Vector-Valued Image Regularization with PDEs: A Common Framework for Different Applications

David Tschumperlé and Rachid Deriche

Abstract—In this paper, we focus on techniques for vector-valued image regularization, based on variational methods and PDEs. Starting from the study of PDE-based formalisms previously proposed in the literature for the regularization of scalar and vector-valued data, we propose a unifying expression that gathers the majority of these previous frameworks into a single generic anisotropic diffusion equation. On one hand, the resulting expression provides a simple interpretation of the regularization process in terms of local filtering with spatially adaptive Gaussian kernels. On the other hand, it naturally disassembles any regularization scheme into the smoothing process itself and the underlying geometry that drives the smoothing. Thus, we can easily specialize our generic expression into different regularization PDEs that fulfill desired smoothing behaviors, depending on the considered application: image restoration, inpainting, magnification, flow visualization, etc. Specific numerical schemes are also proposed, allowing us to implement our regularization framework with accuracy by taking the local filtering properties of the proposed equations into account. Finally, we illustrate the wide range of applications handled by our selected anisotropic diffusion equations with application results on color images.

Index Terms—Diffusion PDEs, color image regularization, denoising, inpainting, vector-valued smoothing, anisotropic filtering, flow visualization.

1 Introduction and State of the Art

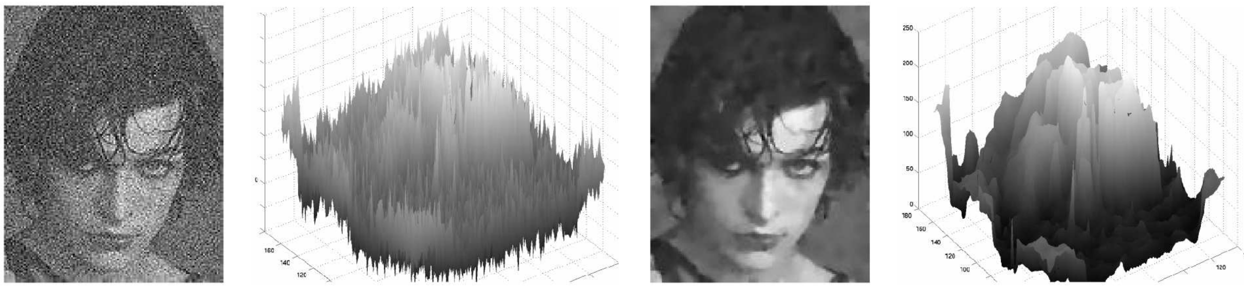
For several years, regularization algorithms have raised a huge interest in the computer vision and image processing community. It basically consists of simplifying a signal or an image, in a way that only interesting features are preserved while unimportant data (considered as "noise") are removed. By the way, such methods have direct applications for *image denoising*, but their abilities to create simplified representations of data are very interesting as well, when dealing with features extraction (edges and corners in images for instance). Actually, it is often one of the key stage performed by high-level algorithms in computer vision or image processing areas, such as object recognition, tracking, etc. Regularization algorithms are used as low-level steps in more complex processing pipelines and their adequations to the considered problems are crucial. For these reasons, a lot of regularization frameworks have already been proposed in the literature. Pioneering works in this area have been initiated, for instance, in [1], [3], [18], [19], [21], [34].

In the late 1980s, the framework of *nonlinear PDEs (partial differential equations)* led to strong improvements in the formalization of regularization methods. First created to describe physical laws and natural motions of mechanic objects and fluids (strings, water, wind [52]), PDEs were already widely studied. Interesting results coming from the fields of physics and mathematics have been recently extended and used to improve data regularization schemes. Nonlinear PDEs succeed in smoothing data while preserving large global features such as contours and corners (discontinuities of the signal) and their use within variational frameworks has opened new ways to handle classical image processing issues (restoration, segmentation, registration, etc.). Thus, many PDE-based schemes have been presented so far in the literature, particularly for the regularization of $2D$ *scalar images* $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ (see, for instance, [2], [4], [27], [30], [34], [37], [51], [53], [54] and references therein).

Another interesting property of nonlinear regularization PDEs such as $\frac{\partial I}{\partial t} = \mathcal{R}$ is the notion of *scale-space* behind: The data are gently regularized step-by-step and a continuous sequence of smoother images $I_{(t)}$ is generated whereas the evolution time t goes by. Obviously, such regularization algorithms must let the less significant data features disappear first, while the interesting ones are preserved as long as they become unimportant themselves within the image. Roughly speaking, regularization PDEs may be seen as *nonlinear filters* that simplify the image little by little and minimize then the image variations. Note, therefore, that they generally do not converge toward a very interesting solution. Most of the time, the image obtained at convergence ($t \rightarrow \infty$) is *constant*, corresponding to an image without any variations: This is actually the most simplified image we can obtain. To avoid this effect, *denoising algorithms* are usually based on a regularization term $\mathcal{R}$ coupled with a *data attachment* term $(I_{\text{noisy}} - I)$, also called *fidelity term*. It avoids the expected solution (regularized image) at convergence to be too different from the original noisy image (not

- D. Tschumperlé is with the Laboratoire GREYC UMR 6072, Equipe Image, Ecole Nationale Supérieure d'Ingénieurs de Caen, 14050 Caen Cedex France. E-mail: David.Tschumperle@greyc.ensicaen.fr.
- R. Deriche is with the Laboratoire Odyssée, INRIA Sophia-Antipolis, 06902 Sophia-Antipolis, France. E-mail: Rachid.Deriche@sophia.inria.fr.

Manuscript received 16 Feb. 2004; revised 19 Aug. 2004; accepted 10 Sept. 2004; published online 10 Feb. 2005.
Recommended for acceptance by S. Soatto.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0086-0204.



(a) (b) (c) (d)

Fig. 1. Example of image restoration with surface area (functional) minimization. (a) Noisy image. (b) Corresponding surface. (c) Restored image. (d) Corresponding surface.

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. In this article, we are mainly interested in the regularization term behavior rather than the fidelity term. For an interesting mathematical study about fidelity terms, please refer to [29], [31].

Extensions of these nonlinear regularization PDEs to *vector-valued images* $\mathbf{I} : \Omega \to \mathbb{R}^n$ have been recently proposed, leading to more elaborated expressions: A *coupling between image channels* generally appears in the equations, through the consideration of a *local vector geometry*, given pointwise by the spectral elements λ_+, λ_- (positive eigenvalues) and θ_+, θ_- (orthogonal eigenvectors) of the 2×2 symmetric and semi-positive-definite matrix, also called *structure tensor* [45], [48], [51], [55]:

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

Each ∇I_j corresponds to the spatial gradient of the *j*th channel (i.e., vector component) of the vector-valued image $\mathbf{I}$. As demonstrated in [55], the structure tensor $\mathbf{G}$ is particularly interesting since the eigenvalues $\lambda_\pm$, respectively, define the local min/max *vector-valued variations* of $\mathbf{I}$ in corresponding spatial directions $\theta_\pm$ (eigenvectors), i.e., the spectral elements of $\mathbf{G}$ define the *local geometry of the vector-valued image discontinuities*. (Note that $\lambda_+ = \|\nabla I\|$ and $\theta_+ = \nabla I / \|\nabla I\|$ *for scalar images*, when $n = 1$).

Starting from this basis, we can classify diffusion PDE's schemes proposed in the literature into one of these three following approaches, related to different interpretation levels of the regularization process, described in Sections 1.1, 1.2, and 1.3 below.

1.1 Functional Minimization

Regularizing an image $\mathbf{I}$ may be seen as the minimization of a functional $E(\mathbf{I})$ measuring a global image variation. The idea is that minimizing this functional will flatten the image variations, then gradually remove the noise:

$$\min_{\mathbf{I}:\Omega \to \mathbb{R}^n} E(\mathbf{I}) = \int_\Omega \phi(\mathcal{N}(\mathbf{I}))\, d\Omega, \tag{1}$$

where $\mathcal{N}(\mathbf{I})$ is a norm related to *local* image variations and $\phi : \mathbb{R} \to \mathbb{R}$ is an increasing function. One often chooses $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ + \lambda_-}$ for vector-valued images [7], [10], [33], [41], [46], [47], but other norms are possible such as $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+}$ [9], [35], [36], or $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ - \lambda_-}$ [38], [49], [50]. For scalar images $I : \Omega \to \mathbb{R}$, these norms naturally reduce to the same expression $\mathcal{N}(I) = \|\nabla I\|$. Then, the minimization of (1) is performed through a gradient descent (PDE), coming from the Euler-Lagrange equations of $E(\mathbf{I})$.

This technique has been widely used in the context of scalar images [4], [15], [16], [24], [25], [54], for instance, by minimizing the area of a surface representing the image (Fig. 1). Corresponding references for vector-valued images are: [10], [22], [33], [37], [39], [42], [44].

