University, 1990.

[35] M.T. Chu. *Matrix differential equations : A continuous realization process for linear algebra problems.* Technical report, Department of Mathematics, North Carolina State University, 1990.

[36] G.H. Cottet and L. Germain. *Image processing through reaction combined with nonlinear diffusion.* Mathematics of Computation, 61(204):659–673, 1993.

[37] 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.

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

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

[40] U. Diewald, T. Preusser, and M. Rumpf. *Anisotropic diffusion in vector field visualization on euclidian domains and surfaces.* IEEE Transactions on Visualization and Computer Graphics, 6(2):139–149, 2000.

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

[42] L. Florack. *Image Structure*. Kluwer Academic Publishers, 1997.

[43] M. A. Förstner and E. Gülch. *A fast operator for detection and precise location of distinct points, corners and centers of circular features.* Proceedings of the Intercommission Workshop of the International Society for Photogrammetry and Remote Sensing, Interlaken, Switzerland, 1987.

[44] B.R. Frieden. *Restoring with maximum likelihood and maximum entropy.* Journal Optical Society, 62:511–518, 1972.

[45] S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 6(6):721–741, 1984.

[46] S. Geman and D.E. McClure. Bayesian image analysis: an application to single photon emission tomography. *Proc. of the Statistical Computing Section p.12-18, Washington D.C*, 1985

[47] G. Gerig, O. Kubler, R. Kikinis, and F. Jolesz. *Nonlinear anisotropic filtering of MRI data.* IEEE TMI, 11(2):221–231, 1992.

[48] G. Gilboa, N. Sochen, and Y Zeevi. *Forward-and-backward diffusion processes for adaptative image enhancement and denoising.* IEEE Transactions on Image Processing, 2002.

[49] G. Gilboa, N. Sochen, and Y. Zeevi. *Regularized shock filters and complex diffusion.* Proceedings of ECCV'02, pages 399–413, Copenhaguen/Denmark.

[50] G. Gilboa, N. Sochen, and Y.Y. Zeevi. *Complex Diffusion Processes for Image Filtering.* Proceedings of ECCV'02, Copenhaguen/Denmark.

[51] G.H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.

[52] P.J. Green. Bayesian reconstruction from emission tomography data using a modified em algorithm. *IEEE Trans. Med. Imaging*, MI-9(1):84–93, March 1990.

[53] J. Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations.* Yale University Press, New Haven, 1923.

[54] B.R. Hunt. *TTe application of constrained least squares estimation to image restoration by digital computer.* IEEE Trans. Computers, C-22(9):805–812, 1973.

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

[56] JPEG Consortium Web Page. *http://www.jpeg.org*

[57] S. Kichenassamy. *The Perona–Malik paradox.* SIAM Journal of Applied Mathematics, 57(5):1328–1342, 1997.

[58] R. Kimmel, R. Malladi, and N. Sochen. *Image processing via the Beltrami operator.* Proceedings of the 3rd Asian Conference on Computer Vision, vol 1, pp.574–581, Hong Kong, 1998.

[59] 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 (IJCV), 39(2):111–129, 2000.

[60] R. Kimmel and N. Sochen. *Geometric-variational approach for color image enhancement and segmentation.* Scale-Space Theories in Computer Vision, Scale-Space'99, vol.1682 of LNCS, pp.295–305. Springer, 1999.

[61] 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.

[62] J.J. Koenderink. *The structure of images.* Biological Cybernetics, 50:363–370, 1984.

[63] P. Kornprobst. *Contributions à la Restauration d'Images et à l'Analyse de Séquences: Approches Variationnelles et Solutions de Viscosité.* PhD thesis, Université de Nice-Sophia Antipolis, 1998.

[64] P. Kornprobst, R. Deriche, and G. Aubert. *Image restoration via PDE's.* First Annual Symposium on Enabling Technologies for Law Enforcement and Security - SPIE Conference 2942, Boston, Massachusetts, USA, 1996.

