

THÈSE

présentée à

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS

pour obtenir le titre de

DOCTEUR EN SCIENCES

École Doctorale Sciences et Technologies de l'Information et de la
Communication

Spécialité

Automatique, Traitement du Signal et de l'Image

soutenue par

David Tschumperlé

le 13 décembre 2002

PDE's Based Regularization of Multivalued Images and Applications

Régularisation d'Images Multivaluées par EDP et Applications

Directeur de thèse : **Rachid Deriche**

Jury

Président

Pr. Olivier Faugeras

Rapporteurs

Pr. Luis Alvarez

Pr. Nir Sochen

Examineurs

Pr. Michel Barlaud

Dr. Luc Robert

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE DES SCIENCES ET TECHNOLOGIES DE
L'INFORMATION ET DE LA COMMUNICATION

Rapport de soutenance
sur la thèse de doctorat en Sciences,
Mention : "Automatique, traitement du signal et des images",
de Monsieur David Tschumperlé

portant sur le sujet : "Régularisation d'images Multivaluées par EDP et Applications."

David Tschumperlé a fait un exposé clair, didactique et pédagogique de ses travaux de recherche, qu'il a su illustrer de nombreux exemples.

Il a très bien su mettre en avant le côté novateur de la modélisation mathématique de questions centrales en traitement d'images qu'il propose, ainsi que les belles expérimentations numériques qui lui ont permis d'en explorer la validité.

Il est apparu clairement qu'il avait fait avancer de manière significative l'état de l'art dans le domaine du filtrage d'images multidimensionnelles et sa présentation a fait ressortir ses contributions personnelles et les nombreuses applications potentielles de son travail.

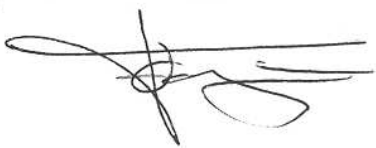
Enfin, il a dégagé en conclusion de son exposé des directions de recherche nouvelles, intéressantes, et prometteuses, notamment dans le domaine des applications aux neurosciences (analyse des connections corticales, calcul des conductivités électriques).

Il a pour terminer très bien répondu aux questions très variées des membres du jury en faisant preuve d'honnêteté et d'une très grande maîtrise du domaine.

En conclusion, les membres du jury sont unanimes à décerner à David Tschumperlé le titre de Docteur de l'Université de Nice Sophia-Antipolis avec la mention "*très honorable*".

Fait à Sophia-Antipolis
Le vendredi 13 décembre 2002

Signature des membres du jury :

 Luis Alvarez
 Nir Sochen
 Nir Sochen
 BARLAUD


Ecole Doctorale STIC de l'Université de Nice-Sophia Antipolis

250 rue Albert Einstein
LEAT BAT 4
LUCIOLES 1
06560 VALBONNE TEL : 04.92.94.28.27

UNIVERSITE DE NICE SOPHIA ANTIPOLIS
UFR SCIENCES
SERVICE SCOLARITE 3EM CYCLE

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

1. TITRE DE LA THESE : « Régularisation d'images Multivaluées par EDP et Applications »

2. NOM ET PRENOM DE L'AUTEUR : Monsieur David TCHUMPERLE

3. MEMBRES DU JURY :

Luis Alvarez
Nir Sochen
Michel Banceland
Rachid Deiche
Luc Robert

4. PRESIDENT DU JURY :

Olivier FAUGERAS

5. DATE DE LA SOUTENANCE :

13/12/02

6. REPRODUCTION DE LA THESE SOUTENUE :



Thèse pouvant être reproduite en l'état



Thèse ne pouvant pas être reproduite



Thèse pouvant être reproduite après correction suggérées au cours de la soutenance

Signature du Président du jury



Secrétariat ED STIC
250 rue Albert Einstein
Lucioles 1 bat 4
06560 Valbonne

Résumé

Nous nous intéressons aux approches par EDP pour la régularisation d'images multivaluées, et leurs applications à une large classe de problèmes d'intérêts. L'étude et la comparaison des méthodes existantes nous permet à la fois de proposer un cadre mathématique commun mieux adapté aux interprétations géométriques locales de ces EDP, mais aussi de concevoir des schémas numériques efficaces pour leur mise en oeuvre. Nous développons de cette façon une nouvelle approche de régularisation multivaluée vérifiant certaines propriétés géométriques locales importantes, qui peut être utilisée dans de nombreuses applications différentes. Nous abordons ensuite le problème lié à la régularisation de données multivaluées contraintes. Un formalisme variationnel est proposé afin de traiter dans un cadre unifié, des données de direction comme les champs de vecteurs unitaires, de matrices de rotation, de tenseurs de diffusion etc. Les solutions apportées sont analysées et utilisées avec succès pour résoudre de nombreux problèmes, notamment la régularisation et l'interpolation d'images couleurs, la visualisation de flots, la régularisation de mouvements rigides estimés à partir de séquences vidéos, et l'aide à la reconstruction de réseaux cohérents de fibres dans la matière blanche du cerveau, à partir de la régularisation d'images d'IRM de diffusion.

Abstract

We are interested in PDE-based approaches for vector-valued image regularization, and its applications for a wide class of interesting image processing problems. The comparative study of existing methods allows us to propose a common mathematical framework, better adapted to understand the underlying diffusion geometry of the regularization processes, as well as design corresponding numerical schemes. Thus we develop a new multivalued image regularization approach that verifies important geometric properties. It can be used in a large range of regularization-related applications. We also tackle the problem of constrained regularization and propose a specific variational formalism unifying in a common framework, the equations acting on direction features : unit vectors, rotation matrices, diffusion tensors, etc. Proposed solutions are analyzed and used with success to solve applications of interest, such as color image regularization and interpolation, flow visualization, regularization of rigid motions estimated from video sequences, and aided reconstruction of coherent fibers network models in the white matter of the brain, using DT-MRI imaging.

Remerciements



La thèse que vous tenez entre les mains a été financée par une bourse MESR, complétée par l'INRIA. Elle a été menée dans le projet ROBOTVIS, devenu ODYSSEE, de l'INRIA Sophia-Antipolis/France : <http://www-sop.inria.fr/odyssee>,

Je tiens à remercier sincèrement les personnes suivantes sans qui je n'aurais pas eu la chance de participer à cette grande aventure qu'est le doctorat :

Tout d'abord, **Mr Olivier Faugeras**, directeur de *ROBOTVIS/ODYSSEE* qui m'a accueilli dans son projet et m'a offert les moyens scientifiques et techniques, qui ont permis l'aboutissement de cette thèse. Je le remercie également d'avoir accepté d'être membre du jury.

Mr Rachid Deriche, mon directeur de thèse, qui m'a encadré patiemment durant ces trois années et m'a fait partager sa grande expérience du domaine de la recherche. Je le remercie de m'avoir soutenu et motivé dans les moments difficiles.

Mr Luis Alvarez, et **Mr Nir Sochen**, rapporteurs de cette thèse qui ont acceptés de lire et de juger mon manuscrit. Je les remercie aussi d'avoir participé cette soutenance, malgré les longues distances qui les séparaient de Nice. Je leur suis reconnaissant de leur patience et de leurs conseils avisés.

Mr Michel Barlaud, et **Mr Luc Robert**, qui ont acceptés aimablement et sans hésitation la proposition d'être membre du jury. Merci pour votre enthousiasme.

Comme une thèse se construit aussi quelques temps avant sa soutenance, je me dois de remercier mes collègues de travail de l'INRIA, avec qui j'ai beaucoup appris durant ces trois années de labeur, dans une ambiance studieuse mais toujours détendue. Ils ont toujours été patients vis à vis de mon tempérament parfois impulsif. Je pense plus particulièrement à Gérardo Hermosillo, Jacques Bride, Frédéric Abad, Christophe Chef d'hôtel, Marie-Cécile Lafont, Bertrand Thirion, Diane Lingrand, François Gaspard. Mais aussi tous les autres : Cyrille, Quentin, Nikos, Jan, Soraya, Maureen, José, Théo, Robert, Thierry, Pierre, Lionel, Lucéro, Manu, Mikael, Thomas, Nicolas, Jean-Philippe, Julia, Louai, Rico, Stéphanie, qui se reconnaîtront.

Pour me ressourcer en dehors du travail, j'ai toujours pu compter sur les personnes suivantes que je remercie également : Laurent, Cedric, Pascale, Christophe, Fred & Delphine,

Hanane, Philippe, Sébastien C., Boris, Julien.

Je n'oublie pas les membres de la famille Tschumperlé qui ont fait de moi quelqu'un de très sympathique, et ce, depuis tout petit : Maman, Papa, Denis, Stéphane, Papy, Mamy.

Un petit coucou pour mes copains d'enfance qui se demandent encore à quoi ont bien pû servir ces trois ans : Renaud, Jean-Marie, Sébastien G., Sébastien H., Valériane, Sarah, et les autres...

Et pour conclure cette partie, je tiens à remercier celle qui a dû me supporter le plus souvent pendant cette thèse, ce qui a dû certainement demander encore plus d'efforts que de l'écrire, j'ai nommé ♡ Laurence ♡, alias **Lolo**. Merci elle pour sa patience (presque) sans limites.

Note : Les quelques citations proposées dans cette thèse sont la propriété intellectuelle de leurs auteurs respectifs.

Contents

Introduction	12
1 General Context and Notations	17
1.1 PDE's and Computer Vision : A quick tour	17
1.1.1 What is Computer Vision ?	17
1.1.2 The framework of PDE's	20
1.2 Mathematical notations	24
1.2.1 Definition of images	24
1.2.2 Image derivatives	25
1.2.3 Calculus of variations and PDE's	27
1.2.4 Definition of tensors	27
2 State of the Art and Motivations	31
2.1 Regularization of scalar images with PDE's	31
2.1.1 Isotropic smoothing	33
2.1.2 Perona-Malik regularization	34
2.1.3 Variational methods and ϕ -functionals	37
2.1.4 Divergence-based PDE's	39
2.1.5 Diffusion PDE's based on oriented 1D Laplacians	41
2.2 Regularization of multivalued images with PDE's	43
2.2.1 Channel by channel regularization	43
2.2.2 Defining a vector geometry	44
2.2.3 Color Total Variation	49
2.2.4 Vector Φ -functionals	50
2.2.5 Coherence Enhancing Diffusion	51
2.2.6 The Beltrami Flow	52
2.2.7 Vector $I_{\xi\xi}$ Diffusion	53
2.2.8 Comparative figures	54
2.3 Regularization PDE's and constraints	59
2.3.1 Direction diffusion	59
2.3.2 Other constrained problems	62
2.4 Open questions on PDE-based regularization methods	64

2.4.1	On the unconstrained regularization	64
2.4.2	On the constrained regularization	68
3	From Variational to Local Formulations	69
3.1	From functionals to divergence forms	69
3.1.1	A generic functional for vector-valued regularization	69
3.1.2	Computing the gradient descent	71
3.1.3	Link with previous multivalued regularization approaches	74
3.2	From divergence forms to oriented Laplacians	77
3.2.1	Geometric interpretation of oriented Laplacians	77
3.2.2	Difference with divergence-based equations	83
3.2.3	Development of the divergence form	84
3.3	A unified expression for multivalued regularization PDE's	89
3.3.1	The general formula	89
3.3.2	The link with other approaches	89
3.4	Defining a new multivalued regularization/enhancement process	93
3.4.1	Vector geometry-adapting regularization	93
3.4.2	Vector-valued edge enhancement	95
3.5	Numerical schemes for multivalued regularization	97
3.5.1	Elementary iterative loop	98
3.5.2	Using a spatial discretization of the Hessian	99
3.5.3	Using local filtering considerations	100
3.5.4	Comparative figure	100
4	Applications of Unconstrained PDE's	103
4.1	Color image restoration	103
4.2	Color image inpainting	107
4.3	Color image magnification	110
4.4	Flow visualization	111
4.5	Other applications and extensions	113
5	Constrained PDE's and the Framework of Orthonormal Vector Sets	119
5.1	Interest of orthonormal vector sets	119
5.1.1	Preliminary notations	120
5.1.2	Orthonormal vector sets and direction features	121
5.2	Regularizing fields of orthonormal vector sets	122
5.2.1	Unconstrained regularization	122
5.2.2	A way of preserving the orthonormal constraints	123
5.3	Regularization of Directions and Rotations	127
5.3.1	Unit vector regularization	127
5.3.2	Orthogonal 3×3 matrices	130
5.3.3	A physical interpretation for 3D orthonormal vectors	132

5.4	Diffusion tensor regularization	134
5.4.1	Direct approach for tensor regularization	134
5.4.2	Spectral regularization approach	137
5.5	Numerical schemes for orthonormal vector set regularization	140
5.6	Extension to general matrix-constrained flows	142
5.6.1	Matrix constraints and exponential maps	142
5.6.2	Three different matrix-valued flows for tensor regularization . . .	143
5.6.3	Comparative figures	145
6	Applications of Constrained PDE's	147
6.1	Direction field regularization	147
6.2	Regularization of estimated camera motions	152
6.3	DT-MRI image regularization	156
	Appendix	157
	Conclusion & Perspectives	165
	Bibliography	173

List of Figures

1.1	Evolution of the computer technology.	18
1.2	Some of the human senses and their computer analogues.	18
1.3	Understanding the 3D behind photographs.	19
1.4	Retrieving image features and shapes : segmentation and regularization algorithms.	20
1.5	Image restoration, treated as the evolution of a surface.	21
1.6	Image segmentation, treated as the evolution of a 2D contour curve.	22
1.7	Image registration, treated as the evolution of a displacement field.	22
1.8	Two other computer vision problems, treated as 3D surface evolutions.	23
1.9	Spatial derivatives of a scalar image I	26
1.10	Diffusion tensor representation with ellipsoids.	29
2.1	Regularization PDE's and the notion of scale-space.	32
2.2	Regularization term for image restoration.	32
2.3	Heat equation applied on a noisy scalar image.	34
2.4	An image contour and its moving vector basis (ξ, η)	35
2.5	Perona-Malik flow applied on a noisy scalar image.	36
2.6	List of different proposed ϕ -functions.	37
2.7	ϕ -function based PDE's applied on a noisy scalar image.	38
2.8	Locally designed PDE's applied on a noisy scalar image.	42
2.9	Channel by channel approach vs Vector-valued PDE's.	44
2.10	Color image with iso-luminance pixels.	46
2.11	A list of possible vector variation norms.	48
2.12	Using vector variation norms for color edge detection.	49
2.13	Comparison of classical vector-valued regularization PDE's (I).	54
2.14	Comparison of classical vector-valued regularization PDE's (II).	56
2.15	Unit vector fields representing directions and chromaticity part of color images.	59
2.16	Regularization of unit vector fields.	62
2.17	Regularization of data defined on non-flat surfaces.	63
2.18	Differences between divergence-based and trace-based regularization.	67
3.1	The link between diffusion tensors and oriented Gaussian kernels.	80

3.2	Example of the PDE flow $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i)$, with non-constant tensor fields $\mathbf{T}$	82
3.3	Using our vector-valued regularization PDE to regularize noisy color images.	96
3.4	Principle of shock filters.	97
3.5	Vector-valued shock filters applied on a color image.	97
3.6	Mixing regularization terms and vector-valued shock filters.	98
3.7	Comparison of the two proposed numerical schemes.	101
4.1	Restoration of a real noisy digital photograph.	105
4.2	Improvement of a lossy compressed color image.	106
4.3	Color image inpainting.	109
4.4	Color image inpainting (detail).	110
4.5	Color image inpainting, used to remove real objects in photographs. . . .	111
4.6	Color image inpainting, used to reconstruct partially coded images. . . .	112
4.7	Magnification of a color cartoon image ($\times 3$).	115
4.8	Magnification of a real color photograph ($\times 4$).	116
4.9	Flow visualization.	117
4.10	Scale-space visualization of a flow.	118
5.1	Fields of orthonormal vector sets.	121
5.2	How to regularize a field $\mathcal{B}$ of orthonormal vector sets ?	122
5.3	Decoupled regularization of orthonormal vector sets.	124
5.4	Anisotropy and constraint-preserving property of the orthonormal-constrained regularization.	127
5.5	The geometric intuition behind the norm constraint.	128
5.6	Fields of 3D orthonormal vector bases.	130
5.7	Decomposition method for rotation field regularization.	131
5.8	Rotation field regularization using orthonormal preserving PDE's.	132
5.9	A solid object $\mathcal{B}$, submitted to physical forces.	133
5.10	Comparison of tensor regularization methods.	136
5.11	Local vector alignment procedure for tensor regularization.	139
5.12	Numerical errors with classical explicit schemes.	140
5.13	Matrix-constrained flow, seen as a tangent flow to a non-flat manifold $\mathcal{M}$. .	142
5.14	Using exponential maps to numerically preserve constraints.	143
5.15	Comparison of different approaches for diffusion tensor regularization. . .	146
6.1	Constrained vs. unconstrained PDE's for direction regularization.	149
6.2	Constructing a scale-space of a direction field.	150
6.3	Norm constrained regularization for chromaticity denoising in color images. .	151
6.4	Principle of our camera motion regularization.	153
6.5	Regularization of video camera motions for 3D virtual object insertion. .	155

6.6	Constructing a structure preserving scale-space of the white matter fibers.	158
6.7	Comparison of 3 different methods for DT-MRI dataset regularization. . .	159
6.8	Principle of the loop <code>cimg_m3x3map</code>	164

Introduction

Les approches variationnelles et les outils EDP (équations aux dérivées partielles) ont depuis la fin des années 1980 soulevés un vif intérêt dans le domaine du traitement d'image. La possibilité de travailler avec des équations *non-linéaires*, a permis à la fois d'améliorer de façon significative l'ensemble des algorithmes essentiels pour le traitement d'image (segmentation, restauration, recalage, etc.), mais aussi de définir un cadre mathématique rigoureux pour leur étude théorique. Nous nous intéressons ici aux méthodes de *régularisation de données*. Ces types d'approches sont très utilisés en traitement d'images car ils permettent d'introduire un à-priori de *régularité* sur les données considérées, et de nombreuses applications concrètes en découlent directement. Ce sujet a été particulièrement bien traité dans le cas de la régularisation *d'images scalaires*, c'est-à-dire en niveau de gris (une valeur par pixel). Plus récemment, des EDP de régularisation agissant sur des image *multivaluées* contraintes ou non contraintes ont été proposées, en se basant sur des formalismes très variés.

Cette thèse se propose de faire un point sur ces nombreuses formulations EDP pour *la régularisation de données multivaluées contraintes et non contraintes*, et de contribuer à étendre certains aspects de ces méthodes. Nous avons organisé ce document de la façon suivante :

Chapitre 1

Nous essayons ici de donner un aperçu rapide de l'intérêt des méthodes variationnelles et EDP pour résoudre des problèmes de traitement d'images, notamment ceux reliés à la vision par ordinateur. Nous définissons aussi les notations mathématiques et les concepts de bases utilisés tout au long de cette thèse.

Chapitre 2

Dans ce chapitre, nous proposons un état de l'art sur les méthodes de régularisation existantes, aussi bien dans le cas scalaire que dans le cas multivalué contraint et non-contraint. Nous nous efforçons de lier ces nombreuses méthodes entre elles en les comparant et en illustrant leur comportement local pour le problème particulier de la restauration d'images couleurs.

Chapitre 3

Nous contribuons à comprendre plus précisément les différentes méthodes de régularisation multivaluées, en proposant un formalisme mathématique commun qui permet de développer et de relier les différents niveaux d'interprétations des EDP de diffusion correspondantes (des approches les plus globales aux plus locales). Cela nous permet à la fois de proposer une nouvelle EDP de

régularisation vérifiant certaines propriétés géométriques souhaitées, mais aussi de proposer un schémas numérique commun pour implémenter toutes ces EDP.

Chapitre 4

Cette partie nous permet d'illustrer de manière pratique l'analyse proposée dans le chapitre précédent. Nous appliquons en effet nos équations de diffusion multivaluées pour résoudre différents problèmes traitant des images couleurs, notamment la restauration et l'interpolation, ainsi que la visualisation de champs de vecteurs.

Chapitre 5

Nous nous intéressons ici à l'extension de nos méthodes au cas *multivalué contraint*. L'unification des EDP de régularisation agissant sur des données de *direction* (vecteurs unitaires, matrices de rotations, etc.) est réalisée grâce à un formalisme variationnel considérant des images d'*ensemble de vecteurs orthonormés*. Cela contribue à étendre les types d'images multivaluées qui peuvent être traitées par des EDP de régularisation. Nous proposons notamment des approches originales permettant de restaurer des champs de matrices de rotation, et des images de tenseurs de diffusion.

Chapitre 6

Les applications de nos EDP multivaluées contraintes sont illustrées dans ce chapitre, avec trois problèmes différents : D'abord nous traitons le cas 'classique' de la restauration de champs de direction et d'images couleurs avec bruit chromatique. Puis nous nous attaquons aux problèmes plus novateurs de la régularisation de mouvements estimés de caméra, afin de permettre des re-projections d'objets virtuels 3D dans des séquences filmées de manière plus réaliste, et de la restauration d'images IRM de tenseurs de diffusion, autorisant en particulier la construction de modèles de réseaux cohérents de fibres dans la matière blanche du cerveau.

Annexe :

Dans cette dernière partie, nous faisons partager notre expérience sur le travail de programmation nécessaire pour implémenter des flots de régularisation par EDP. Nous commentons notamment des programmes en *C*, permettant l'application des algorithmes nouveaux présentés dans le chapitre 3 de cette thèse.

★ Dans ce manuscrit, nous nous sommes toujours efforcés de mettre en avant la *motivation générale* qui nous a amené aux développements mathématiques de nouvelles équations de régularisation. Une des motivations importantes de ce travail de recherche a été de valoriser l'aspect *applicatif* de nos méthodes, en illustrant dès que possible les algorithmes correspondant pour des problèmes concrets de traitement d'images. Nous espérons ainsi avoir soulevé d'une part de nouvelles questions théoriques, et apporté d'autre part des méthodes originales et efficaces, sur la base de l'utilisation des EDP de régularisation ★

Introduction

PDE (partial differential equations) and variational methods have raised a strong interest since the end of the 1980's, in the image processing field. The ability to deal with *nonlinear* equations significantly improved a whole range of image processing algorithms (segmentation, restoration, registration, etc.). But it also defined a rigorous mathematical framework for their theoretical studies. We are interested more particularly in *data regularization methods*. This kind of approaches is very often used to solve image processing problems, because it introduces a prior *regularity knowledge* on the considered data, which have frequently to satisfy smoothness constraints. This subject has been deeply analyzed for the case of *scalar images*, i.e gray level data (a single value per image pixel). More recently, regularization PDE's acting on *multivalued images* have been proposed, relying on various formalisms.

In this thesis, we propose to review this large set of constrained and unconstrained multivalued regularization methods, and contribute to extend some aspects of these approaches. This document is organized as follows :

Chapter 1

We try to give a general idea of variational and PDE-based methods which are useful to solve image processing problems, particularly those linked to *computer vision*. We also define the usual mathematical notations and basic concepts used throughout this thesis.

Chapter 2

In this chapter, we propose a state of the art on existing PDE regularization methods acting on scalar images, and constrained or unconstrained multivalued datasets. We try to link these methods together as much as possible, by comparing and illustrating them with the problem of color image restoration.

Chapter 3

We contribute to understand more precisely the different multivalued regularization methods, by proposing a common mathematical formalism that allows to stress the different interpretation levels (local and global) of the corresponding diffusion PDE's. On one hand, it allows to propose a new multivalued regularization PDE respecting desired geometric properties. On the other hand, it naturally design an unified numerical scheme that can be used to implement all these regularization equations.

Chapter 4

This part illustrates from a practical viewpoint, the analysis proposed in the previous chapter. Indeed, we apply our multivalued diffusion PDE's in order to solve different problems related to color images (restoration and interpolation) and flow visualization.

Chapter 5

We are interested in the extension of our regularization methods to the *constrained multivalued case*. The unification of the diffusion PDE's acting on *direction* features (such as unit vectors, rotation matrices, etc.) is done using a general variational formalism considering images of *orthonormal vector sets*. It contributes to extend the different data types that can be handled by regularization PDE approaches. Particularly, we propose original methods allowing to restore fields of rotation matrices, and diffusion tensor images.

Chapter 6

The application of our multivalued and constrained PDE's is illustrated in this chapter, with three different problems of interests. We first tackle the 'classical' cases of direction field restoration and chromaticity denoising of color images. Then, we address more recent and difficult issues : the regularization of estimated camera motions, in order to project smoothly virtual 3D objects into real movies. Then, the restoration of DT-MRI tensor images, allowing the computation of regularized and coherent models of fiber networks in the white matter of the brain.

Annex :

In this last part, we share our coding experience. We propose *C* source code that implements PDE regularization flows, more particularly the original one presented in chapter 3 of this thesis.

★ In this document, we always tried hard to explain the *general motivation* leading to the mathematical developments of new regularization equations. One of the main motivation of this research work has been the illustration of *applicative aspects* of the proposed methods, in order to deal with various and concrete regularization-related problems, encountered in image processing. Thus, we hope that we raised new theoretical questions and we brought new and efficient regularization methods.

Chapter 1

General Context and Notations



This chapter is intended to describe the general context of PDE's and related computer vision problems. It quickly sums up the important stakes and contributions in this large field and emphasizes the specific interests of such approaches for image processing. We also define the mathematical notations used in the sequel of this manuscript and remind some classical results on variational calculus and PDE evolutions.

1.1 PDE's and Computer Vision : A quick tour

Preliminary note : For three years, my close friends and my family had no precise idea about what I did in front of my computer everyday. This section is intended to popularize computer vision and image processing for them and explain how we can solve some of the related problems within the framework of PDE's and variational tools, as well as it gives the general context where our work takes place. I would like to apologize to the initiated reader for the over-simplified view of this large scientific domain and invite him to go directly to the notations section 1.2.

1.1.1 What is Computer Vision ?

Computers are anywhere. Since the invention of the computer sciences, the technology has raised with an exponential speed and many dreams that were unbelievable before became realizable. Only few years were needed to see the growing power of the processors, the emergence of graphical screens, the extreme miniaturization and the cheap prices of electronic components, allowing the democratization of computers in the industries and at home (Fig.1.1 and [195]).

Nowadays computers assist humans for a large range of tasks : from communication tools (cellular phones, computer networks) to automated industrial processes (car industry, CAD modeling) going through leisures and art (video games and digital painting).

Despite the fact that computers are now powerful enough to execute almost any automated jobs as complex as desired, one of the promised revolution has still to come : The ability for computers *to think and to react to the surrounding environment*, like humans do.



Figure 1.1: Evolution of the computer technology.

Computers can be easily connected with video cameras, micros, and other acquisition peripherals that supply measures of the real world (photons, air vibrations, electromagnetic waves, X-ray, etc.), but the automatic interpretation of these data is not fully resolved yet. For instance, binary data coming from a digital camera is not enough to make a computer seeing, just like the electric current propagating in our optical nerves doesn't suffice to speak about human vision : our brain is the required element to interpret these raw input stimuli. *Simulating the understanding of the digital measures* coming from the input peripherals is the key point to overstep the current limitations of the computer use. Actually, this is a very hard problem and its resolution implies the creation of some *artificial intelligence*. It is one of the important aim of the computer sciences today.

One of the expected way of creating computer intelligence is to copy the different abilities of a working model in the nature : the human (more often than not). In this sense, computer input peripherals are analogous to human senses (Fig.1.2).



Figure 1.2: Some of the human senses and their computer analogues.

Trying to simulate the ability of the brain to treat the informations coming from the outside world has opened new research areas in computer sciences, each of them concerned with specific parts of

our global understanding process (from the lowest level of comprehension to the highest). These branches are for instance : Learning, Reasoning, Speech Recognition, *Computer Vision*.

Computer vision is then the field interested in creating computer algorithms that could *analyze and understand the underlying semantic informations contained in digital images*. These informations of interest include for instance :

- **3D understanding**, or how to perceive the original 3D world behind images shot by a digital camera ? Actually, this problem has been one of the most studied in the computer vision literature (see [71, 73] for a complete theory on this subject). Nowadays, computer algorithms are reaching the goal of retrieving 3D models from a set of photographs (Fig.1.3).

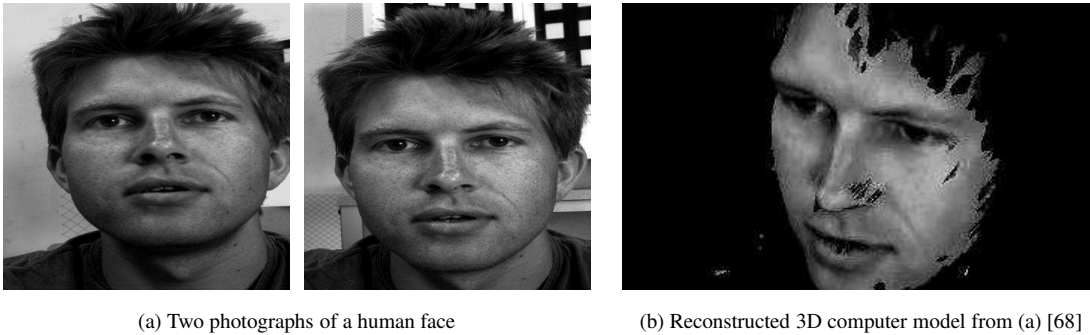


Figure 1.3: Understanding the 3D behind photographs.

- **Shape and object recognition**, or how to discriminate different objects and shapes in a natural image ? This is the matter of *segmentation algorithms*. See [122, 123] for classical works on this subject, with different mathematical formulations. Automated segmentation algorithms are now more and more used to track objects in video sequences (Fig.1.4b).
- **Simple models of images**, or how to simplify images, so that only important image features are preserved ? *Regularization algorithms* are designed to perform such low-level tasks (Fig.1.4c), and are often used as low-level processes before (or in parallel with) any other computer vision algorithm. See [2, 14, 78, 79, 97, 143] for pioneering works in this area. In this thesis, we will focus on these regularization schemes and apply them in order to handle a wide range of image processing problems as for instance : image restoration, interpolation and magnification, flow visualization, simplification of fibers in the white matter of the brain and stabilization of camera motions.

This is of course a non-exhaustive list of computer vision objectives. Other useful schemes have been proposed for instance to find transformations between two images (*image registration algorithms*) or to compute estimated objects velocity in video sequences (*optical flow algorithms*). The point is that one wants to automate specific abilities of the human vision using computers, and even enhance it if possible. Indeed, possible industrial applications are numerous : from the assistance of the medical diagnostics [50, 70] to the detection of drowning people in swimming pools

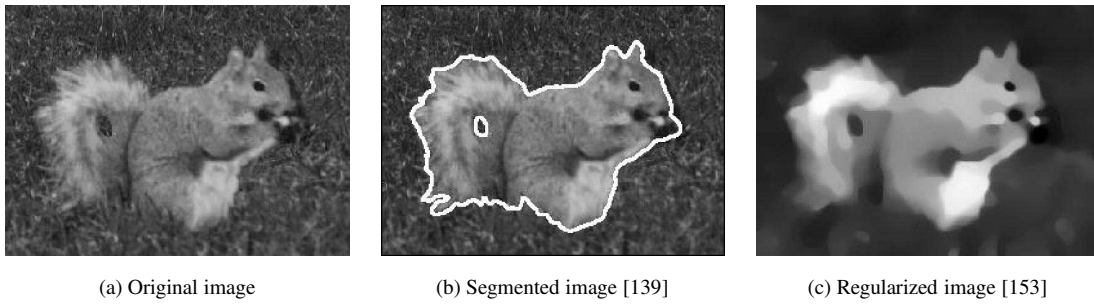


Figure 1.4: Retrieving image features and shapes : segmentation and regularization algorithms.

[144] going through the realization of visual effects for movies and advertising [150], computer vision algorithms are today invading the industry products and the general public market.

The formalization of these algorithms lies on various mathematical frameworks. Some of them are using statistical and discrete theories, while others are based on variational tools and continuous formulations. The fact is that no unique image definitions exist, then a wide variety of theoretical results can be successfully applied to solve computer vision problems. In this thesis, we propose to focus on *PDE-based methods*.

1.1.2 The framework of PDE's

Firstly created to describe physical laws and natural motions of mechanic objects and fluids (strings, water, wind [193]) partial differential equations (PDE's) have been widely studied and extended to other branches of mathematics and physics. This kind of equations is particular in the sense that variables and their derivatives appear in the equation expression. For instance, the following PDE :

$$\frac{\partial^2 \phi(t)}{\partial t^2} + \omega_0^2 \phi(t) = 0$$

states the physical law followed by a simple harmonic oscillator, leading to the motion $\phi(t)$.

PDE's also appear in optimization problems when one wants to minimize energy functionals, via the *Euler-Lagrange equations* (see further section 1.2.3).

In the late 80's, the computer vision community started to be interested in such equations and found interesting properties well adapted to handle classical computer vision problems (see [66] for a state-of-the-art review in 1995). Indeed, PDE's can describe highly non-linear and iterative *continuous object evolutions* like curves, surfaces or vector fields. In the other hand, many computer vision algorithms try to fit *a model* (image, corner, edge, surface, etc.) with *image observations* (photographs, video, MRI data or other complex image modalities) in order to retrieve the observed image semantic.

Representing such models with continuous surfaces, curves or vector fields and evolving them with PDE's has then become a natural way to proceed, and a lot of algorithms based on this idea have been proposed so far in the computer vision literature. We present some classical PDE-based algorithms and briefly explain how the modeling of curves and surfaces may occur.

- **Image regularization** : A noisy 2D image may be considered for instance as a surface I :

$$(x, y) \rightarrow (x, y, I(x, y))$$

With this kind of model, regularizing the image I may be equivalent to find a *smooth* surface similar enough to the original noisy one. This can be done by minimizing energy functionals or directly designing PDE's with specific regularization behaviors that evolve the noisy surface (Fig.1.5). Note that other interesting approaches can be conceived. Regularization PDE-based algorithms constitute the main topic of this thesis and will be analyzed throughout this document, starting from a state of the art, in chapter 2.

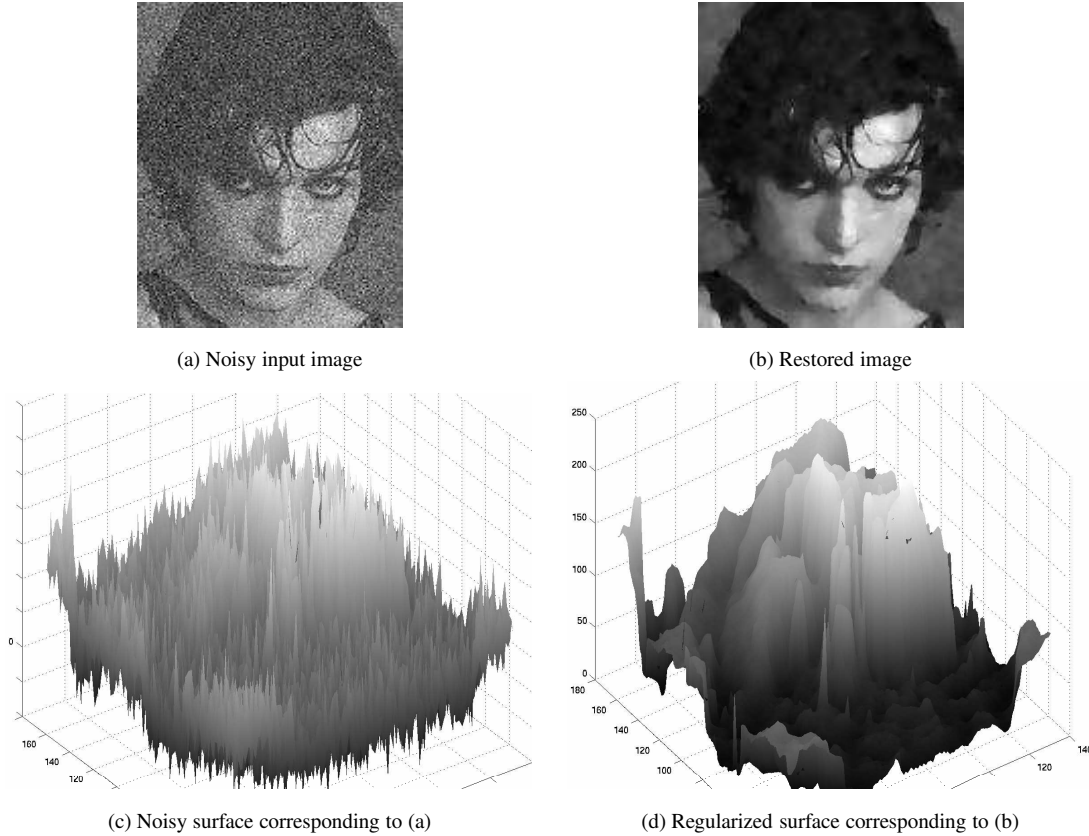


Figure 1.5: Image restoration, treated as the evolution of a surface.

- **Image segmentation** : A classical way of segmenting images with PDE's is to model a closed contour with a 2D curve and then to evolve it from an initial position (random or user-defined) until it fits the exact shape of the objects present in the picture (Fig.1.6). Like image restoration purposes, a PDE is describing the curve evolution and may come from energy minimization or from pertinent local segmentation heuristics.

Abundant literature on this subject can be found for instance in [34, 42, 57, 98, 101, 119, 136, 137, 134, 133, 135, 138, 155, 159, 160].

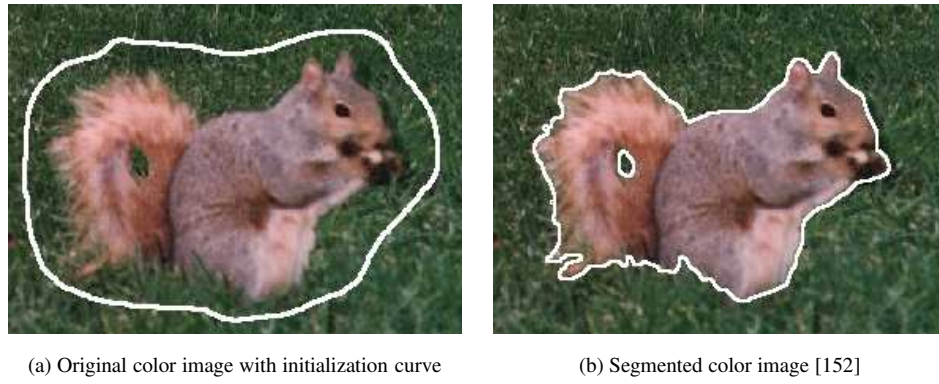


Figure 1.6: Image segmentation, treated as the evolution of a 2D contour curve.

- **Image registration and Optical flow** : The idea is to find a function allowing to *transform an image to another one*. It is particularly used to detect motions in video sequences (by registering consecutive frames), or readjust two images in a way that they fit together (applications in medical images analysis). Here, a vector field models the “pixels motion” between the two images and a PDE is used to describe its evolution until it converges to the expected image transformation (Fig.1.7).

Interesting survey and references on this subject can be found in [5, 3, 6, 11, 13, 15, 17, 18, 49, 56, 67, 74, 94, 98, 109, 117, 124, 151, 172, 183, 192].

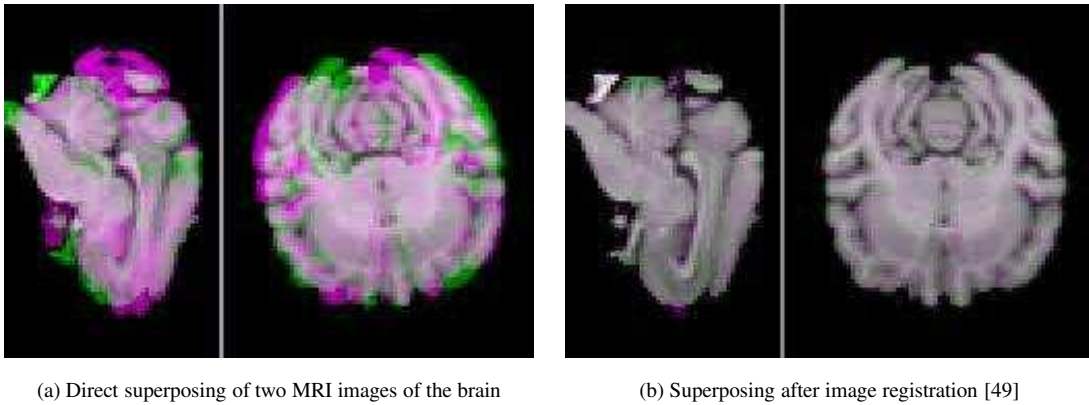
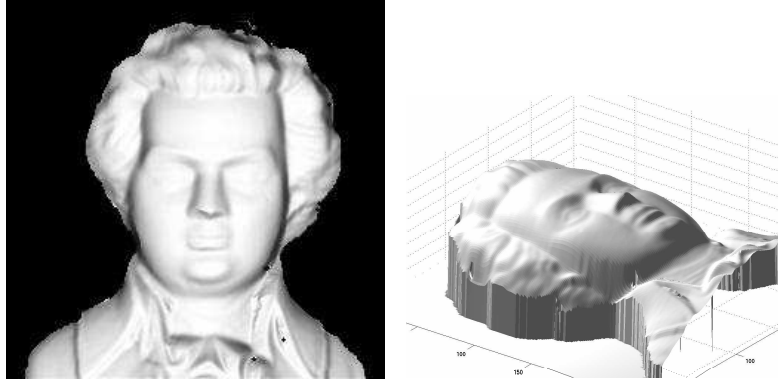


Figure 1.7: Image registration, treated as the evolution of a displacement field.

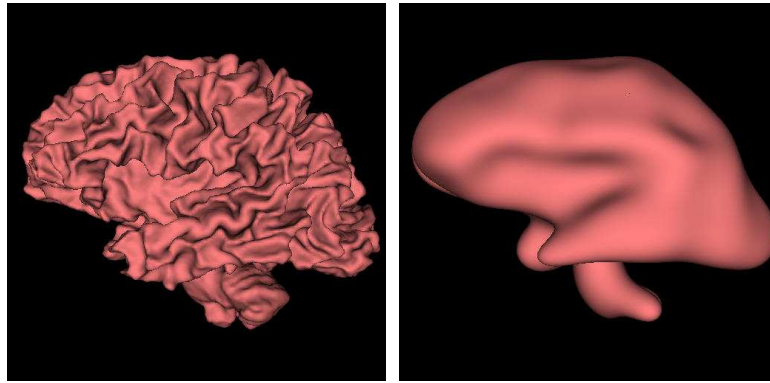
- **Shape from Shading** : This new and challenging problem consists in reconstructing a 3D representation of an object from a *single* photograph of it. It is possible if one looks at the intensity variations of the image pixels due to the shadows and the different illumination conditions during the snapshot. PDE's can describe the flow of an originally flat 3D surface converging to the 3D shape of the real object (Fig.1.8a).

See [95, 96, 148] for a nice panorama of the research in this area.

- **Shape simplification** : Like image restoration, shapes modeled with 3D surfaces can be simplified by regularization PDE's. There are for instance applications in medical imaging, by studying the particular structures of biological objects (for instance the brain, as illustrated in Fig.1.8b and [93]).



(a) Shape from shading [149]



(b) Brain model simplification [87, 93]

Figure 1.8: Two other computer vision problems, treated as 3D surface evolutions.

This application list is obviously incomplete, but it illustrates the high interest in PDE-based methods in computer vision and the large number of applications that follows. Other examples are illustrated for instance in [129, 130].¹

From now on, we will focus on the pioneering and particular problem of *image regularization with PDE's methods*. We close now our general review on computer vision and PDE's formulations and introduce some necessary notations used in the sequel of this manuscript.

¹Some of the images illustrating section 1.1 were kindly provided by C. Chef'd'hotel, G. Hermosillo, E. Prados and M. Rousson / ODYSSEE, INRIA Sophia-Antipolis

1.2 Mathematical notations

1.2.1 Definition of images

When dealing with digital image analysis, one has to properly define the notion of *image*. Nowadays, images on computers are stored using discrete representations of the data but one generally assumes that the discretization is thin enough (in spatial and value steps) to be able to approximate these discrete signals by continuous mathematical functions (or at least piecewise continuous). This is debatable and we refer the reader to [75, 110] for interesting discussions about this metaphysic subject. Nevertheless, the possibility to apply classical mathematical tools as well as the good results obtained with continuous models have often illustrated that this hypothesis is interesting in any way and we will use it from now on.