1.2 Divergence Expressions

A regularization process may be also more locally designed, as a diffusion of pixel values, viewed as chemical concentrations or temperatures [51], [20], and directed by a 2×2 *diffusion tensor* $\mathbf{D}$ (symmetric and definite-positive matrix):

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\,(\mathbf{D} \nabla I_i) \qquad (i = 1..n). \tag{2}$$

It is generally assumed that the spectral elements of $\mathbf{D}$ give the two weights and directions of the local smoothing performed by (2). $\mathbf{D}$ is then specially designed from the spectral elements of the structure tensor $\mathbf{G}$ in order to *anisotropically* smooth $\mathbf{I}$, while taking its intrinsic local geometry into account, preserving its global discontinuities. Anyway, we will show throughout this paper that the interpretation of the PDE (2) in terms of local smoothing is not so obvious. Actually, the spectral shape of the tensors $\mathbf{D}$ is not always representative of the effective smoothing performed by (2). This can be easily understood as follows: Let us consider a simple case of two different "divergence" tensors $\mathbf{D}_1$ and $\mathbf{D}_2$ defined by

$$\mathbf{D}_1 = \frac{\mathbf{Id}}{\|\nabla I\|} \quad \text{and} \quad \mathbf{D}_2 = \frac{1}{\|\nabla I\|^3}\,(\nabla I \nabla I^T).$$

$\mathbf{D}_1$ is isotropic (since it is only a weighted identity matrix) while $\mathbf{D}_2$ is purely anisotropic (only one eigenvalue is nonzero). Nevertheless, it is easy to verify that

$$\mathrm{div}\,(\mathbf{D}_1 \nabla I) = \mathrm{div}(\mathbf{D}_2 \nabla I) = \mathrm{div}\left(\frac{\nabla I}{\|\nabla I\|}\right),$$

which actually corresponds to the well-known *TV minimization* of scalar images: Two tensors with *very different shapes* lead to the same equation, accordingly to the same regularization behavior.

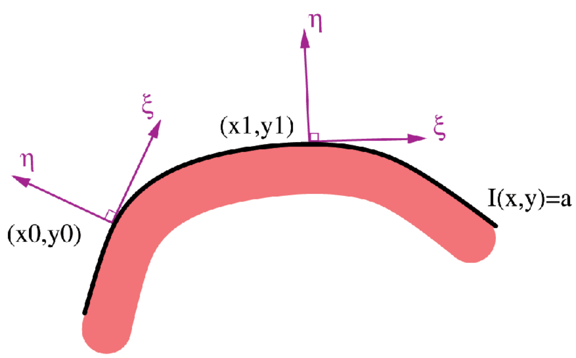


Fig. 2. Principle of regularization techniques based on oriented Laplacians: Two 1D smoothing are done along orthogonal axes ξ and η that are different for each image points.

1.3 Oriented Laplacians

2D image regularization may be finally seen as the simultaneous juxtaposition of two *oriented 1D heat flows*, leading to $1D$ Gaussian smoothing processes along orthonormal directions $\xi \perp \eta$, with different weights c_1 and c_2 [26], [38], [45], [48] (Fig. 2):

$$\frac{\partial \mathbf{I}}{\partial t} = c_1 \frac{\partial^2 \mathbf{I}}{\partial \xi^2} + c_2 \frac{\partial \mathbf{I}}{\partial \eta^2} = c_1\, \mathbf{I}_{\xi\xi} + c_2\, \mathbf{I}_{\eta\eta}. \qquad (3)$$

Like divergence expressions, the smoothing weights c_1, c_2 and directions ξ, η are directly designed from the spectral elements $\lambda_\pm$ and $\theta_\pm$ of $\mathbf{G}$, in order to perform edge-preserving smoothing, mainly along the direction θ_- orthogonal to the image discontinuities.

1.4 Link between the Three Formulations

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

$$\begin{aligned} &(A)\ \min_{I:\Omega\to\mathbb{R}} \int_\Omega \phi(\|\nabla I\|)\, d\Omega \\ \Rightarrow\ &(B)\ \frac{\partial I}{\partial t} = \operatorname{div}\left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|}\ \nabla I\right) \\ \Rightarrow\ &(C)\ \frac{\partial I}{\partial t} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|}\ I_{\xi\xi} + \phi''(\|\nabla I\|)\ I_{\eta\eta}, \end{aligned} \qquad (4)$$

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 corresponding divergence-based tensor $\mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|}\ \mathbf{Id}$.

In this paper, we propose a way to find such links for the more general case of vector-valued regularization based on PDEs. We tackle each of these three interpretation levels (1), (2), and (3) in their more general forms, and derive the corresponding equations. We particularly show that the oriented-Laplacian formalism has an interesting interpretation in terms of *local filtering*, and represents the right smoothing geometry performed by the PDEs. Thus, it allows us to design a new and efficient vector-valued regularization approach, respecting desired local smoothing properties (Section 4), as well as propose new and adapted numerical schemes (Section 6). Finally, we apply our method to solve a wide range of image processing issues, including color image restoration, inpainting, magnification, and flow visualization (Section 7).

2 From Variational to Divergence Forms

We first consider vector-valued image regularization as a variational problem. We want to find the corresponding *divergence-based expression*, i.e., the link $(\mathrm{A}) \Rightarrow (\mathrm{B})$.

2.1 A Generic Functional

Instead of regularizing a functional such as (1) depending on a predefined variation norm $\mathcal{N}(\mathbf{I})$, we would rather propose to minimize this more generic ψ-functional:

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} E(\mathbf{I}) = \int_\Omega \psi(\lambda_+, \lambda_-)\ d\Omega. \qquad (5)$$

As vector-valued images possess two distinct variation estimators λ_+ and λ_- (eigenvalues of the structure tensor $\mathbf{G} = \sum_{j=1}^n \nabla I_j \nabla I_j^T$), it seems natural to minimize a functional defined by a function $\psi : \mathbb{R}^2 \to \mathbb{R}$ of two variables instead of a single one. This is actually a generic extension of the ϕ-function formulation for vector-valued images (4).

2.2 Corresponding Euler-Lagrange Equations

The Euler-Lagrange equations of (5) can be derived and reduced to a simple form of *divergence-based expression* (see Appendix A which can be found on the Computer Society Digital Library at http://computer.org/tkde/archives.htm for details about this Euler-Lagrange derivation):

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\,(\mathbf{D}\nabla I_i) \qquad (i = 1..n), \qquad (6)$$

where the 2×2 diffusion tensor $\mathbf{D}$ is defined as:

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

It results then in a divergence-based equation such as (2), where the diffusion tensor $\mathbf{D}$ is simply defined from the partial derivatives of ψ, and the eigenvectors θ_+, θ_- of $\mathbf{G}$. Note that the tensor $\mathbf{D}$ has the same *orientation* as the structure tensor $\mathbf{G}$ (same eigenvectors).

2.3 Link with Other Approaches

The choice of particular 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 [33], [42]:

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

or the Beltrami flow framework [22]:

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

More generally, our variational approach (5) shows that the eigenvalues of a divergence tensor $\mathbf{D}$ can be seen as the *gradient of a potential function* ψ, linked to the functional (5).

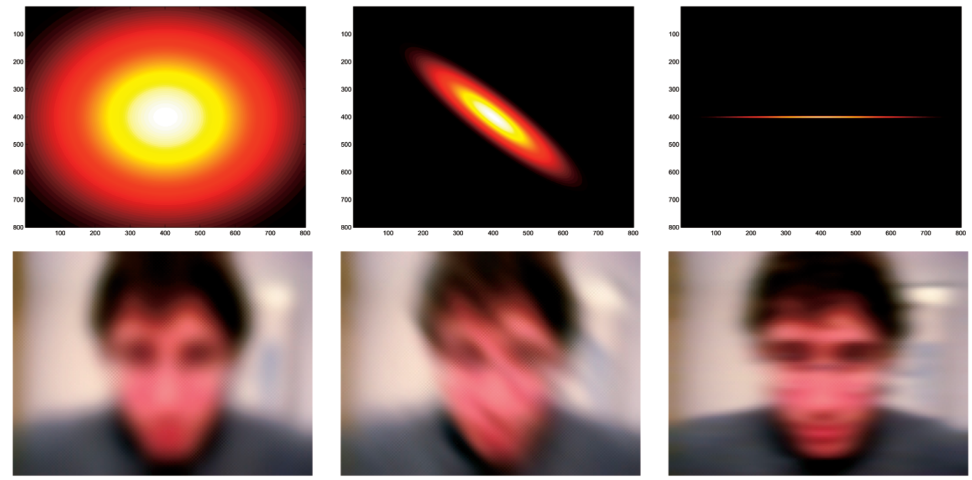


Fig. 3. Trace-based PDEs (7) viewed as convolutions by oriented 2D Gaussians.

If such a potential ψ exists, it is easy to find the energy (5) corresponding to a given divergence-based expression (6).

Note that the problem of the local geometric interpretation of (6) in terms of smoothing weights and directions also applies here. As illustrated by the scalar-valued ϕ-functional case (4), $\mathbf{D}$ may not represent the right smoothing geometry of the regularization process.

3 From Divergences to Oriented Laplacians

We rather want to develop divergence forms as (6) into their corresponding *oriented Laplacian* formulations, i.e., find the link $(\mathrm{B}) \Rightarrow (\mathrm{C})$. Actually, Oriented Laplacians are particularly well designed to geometrically understand the underlying smoothing process performed by the PDE.

3.1 Geometric Meaning of Oriented Laplacians

Let us consider the oriented Laplacian-based equation (3). As $\xi \perp \eta$, this PDE can be equivalently written as:

$$\frac{\partial I_i}{\partial t} = c_1\ I_{i_{\xi\xi}} + c_2\ I_{i_{\eta\eta}} = \mathbf{trace}\ (\mathbf{T}\mathbf{H}_i) \qquad (i = 1..n), \qquad (7)$$

where $\mathbf{H}_i$ is the *Hessian matrix of the vector component* I_i and $\mathbf{T}$ is the 2×2 tensor defined by: $\mathbf{T} = c_1 \xi\xi^T + c_2 \eta\eta^T$, characterized by its two eigenvalues c_1, c_2 and its two corresponding eigenvectors ξ, η. Let us suppose first that $\mathbf{T}$ is a constant tensor over the definition domain Ω.

