INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications

David Tschumperlé — Rachid Deriche

N° 4657

Décembre 2002

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4657--FR+ENG

Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications

David Tschumperlé , Rachid Deriche

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Odyssee

Rapport de recherche n° 4657 — Décembre 2002 — 33 pages

Abstract: This report addresses the problem of vector-valued image regularization with variational methods and PDE's. From the study of existing global and local formalisms, we propose a new framework that unifies a large number of previous methods within a generic local formulation. On one hand, resulting equations are more adapted to analyze the local geometric behaviors of the diffusion processes. On the other hand, it can be used to design a new regularization PDE that takes important local smoothing properties into account. Specific numerical schemes are also naturally emerging from this formulation. Finally, we illustrate the capability of our approach to deal with classical image processing applications, such as color image restoration, inpainting, magnification and flow visualization.

Key-words: Diffusion PDE's, multivalued image regularization, local geometry of images

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications

Résumé : Ce rapport étudie le problème de la régularisation d'images vectorielles par des méthodes variationnelles et EDP. A partir de l'étude des formalismes locaux et globaux qui existent déjà pour traiter ce type de problème, nous proposons une nouvelle vision unificatrice des EDP de lissage, qui permet d'exprimer avec un formalisme commun la plupart des équations de régularisation existantes. D'une part, les équations résultantes sont mieux adaptées pour analyser le comportement géométrique local des processus de diffusion. D'autre part, ce formalisme peut être utilisé pour concevoir une nouvelle EDP de régularisation vectorielle, qui possède des propriétés de lissage locales intéressantes. De plus, des schémas numériques spécifiques émergent naturellement de ce nouveau formalisme. Finalement, nous illustrons les différentes possibilités offerte par notre nouvelle approche, pour l'application à de nombreux problèmes de traitement d'images faisant intervenir des processus de lissage, comme par exemple la restauration d'images couleurs, le 'Inpainting' (remplissage de trous dans l'image), l'interpolation, et la visualisation de flots.

Mots-clés : EDP de diffusion, regularization d'images multivaluées, géometrie locale des images

1 Introduction & Motivation

In the late 80's, anisotropic *regularization PDE's* have raised a strong interest in the field of image processing. The ability to smooth data while preserving large global features such as contours and corners (discontinuities of the signal), has opened new ways to handle classical image processing issues (restoration, segmentation, registration, etc.). Thus, many regularization schemes have been presented so far in the literature, particularly for the case of *2D scalar images* $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ ([1, 18, 19, 28] and references therein).
Extensions of these algorithms to *vector-valued images* $\mathbf{I} : \Omega \to \mathbb{R}^n$ have been recently proposed, leading to more elaborated diffusion PDE's : a *coupling between image channels* 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

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

(also called *structure tensor* [26, 27, 28, 30]). The $\lambda_\pm$ respectively define the local min/max vector-valued variations of $\mathbf{I}$ in corresponding spatial directions $\theta_\pm$, i.e. the local geometry of the image discontinuities. (note that $\lambda_+ = \|\nabla I\|$ and $\theta_+ = \nabla I / \|\nabla I\|$ *for scalar images*, $n = 1$).
Proposed regularization schemes generally lie on one of these three following approaches, related to different interpretation levels :

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 variation will flatten the image, then remove the noise gradually :

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

   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. Then, the minimization is performed through a gradient descent (PDE), coming from the Euler-Lagrange equations of $E(\mathbf{I})$. Corresponding references for vector-valued images can be found in [5, 12, 17, 19, 21, 23, 27],

2. **Divergence expressions** : A regularization process may be also designed more locally, as the diffusion of pixel values, viewed as chemical concentrations

[28, 11] and driven by a 2×2 *diffusion tensor* $\mathbf{D}$ (symmetric and definite-positive matrix) :

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

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 in this paper that the interpretation of the PDE (2) in terms of local smoothing is not so obvious.

3. **Oriented Laplacians** : 2D image regularization may be finally seen as the juxtaposition of two *oriented 1D heat flows*, leading to $1D$ gaussian smoothing processes along orthonormal directions $\mathbf{u}\perp\mathbf{v}$, with different weights c_1 and c_2 [14, 20, 26, 27] :

$$\frac{\partial \mathbf{I}}{\partial t} = c_1 \, \frac{\partial^2 \mathbf{I}}{\partial \mathbf{u}^2} + c_2 \, \frac{\partial^2 \mathbf{I}}{\partial \mathbf{v}^2} = c_1 \; I_{\mathbf{uu}} + c_2 \; I_{\mathbf{vv}} \tag{3}$$

Like divergence expressions, the smoothing weights c_1, c_2 and directions $\mathbf{u}, \mathbf{v}$ 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.

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 ($n = 1$). In this case, the three following approaches are equivalent :

$$\begin{array}{lll} & (1) & \displaystyle\min_{I:\Omega\to\mathbb{R}} \int_\Omega \phi(\|\nabla I\|)\, d\Omega \\ \Rightarrow & (2) & \displaystyle\frac{\partial I}{\partial t} = \operatorname{div}\left(\frac{\phi^{'}(\|\nabla I\|)}{\|\nabla I\|}\;\nabla I\right) \\ \Rightarrow & (3) & \displaystyle\frac{\partial I}{\partial t} = \frac{\phi^{'}(\|\nabla I\|)}{\|\nabla I\|}\; I_{\xi\xi} + \phi^{''}(\|\nabla I\|)\; I_{\eta\eta} \end{array} \tag{4}$$

where $\eta = \nabla I / \|\nabla I\|$ and $\xi \perp \eta$. Note that this regularization 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 equivalences for the more general case of *vector-valued regularization*. We tackle each of these three interpretation levels (1),(2),(3) in its more general form, 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 PDE's. Thus, it allow 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 5). Finally, we apply our method to solve a wide range of image processing issues, including color image restoration, inpainting, magnification, and flow visualization (section 6).

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 (1)⇒(2).

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

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

  where the $\lambda_\pm$ are the eigenvalues of the structure tensor $\mathbf{G} = \sum_{j=1}^n \nabla I_j \nabla I_j^T$, and $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is an increasing function. This is a natural and generic extension for vector-valued images, of the ϕ-function formulation (4).

- **Gradient descent** : The Euler-Lagrange equations of (5) can be derived, and reduce to a simple form of divergence-based expression (see Appendix A) :

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

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

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

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 $\mathbf{G}$.