Let $\Omega \subset \mathbb{R}^p$ be a closed *spatial domain* of dimension p . Ω is the generic name for the definition domain of our images, curves or surfaces. $p \in \mathbb{N}^+$ is the dimension of the underlying space : for the most part of this thesis, $p = 2$ which means that we deal with 2D images. Anyway, we will sometimes consider volumes ($p = 3$) and functions defined on a subset of $\mathbb{R}$ ($p = 1$).

We can now define

- a *scalar image/volume* :

$$I : \left\{ \begin{array}{l} \Omega \subset \mathbb{R}^p \rightarrow \mathbb{R} \\ \mathbf{x} \rightarrow I(\mathbf{x}) \end{array} \right.$$

where $\mathbf{x} = x$ when $p = 1$, $\mathbf{x} = (x, y)$ when $p = 2$ and $\mathbf{x} = (x, y, z)$ when $p = 3$.

Even if pixel values of digital images are discrete and bounded, we assume that $I(\mathbf{x})$ takes its values in the continuous space $\mathbb{R}$. Note that this kind of image can represent only *gray-valued* images or volumes (a single intensity per pixel).

- a *vector-valued image/volume* :

$$\mathbf{I} : \left\{ \begin{array}{l} \Omega \subset \mathbb{R}^p \rightarrow \mathbb{R}^n \\ \mathbf{x} \rightarrow \mathbf{I}(\mathbf{x}) \end{array} \right.$$

Here, each image point is a vector of dimension $n \in \mathbb{N}^+$. Note that color images correspond to $n = 3$, with vector values in (R, G, B) (or another color space, see [147]).

We denote by $I_i : \Omega \rightarrow \mathbb{R}$, the i^{th} vector component of $\mathbf{I}$, also called *image channel* (with $1 \leq i \leq n$). This is obviously a scalar image itself. Note then that

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}(\mathbf{x}) = (I_1(\mathbf{x}), I_2(\mathbf{x}), \dots, I_n(\mathbf{x}))^T$$

Generally, we will denote *multi-valued variables* by **bold** letters. This includes vector-valued as well as matrix-valued images (i.e when $\mathbf{I} : \Omega \rightarrow \mathbb{R}^{p \times q}$)

To be more concise, we will often omit the spatial variable $\mathbf{x}$ in the expressions, i.e we will write $\mathbf{I}$ instead of $\mathbf{I}(\mathbf{x})$, when no ambiguities are possible.

1.2.2 Image derivatives

The derivative of the image I with respect to the variable a is written

$$I_a = \frac{\partial I}{\partial a}$$

For vector-valued images $\mathbf{I}$, we have $\mathbf{I}_a(\mathbf{x}) \in \mathbb{R}^n$ and

$$\mathbf{I}_a = \left(\frac{\partial I_1}{\partial a}, \frac{\partial I_2}{\partial a}, \dots, \frac{\partial I_n}{\partial a} \right)^T$$

The derivation of a *scalar image* I with respect to its spatial coordinates $\mathbf{x}$ is called *the image gradient* and is noted by ∇I :

$$\begin{cases} \nabla I = (I_x, I_y)^T & (\text{when } p = 2) \\ \nabla I = (I_x, I_y, I_z)^T & (\text{when } p = 3) \end{cases}$$

It forms a vector-valued field $\nabla I : \Omega \rightarrow \mathbb{R}^p$ representing the maximum variation directions and magnitudes of the scalar image I . The image of the *gradient norms* $\|\nabla I\|$ is often used in image analysis, since it gives a scalar and pointwise measure of the image variations (Fig.1.9).

$$\begin{cases} \|\nabla I\| = \sqrt{I_x^2 + I_y^2} & (\text{when } p = 2) \\ \|\nabla I\| = \sqrt{I_x^2 + I_y^2 + I_z^2} & (\text{when } p = 3) \end{cases}$$

As for *directional derivatives* in a direction $\mathbf{u} \in \mathbb{R}^p$, we use the notations :

$$I_{\mathbf{u}} = \frac{\partial I}{\partial \mathbf{u}} = \nabla I \cdot \mathbf{u}$$

It is for 2D images ($p = 2$) and 3D volumes ($p = 3$) :

$$\begin{cases} \text{if } p = 2, \mathbf{u} = (u, v)^T, & I_{\mathbf{u}} = uI_x + vI_y \\ \text{if } p = 3, \mathbf{u} = (u, v, w)^T, & I_{\mathbf{u}} = uI_x + vI_y + wI_z \end{cases}$$

In the same way, the second derivative of a scalar image I with respect to a then b is denoted by

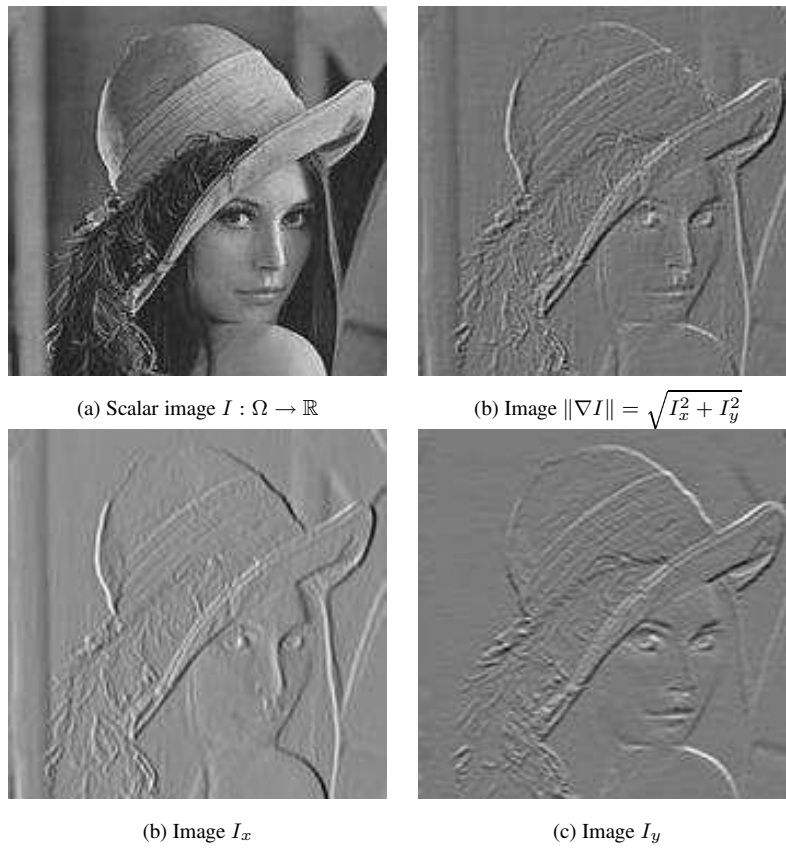
$$I_{ab} = \frac{\partial^2 I}{\partial a \partial b}$$

and we define the *Hessian* of I as the matrix $\mathbf{H}$ of the second derivatives with respect to the spatial coordinates :

$$\mathbf{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix}$$

The matrix $\mathbf{H}$ will be largely used throughout this thesis. We consider that our images are regular enough, $I_{xy} = I_{yx}$. Then, $\mathbf{H}$ is a *symmetric matrix*. We will also use the *Laplacian operator* Δ , defined as

$$\Delta I = \text{trace}(\mathbf{H}) = I_{xx} + I_{yy}$$

Figure 1.9: Spatial derivatives of a scalar image I .

As for *second directional-derivatives* in a direction $\mathbf{u} \in \mathbb{R}^p$, the following notations are equivalent :

$$I_{\mathbf{u}\mathbf{u}} = \frac{\partial^2 I}{\partial \mathbf{u}^2} = \nabla (\nabla I \cdot \mathbf{u}) \cdot \mathbf{u} = \mathbf{u}^T \mathbf{H} \mathbf{u} = \text{trace}(\mathbf{H} \mathbf{u} \mathbf{u}^T)$$

It is for 2D images ($p = 2$) and 3D volumes ($p = 3$) :

$$\left\{ \begin{array}{ll} \text{when } p = 2, \mathbf{u} = (u, v), & I_{\mathbf{u}} = u^2 I_{xx} + 2uv I_{xy} + v^2 I_{yy} \\ \text{when } p = 3, \mathbf{u} = (u, v, w), & I_{\mathbf{u}} = u^2 I_{xx} + 2uv I_{xy} + 2uw I_{xz} + v^2 I_{yy} + 2vw I_{yz} + w^2 I_{zz} \end{array} \right.$$

For a vector-valued image $\mathbf{I}$, the notion of gradient and Hessian is more complex, and we will generally use the gradient and Hessian of its different image channels I_i , respectively written ∇I_i and $\mathbf{H}_i$. Besides, we extend the concept of the gradient norm for vector-valued images, with the *vector gradient norm* $\|\nabla \mathbf{I}\|$:

$$\|\nabla \mathbf{I}\| = \sqrt{\sum_{i=1}^n \|\nabla I_i\|^2} = \sqrt{\mathbf{I}_x \cdot \mathbf{I}_x + \mathbf{I}_y \cdot \mathbf{I}_y} \quad (1.1)$$

Like scalar configurations, the image of $\|\nabla \mathbf{I}\|$ is a useful scalar and pointwise measure of the local vector variations (both in term of vector norms and orientations) of the image $\mathbf{I}$.

1.2.3 Calculus of variations and PDE's

Calculus of variations is a mathematical tool that teaches us how to find extrema of functionals (expressions depending on function integrals rather than simple parameters), for instance :

$$\min_{I:\Omega\rightarrow\mathbb{R}} E(I) = \int_{\Omega} F(x, y, I(x, y), I_x(x, y), I_y(x, y)) d\Omega \quad (1.2)$$

Finding the function I that minimizes the functional $E(I)$ is not a trivial problem. Nevertheless, the *Euler-Lagrange equations* give a necessary condition that must be verified by I to reach a minimum of $E(I)$:

$$\frac{\partial F}{\partial I} - \frac{d}{dx} \frac{\partial F}{\partial I_x} - \frac{d}{dy} \frac{\partial F}{\partial I_y} = 0 \quad (1.3)$$

To avoid the direct and difficult resolution of the PDE (1.3), a classic iterative method is used : the *gradient descent*. Actually, equation (1.3) can be considered as the gradient of the functional $E(I)$. Starting from an initial function I_0 and following the opposite direction of this gradient leads to a local minimizer $I_{\min}$ of $E(I)$:

$$\begin{cases} I_{(t=0)} = I_0 \\ \frac{\partial I}{\partial t} = - \left(\frac{\partial F}{\partial I} - \frac{d}{dx} \frac{\partial F}{\partial I_x} - \frac{d}{dy} \frac{\partial F}{\partial I_y} \right) \end{cases}$$

In the general case when the considered functional $E(I)$ is *not convex*, the starting point I_0 must be carefully chosen, ideally near the global minimum of the functional $E(I)$. Choosing different initializations I_0 may lead to different results (different local minima).

Note that this PDE evolution has been parameterized with an (artificial) *time variable* t . It describes the continuous progression of the function I until it minimizes $E(I)$. Then the PDE velocity vanishes : $\frac{\partial I}{\partial t} = 0$.

The Euler-Lagrange equations make the link between PDE's evolution and gradient descents for functional minimizations. Generally, we will be more interested in the gradient descent itself than the functional minima, and we will often use the term *PDE flows* to describe such evolutions. Please refer to [180] for an exhaustive theory about the calculus of variations.

1.2.4 Definition of tensors

In this thesis, we will generally use the term *tensor* to designate a *symmetric and semi positive-definite matrix*. This is a slight abuse of notations, since a tensor is generally more than a simple matrix (see for instance [1, 166] for detailed tutorials about tensor calculus), but we associate the word '*tensor*' to the particular concept of *diffusion tensors* : indeed, a symmetric and semi positive-definite $n \times n$ matrix $\mathbf{T} \in \mathcal{P}(n)$ may be used to represent *amounts of diffusion* in privileged *spatial directions*.

The generic term *diffusion* describes for instance :

- The spatial dispersions of the error in statistical calculus (in this case, diffusion tensors are also named *covariance matrices*), see for instance [32, 100, 141, 116].

- The local smoothing behaviors of regularization processes. This point will be particularly detailed in chapters 2 and 3. Note that in these chapters, we will sometimes skip the *positive-definiteness* properties, allowing then the tensors to represent *inverse diffusion*.
- The water molecule motion in some biological tissues fibers. The tensors reporting this motion are encountered for instance in DT-MRI imaging (this subject will be tackled in chapter 6).

As this thesis deals with diffusion tensors in two different contexts (DT-MRI images and regularization PDE's), we remind here some classical results that we have to keep in mind.

Let $\mathbf{T} = (t_{ij}) \in \mathcal{P}(n)$ be a $n \times n$ tensor. We have then the following properties :

$$\mathbf{T} \text{ is symmetric} \iff \forall i, j \in [1, n], \quad t_{ij} = t_{ji}$$

$$\mathbf{T} \text{ is semi positive-definite} \iff \forall \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{x}^T \mathbf{T} \mathbf{x} \geq 0$$

Actually the meaningful informations contained in a tensor $\mathbf{T}$ can be retrieved from its *spectral decomposition* giving its eigenvectors $\mathbf{u}_k \in \mathbb{R}^n$ and its eigenvalues λ_k . These elements verify interesting properties :

$$\mathbf{T} \text{ is semi positive-definite} \iff \forall k \in [1, n], \quad \lambda_k \geq 0$$

$$\mathbf{T} \text{ is real and symmetric} \iff \forall k, l \in [1, n], \quad \mathbf{u}_k \cdot \mathbf{u}_l = \delta_{kl} = \begin{cases} 1 & (\text{if } k = l) \\ 0 & (\text{if } k \neq l) \end{cases}$$

It means that the eigenvectors $\mathbf{u}_k$ form an *orthonormal vector basis* in $\mathbb{R}^n$, and $\mathbf{T}$ may be written as :

$$\mathbf{T} = \mathbf{R} \mathbf{\Gamma} \mathbf{R}^T \tag{1.4}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{n \times n}$ is the *diagonal* matrix of the eigenvalues λ_k

$$\mathbf{\Gamma} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

and $\mathbf{R}$ is a *rotation matrix* whose columns $\tilde{\mathbf{u}}_k$ are built from the eigenvectors $\mathbf{u}_k$ of $\mathbf{T}$, such that the determinant of $\mathbf{R}$ is $\det(\mathbf{R}) = +1$.

$$\mathbf{R} = (\tilde{\mathbf{u}}_1 \mid \tilde{\mathbf{u}}_2 \mid \dots \mid \tilde{\mathbf{u}}_n) \quad \text{where} \quad \forall k = 1..n, \quad \tilde{\mathbf{u}}_k = \pm \mathbf{u}_k$$

The formulation (1.4) clearly separates the *orientation* $\mathbf{R}$ and the *diffusivities* $\mathbf{\Gamma}$ of the tensor $\mathbf{T}$. A natural graphical representation of a diffusion tensor $\mathbf{T}$ is then a n -dimensional ellipsoid whose axes and radii are respectively the eigenvectors $\mathbf{u}_k$ and the (positive) eigenvalues λ_k .

Fig.1.10 illustrates this representation for the case of 2D and 3D diffusion tensors.

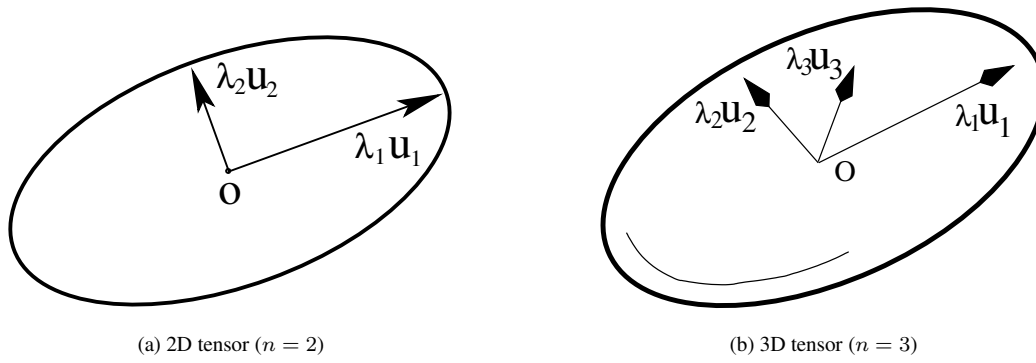


Figure 1.10: Diffusion tensor representation with ellipsoids.

Note also that developing $\mathbf{T}$ in formula (1.4) gives :

$$\mathbf{T} = \sum_{k=1}^n \lambda_k \mathbf{u}_k \mathbf{u}_k^T$$

This second formulation emphasizes the fact that $\mathbf{T}$ is the sum of *weighted elementary orthogonal tensors* ($\mathbf{u}_k \mathbf{u}_k^T$). The n eigenvalues of such elementary tensors ($\mathbf{u}_k \mathbf{u}_k^T$) are respectively :

- 0, associated to arbitrary $n - 1$ orthogonal eigenvectors $\mathbf{v}_l$ such that

$$\mathbf{v}_l \perp \mathbf{u}_k \quad (l = 1 \dots (n - 1))$$

- 1, associated to the eigenvector $\mathbf{u}_k$.

These elementary tensors can be viewed as *thin* ellipsoids with one axe of length 1 and the others of length 0. They are well designed to represent the *orientations* of the vectors $\mathbf{u}_k$, without the vector direction information. A whole tensor $\mathbf{T}$ is simply a combination of these (weighted) orthogonal orientations.

When all the n eigenvalues λ_k of a tensor $\mathbf{T}$ have the same value $\lambda > 0$, there are *no preferred diffusion directions*. Then the tensor is *isotropic*, corresponding to a weighted identity matrix :

$$\mathbf{T} = \sum_{k=1}^n \lambda \mathbf{u}_k \mathbf{u}_k^T = \lambda \mathbf{R} \mathbf{R}^T = \lambda \mathbf{Id}$$

In this case, we can't speak of a specific tensor orientation. The tensor representation is then a *sphere* with a radius λ , and the corresponding diffusion process is done with the same weight in all the directions of the space. Detailed diffusion tensor theory and applications in image processing can be found in [89, 121]

Bah oui, mais pourquoi t'as commencé une thèse, aussi ?

Lolo.

Chapter 2

State of the Art and Motivations



We get to the heart of the matter and survey the range of PDE-based algorithms for image regularization, proposed in the literature. First, we review the classical methods successfully applied for scalar image regularization. Then, we examine and compare the proposed diffusion PDE's dealing with multivalued data, including the recent works on constrained feature regularization (particularly direction vectors). All examples are illustrated and analyzed. Thus, we emphasize the open questions investigated throughout this thesis.

2.1 Regularization of scalar images with PDE's

For several years, regularization algorithms have attracted a growing interest in the computer vision community. It consists in simplifying data in a way that only interesting features are preserved (see Fig.1.4c and Fig.1.5b).

Using regularization terms $\mathcal{R}$ in PDE's formulations like $\frac{\partial I}{\partial t} = \mathcal{R}$, necessarily introduces the additional notion of *scale-space*: the data are iteratively regularized and a continuous sequence of smoother images $I_{(t)}$ is generated whereas the evolution time t goes by. A desired behavior of such regularization algorithms is that the less significant data features disappear first, while the interesting ones are preserved as long as they become unimportant themselves within the image [7, 118, 126, 143, 194]. This concept is illustrated on Fig.2.1: the woman's face contour is firstly well conserved compared to the noise, then it smoothly disappears.

Roughly speaking, regularization PDE's may be seen as *non-linear filters* that simplify the image little by little and minimize then the image variations. Note therefore that they generally don't converge towards 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. However, regularization terms constitute the key elements for solving *ill-posed* computer vision problems [91]: restoration, segmentation, registration, surface reconstruction, etc. It allows variational and PDE-based algorithms to find realistic and smooth solutions. Indeed, when modeling real world's objects on computers using curves or surfaces, one generally expects differentiable models (eventually piecewise differentiable) since these objects

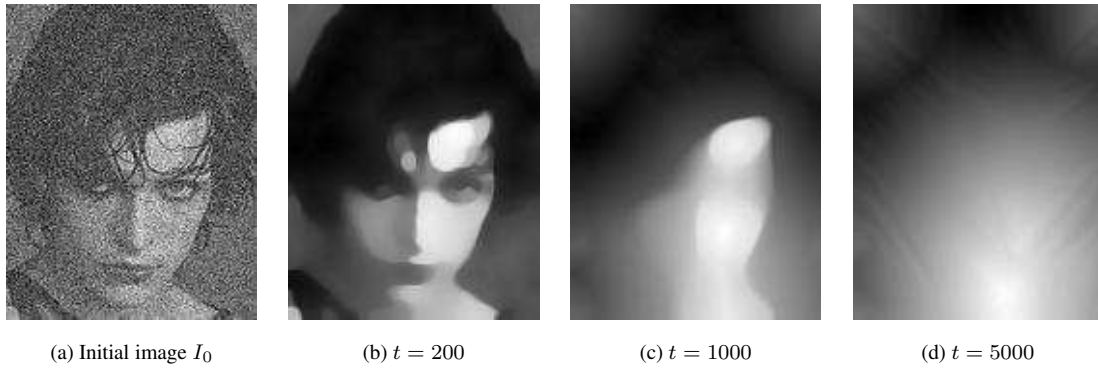


Figure 2.1: Regularization PDE's and the notion of scale-space.

are often quite regular and too many discontinuities should be avoided. This constraint is usually translated into some smoothness properties of the PDE flows, converging into solutions that are regular enough. This is the role of the *additive regularization term* $\mathcal{R}$, which may come from an additive smoothness energy term in the minimizing functional, if any.

Understanding the exact behavior of regularization PDE's has then stakes in the whole range of computer vision algorithms. For that reason, many PDE-based regularization methods acting on scalar images (and more recently on vector-valued images) have been proposed so far in the literature. In the following, we propose to describe some of the most classical ones.

Note that *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*). Basically, it avoids the expected solution at convergence to be too different from the original noisy image (not constant, by the way) (Fig.2.1).

$$\frac{\partial I}{\partial t} = \mathcal{R} + (I_{\text{noisy}} - I)$$

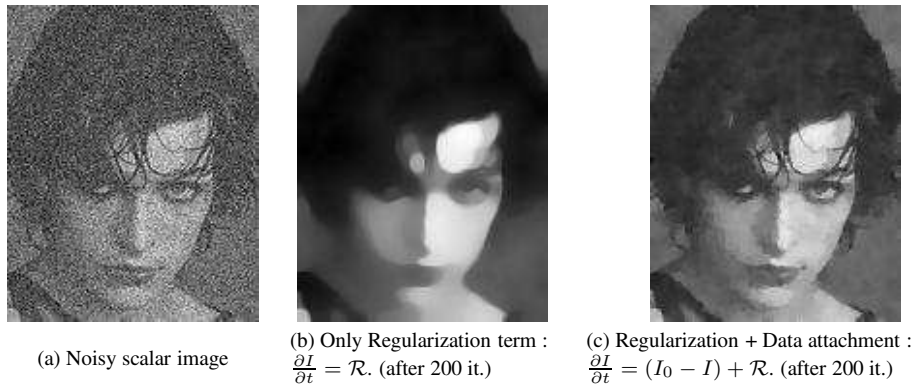


Figure 2.2: Regularization term for image restoration.

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 state of the art, we describe the different classes of PDE methods that regularize *2D images*. The matching piece to these equations for 3D volumes (or higher dimensional fields) is quite similar and can be found in the cited references.

2.1.1 Isotropic smoothing

Isotropic regularization is a natural way to smooth and simplify data and has consequently been reached by several mathematical formulations : From the restoration scheme proposed by Tikhonov in [170] to the classical linear filtering of images (in the Fourier spectral space for instance [76]), the proposed methods all lead to the same regularization behavior. Using a variational formalization, it may be understood like this :

Let $I_{\text{noisy}} : \Omega \rightarrow \mathbb{R}$ be an irregular (noisy) 2D scalar image we want to regularize. The noise is considered as high frequency variations ν with low amplitude, added to the pixels of the regular image I_{regular} .

$$I_{\text{noisy}} = I_{\text{regular}} + \nu$$

To regularize I_{noisy} , a common idea is to minimize its variations, estimated by $\|\nabla I\|$. The corresponding variational problem has been initiated in [170] as the minimization of the functional :

$$\min_{I: \Omega \rightarrow \mathbb{R}} E_{\text{Tikhonov}}(I) = \int_{\Omega} \|\nabla I\|^2 d\Omega \quad (2.1)$$

Note that the original functional in [170] also consists in an additional *data attachment term* $\|I_{\text{noisy}} - AI\|$ originally used to restore images filtered by the linear operator A . Actually, the minimum of such a functional (2.1) is known : a constant image, since then $E_{\text{Tikhonov}}(I_{\text{const}}) = 0$. Nevertheless, the important point we emphasize here is the behavior of *gradient descent of the regularization term* alone. It will be more generally the case in the whole chapters 2 and 3.

Using the Euler-Lagrange equations (1.3) gives the following gradient descent (PDE) that minimizes $E_{\text{Tikhonov}}(I)$, starting from the initial noisy image I_{noisy} .

$$\begin{cases} I_{(t=0)} &= I_{\text{noisy}} \\ \frac{\partial I}{\partial t} &= \Delta I \end{cases} \quad (2.2)$$

The regularization term ends up with the well known *heat equation*, used in physics for instance to describe heat flows through solids. This kind of PDE is called a *diffusion equation*.

Koenderink noticed in [108] that the solution of (2.2) at a particular time t is the convolution of the original image I_{noisy} with a *normalized 2D Gaussian kernel* G_{σ} of variance $\sigma = \sqrt{2t}$:

$$I_{(t)} = I_{\text{noisy}} * G_{\sigma} \quad \text{i.e.} \quad I_{(t)}(x, y) = \iint I_{\text{noisy}}(x - u, y - v) G_{\sigma}(u, v) du dv$$

with

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad \text{and} \quad \sigma = \sqrt{2t}$$

which means that the regularization is *linear* (based on a convolution).

The regularization behavior is also clear : the signal is blurred little by little in an *isotropic* way during the PDE evolution (Fig.2.3). Note that convolving an image by a Gaussian kernel is equiv-

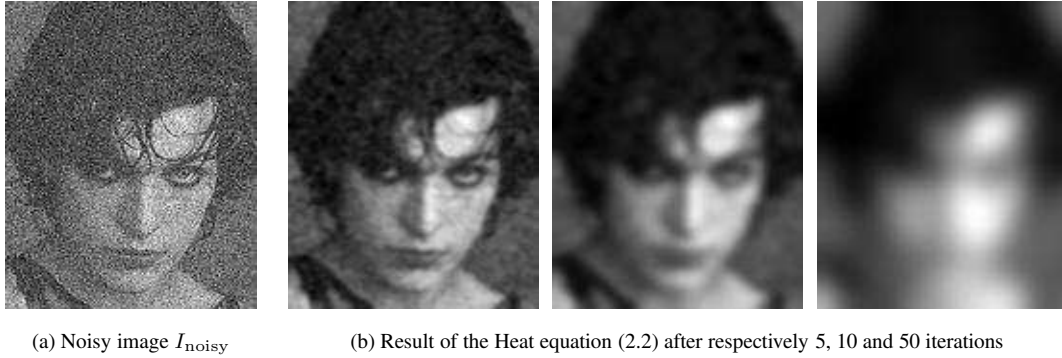


Figure 2.3: Heat equation applied on a noisy scalar image.

alent to multiply the Fourier transform of this image by another Gaussian kernel : the isotropic regularization behaves then as a *low-pass filter* suppressing high frequencies in the image I . Unfortunately, image contours are high frequency signals as well as noise. As illustrated in Fig.2.3, they are quickly blurred by such an isotropic scheme. The need to find more complex non-linear and anisotropic regularization methods has then quickly appeared (in particular for image restoration purposes).

Please note that in this thesis, we will use the term *anisotropic* as the opposite of *isotropic*, i.e to designate a regularization process that doesn't smooth the image with the same weight in all the spatial directions. In the literature, some authors have different definitions. For instance, Weickert [188] introduces the notions of *homogeneous* and *inhomogeneous* filtering, as well as different definitions for the terms *isotropic* and *anisotropic*.

2.1.2 Perona-Malik regularization

To overcome the limitations of linear methods leading to isotropic smoothing, Perona and Malik [143] proposed a nonlinear extension of the heat equation (2.2). The idea is built on the fact that the heat equation can be written in a divergence form :

$$\frac{\partial I}{\partial t} = \Delta I = \text{div}(\nabla I)$$

Adding a function $c(\|\nabla I\|)$ bounded in $[0, 1]$ in the divergence allows to control more precisely the regularization process :

$$\frac{\partial I}{\partial t} = \text{div}(c(\|\nabla I\|)\nabla I) \quad (2.3)$$

where $c : \mathbb{R} \rightarrow \mathbb{R}$ is a decreasing function vanishing on edges (high gradients) in order to stop the diffusion, and close to 1 on quite regular regions (low gradients) for an isotropic smoothing

therein. Perona-Malik proposed :

$$c(\|\nabla I\|) = \exp\left(-\frac{\|\nabla I\|^2}{K^2}\right) \quad (2.4)$$

K is a fixed gradient threshold that differentiates homogeneous areas and regions of contours.

In order to understand the exact diffusion behavior of the PDE (2.3), a specific decomposition of this equation has been proposed in [46, 112] :

$$\frac{\partial I}{\partial t} = c_\xi I_{\xi\xi} + c_\eta I_{\eta\eta} \quad (2.5)$$

where $c_\xi = c(\|\nabla I\|)$ and $c_\eta = c'(\|\nabla I\|)\|\nabla I\| + c(\|\nabla I\|)$, i.e using the proposed function (2.4) :

$$c_\xi = \exp\left(-\frac{\|\nabla I\|^2}{K^2}\right) \quad \text{and} \quad c_\eta = \exp\left(-\frac{\|\nabla I\|^2}{K^2}\right) \left(1 - 2\frac{\|\nabla I\|^2}{K^2}\right)$$

$I_{\xi\xi}$ and $I_{\eta\eta}$ denote the second derivatives of I in orthogonal directions ξ and η and can be seen as 1D oriented Laplacian :

$$I_{\xi\xi} = \frac{\partial^2 I}{\partial \xi^2} = \xi^T \mathbf{H} \xi \quad \text{and} \quad I_{\eta\eta} = \frac{\partial^2 I}{\partial \eta^2} = \eta^T \mathbf{H} \eta \quad \text{where } \mathbf{H} \text{ is the Hessian of } I.$$

The unit vectors η and ξ are respectively defined by the *gradient direction* and its *orthogonal* :

$$\eta = \frac{\nabla I}{\|\nabla I\|} \quad \text{and} \quad \xi = \eta^\perp$$

Note that ξ is *everywhere tangent to the isophote lines* $I(x, y) = a$, i.e to the contours in the image. The set (ξ, η) is then a *moving orthonormal basis* whose configuration depends on the current point coordinate $\mathbf{x} = (x, y)$ (Fig.2.4).

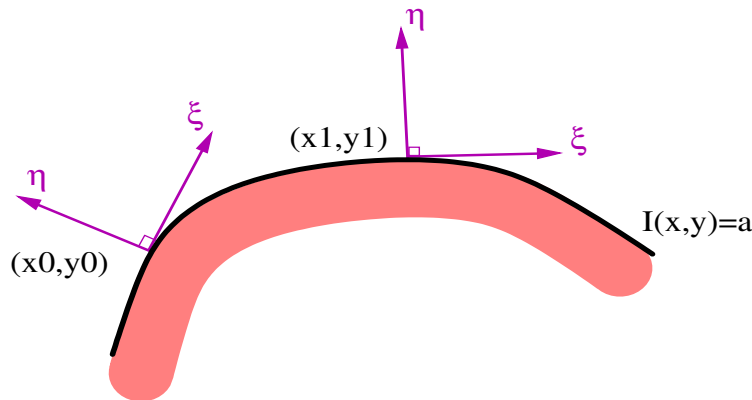


Figure 2.4: An image contour and its moving vector basis (ξ, η) .

Thus, the equation (2.5) : $\frac{\partial I}{\partial t} = c_\xi I_{\xi\xi} + c_\eta I_{\eta\eta}$ can be interpreted as *two coexistent and oriented 1D heat flows* that smooth the image respectively in the direction of the isophotes ξ with a weight c_ξ , and in the direction of the gradient η with a weight c_η .

In this respect, $(\eta, \xi, c_\eta, c_\xi)$ defines the *local diffusion geometry* of the Perona-Malik process.

As $c_\xi \geq c_\eta$, the resulting smoothing is *anisotropic*, mainly directed by the image isophotes and preserving then the image contours. A result of the Perona-Malik flow is illustrated on Fig.2.5.



Figure 2.5: Perona-Malik flow applied on a noisy scalar image.

Image features are obviously better preserved during the PDE flow (2.3) than with the isotropic one (2.2). Nevertheless, a small amount of noise doesn't vanish and is even enhanced.

Unfortunately, the function c proposed in (2.4) may lead to a *negative coefficient* c_η :

$$c_\eta = \exp\left(-\frac{\|\nabla I\|^2}{K^2}\right) \left(1 - 2\frac{\|\nabla I\|^2}{K^2}\right) \quad \text{then } c_\eta < 0, \quad \text{if } \|\nabla I\| > \frac{K}{\sqrt{2}}.$$

It conducts to an *inverse diffusion* on some image points, even for regions where $\|\nabla I\| < K$ (assumed to be a-priori quite homogeneous).

Inverse diffusion is an *unstable process* that *enhances* image features, and in this case, the noise. The Perona-Malik formulation is then ill-posed, despite the quite good experimental results obtained with this method. The complete study of its stability has been deeply investigated in [102, 189, 190] : the authors especially noticed that the classical discretization schemes used to implement the Perona-Malik flow (2.3) behave actually as *numerical stabilizers* of the method.

After all, the Perona-Malik formulation has been the starting impetus for a huge literature on scale space methods using diffusion PDE's. We will quickly review the main approaches that have significantly improved this pioneering work, using the concepts of *variational methods*, *divergence formulations* and *oriented Laplacian based expressions*.

2.1.3 Variational methods and ϕ -functionals

The idea behind regularization with variational methods is as follows. Regularizing images can be done by minimizing energy functionals measuring the global image variations. The acknowledged aim is to suppress low image variations mainly due to the noise, while preserving the high ones representing the image contours. Reference books describing data regularization with variational tools are [37, 46, 19, 122, 156, 188] among others.

The interesting framework of the ϕ -functionals gathers some of these approaches and gives the general way to proceed :

A noisy *scalar image* I_{noisy} can be regularized by minimizing the following ϕ -functional :

$$\min_{I:\Omega\rightarrow\mathbb{R}} E(I) = \int_{\Omega} \phi(\|\nabla I\|) d\Omega \quad (2.6)$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is an *increasing function*, directing the regularization behavior and penalizing high gradients. The minimization is performed via the corresponding *diffusion PDE evolution*, coming from the Euler-Lagrange equations (1.3) of $E(I)$:

$$\begin{cases} I_{(t=0)} = I_{\text{noisy}} \\ \frac{\partial I}{\partial t} = \text{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \end{cases} \quad (2.7)$$

Different choices of functions ϕ lead to different proposed regularization methods. One especially finds the expected Tikhonov and Perona-Malik schemes (Fig.2.6).

Function name	$\phi(s)$	Reference
Tikhonov	s^2	[170]
Perona-Malik	$1 - \exp(-s^2/K^2)$	[143]
Minimal surfaces	$2\sqrt{1+s^2} - 2$	[45]
Geman-McClure	$s^2/(1+s^2)$	[80]
Total Variation	s	[153]
Green	$2\log(\cosh(s))$	[90]

Figure 2.6: List of different proposed ϕ -functions.

Like the Perona-Malik flow, the local diffusion behavior of the ϕ -functionals can be analyzed from a geometric development of the equation (2.7). Indeed, the decomposition of the divergence in term of two simultaneous *oriented 1D heat flows* also applies here :

$$\frac{\partial I}{\partial t} = c_{\xi} I_{\xi\xi} + c_{\eta} I_{\eta\eta} \quad \text{with} \quad c_{\xi} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \quad \text{and} \quad c_{\eta} = \phi''(\|\nabla I\|) \quad (2.8)$$

The smoothing process is then performed along the isophote direction ξ with a weight $\phi'(\|\nabla I\|)/\|\nabla I\|$, and in the gradient direction η with a weight $\phi''(\|\nabla I\|)$.

Considering this alternative formulation (2.8), natural regularization properties should be verified by c_η and c_ξ :

- **Non-inverse diffusion** : To avoid inverse diffusion (e.g Perona-Malik flow), one wishes that the coefficients c_ξ and c_η are always positives. It means that the function ϕ should be *increasing in $\|\nabla I\|$* as well as *convex* :

$$\phi'(\|\nabla I\|) \geq 0 \quad \text{and} \quad \phi''(\|\nabla I\|) \geq 0 \quad (2.9)$$

- **Anisotropic diffusion on the edges** : Blurring the edges (high gradient points) is avoided by smoothing the image preferably in the isophote direction ξ instead of the gradient direction η . The corresponding constraints are translated as

$$\lim_{\|\nabla I\| \rightarrow +\infty} c_\eta = 0 \quad \text{and} \quad \lim_{\|\nabla I\| \rightarrow +\infty} \frac{c_\eta}{c_\xi} = 0 \quad (2.10)$$

- **Isotropic diffusion on regular regions** : In the other hand, ξ and η do not designate coherent directions of image structures on noisy homogeneous regions. The desired smoothing must behave as an isotropic one therein :

$$\lim_{\|\nabla I\| \rightarrow 0} c_\eta = \lim_{\|\nabla I\| \rightarrow 0} c_\xi = \alpha > 0 \quad (2.11)$$

Latest proposed ϕ functions in the literature follow these constraints [45, 90]. Examples illustrating the different regularization behaviors obtained with various ϕ -functions are presented on Fig.2.7. The PDE flows were stopped after a finite number of iterations (160), in order to avoid constant resulting images. Interesting survey and theoretical analysis of the ϕ -functional framework can be also found in [19, 46, 66, 110, 111].



Figure 2.7: ϕ -function based PDE's applied on a noisy scalar image.

Despite the gathering character of this framework, some diffusion behaviors may not be obtainable. For instance, the two diffusion weights c_ξ and c_η defined by (2.8) are not independent but *linked through a ϕ -function* which disables at least one degree of freedom. Fortunately, these limitations can be surpassed by *local formulations* allowing more independence to design regularization PDE's.

2.1.4 Divergence-based PDE's

One level of release has been reached in [19, 8, 35, 112, 156, 188]. The idea is to replace the function $\phi'(\|\nabla I\|)/\|\nabla I\|$ in the divergence of (2.7) by expressions depending on more appropriate image features. This gives more freedom to design regularization PDE's even if we lost the *global meaning* of the regularization process : generally, obtained equations do not correspond to a functional minimization anymore.

For instance, authors in [8] proposed to use a function $g(\|\nabla I * G_\sigma\|)$ depending on the convolved gradient norm $\|\nabla I * G_\sigma\|$, rather than simply considering $\|\nabla I\|$:

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

where

$$G_\sigma = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad \text{is a normalized 2D Gaussian kernel of variance } \sigma.$$

This has initially been done to deal with a well-posed regularization formulation. But, it allows also to respect a more coherent local diffusion geometry by involving a larger neighborhood in the computation of the local image structures that drive the smoothing process.

A major generalization of divergence-based equations has been recently proposed by Weickert [185, 186, 187, 188] : he considered image pixels as chemical concentrations diffusing with respect to some physical laws (Fick Law and continuity equations) and proposed a very generic equation :

$$\frac{\partial I}{\partial t} = \operatorname{div} (\mathbf{D} \nabla I) \tag{2.12}$$

where $\mathbf{D} : \Omega \rightarrow P(2)$ is a field of *diffusion tensors*, i.e symmetric and (semi) positive-definite 2×2 matrices (see section 1.2.4). It defines a *gradient flux* and controls then the local diffusion behavior of the process (2.12). For simplicity, we will now write $\mathbf{D}$ to designate the tensor $\mathbf{D}(\mathbf{x})$ at each point $\mathbf{x} \in \Omega$ of the tensor field.

Note that the ϕ -functional formalism is a particular case of the PDE (2.12), with

$$\mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \mathbf{Id}$$

The author rather proposed to design the diffusion tensors $\mathbf{D}$ for each image point $\mathbf{x} = (x, y)$, by selecting its two eigenvectors $\mathbf{u}, \mathbf{v}$ and eigenvalues λ_1, λ_2 as functions of the spectral elements of a *smoothed structure tensor* $\mathbf{G}_\sigma$:

$$\mathbf{G}_\sigma = (\nabla I \nabla I^T) * G_\sigma \tag{2.13}$$

Note that an alternative method proposed in [83], consists in considering the non-smoothed structure tensor of the smoothed image $I * G_\sigma$. Let us name η^*, ξ^* the eigenvectors of $\mathbf{G}_\sigma$ and μ_1, μ_2 its corresponding eigenvalues.

Note that

$$\lim_{\sigma \rightarrow 0} \xi^* = \xi = \frac{\nabla I^\perp}{\|\nabla I\|} \quad \text{and} \quad \lim_{\sigma \rightarrow 0} \eta^* = \eta = \frac{\nabla I}{\|\nabla I\|}$$

but we have generally

$$\eta^* \neq (\eta * G_\sigma) \quad \text{and} \quad \xi^* \neq (\xi * G_\sigma)$$

The directions η^* and ξ^* are smoothed versions of η and ξ but do not correspond to the gaussian smoothing of η and ξ .

Then, the spectral elements of the diffusion tensor $\mathbf{D}$ are chosen as :

$$\left| \begin{array}{l} \mathbf{u} = \eta^* \\ \mathbf{v} = \xi^* \end{array} \right. \quad \text{and} \quad \left| \begin{array}{l} \lambda_1 = \alpha \\ \lambda_2 = \begin{cases} \alpha & (\text{if } \mu_1 = \mu_2) \\ \alpha + (1 - \alpha) \exp\left(\frac{-C}{(\mu_1 - \mu_2)^2}\right) & \text{else} \end{cases} \end{array} \right. \quad (2.14)$$

($C > 0$ and $\alpha \in [0, 1]$ are fixed thresholds).

The corresponding $\mathbf{D}$ is then computed at each image point as :

$$\mathbf{D} = \lambda_1 \mathbf{u} \mathbf{u}^T + \lambda_2 \mathbf{v} \mathbf{v}^T$$

Weickert assumed that the tensor shapes at each point $\mathbf{x} = (x, y)$ of the field $\mathbf{D}$ give the preferred diffusion geometry. The idea behind the choice (2.14) is then :

- On almost constant regions, we should have $\mu_1 \simeq \mu_2 \simeq 0$ and then $\lambda_1 \simeq \lambda_2 \simeq \alpha$, i.e

$$\mathbf{D} \simeq \alpha \mathbf{Id} \quad \text{where } \mathbf{D} \text{ is the identity matrix.}$$

The tensor $\mathbf{D}$ is defined to be *isotropic* in these regions. Here, one can represent the tensor as a *circle* (2D sphere) of radius α (relations between the tensor shape and the smoothing behavior will be discussed in section 2.4).

- Along image contours, we have

$$\mu_1 \gg \mu_2 \gg 0 \quad \text{and then} \quad \lambda_2 > \lambda_1 > 0$$

The diffusion tensor $\mathbf{D}$ is then *anisotropic*, mainly directed by the smoothed direction ξ^* of the image isophotes. Here, one can represent the tensor as a *stretched ellipse* with two different radii $\mu_1 > \mu_2$.

A result of the flow (2.12) with $\mathbf{D}$ defined as in (2.14) is illustrated in Fig.2.8b. Notice how the Gaussian convolution of the structure tensor allows to create *fiber effects* by reinforcing the *local coherence* of the gradient directions. For this reason, the flow has been named *coherence enhancing flow* in the related papers [185, 186, 187, 188].