Then, the formal solution of the PDE (7) is:

$$I_{i_{(t)}} = I_{i_{(t=0)}}\ *\ G^{(\mathbf{T},t)} \qquad (i = 1..n), \qquad (8)$$

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

$$G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t}\ \exp\left(-\frac{\mathbf{x}^T\mathbf{T}^{-1}\mathbf{x}}{4t}\right) \qquad \text{with} \qquad \mathbf{x} = (x \quad y)^T. \qquad (9)$$

Proof. From the expression (9), we can compute the temporal and spatial derivatives of $G^{(\mathbf{T},t)}$:

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

and

$$\begin{cases} \nabla G^{(\mathbf{T},t)} &= -\frac{1}{8\pi t^2}\ \exp\left(-\frac{\mathbf{x}^T\mathbf{T}^{-1}\mathbf{x}}{4t}\right)\ \mathbf{T}^{-1}\mathbf{x} \\ \mathbf{H}_{G^{(\mathbf{T},t)}} &= -\frac{1}{8\pi t^2}\ \exp\left(-\frac{\mathbf{x}^T\mathbf{T}^{-1}\mathbf{x}}{4t}\right)\mathbf{T}^{-1}\left(\mathbf{Id} - \frac{\mathbf{x}\mathbf{x}^T\mathbf{T}^{-1}}{2t}\right), \end{cases}$$

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

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

And, as the convolution is a linear operation, we have

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

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. □

It is a generalization of the Koenderink's idea [23], who proved this property in the field of computer vision for the *isotropic diffusion tensor* $\mathbf{T} = \mathbf{Id}$, resulting in the well-known *heat flow* equation: $\frac{\partial I_i}{\partial t} = \Delta I_i$.

Fig. 3 illustrates two Gaussian kernels $G^{(\mathbf{T},t)}(x, y)$, respectively, obtained with isotropic and anisotropic tensors $\mathbf{T}$ (*up*) and the corresponding evolutions of the diffusion PDE (7) on a color image (*down*). It is worth to notice that the Gaussian kernels $\mathbf{G}^{(\mathbf{T},t)}$ give the classical

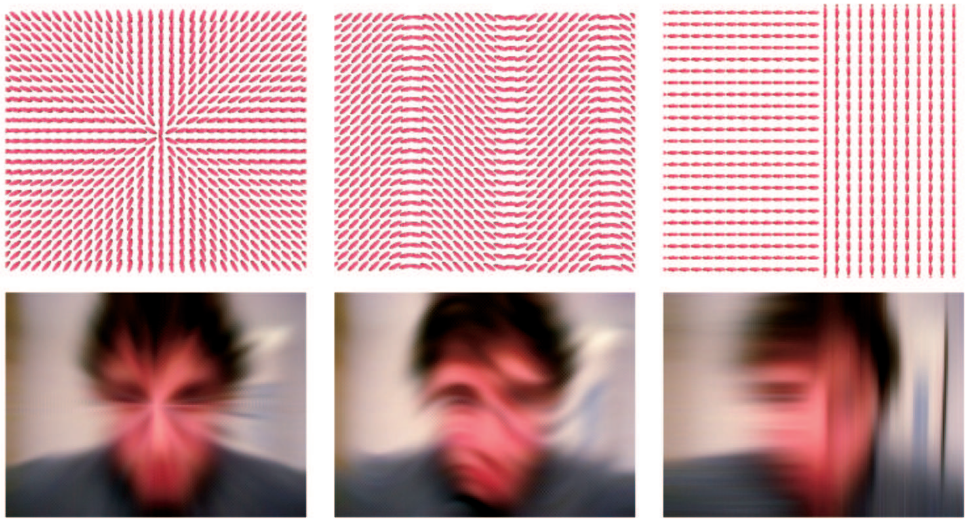


Fig. 4. Trace-based PDEs (7) with nonconstant diffusion tensor fields $\mathbf{T}$. Interpretation in terms of nonlocal filtering.

representations of the tensors $\mathbf{T}$ with ellipsoids. Conversely, it is clear that the tensors $\mathbf{T}$ represent the exact geometry of the smoothing performed by the PDE (7).

When $\mathbf{T}$ is not constant (which is generally the case), i.e., represents a field $\Omega \rightarrow \mathrm{P}(2)$ of variable diffusion tensors, the PDE (7) becomes *nonlinear* and 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. 4 illustrates two examples of spatially varying tensor fields $\mathbf{T}$, represented with fields of ellipsoids (*up*), and the corresponding evolutions of (7) on a color image (*down*). As before, *the shape of each tensor* $\mathbf{T}$ *gives the exact geometry of the local smoothing process* performed by the trace-based PDE (7) point by point.

Note that this local filtering concept makes the link between a generic form of vector-valued diffusion PDEs expressed through a trace operator (7) and the *Bilateral filtering* techniques, as described in [5], [43]. Another similar approach based on non-Gaussian convolution kernels has been also proposed for the specific case of Beltrami Flow in [40].

With the PDE (7), we are naturally disassembling the regularization itself and its underlying smoothing geometry, which is given by the spectral elements of a trace-tensor $\mathbf{T}$. Conversely to divergence equations, the choice of the tensor is unique here: The shape of the trace tensor $\mathbf{T}$ is really giving the correct smoothing geometry performed by the PDE (7).

3.2 Trace-Based and Divergence-Based Tensors

Differences between divergence tensors $\mathbf{D}$ in (2) and trace tensors $\mathbf{T}$ in (7) can be understood as follows: We can develop the divergence equation (2) as:

$$\operatorname{div}(\mathbf{D}\nabla I_i) = \operatorname{trace}(\mathbf{DH}_i) + \nabla I_i^T\, \vec{\mathbf{div}}\,(\mathbf{D}),$$

where $\vec{\mathbf{div}}\,()$ is defined as a divergence operator acting on matrices and returning vectors:

$$\text{if } \mathbf{D} = (d_{ij}), \vec{\mathbf{div}}\,(\mathbf{D}) = \begin{pmatrix} \operatorname{div}(d_{11} \quad d_{12})^T \\ \operatorname{div}(d_{21} \quad d_{22})^T \end{pmatrix}.$$

Then, an additional term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$ appears, connected to the *spatial variation* of the tensor field $\mathbf{D}$. It may perturb the smoothing behavior given by the first part $\operatorname{trace}(\mathbf{DH}_i)$, which actually corresponds to a local smoothing directed by the spectral elements of $\mathbf{D}$. As a result, the divergence-based equation (2) may smooth the image $\mathbf{I}$ with weights and directions that are *different* than the spectral elements of $\mathbf{D}$. This is actually the case for the scalar ϕ-function formulation (4), where the smoothing process does not behave finally and, fortunately, as an isotropic one, despite the isotropic form of the divergence tensor $\mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|}\,\mathbf{Id}$.

3.3 Developing the Divergence Form

Actually, if we consider that the divergence tensor $\mathbf{D}$ only depends on the spectral elements of the structure tensor $\mathbf{G}$:

$$\mathbf{D} = f_1(\lambda_+, \lambda_-)\theta_+\theta_+^T + f_2(\lambda_+, \lambda_-)\theta_-\theta_-^T \tag{10}$$

with $f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$, (which is the case for most of the proposed equations in the literature), then we can develop the corresponding divergence equation $\operatorname{div}(\mathbf{D}\nabla I_i)$ into oriented Laplacians, i.e., this trace-based PDE (full demonstration can be found in Appendix B which can be found on the Computer Society Digital Library at http://computer.org/tkde/archives.htm):

$$\operatorname{div}(\mathbf{D}\nabla I_i) = \sum_{j=1}^{n} \operatorname{trace}\left((\delta_{ij}\mathbf{D} + \mathbf{Q}^{ij})\mathbf{H}_j\right), \tag{11}$$

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

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

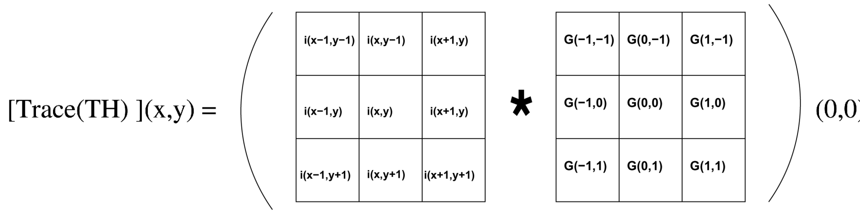
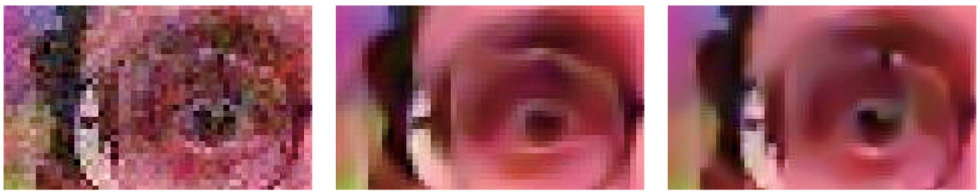


Fig. 5. Numerical schemes.