- **Link with other approaches** : The choice of particular cases of functions ψ leads to previous vector-valued regularization approaches defined as variational methods, such as the whole range of Vector ϕ-functionals [17, 23] :

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

or the Beltrami flow framework [12] :

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

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

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 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 (2)$\Rightarrow$(3). The motivation is that oriented Laplacians are particularly well designed to understand geometrically 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 $\mathbf{u} \perp \mathbf{v}$, this PDE can be equivalently written as :

$$\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}\mathbf{H}_i) \qquad (i = 1..n) \tag{7}$$

where $\mathbf{H}_i$ is the *Hessian matrix of the vector component* I_i and $\mathbf{T}$ is the 2×2 tensor defined as : $\mathbf{T} = c_1 \mathbf{u}\mathbf{u}^T + c_2 \mathbf{v}\mathbf{v}^T$, characterized by its two eigenvalues c_1, c_2 and its two corresponding eigenvectors $\mathbf{u}, \mathbf{v}$. Suppose that $\mathbf{T}$ is a constant tensor over the definition domain Ω. Then, it can be shown [25] that the formal solution of the PDE (7) is :

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

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

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

It is a generalization of the Koenderink's idea [13], who proved this property 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.1 illustrates two gaussian kernels $G^{(\mathbf{T},t)}(x, y)$ obtained respectively with isotropic and anisotropic tensors $\mathbf{T}$ *(left)* and the corresponding evolutions of the diffusion PDE (7) on a color image *(right)*. It is worth to notice that the gaussian kernels $\mathbf{G}^{(\mathbf{T},t)}$ give *exactly* the classical representations of the tensors $\mathbf{T}$ with ellipsoids. Conversely, it is clear that the tensors $\mathbf{T}$ represent the exact smoothing performed by the PDE (7).

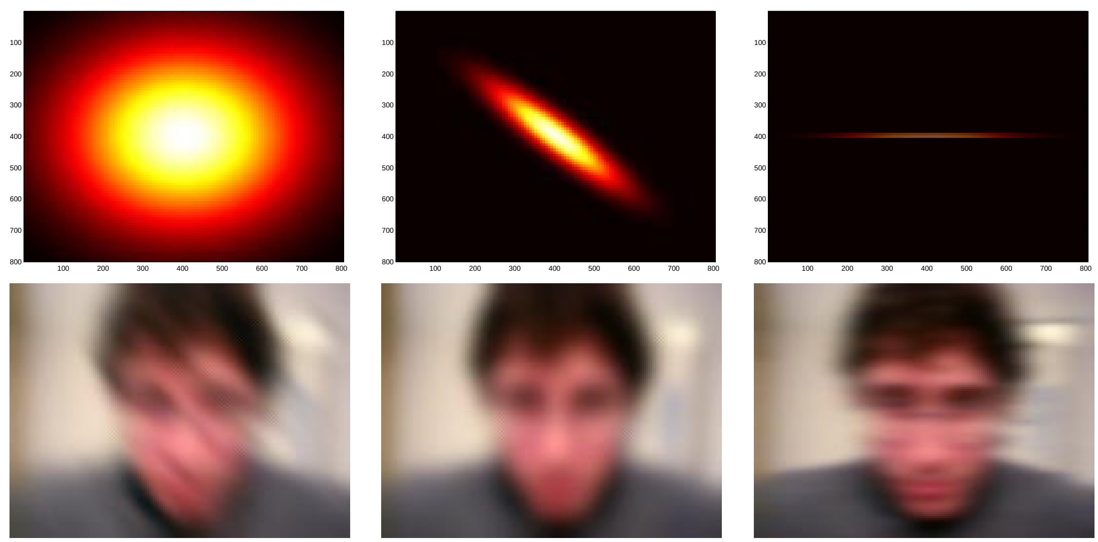


Figure 1: Trace-based PDE's (7) viewed as convolutions by oriented Gaussians.

When $\mathbf{T}$ is not constant (which is generally the case), i.e. represents a field $\Omega \rightarrow P(2)$ of varying 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.2 illustrates two examples of spatially varying tensor fields $\mathbf{T}$, represented with fields of ellipsoids *(left)*, and the corresponding evolutions of (7) on a color image *(right)*. It particularly shows that *the shape of each tensor* $\mathbf{T}$ *is exactly related to the local smoothing behavior* performed pointwise by the trace-based PDE (7).

Note that this local filtering concept makes the link between a generic form of vector-valued diffusion PDE's (7) and *Bilateral filtering* techniques, as described in [2, 24]. Another similar approach based on non-Gaussian convolution kernels has been also proposed for the specific case of Beltrami Flow [22].

Figure 2: Trace-based PDE's (7) with non-constant diffusion tensor fields $\mathbf{T}$.

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 :

$$\mathrm{div}\left(\mathbf{D}\nabla I_i\right) = \mathrm{trace}\left(\mathbf{D}\mathbf{H}_i\right) + \nabla I_i^T \ \vec{\mathbf{div}}\left(\mathbf{D}\right)$$

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

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

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 $\operatorname{trace}(\mathbf{D}\mathbf{H}_i)$, 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 from the spectral elements of $\mathbf{D}$. This is actually the case for the scalar ϕ-function formulation (4), where the smoothing process doesn't 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 $\mathbf{D}$ depends only on the spectral elements of the structure tensor $\mathbf{G}$, such as :

$$\mathbf{D} = f_1(\lambda_+, \lambda_-)\theta_+\theta_+^T + f_2(\lambda_+, \lambda_-)\theta_-\theta_-^T \tag{9}$$

with $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$, (which is the case for 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 (details in Appendix B) :

$$\operatorname{div}(\mathbf{D}\nabla I_i) = \sum_{j=1}^{n} \operatorname{trace}\left((\delta_{ij}\mathbf{D} + \mathbf{Q}^{ij})\mathbf{H}_j\right) \tag{10}$$

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

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

with

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

This development (10) expresses 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 diffusion contributions* that have a simple geometric interpretation in term 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* :

$$\frac{\partial I_i}{\partial t} = \sum_{j=1}^{n} \operatorname{trace}\left(\mathbf{A}^{ij}\mathbf{H}_i\right) \qquad (i = 1..n) \tag{11}$$

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 :

$$\frac{\partial \mathbf{I}}{\partial t} = \vec{\mathbf{trace}}\left(\mathcal{A}\mathcal{H}\right) \tag{12}$$

where $\mathcal{A}$ is the *matrix of diffusion tensors* $\mathbf{A}^{ij}$ (and is itself considered as *symmetric*), and $\mathcal{H}$ is the *vector of Hessian matrices* $\mathbf{H}_j$. The matrix product $\mathcal{A}\mathcal{H}$ in (12) is then seen *sub-matrix per sub-matrix*, and the operator $\vec{\mathbf{trace}}\,()$ returns the vector in $\mathbb{R}^n$, corresponding to the trace of each sub-matrix in the resulting vector of matrices.