2.1.5 Diffusion PDE's based on oriented 1D Laplacians

While divergence based equations are only expressed from first derivative operators (divergence $\text{div}()$, and gradient ∇), second-derivative expressions are rather based on *oriented 1D Laplacians* :

$$\frac{\partial I}{\partial t} = c_1 I_{\mathbf{u}\mathbf{u}} + c_2 I_{\mathbf{v}\mathbf{v}} \quad (2.15)$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, $c_1, c_2 > 0$ and $\mathbf{u} \perp \mathbf{v}$. $I_{\mathbf{u}\mathbf{u}}$ and $I_{\mathbf{v}\mathbf{v}}$ denotes the second derivatives of I in the directions $\mathbf{u}$ and $\mathbf{v}$ and are formally (see section 1.2.2) :

$$I_{\mathbf{u}\mathbf{u}} = \mathbf{u}^T \mathbf{H} \mathbf{u} \quad \text{and} \quad I_{\mathbf{v}\mathbf{v}} = \mathbf{v}^T \mathbf{H} \mathbf{v} \quad \text{where } \mathbf{H} \text{ is the Hessian of } I.$$

Here, the regularization process is considered as two orthogonal and weighted 1D *oriented heat flows*, directed by the vectors $\mathbf{u}$ and $\mathbf{v}$. This is the general matching piece to the equations (2.5) and (2.8) used to interpret Perona-Malik and ϕ -functionals flows. Here, the diffusion behavior is entirely defined by the knowledge of the smoothing directions $\mathbf{u}, \mathbf{v}$ and the corresponding weights c_1 and c_2 .

For instance, the authors of [110, 111] proposed to choose

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

where $g : \mathbb{R} \rightarrow \mathbb{R}$ is a function decreasing to 0 (the diffusion vanishes on high gradients). It allows a *permanent noise removal along the edges* ξ , even on very high gradients since $c_1 = 1$ everywhere. As for low gradient regions, an isotropic smoothing is performed therein as long as g tends to 1. The corresponding equation is :

$$\frac{\partial I}{\partial t} = I_{\xi\xi} + g(\|\nabla I\|) I_{\eta\eta} \quad \text{with} \quad \lim_{\|\nabla I\| \rightarrow 0} g(\|\nabla I\|) = 1 \quad (2.16)$$

However, the constant diffusion along ξ has an effect of over-smoothing the sharp corners.

The general formulation (2.15) allows to find other well-known equations, as *the mean curvature flow*, obtained with $c_1 = 1$, $c_2 = 0$, $\mathbf{u} = \xi$ and $\mathbf{v} = \eta$ [66] :

$$\frac{\partial I}{\partial t} = I_{\xi\xi} \quad (2.17)$$

These two last equations (2.16) and (2.17) are applied on a noisy image in Fig.2.8c,d.

Note that with the formulation (2.15), the ϕ -functional framework (2.7) corresponds to

$$\begin{cases} \mathbf{u} = \xi \\ \mathbf{v} = \eta \end{cases} \quad \text{and} \quad \begin{cases} c_1 = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \\ c_2 = \phi''(\|\nabla I\|) \end{cases}$$

As we may notice, there is also a close link between oriented Laplacians and diffusion tensors. Actually, second derivatives expressions like (2.15) are based on two smoothing orientations $\mathbf{u}$

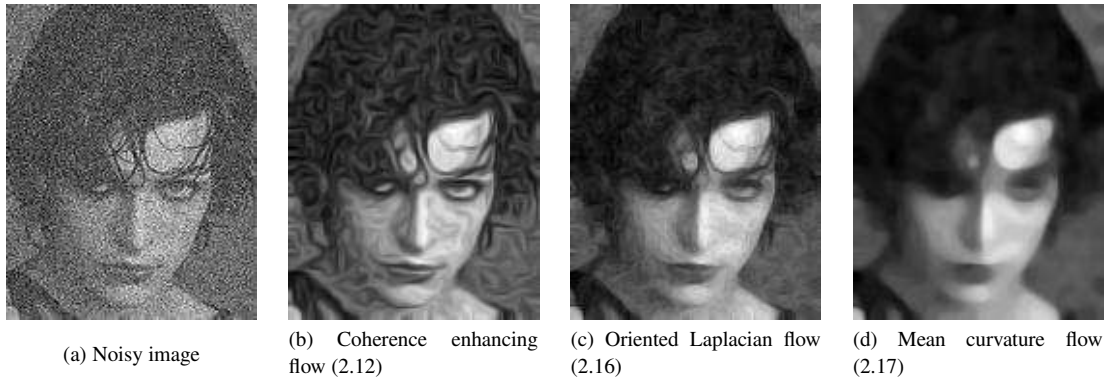


Figure 2.8: Locally designed PDE's applied on a noisy scalar image.

and $\mathbf{v}$, as well as two diffusion weights c_1 and c_2 . Diffusion tensors are well adapted to represent such geometric parameters (see section 1.2.4) and can be used to write the equation (2.15) in a more convenient form :

$$\begin{aligned} \frac{\partial I}{\partial t} &= c_1 I_{\mathbf{u}\mathbf{u}} + c_2 I_{\mathbf{v}\mathbf{v}} \\ &= \text{trace}(\mathbf{T}\mathbf{H}) \end{aligned} \quad (2.18)$$

where

$$\mathbf{T} = c_1 \mathbf{u}\mathbf{u}^T + c_2 \mathbf{v}\mathbf{v}^T$$

is the 2×2 symmetric matrix whose eigenvalues are c_1 and c_2 and respective eigenvectors are $\mathbf{u}$ and $\mathbf{v}$. More discussions about the relation between the formulations (2.12) and (2.18), both written with a diffusion tensor can be found in section (2.4) and chapter 3.

Finally, we cannot conclude this state of the art on regularization methods acting on scalar images, without referring the reader to the following papers and books, which propose other interesting formalisms and viewpoints on image regularization. We can cite for instance the works of Kimmel, Sochen & Malladi for scalar images [165] (we will go back on this interesting formulation for the case of vector-valued images), but also Black & Sapiro [28], Caselles & Morel [35], Cottet & Germain [59], Ng [125], Nikolova [127, 128], Krim [92], etc.

2.2 Regularization of multivalued images with PDE's

Recently and thanks to increased computer memory capacities, the problem of regularizing images of *vector-valued features* has become an active research topic. Of course, the large number of related applications explains this sudden enthusiasm. Actually, vector-valued images may be encountered with a lot of different and interesting datasets. Here are some of them, with a quick link to reference papers :

- *Digital color images* : a color pixel may be seen as a 3D vector (R, G, B) and color images can be likened to vector fields. Vector-valued regularization flows allow noise removal in color images [31, 104, 157, 165, 184]. Moreover, these vector-valued PDE's may be used to fill undesired holes in color images allowing nonlinear interpolation schemes. This process, commonly named *image inpainting*, is very interesting to assist image restoration processes [26, 41].
- *Optical flow and Direction fields* : Optical flows algorithms lead to 2D or 3D vector fields representing pixel motions between two images. Thus, adapted vector-valued regularization terms may be used for better estimation of optical flows. In the particular *constrained* case, vector-valued regularization PDE's can also successfully deal with fields of *unit vectors*, also called *direction fields* [39, 106, 142, 168, 167].
- *Fields of diffusion tensors* : DT-MRI images are quite noisy *matrix-valued* fields encountered in medical imaging. Regularizing such images is a challenging issue to better understand the anatomical structure of the brain [61, 48, 146, 181]. This will be considered in details in chapters 5 and 6.

As the PDE framework has been powerful enough to deal with the regularization of scalar images (section 2.1), many authors proposed to extend the existing scalar schemes to regularize noisy *vector-valued* 2D images as

$$\mathbf{I}_{\text{noisy}} : \Omega \rightarrow \mathbb{R}^n \quad \text{where } \Omega \subset \mathbb{R}^2$$

In this section, we review the principal algorithms proposed in this field and illustrate their applications on color test images ($n = 3$) with synthetic noise. Of course, these vector-valued schemes also apply for the particular case when $n = 1$ (scalar images) and often reduce then to their scalar equivalents. Note also that we first consider the problem of *unconstrained* vector-valued datasets. This means for instance that no links between vector components are taken into account. A survey of the recent *constrained* regularization problem using PDE's will be proposed in section 2.3.

2.2.1 Channel by channel regularization

A very common idea to restore vector-valued images is to use classical scalar diffusion PDE's on each channel I_i of a noisy image $\mathbf{I}_{\text{noisy}}$. But one fastly notices that this scheme is useless, since each image channel I_i evolves independently with *different smoothing geometries*.

In particular, the image isophotes have different directions $\xi_i = \nabla I_i^\perp / \|\nabla I_i\|$ for each vector components I_i . Even if the ξ_i are relatively close between each-others, the diffusion process will

not behave in a coherent way and a high risk of *vector components blending* (that will blur the edges) may occur.

This is very clear for the cartoon picture in Fig.2.9 : For this kind of images, the channel isophotes ξ_i are mainly in the same direction in each channel of the clear image $\mathbf{I}_{\text{clear}}$ (since a cartoon image is almost a piecewise constant function). But, we added a non-Gaussian additive noise on $\mathbf{I}_{\text{clear}}$ that highly perturbed these direction ξ_i , independently in each image channel (Fig.2.9a). As a result, the channel by channel diffusion smooths the image contours faster (Fig.2.9b) than with a vector-valued PDE (Fig.2.9c), in a way that could be assimilated to *color blending*.

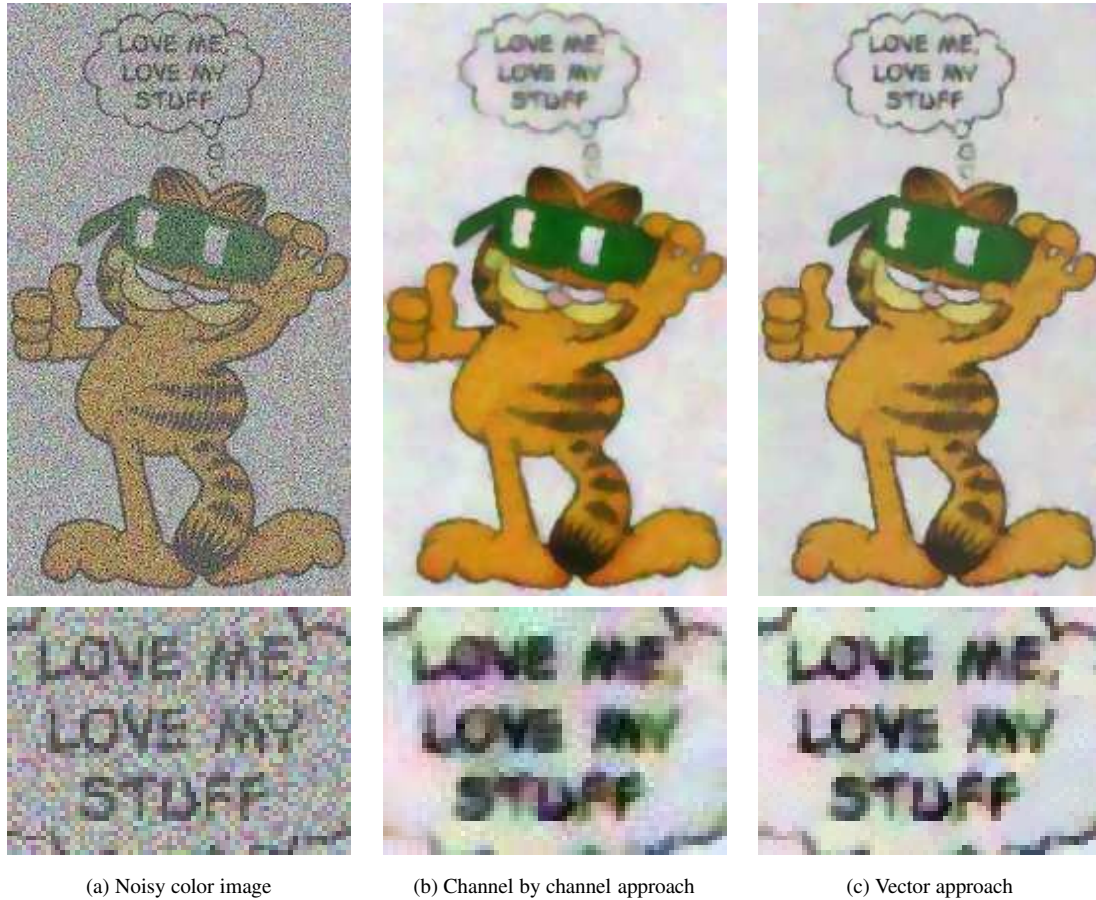


Figure 2.9: Channel by channel approach vs Vector-valued PDE's.

2.2.2 Defining a vector geometry

To avoid this blending effect, the regularization process must be driven in a common and coherent way for all the vector image channels I_i . As we noticed in section 2.1, the diffusion geometry is defined from the *local geometry of the image structures*. This is done intrinsically (by variational methods), by geometry-adapting diffusion tensors or oriented Laplacian.

These three definition are not always equivalent, but they designate the important local parameters of the regularization schemes. Extending scalar algorithms to the multivalued case needs to define

such *local vector geometries* that describe vector-valued image variations and structures. It has to be defined on each point $\mathbf{x} = (x, y) \in \Omega$ by :

- A *vector gradient norm* $\mathcal{N}$ that detects edges and corners when its value becomes high. $\mathcal{N}$ should naturally reduce to $\|\nabla I\|$ for scalar images (when $n = 1$).
- Two corresponding *variation orientations* θ_+ and θ_- that are respectively orthogonal and tangent to the vector edges, if any. In the same way, they should have the desired property of reducing to the gradient and isophote directions for scalar images :

$$\text{if } n = 1, \quad \theta_- = \xi = \frac{\nabla I^\perp}{\|\nabla I\|} \quad \text{and} \quad \theta_+ = \eta = \frac{\nabla I}{\|\nabla I\|}$$

Note that in the general case when $n > 1$, we can't talk about *vector isophotes* : Contrary to scalar images, paths of constant (vector-valued) intensity may not exist.

In order to construct such a vector geometry, different approaches can be considered :

1. Computing the vector geometry from a reduced scalar version of the vector image :

One method would be to compute first a scalar image $f(\mathbf{I})$, using a function

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

that could models the *human perception* of vector edges, if it exists. It is particularly conceivable for color images : One could choose the lightness function (perceptual response to the luminance) coming from the *CIELAB* color base [58, 147] :

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

and $g : \mathbb{R} \rightarrow \mathbb{R}$ is defined by

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

Then we may define the local vector geometry as

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

However, there are mathematically no functions f that can detect all possible vector variations. For instance, the luminance function defined above is not able to detect *iso-luminance* vector contours. It is the case on Fig.2.10 : the contours inside the colored yin-yang symbol won't be detected by $\mathcal{N} = \|\nabla f(\mathbf{I})\|$, since $f(\mathbf{I})$ is constant therein.

Here, the diffusion is going to be isotropic and the existing color edges inside the yin-yang will not be preserved.

Note that if your printer converts colors to gray-levels using the luminance or lightness function, you won't be able to see correctly the color image in Fig.2.10d.

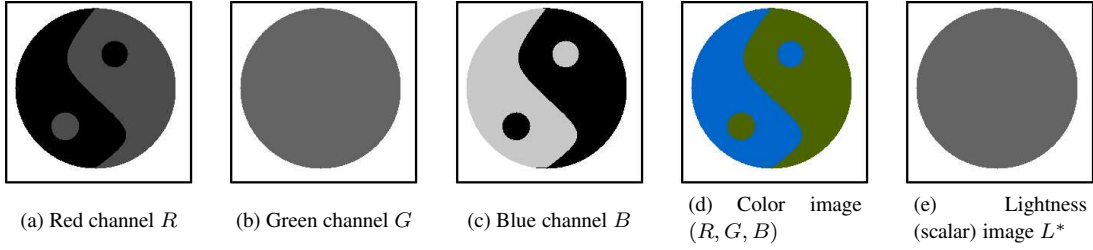


Figure 2.10: Color image with iso-luminance pixels.

2. Di Zenzo multivalued geometry :

In order to surpass this limitation, another solution has been proposed by Di Zenzo in [196]. He considers a multivalued image $\mathbf{I}$ as a $2D \rightarrow n$ -D vector field, and looks for the local variations of the vector norm $\|d\mathbf{I}\|^2$, mainly given by a variation matrix $\mathbf{G} = (g_{i,j})$.

we get :

$$d\mathbf{I} = \mathbf{I}_x dx + \mathbf{I}_y dy \quad (\in \mathbb{R}^n)$$

then

$$\begin{aligned} \|d\mathbf{I}\|^2 &= d\mathbf{I}^T d\mathbf{I} \\ &= \|\mathbf{I}_x\|^2 dx^2 + 2 \mathbf{I}_x^T \mathbf{I}_y dx dy + \|\mathbf{I}_y\|^2 dy^2 \end{aligned}$$

i.e.

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

Note that $\mathbf{G}$ is the sum of the *scalar structure tensors* $\nabla I_i \nabla I_i^T$ of each image channel I_i , as defined in section 2.1.4. It can be named itself a *vector-valued structure tensor*. Indeed, it is easy to see that $\mathbf{G}$ is symmetric as well as semi positive-definite. Its coefficients are :

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

From now on, we will equally use the terms *Di Zenzo matrix* or *structure tensor* to designate this matrix $\mathbf{G}$.

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

$$\mathbf{G} = \begin{pmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{pmatrix} \quad (2.19)$$

The interesting point about $\mathbf{G}$ is that its positive eigenvalues $\lambda_{+/-}$ are *the maximum and the minimum of $\|d\mathbf{I}\|^2$* while the orthogonal eigenvectors θ_+ and θ_- are the corresponding

variation orientations, and are formally given by :

$$\lambda_{+/-} = \frac{g_{11} + g_{22} \pm \sqrt{\Delta}}{2} \quad (2.20)$$

and

$$\theta_{+/-} \parallel \begin{pmatrix} 2 g_{12} \\ g_{22} - g_{11} \pm \sqrt{\Delta} \end{pmatrix}$$

where $\Delta = (g_{11} - g_{22})^2 + 4 g_{12}^2$.

With this simple and efficient approach, Di Zenzo opened a natural way to deal with the local vector geometry of vector-valued images, through the use of the *oriented orthogonal basis* (θ_+, θ_-) and the *variations measures* $\lambda_{\pm}$. These eigenvalues are indeed well adapted to discriminate different geometric cases :

1. If $\lambda_+ \simeq \lambda_- \simeq 0$, there are very few vector variations around the current point $\mathbf{x} = (x, y)$: the region is *almost flat* and doesn't contain any edges or corners (it is the case for the inside of the strips in Figure 2.11a). For this configuration, the variation norm $\mathcal{N}$ we have to define should be low.
2. If $\lambda_+ \gg \lambda_-$, there are a lot of vector variations. The current point may be located on a *vector edge* (it is the case for the edges of the strips in Figure 2.11a). For this configuration, the variation norm $\mathcal{N}$ should be high.
3. If $\lambda_+ \simeq \lambda_- \gg 0$, we are located on a *saddle point of the vector surface*, which can possibly be a *vector corner* in the image (the intersections of the strips in Figure 2.11a). In this case $\mathcal{N}$ should be even higher than the case 2. Regularization algorithms have indeed a tendency to smooth corners fastly. A very high variation estimated on corner points would attenuate the smoothing there, which is a desired effect.

Actually, proposed regularization algorithms acting on multivalued images are implicitly or explicitly based on these Di Zenzo's attributes. In particular, three different choices of vector gradient norms $\mathcal{N}$ have been proposed so far in the literature :

1. $\mathcal{N} = \sqrt{\lambda_+}$, as a natural extension of the scalar gradient norm viewed as *the value of maximum variations* [30, 154, 155] (Figure 2.11b).
2. $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$, also called *coherence norm*, have been chosen in [157, 181, 184]. Note that this norm fails to detect discontinuities that are saddle points of the vector-valued surface (Figure 2.11c). This may perturb regularization processes since certain sharp corners will be considered as quite homogeneous regions and will be probably smoothed.
3. $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$, also denoted by $\|\nabla \mathbf{I}\|$ is often chosen [25, 31, 140, 167, 175, 176] since it detects edges and corners in a good way, and is easy to compute (doesn't require an eigenvalue decomposition of $\mathbf{G}$) :

$$\mathcal{N}_+ = \|\nabla \mathbf{I}\| = \sqrt{\text{trace}(\mathbf{G})} = \sqrt{\sum_{i=1}^n \|\nabla I_i\|^2} \quad (2.21)$$

Moreover, the norm $\mathcal{N}_+$ has the property of giving preferences to certain corners (Figure 2.11d). This is very interesting for image restoration purposes, since the smoothing will be attenuated on these points.

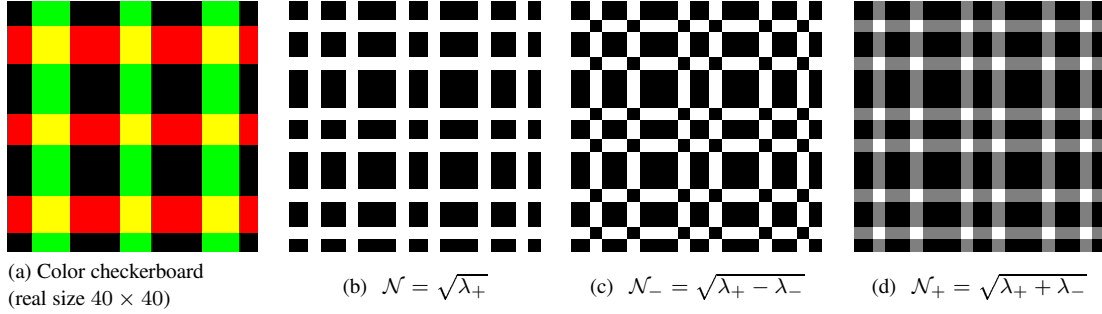


Figure 2.11: A list of possible vector variation norms.

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

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

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

$$\begin{cases} \theta_-^1 = \xi = \frac{\nabla I^\perp}{\|\nabla I\|} \\ \theta_+^1 = \eta = \frac{\nabla I}{\|\nabla I\|} \end{cases} \quad \text{associated to} \quad \begin{cases} \lambda_-^1 = 0 \\ \lambda_+^1 = \|\nabla I\|^2 \end{cases}$$

It particularly means that the three above defined norms $\mathcal{N}_+$, $\mathcal{N}_-$ and $\mathcal{N}$ all reduce to $\|\nabla I\|$ in the scalar case.

Once a local vector geometry is defined, we can use it as a measure in many image analysis processes acting on vector images (and not only for regularization schemes).

For instance, color edge detection can be performed by finding the local maxima of $\mathcal{N}$ in the θ_+ direction (Figure 2.12 and [114, 174, 177]). This vector geometry computation has also been integrated for color image segmentation purposes in [154, 155].

In the following sections, we will analyze the recently proposed diffusion PDE's acting on vector-valued images. We will use the previous notations ξ, η to refer to the *scalar local geometry* (see section 2.1.2), and $\theta_\pm, \lambda_\pm$ and the matrix $\mathbf{G}$ to refer to the Di Zenzo's *vector local geometry*. This survey ends up with two comparative figures Fig.2.13 and Fig.2.14 illustrating the different methods applied on synthetic and real color images.

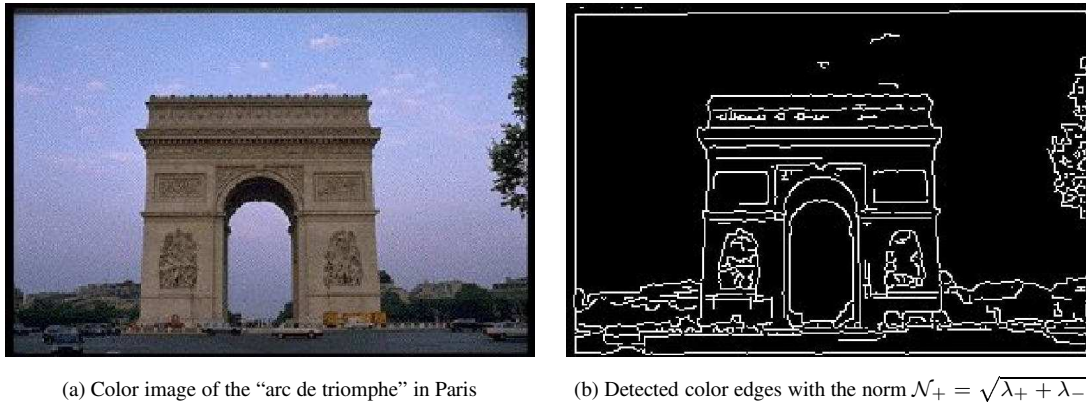


Figure 2.12: Using vector variation norms for color edge detection.

2.2.3 Color Total Variation

In order to regularize color images ($n = 3$), Blomgren and Chan [31] proposed to minimize a measure of a *Color Total Variation* CTV_n . This is a variational formulation that is the vector-valued matching piece of the *total variation* formalism, largely used to regularize scalar images (it corresponds to the ϕ -functional framework with $\phi(s) = s$, see section 2.1.3).

The color total variation has been formulated as the minimization :

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} CTV_n(\mathbf{I}) = \sqrt{\sum_{i=1}^n \left[\int_{\Omega} \|\nabla I_i\| d\Omega \right]^2}. \quad (2.22)$$

Minimizing the CTV_n leads to the following vector-valued diffusion PDE's (written in a component by component style), coming from the Euler-Lagrange equations of (2.22) :

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} = \frac{\int_{\Omega} \|\nabla I_i\|}{CTV_n(\mathbf{I})} \operatorname{div} \left(\frac{\nabla I_i}{\|\nabla I_i\|} \right) \end{cases} \quad (i = 1..n) \quad (2.23)$$

In order to understand the behavior of this regularization equation, we can decompose (2.23) and express it with directional 1D-Laplacian, as we did in section 2.1.3 for the ϕ -functional approach. Actually, if we introduce the isophote directions $\xi_i = \nabla I_i^\perp / \|\nabla I_i\|$ in the channels I_i , the PDE (2.23) writes :

$$\frac{\partial I_i}{\partial t} = A_i \frac{I_{\xi_i \xi_i}}{\|\nabla I_i\|} \quad \text{where} \quad A_i = \frac{\int_{\Omega} \|\nabla I_i\|}{TV_n(\mathbf{I})}$$

The interpretation is immediate. the diffusion is similar to a *channel by channel Total Variation*, weighted by a coupling term A_i which is *constant for a whole channel* I_i . Each channel is more or less denoising depending of its own total variation $\int_{\Omega} \|\nabla I_i\|$.

It means that *no local vector interactions* are considered. Noisy vector edges that do not clearly appear in each image channel I_i may be smoothed, exactly as the channel by channel diffusion behaves.

Note also that the smoothing is always done along the isophote directions ξ_i and never along the gradient direction η_i . It doesn't allow isotropic diffusion in noisy homogeneous regions. Surprisingly, we notice anyway that the regularization obtained in flat regions behaves almost as *isotropic* ones (Fig.2.13b). This remarkable property is due to the *color blending effect* which is the peculiarity of decoupled diffusion methods : each channel I_i diffuses in such different directions ξ_i in flat regions that it leads to *isotropic-like* diffusion.

Besides, the main noticeable point of the CTV_n method is that minimizing a vector coupled functional *does not necessarily lead* to a PDE that implicitly considers a *local vector geometry*.

2.2.4 Vector Φ -functionals

Another natural extension of variational methods for multivalued image regularization is the one proposed in [154, 155, 161, 168]. The general idea is to consider that the image variations are given by a *vector gradient norm*, as $\mathcal{N}$, $\mathcal{N}_+$ or $\mathcal{N}_-$, defined in section 2.2.2.

Then, one minimizes a functional measuring global image variations :

$$\min_{\mathbf{I}:\Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \phi(\mathcal{N}_{\pm}) d\Omega \quad (2.24)$$

More particularly, the cited authors focused on the norm $\mathcal{N}_+ = \|\nabla \mathbf{I}\|$:

$$\|\nabla \mathbf{I}\| = \mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-} = \sqrt{\sum_{i=1}^n \|\nabla I_i\|^2} \quad (2.25)$$

and minimize then the following ϕ -functional :

$$\min_{\mathbf{I}:\Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \phi(\|\nabla \mathbf{I}\|) d\Omega$$

This is performed via the corresponding Euler-Lagrange multivalued equations :

$$\begin{cases} \mathbf{I}_{(t=0)} &= \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} &= \text{div} \left(\frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|} \nabla I_i \right) \quad (i = 1..n) \end{cases} \quad (2.26)$$

This is a very similar expression to the ϕ -functional PDE (2.7) used for scalar image regularization. Unfortunately, the local geometric behavior of the equation (2.26) is not as easily understandable as in the scalar case. Indeed, the raw development of (2.26) in term of second derivatives introduces additional *vector coupling* terms which hampers the diffusion interpretation. Analyzing this local diffusion comportment is one of the aims of chapter 3.

2.2.5 Coherence Enhancing Diffusion

In [184, 186, 188], Weickert proposed to extend his coherence enhancing scheme (2.12), in order to deal with vector-valued images, by choosing a *common diffusion tensor* $\mathbf{D}$ for all image channels I_i :

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) \end{cases} \quad (i = 1..n) \quad (2.27)$$

where $\mathbf{D} = \lambda_1 \mathbf{u}\mathbf{u}^T + \lambda_2 \mathbf{v}\mathbf{v}^T$ is constructed such that its spectral elements λ_1, λ_2 (eigenvalues) and $\mathbf{u}, \mathbf{v}$ (eigenvectors) take the Di Zenzo's local vector geometry into account ($\mathbf{D}$ is *pointwise* defined). Rather than defining $\mathbf{D}$ directly from the eigenvalues/eigenvectors of the structure tensor $\mathbf{G}$, the author uses a smoothed version $\mathbf{G}^*$ of $\mathbf{G}$:

$$\mathbf{G}^* = \left(\sum_{i=1}^n \nabla I_i \nabla I_i^T \right) * G_\sigma \quad \text{where} \quad G_\sigma = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

This is indeed a very good idea allowing to retrieve a more *coherent local vector geometry*.

The spectral elements of $\mathbf{D}$ are then chosen in a similar way to the scalar case (2.14) :

$$\left| \begin{array}{l} \mathbf{u} = \theta_+^* \\ \mathbf{v} = \theta_-^* \end{array} \right. \quad \text{and} \quad \left| \begin{array}{l} \lambda_1 = \alpha \\ \lambda_2 = \begin{cases} \alpha & (\text{if } \lambda_+^* = \lambda_-^*) \\ \alpha + (1 - \alpha) \exp\left(\frac{-C}{(\lambda_+^* - \lambda_-^*)^2}\right) & \text{else} \end{cases} \end{array} \right. \quad (2.28)$$

where the $\lambda_\pm^*$ and $\theta_\pm^*$ are the eigenvalues and corresponding eigenvectors of the smoothed structure tensors $\mathbf{G}^*$.

The assumption is that the diffusion tensors give the general diffusion geometry of the PDE (2.27). Following the proposal (2.28) we notice that the local vector geometry is detected using a smoothed version $\mathcal{N}_-^* = \sqrt{\lambda_+^* - \lambda_-^*}$ of the *coherence norm* $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$:

- On flat regions the diffusion tensor is almost isotropic. It occurs when

$$\mathcal{N}_-^* \simeq 0 \quad \implies \quad \mathbf{D} \simeq \alpha \mathbf{Id}$$

- Near the “edges” ($\mathcal{N}_-^* \gg 0$), the diffusion tensor is anisotropic, mainly directed by the vector contours θ_-^* since $\lambda_2 \gg \lambda_1$, but also in the direction θ_+ with a constant weight α . Remind that the norm $\mathcal{N}_-$ may be inadequate for certain image corners (Fig.2.11).

Note that the vector ϕ -functional formulation (2.25) is a particular case of the divergence equation (2.27) with the following isotropic diffusion tensor field $\mathbf{D}$:

$$\mathbf{D} = \frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|} \mathbf{Id}$$

where $\mathbf{Id}$ is the identity matrix. As we will see in further sections, it doesn't mean that the corresponding regularization process behaves as an isotropic one. Note that like the scalar case, a divergence-based expression with a generic diffusion tensor $\mathbf{D}$ may not correspond to a known variational formulation.

2.2.6 The Beltrami Flow

With a variational approach, different from the ϕ -functional one, Kimmel, Sochen and Malladi [104, 164, 165] found another interesting particular case of the divergence-based equation (2.27). Considering a vector-valued image $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$ as a *2D surface embedded in a $(n + 2)D$ space* :

$$(x, y) \longrightarrow (x, y, I_1(x, y), I_2(x, y), \dots, I_n(x, y)) ,$$

the authors proposed to minimize this generic *Polyakov action* :

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \sigma \sqrt{g} g^{\mu\nu} \partial_{\mu} X^i \partial_{\nu} X^j h_{ij}(\mathbf{X}) d\Omega \quad (2.29)$$

The functional (2.29) is actually a physical measure of the surface area, and can be written more simply for our case of vector-valued images, defined on a flat 2D domain Ω :

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \sqrt{\det(\mathbf{Id} + \mathbf{G})} d\Omega$$

where $\mathbf{G}$ is the Di Zenzo matrix defined in (2.19) and $\mathbf{Id}$ is the identity matrix. We notice that $\det(\mathbf{Id} + \mathbf{G}) = (1 + \lambda_+)(1 + \lambda_-)$, i.e the minimization problem is

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \sqrt{(1 + \lambda_+)(1 + \lambda_-)} d\Omega$$

The corresponding gradient descent, with respect to the surface metric $(\mathbf{Id} + \mathbf{G})$, ends up in the following diffusion PDE called *Beltrami Flow* :

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} = \frac{1}{\sqrt{\det(\mathbf{Id} + \mathbf{G})}} \operatorname{div} \left(\sqrt{\det(\mathbf{Id} + \mathbf{G})} (\mathbf{Id} + \mathbf{G})^{-1} \nabla I_i \right) \quad (i = 1 \dots n) \end{cases} \quad (2.30)$$

This flow is minimizing the global area of the surface representing the vector-valued image, with respect to the surface metric. The resulting equation (2.30) is then a *weighted version* of the coherence enhancing diffusion (2.27), with the following anisotropic diffusion tensor :

$$\mathbf{D} = \sqrt{\det(\mathbf{Id} + \mathbf{G})} (\mathbf{Id} + \mathbf{G})^{-1}$$

One can easily verify that the spectral elements $\lambda_{1/2}$ (eigenvalues) and $\mathbf{u}, \mathbf{v}$ (eigenvectors) of this diffusion tensor $\mathbf{D}$ can be expressed from the Di Zenzo's geometry attributes :

$$\left| \begin{array}{l} \lambda_1 = \sqrt{\frac{1 + \lambda_-}{1 + \lambda_+}} \\ \lambda_2 = \sqrt{\frac{1 + \lambda_+}{1 + \lambda_-}} \end{array} \right. \quad \text{and} \quad \left| \begin{array}{l} \mathbf{u} = \theta_+ \\ \mathbf{v} = \theta_- \end{array} \right.$$

The diffusion tensor shape is then adapting with respect to the local vector geometry of the image, using implicitly the *coherence norm* $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ to discriminate flat regions and contours :

- When $\lambda_+ \simeq \lambda_-$ (i.e $\mathcal{N}_- \simeq 0$), the current point is supposed to be on a quite flat region, the tensor $\mathbf{D}$ becomes isotropic, i.e $\lambda_1 \simeq \lambda_2 \simeq 1$.
- Near edges $\lambda_+ \gg \lambda_-$ (i.e $\mathcal{N}_- \gg 0$), the diffusion tensor is mainly directed by the vector edge direction θ_- .

As for the general weighting term of the equation (2.30),

$$\frac{1}{\sqrt{\det(\mathbf{Id} + \mathbf{G})}} = \frac{1}{\sqrt{(1 + \lambda_+)(1 + \lambda_-)}}$$

It quickly decreases the amount of diffusion near high vector variations, and vector edges are preserved for a long time during the flow (and unfortunately noisy sharp edges too).

2.2.7 Vector $\mathbf{I}_{\xi\xi}$ Diffusion

In [157], Ringach and Sapiro proposed an extension of the mean curvature equation (2.17) $I_t = I_{\xi\xi}$ for vector-valued images, using an oriented Laplacian. Conversely to the previous methods, their equation is directly based on a *second-derivative expression*, unlike variational or divergence based approaches. They naturally used the Di Zenzo vector geometry to design this vector-valued regularization PDE :

$$\frac{\partial \mathbf{I}}{\partial t} = g(\lambda_+ - \lambda_-) \mathbf{I}_{\theta_- \theta_-} \quad (2.31)$$

where $g : \mathbb{R} \rightarrow \mathbb{R}$ is a positive decreasing function, avoiding the smoothing of high gradients regions. Note that the coherence norm $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ is also used here, in order to discriminate homogeneous regions and image contours.

It was one of the first attempts to construct an oriented Laplacian vector PDE directly from a local vector geometry viewpoint. At a given point, all channels I_i are smoothed along a *common vector edge direction* with a *common intensity*. Despite this great designing, some drawbacks subsist :

- The coherence norm $\mathcal{N}_- = \sqrt{\lambda_+ - \lambda_-}$ is not always adapted to detect certain vector corners (Figure 2.11d).
- In flat regions ($\mathcal{N}_- \rightarrow 0$), the diffusion is made along a single direction θ_- , which is mainly directed by the noise since no coherent structures exist in these regions. *Texture effects* may result from this mono-directional smoothing. This is particularly true since contrary to decoupled regularizations, vector components are *not blended* with this method (the diffusions in all image channels I_i follow a common direction).

We may also notice that the regularizing flow (2.31) can be written using a *trace operator* coupled with a *diffusion tensor*, exactly as the case of oriented Laplacians for scalar images (see section 2.1.5) :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i) \quad \text{where} \quad \mathbf{T} = g(\lambda_+ - \lambda_-) \theta_- \theta_-^T$$

This will be more particularly discussed in the open questions (section 2.4).

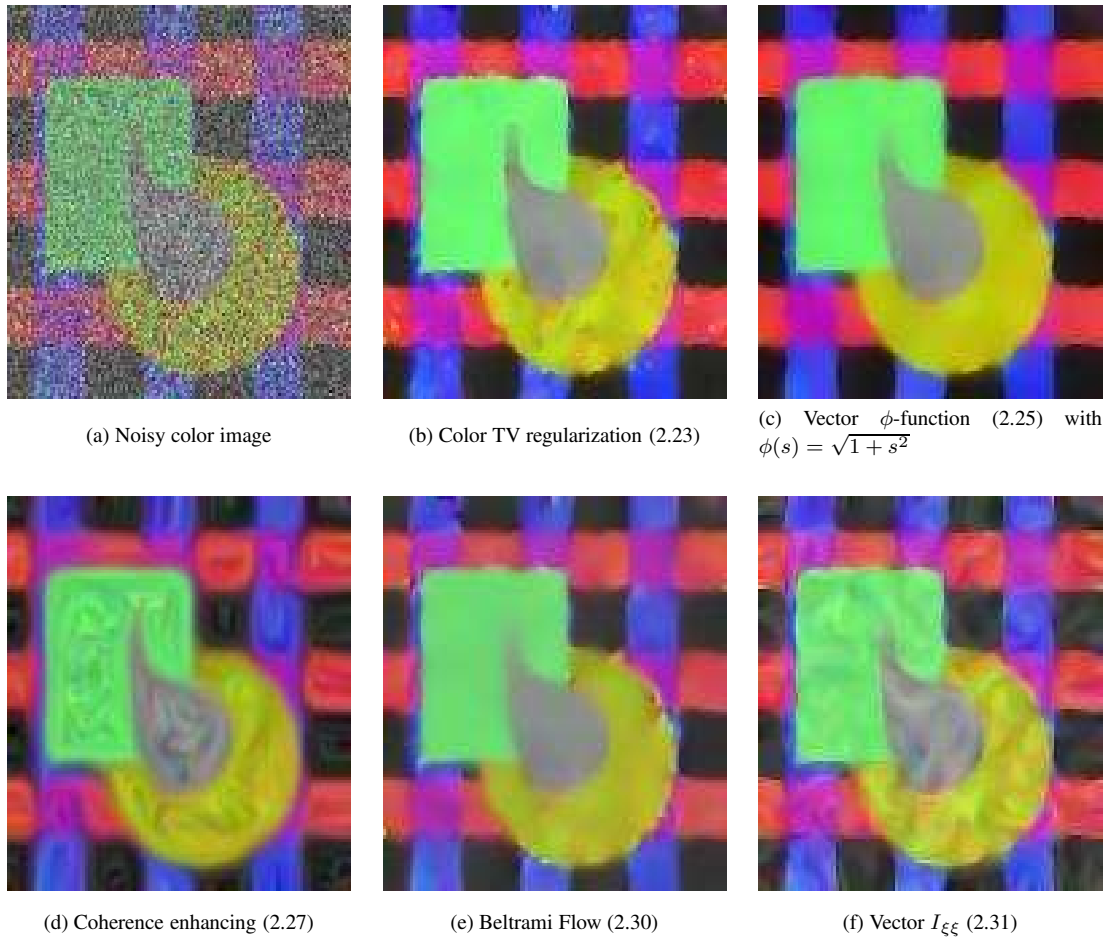


Figure 2.13: Comparison of classical vector-valued regularization PDE's (I).

2.2.8 Comparative figures

We close our survey on multivalued regularization PDE's with a comparative figure, intended to highlight the different compartments of the (proposed above) equations. Obviously, the aim is not to discriminate bad and good methods since each approach is more or less adapted to certain image configurations. We rather want to put the accent on the use of a local vector geometry in the equations and its effect on the obtained results.

The experiments have been carried out with the following protocol :

- **Types of images** : We applied the different regularization algorithms (CTV, Vector Φ -functional, Coherence enhancing, Beltrami flow, Vector $I_{\xi\xi}$) on a synthetic color image (128×128) and a real color photograph (290×290). Both images are initially clear but we added artificial noise into them. The test images and the PDE evolutions are considered in the (R, G, B) color space.
- **Model of noise** : The added noise is high intensity variations and is a mix of several different stochastic processes :

1. Correlated and uncorrelated uniform noises :

$$\forall \mathbf{x} \in \Omega, \quad \begin{cases} R(\mathbf{x}) = R(\mathbf{x}) + \delta_R + \delta \\ G(\mathbf{x}) = G(\mathbf{x}) + \delta_G + \delta \\ B(\mathbf{x}) = B(\mathbf{x}) + \delta_B + \delta \end{cases}$$

where $\delta_R, \delta_G, \delta_B$ and δ are four different *uniform* distributions.

2. A Gaussian noise on the vector directions and norms :

The principle is to decompose each vector $\mathbf{I} = (R, G, B)^T$ into its orientation part $\vec{i}$ and its norm $\|\mathbf{I}\|$:

$$\vec{i} = \frac{1}{\sqrt{R^2 + G^2 + B^2}} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \text{and} \quad \|\mathbf{I}\| = \sqrt{R^2 + G^2 + B^2}$$

For color images, this decomposition splits the image into :

- The scalar image $\|\mathbf{I}\| : \Omega \rightarrow \mathbb{R}$, representative of the *brightness* informations of the colors.
- The vector-valued image $\vec{i} : \Omega \rightarrow \mathbb{S}^2(\subset \mathbb{R}^3)$, representative of the *chromaticity* informations of the colors.

Then, a Gaussian noise is added in both fields $\vec{i}$ and $\|\mathbf{I}\|$. The corresponding noisy color image $\mathbf{I}_{\text{noisy}}$ is then constructed as :

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}_{\text{noisy}}(\mathbf{x}) = \vec{i}_{\text{noisy}}(\mathbf{x}) \|\mathbf{I}\|_{\text{noisy}}(\mathbf{x})$$

This leads to a *non-Gaussian noise* in the (R, G, B) space, with an effect of *chromaticity* noise as well as *brightness* noise.