(a) (b) (c)

Fig. 6. Comparisons of numerical schemes. (a) Noisy image, (b) scheme using Hessian discretization, and (c) scheme using local filtering techniques.

with

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

This development (11) expresses a whole range of previously proposed vector-valued regularization algorithms (variational and divergence based PDEs) into an extended trace-based equation, composed of *several channel-diffusion contributions* that have direct geometric interpretations in terms of local filtering. The interesting point is that *additional diffusion tensors* $\mathbf{Q}^{ij}$ are appearing and contribute to *modify the smoothing behavior* which is finally *not given by the initial divergence tensor* $\mathbf{D}$.

4 A Unified Expression

From these previous developments, we can now define a *generic vector-valued regularization PDE*:

$$\frac{\partial I_i}{\partial t} = \sum_{j=1}^{n} \text{trace}\,(\mathbf{A}^{ij}\mathbf{H}i) \qquad (i = 1..n), \tag{12}$$

where the $\mathbf{A}^{ij}$ forms a family of 2×2 symmetric matrices, and the $\mathbf{H}_i$ designate the Hessian matrices of I_i. Actually, this expression can be equivalently written with a slight abuse of notations, in a *super-matrix* form:

$$\frac{\partial \mathbf{I}}{\partial t} = \overrightarrow{\mathbf{trace}}\,(\mathcal{AH}), \tag{13}$$

where $\mathcal{A}$ is the *matrix of diffusion tensors* $\mathbf{A}^{ij}$ (and is itself *symmetric*), and $\mathcal{H}$ is the *vector of Hessian matrices* $\mathbf{H}_j$. The matrix product $\mathcal{AH}$ in (13) is then seen *submatrix by submatrix*, and the operator $\overrightarrow{\mathbf{trace}}()$ returns the vector in $\mathbb{R}^n$, corresponding to the trace of each submatrix in the resulting vector of matrices.

4.1 Link with Previous Expressions

The PDE (12) is a unifying equation that can be used to describe a wide range of vector-valued regularization:

- First, it develops both variational and divergence-based approaches (that can be written as
$$\frac{\partial I_i}{\partial t} = \text{div}\,(\mathbf{D}\nabla I_i),$$
as developed in Section 2) into a very local formulation. This particularly includes the works done in [10], [20], [22], [33], [37], [39], [42], [48], [51] among others. As described above, the 2×2 tensors $\mathbf{A}^{ij}$ are then defined to be $\mathbf{A}^{ij} = \delta_{ij}\mathbf{D} + \mathbf{Q}^{ij}$. Note that the $\mathbf{Q}^{ij}$ $(i \neq j)$ corresponds here to diffusion contributions of other channels I_j in the current one I_i. This kind of *diffusion energy transfer* can be considered as a particular *coupling* of the corresponding vector-valued diffusion PDE.
- Second, the PDE (12) also gathers the oriented-Laplacian formulations $\frac{\partial I_i}{\partial t} = \text{trace}\,(\mathbf{TH}_i)$, by choos-

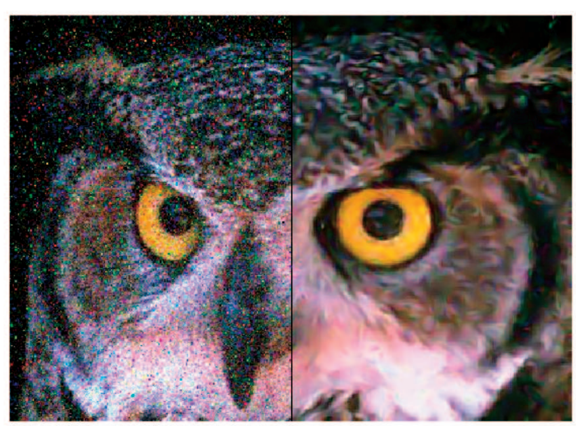
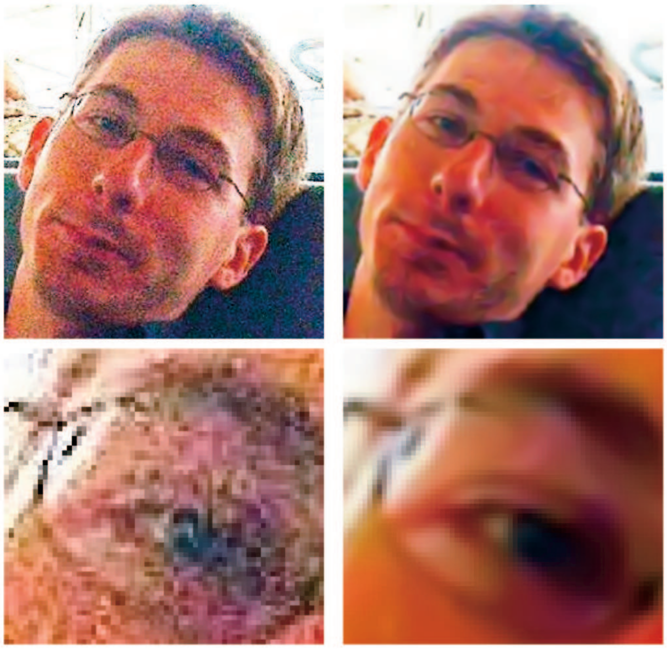


Fig. 7. Using our vector-valued regularization PDE's, to restore a color image artificially degraded (added a mixture of Salt & Pepper and Gaussian noises).



(a) (b)

Fig. 8. Using our vector-valued regularization PDE (14), for color image restoration. (a) Noisy color image and (b) restored color image.

ing $A^{ij} = \delta_{ij}\mathbf{T}$. In this case, the supermatrix $\mathcal{A}$ is diagonal and no diffusion energy transfer occurs between image channels I_i. The vector coupling is only done through the use of the structure tensor $\mathbf{G}$ for the computation of the local smoothing geometry. This unifies the formulations proposed in [26], [38], [45], [48].

5 A New Regularization PDE

We propose now to design a new vector-valued regularization PDE that follows desired local geometric properties (particularly for image denoising). These constraints will naturally define a specific form of regularization PDE, from the very generic form (12):

- We do not want to mix diffusion contributions between image channels. The desired coupling between vector components I_i should only appear in the diffusion PDE through the computation of the structure tensor $\mathbf{G}$, in order to control the local smoothing behavior of the regularization process. This means we have to define only one diffusion tensor $\mathbf{A}$, then choose $\mathbf{A}^{ij} = \delta_{ij}\mathbf{A}$. Undesired coupling terms are then avoided.
- On homogeneous regions (corresponding to low vector variations), we want to perform an *isotropic smoothing*, i.e., a 2D heat flow that smoothes the noise efficiently with no-preferred directions: $\frac{\partial I_i}{\partial t} \simeq \Delta I_i = \mathrm{trace}(\mathbf{H}_i)$. The tensor $\mathbf{A}$ must then be *isotropic* in these regions:

$$\lim_{(\lambda_+ + \lambda_-)\to 0} \mathbf{A} = \alpha\mathbf{Id}.$$

(a) (b)

Fig. 9. Using our vector-valued regularization PDE (14), for improvement of lossy compressed images. (a) Lossy compressed JPEG image and (b) Improved color image.

- On vector edges (corresponding to high vector variations), we want to perform an *anisotropic smoothing along the vector edges θ_-*, in order to preserve them while removing the noise: $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\beta\theta_-{\theta_-}^T\mathbf{H}_i)$, where β is a function decreasing anyway for very high variations, avoiding the oversmoothing of sharp corners. The tensor $\mathbf{A}$ must be *anisotropic* in these regions:

$$\lim_{(\lambda_+ + \lambda_-)\to \infty} \mathbf{A} = \beta\theta_-\theta_-^T.$$

The following multivalued regularization PDE respects all these local geometric properties:

$$\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}\mathbf{H}_i) \qquad (i = 1..n), \tag{14}$$

where $\mathbf{T}$ is the tensor field defined pointwise as:

$$\mathbf{T} = f_-\left(\sqrt{\lambda_+^* + \lambda_-^*}\right)\theta_-^*{\theta_-^*}^T + f_+\left(\sqrt{\lambda_+^* + \lambda_-^*}\right)\theta_+^*{\theta_+^*}^T.$$

$\lambda_\pm^*$ and $\theta_\pm^*$ are defined to be the spectral elements of $\mathbf{G}_\sigma = \mathbf{G} * G_\sigma$, a *Gaussian smoothed version of the structure tensor* $\mathbf{G}$, allowing us to retrieve a more coherent vector-geometry and giving a better approximation of the vector discontinuities directions (see also [51]). For our experiments in Section 7, we chose

$$f_+(s) = \frac{1}{1+s^2} \quad \text{and} \quad f_-(s) = \frac{1}{\sqrt{1+s^2}}.$$

This is, of course, one possible "empiric" choice (inspired from the *hypersurface formulation* of the scalar case [4]) that verifies the above geometric properties, relying on practical experience.