4.1 Link with previous expressions

The PDE (11) 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 $\frac{\partial I_i}{\partial t} = \operatorname{div}\left(\mathbf{D}\nabla I_i\right)$, as developed in section 2) into a very local formulation. This particularly includes the works done in [5, 11, 12, 17, 19, 21, 23, 27, 28] 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 *diffusion energy transfer* can be considered as a particular *coupling* of the corresponding vector-valued diffusion PDE.

- Second, the PDE (11) also gathers the oriented-Laplacian formulations $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}\mathbf{H}_i)$, by choosing $A^{ij} = \delta_{ij}\mathbf{T}$. In this case, the matrix $\mathcal{A}$ 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 [14, 20, 26, 27].

4.2 A new regularization PDE

We propose now to design a new vector-valued regularization PDE that follows desired local geometric properties. These properties will naturally define a specific form of regularization PDE, from the very generic form (11) :

- We don't 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 that 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* therein, i.e. a 2D heat flow that smooth the noise efficiently with no-preferred directions : $\frac{\partial I_i}{\partial t} \simeq \Delta I_i = \mathrm{trace}\,(\mathbf{H}_i)$. It means that the tensor $\mathbf{A}$ must be *isotropic* in these regions :

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

- 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 : $\frac{\partial I_i}{\partial t} = \mathrm{trace}\left(\beta\theta_-{\theta_-}^T\mathbf{H}_i\right)$, where β is a function decreasing anyway for very high variations, avoiding sharp corners oversmoothing. This means that the tensor $\mathbf{A}$ must be *anisotropic* in these regions :

$$\lim_{(\lambda_+ + \lambda_-)\to 0} \mathbf{A} = \beta\theta_-\theta_-^T$$

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

$$\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{T}\mathbf{H}_i) \qquad (i = 1..n) \tag{13}$$

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 to retrieve a more coherent vector-geometry that gives a better approximation of the vector discontinuities directions (see also [28]). For our experiments in section 6, we chose

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

This is of course one possible choice (inspired from the *hyper-surface formulation* of the scalar case [1]) 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 process performed by (13). 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.

5 Numerical schemes

The numerical implementation of the PDE (13) can be done with classical numerical schemes, based on spatial discretizations with centered finite differences of the gradients and the Hessians [15]. 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 mask $\mathbf{G}^{(\mathbf{T},t)}$ corresponding to the tensor $\mathbf{T}$, defined by (13). Then, we apply it on each local neighborhood $I_i(x, y)$:

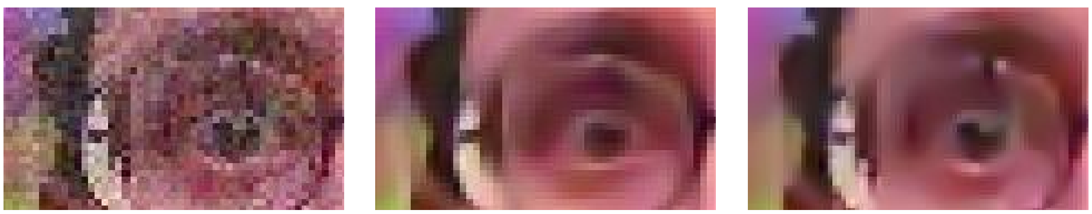
$$[\mathrm{Trace(TH)}](x,y) = \left(\begin{array}{|c|c|c|}\hline i(x-1,y-1) & i(x,y-1) & i(x+1,y) \\ \hline i(x-1,y) & i(x,y) & i(x+1,y) \\ \hline i(x-1,y+1) & i(x,y+1) & i(x+1,y+1) \\ \hline \end{array} \ * \ \begin{array}{|c|c|c|}\hline G(-1,-1) & G(0,-1) & G(1,-1) \\ \hline G(-1,0) & G(0,0) & G(1,0) \\ \hline G(-1,1) & G(0,1) & G(1,1) \\ \hline \end{array} \right)(0,0)$$

Main advantages of this numerical scheme are :

1. It 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.3).

As for 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.



(a) Noisy image ()

(b) Scheme using Hessian discretizations

(c) Scheme using local filtering techniques

Figure 3: Comparisons of numerical schemes.

6 Applications

We illustrate here the wide range of image processing related applications that can be handled by our presented approach, through our vector-valued regularization PDE (13) :

- **Color image restoration** : Despite the apparition of digital cameras, color image restoration may be still needed. Fig.4 represents a digital photograph with *real noise*, due to the bad lightning conditions during the snapshot. Our vector-valued regularization PDE can successfully remove the noise, while preserving the global features of the image.

- **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 (13) significantly improves the quality of such degraded images (Fig.5).

- **Color image inpainting** : Recently, an interesting application of diffusion PDE's named *image inpainting*, has been proposed in [4, 7, 8, 9]. It consists in filling undesired holes (defined by 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 (13) 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.6), removing real objects in photographs (Fig.7) or reconstruct partially coded images for image compression purposes (Fig.8).

- **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 (13) on the image (excepted on the original known pixels), we can retrieve non-linear magnified images without jagging or bloc effects, inherent to classical linear interpolation techniques (Fig.9).

- **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.10a), 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 (13) but where $\mathbf{T}$ is directed by the directions of $\mathcal{F}$, instead of the local geometry of $\mathbf{I}$:

$$\frac{\partial I_i}{\partial t} = \text{trace}\left(\left[\frac{1}{\|\mathcal{F}\|}\mathcal{F}\mathcal{F}^T\right]\mathbf{H}_i\right) \qquad (i = 1..n) \tag{14}$$

  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.11). Here, our used regularization equation (14) *ensures that the smoothing of the pixels is done exactly in the direction of the flow* $\mathcal{F}$. This is not the case in [3, 6, 10], 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.