The global resulting noise is finally not only Gaussian nor additive and should not favor any of the algorithm.

- **Algorithms parameters** : Classically, the choice of the parameters for a regularization algorithm is a very difficult issue. Indeed, the stopping evolution time (i.e the number of iterations) and the different thresholds, used in PDE expressions, play an important role for the quality of the resulting images. As the different regularization expressions have different PDE velocity scales, the use of a common data attachment term would not be suitable here.

Our experimental protocol has then been chosen as follows : we ran each algorithm with several set of parameters and saved each PDE iteration until 1000 iterations (which always corresponded to *over-smoothed* images). The results presented in Fig.2.13 and 2.14 are simply the *best looking images* (from our human perception viewpoint) in the corresponding iterated image list. Our desired aim is that the resulting artefacts in the result images are representative of the different methods. The automatic decision of a stopping iteration is indeed a very hard problem. It should imply the knowledge of some human-based criterion of image coherence.



(a) Noisy color image



(b) Color TV regularization (2.23)

(c) Vector ϕ -function (2.25) with $\phi(s) = \sqrt{1 + s^2}$ 

(d) Coherence enhancing diffusion (2.27)



(e) Beltrami Flow (2.30)

(f) Vector $I_{\xi\xi}$ (2.31)

Figure 2.14: Comparison of classical vector-valued regularization PDE's (II).

Analysis of the results : The two comparative figures Fig.2.13 and Fig.2.14 illustrate the experimental behaviors of the studied vector-valued regularization equations. They can be interpreted as follows :

- **Color total variation [31] :**

Due to the almost uncorrelated diffusion, some noisy vector edges are not well recovered since colors are blended (different diffusion in each color channel). This is particularly noticeable in the synthetic image Fig.2.13b for the edges of the yellow circle and the bottom part of the grey object.

The non-isotropic diffusion on flat regions should have created texture effects therein. Fortunately, isophote directions ξ_i are very different in these regions and the decoupled anisotropic diffusion that blends colors, creates an isotropic-like effect, allowing to retrieve the homogeneous regions quite well.

The staircasing effect, inherent to total variation methods is also noticeable in almost flat regions in the real photograph Fig.2.14b (look precisely at the shoulders and the back of the woman).

- **Vector ϕ -function [25, 140, 154, 155, 161, 168] :**

The ϕ -function used here is $\phi(s) = \sqrt{1 + s^2}$ corresponding to the *hyper-surface* formulation of the scalar case [45]. The retrieved result is very good for the synthetic image (Fig.2.13c). Actually, this vector ϕ -functional acts almost like its scalar counterpart, and performs an isotropic smoothing in flat regions, while anisotropically denoising the image contours. Nevertheless, this formulation well adapted to retrieve piecewise-constant regions (which is the configuration of the original synthetic image) gives sometimes a synthetic aspect to restored real photographs (look at the constant intensity patches appearing in the background of Fig.2.14c).

- **Coherence enhancing diffusion [184, 186, 188] :**

This equation was initially designed to take into account coherent image structures, in order to drive the diffusion. It is particularly done thanks to the convolution of the structure tensor $\mathbf{G}$ by a Gaussian kernel G_σ .

When dealing with noisy images, the variance σ of G_σ should be low, since trying to find coherent structures in noisy flat regions would create *fiber effects* (as illustrated in Fig.2.13d). (remind that the author proposed the equation (2.27) in [184, 186, 188] to enhance the “impressionism” style of color images).

On the other hand, using a low variance σ for the Gaussian kernel G_σ disables the detection of large structures. We used $\sigma = 0.8$ to restore the photograph in Fig.2.14. In this case, the PDE (2.27) behaves almost as a ϕ -functional one (Fig.2.14), even if some texture appears in the image background (since σ is not zero at all).

- **Beltrami flow** [104, 164] :

Experimentally, this PDE is one of the formulation that better conserves the edges during the regularization flow. This is mainly due to the factor $1/\sqrt{\det(\mathbf{Id} + \mathbf{G})}$ in the PDE (2.30), that quickly decreases the amount of diffusion near image contours.

This is perceptible for images with high noise : some vector contours may preserve a sharp and discontinuous aspect (Fig.2.13e). When used on a less noisy image Fig.2.14e, the obtained result is well restored. Note how the background is isotropically smoothed in Fig.2.14e. However, the smoothing has sometimes a tendency to stop on high variations (look for instance the interior of the woman's hair).

- **Vector $I_{\xi\xi}$** [157] :

This unidirectional smoothing process suffers from high noise, since texture effects quickly appear in noisy flat regions (Fig.2.13f) : The diffusion is always driven by a single direction and isotropic smoothing is unfortunately never performed. As for the edges, they are well preserved while being denoised, thanks to this explicit diffusion along the direction θ_- , which is indeed the right orientation of the vector edges.

This comparative figure underlines the fact that equations proposed for vector-valued image regularization, based on a vector-valued geometry, should naturally respect the following regularization properties (they are actually the same as the ones defined for the ϕ -function framework in section 2.1.3) :

- *Isotropic smoothing in homogeneous regions*, allowing to remove the noise efficiently therein, and avoiding the apparition of undesired image structures.
- Diffusion of the intensities *along the vector edges θ_- on regions of high vector variations*, in order to preserve the vector edges while removing the noise.
- *Decreased diffusion on sharp corners* (very high variation regions), in order to prevent the corner erosion.

These properties will be further considered as basic principles, when we will design our new vector-valued diffusion PDE, in chapter 3.

2.3 Regularization PDE's and constraints

More recently, the problem of regularizing multivalued *constrained* datasets has opened new theoretical and practical problems. Indeed, a PDE flow acting on multivalued constrained data is not a straightforward generalization of its unconstrained counterpart : we have to consider the fact that the vector data lie on *a constrained and generally non-flat manifold*. This yields significant modifications of the corresponding regularization equations, by adding *coupling terms* that allow the constraints to be preserved during the PDE flow. In the following, we review some of the recent related works, particularly the ones dealing with the *unitary norm constraint* [25, 39, 107, 142, 167, 182].

2.3.1 Direction diffusion

Let us consider an image of *unit vectors* :

$$\mathbf{I} : \Omega \rightarrow \mathbb{R}^n \quad \text{such as} \quad \forall \mathbf{x} \in \Omega, \quad \|\mathbf{I}(\mathbf{x})\| = 1 \quad (2.32)$$

This unitary norm constraint (2.32) means that each point $\mathbf{I}(\mathbf{x})$ of the image belongs to the specific manifold $S^{n-1} \subset \mathbb{R}^n$, also named *unit sphere*.

For instance, the following figure Fig.2.15 illustrates two interesting examples of such fields.

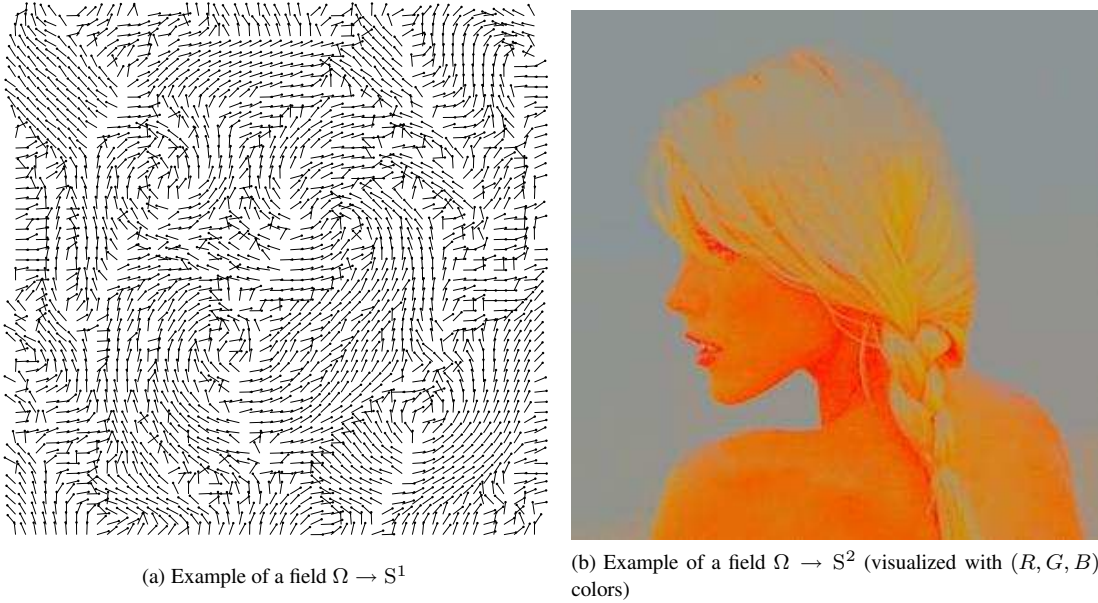


Figure 2.15: Unit vector fields representing directions and chromaticity part of color images.

As described in section 2.2.8, the unit vectors $\vec{i}$ corresponding to the directions of color vectors (R, G, B) ,

$$\vec{i} = \frac{1}{\sqrt{R^2 + G^2 + B^2}} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

are typical features of the *color chromaticity* of the color pixels, without the brightness information (Fig.2.15b). It is a field of 3D vectors and has been represented using the (R, G, B) color space in Fig.2.15b. Note how this image only represent the basic colors of the initial image (Fig.2.14).

It is interesting to anisotropically smooth such unit vector images $\mathbf{I}$ while *preserving the unitary norm constraint*. By doing this we are able to regularize *directions fields*, as well as *chromaticity noise* in color images.

A first approach consists in using classical *unconstrained* regularization PDE's on $\mathbf{I}$, then *re-project* each vector $\mathbf{I}(\mathbf{x})$ into the unit sphere S^{n-1} at the end of the regularization process, or after each PDE iteration. This method is computationally expensive, as well as imprecise : nothing prevents some direction vectors to *vanish numerically* during the PDE evolution (especially if one re-projects after several PDE iterations). Thus, the re-projection on S^{n-1} will necessarily loose the direction information of the considered (null) vectors. To avoid this re-projection trick, we have to *formally follow the unit sphere manifold* S^{n-1} during the regularizing flow.

For this purpose, several approaches have been already proposed in the literature. Actually, the main starting proposal has been suggested by Perona [142]. He restricted to an image of 2D directions ($n = 2$) and proposed to compute the field of the polar angles θ corresponding to the direction field $\mathbf{I}$. Then, this scalar image can be regularized by a classical heat flow :

$$\begin{cases} \theta_{(t=0)} = \theta_{\text{noisy}} \\ \frac{\partial \theta}{\partial t} = \Delta \theta \end{cases} \quad (2.33)$$

The polar characteristic of θ implies that we have to take care of the 2Π -periodicity of the angles during the regularization flow. Thus, Perona proposed then a special numerical scheme, that handles this problem :

$$\frac{\partial \theta_k}{\partial t} = \lambda (\sin(\theta_{k+1} - \theta_k) + \sin(\theta_{k-1} - \theta_k))$$

Despite the good results obtained by this scheme, some points are limiting the method :

- The Laplacian operator forces the diffusion process to be isotropic, blurring then the discontinuities in the direction field θ .
- Polar angles are well adapted for 2D direction vectors, but the number of required angles increases when considering higher dimension vectors ($n > 2$). In this case, the coupling between these orientation angles should be taken into account, in a way that is not defined by the Perona formulation.

In order to outperform these limitations, anisotropic methods acting directly on the direction field $\mathbf{I}$ have been proposed for instance in [39, 43, 106, 168, 182]. The idea is to find regularization PDE flows that naturally preserve the norm constraint (2.32).

Different formal solutions have been proposed in the literature, leading to very similar equations :

- Tang-Sapiro-Caselles in [168] proposed to minimize the following functional, coming from the theory of harmonic maps in liquid crystals :

$$\min_{\mathbf{I} : \Omega \rightarrow S^{n-1}} E(\mathbf{I}) = \int_{\Omega} \|\nabla \mathbf{I}\|^r d\Omega \quad \text{where } r > 0 \quad (2.34)$$

The norm $\|\nabla \mathbf{I}\| = \mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$ is the one defined in (2.21) (section 2.2.2).

Note that this minimization is constrained since we look for $\forall \mathbf{x} \in \Omega, \mathbf{I}(\mathbf{x}) \in S^{n-1}$.

According to the authors, the corresponding Euler-Lagrange equations of (2.34) end up with

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} = \text{div} (\|\nabla \mathbf{I}\|^{r-2} \nabla I_i) + I_i \|\nabla \mathbf{I}\|^r \end{cases} \quad (i = 1..n) \quad (2.35)$$

- Chan-Kang-Shen in [43] started the idea of denoising color bases whose certain components describe orientations (this is the case for instance with the *Hue* in the (H, S, V) color space). They end up with a particular case of (2.35) with $p = 1$ (total variation formulation).
- Other variants of the norm constrained equation (2.35) have been also proposed, for instance in [27, 106, 182], fitting the Beltrami Framework (defined in section 2.2.6) or improving numerical schemes.

An example of direction field regularization is illustrated in Fig.2.16, using the flow (2.35) (with $r = 1$). Note how the additional term $I_i \|\nabla I_i\|^r$ in the PDE allows to preserve the unit norm constraint (Fig.2.16c), while the corresponding unconstrained regularization decreases the length of the vectors (especially along the contour of the circle structure, in Fig.2.16b).

We will go back on the direction diffusion case in chapter 5, with a generalization of orientation diffusion techniques for specific fields of orientation data, and the proposal of adapted numerical schemes to implement these kinds of algorithms. Experimental results and real applications of direction field regularization can be also found in chapter 6.

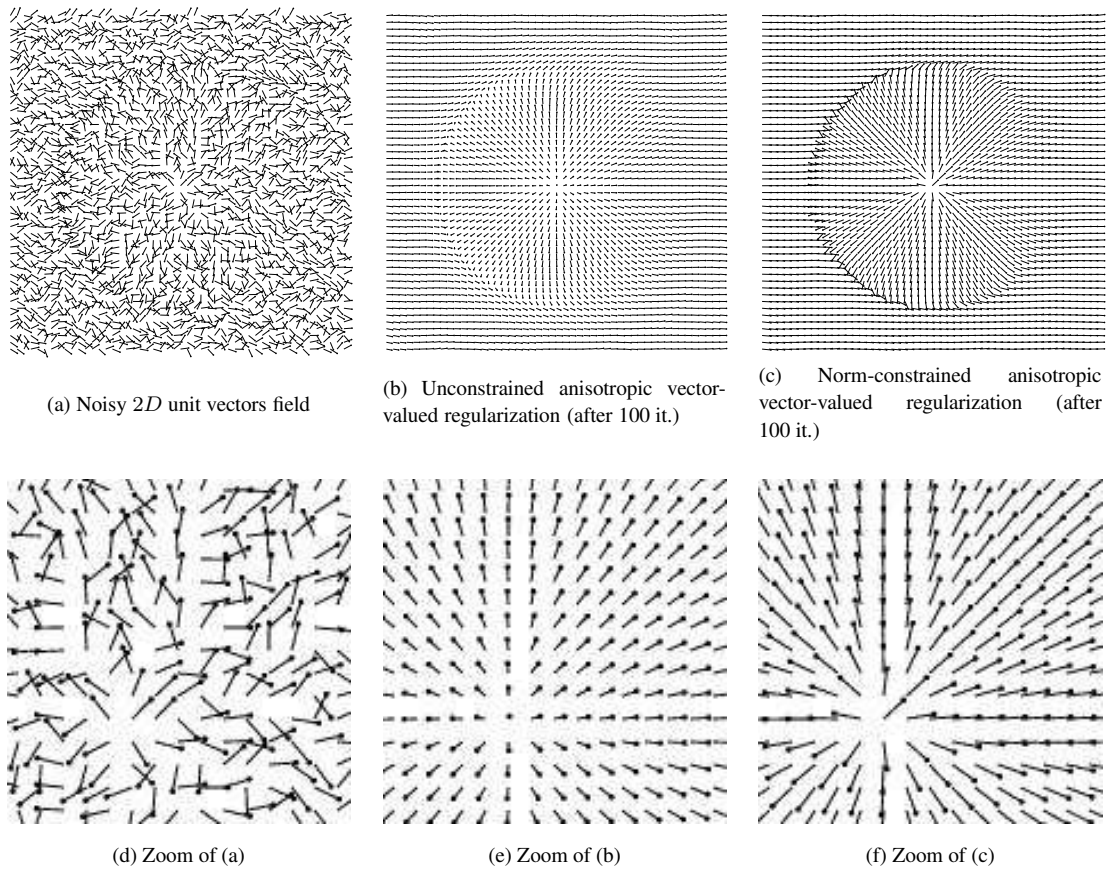


Figure 2.16: Regularization of unit vector fields.

2.3.2 Other constrained problems

We cannot conclude this survey on constrained regularization methods without referencing these following PDE based approaches, preserving other constraint types :

- In [27], the images are constrained to lie on specific manifolds, defined by *implicit surfaces via level-set functions*. The constraints here do not concern the image features themselves, but the definition domain Ω of the images. This new and innovative approach allows to regularize features measured on surfaces (this is the case for instance when dealing with MEG-EEG datasets, as described in [55, 72]).

Fig.2.17 illustrates the restoration of a noisy image defined on a torus. ¹

More particularly, this method is useful when no simple continuous mappings between $\mathbb{R}^2$ and the implicit surface are known.

- in [140], the authors proposed to deal with probability vectors, i.e where the sum of the vector components is constrained to be 1. Actually, this constraint is *linear* and is intrinsic.

¹This figure has been kindly provided by L. Lopez Perez / ODYSSEE, INRIA-Sophia Antipolis.

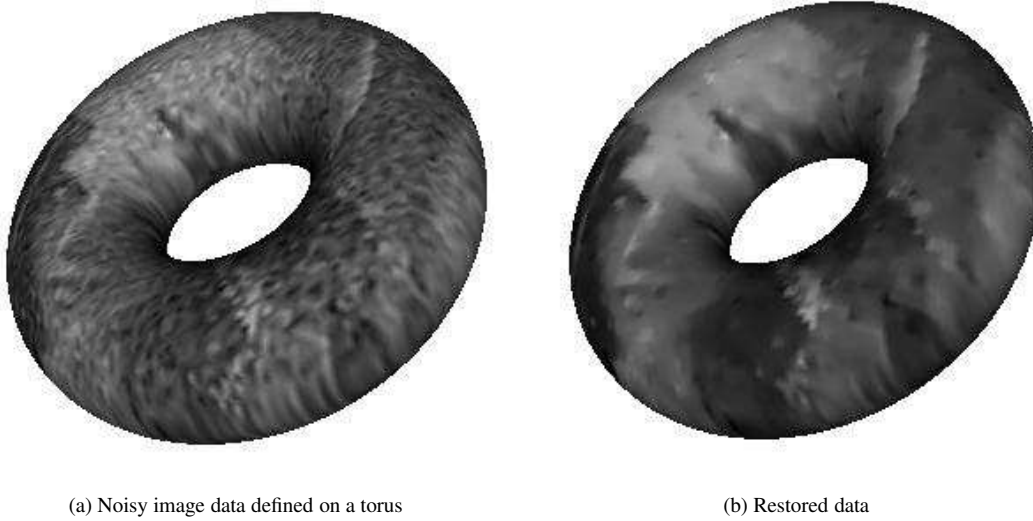


Figure 2.17: Regularization of data defined on non-flat surfaces.

sically preserved by the proposed unconstrained anisotropic diffusion PDE. The equations are then applied to improve image segmentation techniques based on Bayesian approaches.

- in [51, 52, 53, 54], the author theoretically studies several PDE-flows acting on matrix fields and preserving some particular spectral constraints. Our constrained regularization framework and its development, proposed in chapter 5 will cross some of the ideas raised in these papers. Moreover, we will illustrate possible concrete applications of these constrained equations (chapter 6).

2.4 Open questions on PDE-based regularization methods

In this section, we close our general survey on regularization PDE's and raise the important questions investigated in the next chapters of this document.

2.4.1 On the unconstrained regularization

As we noticed, three different levels of design have been generally considered so far in the literature, for the problem of image regularization :

1. Variational formulations :

The regularization is seen as *the minimization of a functional* $E(\mathbf{I})$ measuring global image variations. These variations are expressed as the integral of functions depending on a *local variation norm* $\mathcal{N}(\mathbf{I})$, which can be for instance :

- Scalar values : $\mathcal{N} = \|\nabla I\|$.
- Vector values : $\mathcal{N} = \sqrt{\lambda_+}$, or $\mathcal{N} = \sqrt{\lambda_+ + \lambda_-}$, or $\mathcal{N} = \sqrt{\lambda_+ - \lambda_-}$, as defined in section 2.2.2.

The regularizing functional has then the following form for multivalued images :

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

The function ϕ parameterizes the regularization behavior and the corresponding PDE evolution is found thanks to the *Euler-Lagrange* equations of $E(\mathbf{I})$.

Question : How to choose a well adapted ϕ -function to deal with specific applications ? Understanding the exact local regularization behavior performed by the gradient descent of a ϕ -functional by looking only at the ϕ -function form is not trivial, especially for the case of vector-valued regularization.

2. Divergence formulations :

The regularization is seen as *the chemical diffusion of image pixels* following a gradient flux, defined by a *diffusion tensor* $\mathbf{D}$.

$$\frac{\partial I}{\partial t} = \text{div}(\mathbf{D}\nabla I) \quad (2.36)$$

Here, the 2×2 matrix $\mathbf{D}$ parameterizes the regularization behavior of the PDE (2.36), and is often constructed by choosing its eigenvalues λ_1, λ_2 , and its corresponding eigenvectors $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{D} = \lambda_1 \mathbf{u}\mathbf{u}^T + \lambda_2 \mathbf{v}\mathbf{v}^T$$

It is commonly assumed that the shape of the diffusion tensor $\mathbf{D}$ gives the *local smoothing geometry of the regularization process*. The eigenvalues λ_1, λ_2 corresponding to the amount

of diffusion in the corresponding eigenvectors directions $\mathbf{u}$ and $\mathbf{v}$. Thus, they are naturally designed as functions of the *local geometry* of the considered images (scalar or vector-valued).

Question : Some fields of pointwise *isotropic* diffusion tensors $\mathbf{D}$ may lead to *anisotropic* diffusion. The usual example of the scalar ϕ -functional framework, described in section 2.1.3 illustrates this point :

$$\min_{I : \Omega \rightarrow \mathbb{R}} \int_{\Omega} \phi(\|\nabla I\|) d\Omega \quad \Longrightarrow \quad \frac{\partial I}{\partial t} = \operatorname{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \quad (2.37)$$

This scalar regularization PDE (2.37) corresponds to the divergence-based equation (2.36), by considering the field of the following tensors $\mathbf{D}$:

$$\mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \mathbf{Id} \quad (2.38)$$

Each tensor of the field $\mathbf{D}$ is *isotropic* and can be represented by a cercle of radius $\phi'(\|\nabla I\|)/\|\nabla I\|$. However, the ϕ -functionals *do not behave as isotropic filters* (compare the different results obtained in sections 2.1.1 and 2.1.3 for instance). The reason is simply that :

$$\operatorname{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \neq \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \Delta I$$

It means that the shape of the diffusion tensor $\mathbf{D}$ in the equation (2.36) *cannot be directly interpreted* to be the local diffusion geometry of the regularization process.

This can be better understood if we also remark that two different matrices $\mathbf{D}$ lead exactly to the same ϕ -functional based PDE (2.37), for the case of scalar images :

$$\mathbf{D}_1 = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \mathbf{Id} \quad \text{and} \quad \mathbf{D}_2 = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|^3} \nabla I \nabla I^T$$

Indeed, since ∇I is an eigenvector of $\nabla I \nabla I^T$:

$$(\nabla I \nabla I^T) \nabla I = \|\nabla I\|^2 \nabla I$$

then

$$\mathbf{D}_1 \nabla I = \mathbf{D}_2 \nabla I = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I$$

Thus, the following flows are equivalent :

$$\begin{aligned} \frac{\partial I}{\partial t} &= \operatorname{div} (\mathbf{D}_1 \nabla I) \\ &= \operatorname{div} (\mathbf{D}_2 \nabla I) \\ &= \operatorname{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \end{aligned}$$

In this case, $\mathbf{D}_1$ is isotropic and $\mathbf{D}_2$ is anisotropic but both perform the same anisotropic diffusion. Then, we cannot be sure of the exact diffusion geometry of the flow (2.36) by looking only at the diffusion tensor shape.

3. Second-derivatives formulations :

Here, the regularization process is seen as two *1D oriented heat flows* $I_{\mathbf{uu}}$ and $I_{\mathbf{vv}}$ in orthogonal directions $\mathbf{u}$ and $\mathbf{v}$, with respective weights c_1 and c_2 :

$$\frac{\partial I}{\partial t} = c_1 I_{\mathbf{uu}} + c_2 I_{\mathbf{vv}} \quad \text{with} \quad \mathbf{u} \perp \mathbf{v}$$

As we noticed in the end of the section 2.1.5, this kind of equation can be written with a diffusion tensor $\mathbf{T}$:

$$\frac{\partial I}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}) \quad \text{where} \quad \mathbf{T} = c_1 \mathbf{uu}^T + c_2 \mathbf{vv}^T \quad (2.39)$$

and $\mathbf{H}$ is the Hessian of the image I (see section 1.2.2).

Indeed, we have

$$\begin{aligned} \text{trace}(\mathbf{T}\mathbf{H}) &= \text{trace}((c_1 \mathbf{uu}^T + c_2 \mathbf{vv}^T)\mathbf{H}) \\ &= c_1 \text{trace}(\mathbf{uu}^T \mathbf{H}) + c_2 \text{trace}(\mathbf{vv}^T \mathbf{H}) \\ &= c_1 (\mathbf{u}^T \mathbf{H} \mathbf{u}) + c_2 (\mathbf{v}^T \mathbf{H} \mathbf{v}) \\ &= c_1 I_{\mathbf{uu}} + c_2 I_{\mathbf{vv}} \end{aligned}$$

□

The matrix $\mathbf{T}$ is a diffusion tensor whose eigenvalues are c_1, c_2 and corresponding (orthogonal) eigenvectors are $\mathbf{u}$ and $\mathbf{v}$. Thus, it is well adapted to describe the geometry of the regularization processes based on oriented *1D* Laplacian.

Question : Are both tensors $\mathbf{D}$ and $\mathbf{T}$ equivalent in the formulations

$$\frac{\partial I}{\partial t} = \text{div}(\mathbf{D}\nabla I) \quad \text{and} \quad \frac{\partial I}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}) \quad ?$$

The answer is negative. Indeed, if we consider again the ϕ -functional regularization framework for the *scalar case*, we get the following equivalent expressions for the gradient descent (see section 2.1.3) :

$$\begin{aligned} \frac{\partial I}{\partial t} &= \text{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \\ &= \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} I_{\xi\xi} + \phi''(\|\nabla I\|) I_{\eta\eta} \\ &= \text{trace} \left(\left[\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \xi\xi^T + \phi''(\|\nabla I\|) \eta\eta^T \right] \mathbf{H} \right) \end{aligned}$$

In this case, the two diffusion tensors $\mathbf{D}$ and $\mathbf{T}$ corresponding to the same ϕ -functional regularization equation are :

$$\text{div}(\mathbf{D}\nabla I) = \text{trace}(\mathbf{T}\mathbf{H}) \quad \text{with} \quad \begin{cases} \mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \mathbf{Id} \\ \mathbf{T} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \xi\xi^T + \phi''(\|\nabla I\|) \eta\eta^T \end{cases}$$

This is illustrated on Fig.2.18, where we represented the behaviors of two scalar regularization processes :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i) \quad \text{and} \quad \frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D}\nabla I_i) \quad (i = 1..n)$$

where

$$\mathbf{D} = \frac{1}{\|\nabla I\|} \mathbf{Id}$$

One uses the trace formulation (Fig.2.18b), while the other is written with a divergence operator (Fig.2.18c). They are based on the same field $\mathbf{D}$ of *isotropic* diffusion tensors. Note how the result based on the divergence equation seems to be anisotropically restored, despite the isotropic form of the diffusion tensor $\mathbf{D}$ (Fig.2.18c). It is not the case for the trace-based equation (Fig.2.18b).

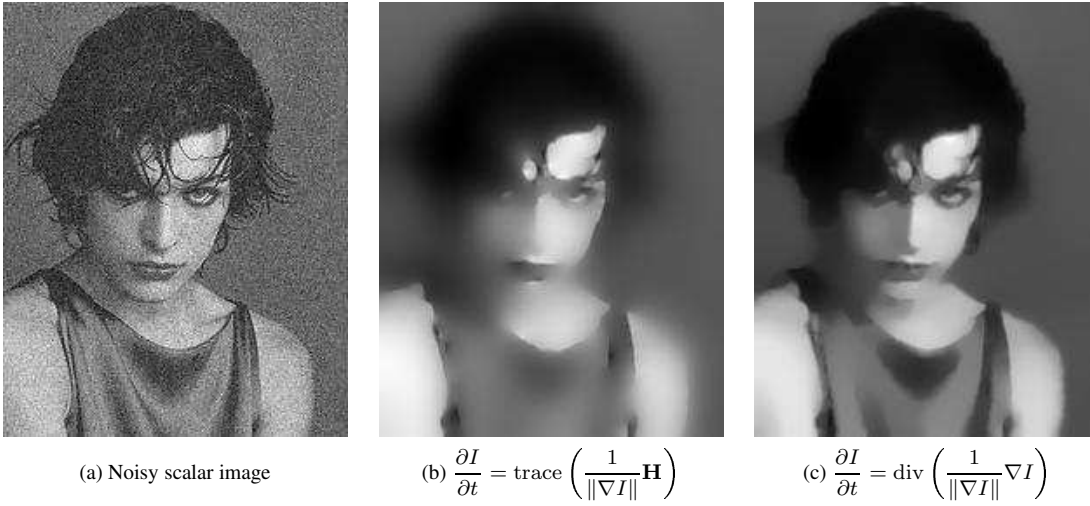


Figure 2.18: Differences between divergence-based and trace-based regularization.

The link between $\mathbf{T}$ and $\mathbf{D}$ is known for scalar schemes, but not in the general case of multivalued image regularization. What is the correspondence between these two tensors $\mathbf{D}$ and $\mathbf{T}$?

In chapter 3 we contribute to answer to these questions about unconstrained PDE regularization behaviors. We will particularly want to underline the fact that the local geometric interpretation is better given by the tensor shape of the trace-based PDE (2.39), rather than the divergence-based one (2.36). This will be particularly applied in chapter 4, where we will design regularization PDE's adapted to specific problems, related to color image restoration.

2.4.2 On the constrained regularization

Concerning the problem of regularizing constrained features, we have focused our study on *orientation features*, as the unitary norm constraint allows to deal with. Two complementary questions follow naturally our review of direction diffusion methods :

1. How to deal, within a common framework, with the regularization of orientation data that are *more complex* than unit vectors ? One may particularly think of *rotation matrices*, or *diffusion tensors*, that symbolize multi-dimensional orientation features.
2. How can we find *accurate numerical schemes*, avoiding the necessary re-projection step on the corresponding constrained manifolds ? The recent report [182] is the only one that proposed an alternative solution to this reprojection problem for the unitary norm constraint.

These two questions will be tackled in chapter 5, leading to the formalization of a unifying framework used to regularize fields of unit vectors or rotation matrices, via the *orthonormal vector set constraint*. Moreover, the physical interpretation of the resulting equations will help to construct a natural and fast numerical scheme to implement these PDE evolutions, and applications will be finally illustrated in chapter 6.

Bon, ta thèse ne s'écrit pas très vite mais c'est pas plus mal, ça va te mettre la pression pour le mois de septembre.

R. Deriche.

Chapter 3

From Variational to Local Formulations



In this part, we propose a new approach to unify with a common diffusion equation, the range of vector-valued regularization PDE's described earlier. This is done by developing each of the three formulations (variational methods, divergence forms and oriented Laplacians) into a generic form, based on diffusion tensors and the trace operator. This formalism is an important step to better understand the local behaviors of regularization PDE's, thanks to a partial geometric interpretation of the developed equation. It helps as well to design a new and efficient vector-valued PDE, respecting natural regularization properties. Finally, it induces interesting numerical schemes for the implementation of these various regularization algorithms.

3.1 From functionals to divergence forms

In this section, we consider the regularization of 2D multivalued images $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$, as a *variational problem*. We propose to minimize a generic regularization functional depending of the Di Zenzo's variation indicators λ_+ and λ_- (eigenvalues of the structure tensor $\mathbf{G}$), measuring the vector-valued image variations. We show that the corresponding Euler-Lagrange derivation leads to a specific case of a divergence-based equation $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i)$ where the diffusion tensor field $\mathbf{D}$ is simply expressed from the spectral elements of $\mathbf{G}$.

3.1.1 A generic functional for vector-valued regularization

Image regularization based on variational methods consists in the minimization of energy functionals that measure *image variations*. The idea is that minimizing this kind of functional will flatten the variations, then remove the noise.

In the case of scalar images $I : \Omega \rightarrow \mathbb{R}$, this is usually done through the use of ϕ -functionals, depending on the *gradient norm* $\|\nabla I\|$:

$$\min_{I: \Omega \rightarrow \mathbb{R}} E(I) = \int_{\Omega} \phi(\|\nabla I\|) d\Omega \quad (3.1)$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is an increasing function penalizing high variations for the minimization process.

In the case of vector-valued images $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$, the local vector variations are defined by the *two eigenvalues* λ_+ and λ_- of the *structure tensor* $\mathbf{G}$:

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{pmatrix} \quad \text{with} \quad \begin{cases} g_{11} = \sum_{k=1}^n I_{k_x}^2 \\ g_{12} = \sum_{k=1}^n I_{k_x} I_{k_y} \\ g_{22} = \sum_{k=1}^n I_{k_y}^2 \end{cases} \quad (3.2)$$

A way of adapting variational methods to multivalued-images is then to define a vector-valued norm $\mathcal{N}(\lambda_+, \lambda_-)$ that has the same role as the scalar gradient norm $\|\nabla I\|$. Then one minimizes a regularizing ϕ -functional depending on this norm :

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \phi(\mathcal{N}) \, d\Omega \quad (3.3)$$

In the literature, the vector variation indicator $\mathcal{N}$ is usually chosen to be one of these three expressions :

$$\mathcal{N}_1 = \sqrt{\lambda_+} \quad \text{or} \quad \mathcal{N}_2 = \sqrt{\lambda_+ - \lambda_-} \quad \text{or} \quad \mathcal{N}_3 = \sqrt{\lambda_+ + \lambda_-}$$

Actually, the a-priori choice for a variation norm limits the complexity of the regularizing functional, for the multivalued case. Since the vector variations are given by *two* parameters λ_+ and λ_- , we should rather consider the more general problem of minimizing this ψ -functional :

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \psi(\lambda_+, \lambda_-) \, d\Omega \quad (3.4)$$

where $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a function that should penalizes high variations. It is a natural extension of the scalar ϕ -functions for vector-valued images. The approach (3.3) is obviously a particular case of the minimization (3.4), with the following choices for ψ :

$$\begin{aligned} \psi_1(\lambda_+, \lambda_-) &= \phi(\sqrt{\lambda_+}) \\ \text{or } \psi_2(\lambda_+, \lambda_-) &= \phi(\sqrt{\lambda_+ - \lambda_-}) \\ \text{or } \psi_3(\lambda_+, \lambda_-) &= \phi(\sqrt{\lambda_+ + \lambda_-}) \end{aligned}$$

Note also that for the particular case of scalar image regularization ($n = 1$), the ψ -functional (3.4) naturally reduces to a classical ϕ -functional formulation (3.1), since then $\lambda_- = 0$.

Usually, the regularizing minimizations are performed with a *gradient descent*, coming from the Euler-Lagrange expression of the considered functionals. Let us compute the one related to the functional (3.4)

3.1.2 Computing the gradient descent

The Euler-Lagrange equations corresponding to the functional (3.4) are :

$$\frac{\partial I_i}{\partial t} = \operatorname{div} \begin{pmatrix} \frac{\partial \psi}{\partial I_{i_x}} \\ \frac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} \quad (i = 1..n) \quad (3.5)$$

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 :

$$\begin{pmatrix} \frac{\partial \psi}{\partial I_{i_x}} \\ \frac{\partial \psi}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} \frac{\partial \lambda_+}{\partial I_{i_x}} & \frac{\partial \lambda_-}{\partial I_{i_x}} \\ \frac{\partial \lambda_+}{\partial I_{i_y}} & \frac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} \begin{pmatrix} \frac{\partial \psi}{\partial \lambda_+} \\ \frac{\partial \psi}{\partial \lambda_-} \end{pmatrix} \quad (3.6)$$

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} :

$$\frac{\partial \lambda_{\pm}}{\partial I_{i_x}} = \sum_{k,l} \frac{\partial \lambda_{\pm}}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_x}} \quad \text{and} \quad \frac{\partial \lambda_{\pm}}{\partial I_{i_y}} = \sum_{k,l} \frac{\partial \lambda_{\pm}}{\partial g_{kl}} \frac{\partial g_{kl}}{\partial I_{i_y}} \quad (3.7)$$

Considering formula (3.2), the expressions $\frac{\partial g_{kl}}{\partial I_{i_x}}$ and $\frac{\partial g_{kl}}{\partial I_{i_y}}$ are particularly simple :

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

i.e (3.7) can be written as :

$$\begin{pmatrix} \frac{\partial \lambda_{\pm}}{\partial I_{i_x}} \\ \frac{\partial \lambda_{\pm}}{\partial I_{i_y}} \end{pmatrix} = \begin{pmatrix} 2\frac{\partial \lambda_{\pm}}{\partial g_{11}} & \frac{\partial \lambda_{\pm}}{\partial g_{12}} \\ \frac{\partial \lambda_{\pm}}{\partial g_{12}} & 2\frac{\partial \lambda_{\pm}}{\partial g_{22}} \end{pmatrix} \nabla I_i \quad (3.8)$$

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 :

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial g_{kl}} &= \frac{\partial \lambda_+}{\partial g_{kl}} \theta_+ \theta_+^T + \frac{\partial \lambda_-}{\partial g_{kl}} \theta_- \theta_-^T \\ &+ \lambda_+ \frac{\partial \theta_+}{\partial g_{kl}} \theta_+^T + \lambda_- \frac{\partial \theta_-}{\partial g_{kl}} \theta_-^T \\ &+ \lambda_+ \theta_+ \frac{\partial \theta_+^T}{\partial g_{kl}} + \lambda_- \theta_- \frac{\partial \theta_-^T}{\partial g_{kl}} \end{aligned} \quad (3.9)$$

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

$$\begin{cases} \theta_+^T \theta_+ = \theta_-^T \theta_- = 1 \\ \theta_+^T \theta_- = \theta_-^T \theta_+ = 0 \end{cases} \quad \text{and} \quad \begin{cases} \frac{\partial \theta_+^T}{\partial g_{kl}} \theta_+ = \theta_+^T \frac{\partial \theta_+}{\partial g_{kl}} = 0 \\ \frac{\partial \theta_-^T}{\partial g_{kl}} \theta_- = \theta_-^T \frac{\partial \theta_-}{\partial g_{kl}} = 0 \end{cases} \quad (3.10)$$

We first multiply the equation (3.9) by $\theta_{\pm}^T$ at the left, by $\theta_{\pm}$ at the right, then use the properties (3.10). It allows high simplifications, and leads to these two relations :

$$\frac{\partial \lambda_+}{\partial g_{kl}} = \theta_+^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_+ \quad \text{and} \quad \frac{\partial \lambda_-}{\partial g_{kl}} = \theta_-^T \frac{\partial \mathbf{G}}{\partial g_{kl}} \theta_- \quad (3.11)$$

Equations (3.11) 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 [132] 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 :

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

With all these elements, we can express (3.8) as :

$$\begin{pmatrix} \frac{\partial \lambda_+}{\partial I_{i_x}} \\ \frac{\partial \lambda_+}{\partial I_{i_y}} \end{pmatrix} = 2 \theta_+ \theta_+^T \nabla I_i \quad \text{and} \quad \begin{pmatrix} \frac{\partial \lambda_-}{\partial I_{i_x}} \\ \frac{\partial \lambda_-}{\partial I_{i_y}} \end{pmatrix} = 2 \theta_- \theta_-^T \nabla I_i \quad (3.12)$$

Finally, replacing (3.12) in the Euler-Lagrange equations (3.6) and (3.5), gives the vector-valued gradient descent of the functional (3.4) :

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} \int_{\Omega} \psi(\lambda_+, \lambda_-) d\Omega \quad \Longrightarrow \quad \frac{\partial I_i}{\partial t} = 2 \operatorname{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \psi}{\partial \lambda_-} \theta_- \theta_-^T \right] \nabla I_i \right) \quad (3.13)$$

(for $i = 1..n$)

□

Note that (3.13) is a divergence-based equation such that :

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

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

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

associated to these corresponding orthonormal eigenvectors :

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

They are *directly linked* with the spectral elements of the structure tensor $\mathbf{G}$, and the partial derivatives of the function ψ . We have to mention that λ_1 or λ_2 may be *negative*, if the function ψ is not chosen to be strictly increasing, leading to *negative-definite* diffusion tensors.

In this chapter, we will often use the concept of diffusion tensor in order to designate real symmetric matrices, without considering the semi-positive constraint, i.e tensors that may represent inverse diffusion.

With the demonstration (3.13), we extend the very recent work in [192], where the authors reached a particular case of diffusion tensors $\mathbf{D}$, using a variational formulation :

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} \int_{\Omega} \operatorname{trace} (\phi(\lambda_+) \theta_+ \theta_+^T + \phi(\lambda_-) \theta_- \theta_-^T) d\Omega \quad \Longrightarrow \quad \frac{\partial I_i}{\partial t} = \operatorname{div} (\mathbf{D} \nabla I_i)$$

where

$$\mathbf{D} = \phi'(\lambda_+) \theta_+ \theta_+^T + \phi'(\lambda_-) \theta_- \theta_-^T$$

Here, the *same function* $\phi' : \mathbb{R} \rightarrow \mathbb{R}$ appeared for each eigenvalue of the obtained tensor $\mathbf{D}$.

With our equation (3.13) we clearly separate the two eigenvalues of the diffusion tensor $\mathbf{D}$, as two different functions of λ_+ and λ_- , expressed with the partial derivatives of the ψ -function of the minimizing functional (3.4). Going back from a generic divergence expression $\frac{\partial I_i}{\partial t} = \operatorname{div} (\mathbf{T} \nabla I_i)$, to the corresponding variational formulation may be done by finding a *potential* $\psi(\lambda_+, \lambda_-)$ whose gradient $(\frac{\partial \psi}{\partial \lambda_+}, \frac{\partial \psi}{\partial \lambda_-})^T$ corresponds to the two eigenvalues of the tensor $\mathbf{T}$ inside the divergence operator. The related minimizing functional would then be (3.4)

Note that like the scalar ϕ -functional case, the ψ -function in (3.4) defines the regularization behavior of the resulting PDE.

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 :

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

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.

3.1.3 Link with previous multivalued regularization approaches

We illustrate there how previously proposed vector-valued regularization PDE's may be interpreted within our generic divergence-based equation (3.13).

1. Vector ϕ -functionals [25, 140, 154, 155, 161, 168] :

The vector ϕ -functional formalism is

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} \int_{\Omega} \phi(\|\nabla \mathbf{I}\|) d\Omega \quad \Longrightarrow \quad \frac{\partial I_i}{\partial t} = \operatorname{div} \left(\frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|} \nabla I_i \right) \quad (3.14)$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ and $\|\nabla \mathbf{I}\|^2 = \mathcal{N}_+ = \sum_{i=1}^n \|\nabla I_i\|^2 = \sqrt{\lambda_+ + \lambda_-}$.

Using our expression (3.13), it can be viewed as the diffusion process :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i) \quad \text{with} \quad \mathbf{D} = \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{\sqrt{\lambda_+ + \lambda_-}} \theta_+ \theta_+^T + \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{\sqrt{\lambda_+ + \lambda_-}} \theta_- \theta_-^T$$

i.e $\mathbf{D}$ is the *isotropic* diffusion tensor, with equal eigenvalues $\frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|}$.