The point is that we can easily adapt the weighting functions f_+ and f_- to obtain regularization behaviors for specific problems, since we are sure of the local smoothing

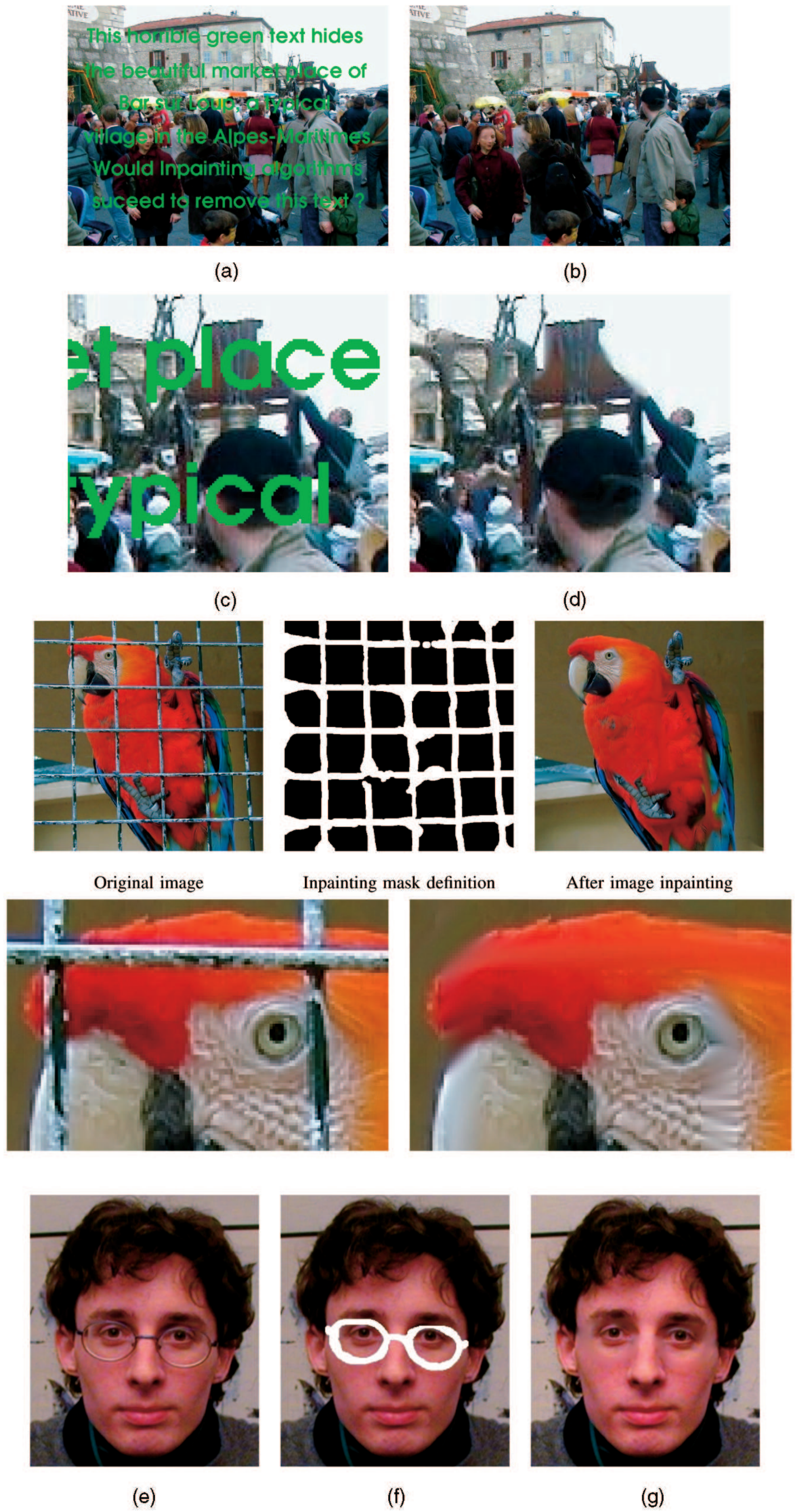


Fig. 10. Using vector-valued regularization PDE's for color image inpainting (2). (a) Image with undesired text. (b) Inpainted color image. (c) Zoom of (a). (d) Zoom of (b). (e) Original color image. (f) Image + Inpainting mask. (g) Inpainted image.

process performed by (14). This vector-valued regularization equation smoothes the image in coherent spatial directions and preserves then well the edges, by allowing only the necessary geometric coupling between vector channels I_i. Its form has steadily followed the local analysis of classical multivalued regularization algorithms.

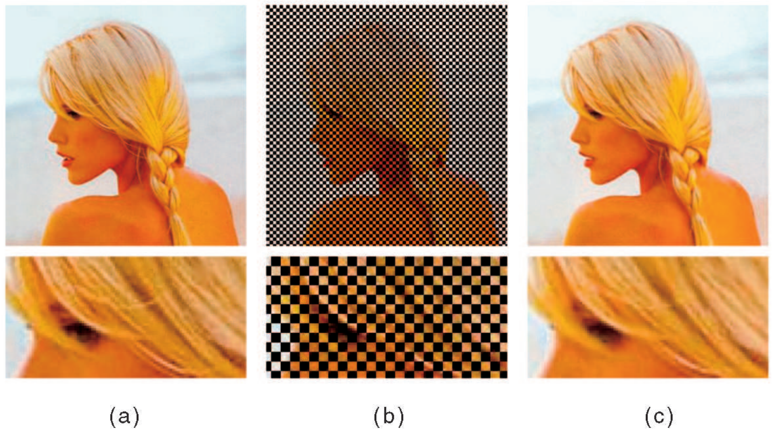


Fig. 11. Using vector-valued regularization PDEs for image reconstruction. (a) Color image, (b) removing 50 percent of the pixels, and (c) reconstructured image.

6 Numerical Schemes

The numerical implementation of the PDE (14) can be done with classical numerical schemes, based on spatial discretizations with centered finite differences of the gradients and the Hessians [28]. Here, we propose an alternative approach based on the local filtering interpretation of trace-based equations (7), proposed in Section 3. The idea is as follows: The smoothing can be locally performed by applying a spatially varying mask over the image. For each point (x, y) of the image $\mathbf{I}$, we compute the oriented Gaussian kernel $\mathbf{G}^{(\mathbf{T},t)}$ corresponding to the tensor $\mathbf{T}$, defined by (14). Then, we apply it on each local neighborhood $I_i(x, y)$, as shown in Fig. 5.

The main advantages of this numerical scheme are:

1. It numerically preserves the *maximum principle* since the local filtering is done only with *normalized kernels*.
2. It is more precise, since the computed local kernel corresponds exactly to the smoothing to perform. No (imprecise) second derivatives have to be computed (Fig. 6), and local filtering kernel is better oriented.

As for the shortcomings of this scheme, we have to mention that it is more time-consuming, since we have to compute a different Gaussian kernel (i.e., exponential functions) at each image point and for each iteration. For our experiments, we chose 5×5 convolution kernels (Fig. 6). Note how edge details are better preserved in Fig. 6c (look at the glint inside the eye).

7 Applications

We illustrate here the wide range of image processing applications that we can handle with our presented approach, through our vector-valued regularization PDE (14).

7.1 Color Image Restoration

Despite the emergence of digital cameras, color image restoration may be still needed. Fig. 8 represents a digital photograph with *real noise*, due to the bad lighting conditions during the snapshot. Our vector-valued regularization PDE can successfully remove the noise, while preserving the global features of the image (see also Fig. 7).

7.2 Improvement of Lossy Compressed Images

Digital images, due to their big memory size, are often stored in a more compact form obtained with *lossy compression algorithms* (JPEG being the most popular). It often introduces visible image artefacts: For instance, bloc effects are classical JPEG drawbacks. Using our flow (14) significantly improves the quality of such degraded images (Fig. 9). In this case, we chose a high parameter σ (variance of the structure tensor presmoothing), since a lot of structures in this image are quite linear. It helps then to retrieve linear structure, such as the gnome's hair.

7.3 Color Image Inpainting

Recently, an interesting application of diffusion PDEs named *image inpainting*, has been proposed in [8], [12], [13], [14]. It consists of filling undesired holes (defined by

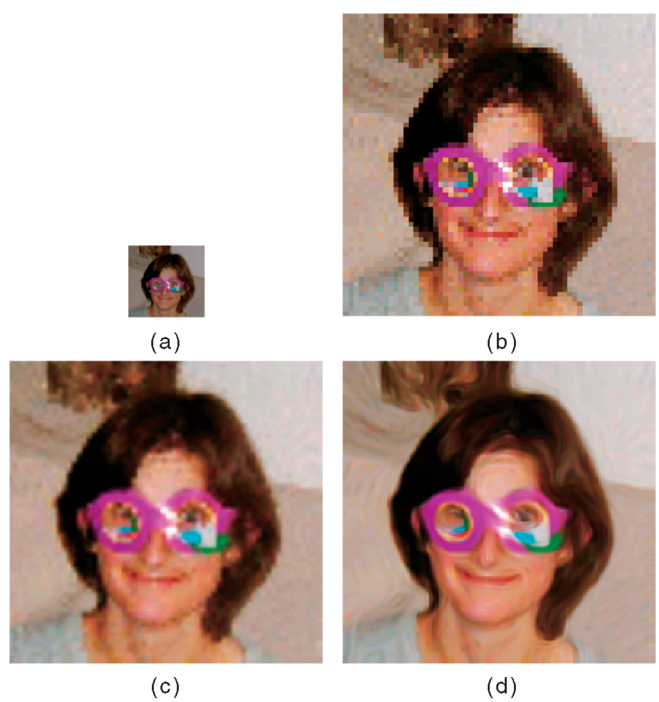
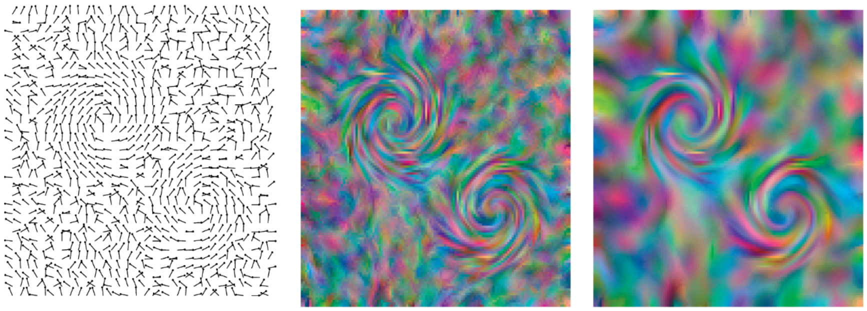


Fig. 12. Using vector-valued regularization PDE' for image magnification ($\times 4$). (a) Original color image (64×64). (b) Bloc interpolation. (c) Linear interpolation. (d) Interpolation with PDEs.