Conclusion & 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 PDE's. 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.

7 Appendix A

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

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\begin{pmatrix} \dfrac{\partial \psi}{\partial I_{i_x}} \\ \\ \dfrac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} \qquad\qquad (i = 1..n) \tag{15}$$

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 :

$$\begin{pmatrix} \dfrac{\partial \psi}{\partial I_{i_x}} \\ \\ \dfrac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \lambda_+}{\partial I_{i_x}} & \dfrac{\partial \lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_+}{\partial I_{i_y}} & \dfrac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} \begin{pmatrix} \dfrac{\partial \psi}{\partial \lambda_+} \\ \\ \dfrac{\partial \psi}{\partial \lambda_-} \end{pmatrix} \tag{16}$$

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} :

$$\frac{\partial \lambda_\pm}{\partial I_{i_x}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_x}} \qquad \text{and} \qquad \frac{\partial \lambda_\pm}{\partial I_{i_y}} = \sum_{k,l} \frac{\partial \lambda_\pm}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_y}} \tag{17}$$

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

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

i.e (17) can be written as :

$$\begin{pmatrix} \dfrac{\partial \lambda_\pm}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} 2\dfrac{\partial \lambda_\pm}{\partial g_{11}} & \dfrac{\partial \lambda_\pm}{\partial g_{12}} \\ \\ \dfrac{\partial \lambda_\pm}{\partial g_{12}} & 2\dfrac{\partial \lambda_\pm}{\partial g_{22}} \end{pmatrix} \nabla I_i \tag{18}$$

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 :

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial g_{kl}} &= \frac{\partial \lambda_+}{\partial g_{kl}} \; \theta_+ \theta_+^T + \frac{\partial \lambda_-}{\partial g_{kl}} \; \theta_- \theta_-^T \\ &+ \lambda_+ \; \frac{\partial \theta_+}{\partial g_{kl}} \theta_+^T + \lambda_- \; \frac{\partial \theta_-}{\partial g_{kl}} \theta_-^T \\ &+ \lambda_+ \; \theta_+ \frac{\partial \theta_+^T}{\partial g_{kl}} + \lambda_- \; \theta_- \frac{\partial \theta_-^T}{\partial g_{kl}} \end{aligned} \tag{19}$$

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

$$\left\{ \begin{array}{l} \theta_+^T \theta_+ = \theta_-^T \theta_- = 1 \\ \\ \theta_+^T \theta_- = \theta_-^T \theta_+ = 0 \end{array} \right. \qquad \text{and} \qquad \left\{ \begin{array}{l} \dfrac{\partial \theta_+^T}{\partial g_{kl}} \theta_+ = \theta_+^T \dfrac{\partial \theta_+}{\partial g_{kl}} = 0 \\ \\ \dfrac{\partial \theta_-^T}{\partial g_{kl}} \theta_- = \theta_-^T \dfrac{\partial \theta_-}{\partial g_{kl}} = 0 \end{array} \right. \tag{20}$$

We first multiply the equation (19) by $\theta_\pm^T$ at the left, by $\theta_\pm$ at the right, then use the properties (20). It allows high simplifications, and leads to these two relations :

$$\frac{\partial \lambda_+}{\partial g_{kl}} = \theta_+^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_+ \qquad \text{and} \qquad \frac{\partial \lambda_-}{\partial g_{kl}} = \theta_-^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_- \tag{21}$$

Equations (21) 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 [16] 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 :

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

With all these elements, we can express (18) as :

$$\begin{pmatrix} \dfrac{\partial \lambda_+}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_+}{\partial I_{i_y}} \end{pmatrix} = 2\ \theta_+\theta_+^T \nabla I_i \qquad \text{and} \qquad \begin{pmatrix} \dfrac{\partial \lambda_-}{\partial I_{i_x}} \\ \\ \dfrac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} = 2\ \theta_-\theta_-^T \nabla I_i \tag{22}$$

Finally, replacing (22) in the Euler-Lagrange equations (16) and (15), gives the vector-valued gradient descent of the functional (5) :

$$\min_{\mathbf{I}:\Omega\to\mathbb{R}^n} \int_\Omega \psi(\lambda_+, \lambda_-)\, d\Omega \qquad \Longrightarrow \qquad \frac{\partial I_i}{\partial t} = 2\ \mathrm{div}\left(\left[\frac{\partial \psi}{\partial \lambda_+} \theta_+\theta_+^T + \frac{\partial \psi}{\partial \lambda_-} \theta_-\theta_-^T \right] \nabla I_i \right) \tag{23}$$

(for $i = 1..n$) □

Note that (23) is a divergence-based equation such that :

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

$\mathbf{D} \in \mathrm{P}(2)$ 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 :

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

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.

8 Appendix B

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

$$\frac{\partial I_i}{\partial t} = \mathrm{div}\left(\mathbf{D}\nabla I_i\right) \qquad\qquad (i = 1..n) \tag{24}$$

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 $\mathrm{div}\,() = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$ is itself a first order derivative operator, we should be able to write (24) 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 $\frac{\partial I_i}{\partial t} = \mathrm{trace}\,(\mathbf{D}\mathbf{H}_i)$.

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* :

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

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* $\mathbf{D}$:

$$\mathrm{div}\left(\mathbf{D}\nabla I_i\right) = \mathrm{trace}\left(\mathbf{D}\mathbf{H}_{I_i}\right) + \nabla I_i^T\ \vec{\mathbf{div}}\left(\mathbf{D}\right) \tag{25}$$

where $\vec{\mathbf{div}}\,()$ is the *matrix divergence*.
A natural idea is then to decompose the additional term $\nabla I_i^T\,\vec{\mathbf{div}}\,(\mathbf{D})$ 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

$$\mathbf{D} = f_1(\lambda_+, \lambda_-)\;\theta_+\theta_+^T + f_2(\lambda_+, \lambda_-)\;\theta_-\theta_-^T \quad \text{with} \quad f_{1/2} : \mathbb{R}^2 \to \mathbb{R} \tag{26}$$

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 :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}\left(\mathbf{D}\nabla I_i\right) \qquad \text{with} \quad (26) \quad \text{and} \quad \left\{ \begin{array}{rcl} f_1(\lambda_+, \lambda_-) & = & 2\dfrac{\partial \psi}{\partial \lambda_+} \\ \\ f_2(\lambda_+, \lambda_-) & = & 2\dfrac{\partial \psi}{\partial \lambda_-} \end{array} \right.$$