If we want to going back to a ψ -functional formulation (3.4), we have to find a function $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ that verifies :

$$2 \frac{\partial \psi}{\partial \lambda_+} = 2 \frac{\partial \psi}{\partial \lambda_-} = \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{\sqrt{\lambda_+ + \lambda_-}}$$

This is indeed verified by the expected function :

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

Then, we find again the minimizing functional (3.14).

Notice that using functionals depending only on the norm $\|\nabla \mathbf{I}\|$ will *necessarily* leads to *isotropic diffusion tensors* $\mathbf{D}$ in the resulting divergence based PDE's (3.13). It emphasizes again the fact that isotropic diffusion tensor may lead to anisotropic regularization. The tensor shape is not interpretable if one wants to predict the regularization behavior of the PDE flow (3.13).

2. Coherence enhancing diffusion [184, 186, 188] :

The coherence enhancing diffusion formalism is :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i) \quad \text{where} \quad \mathbf{D} = \alpha \theta_+ \theta_+^T + \left(\alpha + (1 - \alpha) e^{\left(-\frac{C}{(\lambda_+ - \lambda_-)^2} \right)} \right) \theta_- \theta_-^T \quad (3.15)$$

In the initial method, the spectral elements $\lambda_{\pm}$ and $\theta_{\pm}$ were computed from a *smoothed* version $\mathbf{G}_{\sigma} = \mathbf{G} * G_{\sigma}$ of the structure tensor field $\mathbf{G}$ (where G_{σ} is a normalized Gaussian kernel). Here, we limit our analogy for a *non-smoothed* version of $\mathbf{G}$ (i.e $\sigma \rightarrow 0$).

This approach has been proposed in a direct divergence-based form, without any knowledge of the corresponding minimizing functional. Actually, if the variational problem exists, one should look for a function $\psi(\lambda_+, \lambda_-)$ such that :

$$\frac{\partial \psi}{\partial \lambda_+} = \alpha \quad \text{and} \quad \frac{\partial \psi}{\partial \lambda_-} = \alpha + (1 - \alpha) \exp \left(-\frac{C}{(\lambda_+ - \lambda_-)^2} \right)$$

Let us consider the 2D vector field $\mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined by :

$$\mathcal{F}(\lambda_+, \lambda_-) = \begin{pmatrix} \alpha \\ \alpha + (1 - \alpha) \exp \left(-\frac{C}{(\lambda_+ - \lambda_-)^2} \right) \end{pmatrix}$$

It means that $\mathcal{F}$ should be the *gradient* of a two dimensional potential $\psi(\lambda_+, \lambda_-)$. This is possible iff its curl is null. The actual curl of the field $\mathcal{F}$ is computed as :

$$\begin{aligned} \text{curl}(\mathcal{F}) &= \frac{\partial \mathcal{F}_2}{\partial \lambda_+} - \frac{\partial \mathcal{F}_1}{\partial \lambda_-} \\ &= \frac{2C(1 - \alpha)}{(\lambda_+ - \lambda_-)^3} \exp \left(-\frac{C}{(\lambda_+ - \lambda_-)^2} \right) \\ &\neq 0 \end{aligned}$$

since $\alpha \neq 1$ and $C \neq 0$.

Unfortunately, the coherence enhancing diffusion cannot be expressed as the gradient descent of a ψ -minimizing functional as (3.4). It particularly shows that a direct and local designing of divergence-based PDE's offers more freedom to create different regularization behaviors that couldn't be reached with variational formulations.

3. Beltrami flow [104, 164] :

The Beltrami flow formalism for 2D vector image regularization is :

$$\min_{\mathbf{I} : \Omega \rightarrow \mathbb{R}^n} \int_{\Omega} \sqrt{(1 + \lambda_+)(1 + \lambda_-)} d\Omega \quad \implies \quad \frac{\partial I_i}{\partial t} = \frac{1}{\sqrt{(1 + \lambda_+)(1 + \lambda_-)}} \text{div}(\mathbf{D} \nabla I_i) \quad (3.16)$$

where the diffusion tensor $\mathbf{D}$ can be expressed as (see section 2.2.6) :

$$\mathbf{D} = \sqrt{\frac{1 + \lambda_-}{1 + \lambda_+}} \theta_+ \theta_+^T + \sqrt{\frac{1 + \lambda_+}{1 + \lambda_-}} \theta_- \theta_-^T$$

Note that the factor $1/\sqrt{(1 + \lambda_+)(1 + \lambda_-)}$ is only appearing because the gradient descent is computed with respect to the metric of the image surface, rather than the classical Euclidean metric.

If we are only starting from the divergence form and we are looking to find the corresponding variational ψ -functional formulation (3.4), we should look for a function $\psi(\lambda_+, \lambda_-)$ that verifies :

$$2\frac{\partial\psi}{\partial\lambda_+} = \sqrt{\frac{1+\lambda_-}{1+\lambda_+}} \quad \text{and} \quad 2\frac{\partial\psi}{\partial\lambda_-} = \sqrt{\frac{1+\lambda_+}{1+\lambda_-}}$$

One can easily verify that the ψ -function verifying these properties is indeed

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

□

We addressed the problem of linking the diffusion tensors $\mathbf{D}$ appearing in divergence-based regularization PDE's, with a corresponding ψ -functional minimization. We show how previously proposed variational methods for multivalued image regularization may be expressed within this framework, with particular choices of ψ -functions. But one step left has to be done : how can we link these generic divergence equations to regularization formulations based on *oriented 1D Laplacians*, as described in 2.4.1 ? This is the matter of the following part.

3.2 From divergence forms to oriented Laplacians

As we noticed in section 2.4, there is a close link between *oriented 1D Laplacians* (i.e directional second derivatives) and the *trace-based equations* :

$$\frac{\partial \mathbf{I}}{\partial t} = c_1 \mathbf{I}_{\mathbf{u}\mathbf{u}} + c_2 \mathbf{I}_{\mathbf{v}\mathbf{v}} \quad \Longleftrightarrow \quad \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) \quad (i = 1..n) \quad (3.17)$$

where $\mathbf{H}_i$ is the Hessian of the image $\mathbf{I}$ and $\mathbf{T}$ is the diffusion tensor whose eigenvalues are c_1, c_2 and corresponding orthogonal eigenvectors are $\mathbf{u}, \mathbf{v}$:

$$\mathbf{T} = c_1 \mathbf{u}\mathbf{u}^T + c_2 \mathbf{v}\mathbf{v}^T$$

In the followings, we first show the intuition behind oriented Laplacians and trace-based PDE's (3.17), in terms of *local geometric smoothing*. Then, we will develop PDE's expressed with divergence forms, into a unifying trace-based equation that gathers previously proposed multivalued regularization methods. Our believe is that the tensor shapes in trace-based equations give the *real* local geometry of the diffusion process, contrary to the ones in divergence-based PDE's.

3.2.1 Geometric interpretation of oriented Laplacians

In section 2.1.1, we reminded the work of Koenderink [108] who noticed that the well know 2D heat equation acting on scalar images $I : \Omega \rightarrow \mathbb{R}$ may be seen as a convolution process. It can be naturally extended for multivalued images $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$, using a channel by channel convolution. The following equation

$$\begin{cases} \mathbf{I}_{(t=0)} &= \mathbf{I}_0 \\ \frac{\partial I_i}{\partial t} &= \Delta I_i \end{cases} \quad (i = 1..n) \quad (3.18)$$

has a simple analytic solution at time t :

$$I_{i(t)} = I_{i_0} * G^t \quad (i = 1..n)$$

where G^t is a 2D normalized Gaussian kernel whose expression is :

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

This simple diffusion process is *isotropic*, since it smooths the image with the same weight in all directions. Note that

$$\Delta I_i = \text{trace}(\mathbf{Id}\mathbf{H}_i)$$

i.e, (3.18) can be seen as a trace-based equation as (3.17) where the diffusion tensor $\mathbf{T}$ in the trace operator is the *identity matrix* $\mathbf{Id}$, i.e an *isotropic tensor*.

Actually, we can prove the same kind of result for the larger class of *anisotropic* smoothing processes, similar to (3.18) but with a generic diffusion tensor $\mathbf{T}$ in the trace operator, instead of

the identity matrix $\mathbf{Id}$:

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_0 \\ \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i) \end{cases} \quad (i = 1..n) \quad (3.19)$$

where $\mathbf{T}$ is a *constant* diffusion tensor over the domain Ω .

Similarly to the heat equation, the solution of (3.19) is the convolution of the multivalued image $\mathbf{I}$ (channel by channel) by a particular *2D oriented Gaussian kernel* $G^{(\mathbf{T},t)}$:

$$I_{i(t)} = I_{i_0} * G^{(\mathbf{T},t)} \quad (i = 1..n)$$

where $G^{(\mathbf{T},t)}$ is defined as :

$$G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t} \exp\left(-\frac{\mathbf{x}^T \mathbf{T}^{-1} \mathbf{x}}{4t}\right) \quad \text{where } \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.20)$$

This normalized Gaussian function is oriented by the two eigenvectors of the diffusion tensor $\mathbf{T}$, and has corresponding variances depending both of the eigenvalues of $\mathbf{T}$ and the diffusion time t .

Proof : From the expression (3.20), 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

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

And as the convolution is a linear operation, we have

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

as well as

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

since the Gaussian function $G^{(\mathbf{T}, t)}$ is normalized. $\square$

It is very interesting, since it makes the link between a certain class of anisotropic diffusion PDE's and classical filtering techniques :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) \iff I_{i(t)} = I_{i(t=0)} * G^{(\mathbf{T}, t)} \quad (3.21)$$

with

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

This is illustrated on Fig.3.1, for different tensors $\mathbf{T}$. In the figure captions, we denoted by λ_1, λ_2 the eigenvalues of $\mathbf{T}$, and by $\theta \in \mathbb{R}$ the polar angle of the eigenvector $\mathbf{u}_1$ associated with λ_1 :

$$\mathbf{u}_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad \text{and} \quad \mathbf{u}_2 = \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix} \perp \mathbf{u}_1 \quad \text{i.e.} \quad \mathbf{T} = \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T + \lambda_2 \mathbf{u}_2 \mathbf{u}_2^T$$

Actually, we may notice that the Gaussian kernel images are *exactly* the ellipsoids that would be used to represent the diffusion tensor $\mathbf{T}$ (see section 1.2).

Note that the shape of the tensor $T = c_1 \mathbf{u}\mathbf{u}^T + c_2 \mathbf{v}\mathbf{v}^T$ can be easily understood from the oriented laplacian version of the trace-based equation (3.17) :

$$\frac{\partial I}{\partial t} = c_1 \mathbf{I}_{\mathbf{u}\mathbf{u}} + c_2 \mathbf{I}_{\mathbf{v}\mathbf{v}} \quad \text{where } \mathbf{u} \perp \mathbf{v}$$

This equation can be seen as two coexistent smoothing processes in orthogonal directions $\mathbf{u}$ and $\mathbf{v}$, weighted respectively by c_1 and c_2 . It means that the corresponding gaussian kernel $G^{(\mathbf{T}, t)}$ used for the convolution, is naturally stretched in the direction $\mathbf{u}$ if $c_1 > c_2$, and the direction $\mathbf{v}$ if $c_1 < c_2$, leading to clearly anisotropic smoothing effect.

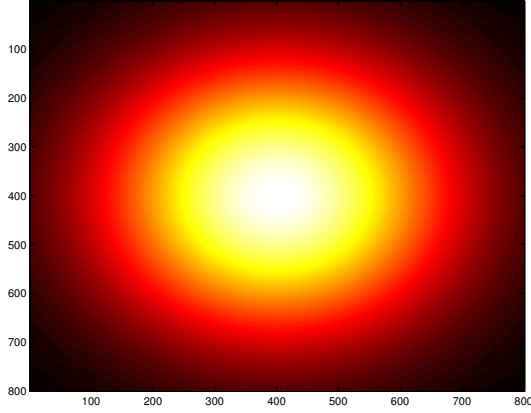
Note also that the extension of (3.21) for images defined in higher dimension domains $\Omega \subset \mathbb{R}^p$ (where $p > 2$) is easy and leads to the same equation : $\mathbf{T}$ is then a $p \times p$ tensor, and the corresponding convolution mask is a p -dimensional oriented Gaussian function.

Suppose now that $\mathbf{T}$ is *not constant* but models a *general field* of diffusion tensors. Then, the trace equation (3.21) becomes *nonlinear*, and doesn't correspond anymore to the convolution of the image $\mathbf{I}$ by a Gaussian kernel.

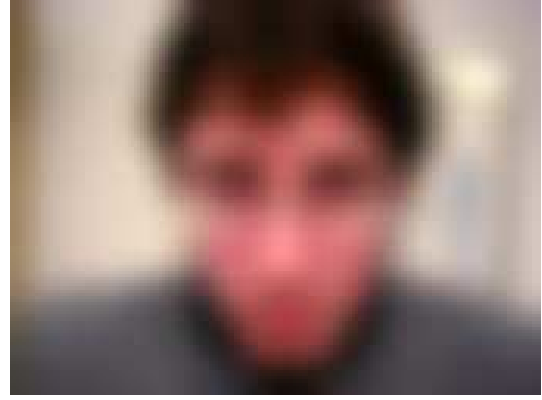
Nevertheless, the fact is that we can see this nonlinear process as the application of *very local convolutions* at each image point. This is a way of applying temporally and spatially varying masks over the image. This can be assimilated to *local filtering* methods, as proposed in [20, 171] (concept of *Bilateral Filtering*), and [163] (*Filters and Short Time Kernels* for the Beltrami flow).

Two examples of the trace-based equation $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i)$ with spatially varying tensor fields $\mathbf{T}$ are illustrated on Fig.3.2. The PDE has been applied on a color image ($n = 3$).

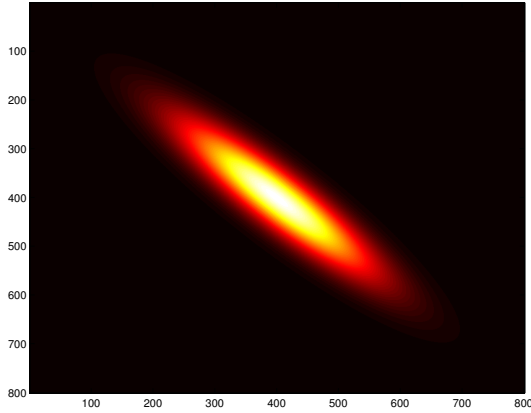
Note how the form of the tensor field $\mathbf{T}$ gives precisely *the geometry of the local smoothing* that is applied on the image. With the trace-based PDE (3.21), it is then very easy to predict the result



(a1) Gaussian kernel $\mathbf{G}^{(\mathbf{T}_1, t)}$ with $t = 1$, $\lambda_1 = \lambda_2 = 1$ and $\theta \in \mathbb{R}$.



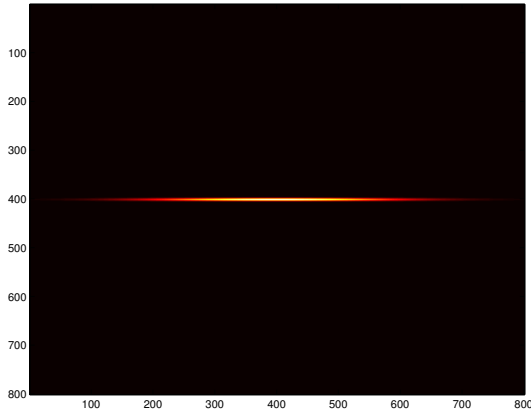
(a2) Corresponding PDE flow $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_1 \mathbf{H}_i)$, at $t = 1$



(b1) Gaussian kernel $\mathbf{G}^{(\mathbf{T}_2, t)}$ with $t = 1$, $\lambda_1 = 1, \lambda_2 = 0.05$ and $\theta = -\frac{\pi}{4}$.



(b2) Corresponding PDE flow $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_2 \mathbf{H}_i)$, at $t = 1$



(c1) Gaussian kernel $\mathbf{G}^{(\mathbf{T}_3, t)}$ with $t = 1$, $\lambda_1 = 1, \lambda_2 = 0.002$ and $\theta = 0$.



(c2) Corresponding PDE flow $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_3 \mathbf{H}_i)$, at $t = 1$

Figure 3.1: The link between diffusion tensors and oriented Gaussian kernels.

of such regularization processes, only by looking at the field of diffusion tensors $\mathbf{T}$, used in the trace operator.

Thus, one may design specific diffusion tensor fields $\mathbf{T}$, in order to adapt the regularization to the local geometry of the considered image, or even take into account some geometric features coming from additional datasets, allowing to introduce *regularization priors*.

In the next section, we will show that the generic multivalued regularization PDE's as $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D}\nabla I_i)$ which gathers a lot of proposed multivalued approaches, can be written in an extended trace-based equation, using diffusion tensors.

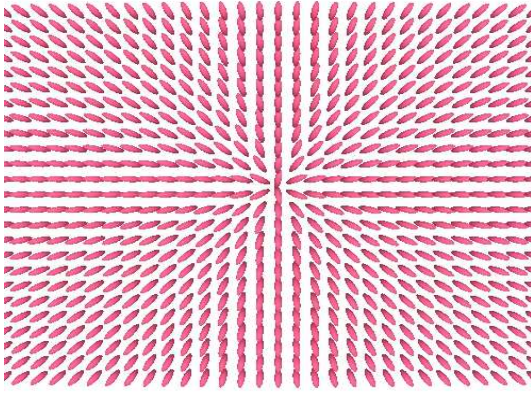
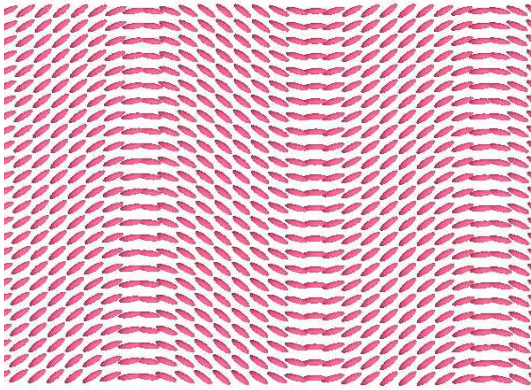
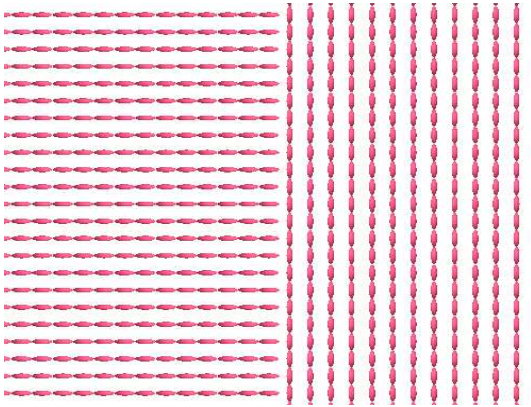
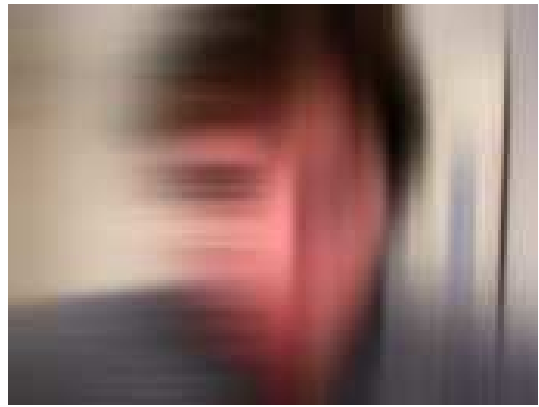
(a) Tensor field $\mathbf{T}_1 : \Omega \rightarrow \mathbb{P}(2)$ (b) Smoothed image with $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_1 \mathbf{H}_i)$ (c) Direction field $\mathbf{T}_2 : \Omega \rightarrow \mathbb{P}(2)$ (d) Smoothed image with $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_2 \mathbf{H}_i)$ (c) Direction field $\mathbf{T}_3 : \Omega \rightarrow \mathbb{P}(2)$ (d) Smoothed image with $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}_3 \mathbf{H}_i)$

Figure 3.2: Example of the PDE flow $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i)$, with non-constant tensor fields $\mathbf{T}$.

3.2.2 Difference with divergence-based equations

To understand the link between divergence based equations $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D}\nabla I_i)$, and nonlinear trace-based regularizations $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i)$, let us denote the field of diffusion tensor $\mathbf{D}$ by :

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

Note that a, b, c are functions $\Omega \rightarrow \mathbb{R}$. We have then :

$$\begin{aligned} \frac{\partial I_i}{\partial t} &= \text{div}(\mathbf{D}\nabla I_i) \\ &= \text{div} \begin{pmatrix} a I_{i_x} + b I_{i_y} \\ b I_{i_x} + c I_{i_y} \end{pmatrix} \\ &= \frac{\partial}{\partial x} (a I_{i_x} + b I_{i_y}) + \frac{\partial}{\partial y} (b I_{i_x} + c I_{i_y}) \\ &= a I_{i_{xx}} + 2b I_{i_{xy}} + c I_{i_{yy}} + I_{i_x} \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \right) + I_{i_y} \left(\frac{\partial b}{\partial x} + \frac{\partial c}{\partial y} \right) \\ &= \text{trace}(\mathbf{D}\mathbf{H}_i) + \nabla I_i^T \vec{\text{div}}(\mathbf{D}) \end{aligned} \quad (3.22)$$

where we defined $\vec{\text{div}}()$ as the divergence operator acting on matrix fields, returning variation vectors :

$$\vec{\text{div}} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \text{div} \begin{pmatrix} a \\ b \end{pmatrix} \\ \text{div} \begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_x + b_y \\ c_x + d_y \end{pmatrix}$$

- If $\mathbf{D}$ is a field of *constant* diffusion tensors, we have $\vec{\text{div}}(\mathbf{D}) = \vec{0}$ and (3.22) simplifies to :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i)$$

In this case, the two diffusion processes (3.22) and (3.19) are equivalent, and the diffusion tensor $\mathbf{D}$ is really related to the local smoothing behavior of the divergence equation $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D}\nabla I_i)$.

- In the more general case when $\mathbf{D}$ spatially varies, an *additional term* $\nabla I_i^T \vec{\text{div}}(\mathbf{D})$ depending on the *variations of* $\mathbf{D}$ differentiates the two diffusion methods expressed with the trace operator (3.19) and with the divergence-based operator :

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

This additional term may not be negligible in regard to the trace term $\text{trace}(\mathbf{D}\mathbf{H}_i)$ that smoothes locally the image with respect to the shape of $\mathbf{D}$, as we noticed in the previous section 3.2.1

Actually, it is possible to develop $\nabla I_i^T \vec{\text{div}}(\mathbf{D})$ in order to express it also with a trace operator. This additional trace-based term perturbs then the initial smoothing given by $\text{trace}(\mathbf{D}\mathbf{H}_i)$.

In the next section, we show that the resulting trace-based equation will not be driven only by the tensor $\mathbf{D}$, but will depend also on *different diffusion tensors* $\mathbf{Q}^{ij}$ that may change the diffusion directions.

Moreover, it will be used to unify trace-based and divergence-based equations, in a common formalism.

3.2.3 Development of the divergence form

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

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) \quad (i = 1..n) \quad (3.23)$$

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 $\text{div}() = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$ is itself a first order derivative operator, we should be able to write (3.23) 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, such as (3.19).

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* :

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

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* $\mathbf{D}$:

$$\text{div}(\mathbf{D} \nabla I_i) = \text{trace}(\mathbf{D} \mathbf{H}_{I_i}) + \nabla I_i^T \vec{\text{div}}(\mathbf{D}) \quad (3.24)$$

where $\vec{\text{div}}()$ is the *matrix divergence*.

A natural idea is then to decompose the additional term $\nabla I_i^T \vec{\text{div}}(\mathbf{D})$ 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

$$\mathbf{D} = f_1(\lambda_+, \lambda_-) \theta_+ \theta_+^T + f_2(\lambda_+, \lambda_-) \theta_- \theta_-^T \quad \text{with} \quad f_{1/2} : \mathbb{R}^2 \rightarrow \mathbb{R} \quad (3.25)$$

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, this is the case for our generic ψ -functional regular-

ization PDE in section 3.1, where

$$\frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i) \quad \text{with} \quad (3.25) \quad \text{and} \quad \begin{cases} f_1(\lambda_+, \lambda_-) = 2 \frac{\partial \psi}{\partial \lambda_+} \\ f_2(\lambda_+, \lambda_-) = 2 \frac{\partial \psi}{\partial \lambda_-} \end{cases}$$

In order to develop the additional diffusion term $\nabla I_i^T \vec{\operatorname{div}}(\mathbf{D})$ in the equation (3.24), we propose to write $\mathbf{D}$ as a linear combination of $\mathbf{G}$ and $\mathbf{Id}$:

$$\mathbf{D} = \alpha(\lambda_+, \lambda_-)\mathbf{G} + \beta(\lambda_+, \lambda_-)\mathbf{Id} \quad (3.26)$$

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

$$\alpha = \frac{f_1(\lambda_+, \lambda_-) - f_2(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \quad \text{and} \quad \beta = \frac{\lambda_+ f_2(\lambda_+, \lambda_-) - \lambda_- f_1(\lambda_+, \lambda_-)}{\lambda_+ - \lambda_-} \quad (3.27)$$

Indeed, we have

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

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

$$\vec{\operatorname{div}}(\mathbf{D}) = \alpha \vec{\operatorname{div}}(\mathbf{G}) + \mathbf{G} \nabla \alpha + \nabla \beta \quad (3.28)$$

and the additional term of the equation (3.24) would be computed as :

$$\begin{aligned} \nabla I_i^T \vec{\operatorname{div}}(\mathbf{D}) &= \operatorname{trace} \left(\vec{\operatorname{div}}(\mathbf{D}) \nabla I_i^T \right) \\ &= \alpha \operatorname{trace} \left(\vec{\operatorname{div}}(\mathbf{G}) \nabla I_i^T \right) \end{aligned} \quad (3.29)$$

$$+ \operatorname{trace}(\mathbf{G} \nabla \alpha \nabla I_i^T) \quad (3.30)$$

$$+ \operatorname{trace}(\nabla \beta \nabla I_i^T) \quad (3.31)$$

In the following, we propose to find formal expressions of (3.29), (3.30) and (3.31).

- 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 :

$$\begin{aligned}
\vec{\text{div}}(\mathbf{G}) &= \sum_{j=1}^n \vec{\text{div}} \begin{pmatrix} I_{jx}^2 & I_{jx}I_{jy} \\ I_{jx}I_{jy} & I_{jy}^2 \end{pmatrix} \\
&= \sum_{j=1}^n \begin{pmatrix} 2 I_{jx}I_{jxx} + I_{jx}I_{jyy} + I_{jy}I_{jxy} \\ I_{jx}I_{jxy} + I_{jy}I_{jxx} + 2 I_{jy}I_{jyy} \end{pmatrix} \\
&= \sum_{j=1}^n \begin{pmatrix} I_{jx}(I_{jxx} + I_{jyy}) \\ I_{jy}(I_{jxx} + I_{jyy}) \end{pmatrix} + \begin{pmatrix} I_{jx}I_{jxx} + I_{jy}I_{jxy} \\ I_{jx}I_{jxy} + I_{jy}I_{jyy} \end{pmatrix} \\
&= \sum_{j=1}^n \Delta I_j \nabla I_j + \mathbf{H}_j \nabla I_j
\end{aligned}$$

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 3.29 as :

$$\alpha \text{trace} \left(\vec{\text{div}}(\mathbf{G}) \nabla I_i^T \right) = \sum_{j=1}^n \alpha \text{trace} \left(\mathbf{H}_j [\nabla I_i^T \nabla I_j \mathbf{Id} + \nabla I_j \nabla I_i^T] \right) \quad (3.32)$$

□

• We finally have to compute $\nabla \alpha$ and $\nabla \beta$, in the expression (3.30) and (3.31). This can be done by the decomposition :

$$\nabla \alpha = \frac{\partial \alpha}{\partial \lambda_+} \nabla \lambda_+ + \frac{\partial \alpha}{\partial \lambda_-} \nabla \lambda_- \quad \text{and} \quad \nabla \beta = \frac{\partial \beta}{\partial \lambda_+} \nabla \lambda_+ + \frac{\partial \beta}{\partial \lambda_-} \nabla \lambda_- \quad (3.33)$$

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

$$\begin{aligned}
\nabla \lambda_{\pm} &= \begin{pmatrix} \lambda_{\pm x} \\ \lambda_{\pm y} \end{pmatrix} \\
&= \sum_{j=1}^n \begin{pmatrix} \frac{\partial \lambda_{\pm}}{\partial I_{jx}} I_{jxx} + \frac{\partial \lambda_{\pm}}{\partial I_{jy}} I_{jxy} \\ \frac{\partial \lambda_{\pm}}{\partial I_{jx}} I_{jxy} + \frac{\partial \lambda_{\pm}}{\partial I_{jy}} I_{jyy} \end{pmatrix} \\
&= \sum_{j=1}^n \mathbf{H}_{I_j} \begin{pmatrix} \frac{\partial \lambda_{\pm}}{\partial I_{xj}} \\ \frac{\partial \lambda_{\pm}}{\partial I_{yj}} \end{pmatrix}
\end{aligned}$$

In section 3.1, we derivated eigenvalues of a structure tensor $\mathbf{G}$, with respect to the spatial image derivatives. We ended up with the following relation :

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

Then,

$$\nabla \lambda_{\pm} = \sum_{j=1}^n 2\mathbf{H}_j \theta_{\pm} \theta_{\pm}^T \nabla I_j \quad (3.34)$$

We can replace (3.34) into the expressions of (3.33), in order to find the spatial gradients of α and β :

$$\begin{cases} \nabla \alpha = \sum_{j=1}^n 2\mathbf{H}_j \left(\frac{\partial \alpha}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \alpha}{\partial \lambda_-} \theta_+ \theta_+^T \right) \nabla I_j \\ \nabla \beta = \sum_{j=1}^n 2\mathbf{H}_j \left(\frac{\partial \beta}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \beta}{\partial \lambda_-} \theta_+ \theta_+^T \right) \nabla I_j \end{cases} \quad (3.35)$$

Using (3.35), we finally compute the two missing parts (3.30) and (3.31) of the additional term $\nabla I_i^T \vec{\text{div}}(\mathbf{D})$:

$$\begin{cases} \text{trace}(\mathbf{G} \nabla \alpha \nabla I_i^T) = \sum_{j=1}^n \text{trace} \left(2 \mathbf{G} \mathbf{H}_j \left(\frac{\partial \alpha}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \alpha}{\partial \lambda_-} \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \right) \\ \text{trace}(\nabla \beta \nabla I_i^T) = \sum_{j=1}^n \text{trace} \left(2 \mathbf{H}_j \left(\frac{\partial \beta}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \beta}{\partial \lambda_-} \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \right) \end{cases} \quad (3.36)$$

□

- The final step consists in putting together the equations (3.32) and (3.36), in order to express the additional term $\nabla I_i^T \vec{\text{div}}(\mathbf{D})$ in the PDE (3.24).

$$\nabla I_i^T \vec{\text{div}}(\mathbf{D}) = \sum_{j=1}^n \text{trace}(\mathbf{H}_j \mathbf{P}^{ij}) \quad (3.37)$$

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

$$\begin{aligned} \mathbf{P}^{ij} &= \alpha \nabla I_i^T \nabla I_j \mathbf{Id} \\ &+ 2 \left(\frac{\partial \alpha}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \alpha}{\partial \lambda_-} \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \mathbf{G} \\ &+ 2 \left(\left(\alpha + \frac{\partial \beta}{\partial \lambda_+} \right) \theta_+ \theta_+^T + \left(\alpha + \frac{\partial \beta}{\partial \lambda_-} \right) \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \end{aligned} \quad (3.38)$$

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 (3.37) only with symmetric matrices, in order to interpret it as a sum of local smoothing processes oriented by *diffusion tensors* (as described in section 3.2). Fortunately, the trace operator has this simple property :

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

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

$$\mathbf{Q}^{ij} = \frac{\mathbf{P}^{ij} + \mathbf{P}^{ijT}}{2} \quad (3.39)$$

and we have :

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

Finally, the divergence-based PDE (3.24) can be written as :

$$\text{div}(\mathbf{D} \nabla I_i) = \sum_{j=1}^n \text{trace}((\delta_{ij} \mathbf{D} + \mathbf{Q}^{ij}) \mathbf{H}_j) \quad (3.40)$$

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

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

□

The regularization PDE (3.40) is equivalent to the divergence-based equation $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i)$, but with a trace-based formulation.

It clearly shows that the local smoothing behavior of a divergence-based PDE as $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i)$ is generally not given by the single diffusion tensor $\mathbf{D}$, but results from a sum of several diffusion contributions of each channel I_j in each channel I_i (with $i, j = 1 \dots n$), involving additional tensors $\mathbf{Q}^{ij}$.

The local geometric interpretation of (3.40) is not as simple as the single trace equation $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i)$, but it gives an idea of the coupling between vector components (through the matrices $\mathbf{Q}^{ij}$, with $i \neq j$), as well as the basic smoothing directions in each channel I_i (through the matrices $\mathbf{Q}^{ii}$).

Actually, the fact is that (3.40) is purely *pointwise defined*, and is then more locally significant than the corresponding divergence form, which implicitly introduces the notion of *diffusion tensor variations*, through the differential divergence operator. It is an interesting stage that makes the link between the formalism of classical nonlinear regularization PDE's and the approach of bilateral filtering, as proposed in [20, 171, 163].

Moreover, the equation (3.40) gives a general form of vector-valued regularization PDE's, as it will be detailed now.

3.3 A unified expression for multivalued regularization PDE's

3.3.1 The general formula

In previous sections, we first derivated regularizing functionals into specific divergence-based equations. Then, we developed general divergence-based PDE's into specific trace-based equations. At each development step, we get into more local interpretations. Reaching the level of trace-based equations, we can now define a general form that unifies the three different levels of regularization approaches.

We define the following *generic multivalued regularization PDE* as :

$$\frac{\partial I_i}{\partial t} = \sum_{j=1}^n \text{trace}(\mathbf{A}^{ij} \mathbf{H}_i) \quad (i = 1..n) \quad (3.41)$$

where the $\mathbf{A}^{ij}$ forms a family of 2×2 symmetric matrices.

Actually, this expression can be written with a slight abuse of notations, in a *super-matrix* form¹ :

$$\boxed{\frac{\partial \mathbf{I}}{\partial t} = \vec{\text{trace}}(\mathcal{A}\mathcal{H})} \quad (3.42)$$

where $\mathcal{A}$ is a *matrix of diffusion tensors* $\mathbf{A}^{ij}$ (and is itself considered as *symmetric*) :

$$\mathcal{A} = \begin{pmatrix} \mathbf{A}^{11} & \dots & \mathbf{A}^{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{1n} & \dots & \mathbf{A}^{nn} \end{pmatrix}$$

and $\mathcal{H}$ is a *vector of Hessian matrices* $\mathbf{H}_j$:

$$\mathcal{H} = \begin{pmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_n \end{pmatrix}$$

□

The matrix product $\mathcal{A}\mathcal{H}$ is seen *sub-matrix per sub-matrix* there, and the operator $\vec{\text{trace}}()$ returns the vector in $\mathbb{R}^n$, corresponding to the trace of each sub-matrix in the resulting vector of matrices.

3.3.2 The link with other approaches

The equation (3.42) is actually a *natural extension* of the studied trace-based PDE $\frac{\partial I}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i)$. Its form follows readily the decomposition of divergence-based equations, and unifies all the previously proposed multivalued regularization PDE's :

¹This notation has been introduced by N. Sochen, after fruitful discussions.

- **Variational formulations and divergence-based expression :**

The equation (3.42) can be used to express regularization PDE's based on variational formulations and divergence-based approaches (see section 3.2.3). Indeed, we can define the whole range of vector-valued divergence-based equations with trace operators :

$$\frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i) \quad \Longleftrightarrow \quad \frac{\partial \mathbf{I}}{\partial t} = \operatorname{trace}(\mathcal{A} \mathcal{H})$$

with the following definitions for the $\mathbf{A}^{ij}$:

$$\mathbf{A}^{ij} = \delta_{ij} \mathbf{D} + \mathbf{Q}^{ij}$$

where the $\mathbf{Q}^{ij}$ are the symmetric parts of the matrices $\mathbf{P}^{ij}$, defined as :

$$\begin{aligned} \mathbf{P}^{ij} &= \alpha \nabla I_i^T \nabla I_j \mathbf{Id} \\ &+ 2 \left(\frac{\partial \alpha}{\partial \lambda_+} \theta_+ \theta_+^T + \frac{\partial \alpha}{\partial \lambda_-} \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \mathbf{G} \\ &+ 2 \left(\left(\alpha + \frac{\partial \beta}{\partial \lambda_+} \right) \theta_+ \theta_+^T + \left(\alpha + \frac{\partial \beta}{\partial \lambda_-} \right) \theta_- \theta_-^T \right) \nabla I_j \nabla I_i^T \end{aligned} \quad (3.43)$$

This includes more particularly, the different formulations handled by the ψ -functional framework (section 3.1), such as the Beltrami flow and the Vector ϕ -functionals, but also formulations directly based on divergence forms, such as the Weickert coherence enhancing PDE's [188].

Note that in general, the $\mathbf{A}^{ij}$ (where $i \neq j$) are *non-null*, which means that divergence based equations offers a complex *diffusion coupling* between image channels I_i . This kind of coupling may not be desired for regularization purposes.

- **Multivalued oriented 1D Laplacians :**

The equation (3.42) can be also used to represent regularization PDE's based on oriented 1D Laplacians, as described in section 2.4 :

$$\frac{\partial I_i}{\partial t} = \operatorname{trace}(\mathbf{T} \mathbf{H}_i) \quad \Longleftrightarrow \quad \frac{\partial \mathbf{I}}{\partial t} = \operatorname{trace}(\mathcal{A} \mathcal{H})$$

with the following definitions for the $\mathbf{A}^{ij}$:

$$\mathbf{A}^{ij} = \delta_{ij} \mathbf{T}$$

It means that the matrix $\mathcal{A}$ is *diagonal*. In this case, the $\mathbf{A}^{ij}$ ($i \neq j$) are null and no diffusion transfers between the different image channels occurs.

This particular case includes for instance the work of Sapiro-Ringach [157] and Kornprobst-Deriche [112].

Clearly, the matrix $\mathcal{A}$ in (3.42) gives important informations about the exact regularization behaviors of the PDE flow. It can be seen as an extension of the diffusion tensor $\mathbf{T}$ in trace-based

equations like $\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i)$, where the sub-matrices $\mathbf{A}^{ii}$ of $\mathcal{A}$ correspond to the diffusion performed in the current channel I_i , while the $\mathbf{A}^{ij}$ ($i \neq j$) are representative of a kind of *diffusion energy transfer* between the vector components I_i and I_j .

$\mathcal{A}$ is then a kind of *super diffusion tensor*, characterizing the regularization process (3.42). Just like classical diffusion tensors, a natural idea would be then to *diagonalize* it, in a bloc-matrix form. This is an interesting new problem that opens perspectives in the study of regularization PDE's. In this thesis, we didn't go deeper into this "spectral decomposition" of the matrix $\mathcal{A}$.

- Illustration for the scalar case :

Anyway, we illustrate that the classical case of *scalar image regularization* ($n = 1$), based on ϕ -functions, illustrates the important role of $\mathcal{A}$ in order to understand the regularization behavior of equation (3.42). Indeed, the PDE reduces then to :

$$\frac{\partial I}{\partial t} = \text{trace}(\mathbf{A}^{11}\mathbf{H}) = \text{trace}([\mathbf{D} + \mathbf{Q}^{11}]\mathbf{H})$$

and $\mathbf{Q}^{11}$ can be computed from (3.38) :

$$\mathbf{Q}^{11} = \|\nabla I\|^2 \left[\left(3\alpha + 2\|\nabla I\|^2 \frac{\partial \alpha}{\partial \lambda_+} + 2\frac{\partial \beta}{\partial \lambda_+} \right) \eta\eta^T + \alpha \xi\xi^T \right]$$

Considering the ϕ -function framework, we have $\mathbf{D} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|}$. Following the notations (3.27), we get :

$$\alpha = 0 \quad \text{and} \quad \beta = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} = \frac{\phi'(\sqrt{\lambda_+})}{\sqrt{\lambda_+}}$$

and $\mathbf{Q}^{11}$ is in this case :

$$\mathbf{Q}^{11} = \left[\phi''(\|\nabla I\|) - \frac{1}{\|\nabla I\|} \phi'(\|\nabla I\|) \right] \eta\eta^T$$

and the diffusion tensor $\mathbf{A}^{11}$ is written as :

$$\mathbf{A}^{11} = \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \xi\xi^T + \phi''(\|\nabla I\|) \eta\eta^T$$

This tensor $\mathbf{A}^{11}$ gives the *exact local geometry* of the ϕ -functional diffusion process. It is very different from the initial tensor $\mathbf{D}$, present in the divergence term.

$$\begin{aligned} \frac{\partial I}{\partial t} &= \text{div} \left(\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \nabla I \right) \\ &= \text{trace} \left(\left[\frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} \xi\xi^T + \phi''(\|\nabla I\|) \eta\eta^T \right] \mathbf{H} \right) \\ &= \frac{\phi'(\|\nabla I\|)}{\|\nabla I\|} I_{\xi\xi} + \phi''(\|\nabla I\|) I_{\eta\eta} \end{aligned}$$

Using a more general viewpoint, we find again the divergence decomposition into oriented Laplacians that explains the anisotropic behavior of ϕ -function based regularizing PDE's.

Our approach is more global since it can decompose *multivalued* regularization PDE's into trace-based equations.

Using the generic form of regularizing PDE's (3.42), we propose now to construct a new and efficient multivalued regularization process, designed on some important local geometric constraints.

3.4 Defining a new multivalued regularization/enhancement process

Following our previous study on the multivalued regularization PDE's, we propose now to define a new diffusion PDE, based on desired *local geometric properties* that are desirable for a regularization process. These geometric behaviors will be naturally expressed through specific constraints on the subtensors of $\mathcal{A}$, in (3.42)

3.4.1 Vector geometry-adapting regularization

Our approach is based on the fact that we can define the exact smoothing behavior of a multivalued regularization process like the one defined in (3.42) :

$$\frac{\partial \mathbf{I}}{\partial t} = \text{trace}(\mathcal{A}\mathcal{H}) \quad (3.44)$$

We want to design locally the corresponding matrix $\mathcal{A}$, in order to adapt the smoothing to the local vector geometry of the image. The list of desired geometric properties is as follows :

1. We don't want to mix diffusion terms between different image channels. It corresponds to avoid *diffusion transfer* between the vector components. Then, we wish that :

$$\forall i, j \mid i \neq j, \quad \mathbf{A}^{ij} = \mathbf{0} \quad (\text{the null matrix})$$

It means that the super-matrix $\mathcal{A}$ is *diagonal*, and that we are only defining n matrices $\mathbf{A}^{ii}$. Actually, it allows us to remove the vector coupling between image components, i.e we assume that we chose an initial vector basis that is adapted to the regularization process (note that choosing the right initial vector basis is another difficult topic and is not discussed in this thesis).

2. Vector components should evolve with the same different behaviors, since they have a-priori the same importance. It particularly means that the tensors $\mathbf{A}^{ii}$ have to be equal to a single tensor $\mathbf{A}$. We define its eigenvalues by $f_+ : \Omega \rightarrow \mathbb{R}$ and $f_- : \Omega \rightarrow \mathbb{R}$, and its associated orthonormal eigenvectors by $\mathbf{u}_+$ and $\mathbf{v}_+$:

$$\mathbf{A} = f_+ \mathbf{u}_+ \mathbf{u}_+^T + f_- \mathbf{u}_- \mathbf{u}_-^T$$

3. The diffusion geometry has to be vector-valued, i.e computed from the spectral elements $\lambda_{\pm}$ and $\theta_{\pm}$ of the structure tensor $\mathbf{G}$. We will especially use the variation norm $\mathcal{N}_+ = \sqrt{\lambda_+ + \lambda_-}$ in order to discriminate the different types of image regions (for reason explained in section 2.2.2).