[65] P. Kornprobst, R. Deriche, and G. Aubert. *Nonlinear operators in image restoration.* Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp.325–331, Puerto Rico, 1997.

[66] A. Koschan. *A comparative study on color edge detection.* Proceedings of the 2nd Asian Conference on Computer Vision, ACCV'95, pages 574–578.

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

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

[69] L. Lucido, R. Deriche, L. Alvarez, and V. Rigaud. *Sur quelques schémas numériques de résolution d'équations aux dérivées partielles pour le traitement d'images.* Rapport de Recherche 3192, INRIA, 1997.

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

[71] Y. Meyer *Oscillatory patterns in image processing and nonlinear evolution equations.* University Lecture Series, vol. 22, American Mathematical Society, Providence, 2001.

[72] M. Nielsen, L. Florack, and R. Deriche. *Regularization, scale-space and edge detection filters.* Journal of Mathematical Imaging and Vision (JMIV), 7(4):291–308, 1997.

[73] M. Nikolova. *Local strong homogeneity of a regularized estimator.* SIAM Journal of Applied Mathematics, 61(2):633–658, 2000.

[74] M. Nikolova. *Image restoration by minimizing objective functions with nonsmooth data-fidelity terms.* IEEE Workshop on Variational and Level Set Methods (VLSM), pp. 11–19, Vancouver, Canada, 2001.

[75] M. Nikolova and M. Ng. *Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning.* Proceedings of the International Conference on Image Processing. IEEE Signal Processing Society, 2001.

[76] 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, 1990.

[77] T. Papadopoulo and M.I.A. Lourakis. *Estimating the jacobian of the singular value decomposition: Theory and applications.* Research Report 3961, INRIA Sophia-Antipolis, 2000.

[78] A. Pardo and G. Sapiro. *Vector probability diffusion.* Proceedings of the International Conference on Image Processing (ICIP). IEEE Signal Processing Society, 2000.

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

[80] P. Perona and J. Malik. *Scale-space and edge detection using anisotropic diffusion.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 12(7):629–639, 1990.

[81] C.A. Poynton. Poynton's colour FAQ (www.inforamp.net/ poynton). Web page, 1995.

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

[83] 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.

[84] Bart M. ter Haar Romeny. *Geometry-driven diffusion in computer vision*. Computational imaging and vision. Kluwer Academic Publishers, 1994.

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

[86] G. Sapiro. *Vector-Valued Active Contours.* Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp.680–685, San Francisco, 1996. IEEE.

[87] G. Sapiro. *Color snakes.* Computer Vision and Image Understanding, 68(2), 1997.

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

[89] G. Sapiro and D.L. Ringach. *Anisotropic diffusion of multivalued images with applications to color filtering.* IEEE Transactions on Image Processing, 5(11):1582–1585, 1996.

[90] H. Scharr and J. Weickert. *An anisotropic diffusion algorithm with optimized rotation invariance.* G. Sommer, N. Krüger, C. Perwass (Eds.), Mustererkennung 2000, Springer, Berlin, pp.460–467, 2000.

[91] J. Shah. *Curve evolution and segmentation functionals: Applications to color images.* Proceedings of the International Conference on Image Processing (ICIP), pp.461–464, 1996.

[92] N. Sochen, R. Kimmel, and A.M. Bruckstein. *Diffusions and confusions in signal and image processing.* Journal of Mathematical Imaging and Vision (JMIV), 14(3):195–209, 2001.

[93] N. Sochen. *On affine invariance in the Beltrami framework for vision.* IEEE Workshop on Variational and Level Set Methods (VLSM), pp.51–56, Vancouver/Canada.

[94] N. Sochen, R. Kimmel, and R. Malladi. *A geometrical framework for low level vision.* IEEE Transaction on Image Processing, Special Issue on PDE based Image Processing, 7(3):310–318, 1998.

[95] 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.

[96] B. Tang, G. Sapiro, and V. Caselles. *Direction diffusion.* International Conference on Computer Vision (ICCV), 1998.