In order to develop the additional diffusion term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$ in the equation (25), we propose to write $\mathbf{D}$ as a linear combination of $\mathbf{G}$ and $\mathbf{Id}$:

$$\mathbf{D} = \alpha(\lambda_+, \lambda_-)\mathbf{G} + \beta(\lambda_+, \lambda_-)\mathbf{Id} \tag{27}$$

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

$$\alpha = \frac{f_1(\lambda_+, \lambda_-) - f_2(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \qquad \text{and} \qquad \beta = \frac{\lambda_+ f_2(\lambda_+, \lambda_-) - \lambda_- f_1(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \tag{28}$$

Indeed, we have

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

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 (25), 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 :

$$\vec{\mathbf{div}}\,(\mathbf{D}) = \alpha\vec{\mathbf{div}}\,(\mathbf{G}) + \mathbf{G}\nabla\alpha + \nabla\beta \tag{29}$$

and the additional term of the equation (25) would be computed as :

$$\begin{aligned}
\nabla I^T \vec{\mathbf{div}}\,(\mathbf{D}) &= \text{trace}\left(\vec{\mathbf{div}}\,(\mathbf{D})\,\nabla I_i^T\right) \\
&= \alpha\text{trace}\left(\vec{\mathbf{div}}\,(\mathbf{G})\,\nabla I_i^T\right) && (30)\\
&+ \text{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) && (31)\\
&+ \text{trace}\left(\nabla\beta\nabla I_i^T\right) && (32)
\end{aligned}$$

In the following, we propose to find formal expressions of (30), (31) and (32).

- 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 30 as :

$$\alpha \text{trace}\left(\vec{\mathbf{div}}\,(\mathbf{G})\,\nabla {I_i}^T\right) = \sum_{j=1}^{n} \alpha \text{trace}\left(\mathbf{H}_j\left[\nabla {I_i}^T \nabla I_j \mathbf{Id} + \nabla I_j \nabla {I_i}^T\right]\right) \tag{33}$$

□

• We finally have to compute $\nabla\alpha$ and $\nabla\beta$, in the expression (31) and (32). This can be done by the decomposition :

$$\nabla\alpha = \frac{\partial\alpha}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\alpha}{\partial\lambda_-}\nabla\lambda_- \qquad \text{and} \qquad \nabla\beta = \frac{\partial\beta}{\partial\lambda_+}\nabla\lambda_+ + \frac{\partial\beta}{\partial\lambda_-}\nabla\lambda_- \tag{34}$$

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,

$$\nabla\lambda_\pm = \sum_{j=1}^{n} 2\mathbf{H}_j \theta_\pm\theta_\pm^T \nabla I_j \tag{35}$$

We can replace (35) into the expressions of (34), in order to find the spatial gradients of α and β :

$$\left\{ \begin{aligned} \nabla\alpha &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\alpha}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j \\ \nabla\beta &= \sum_{j=1}^{n} 2\mathbf{H}_j \left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T + \frac{\partial\beta}{\partial\lambda_-}\theta_+\theta_+^T\right)\nabla I_j \end{aligned} \right. \tag{36}$$

Using (36), we finally compute the two missing parts (31) and (32) of the additional term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$:

$$\left\{\begin{array}{rcl} \operatorname{trace}\left(\mathbf{G}\nabla\alpha\nabla I_i^T\right) & = & \displaystyle\sum_{j=1}^{n}\operatorname{trace}\left(2\;\mathbf{G}\mathbf{H}_j\left(\frac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\alpha}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \\ \\ \operatorname{trace}\left(\nabla\beta\nabla I_i^T\right) & = & \displaystyle\sum_{j=1}^{n}\operatorname{trace}\left(2\;\mathbf{H}_j\left(\frac{\partial\beta}{\partial\lambda_+}\theta_+\theta_+^T+\frac{\partial\beta}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\right) \end{array}\right. \tag{37}$$

$\square$

- The final step consists in putting together the equations (33) and (37), in order to express the additional term $\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D})$ in the PDE (25).

$$\nabla I_i^T \vec{\mathbf{div}}\,(\mathbf{D}) = \sum_{j=1}^{n}\operatorname{trace}\left(\mathbf{H}_j\mathbf{P}^{ij}\right) \tag{38}$$

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

$$\begin{array}{rcl} \mathbf{P}^{ij} & = & \alpha\;\nabla I_i^T\nabla I_j\mathbf{Id} \\ & + & 2\left(\dfrac{\partial\alpha}{\partial\lambda_+}\theta_+\theta_+^T+\dfrac{\partial\alpha}{\partial\lambda_-}\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T\mathbf{G} \\ & + & 2\left((\alpha+\dfrac{\partial\beta}{\partial\lambda_+})\theta_+\theta_+^T+(\alpha+\dfrac{\partial\beta}{\partial\lambda_-})\theta_-\theta_-^T\right)\nabla I_j\nabla I_i^T \end{array} \tag{39}$$

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 (38) 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 :

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

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

$$\mathbf{Q}^{ij} \quad = \quad \frac{\mathbf{P}^{ij}+\mathbf{P}^{ij^T}}{2} \tag{40}$$

and we have :

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

Finally, the divergence-based PDE (25) can be written as :

$$\mathrm{div}\,(\mathbf{D}\nabla I_i) = \sum_{j=1}^{n} \mathrm{trace}\left((\delta_{ij}\mathbf{D}+\mathbf{Q}^{ij})\mathbf{H}_j\right) \tag{41}$$

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

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

$\square$

The regularization PDE (41) is equivalent to the divergence-based equation $\frac{\partial I_i}{\partial t} = \mathrm{div}\,(\mathbf{D}\nabla I_i)$, but with a trace-based formulation.

References

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

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

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

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In Kurt Akeley, editor, *Proceedings of the SIGGRAPH*, pages 417–424. ACM Press, ACM SIGGRAPH, Addison Wesley Longman, 2000.

[5] P. Blomgren and T.F. Chan. Color tv: Total variation methods for restoration of vector-valued images. *IEEE Trans. Imag. Proc.*, 7(3):304–309, 1998. Special Issue on Partial Differential Equations and Geometry-Driven Diffusion in Image Processing and Analysis.

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

[7] T. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature based inpainting. *SIAM J. Appl. Math.*, 2002.