Following the idea of Weickert's method [184, 186, 188], we also compute the eigenvalues $\lambda_{\pm}$ and the eigenvectors $\theta_{\pm}$ from a *smoothed version* $\mathbf{G}_{\sigma}$ of the structure tensor field $\mathbf{G}$:

$$\mathbf{G}_{\sigma} = \mathbf{G} * G_{\sigma} \quad \text{where} \quad G_{\sigma} = \frac{1}{4\pi t} \exp\left(-\frac{x^2 + y^2}{4t}\right)$$

We denote by $\lambda_{\pm}^*$ and $\theta_{\pm}^*$ the spectral elements of $\mathbf{G}_{\sigma}$.

Choosing these smoothed spectral elements in order to design $\mathbf{A}$ improves the regularization process thanks to the use of a *more coherent diffusion geometry*.

4. On homogeneous zones (low vector variations regions), we would like to perform an *isotropic smoothing* therein, in order to clear the noise efficiently :

$$\frac{\partial I_i}{\partial t} \simeq \Delta I_i = \text{trace}(\mathbf{H}_i) \quad \text{i.e} \quad \lim_{\mathcal{N}_+^* \rightarrow 0} \mathbf{A} = \mathbf{Id}$$

which also means that

$$\lim_{\mathcal{N}_+^* \rightarrow 0} f_+(\mathcal{N}_+^*) = \lim_{\mathcal{N}_+^* \rightarrow 0} f_-(\mathcal{N}_+^*) = 1$$

5. On vector edges (high variation regions), we would like to perform an *anisotropic smoothing along the vector edges*, in order to preserve them while removing the noise. Thus, the diffusion tensor $\mathbf{A}$ should be mainly directed by the vector edge orientation θ_-^* , in these regions :

$$\frac{\partial I_i}{\partial t} \simeq f_-(\mathcal{N}_+^*) \theta_- \theta_-^T = \text{trace}(f_-(\mathcal{N}_+^*) \theta_- \theta_-^T \mathbf{H}_i)$$

i.e

$$\lim_{\mathcal{N}_+^* \rightarrow +\infty} \frac{f_-(\mathcal{N}_+^*)}{f_+(\mathcal{N}_+^*)} = 0$$

$f_- : \mathbb{R} \rightarrow \mathbb{R}$ is then a function decreasing for very high vector variations (as f_+), in order to preserve *sharp corners*.

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

$$\left\{ \begin{array}{l} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} = \text{trace} \left(\left[f_+ \left(\sqrt{\lambda_+^* + \lambda_-^*} \right) \theta_-^* \theta_-^{*T} + f_- \left(\sqrt{\lambda_+^* + \lambda_-^*} \right) \theta_+^* \theta_+^{*T} \right] \mathbf{H}_i \right) \end{array} \right. \quad (3.45)$$

$(i = 1..n).$

The $\lambda_{\pm}^*$ and $\theta_{\pm}^*$ are the eigenvalues and eigenvectors of the smoothed structure tensor $\mathbf{G}_{\sigma}$:

$$\mathbf{G}_{\sigma} = \left(\sum_{j=1}^n \nabla I_j \nabla I_j^T \right) * G_{\sigma}$$

and the $f_{\pm} : \Omega \rightarrow \mathbb{R}$ are two functions that weight the smoothing process in the corresponding directions $\theta_{\pm}^*$. We propose for instance these two functions, inspired from the *hypersurface formulation* of the scalar case (see chapter 2) :

$$\left\{ \begin{array}{l} f_-(s) = \frac{1}{\sqrt{1+s^2}} \\ f_+(s) = \frac{1}{1+s^2} \end{array} \right. \quad (3.46)$$

□

This is of course one possible choice that verifies the above geometric properties, relying on practical experience. The point is that we can easily adapt f_- and f_+ to obtain regularization behaviors for specific problems.

This new multivalued regularization equation (3.45) smoothes locally the image, using a coherent vector-geometry directions and preserves well the edges, while avoiding undesired coupling between vector components. It can be also written in an *oriented Laplacian* style :

$$\frac{\partial \mathbf{I}}{\partial t} = f_- \mathbf{I}_{\theta_-^* \theta_-^*} + f_+ \mathbf{I}_{\theta_+^* \theta_+^*}$$

With this expression, it is clear that the regularization process consists in two oriented vector-valued heat flows, along the directions θ_+^* and θ_-^* , with corresponding weights f_+ and f_- .

Its form has steadily followed the local analysis of classical multivalued regularization algorithms. Its main interest is the knowledge of the *exact smoothing directions*.

Applying this new regularizing PDE (3.45) on the images with artificial noise presented on the comparative figures Fig.2.13 and Fig.2.14 in chapter 2 can be found below (Fig.3.3)

3.4.2 Vector-valued edge enhancement

Reducing the blurred edges can be a part of an image restoration process. We propose to add an additional *vector edge enhancement term* to our multivalued regularization PDE (3.45), based on the well-known *shock filter* formalism [9, 131].

The scalar shock filter method has been designed to enhance blurred edges in gray-valued images *without any knowledge of convolution masks that caused the blur* (Fig. 3.4).

It operates by raising the signal in the gradient direction $\eta = \frac{\nabla I}{\|\nabla I\|}$:

$$\begin{aligned} \frac{\partial I}{\partial t} &= -\text{sign}(I_{\eta\eta}) \|\nabla I\| \\ &= -\text{sign}(\eta^T \mathbf{H} \eta) \|\nabla I\| \end{aligned}$$

When dealing with vector-valued images, we would like naturally that the shock filter raises each vector channel I_i in a *common direction* θ_+^* of the vector discontinuities (θ_+^* is the main eigenvector of the smoothed structure tensor $\mathbf{G}_\sigma$).

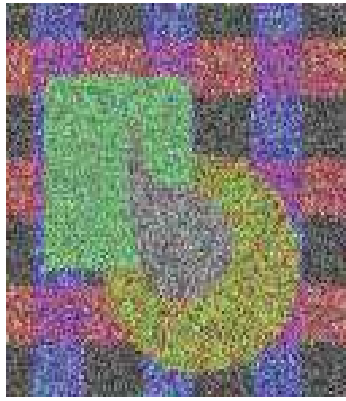
We also add a weighting function that adapts the intensity of the shock filter process, allowing to enhance vector edges, while keeping homogeneous regions unchanged.

This leads to the following *vector shock filter* equation :

$$\frac{\partial I_i}{\partial t} = - \left(1 - g \left(\sqrt{\lambda_+^* + \lambda_-^*} \right) \right) \text{sign} \left(\theta_+^{*T} \mathbf{H}_i \theta_+^* \right) |I_{i_{\theta_+^*}}| \quad (3.47)$$

where $g : \mathbb{R} \rightarrow [0, 1]$ is a decreasing function.

□



(a) Noisy synthetic color image



(b) Restored image, using the PDE (3.45)



(c) Noisy color photograph



(d) Restored image, using the PDE (3.45)

Figure 3.3: Using our vector-valued regularization PDE to regularize noisy color images.

We illustrate on Fig.3.5 the application of the vector shock filters (3.47) on a blurred color image.

Notice that shock filters cannot create new image structures, but only acts as an edge sharpener. Processed images with shock filters often look like cartoon images, since the PDE (3.47) tends to create *piecewise constant solutions*.

Combining the regularization term (3.45) with the vector-valued shock filter term (3.47) allows to *denoise and enhance vector edges* in a common PDE process. In Fig.3.6, we applied this method to restore and enhance a noisy and blurred color image. Using only regularization PDE's removes well the noise (Fig.3.6b), but the resulting image is still blurred. The coupled regularization+shock equation allows to retrieve a denoised image with sharper contours (Fig.3.6c).

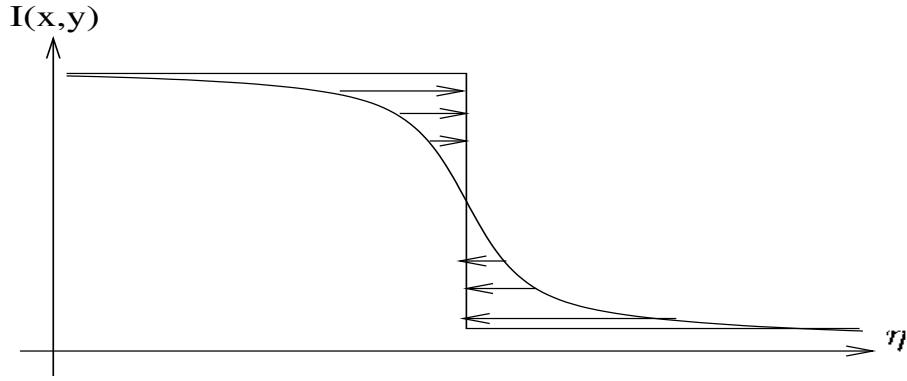
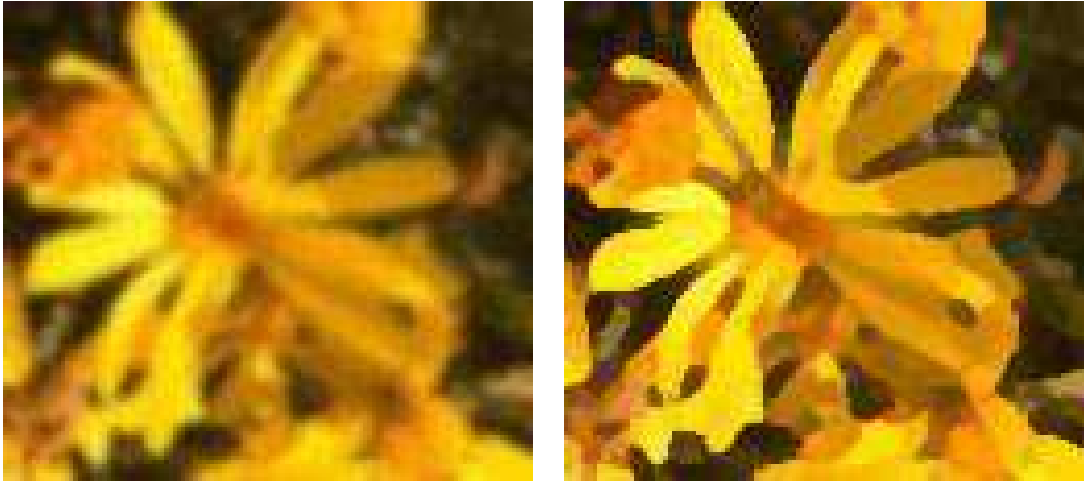


Figure 3.4: Principle of shock filters.



(a) Blurred color image

(b) Enhanced image, with vector shock filters

Figure 3.5: Vector-valued shock filters applied on a color image.

3.5 Numerical schemes for multivalued regularization

We propose two working numerical schemes, which can be used to implement the general multivalued equation (3.44), proposed in section 3.41 :

$$\frac{\partial \mathbf{I}}{\partial t} = \vec{\text{trace}}(\mathcal{A}\mathcal{H}) \quad \text{i.e.} \quad \frac{\partial I_i}{\partial t} = \sum_{j=1}^n \text{trace}(\mathbf{A}^{ij} \mathbf{H}_j) \quad (i = 1..n) \quad (3.48)$$

More particularly, these schemes have been used to implement our new regularization equation (application results can be found in chapter 4) :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) \quad \text{where } \mathbf{T} \text{ is defined as (3.45)} \quad (3.49)$$

Please note that these schemes are given as they are. The exhaustive mathematical study (uniqueness, convergence, etc.) has not been done in this thesis and could be envisaged for future works.



Figure 3.6: Mixing regularization terms and vector-valued shock filters.

3.5.1 Elementary iterative loop

For both methods, a common iterative computation loop based on classical explicit schemes for PDE implementation is used :

Considering a vector-valued image $\mathbf{I}^k$ at a particular PDE iteration of (3.48), we compute the iterated image $\mathbf{I}^{k+1}$ by following these ordered steps :

1. Compute the fields of the first spatial derivatives $\mathbf{I}_x^k$ and $\mathbf{I}_y^k$. This is done by a simple centered scheme, based on finite differences :

$$\begin{cases} I_{i_x}^k(x, y) &= 0.5 \times (I_i^k(x+1, y) - I_i^k(x-1, y)) \\ I_{i_y}^k(x, y) &= 0.5 \times (I_i^k(x, y+1) - I_i^k(x, y-1)) \end{cases}$$

Indeed, more complex first derivative estimations could be used here, as for instance the methods proposed in [63, 120, 158].

2. Compute the structure tensor field $\mathbf{G}^k$, computed as :

$$\mathbf{G}^k = \begin{pmatrix} g_{11}^k & g_{12}^k \\ g_{12}^k & g_{22}^k \end{pmatrix} \quad \text{with} \quad \begin{cases} g_{11}^k &= \sum_{j=1}^n I_{i_x}^k{}^2 \\ g_{12}^k &= \sum_{j=1}^n I_{i_x}^k I_{i_y}^k \\ g_{22}^k &= \sum_{j=1}^n I_{i_y}^k{}^2 \end{cases}$$

3. Compute the smoothed structure tensor field

$$\mathbf{G}_\sigma^k = \mathbf{G}^k * G_\sigma \quad \text{where } G_\sigma \text{ is a normalized 2D Gaussian kernel,}$$

as well as the corresponding fields of the eigenvalues $\lambda_\pm^*$ and eigenvectors $\theta_\pm^*$ of $\mathbf{G}_\sigma$. For this purpose, we can profitably replace the convolution by the application of a recursive implementation of a Gaussian filter, as described in [65].

4. Compute the matrix $\mathcal{A}$: for the general case (3.48), it consists in several sub-diffusion tensors $\mathbf{A}^{ij}$. For the simpler case (3.49), only a single diffusion tensor $\mathbf{T}$ is needed. Note that $\mathcal{A}$ depends *only* on the spectral elements of $\mathbf{G}_\sigma$, which have been already computed.
5. Compute the PDE velocity

$$\beta = \vec{\text{trace}}(\mathcal{A}\mathcal{H})$$

This is done by estimating each sub-term such as $\text{trace}(\mathbf{A}^{ij}\mathbf{H}_j)$, using one of the two different methods proposed in the sections 3.5.2 and 3.5.3. Note that for the PDE (3.49), only one trace term is needed.

6. Finally, the new iterated image $\mathbf{I}^{k+1}$ is estimated with a classical explicit scheme :

$$\mathbf{I}^{k+1} = \mathbf{I}^k + \text{dt } \beta$$

Particularly, we propose to choose the time-step dt in an *adapting* way. More precisely, it is computed in order to limit the amount of pixel variations at each PDE iteration :

$$\text{dt} = \frac{\beta_{\max}}{\max_{\mathbf{x} \in \Omega} \|\beta(\mathbf{x})\|}$$

The parameter $\beta_{\max}$ is the maximum variation allowed for the points of the regularizing image, during one iteration. It can be fixed by hand, or computed at each iteration as a fixed percentage of the maximum intensity variation in the processing image. On one hand, it limits discontinuous variations of the image (vector) intensities, during the PDE flow. On the other hand, it speeds up the process by ensuring that at least one image point is evolving with a speed $\beta_{\max}$.

Note that this adapting time-step technique can be generally used for the implementation of other various PDE flows.

This computation loop is then repeated for a fixed number of iterations.

3.5.2 Using a spatial discretization of the Hessian

This is the most simple and fast solution, based on a direct discretization of the vector $\mathcal{H}$, whose elements are the Hessian matrices $\mathbf{H}_j$ of the different image channels I_j . As for first derivatives, those Hessians are estimated with classical finite (centered) difference schemes :

$$\mathbf{H}_i = \begin{pmatrix} I_{i_{xx}}^k & I_{i_{xy}}^k \\ I_{i_{xy}}^k & I_{i_{yy}}^k \end{pmatrix} \quad \text{with} \quad \begin{cases} I_{i_{xx}}^k &= I_i^k(x+1, y) + I_i^k(x-1, y) - 2 I_i^k(x, y) \\ I_{i_{yy}}^k &= I_i^k(x, y+1) + I_i^k(x, y-1) - 2 I_i^k(x, y) \\ I_{i_{xy}}^k &= 0.25 \times (I_i^k(x+1, y+1) + I_i^k(x-1, y-1) \\ &\quad - I_i^k(x-1, y+1) - I_i^k(x+1, y-1)) \end{cases}$$

Then, we can easily compute the term $\vec{\text{trace}}(\mathcal{A}\mathcal{H})$ of the different diffusion contributions in the PDE velocity β , using simple matrix multiplications.

Experimentally, we noticed that this scheme gives very similar results as the classical 0.5 centered schemes for divergence discretization, as proposed in [5, 12].

3.5.3 Using local filtering considerations

Another method to estimate a term such as $\text{trace}(\mathbf{T}\mathbf{H}_i)$ at a certain time t , is to follow the geometric interpretation in term of local filtering, as described in section (3.2.1). We quickly remind the idea. If $\mathbf{T}$ is a constant diffusion tensor, we have :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) \iff I_{i(t)} = I_{i(t=0)} * G_{\sigma}^{(\mathbf{T},t)} \quad \text{with} \quad G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t} \exp\left(-\frac{\mathbf{x}^T \mathbf{T}^{-1} \mathbf{x}}{4t}\right)$$

If $\mathbf{T}$ is spatially varying (which is obviously the case for (3.48) and (3.49)), one may consider the regularization process as the application of *local convolution masks* over the image, leading to schemes similar to *local and Bilateral filtering* [20, 163, 171].

Estimating a term as $\text{trace}(\mathbf{T}\mathbf{H}_i)$ can then be done as follows. For each point $\mathbf{x} = (x, y) \in \Omega$:

- We first compute the local convolution mask $G^{(\mathbf{T},dt)}$, defining the local geometry of the smoothing that must be performed at the point $\mathbf{x}$. Its form follows the corresponding diffusion tensor $\mathbf{T}(\mathbf{x})$, inside the trace operator.

Then, the term $\text{trace}(\mathbf{T}\mathbf{H}_i)$ is estimated as the local application of this mask $G^{(\mathbf{T},dt)}$, in the local neighborhood of $\mathbf{x}$ (illustrated below with a 3×3 mask $\mathbf{G} = \mathbf{G}^{(\mathbf{T},\mathbf{H}_i)}$) :

$$[\text{Trace}(\mathbf{T}\mathbf{H})](x,y) = \left(\begin{array}{|c|c|c|} \hline i(x-1,y-1) & i(x,y-1) & i(x+1,y-1) \\ \hline i(x-1,y) & i(x,y) & i(x+1,y) \\ \hline i(x-1,y+1) & i(x,y+1) & i(x+1,y+1) \\ \hline \end{array} \star \begin{array}{|c|c|c|} \hline \mathbf{G}(-1,-1) & \mathbf{G}(0,-1) & \mathbf{G}(1,-1) \\ \hline \mathbf{G}(-1,0) & \mathbf{G}(0,0) & \mathbf{G}(1,0) \\ \hline \mathbf{G}(-1,1) & \mathbf{G}(0,1) & \mathbf{G}(1,1) \\ \hline \end{array} \right) (0,0)$$

- This local filtering is repeated for each trace $(\mathbf{A}^{ij}\mathbf{H}_j)$ in the vector $\vec{\text{trace}}(\mathcal{AH})$, in order to estimate all these subsequent terms.

The size of the convolution mask is an important factor of the computational time of the algorithm. For our experiments, we used 3×3 masks.

A noticeable point is that this scheme naturally *preserves the maximum principle* for the PDE (3.49) (containing only one diffusion tensor). In this case, it is equivalent to apply locally a *normalized smoothing mask* directly on the image $\mathbf{I}$, then the pixel values cannot step out of the value range of the initial image $\mathbf{I}_{(t=0)}$.

3.5.4 Comparative figure

We applied our vector-valued regularization PDE (3.45) on a noisy color image, with the two different numerical schemes, in order to illustrate the advantages and drawbacks of each proposed schemes (Fig.3.7).

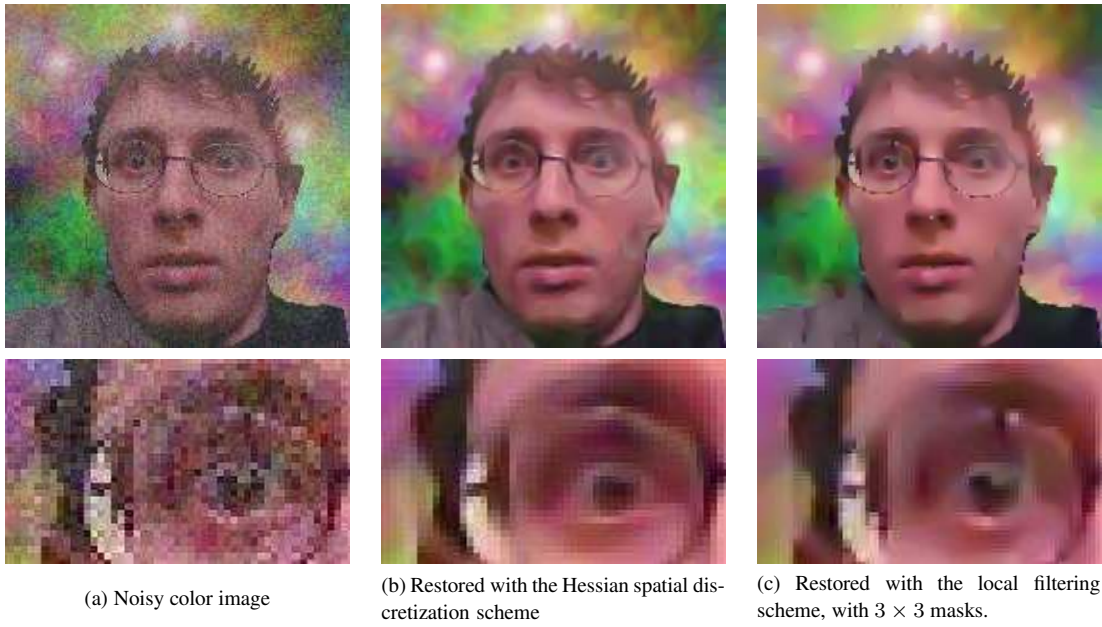


Figure 3.7: Comparison of the two proposed numerical schemes.

The size of the considered images is 180×180 , and the color base used for the restoration is (R, G, B) . For both schemes, the regularizing flow (3.45) has been carried out after 130 iterations. The two obtained results are well restored : noise is removed, while important image structures are preserved.

Anyway, we notice that we lose more quickly details with the classical scheme, based on the Hessian discretization (3.5.2). As for our new local filtering-based scheme (3.5.3), very thin details are preserved over the time, as for instance the *one pixel wide* edge of the glasses. On the other hand, the computation time is in favor of the first scheme. Restoration times for this image are indeed 13.64 s for the first scheme, and 1m56.45 s for the second scheme.

In this chapter, we proposed a new framework that can handle a lot of previously proposed vector-valued regularization methods, using PDE's. We also proposed a new equation, based on geometric priors we want to follow, as well as new numerical schemes. Let us illustrate now, how we can apply it for various image processing problems.

C'est bien d'avoir une logique abstraite, mais il faudrait que ça te serve dans la vie courante, mon garçon.

Maman.

Chapter 4

Applications of Unconstrained PDE's



We illustrate how our generic multi-valued regularization framework (proposed in the previous chapter) can be successfully exploited in order to tackle different problems of interest. First, we use it to restore degraded color images : this is indeed the most natural application of vector-valued diffusion PDE's. But we also propose similar PDE-based methods to handle the problem of color image inpainting, magnification and interpolation, as well as the visualization of flows. Experimental results are commented and illustrated throughout this chapter.

4.1 Color image restoration

Image restoration has been one of the first concrete application since the apparition of scalar regularization PDE's for image processing. Let us consider an image I_{noisy} , corrupted by noise. The application of an anisotropic diffusion PDE on I_{noisy} consists in simplifying the data little by little, such that the low variations disappear first while the sharp discontinuities are preserved (scale-space principle). As a result, the noise is removed in the image, while the important structures (edges, corners) are maintained.

The recent extension of these algorithms to *vector-valued data* has particularly opened the right way to restore degraded *color images*. Despite the growing quality of analogical and digital cameras, noisy color images are still encountered in a lot of situations :

- **Old color photographs/videos :**

The appearance of color photographs dates from 1861, with the "color separation" method of the Scottish physicist James Clerk-Maxwell. It has been generalized afterwards with the first commercial color film, the Autochrome plates, manufactured by Lumiere brothers in France. As for color televisions, they have been marketed since 1951. This means that many historic color documents with noise exists (due to the old and rudimentary used technologies, or to the bad storage conditions), and may be in their interests to be scanned into a digital representation, then automatically restored.

- **Digital color photographs :**

The recent apparition of scanners and digital cameras has permitted to obtain high-quality snapshots and to store them directly on computers. One may think that the image quality has reached its highest level and that image restoration won't be necessary anymore. This is wrong. We illustrate in Fig.4.1a the detail of a *real digital color image*, obtained with an (expensive) digital camera. The noise here has not been artificially added, but results from the *very-low luminosity conditions* of the scene during the snapshot : unfortunately, the camera flash has not been released here.

In this case, a color restoration process is interesting. Fig.4.1b shows how the application of a well-designed vector-valued regularization scheme can be helpful to restore such degraded digital images. We used the following regularization equation (proposed in chapter 3 :

$$\begin{cases} \mathbf{I}_{(t=0)} &= \mathbf{I}_{\text{noisy}} \\ \frac{\partial I_i}{\partial t} &= \text{trace}(\mathbf{D}\mathbf{H}_i) \quad (i = 1, 2, 3) \end{cases} \quad (4.1)$$

where $\mathbf{H}_i$ is the Hessian of the color channel I_i (R, G or B) and $\mathbf{D}$ is the following 2×2 diffusion tensor :

$$\mathbf{D} = \frac{1}{\sqrt{1 + \lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T} + \frac{1}{1 + \lambda_+^* + \lambda_-^*} \theta_+^* \theta_+^{*T}$$

Remember that $\lambda_{\pm}^*$ and $\theta_{\pm}^*$ are the eigenvalues and the corresponding eigenvectors of the smoothed structure tensor $\mathbf{G}_{\sigma} = (\sum_{i=1}^n \nabla I_i \nabla I_i^T) * G_{\sigma}$, which is representative of a coherent local vector geometry in the image $\mathbf{I}$ (see chapters 2 and 3 for details).

The size of the initial noisy picture (Fig.4.1a) is 293×306 pixels. The restored image (Fig.4.1b) has been obtained with the application of the PDE flow (4.1) during 26 iterations with an adaptive time-step. Note how the restored image is naturally denoised without losing important image informations.

- **Lossy compressed images :**

Another modern application of color image regularization techniques can be found when dealing with compression algorithms. 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 [99]). These kinds of compression algorithms save a lot of memory space but introduce more or less visible *image artefacts*. For instance, color images compressed with the JPEG method often display visible *bloc effects*, as illustrated on Fig.4.2a. The used image here is a 376×297 pixels color image, compressed in JPEG format with a quality ratio of 10%.

As illustrated on Fig.4.2b, the use of our regularization PDE flow (4.1) improves the quality of this blocky image by reinforcing the image structures at the expense of the undesired bloc artefacts.



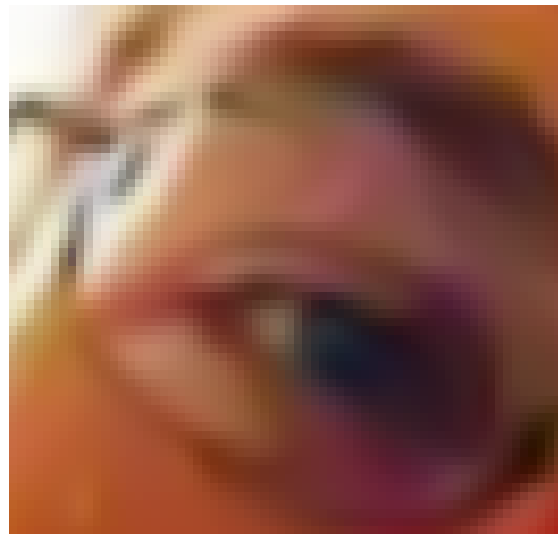
(a) Digital color image with real noise



(b) Restored image with our regularization method (4.1)



(c) Detail of the noisy image (a)



(d) Detail of the restored image (b)

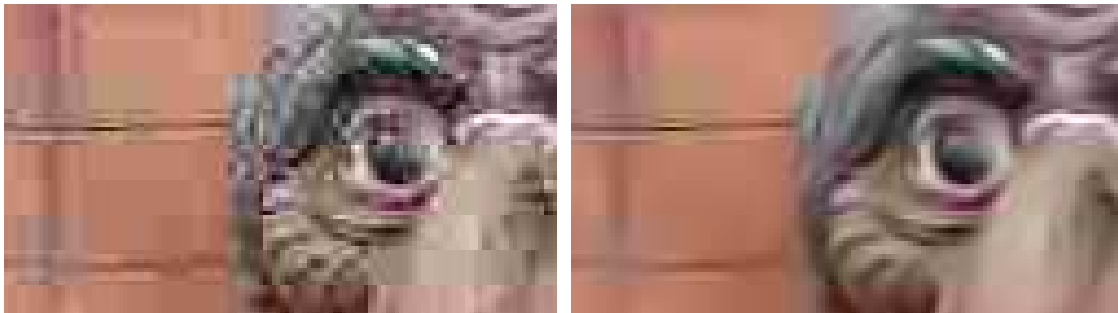
Figure 4.1: Restoration of a real noisy digital photograph.



(a) Compressed JPEG image (10% quality)



(b) Improved image with the PDE flow (4.1)



(c) Detail of compressed (left) and improved (right) images.

Figure 4.2: Improvement of a lossy compressed color image.

In both cases, the used numerical scheme is the one proposed in chapter 3, using local convolutions with geometry-adapting kernels. Actually, this scheme is well designed to preserve thin structures (such as the two pixels wide contour of the glasses in Fig.4.1, or the hair of the gnome in Fig.4.2), despite its expensive computation time (8 to 9 times slower than a classical Hessian discretization).

4.2 Color image inpainting

Diffusion PDE's are not limited to noise suppression. Recently, a new interesting application of regularization schemes named *image inpainting*, has been proposed in [26, 38, 40, 41, 44]. It consists in filling undesired holes in an image $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$, by interpolating the data located at the neighborhood of the holes. The user has only to define a *binary mask* $M : \Omega \rightarrow \{0, 1\}$, locating the desired regions to fill in the image $\mathbf{I}$. Then, a PDE-based algorithm is used to interpolate the data in a way that the image structures are coherently completed and fill the holes. This is obviously a non-trivial inverse problem that raises a lot of subjacent questions and a whole thesis could be not enough to answer them.

As for us, we will limit ourselves to the simple application of an adapted regularization PDE, specially designed to find a working and acceptable solution :

In chapter 3, we proposed PDE's that have the desired property of diffusing the image intensities in local neighborhoods, directed by some prior directions. Color image inpainting can then be done by applying these kind of vector-valued PDE *only on points or areas defined in the inpainting mask* M , while keeping the other image regions unchanged :

$$\left\{ \begin{array}{l} \mathbf{I}_{(t=0)} = \mathbf{I}_{\text{original}} \\ \forall (x, y) \in \Omega, \forall i = 1, 2, 3, \end{array} \right. \left| \begin{array}{ll} \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i) & \text{if } M(x, y) = 1 \\ \frac{\partial I_i}{\partial t} = 0 & \text{if } M(x, y) = 0 \end{array} \right. \quad (4.2)$$

with the following diffusion tensor

$$\mathbf{D} = \frac{1}{\sqrt{1 + \lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T}$$

As above, $\lambda_{\pm}^*, \theta_{\pm}^*$ are the eigenvalues and the corresponding eigenvectors of the smoothed structure tensor $\mathbf{G}_{\sigma} = (\sum_{i=1}^n \nabla I_i \nabla I_i^T) * G_{\sigma}$. This indicates the use of a real local vector geometry for the inpainting process, leading to coherent colors completions in the inpainted holes.

Contrary to the equation (4.1) used to restore color images (previous section), we don't allow isotropic smoothing here, even on flat regions (the diffusion is always done along a single direction θ_-^*). Indeed, we want only to fill-in the holes by linking existing structures together. Isotropic smoothing is well adapted to remove quickly the noise but has to be avoided when diffusing image structures (there would be a risk of structure blurring).

The parameter σ depends on the size of the region to fill in : Blurring the structure tensor field $\mathbf{G}$ emphasizes the important image structures within a larger local neighborhood and helps for the PDE interpolation. Structures that have been split by large holes may be recovered with a large variance σ .

Inpainting methods have a wide range of applications. We applied our equation (4.2) in order to tackle the following problems :

- **Removing text and advertisements from color images :**

Our algorithm can remove text or ads present in a color image. This is illustrated on Fig.4.3a, with a real photograph containing undesired text (size 640×480 pixels). Here the inpainting mask is obviously the same as the region covered by the green text.

The result of the inpainting process is shown on Fig.4.3b (needed 100 PDE iterations, with an adaptive time-step). We finally retrieve a quite realistic image. Nevertheless, interpolating image colors cannot create artificial structures that have been hidden by the inpainting mask. It is for instance visible with the eyes of the foreground woman, which have not been recovered (actually, this would be hardly possible).

- **Removing real objects :**

With the same reasoning, we can create an inpainting mask on purpose, hiding a whole object present in the image in order to make it disappear. It may have several applications for visual effects in the domain of movies and advertising. An example is illustrated in Fig.4.5 : we especially designed the inpainting mask to remove the glasses of the elegant man. The removal needed 300 iterations of (4.2), with an adaptive time-step.

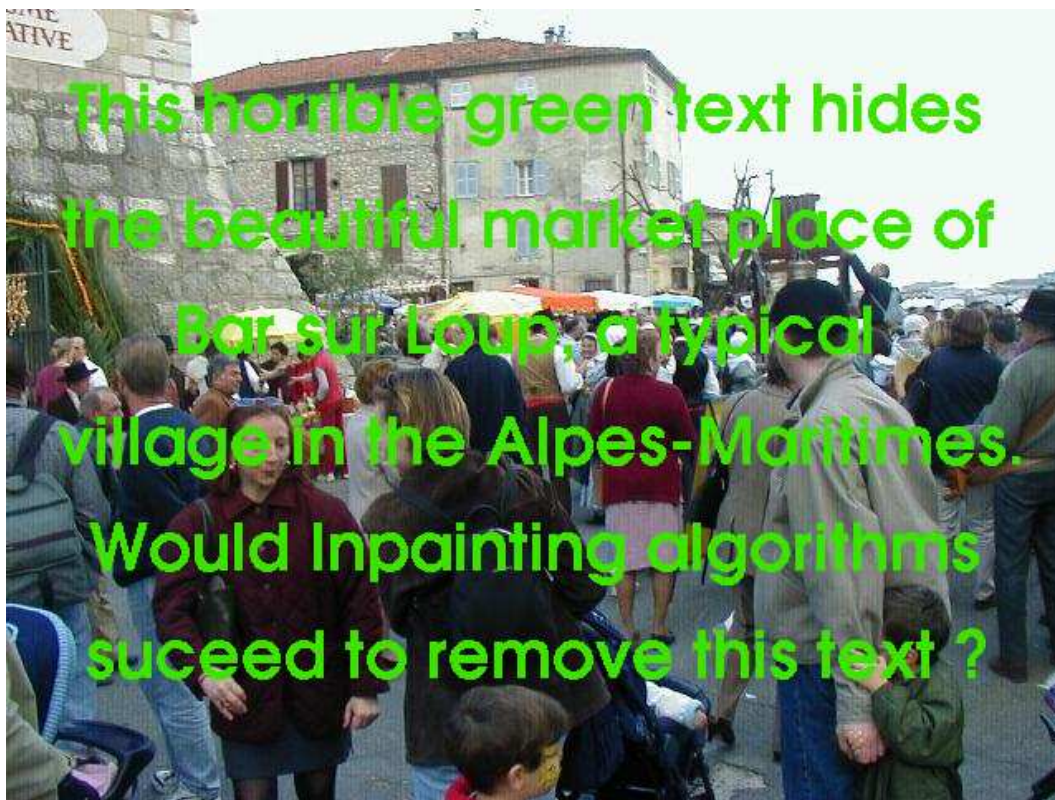
- **Image reconstruction from incomplete data :**

With this example, we want to illustrate the high potential of non-linear inpainting algorithms, for partial image reconstruction. We defined the inpainting mask as a checkerboard with small cases (4 pixels wide), allowing to retrieve only 50% of the initial image informations (Fig.4.6a,b).

The result of our inpainting flow (4.2) is illustrated on Fig.4.6d. The color image is very well reconstructed. Notice how the image structures have been correctly interpolated (particularly the woman's hair).

We have also compared our non-linear scheme with a classical linear method, consisting in reconstructing the missing part of the images with linear interpolations (Fig.4.6c). Embarrassing jaggling effects appear and spoil the general aspect of the "restored" image. The non-linear anisotropic diffusion done by our PDE method (4.2) results in a higher quality reconstruction.

Here, 70 iterations were needed to obtained the final result. Despite the high power of recent processors, it still takes few seconds on recent computers. Nevertheless, one can imagine that it could be done in real time with dedicated hardware and used as a part of a video decompression algorithm (see also [22, 88] for interesting video compression methods).



(a) Original degraded color image



(b) Result of the color inpainting algorithm (4.2)

Figure 4.3: Color image inpainting.



(a) Original degraded color image

(b) Result of the color inpainting algorithm (4.2)

Figure 4.4: Color image inpainting (detail).

Inpainting methods could be also applied for the restoration of old images and videos that have been badly stored and that contain *scratches and stains*. For reference articles about other specific inpainting formulations, please refer to [26, 38, 40, 41, 44].

4.3 Color image magnification

Image magnification consists in computing a large image from a smaller one, with interpolation methods. Moreover, one wishes that the magnified images seem to be quite realistic. In this sense, the problem is similar to image inpainting, where the inpainting mask is the boolean inverse of a regular grid :

Indeed, let us consider the small color image $\mathbf{I}_{\text{small}} : \omega = [h \times l] \rightarrow \mathbb{R}^3$. Magnifying the image $\mathbf{I}_{\text{small}}$ by a factor k with a PDE-based method is equivalently done by inpainting the image $\mathbf{I} : \Omega = [kh \times kl] \rightarrow \mathbb{R}^3$ by an inpainting mask $M : \Omega \rightarrow \{0, 1\}$ defined by :

$$\forall (x, y) \in \Omega, \quad M(x, y) = \begin{cases} 0 & \text{if } (x \bmod k) \wedge (y \bmod k) \\ 1 & \text{else} \end{cases} \quad \text{and} \quad \mathbf{I}(x, y) = \mathbf{I}_{\text{small}}\left(\frac{x}{k}, \frac{y}{k}\right)$$

where $\mathbf{I}_{\text{small}}(x/y, y/k)$ is computed through a classical interpolation operator. For our experiments, we considered a linear interpolation.

Then, the magnification is done by applying the inpainting PDE (4.2) on the image $\mathbf{I}$. Note that high magnification scales require higher structure tensor smoothing (i.e high variance σ).

We applied our magnification method on two different images :

1. The first one is a synthetic (96×97 pixels) cartoon image, magnified by a factor $\times 3$. It clearly shows how the classical jaggging effect obtained with simple interpolation schemes disappears with the non-linear PDE-based magnification method (Fig.4.7).



Figure 4.5: Color image inpainting, used to remove real objects in photographs.

2. The second one is a detail (242×244 pixels) of a real digital photograph and has been magnified by a factor $\times 4$ (Fig.4.8).

The picture Fig.4.8c is the result of a direct magnification of factor $\times 4$, using the PDE (4.2). The picture Fig.4.8d is the result of a two step magnification of factors $\times 2$, using the PDE (4.2). Actually, both results are quite similar.

Note that like inpainting methods, PDE-based magnification is not able to create new image structures but succeeds in linking the existing ones in a realistic way.

4.4 Flow visualization

As a last example of the application of *unconstrained* diffusion PDE's, we propose to deal with flow visualization. Considering a 2D vector field $\mathcal{F} : \Omega \rightarrow \mathbb{R}^2$, we have several ways to visualize it :



Figure 4.6: Color image inpainting, used to reconstruct partially coded images.

1. **Using vectors :** This is illustrated on Fig.4.9a. It is a basic representation of the flow, where we represent each vector $\mathcal{F}(\mathbf{x})$ with a graphic line. This simple representation has some drawbacks :
 - One has often to subsample the vector field $\mathcal{F}$ in order to represent it. The vectorial representation size is not well adapted for displaying large fields $\mathcal{F}$. For instance, the figure Fig.4.9a is a sub-sampling of an original field $\mathcal{F}$ by a factor 2 (original size 128×128).
 - The perception of thin flow structures is not easy with dense vectorial graphics.
2. **Using anisotropic diffusion PDE's :** These problems are solved with the use of regularization PDE's. Starting from a purely noisy (color) image, the idea is to diffuse it *in flow direction*, in order to make the flow structures appear :

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H}_i) \quad \text{where} \quad \mathbf{D} = \frac{1}{\|\mathcal{F}\|} \mathcal{F}\mathcal{F}^T \quad (4.3)$$

With this equation (4.3), we perform the image intensity diffusion only in the unit vector $\mathcal{F}/\|\mathcal{F}\|$, with a weight $\|\mathcal{F}\|$. It is equivalently written as :

$$\frac{\partial \mathbf{I}}{\partial t} = \|\mathcal{F}\| \mathbf{I}_{\frac{\mathcal{F}}{\|\mathcal{F}\|} \frac{\mathcal{F}}{\|\mathcal{F}\|}} = \|\mathcal{F}\| \frac{\partial^2 \mathbf{I}}{\partial (\frac{\mathcal{F}}{\|\mathcal{F}\|})^2}$$

which is an $1D$ heat equation, oriented by the flow $\mathcal{F}$.

Whereas the evolution time t of the PDE goes by, the thin flow structures are disappearing while the big ones are preserved. Thus, this constructs a *multi-scale* visualization of the considered flow $\mathcal{F}$ (Fig.4.9b and Fig.4.10).

Here, our used regularization equation (4.3) ensures that the smoothing of the pixels is done exactly in the flow direction $\mathcal{F}$. Similar interesting methods have been proposed in [24, 33, 69], using a divergence-based expression to diffuse the random images, as well as a transport term. As described in chapter 3, the use of a diffusion tensor such as $\mathbf{D}$ in (4.3), with a divergence based equation $\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D}\nabla I_i)$ may not be adapted to smooth the noisy image exactly in the specified directions of the flow $\mathcal{F}$. With our regularization PDE (4.3), we ensure that at each image point $\mathbf{x} = (x, y) \in \Omega$, *the smoothing of the image is exactly done in the direction of $\mathcal{F}(x, y)$* . This is important, since using divergence-based expressions would introduce smoothing in false directions.

4.5 Other applications and extensions

In this chapter, we illustrated the high number of possible applications of vector-valued regularization PDE's. We particularly showed how to adapt the diffusion tensors used in the trace-based equation, in order to deal with specific problems. We focused on *color images*, but these accurately designed diffusion PDE's have interest in many other domains : multi-spectral radar imagery, optical flow regularization, etc.

We defined all our regularization schemes in a way that they behave coherently with the underlying vector geometry of the considered images. Note that no constraints on the type of vector data we regularize, are taken into account. For all the presented applications here, this is sufficient since color pixels are a-priori unconstrained data.

But sometimes, we have some a-priori knowledge about the image point features : they can be for instance unit vectors, rotation matrices or diffusion tensors (see the survey in section.2.3). Using directly unconstrained regularization PDE's as we defined above may be not well suited to this particular class of problems.

Now, we will introduce a mathematical framework (chapter 5) that can handle a useful type of (orientation) constraints. This complementary part will permit to deal with constrained schemes that have the same regularization properties as the unconstrained ones, while preserving the inherent form of the image points. We will finally use it to tackle interesting applications, as illustrated in chapter 6.