(a) (b) (c)

Fig. 13. Using our vector-valued regularization PDE (15), for flow visualization (1). (a) Flow visualization with arrows. (b) Flow visualization with diffusion PDEs (5 iter.). (c) Flow visualization with diffusion PDEs (15 iter.).

the user) in an image by *interpolating the data* located at the neighborhood of the holes. It is possible to do that by applying our PDE (14) only in the holes to fill: boundaries pixels will be diffused until they completely fill the missing regions, in a *structure-preserving way*. Important issues may be solved with this kind of algorithms as, for instance: removing text on images (Fig. 10), removing real objects in photographs (Fig. 10) or reconstruct partially coded images for image compression purposes (Fig. 11).

7.4 Color Image Magnification

With the same techniques, one can easily perform image magnification. Starting from a linear interpolation of a small image, and applying our PDE (14) on the image (excepted on the original known pixels), we can retrieve nonlinear magnified images without jagging or bloc effects, inherent to classical linear interpolation techniques (Fig. 12).

7.5 Flow Visualization

Considering a 2D vector field $\mathcal{F} : \Omega \rightarrow \mathbb{R}^2$, we have several ways to visualize it. We can first use vectorial graphics (Fig. 13a), but we have to subsample the field since this kind of representation is not adapted to represent big flows. A better solution is as follows: We smooth a completely noisy (color) image $\mathbf{I}$, with a regularizing flow equivalent to (14) but where $\mathbf{T}$ is directed by the directions of $\mathcal{F}$, instead of the local geometry of $\mathbf{I}$:

$$\frac{\partial I_i}{\partial t} = \mathrm{trace}\left(\left[\frac{1}{\|\mathcal{F}\|}\mathcal{F}\mathcal{F}^T\right]\mathbf{H}_i\right) \quad (i = 1..n). \tag{15}$$

Whereas the PDE evolution time t goes by, more global structures of the flow $\mathcal{F}$ appear, i.e., a visualization *scale-space* of $\mathcal{F}$ is constructed (Fig. 14). Here, our used regularization equation (15) *ensures that the smoothing of the pixels is done exactly in the direction of the flow* $\mathcal{F}$. This is not the case in [6], [11], [17], where the authors based their equations on a divergence 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 3.

8 Conclusion and Perspectives

In this paper, we proposed a new formalism allowing to express a large set of previous vector-valued regularization approaches within a common local expression. This formulation is particularly adapted to understand the local smoothing behavior of diffusion PDEs. Indeed, it explains the link between the diffusion tensor shapes in divergence or trace-based equations, and the actual smoothing performed by these processes, in term of local filtering. From this general study, we defined a new and particular regularization equation, based on the respect of a coherent anisotropic smoothing preserving the global features of vector images. We proposed as well specific numerical schemes adapted for accurate implementations. The application to several problems related to color images and flow visualization illustrated the efficiency of our method to deal with concrete cases based on the use of vector-valued regularization processes.

Note 1: Other applications results and color demos can be found at the following URL: http://www-sop.inria.fr/odyssee/research/tschumperle-deriche:02d/appliu/index.html.

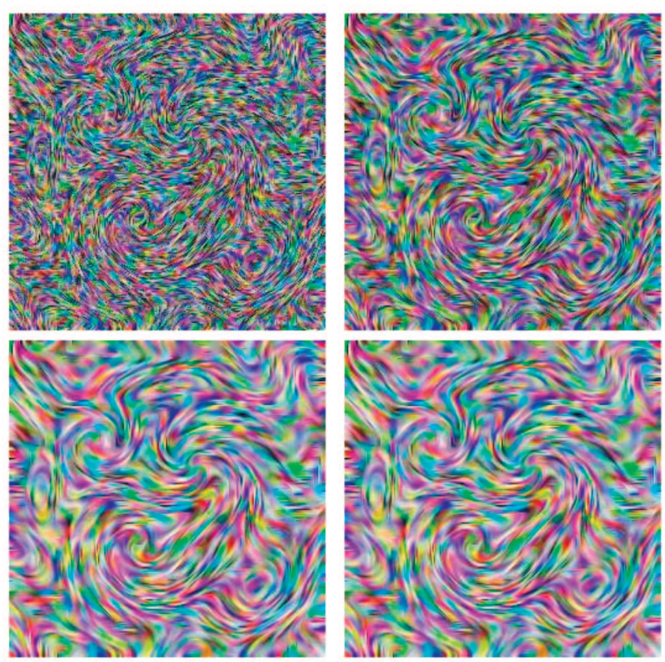


Fig. 14. Using our vector-valued regularization PDE (15), for flow visualization (2).

Note 2: The implementation of the proposed equation, as well as the code source is a part of the **CImg Library**, a powerful and open-source C++ Image Processing Library, located at: http://cimg.sourceforge.net.

REFERENCES

[1] J.F. Abramatic and L.M. Silverman, "Non Linear Restoration of Noisy Images," *IEEE Trans. Patern Analysis and Machine Intelligence,* vol. 4, no. 2, pp. 141-149, 1982.

[2] 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.

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

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

[5] D. Barash, "Bilateral Filtering and Anisotropic Diffusion: Towards a Unified Viewpoint," technical report, HP Laboratories, Israel, 2000.

[6] J. Becker, T. Preusser, and M. Rumpf, "Pde Methods in Flow Simulation Post Processing," *Computing and Visualization in Science,* vol. 3, no. 3, pp. 159-167, 2000.

[7] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro, "Variational Problems and Partial Differential Equations on Implicit Surfaces: The Framework and Examples in Image Processing and Pattern Formation," UCLA research report, June 2000.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," *Proc. SIGGRAPH,* K. Akeley, ed., pp. 417-424, 2000.

[9] P. Blomgren, "Total Variation Methods for Restoration of Vector Valued Images," PhD thesis, Dept. Math., Univ. of California, Los Angeles, June 1998

[10] P. Blomgren and T.F. Chan, "Color TV: Total Variation Methods for Restoration of Vector-Valued Images," *IEEE Trans. Image Processing,* special issue on partial differential equations and geometry-driven diffusion in image processing and analysis, vol. 7, no. 3, pp. 304-309, 1998.

[11] D. Buerkle, T. Preusser, and M. Rumpf, "Transport and Diffusion in Timedependent Flow Visualization," *Proc. IEEE Visualization,* 2001.

[12] T. Chan, S.H. Kang, and J. Shen, "Euler's Elastica and Curvature Based Inpainting," *SIAM J. Applied Math.,* 2002.

[13] T. Chan and J. Shen, "Mathematical Models for Local Deterministic Inpaintings," Technical Report 00-11, Dept. Math., UCLA, Los Angeles, Mar. 2000.

[14] T.F. Chan and J. Shen, "Non-Texture Inpainting by Curvature-Driven Diffusions (CDD)," *J. Visual Comm. and Image Representation,* vol. 12, no. 4, pp. 436-449, 2001.

[15] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic Edge-Preserving Regularization in Computed Imaging," *IEEE Trans. Image Processing,* vol. 6, no. 2, pp. 298-311, 1997.

[16] R. Deriche and O. Faugeras, "Les EDP en Traitement des Images et Vision par Ordinateur," technical report, INRIA, Nov. 1995.

[17] U. Diewald, T. Preusser, and M. Rumpf, "Anisotropic Diffusion in Vector Field Visualization on Euclidian Domains and Surfaces," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 2, pp. 139-149, Mar./Apr. 2000.

[18] B.R. Frieden, "Restoring with Maximum Likelihood and Maximum Entropy," *J. Optical Soc. Am.,* vol. 62, pp. 511-518, 1972.

[19] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, 1984.

[20] G. Gerig, O. Kubler, R. Kikinis, and F. Jolesz, "Nonlinear Anisotropic Filtering of MRI Data," *IEEE Trans. Medical Imaging,* vol. 11, no. 2, pp. 221-231, 1992.

[21] B.R. Hunt, "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer," *IEEE Trans. Computers,* vol. 22, no. 9, pp. 805-812, 1973.

[22] R. Kimmel, R. Malladi, and N. Sochen, "Images as Embedded Maps and Minimal Surfaces: Movies, Color, Texture, and Volumetric Medical Images," *Int'l J. Computer Vision,* vol. 39, no. 2, pp. 111-129, Sept. 2000.