[97] 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 (IJCV), 36(2):149–161, 2000.

[98] S. Teboul, L. Blanc-Féraud, G. Aubert, and M. Barlaud. *Variational approach for edge-preserving regularization using coupled PDE's.* IEEE Transaction on Image Processing, Special Issue on PDE based Image Processing, 7(3):387–397, 1998.

[99] A.N. Tikhonov. *Regularization of incorrectly posed problems.* Soviet. Math. Dokl., 4:1624–1627, 1963.

[100] C. Tomasi and R. Manduchi. *Bilateral filtering for gray and color images.* Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp.839–846, 1998.

[101] D. Tschumperlé. *PDE's Based Regularization of Multivalued Images and Applications*. PhD thesis, Université de Nice-Sophia Antipolis, 2002.

[102] D. Tschumperlé. *Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's.* International Journal of Computer Vision (IJCV), vol.68, No.1, pp.65-82, 2006.

[103] D. Tschumperlé and R. Deriche. *Constrained and unconstrained PDE's for vector image restoration.* Proceedings of the 10th Scandinavian Conference on Image Analysis (SCIA), pp.153–160, Bergen, Norway, 2001.

[104] D. Tschumperlé and R. Deriche. *Diffusion tensor regularization with constraints preservation.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai Marriott, Hawaii, 2001.

[105] D. Tschumperlé and R. Deriche. *Regularization of orthonormal vector sets using coupled PDE's.* IEEE Workshop on Variational and Level Set Methods (VLSM), pp.3–10, Vancouver, Canada, 2001.

[106] 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, 2002.

[107] D. Tschumperlé and R. Deriche. *Diffusion PDE's on Vector-Valued images.* IEEE Signal Processing Magazine, 19(5):16–25, 2002.

[108] D. Tschumperlé and R. Deriche. *Orthonormal Vector Sets Regularization with PDE's and Aplications.* Internation Journal of Computer Vision (IJCV), 2002.

[109] D. Tschumperlé and R. Deriche *Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol.1, pp-651–656, 2003.

[110] 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 (PAMI), Vol.27, No.4, 2005.

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

[112] L.A. Vese and S. Osher. *Numerical methods for p-harmonic flows and applications to image processing.* CAM Report 01-22, UCLA, 2001.

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

[114] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, University of Kaiserslautern, Germany, Laboratory of Technomathematics, January 1996.

[115] J. Weickert. *Theoretical foundations of anisotropic diffusion in image processing.* Computing Supplement, 11:221–236, 1996.

[116] J. Weickert. *Coherence-enhancing diffusion of colour images.* 7th National Symposium on Pattern Recognition and Image Analysis, April 1997.

[117] J. Weickert. *A review of nonlinear diffusion filtering.* Scale-Space Theory in Computer Vision, Lecture Notes in Computer Science (Springer, Berlin), 1252:3–28, 1997.

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

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

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

[121] J. Weickert and B. Benhamouda. *A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox* Advances in Computer Vision, Springer, pages 1–10, Wien, 1997.

[122] J. Weickert and B. Benhamouda. *Why the Perona-Malik filter works.* Technical Report 97/22, Department of Computer Science, University of Copenhagen, 1997.

[123] 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.

[124] J. Weickert and C. Schnörr. *A theoretical framework for convex regularizers in pde-based computation of image motion.* International Journal of Computer Vision (IJCV), 45(3):245–264, 2001.

[125] 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.

[126] P. Wesseling. *Principles of Computational Fluid Dynamics.* Springer-Verlag Berlin Heidelberg New York., 2000.

[127] A.P. Witkin. *Scale-space filtering.* International Joint Conference on Artificial Intelligence, pp.1019–1021, 1983.

[128] A. Yezzi. *Modified curvature motion for image smoothing and enhancement.* IEEE Transactions on Image Processing, pp 345–352, vol. 7, No 3, 1998.