Je sais pas ce que t'en pense, mais je trouve que toutes ces méthodes EDP, ça donne des résultats qui ont l'air trop synthétique, t'es pas d'accord ?

J. Bride.

(a) Color image at original size (96×97)

(b) Bloc magnification



(c) Linear magnification



(d) PDE-based magnification

Figure 4.7: Magnification of a color cartoon image ($\times 3$).

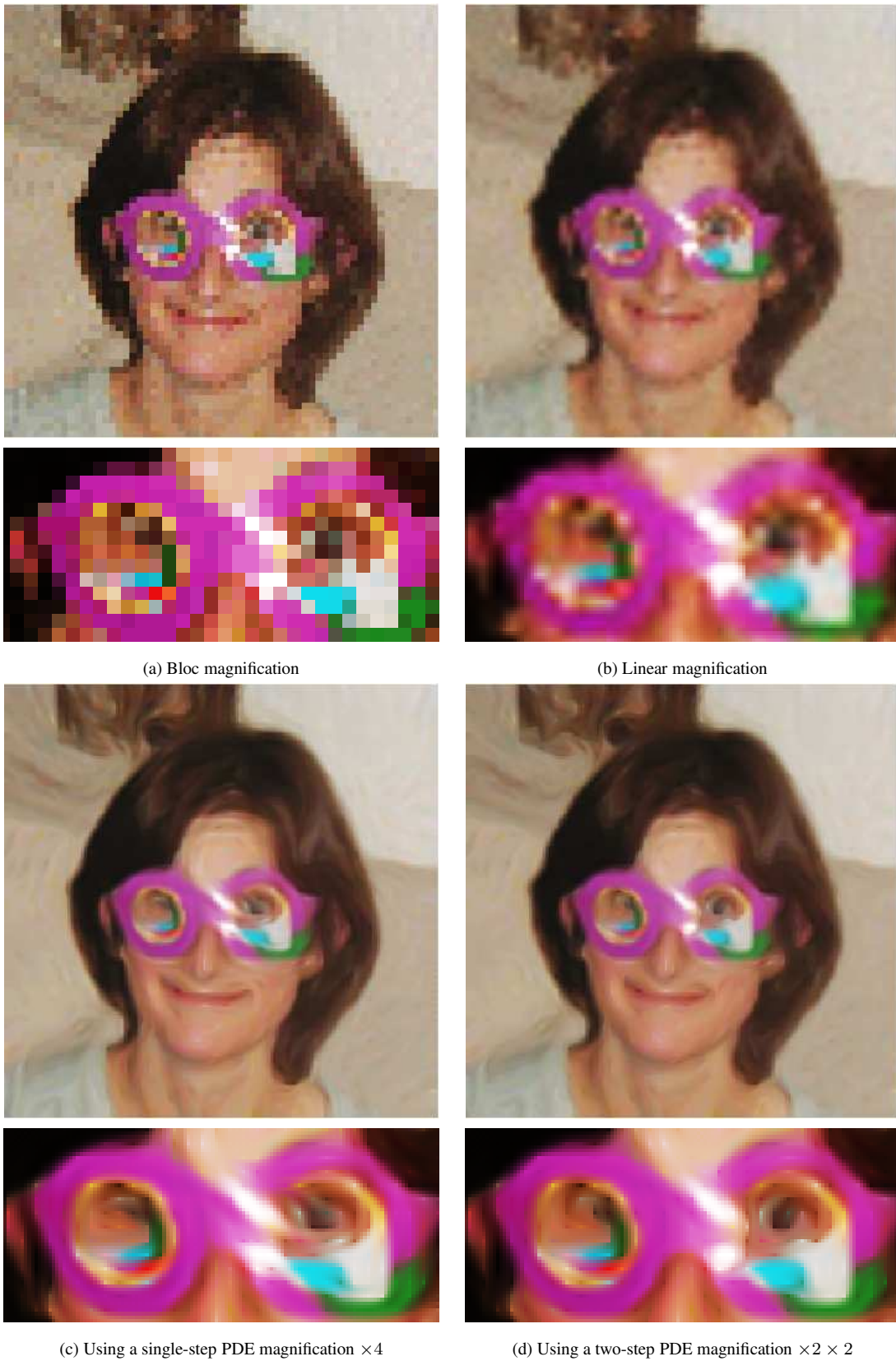
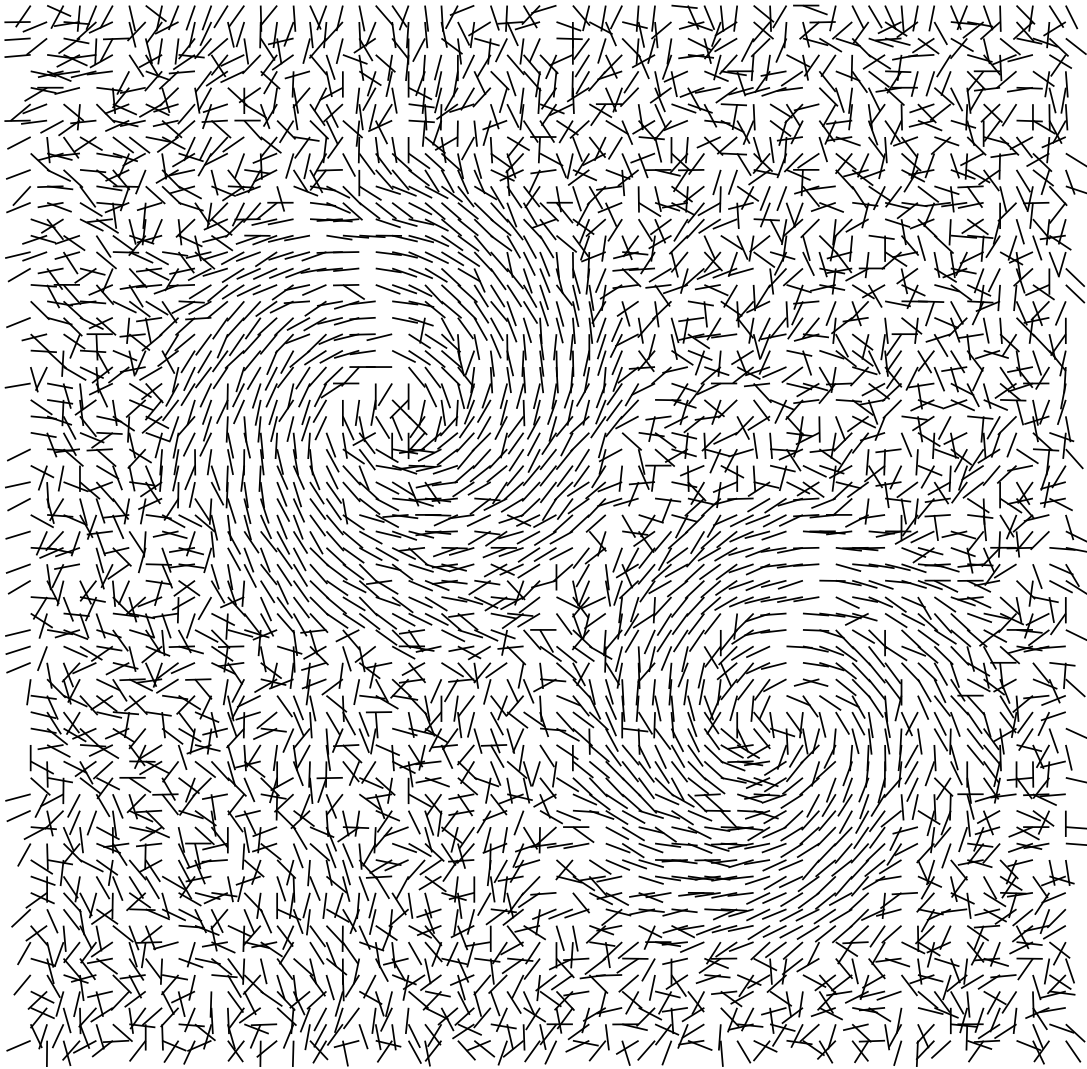
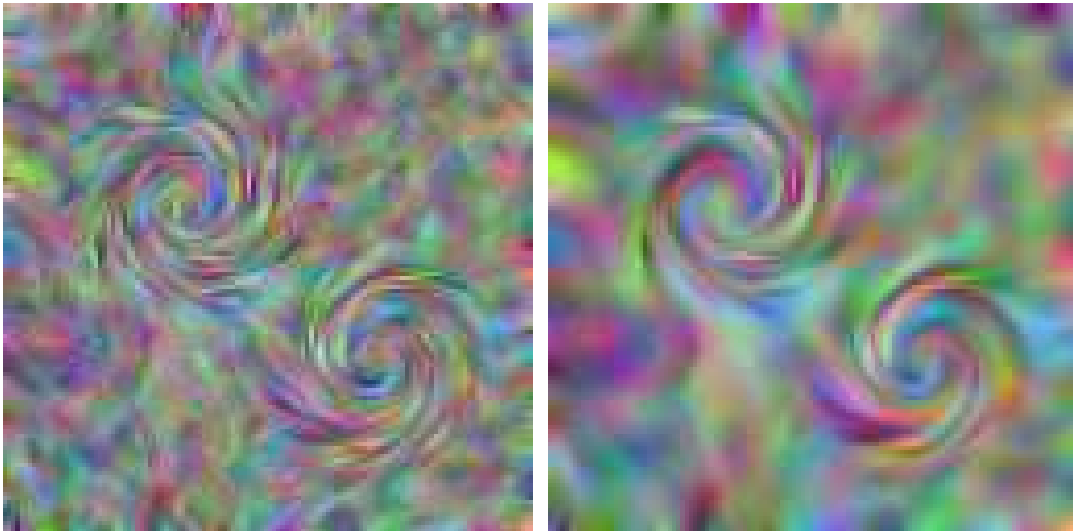


Figure 4.8: Magnification of a real color photograph ($\times 4$).



(a) Original flow, represented with graphic vectors



(b) Visualization using regularization PDE (after respectively 5 and 10 iterations)

Figure 4.9: Flow visualization.

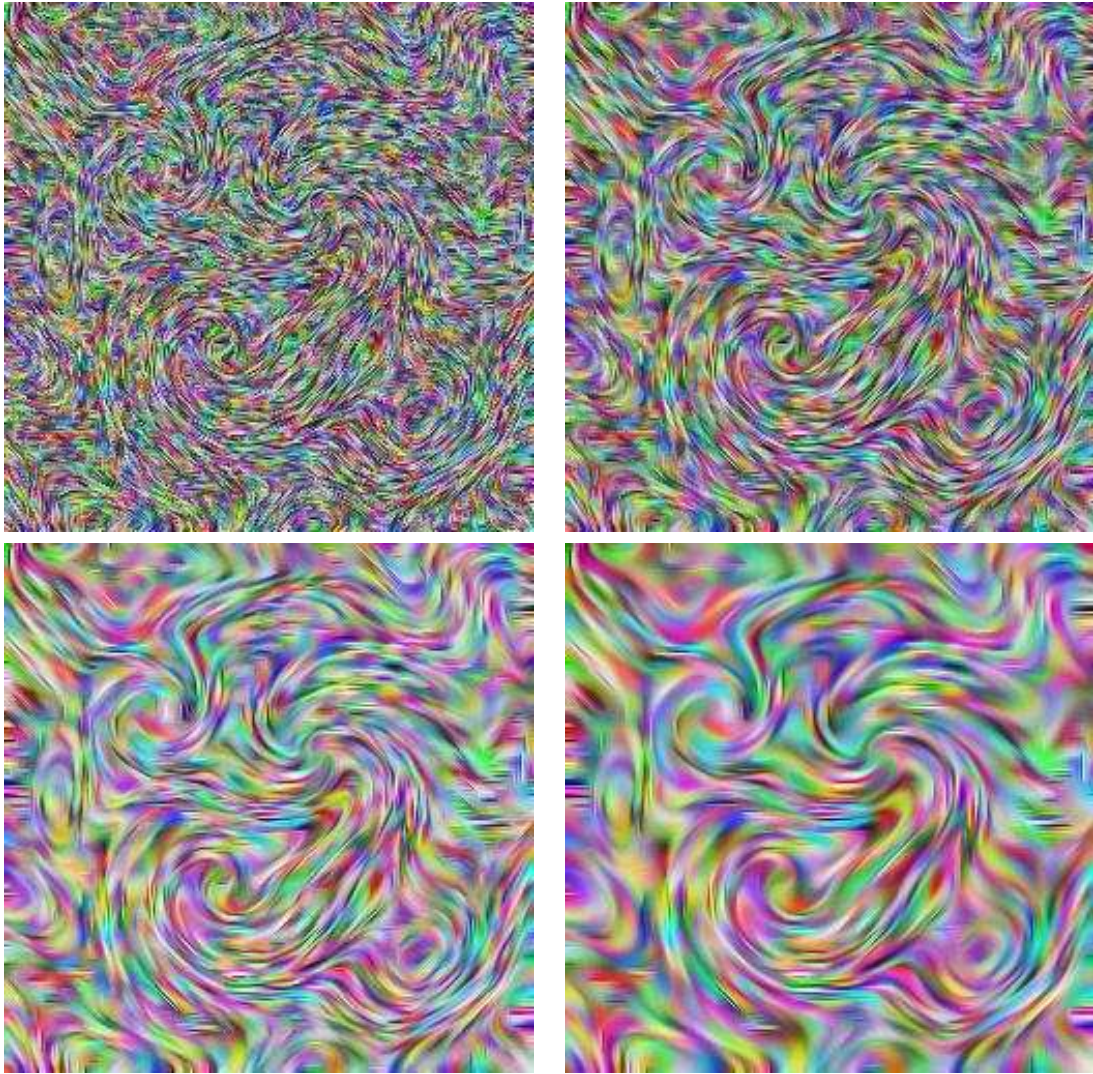


Figure 4.10: Scale-space visualization of a flow.

Chapter 5

Constrained PDE's and the Framework of Orthonormal Vector Sets



In this chapter, we propose to take into account some prior constraints on the data we regularize. More particularly, we are interested in regularizing fields of orthonormal vector sets, where each point is defined as multiple orthogonal and unit vectors. It can indeed represent a lot of interesting orientation features. The derivation of a specific ψ -functional framework leads to a multivalued regularization PDE preserving these orthonormal constraints and extending naturally the recent works on direction diffusion. Then, we study some particular cases of direction features that can be handled by this new formalism : unit vectors, orthogonal matrices and diffusion tensor orientations. Specific numerical schemes are finally proposed, allowing to implement all these constrained PDE's and proposing an elegant solution to the classical reprojection problem.

5.1 Interest of orthonormal vector sets

In the previous parts of this document, we studied anisotropic diffusion PDE's acting on *unconstrained* multi-valued data, i.e whose vector components are not linked together. Such methods have proven their efficiencies for regularization of a large variety of images, including color pictures, optical flows, etc.

Some of these diffusion PDE's were recently extended within various frameworks, in order to deal with fields of *unit vectors*, taking the unitary norm constraints into account.

$$\forall \mathbf{x} \in \Omega, \quad \|\mathbf{I}(\mathbf{x})\| = 1$$

This is an important new domain that has been addressed for the regularization of direction fields and color images expressed in the chromaticity-brightness or HSV spaces. Proposed methods were based on the decomposition into polar angles [142], the harmonic maps theory [167, 168], the total variation framework [39] or the geometric Beltrami operator [106, 107] (see section 2.3 for a state of the art with more references).

The derivation of the corresponding norm constrained regularizing flows is a challenging problem, which is quite different than the unconstrained case, in the sense that the data features are known to be constrained to the specific *unit sphere* manifold $S^{n-1} \subset \mathbb{R}^n$. It usually leads to sets of vector PDE's where the preservation of the norm constraint (that involves a link between vector components I_i) appears as an *additive coupling term*.

In this chapter, we go one step further and study the constrained regularization of more complex orientation features modeled by *orthonormal vector sets*, and its application on fields of unit vectors, rotation matrices and diffusion tensors.

We first develop a variational approach of the regularization, and detail the derivation of the corresponding constrained equations, thanks to the use of Lagrange multipliers. Then we study each interesting particular case of this formalism. Finally, a physical interpretation of the proposed constrained regularization process will be used to design accurate numerical schemes, avoiding the classical reprojection step problem.

5.1.1 Preliminary notations

Let us consider m vector-valued images

$$\mathbf{I}^{[k]} : \Omega \rightarrow \mathbb{R}^n \quad (1 \leq k \leq m, \quad n \in \mathbb{N}^+ \quad \text{and} \quad \Omega \subset \mathbb{R}^p)$$

As in previous chapters, we use the notation $I_i^{[k]}$ to designate the i^{th} channel of the multivalued image $\mathbf{I}^{[k]}$:

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}^{[k]}(\mathbf{x}) = \begin{pmatrix} I_1^{[k]}(\mathbf{x}) \\ I_2^{[k]}(\mathbf{x}) \\ \vdots \\ I_n^{[k]}(\mathbf{x}) \end{pmatrix}$$

We are particularly interested in the set

$$\mathcal{B} = \left\{ \mathbf{I}^{[k]} \mid 1 \leq k \leq m \right\}$$

of the m vector-valued images $\mathbf{I}^{[k]}$. It can be seen itself as a field, where each point is a *vector set* :

$$\forall \mathbf{x} \in \Omega, \quad \mathcal{B}(\mathbf{x}) = \left\{ \mathbf{I}^{[1]}(\mathbf{x}), \mathbf{I}^{[2]}(\mathbf{x}), \dots, \mathbf{I}^{[m]}(\mathbf{x}) \right\}$$

Suppose now that the following *orthonormal constraints* between the vectors $\mathbf{I}^{[k]}$ are also verified :

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}^{[k]}(\mathbf{x}) \cdot \mathbf{I}^{[l]}(\mathbf{x}) = \delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases} \quad (5.1)$$

where $\mathbf{I}^{[k]}(\mathbf{x}) \cdot \mathbf{I}^{[l]}(\mathbf{x}) = \sum_{i=1}^n I_i^{[k]}(\mathbf{x}) I_i^{[l]}(\mathbf{x})$ is the usual dot product in $\mathbb{R}^n$.

Then, $\forall \mathbf{x} \in \Omega$, $\mathcal{B}(\mathbf{x})$ is an *orthonormal vector set* composed of m orthogonal and unit vectors of dimension n .

Note that a particularly interesting case is reached when $m = n$, since $\mathcal{B}$ is then an *orthonormal vector basis* in $\mathbb{R}^n$.

5.1.2 Orthonormal vector sets and direction features

In the following, we will propose a general approach to regularize any datasets that can be expressed as a field $\mathcal{B}$ of orthonormal vector sets, using coupled anisotropic diffusion PDE's. This idea is motivated by the fact that orthonormal vector sets can indeed represent various *direction features*, including :

- **Unitary vectors** : When the set $\mathcal{B}$ is restricted to a single vector image $\mathcal{B} = \{\mathbf{I}\}$ (i.e $m = 1$), the orthonormal constraints (5.1) reduce to

$$\forall \mathbf{x} \in \Omega, \quad \|\mathbf{I}(\mathbf{x})\|^2 = 1$$

which is the unitary norm constraint characterizing fields of direction vectors (Fig.5.1a).

- **Orthogonal matrices** : The columns of an orthogonal matrix $\mathbf{R} \in \mathcal{O}(n)$ are unit vectors that form an orthonormal vector basis. Then, $\mathbf{R}$ can be equivalently represented by an orthonormal vector set $\mathcal{B}$ with $m = n$ (the matrix dimension), as illustrated on Fig.5.1b.

Roughly speaking, we can see $\mathcal{O}(n)$ as a kind of bi-dimensional extension of S^{n-1} . Both manifolds have the same non-flat structure, due to orthonormal constraints.

More particularly, we will use such orthonormal vector bases to deal with rotation matrices $\mathbf{R} \in \text{SO}(n) \subset \mathcal{O}(n)$.

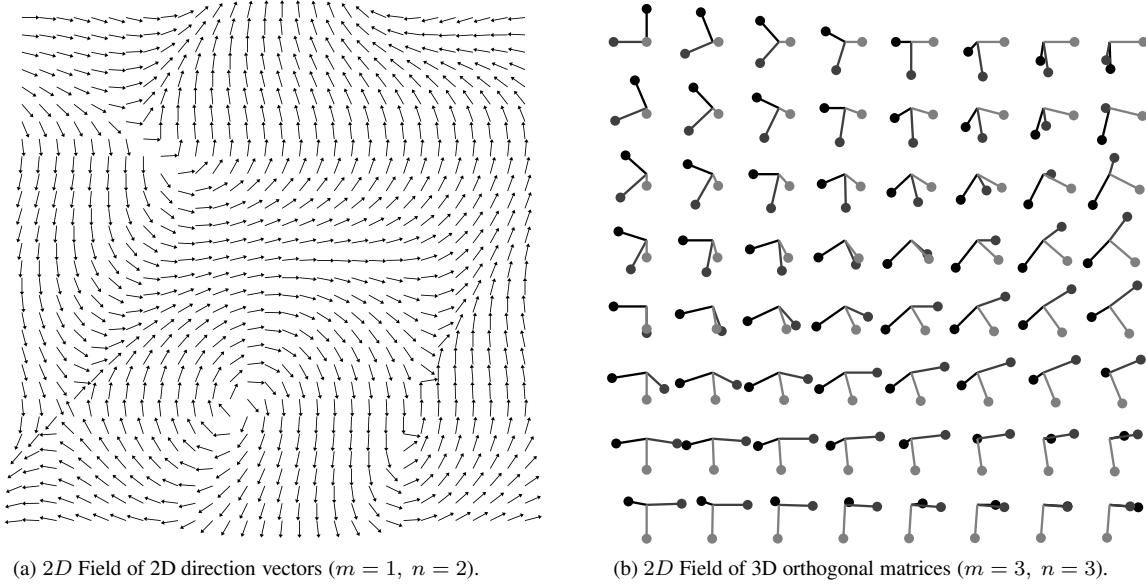


Figure 5.1: Fields of orthonormal vector sets.

These particular cases will be detailed further, respectively in sections 5.3.1 and 5.3.2. First of all, let us develop a variational formulation that addresses the regularization of *general orthonormal vector sets* ($m, n \in \mathbb{N}^+$).

5.2 Regularizing fields of orthonormal vector sets

We consider an initial “noisy” image $\mathcal{B}_0$ of *orthonormal vector sets*, verifying the constraints (5.1).

$$\forall \mathbf{x} \in \Omega, \quad \mathcal{B}_0(\mathbf{x}) = \left\{ \mathbf{I}_0^{[1]}(\mathbf{x}), \mathbf{I}_0^{[2]}(\mathbf{x}), \dots, \mathbf{I}_0^{[m]}(\mathbf{x}) \right\}$$

The idea is to regularize $\mathcal{B}_0$, using a variational flow that preserves the orthonormal structure of the vector sets (Fig.5.2).

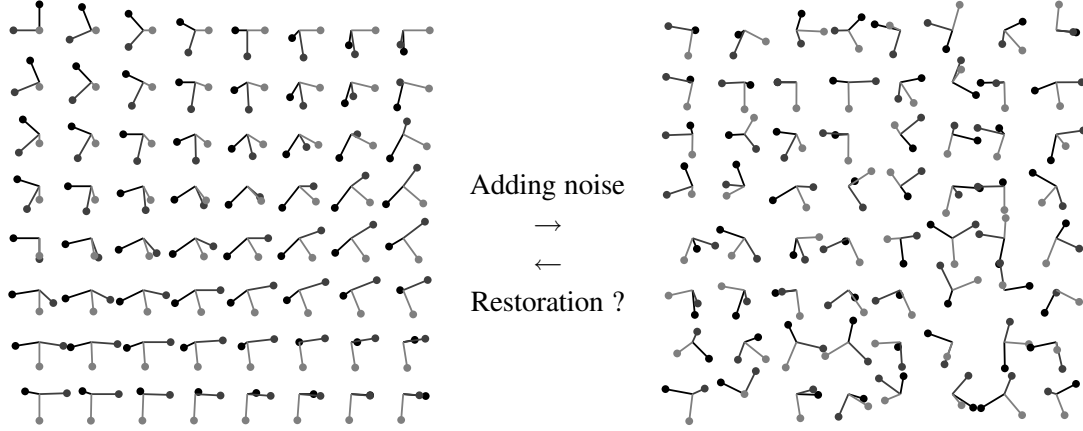


Figure 5.2: How to regularize a field $\mathcal{B}$ of orthonormal vector sets ?

5.2.1 Unconstrained regularization

We propose to find $\mathcal{B}$ as the solution of an energy minimization, following our idea of ψ -function diffusion, used to restore unconstrained multivalued images (see chapter 3). We quickly remind the idea.

Each noisy vector-valued image $\mathbf{I}_0^{[k]}$ of the set $\mathcal{B}_0$ can be anisotropically restored (denoising with preservation of discontinuities), by minimizing the following ψ -functional :

$$\min_{\mathbf{I}^{[k]}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}^{[k]}) = \int_{\Omega} \left(\frac{\alpha}{2} \|\mathbf{I}^{[k]} - \mathbf{I}_0^{[k]}\|^2 + \psi(\lambda_+^{[k]}, \lambda_-^{[k]}) \right) d\Omega \quad (5.2)$$

where $\lambda_+^{[k]}, \lambda_-^{[k]}$ are the eigenvalues of the structure tensor $\mathbf{G}^{[k]} = \sum_{j=1}^n \nabla I_j^{[k]} \nabla I_j^{[k]T}$, and measure the local vector variations both in norms and orientations, in the multivalued image $\mathbf{I}^{[k]}$.

Contrary to chapters 2 and 3, we are not only interested by the behavior of a regularization term, but we would like to find a *steady-state* regular solution which is not constant. This is the reason why we also consider a *data attachment term* through the fixed parameter $\alpha \in \mathbb{R}$, in the functional (5.2). It prevents the final solution from being too different from the initial given field $\mathbf{I}_0^{[k]}$.

The function $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a *diffusion function*, which controls the regularization behavior. It is a natural extension of the ϕ -functional framework, already proposed in the literature related to scalar image restoration. Choosing the right ψ -function depends on the desired regularization

behavior and we refer the reader to chapters 2 and 3 for more informations about ψ -functionals.

One way of minimizing the functional $E(\mathbf{I}^{[k]})$, is to compute the corresponding *vector Lagrangian* $\mathcal{L}^{[k]} \in \mathbb{R}^n$ which is in this case (using a component by component writing style) :

$$\mathcal{L}_i^{[k]} = \alpha (I_i^{[k]} - I_{i_0}^{[k]}) - \text{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+^{[k]}} \theta_+^{[k]} \theta_+^{[k]T} + \frac{\partial \psi}{\partial \lambda_-^{[k]}} \theta_-^{[k]} \theta_-^{[k]T} \right] \nabla I_i^{[k]} \right)$$

where the $\theta_{\pm}^{[k]}$ are the eigenvectors of the structure tensor $\mathbf{G}^{[k]}$.

Then, one uses m vector gradient descents until steady state : $\frac{\partial \mathbf{I}^{[k]}}{\partial t} = -\mathcal{L}^{[k]}$, i.e the $m \times n$ following PDE's :

$$\begin{cases} \mathbf{I}_{(t=0)}^{[k]} = \mathbf{I}_0^{[k]} \\ \frac{\partial I_i^{[k]}}{\partial t} = \alpha (I_{i_0}^{[k]} - I_i^{[k]}) + \text{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+^{[k]}} \theta_+^{[k]} \theta_+^{[k]T} + \frac{\partial \psi}{\partial \lambda_-^{[k]}} \theta_-^{[k]} \theta_-^{[k]T} \right] \nabla I_i^{[k]} \right) \end{cases} \quad (5.3)$$

($1 \leq i \leq n$ and $1 \leq k \leq m$).

For our purpose of orthonormal vector set regularization, one could naively apply such diffusion PDE's (5.3) on each vector $\mathbf{I}_0^{[k]}$ of the orthonormal vector set $\mathcal{B}_0$, then reconstruct the regularized vector set image $\mathcal{B}$ with the resulting smoothed vectors.

A result of this method is illustrated on Fig.5.3. Two regularizing PDE's (5.3) were applied on each component of a 2D orthonormal vector base field $\mathcal{B} = \{\mathbf{I}^{[1]}, \mathbf{I}^{[2]}\}$, which has been chosen to be a mixture of direct and indirect bases. For illustration purposes, we chose a simple Tikhonov-like ψ -function $\psi(\lambda_+, \lambda_-) = \lambda_+ + \lambda_-$, leading to an isotropic smoothing.

Unfortunately, this decoupled regularization method breaks the orthonormal properties : vector norms and orthogonal angles are not intrinsically preserved by an unconstrained regularization PDE as (5.3). Thus, we have to explicitly introduce *orthonormal constraints*, in the minimization process.

Note that conversely to regularization methods acting on S^{n-1} that led to PDE's with coupling terms between vector components, our problem is more general since the equations must also consider an *additional orthogonal coupling between the different vectors themselves*.

5.2.2 A way of preserving the orthonormal constraints

In order to regularize the field of orthonormal vector set $\mathcal{B}_0$ while preserving the orthonormal properties (5.1), we propose a constrained minimization of the following functional :

$$\min_{\mathcal{B}} E(\mathcal{B}) = \sum_{k=1}^m E(\mathbf{I}^{[k]})$$

which can also be written :

$$E(\mathcal{B}) = \int_{\Omega} \sum_{k=1}^m \left(\frac{\alpha}{2} \|\mathbf{I}^{[k]} - \mathbf{I}_0^{[k]}\|^2 + \psi(\lambda_+^{[k]}, \lambda_-^{[k]}) \right) d\Omega \quad (5.4)$$

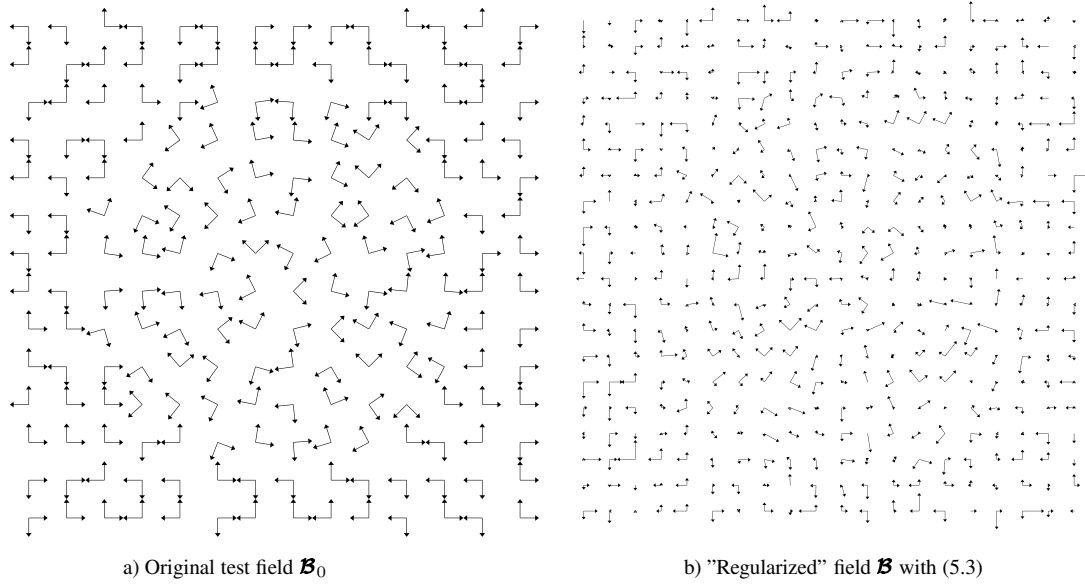


Figure 5.3: Decoupled regularization of orthonormal vector sets.

with respect to the m vector images $\mathbf{I}^{[k]}$, subject to the orthonormal constraints :

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}^{[p]}(\mathbf{x}) \cdot \mathbf{I}^{[q]}(\mathbf{x}) = \delta_{pq} = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases} \quad (5.1)$$

Note that the m Lagrangian vectors $\mathcal{L}^{[k]}$ of the energy $E(\mathcal{B})$ are obviously the same as in section 5.2.1, i.e

$$\mathcal{L}_i^{[k]} = \alpha (I_i^{[k]} - I_{i_0}^{[k]}) - \text{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+^{[k]}} \theta_+^{[k]} \theta_+^{[k]T} + \frac{\partial \psi}{\partial \lambda_-^{[k]}} \theta_-^{[k]} \theta_-^{[k]T} \right] \nabla I_i^{[k]} \right)$$

It is then equivalent to associate at each vector $\mathbf{I}^{[k]}$ an energy functional $E(\mathbf{I}^{[k]})$ as defined in (5.2). The orthonormal constraints are then introduced by adding m^2 Lagrange multipliers $\lambda_{pq} : \Omega \rightarrow \mathbb{R}$ (where $p, q \in [1, m]$) to the functional $E(\mathcal{B})$, where each λ_{pq} is associated with the constraint :

$$\forall \mathbf{x} \in \Omega, \quad \mathbf{I}^{[p]}(\mathbf{x}) \cdot \mathbf{I}^{[q]}(\mathbf{x}) = \delta_{pq}$$

It leads to the *unconstrained minimization* of the following functional, with respect to the $\mathbf{I}^{[k]}$ and λ_{pq} :

$$E^*(\mathcal{B}, \lambda) = E(\mathcal{B}) + \int_{\Omega} \sum_{(p,q) \in [1,m]} \lambda_{pq} (\mathbf{I}^{[p]} \cdot \mathbf{I}^{[q]} - \delta_{pq}) d\Omega$$

In fact, as the dot product and the δ_{pq} are symmetric, the constraints $\mathbf{I}^{[p]} \cdot \mathbf{I}^{[q]} = \delta_{pq}$ and $\mathbf{I}^{[q]} \cdot \mathbf{I}^{[p]} = \delta_{qp}$ are the same, and the two corresponding Lagrange multipliers λ_{pq} and λ_{qp} are then equal.

When the constrained minimum is reached, the Euler-Lagrange equations corresponding to $E^*(\mathcal{B}, \lambda)$ with respect to $\mathbf{I}^{[k]}$ are : $\forall k \in [1, m]$,

$$\begin{aligned} 0 &= \mathcal{L}^{[k]} + \sum_{(p,q)} \lambda_{pq} \frac{\partial \mathbf{I}^{[p]}}{\partial \mathbf{I}^{[k]}} \cdot \mathbf{I}^{[q]} + \sum_{(p,q)} \lambda_{qp} \frac{\partial \mathbf{I}^{[q]}}{\partial \mathbf{I}^{[k]}} \cdot \mathbf{I}^{[p]} \\ &= \mathcal{L}^{[k]} + \sum_{q=1}^m \lambda_{kq} \mathbf{I}^{[q]} + \sum_{p=1}^m \lambda_{pk} \mathbf{I}^{[p]} \\ &= \mathcal{L}^{[k]} + 2 \sum_{l=1}^m \lambda_{kl} \mathbf{I}^{[l]} \end{aligned}$$

and the final set of Euler-Lagrange equations of $E^*(\mathcal{B}, \lambda)$ with respect to $\mathbf{I}^{[k]}$ and λ_{pq} is written :

$$\begin{cases} \mathcal{L}^{[k]} + 2 \sum_{l=1}^m \lambda_{kl} \mathbf{I}^{[l]} = 0 & (a) \\ \mathbf{I}^{[p]} \cdot \mathbf{I}^{[q]} = \delta_{pq} & (k, p, q \in [1, m]) \quad (b) \end{cases} \quad (5.5)$$

Thanks to the quadratic form of the orthonormal constraints, we can find formally the λ_{kl} reached at the minimum : we take the dot product of the l^{th} equation of (5.5a) with the vector $\mathbf{I}^{[k]}$:

$$\mathcal{L}^{[l]} \cdot \mathbf{I}^{[k]} + 2 \sum_{p=1}^m \lambda_{pl} \mathbf{I}^{[p]} \cdot \mathbf{I}^{[k]} = 0$$

then simplify it using the orthonormal relations (5.5b) :

$$\lambda_{kl} = -\frac{\mathcal{L}^{[l]} \cdot \mathbf{I}^{[k]}}{2}$$

Finally, replacing the λ_{kl} in (5.5a) gives a closed form of the vector gradient descent that minimizes (5.4) while *preserving the orthonormal constraints* (5.1) :

$$\frac{\partial \mathbf{I}^{[k]}}{\partial t} = -\mathcal{L}^{[k]} + \sum_{l=1}^m \left(\mathcal{L}^{[l]} \cdot \mathbf{I}^{[k]} \right) \mathbf{I}^{[l]} \quad (5.6)$$

where

$$\mathcal{L}_i^{[k]} = \alpha (I_i^{[k]} - I_{i_0}^{[k]}) - \text{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+^{[k]}} \theta_+^{[k]} \theta_+^{[k]T} + \frac{\partial \psi}{\partial \lambda_-^{[k]}} \theta_-^{[k]} \theta_-^{[k]T} \right] \nabla I_i^{[k]} \right) \quad (5.7)$$

(with $i = 1..n$).

□

The obtained equation (5.6) is a set of m coupled vector PDE's, i.e $m \times n$ scalar regularization equations where the coupling between vectors *and* vector components is clearly present. It allows to regularize any field of orthonormal vector sets, preserving the orthonormal structure of the vectors during the PDE evolution.

Note : The k^{th} Lagrangian vector $\mathcal{L}^{[k]}$ of the unconstrained functional $E(\mathcal{B})$, defined as (5.7) can be seen as a pure *diffusion force*, acting on the vector $\mathbf{I}^{[k]}$. A physical interpretation of these single forces will be provided in section 5.3.3.

Actually, one can note the clear separation in the PDE (5.6) between *the unconstrained Lagrangians* $\mathcal{L}^{[k]}$ which are responsible for the regularization behavior, and *the coupling term* $\sum_{l=1}^m \left(\mathcal{L}^{[l]} \cdot \mathbf{I}^{[k]} \right) \mathbf{I}^{[l]}$ which allows the *orthonormal constraints* to be *preserved*.

This opens interesting possibilities.

We may for instance replace the ψ -function based Lagrangian term (5.7) by other regularization terms more adapted to certain problems, even if it doesn't come from variational principles. For instance, we can use our trace-based equation proposed in chapter 3, but also classical multivalued regularization as those described in chapter 2. One can also think to use this general equation (5.6) to solve other problems involving orthonormal constraints, such as image matching or edge enhancement.

The figure Fig.5.4 illustrates the role of the orthonormal constraint preservation during the PDE flow (5.6), as well as its anisotropic regularizing behavior. We applied the PDE (5.6) on two different kind of orthonormal vector sets :

- **a 2D field of 2D orthonormal bases** (Fig.5.4a,b,c) : The image is initially clear, but we illustrate how the unconstrained anisotropic regularization change the image anyway. As expected, the unconstrained regularization equation reduces the vector norm and doesn't preserve the orthogonal angles. Using an orthonormal-constrained PDE (5.6) allows to retrieve an orthonormal-preserved result.
- **a 2D field of 3D orthonormal bases** (Fig.5.4d,e,f) : The field illustrated here contains a noisy *triple-junction*, which is a structure that is usually quite hard to restore. Anyway, the use of an *anisotropic* regularization term (5.7) allows to retrieve these discontinuities very well, while preserving the orthonormal constraints.

Note that this kind of field may interestingly represent a field of *rotation matrices*, as it will be discussed in detail in section 5.3.2.

From now on, we will study some particular cases of orthonormal vector sets, and the corresponding constrained regularization equations (5.6). We will also link our approach with related works previously proposed in the literature.

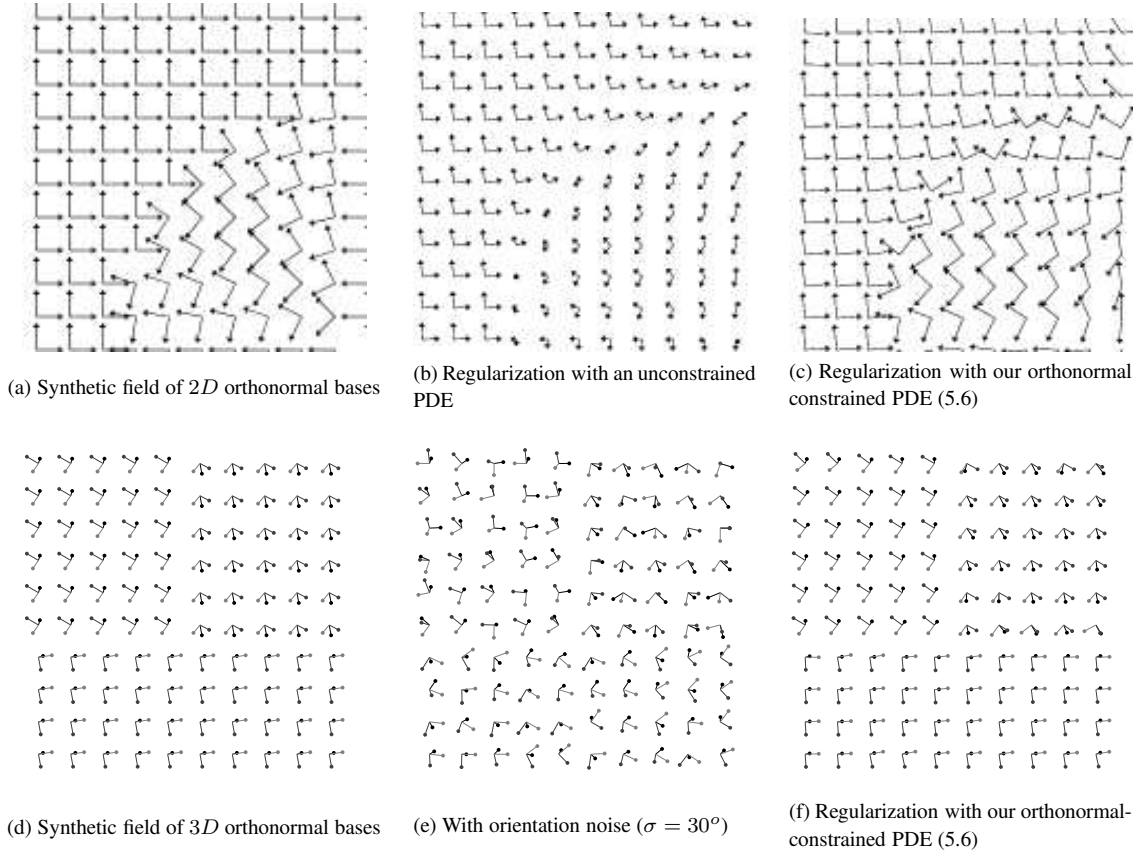


Figure 5.4: Anisotropy and constraint-preserving property of the orthonormal-constrained regularization.

5.3 Regularization of Directions and Rotations

5.3.1 Unit vector regularization

Vector direction diffusion has already been studied in [27, 39, 142, 168, 174, 182]. It consists in regularizing fields $\mathbf{I}$ of unit vectors $\mathbf{I}(x) \in S^{n-1}$. Actually, this problem can be seen as a particular case of our orthonormal vector set framework, where the vector sets $\mathcal{B}(\mathbf{x})$ are restricted to a single vector $\mathcal{B}(\mathbf{x}) = \{\mathbf{I}(\mathbf{x})\}$.

Considering the general orthonormal vector set evolution (5.6) and reducing it to the norm constraint, we find the following unit norm regularization PDE flow :

$$\frac{\partial \mathbf{I}}{\partial t} = -\mathcal{L} + (\mathcal{L} \cdot \mathbf{I}) \mathbf{I} \quad (5.8)$$

where

$$\mathcal{L}_i^{[k]} = \alpha (I_i^{[k]} - I_{i_0}^{[k]}) - \text{div} \left(\left[\frac{\partial \psi}{\partial \lambda_+^{[k]}} \theta_+^{[k]} \theta_+^{[k]T} + \frac{\partial \psi}{\partial \lambda_-^{[k]}} \theta_-^{[k]} \theta_-^{[k]T} \right] \nabla I_i^{[k]} \right) \quad (i = 1..n)$$

□

This PDE (5.8) is an interesting particular case of our proposed variational framework based on the preservation of orthonormal vector sets constraints. It can be used to regularize any field of direction vectors with unit norms, with a very generic anisotropic term $\mathcal{L}$.