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

[9] T.F. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (cdd). *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001.

[10] 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.

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

[12] 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*, 39(2):111–129, September 2000.

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

[14] P. Kornprobst, R. Deriche, and G. Aubert. Nonlinear operators in image restoration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 325–331, Puerto Rico, June 1997. IEEE Computer Society, IEEE.

[15] Laurence Lucido, Rachid Deriche, Luis Alvarez, and Vincent Rigaud. Sur quelques schémas numériques de résolution d'équations aux dérivées partielles pour le traitement d'images. Rapport de Recherche 3192, INRIA, June 1997.

[16] 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.

[17] A. Pardo and G. Sapiro. Vector probability diffusion. In *Proceedings of the International Conference on Image Processing*. IEEE Signal Processing Society, September 2000.

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

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

[20] 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.

[21] J. Shah. Curve evolution and segmentation functionals: Applications to color images. In *Proceedings of the International Conference on Image Processing*, pages 461–464, 1996.

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

[23] B. Tang, G. Sapiro, and V. Caselles. Diffusion of general data on non-flat manifolds via harmonic maps theory : The direction diffusion case. *The International Journal of Computer Vision*, 36(2):149–161, February 2000.

[24] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 839–846, January 1998.

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

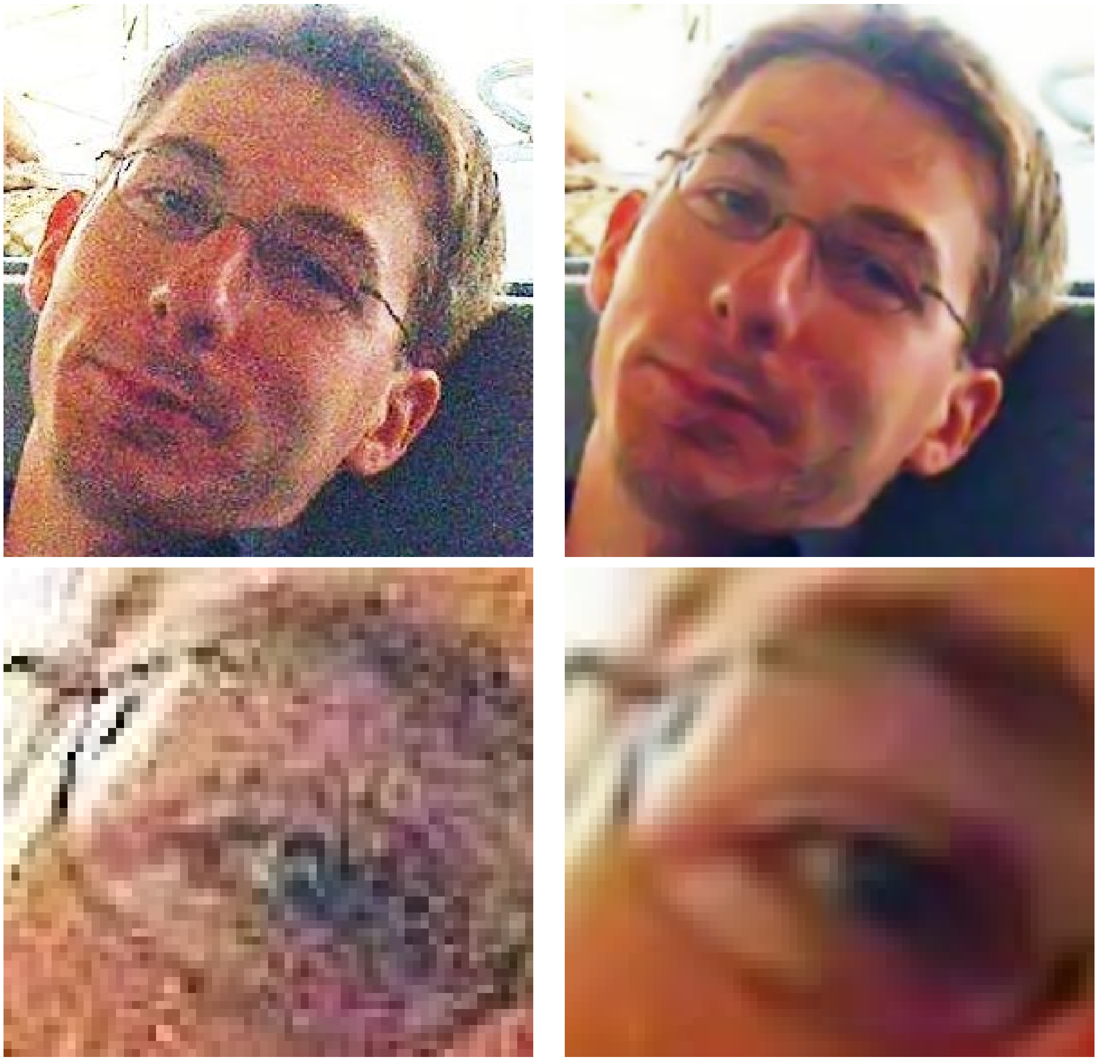
[26] D. Tschumperlé and R. Deriche. Constrained and unconstrained pde's for vector image restoration. In Ivar Austvoll, editor, *Proceedings of the 10th Scandinavian Conference on Image Analysis*, pages 153–160, Bergen, Norway, June 2001.

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

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

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

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



(a) Noisy color image

(b) Restored color image

Figure 4: Using vector-valued regularization PDE's, for color image restoration.

(c) Lossy compressed JPEG image (d) Improved color image

Figure 5: Using vector-valued regularization PDE's, for improvement of lossy compressed images.