[23] J.J. Koenderink, "The Structure of Images," *Biological Cybernetics,* vol. 50, pp. 363-370, 1984.

[24] 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.

[25] P. Kornprobst, R. Deriche, and G. Aubert, "Image Restoration via PDE's," *Proc. First Ann. Symp. Enabling Technologies for Law Enforcement and Security—SPIE Conf. 2942: Investigative Image Processing,* Nov. 1996.

[26] P. Kornprobst, R. Deriche, and G. Aubert, "Nonlinear Operators in Image Restoration" *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 325-331, 1997.

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

[28] 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, June 1997.

[29] Y. Meyer, "Oscillatory Patterns in Image Processing and Nonlinear Evolution Equations," Univ. Lecture Series, vol. 22, Am. Math. Soc., Providence, R.I., 2001.

[30] M. Nielsen, L. Florack, and R. Deriche, "Regularization, Scale-Space and Edge Detection Filters," *J. Math. Imaging and Vision,* vol. 7, no. 4, pp. 291-308, 1997. Related ECCV:96 article available at: ftp://ftp-sop.inria.fr/robotvis/pub/html/Papers/nielsen-florack-etal:96.ps.gz.

[31] M. Nikolova, "Image Restoration by Minimizing Objective Functions with Nonsmooth Data-Fidelity Terms," *Proc. IEEE Workshop Variational and Level Set Methods,* pp. 11-19, 2001.

[32] T. Papadopoulo and M.I. A. Lourakis, "Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications," Research Report 3961, INRIA Sophia-Antipolis, June 2000.

[33] A. Pardo and G. Sapiro, "Vector Probability Diffusion," *Proc. Int'l Conf. Image Processing,* 2000.

[34] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, July 1990.

[35] G. Sapiro, "Vector-Valued Active Contours," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 680-685, 1996.

[36] G. Sapiro, "Color Snakes," *Computer Vision and Image Understanding,* vol. 68, no. 2, 1997.

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

[38] G. Sapiro and D.L. Ringach, "Anisotropic Diffusion of Multivalued Images with Applications to Color Filtering," *IEEE Trans. Image Processing,* vol. 5, no. 11, pp. 1582-1585, 1996.

[39] J. Shah, "Curve Evolution and Segmentation Functionals: Applications to Color Images," *Proc. Int'l Conf. Image Processing,* 461-464, 1996.

[40] N. Sochen, R. Kimmel, and A.M. Bruckstein, "Diffusions and Confusions in Signal and Image Processing," *J. Math. Imaging and Vision,* vol. 14, no. 3, pp. 195-209, 2001.

[41] B. Tang, G. Sapiro, and V. Caselles, "Direction Diffusion," *Proc. Int'l Conf. Computer Vision,* 1998.

[42] B. Tang, G. Sapiro, and V. Caselles, "Diffusion of General Data on Non-Flat Manifolds via Harmonic Maps Theory: The Direction Diffusion Case," *The Int'l J. Computer Vision,* vol. 36, no. 2, pp. 149-161, Feb. 2000.

[43] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 839-846, Jan. 1998.

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

[45] D. Tschumperlé and R. Deriche, "Constrained and Unconstrained PDE's for Vector Image Restoration," *Proc. 10th Scandinavian Conf. Image Analysis,* Ivar Austvoll, ed., pp. 153-160, 2001.

[46] D. Tschumperlé and R. Deriche, "Diffusion Tensor Regularization with Constraints Preservation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2001.

[47] D. Tschumperlé and R. Deriche, "Regularization of Orthonormal Vector Sets Using Coupled PDE's," *Proc. IEEE Workshop Variational and Level Set Methods,* pp. 3-10, 2001.

[48] D. Tschumperlé and R. Deriche, "Diffusion PDE's on Vector-Valued Images," *IEEE Signal Processing Magazine,* vol. 19, no. 5, pp. 16-25, 2002.

[49] B. Vemuri, Y. Chen, M. Rao, T. McGraw, T. Mareci, and Z. Wang, "Fiber Tract Mapping from Diffusion Tensor MRI," *Proc. First IEEE Workshop Variational and Level Set Methods in Computer Vision,* 2001.
[50] J. Weickert, "Anisotropic Diffusion in Image Processing," PhD thesis, Univ. of Kaiserslautern, Germany, Laboratory of Technomath., Jan. 1996.
[51] J. Weickert, *Anisotropic Diffusion in Image Processing.* Stuttgart: Teubner-Verlag, 1998.
[52] P. Wesseling, *Principles of Computational Fluid Dynamics.* Springer-Verlag, 2000.
[53] A.P. Witkin, "Scale-Space Filtering," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 1019-1021, 1983.
[54] A. Yezzi, "Modified Curvature Motion for Image Smoothing and Enhancement," *IEEE Trans. Image Processing,* pp 345-352, vol. 7, no. 3, Mar. 1998.
[55] S. Di Zenzo, "A Note on the Gradient of a Multi-Image," *Computer Vision, Graphics, and Image Processing,* vol. 33, pp. 116-125, 1986.

David Tschumperlé gradutated from Ecole Supérieure en Sciences Informatiques (ESSI), Sophia Antipolis in 1999, and received the PhD degree in computer sciences from the University of Nice in 2002. He is currently a CNRS researcher in the IMAGE group of the GREYC Lab, in Caen/France. He is mainly interested in variational tools and PDEs to solve computer vision and image processing problems, particularly related to multidimensionnal data and medical images. He is the author of the CImg Library (http://cimg.sourceforge.net), a powerful, open-source and multiplateform image processing library.

Rachid Deriche graduated from Ecole Nationale Supérieure des Télécommunications, Paris, in 1979, and received the PhD degree in mathematics from the University of Paris IX, Dauphine, in 1982. He is currently a research director at INRIA Sophia-Antipolis in the Computer and Biological Vision Group. His research interests are in image processing, computer and biological vision and include, in particular, the area related to variational methods and partial differential equations for vision. More generally, he is very interested in the applications of mathematics to computer vision and image processing. He has authored and coauthored more than 100 scientific papers. To find out more about his research and some selected publications, go to: http://www-sop.inria.fr/odyssee/team/Rachid.Deriche/.

I. Appendix A

The Euler-Lagrange equations corresponding to the functional (5) are :

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

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 :

$$\begin{pmatrix} \dfrac{\partial \psi}{\partial I_{i_x}} \\ \\ \dfrac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \lambda_+}{\partial I_{i_x}} & \dfrac{\partial \lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_+}{\partial I_{i_y}} & \dfrac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} \begin{pmatrix} \dfrac{\partial \psi}{\partial \lambda_+} \\ \\ \dfrac{\partial \psi}{\partial \lambda_-} \end{pmatrix} \tag{2}$$

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} :

$$\frac{\partial \lambda_\pm}{\partial I_{i_x}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_x}} \qquad \text{and} \qquad \frac{\partial \lambda_\pm}{\partial I_{i_y}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_y}} \tag{3}$$

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

$$\begin{cases} \dfrac{\partial g_{11}}{\partial I_{i_x}} = 2 I_{i_x} \\ \\ \dfrac{\partial g_{11}}{\partial I_{i_y}} = 0 \end{cases} \quad \text{and} \quad \begin{cases} \dfrac{\partial g_{12}}{\partial I_{i_x}} = I_{i_y} \\ \\ \dfrac{\partial g_{12}}{\partial I_{i_y}} = I_{i_x} \end{cases} \quad \text{and} \quad \begin{cases} \dfrac{\partial g_{22}}{\partial I_{i_x}} = 0 \\ \\ \dfrac{\partial g_{22}}{\partial I_{i_y}} = 2 I_{i_y} \end{cases}$$

i.e (18) can be written as :

$$\begin{pmatrix} \dfrac{\partial \lambda_\pm}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} 2\dfrac{\partial \lambda_\pm}{\partial g_{11}} & \dfrac{\partial \lambda_\pm}{\partial g_{12}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial g_{12}} & 2\dfrac{\partial \lambda_\pm}{\partial g_{22}} \end{pmatrix} \nabla I_i \tag{4}$$

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 :

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial g_{kl}} &= \frac{\partial \lambda_+}{\partial g_{kl}} \, \theta_+ \theta_+^T + \frac{\partial \lambda_-}{\partial g_{kl}} \, \theta_- \theta_-^T \\ &+ \lambda_+ \, \frac{\partial \theta_+}{\partial g_{kl}} \theta_+^T + \lambda_- \, \frac{\partial \theta_-}{\partial g_{kl}} \theta_-^T \\ &+ \lambda_+ \, \theta_+ \frac{\partial \theta_+^T}{\partial g_{kl}} + \lambda_- \, \theta_- \frac{\partial \theta_-^T}{\partial g_{kl}} \end{aligned} \tag{5}$$

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

$$\begin{cases} \theta_+^T\theta_+ = \theta_-^T\theta_- = 1 \\ \\ \theta_+^T\theta_- = \theta_-^T\theta_+ = 0 \end{cases} \qquad \text{and} \qquad \begin{cases} \dfrac{\partial\theta_+^T}{\partial g_{kl}}\theta_+ = \theta_+^T\dfrac{\partial\theta_+}{\partial g_{kl}} = 0 \\ \\ \dfrac{\partial\theta_-^T}{\partial g_{kl}}\theta_- = \theta_-^T\dfrac{\partial\theta_-}{\partial g_{kl}} = 0 \end{cases} \tag{6}$$