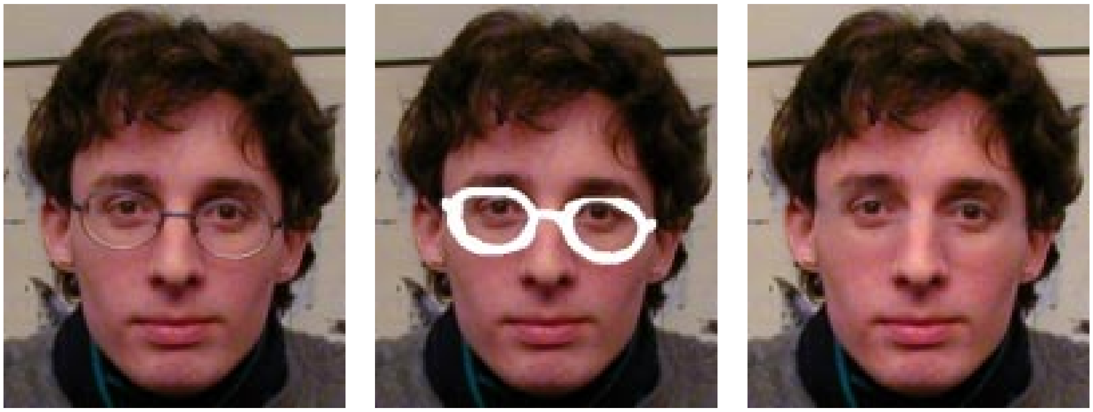
(a) Image with undesired text

(b) Inpainted color image

(c) Zoom of (a)

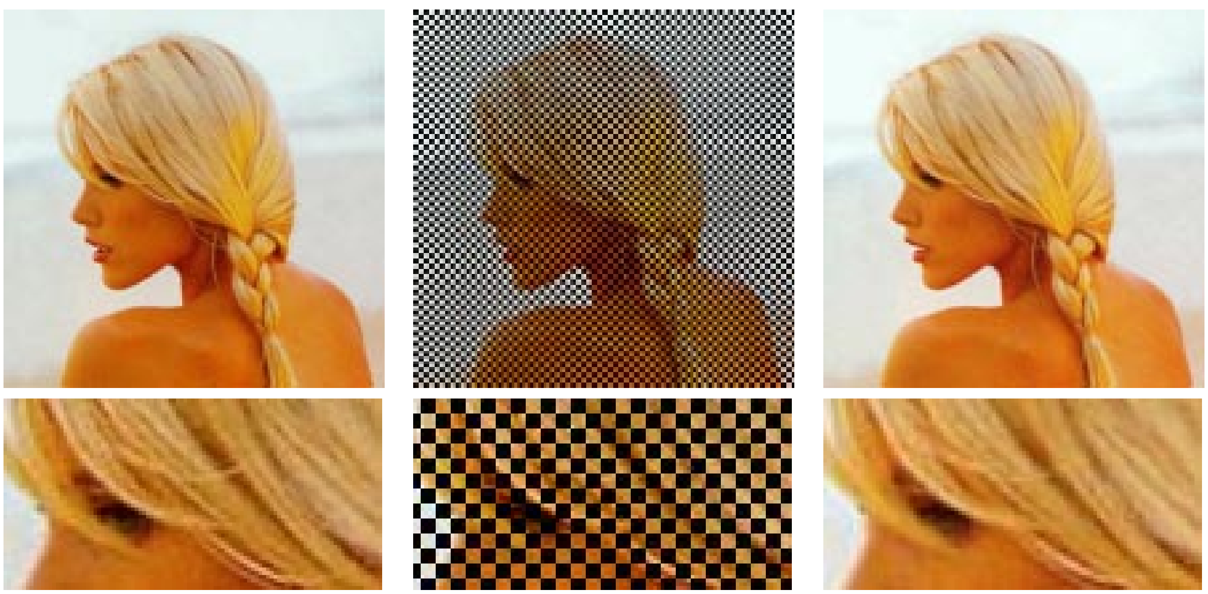
(d) Zoom of (b)

Figure 6: Using vector-valued regularization PDE's for color image inpainting (1).



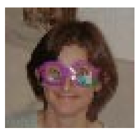
(e) Original color image (f) Image + Inpainting mask (g) Inpainted image

Figure 7: Using vector-valued regularization PDE's for color image inpainting (2).



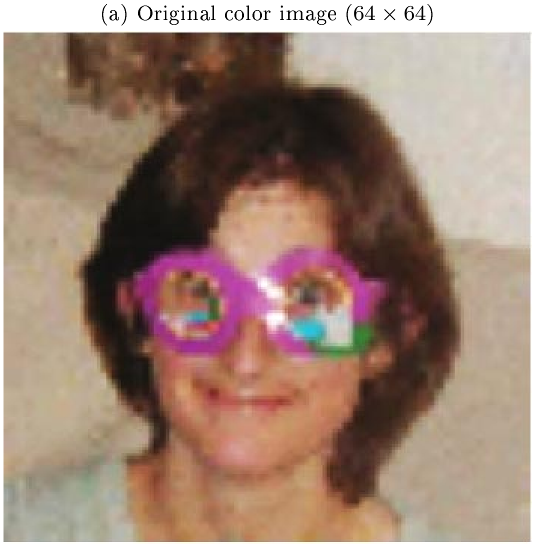
(a) Color image (b) Removing 50% of the pixels (c) Reconstructed image

Figure 8: Using vector-valued regularization PDE's for image reconstruction.

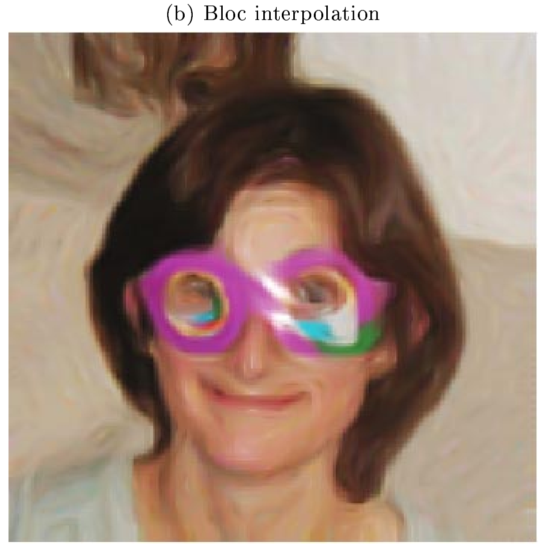


(a) Original color image (64 × 64)

(b) Bloc interpolation

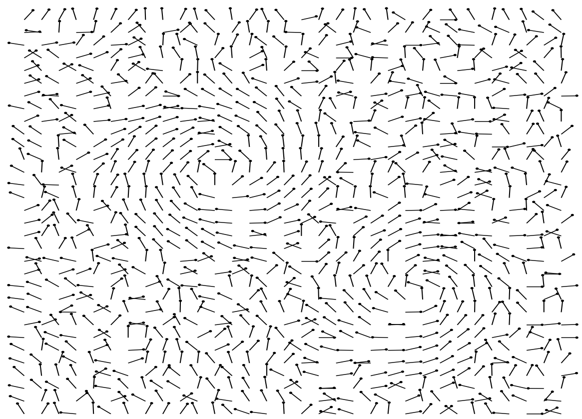


(c) Linear interpolation

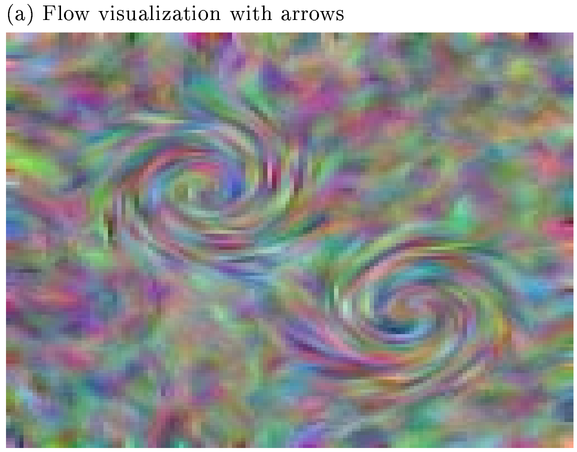


(d) Interpolation with PDE's

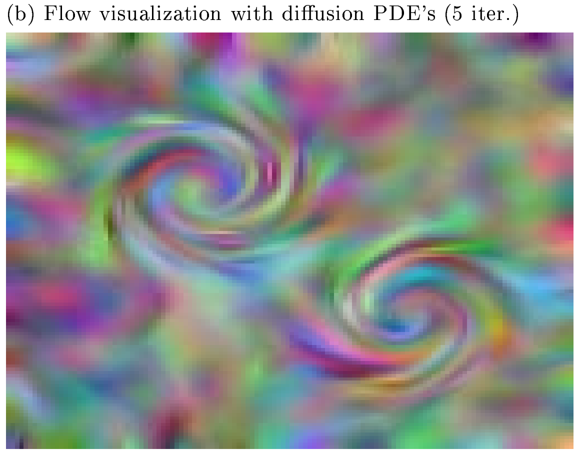
Figure 9: Using vector-valued regularization PDE's for image magnification (×4).



(a) Flow visualization with arrows



(b) Flow visualization with diffusion PDE's (5 iter.)



(c) Flow visualization with diffusion PDE's (15 iter.)

Figure 10: Using vector-valued regularization PDE's, for flow visualization (1).

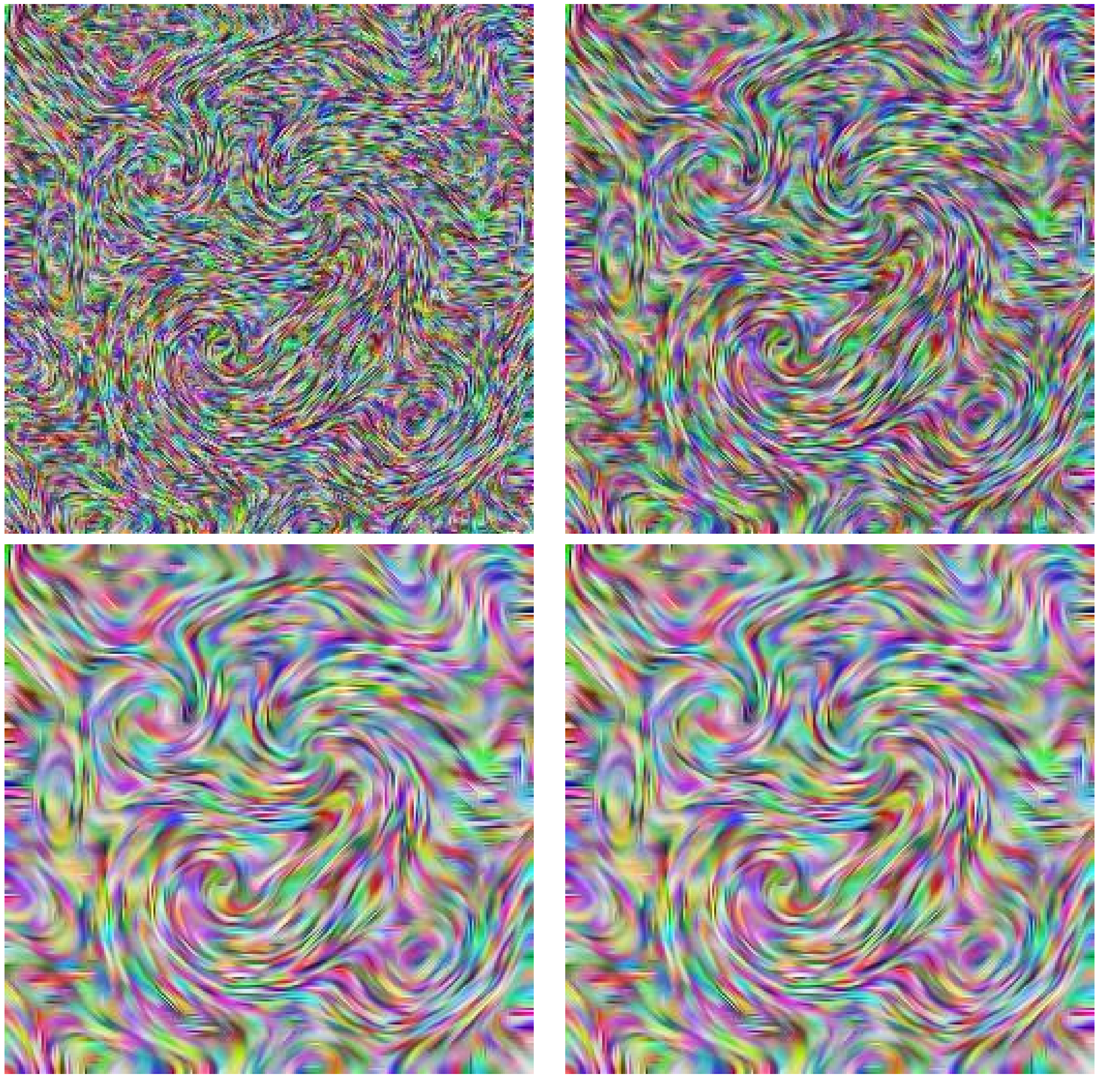


Figure 11: Using vector-valued regularization PDE's, for flow visualization (2).

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399