We first multiply the equation (5) by $\theta_\pm^T$ at the left, by $\theta_\pm$ at the right, then use the properties (6). It allows high simplifications, and leads to these two relations :

$$\frac{\partial\lambda_+}{\partial g_{kl}} = \theta_+^T\frac{\partial\mathbf{G}}{\partial g_{kl}}\theta_+ \qquad \text{and} \qquad \frac{\partial\lambda_-}{\partial g_{kl}} = \theta_-^T\frac{\partial\mathbf{G}}{\partial g_{kl}}\theta_- \tag{7}$$

Equations (7) 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 [32] 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 :

$$\frac{\partial\mathbf{G}}{\partial g_{11}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad , \quad \frac{\partial\mathbf{G}}{\partial g_{12}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial\mathbf{G}}{\partial g_{22}} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

With all these elements, we can express (4) as :

$$\begin{pmatrix} \dfrac{\partial\lambda_+}{\partial I_{i_x}} \\ \\ \dfrac{\partial\lambda_+}{\partial I_{i_y}} \end{pmatrix} = 2\ \theta_+\theta_+^T\nabla I_i \qquad \text{and} \qquad \begin{pmatrix} \dfrac{\partial\lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial\lambda_-}{\partial I_{i_y}} \end{pmatrix} = 2\ \theta_-\theta_-^T\nabla I_i \tag{8}$$

Finally, replacing (8) in the Euler-Lagrange equations (2) and (1), gives the vector-valued gradient descent of the functional (5) :

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} \int_\Omega \psi(\lambda_+,\lambda_-)\ d\Omega \qquad \Longrightarrow \qquad \frac{\partial I_i}{\partial t} = 2\ \mathrm{div}\left(\left[\frac{\partial\psi}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\psi}{\partial\lambda_-}\theta_-\theta_-^T\right]\nabla I_i\right) \tag{9}$$

(for $i = 1..n$) $\square$

Note that (9) is a divergence-based equation such that :

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

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

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

associated to these corresponding orthonormal eigenvectors :

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

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 :

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

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.

II. Appendix B

Most divergence-based regularization PDE's acting on multivalued images have the following form :

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

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 (10) 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* $\mathbf{D}$:

$$\operatorname{div}\left(\mathbf{D}\nabla I_i\right) = \operatorname{trace}\left(\mathbf{D}\mathbf{H}_{I_i}\right) + \nabla I_i^T \ \vec{\mathbf{div}}\left(\mathbf{D}\right) \tag{11}$$

where $\vec{\mathbf{div}}\left(\right)$ 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

$$\mathbf{D} = f_1(\lambda_+,\lambda_-)\ \theta_+\theta_+^T + f_2(\lambda_+,\lambda_-)\ \theta_-\theta_-^T \quad \text{with} \quad f_{1/2}:\mathbb{R}^2\rightarrow\mathbb{R} \tag{12}$$

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 :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\left(\mathbf{D}\nabla I_i\right) \quad \text{with} \quad (12) \quad \text{and} \quad \left\{ \begin{array}{rcl} f_1(\lambda_+,\lambda_-) & = & 2\dfrac{\partial\psi}{\partial\lambda_+} \\ \\ f_2(\lambda_+,\lambda_-) & = & 2\dfrac{\partial\psi}{\partial\lambda_-} \end{array}\right.$$

In order to develop the additional diffusion term $\nabla I_i^T \vec{\mathbf{div}}\left(\mathbf{D}\right)$ in the equation (11), we propose to write $\mathbf{D}$ as a linear combination of $\mathbf{G}$ and $\mathbf{Id}$:

$$\mathbf{D} = \alpha(\lambda_+,\lambda_-)\mathbf{G} + \beta(\lambda_+,\lambda_-)\mathbf{Id} \tag{13}$$

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

$$\alpha = \frac{f_1(\lambda_+,\lambda_-) - f_2(\lambda_+,\lambda_-)}{\lambda_+ - \lambda_-} \qquad \text{and} \qquad \beta = \frac{\lambda_+ f_2(\lambda_+,\lambda_-) - \lambda_- f_1(\lambda_+,\lambda_-)}{\lambda_+ - \lambda_-} \tag{14}$$

Indeed, we have

$$\begin{aligned}
\alpha\mathbf{G}+\beta\mathbf{Id} &= \frac{f_1-f_2}{\lambda_+-\lambda_-}\left(\lambda_+\ \theta_+\theta_+^T+\lambda_-\ \theta_-\theta_-^T\right)+\frac{\lambda_+f_2-\lambda_-f_1}{\lambda_+-\lambda_-}\left(\theta_+\theta_+^T+\theta_-\theta_-^T\right)\\
&= \frac{1}{\lambda_+-\lambda_-}\left[\theta_+\theta_+^T\left(\lambda_+f_1-\lambda_-f_1\right)+\theta_-\theta_-^T\left(\lambda_+f_2-\lambda_-f_2\right)\right]\\
&= f_1\ \theta_+\theta_+^T+f_2\ \theta_-\theta_-^T\\
&= \mathbf{D} \qquad\qquad \square
\end{aligned}$$

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 (11), 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 $\vec{\mathbf{div}}\,(\mathbf{D})$ into :

$$\vec{\mathbf{div}}\,(\mathbf{D})=\alpha\vec{\mathbf{div}}\,(\mathbf{G})+\mathbf{G}\nabla\alpha+\nabla\beta \tag{15}$$

and the additional term of the equation (11) would be computed as :

$$\begin{aligned}
\nabla I^T\vec{\mathbf{div}}\,(\mathbf{D}) &= \mathrm{trace}\left(\vec{\mathbf{div}}\,(\mathbf{D})\,\nabla I_i^T\right)\\
&= \alpha\mathrm{trace}\left(\vec{\mathbf{div}}\,(\mathbf{G})\,\nabla I_i^T\right) && (16)\\
&+ \mathrm{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) && (17)\\
&+ \mathrm{trace}\left(\nabla\beta\nabla I_i^T\right) && (18)
\end{aligned}$$

In the following, we propose to find formal expressions of (16), (17) and (18).

- 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 :

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

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 16 as :

$$\alpha\mathrm{trace}\left(\vec{\mathbf{div}}\,(\mathbf{G})\,\nabla I_i{}^T\right)=\sum_{j=1}^{n}\alpha\mathrm{trace}\left(\mathbf{H}_j\left[\nabla I_i{}^T\nabla I_j\mathbf{Id}+\nabla I_j\nabla I_i{}^T\right]\right) \tag{19}$$

$\square$

• We finally have to compute $\nabla\alpha$ and $\nabla\beta$, in the expression (17) and (18). This can be done by the decomposition :

$$\nabla\alpha = \frac{\partial\alpha}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\alpha}{\partial\lambda_-}\nabla\lambda_- \qquad \text{and} \qquad \nabla\beta = \frac{\partial\beta}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\beta}{\partial\lambda_-}\nabla\lambda_- \tag{20}$$

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

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

In Appendix A, we derivated 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,

$$\nabla\lambda_\pm = \sum_{j=1}^{n} 2\mathbf{H}_j\theta_\pm\theta_\pm^T\nabla I_j \tag{21}$$

We can replace (21) into the expressions of (20), in order to find the spatial gradients of α and β :

$$\left\{\begin{aligned} \nabla\alpha &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\alpha}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j \\ \nabla\beta &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\beta}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j \end{aligned}\right. \tag{22}$$

Using (22), we finally compute the two missing parts (17) and (18) of the additional term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$:

$$\left\{\begin{aligned} \operatorname{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) &= \sum_{j=1}^{n} \operatorname{trace}\left(2\ \mathbf{G}\mathbf{H}_j\left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\alpha}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \\ \operatorname{trace}\left(\nabla\beta\nabla I_i^T\right) &= \sum_{j=1}^{n} \operatorname{trace}\left(2\ \mathbf{H}_j\left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\beta}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \end{aligned}\right. \tag{23}$$

□

• The final step consists in putting together the equations (19) and (23), in order to express the additional term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$ in the PDE (11).

$$\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D}) = \sum_{j=1}^{n} \operatorname{trace}\left(\mathbf{H}_j\mathbf{P}^{ij}\right) \tag{24}$$

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

$$\begin{aligned} \mathbf{P}^{ij} &= \alpha \, \nabla I_i^T \nabla I_j \mathbf{Id} \\ &+ 2 \left(\frac{\partial \alpha}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \alpha}{\partial \lambda_-} \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \mathbf{G} \\ &+ 2 \left((\alpha + \frac{\partial \beta}{\partial \lambda_+}) \theta_+ \theta_+^T + (\alpha + \frac{\partial \beta}{\partial \lambda_-}) \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \end{aligned} \qquad (25)$$

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 (24) 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 :

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

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 $\mathbf{P}^{ij}$:

$$\mathbf{Q}^{ij} = \frac{\mathbf{P}^{ij} + \mathbf{P}^{ij^T}}{2} \qquad (26)$$

and we have :

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

Finally, the divergence-based PDE (11) can be written as :

$$\operatorname{div}(\mathbf{D} \nabla I_i) = \sum_{j=1}^{n} \operatorname{trace}\left((\delta_{ij} \mathbf{D} + \mathbf{Q}^{ij}) \mathbf{H}_j \right) \qquad (27)$$

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

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

$\square$

The regularization PDE (27) is equivalent to the divergence-based equation $\frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i)$, but with a trace-based formulation.