• Geometric interpretation :

Starting from an initial image of direction vectors $\mathbf{I}_0$, we want that each unit norm is preserved during the PDE evolution, i.e :

$$\forall \mathbf{x} \in \Omega, \forall t, \quad \|\mathbf{I}(\mathbf{x})\|^2 = 1 \quad \implies \quad 2 \mathbf{I}(\mathbf{x}) \cdot \frac{\partial \mathbf{I}(\mathbf{x})}{\partial t} = 0$$

It means that the PDE *velocity vector* $\frac{\partial \mathbf{I}(\mathbf{x})}{\partial t}$ must be always orthogonal to the vector $\mathbf{I}(\mathbf{x})$, in order to preserve its norm, i.e that $\mathbf{I}(\mathbf{x})$ performs a pure *rotation*. Suppose then we have a general unconstrained vector PDE, that doesn't preserve the unit norms of the $\mathbf{I}(\mathbf{x})$:

$$\frac{\partial \mathbf{I}}{\partial t} = \mathbf{v} \quad \text{where } \mathbf{v} \in \mathbb{R}^n$$

Adding the norm constraint can be naturally done by *projecting the velocity* $\mathbf{v}$ to the hyperplane, orthogonal to $\mathbf{I}$, which is formally :

$$\begin{aligned} \mathcal{P}_{\mathbf{I}}^\perp(\mathbf{v}) &= (\mathbf{Id} - \mathbf{I} \mathbf{I}^T) \mathbf{v} \\ &= \mathbf{v} - (\mathbf{v} \cdot \mathbf{I}) \mathbf{I} \end{aligned}$$

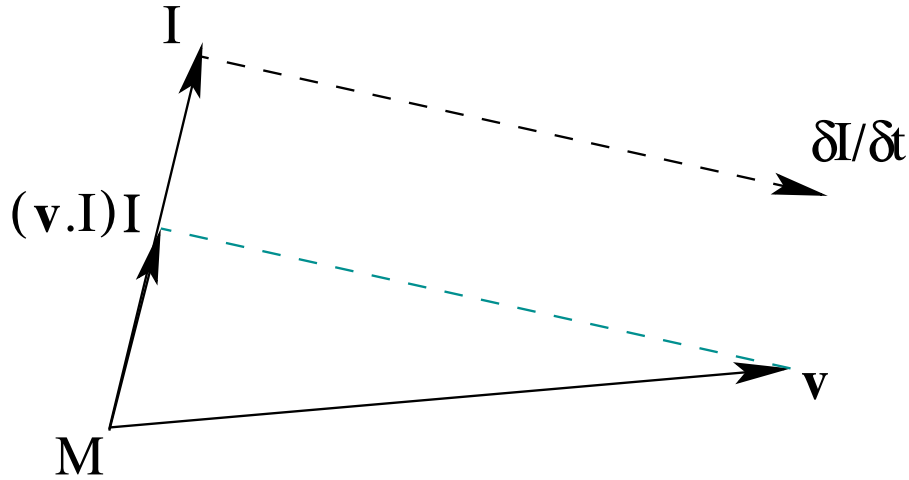


Figure 5.5: The geometric intuition behind the norm constraint.

This new velocity is obviously orthogonal to the vector $\mathbf{I}$ and then, the following PDE ensures the preservation of the unit norm during the flow :

$$\frac{\partial \mathbf{I}}{\partial t} = \mathbf{v} - (\mathbf{v} \cdot \mathbf{I}) \mathbf{I} \tag{5.9}$$

This local geometric viewpoint is another way to find exactly the norm-constrained flow (5.8), coming from the orthonormal vector set formalism. The figure Fig.5.5 illustrates this simple geometric intuition : during the PDE evolution, $\mathbf{I}(\mathbf{x})$ does a *pure rotation* and preserves then its norm.

• The ϕ -function case :

A link with the previous works on direction diffusion can be made, if we consider the particular case of the ϕ -functional framework, i.e the following unconstrained Lagrangian term $\mathcal{L}$:

$$\psi(\lambda_+, \lambda_-) = \phi(\sqrt{\lambda_+ + \lambda_-}) \implies \mathcal{L}_i = \operatorname{div} \left(\frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|} \nabla I_i \right) \quad (i = 1..n)$$

From the spatial derivations of $\|\mathbf{I}(\mathbf{x})\|^2 = 1$, we find :

$$\forall a \in [1, p], \quad \mathbf{I} \cdot \frac{\partial \mathbf{I}}{\partial x_a} = 0 \quad \text{and} \quad \Delta \mathbf{I} \cdot \mathbf{I} = -\|\nabla \mathbf{I}\|^2 \quad (5.10)$$

Developing the divergence in each $\mathcal{L}_i$:

$$\operatorname{div} (A \nabla I_i) = A \Delta I_i + \nabla A \cdot \nabla I_i$$

where $A = \frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|}$. If we note by $\mathbf{d}$ the vector defined by $d_i = \operatorname{div} (A \nabla I_i)$, with $i = 1..n$:

$$\begin{aligned} \mathbf{d} \cdot \mathbf{I} &= A \Delta \mathbf{I} \cdot \mathbf{I} + \sum_{i=1}^n \sum_{a=1}^p \frac{\partial A}{\partial x_a} \frac{\partial I_i}{\partial x_a} I_i \\ &= A \Delta \mathbf{I} \cdot \mathbf{I} + \sum_{a=1}^p \frac{\partial A}{\partial x_a} \frac{\partial \mathbf{I}}{\partial x_a} \cdot \mathbf{I} \end{aligned}$$

The equations (5.10) allow the simplification :

$$\mathbf{d} \cdot \mathbf{I} = -\phi'(\|\nabla \mathbf{I}\|) \|\nabla \mathbf{I}\|$$

Then, the diffusion PDE (5.8) becomes in the particular case of the ϕ -functionals :

$$\begin{aligned} \frac{\partial I_i}{\partial t} &= \operatorname{div} \left(\frac{\phi'(\|\nabla \mathbf{I}\|)}{\|\nabla \mathbf{I}\|} \nabla I_i \right) + \phi'(\|\nabla \mathbf{I}\|) \|\nabla \mathbf{I}\| I_i \\ &+ \alpha (I_{i_0} - (\mathbf{I}_0 \cdot \mathbf{I}) I_i) \end{aligned} \quad (5.11)$$

□

Direction diffusion PDE's already proposed in the literature [27, 39, 168] are a restriction of (5.11) to

$$\alpha = 0 \quad \text{and} \quad \phi(s) = s^r \quad (r = 1, 2)$$

Actually, our proposed method (5.8) is more general since we can choose a more complex unconstrained Lagrangian vector $\mathcal{L}$, adapted to the regularization problem of the problems we consider.

5.3.2 Orthogonal 3×3 matrices

We are now interested in another particular case of orthonormal vector sets : 3D orthonormal vector bases. We consider then orthonormal vector bases fields $\mathcal{B}$ with $m = n = 3$ and for simplicity reasons, we denote the three basis vectors by :

$$\mathbf{I} = \mathbf{I}^{[1]}, \quad \mathbf{J} = \mathbf{I}^{[2]} \quad \text{and} \quad \mathbf{K} = \mathbf{I}^{[3]} \quad \text{then} \quad \mathcal{B} = \{ \mathbf{I}, \mathbf{J}, \mathbf{K} \}$$

Such datasets can represent fields of orthogonal matrices $\mathbf{R} \in O(3)$, since the columns of such matrices form an orthonormal vector basis $\mathcal{B}$.

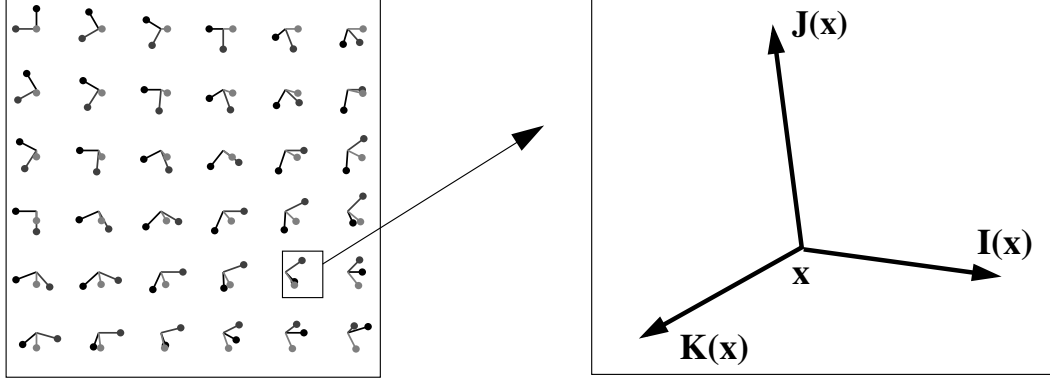


Figure 5.6: Fields of 3D orthonormal vector bases.

Note that the sign of the determinant $\det(\mathbf{R})$ is an information that can be retrieved from the configuration of the orthonormal vector basis $\mathcal{B}$, which is *direct* (then $\det(\mathbf{R}) = +1$) or *indirect* (then $\det(\mathbf{R}) = -1$). More particularly, it means that we are able to discriminate rotations ($\det(\mathbf{R}) = +1$ and $\mathbf{R} \in SO(n)$), and rotoinversions ($\det(\mathbf{R}) = -1$ and $\mathbf{R} \in O(n) \setminus SO(n)$) when modeling orthogonal matrices with orthonormal vector sets $\mathcal{B}$.

In order to regularize $\mathcal{B}$ while preserving possible discontinuities, we minimize the functional (5.4), with $m = n = 3$:

$$E(\mathcal{B}) = \int_{\Omega} \frac{\alpha}{2} (\|\mathbf{I} - \mathbf{I}_0\|^2 + \|\mathbf{J} - \mathbf{J}_0\|^2 + \|\mathbf{K} - \mathbf{K}_0\|^2) + \psi(\lambda_+^{\mathbf{I}}, \lambda_-^{\mathbf{I}}) + \psi(\lambda_+^{\mathbf{J}}, \lambda_-^{\mathbf{J}}) + \psi(\lambda_+^{\mathbf{K}}, \lambda_-^{\mathbf{K}}) d\Omega$$

where the $\lambda_{\pm}^{\mathbf{I}}$ are the two eigenvalues of the associated structure tensor $\mathbf{G}^{\mathbf{I}} = \sum_{j=1}^n \nabla I_j \nabla I_j^T$, as described in chapter 3.

Using the general solution (5.6), we can write the corresponding constrained set of 3D vector diffusion PDE's :

$$\begin{cases} \mathbf{I}_t = \mathcal{L}^{\mathbf{I}} - (\mathcal{L}^{\mathbf{I}} \cdot \mathbf{I}) \mathbf{I} - (\mathcal{L}^{\mathbf{J}} \cdot \mathbf{I}) \mathbf{J} - (\mathcal{L}^{\mathbf{K}} \cdot \mathbf{I}) \mathbf{K} \\ \mathbf{J}_t = \mathcal{L}^{\mathbf{J}} - (\mathcal{L}^{\mathbf{I}} \cdot \mathbf{J}) \mathbf{I} - (\mathcal{L}^{\mathbf{J}} \cdot \mathbf{J}) \mathbf{J} - (\mathcal{L}^{\mathbf{K}} \cdot \mathbf{J}) \mathbf{K} \\ \mathbf{K}_t = \mathcal{L}^{\mathbf{K}} - (\mathcal{L}^{\mathbf{I}} \cdot \mathbf{K}) \mathbf{I} - (\mathcal{L}^{\mathbf{J}} \cdot \mathbf{K}) \mathbf{J} - (\mathcal{L}^{\mathbf{K}} \cdot \mathbf{K}) \mathbf{K} \end{cases} \quad (5.12)$$

where $\mathcal{L}^{\mathbf{I}}, \mathcal{L}^{\mathbf{J}}, \mathcal{L}^{\mathbf{K}}$ are the unconstrained functional Lagrangian vectors, associated to each vector $\mathbf{I}, \mathbf{J}, \mathbf{K}$ and defined by (5.7). Note that this equation can be equivalently written with a *matrix PDE flow* :

$$\frac{\partial \mathbf{R}}{\partial t} = -\mathbf{L} + \mathbf{R} \mathbf{L}^T \mathbf{R} \quad (5.13)$$

where the matrices $\mathbf{R}$ and $\mathbf{L}$ are defined column by column :

$$\mathbf{R} = (\mathbf{I} \mid \mathbf{J} \mid \mathbf{K}) \quad \text{and} \quad \mathbf{L} = (\mathcal{L}^{\mathbf{I}} \mid \mathcal{L}^{\mathbf{J}} \mid \mathcal{L}^{\mathbf{K}})$$

□

The equation (5.13) corresponds then to an *orthogonal matrix-preserving regularizing PDE*. Note that its extension to higher matrix dimensions $O(n)$ is also valid. Developing (5.13) with

$$\mathbf{R} = (\mathbf{I}^{[1]} \mid \dots \mid \mathbf{I}^{[n]}) \quad \text{and} \quad \mathcal{L} = (\mathcal{L}^{[1]} \mid \dots \mid \mathcal{L}^{[n]})$$

gives the expression of the general orthonormal vector sets evolution (5.6) for $m = n$ (see also [48] for interesting developments on other matrix-valued flows).

• Comparison with other approaches for rotation regularization :

When dealing with rotation matrices, a natural idea is to decompose these matrices into more simple data that are easy to regularize (usually Euler angles, unit quaternions or rotation vectors), then reconstruct the final rotation field from smoothed versions of these data (Fig.5.7).

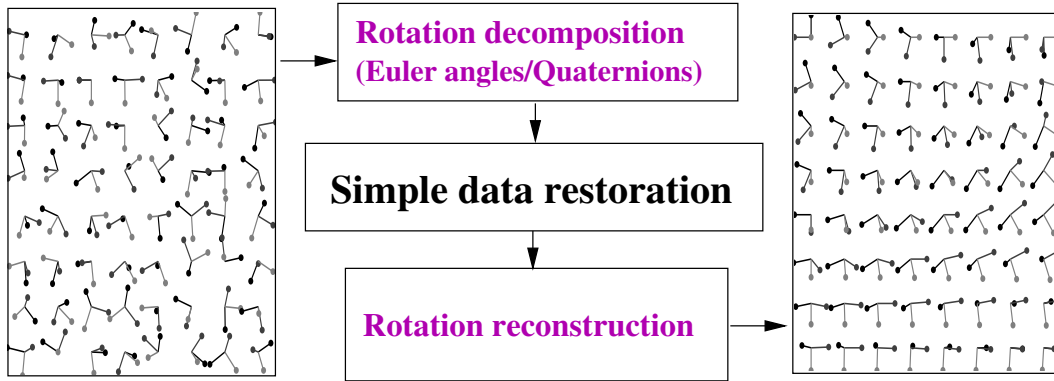


Figure 5.7: Decomposition method for rotation field regularization.

However this method has some drawbacks :

1. The conversions often induce *numerical imprecisions*.
2. The rotation decomposition *is not unique*. It would introduce annoying *discontinuities* in the decomposed data, even if the initial rotation field is perfectly smooth. These discontinuities are coming from :
 - The 2π -periodicity ambiguity in the Euler angles or in the rotation vector norms.
 - The double representation of a single rotation by two equivalent quaternions q and $-q$.

In both cases, these new created discontinuities have a large influence on the anisotropic regularization behavior and perturb the diffusion process.

Actually, using the orthogonal-matrix preserving flow (5.13) coming from the framework of orthonormal vector sets, solves this problem : we apply directly the PDE (5.13) on the rotation matrices sequence in order to regularize it. No rotation decompositions are needed anymore, and there are no false discontinuities since we work directly on the matrix coefficients which form a *unique representation* of a rotation $\mathbf{R}$ (Fig.5.8).

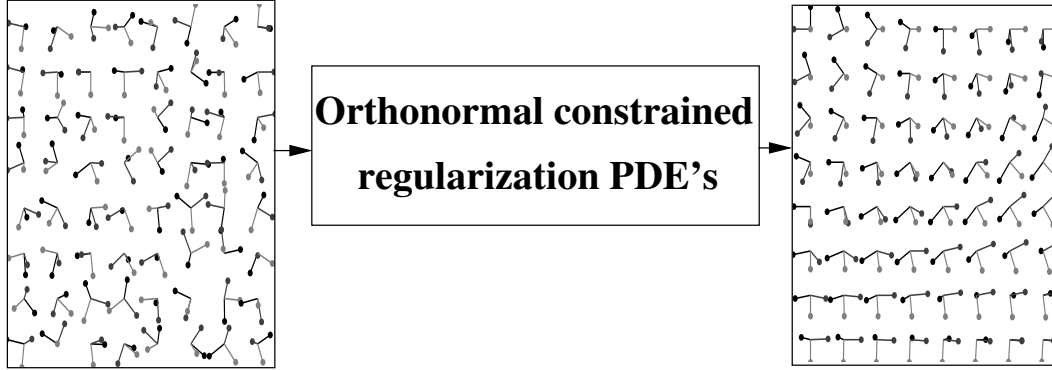


Figure 5.8: Rotation field regularization using orthonormal preserving PDE's.

5.3.3 A physical interpretation for 3D orthonormal vectors

The orthonormal vector set equation (5.6) has an intuitive physical interpretation for the case of 3D orthonormal bases, which is as follows :

The orthonormal vector sets $\mathcal{B}(\mathbf{x}) = \{ \mathbf{I}(\mathbf{x}), \mathbf{J}(\mathbf{x}), \mathbf{K}(\mathbf{x}) \}$ can be seen as solid objects composed of three orthogonal rigid stems of unit length, fixed at the same point $\mathbf{x}$, and submitted to forces $\mathbf{f}^{\mathbf{I}}$, $\mathbf{f}^{\mathbf{J}}$ and $\mathbf{f}^{\mathbf{K}}$ respectively (Fig.5.9).

A rotation around $\mathbf{x}$ is obviously the only motion that $\mathcal{B}$ can perform.

From a mechanical viewpoint, each force $\mathbf{f}^{\mathbf{I}}$, $\mathbf{f}^{\mathbf{J}}$ and $\mathbf{f}^{\mathbf{K}}$ induces a mechanic momentum on this object :

$$\omega_{\mathbf{I}} = \mathbf{I} \times \mathbf{f}^{\mathbf{I}}, \quad \omega_{\mathbf{J}} = \mathbf{J} \times \mathbf{f}^{\mathbf{J}}, \quad \text{and} \quad \omega_{\mathbf{K}} = \mathbf{K} \times \mathbf{f}^{\mathbf{K}}$$

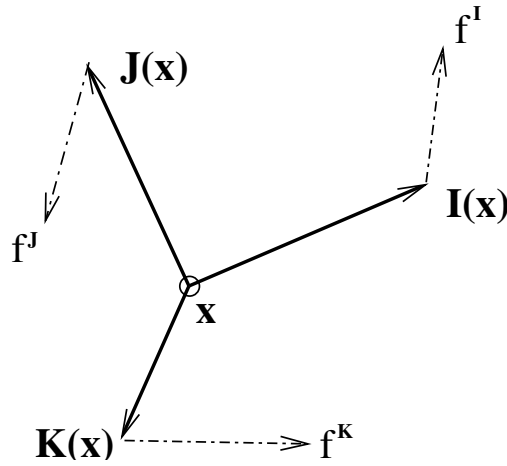
Where $\times$ designates the usual cross product in $\mathbb{R}^3$. Then, the total momentum ω applied to the object $\mathcal{B}$ is given by :

$$\omega = \omega_{\mathbf{I}} + \omega_{\mathbf{J}} + \omega_{\mathbf{K}} \tag{5.14}$$

i.e

$$\omega = (\mathbf{I} \times \mathbf{f}^{\mathbf{I}}) + (\mathbf{J} \times \mathbf{f}^{\mathbf{J}}) + (\mathbf{K} \times \mathbf{f}^{\mathbf{K}}) \tag{5.15}$$

If we suppose that $\mathcal{B}$ has an unit moment of inertia, we can express the velocities $\mathbf{v}^{\mathbf{I}}$, $\mathbf{v}^{\mathbf{J}}$ and $\mathbf{v}^{\mathbf{K}}$

Figure 5.9: A solid object $\mathcal{B}$, submitted to physical forces.

at each free extremity of the stems, corresponding to *the constrained motion* of the solid :

$$\begin{cases} \mathbf{v}^I = \boldsymbol{\omega} \times \mathbf{I} \\ \mathbf{v}^J = \boldsymbol{\omega} \times \mathbf{J} \\ \mathbf{v}^K = \boldsymbol{\omega} \times \mathbf{K} \end{cases}$$

Developing these expressions, using the double vector product formula

$$\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) = (\mathbf{u} \cdot \mathbf{w}) \mathbf{v} - (\mathbf{u} \cdot \mathbf{v}) \mathbf{w}$$

and the orthogonal properties $\mathbf{I}^{[k]} \cdot \mathbf{I}^{[l]} = \delta_{kl}$, leads to :

$$\begin{cases} \mathbf{v}^I &= \mathbf{f}^I - (\mathbf{f}^I \cdot \mathbf{I}) \mathbf{I} - (\mathbf{f}^J \cdot \mathbf{I}) \mathbf{J} - (\mathbf{f}^K \cdot \mathbf{I}) \mathbf{K} \\ \mathbf{v}^J &= \mathbf{f}^J - (\mathbf{f}^I \cdot \mathbf{J}) \mathbf{I} - (\mathbf{f}^J \cdot \mathbf{J}) \mathbf{J} - (\mathbf{f}^K \cdot \mathbf{J}) \mathbf{K} \\ \mathbf{v}^K &= \mathbf{f}^K - (\mathbf{f}^I \cdot \mathbf{K}) \mathbf{I} - (\mathbf{f}^J \cdot \mathbf{K}) \mathbf{J} - (\mathbf{f}^K \cdot \mathbf{K}) \mathbf{K} \end{cases}$$

A velocity is an infinitesimal variation of a vector during the time ∂t :

$$\frac{\partial \mathbf{I}}{\partial t} = \mathbf{v}^I, \quad \frac{\partial \mathbf{J}}{\partial t} = \mathbf{v}^J, \quad \frac{\partial \mathbf{K}}{\partial t} = \mathbf{v}^K$$

If we choose the forces $\mathbf{f}^I$, $\mathbf{f}^J$, $\mathbf{f}^K$ to be defined by (5.7), we find then the expected regularization PDE (5.6) that preserves the orthogonal constraints : the functional (5.4) can then be seen as a mechanic energy associated to a rigid object $\mathcal{B}$, submitted to three *pure diffusion forces* $\mathbf{f}^I$, $\mathbf{f}^J$, $\mathbf{f}^K$. The obtained PDE's are expressions of the *instant rotations* applied to $\mathcal{B}$ in order to minimize this energy.

In this section (5.3), we studied two particular cases (unit vectors and orthogonal matrices) that can be directly handled by the orthonormal vector set regularization framework (5.6). Applications of the regularization for these two type of fields will be illustrated in chapter 6.

Let us now consider the regularization of a more complex and interesting orientation feature : *diffusion tensors*.

5.4 Diffusion tensor regularization

We are now interested in regularizing noisy diffusion tensor fields $\mathbf{T}$ defined on a continuous domain Ω of $\mathbb{R}^p$ (usually $p = 1, 2, 3$). Diffusion tensors are complex orientation features (see section 1.2.4) encountered in a lot of different datasets. This work has been motivated by the restoration of 3D DT-MRI medical images of the brain (as described in [115, 145] and illustrated on chapter 6), but can be used without restriction to general symmetric and semi-positive definite $n \times n$ matrices, as for instance covariance matrices or structure tensors [187, 77, 118]. Diffusion tensors are also encountered in fields of statistical parameters in Doppler spectral analysis [21] ¹. The problem of diffusion tensor regularization has been already considered in [61, 181], with quite different approaches.

5.4.1 Direct approach for tensor regularization

A first and naive approach would be to consider a tensor field $\mathbf{T} : \Omega \rightarrow \mathcal{P}(n)$, as a multi-valued image where each point of the field has n^2 components. Due to the symmetric property of the matrices $\mathbf{T}$, we can even reduce the vector dimension to $n(n+1)/2$.

Then, a natural idea would be to evolve the tensor components $T_{i,j}$ with classic vector-valued diffusion PDE's, as for instance the scheme proposed in chapter 3 :

$$\frac{\partial T_{i,j}}{\partial t} = \text{trace} \left(\left[\frac{1}{\sqrt{\lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T} + \frac{1}{\lambda_+^* + \lambda_-^*} \theta_+^* \theta_+^{*T} \right] \mathbf{H}_{i,j} \right) \quad (5.16)$$

where $\mathbf{H}_{i,j}$ is the Hessian of the matrix component $T_{i,j}$, and $\lambda_{\pm}^*$, $\theta_{\pm}^*$ are the spectral element of a smoothed structure tensor $G = \sum_{i,j} \nabla T_{i,j} \nabla T_{i,j}^T$.

It is also worth to mention that the PDE (5.16) *intrinsically preserves the matrix symmetry* and can then be applied only on the upper triangular part of $\mathbf{T}$.

As expected, the semi-positive definite constraint needs more attention. Note that as the corresponding constrained space $\mathcal{P}(n)$ is a *cone*, any linear combination of elements of $\mathcal{P}(n)$ is itself positive-definite. This is often the case for classical regularization PDE's. Anyway, we propose here a simple method allowing to deal with general PDE's, as well as ensuring the numerical preservation of the positive constraint during a PDE flow.

Indeed, a simple way to *numerically* preserve it is to reproject after each PDE iteration, the regularizing matrices $\mathbf{T}$ into the semi-positive space. This is done as follows.

Considering the spectral decomposition of a symmetric matrix $\mathbf{T}$:

$$\mathbf{T} = \mathbf{U} \mathbf{\Gamma} \mathbf{U}^T$$

where $\mathbf{U} \in \text{SO}(n)$ is a rotation and $\mathbf{\Gamma}$ is a diagonal matrix with supposed negative eigenvalues ($\mathbf{T}$ has stepped out from the positive-matrix space), we can compute the projection $\mathcal{P}(\mathbf{T})$ of $\mathbf{T}$ into the constrained space of semi-positive matrices, (with respect to the Frobenius norm) as :

$$\mathcal{P}(\mathbf{T}) = \mathbf{U} \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) \mathbf{U}^T$$

¹We thank F. Barbaresco who recently brought this problem to our attention.

where

$$\begin{cases} \tilde{\lambda}_l = \lambda_l & \text{if } \lambda_l \geq 0 \\ \tilde{\lambda}_l = 0 & \text{if } \lambda_l < 0 \end{cases}$$

Proof : Let $\mathbf{A} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$ be a symmetric matrix ($\mathbf{U}$ is orthogonal and $\mathbf{\Gamma} = \text{diag}(\lambda_l)$).

We are looking for the semi-positive definite matrix $\mathbf{B}$ that minimizes the Frobenius norm :

$$\|\mathbf{A} - \mathbf{B}\|_F^2 = \|\mathbf{U}\mathbf{\Gamma}\mathbf{U}^T - \mathbf{B}\|_F^2 = \|\mathbf{\Gamma} - \mathbf{U}^T\mathbf{B}\mathbf{U}\|_F^2$$

(the norm is independent of the basis).

$\mathbf{U}^T\mathbf{B}\mathbf{U}$ is symmetric. Let us call $\mathbf{\Gamma}'$ the matrix of its diagonal elements and $\mathbf{S}$ its lower triangular matrix such as :

$$\mathbf{U}^T\mathbf{B}\mathbf{U} = \mathbf{\Gamma}' + \mathbf{S} + \mathbf{S}^T$$

Then, we have :

$$\|\mathbf{A} - \mathbf{B}\|_F^2 = \|\mathbf{\Gamma} - \mathbf{\Gamma}'\|_F^2 + 2\|\mathbf{S}\|_F^2$$

The minimum is reached when

$$\mathbf{S} = 0, \quad \text{i.e.} \quad \mathbf{B} = \mathbf{U}\mathbf{\Gamma}'\mathbf{U}^T$$

and then

$$\|\mathbf{A} - \mathbf{B}\|_F^2 = \|\mathbf{\Gamma} - \mathbf{\Gamma}'\|_F^2 = \sum_{i=1}^n (\lambda_i - \mu_i)^2$$

where $\mu_1, \dots, \mu_n$ are the positive eigenvalues of $\mathbf{B}$. Finally,

$$\|\mathbf{A} - \mathbf{B}\|_F^2 \text{ is minimum} \quad \Longleftrightarrow \quad \begin{cases} \mu_i = \lambda_l & (\text{if } \lambda_l \geq 0) \\ \mu_i = 0 & (\text{if } \lambda_l < 0) \end{cases}$$

□

This reprojection step numerically ensures the semi-positive property of the tensors $\mathbf{T}$ during the PDE flow (5.16). It can be used for general regularization PDE evolutions. Note that, as described in [191], a wide variety of regularizing flows actually preserves intrinsically this positive-constraint.

However, this technique has some drawbacks :

- When needed, the reprojection requires a spectral decomposition of the matrix field $\mathbf{T}$, in order to get the eigenvalues λ_l . This is a very time consuming procedure.
- We do not have a direct control on the spectral elements of the tensors, which are the essential features of interest that characterizes $\mathbf{T}$: they decompose the tensor informations (orientations $\mathbf{U}$ and diffusivity $\mathbf{\Gamma}$) and are relevant data for designing an adapted regularization process.

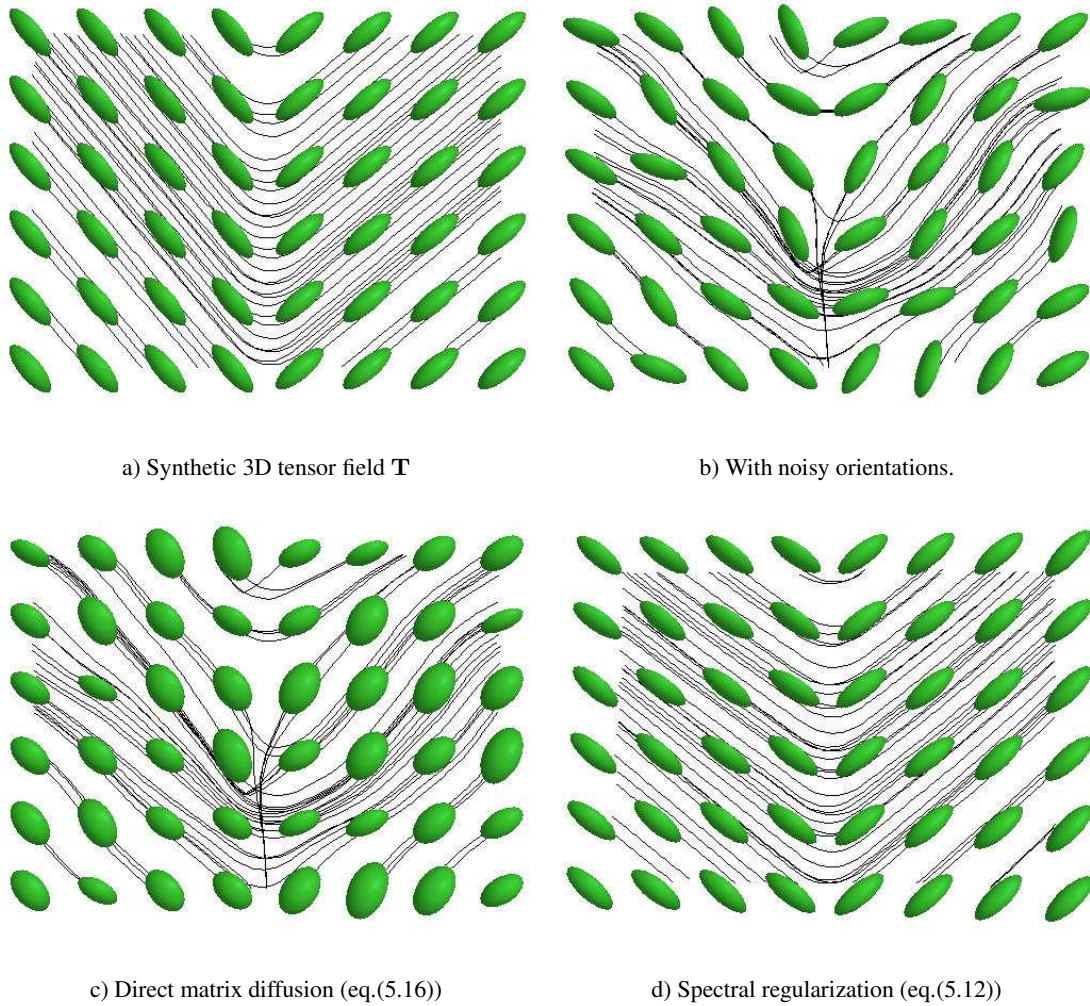


Figure 5.10: Comparison of tensor regularization methods.

One consequence of this direct diffusion matrix method is illustrated in the Fig.5.10.

A noisy synthetic tensor field $\mathbf{T}$ has been regularized using a direct matrix diffusion (5.16) (Fig.5.10c). Note that for illustration purposes, the noise has been mainly added to the tensor directions $\mathbf{U}$.

Clearly, there is an *eigenvalue swelling effect*, due to the fact that the diffusion PDE (5.16) regularizes tensor orientations and diffusivities in a coupled way, that is not desirable in the general case. As an effect, neighboring tensors with near orthogonal directions are not aligning themselves, but swell. A high risk of losing the tensor orientations is encountered : the tensors are indeed converging to *identity matrices* in those cases.

In Fig.5.10d, we rather used a *spectral regularization* method that will be described in the next section 5.4.2. It allows to retrieve more coherent diffusion tensors, thanks to the separation between tensor orientations and diffusivities for the regularization process.

Fig.5.10 illustrates this problem : the black lines represented behind the tensors, are the lines

following the main eigenvector λ_1 of each $\mathbf{T}(\mathbf{x})$.

An analogy can be made with chromaticity denoising in color images. The direct regularization of color vectors (R, G, B) regularizes in a coupled way the orientation and the brightness features of the colors. A separate regularization can be useful to retrieve more detailed images (as it will be illustrated on chapter 6).

This is particularly true for our application of interest : the regularization of DT-MRI images. One of the aims is indeed to retrieve the fiber networks, by following the main directions of tensors $\mathbf{T}$ defined on a volume $\Omega \subset \mathbb{R}^3$. Using a direct regularization, as (5.16) would lose some fiber directions, which is an undesired property.

5.4.2 Spectral regularization approach

In order to avoid the undesired swelling eigenvalue effect due to direct tensor regularization, we rather propose to work directly on the spectral decomposition of the noisy tensor field $\mathbf{T}$, while preserving the tensor constraints *in the spectral space*.

This is justified by the fact that spectral elements of diffusion tensors $\mathbf{T} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$ are the important data that provide significant structural informations :

- For DT-MRI images, the diagonal matrix $\mathbf{\Gamma} = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ measures the water molecule velocity in the brain fibers, while the tensor orientation $\mathbf{U}$ provides important clues to the structure and geometric organization of these fibers.

Significant physiological values can also be computed from $\mathbf{\Gamma}$ [115] :

- Mean diffusivity : $Tr = \lambda_1 + \lambda_2 + \lambda_3$,
- Partial anisotropy : $FA = \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_1 - \lambda_3)^2 + (\lambda_2 - \lambda_3)^2}{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}}$,
- Volume ratio : $VR = 27 \lambda_1 \lambda_2 \lambda_3 / (\lambda_1 + \lambda_2 + \lambda_3)$.
- For structure tensors of color images [187], $\mathbf{\Gamma}$ and $\mathbf{U}$ measure the color variations and their corresponding directions.
- For covariance matrices, $\mathbf{\Gamma}$ represents the standard deviations of the data dispersion along the main axes given by $\mathbf{U}$.

These few examples clearly illustrate how the spectral decomposition of a tensor is directly related to a better understanding of its structure. Moreover, the tensor constraints (semi-positive definiteness and symmetry) can be easily *expressed in the spectral domain* :

$$\left\{ \begin{array}{ll} \text{Semi-positivity :} & \forall l \quad \lambda_l \geq 0 \\ \text{Orthogonality :} & \forall k, l, \quad \mathbf{u}^{[k]} \cdot \mathbf{u}^{[l]} = \delta_{k,l} \end{array} \right. \quad (5.17)$$

The matrix $\mathbf{\Gamma}$ is then directly linked to the tensor semi-positivity while the orthogonal property of the matrix $\mathbf{U}$ is related to the symmetry of the corresponding tensor $\mathbf{T}$.

Our spectral method is based on two constrained and coupled regularizations acting on $\mathbf{\Gamma}$ and $\mathbf{U}$:

• **Regularization of the tensor diffusivities :**

Different anisotropic PDE's can be used to regularize the tensor diffusivities $\mathbf{D} = \text{diag}(\lambda_l)$, depending on the considered application. For instance, the following diffusion schemes could be considered for analysis :

- Process each eigenvalue λ_l *separately*, with classical scalar regularization schemes (see chapter 2).
- Process the vector $\boldsymbol{\lambda} = (\lambda_l)$ using vector-valued diffusion PDE's (as those proposed in chapter 3).
- Include a-priori spectral informations inside the diffusion equation, in order to drive the diffusion process. For instance, it could be done like this, for DT-MRI regularization purposes :

$$\frac{\partial \lambda_l}{\partial t} = \text{div}(\mathbf{D}(\lambda_l, FA, VR, \dots) \nabla \lambda_l)$$

where $\mathbf{D}$ is a diffusion tensor that drives the regularization process.

Working on the spectral domain allows much more freedom in the choice of the diffusion terms acting on the tensor diffusivities λ_i of $\mathbf{T}$, than a direct matrix restoration.

Note that the semi-positivity constraint of the eigenvalues can be imposed simply by using a discretized scheme that satisfies the *maximum and minimum principle* [23]. Starting from an initial semi-positive tensor field $\mathbf{T}_0$, the tensor eigenvalues will always be positive.

• **Regularization of the tensor orientations :**

The difficult part of our spectral regularization method would come from the preservation of the orthogonality of $\mathbf{U}$ during the regularizing flow. Indeed, using unconstrained vector diffusion PDE's on each eigenvector $\mathbf{u}^{[l]}$ (the columns of $\mathbf{U}$) is not suitable here, since it doesn't preserve intrinsically the orthonormal constraints of $\mathbf{U}$.

Following the idea of our orthonormal-preserving PDE (section 5.3.2, we can directly provide a simple and efficient way to restore the field of tensor orientation $\mathbf{U}$:

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{L} + \mathbf{U} \mathbf{L}^T \mathbf{U} \quad (5.18)$$

where $\mathbf{L}$ is the matrix corresponding to an *unconstrained Lagrangian* describing the regularization process. For instance, we can choose :

$$\mathbf{L}_{ij} = \text{trace}(\mathbf{D} \mathbf{H}_{ij})$$

as described in chapter 3.

• **A local alignment method :**

When dealing with diffusion tensors, one has to take care of the non-uniqueness of the spectral decomposition :

$$\mathbf{T} = \sum_{k=1}^n \lambda_k \mathbf{u}^{[k]} \mathbf{u}^{[k]T}$$

Flipping one eigenvector direction while keeping its orientation (i.e considering $-\mathbf{u}^{[l]}$ instead of $\mathbf{u}^{[l]}$) gives the same tensor $\mathbf{T}$: 2^n configurations actually can represent its orientation $\mathbf{U}$.

This means that a constant tensor field may be decomposed into *highly discontinuous* orientation fields $\mathbf{U}$, disturbing the anisotropic regularization process with false discontinuity detections.

To overcome this problem, a *local eigenvector alignment process* can be made before applying the PDE on each tensor of the field $\mathbf{T}$. The idea is to align the neighboring eigenvector directions with the current one. This is done by minimizing the angles between them, constraining the dot product to be positive by flipping the neighboring eigenvectors if necessary :

$$\forall N \in \mathcal{V}(M), \quad \tilde{\mathbf{u}}^{[i]}(N) = \text{sign}(\mathbf{u}^{[i]}(N) \cdot \mathbf{u}^{[i]}(M)) \mathbf{u}^{[i]}(N)$$

where $\mathcal{V}(M)$ is a neighborhood of $\mathbf{x}$ (Fig.5.11), and the $\mathbf{u}^{[i]}$ are the eigenvectors of $\mathbf{T}$, i.e. the columns of the orthogonal matrix $\mathbf{U}$.

This local operation allows to act on the vector orientations while ignoring the direction information. Then, we can apply the orthogonal constrained diffusion PDE eq.(5.18). The importance of this procedure will be shown in chapter 6, with real DT-MRI datasets regularization. Moreover, an alternative solution avoiding this pre-alignment step will be proposed for the case of diffusion tensor regularization, thanks to the consideration of *isospectral flows*.

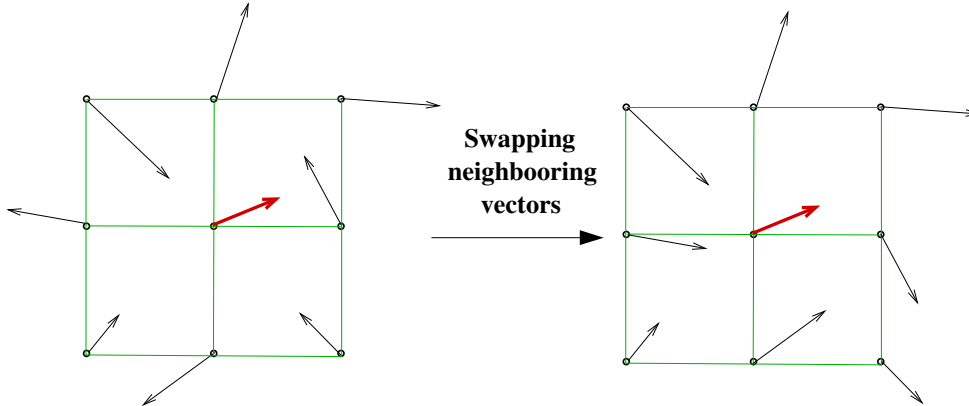


Figure 5.11: Local vector alignment procedure for tensor regularization.

5.5 Numerical schemes for orthonormal vector set regularization

All the generic PDE flows (5.6), (5.11), (5.13) acting on orthonormal vector sets have this particular form :

$$\frac{\partial \mathbf{I}^{[k]}}{\partial t} = \boldsymbol{\beta}^{[k]} \quad \text{with} \quad \begin{cases} \boldsymbol{\beta}^{[k]} \perp \mathbf{I}^{[k]} \\ \|\mathbf{I}^{[k]}\| = 1 \end{cases}$$

Indeed, if we use the general expression (5.6) of orthonormal vector sets evolution for $\boldsymbol{\beta}^{[k]}$:

$$\begin{aligned} \boldsymbol{\beta}^{[k]} \cdot \mathbf{I}^{[k]} &= \left(-\mathcal{L}^{[k]} + \sum_{l=1}^m \left(\mathcal{L}^{[l]} \cdot \mathbf{I}^{[k]} \right) \mathbf{I}^{[l]} \right) \cdot \mathbf{I}^{[k]} \\ &= -\mathcal{L}^{[k]} \cdot \mathbf{I}^{[k]} + \mathcal{L}^{[k]} \cdot \mathbf{I}^{[k]} \\ &= 0 \end{aligned}$$

The PDE velocity $\boldsymbol{\beta}^{[k]}$ is then anytime *orthogonal to the current vector* $\mathbf{I}^{[k]}$ (It is generally the case for vector PDE's acting on *orientation features*, as in [39, 106, 142, 168, 174, 182]).

This means that the vector $\mathbf{I}^{[k]}(\mathbf{x})$ should theoretically perform a rotation motion around $\mathbf{x}$, preserving its norm. But using *classic explicit schemes* as

$$\mathbf{I}_{(t+dt)}^{[k]} = \mathbf{I}_{(t)}^{[k]} + dt \boldsymbol{\beta}^{[k]} \quad \text{where} \quad 0 < dt \ll 1$$

leads to numerical errors since the underlying manifold is non-flat and the unit vectors $\mathbf{I}^{[k]}$ will step out from the corresponding constrained non-flat manifold (Fig.5.12).

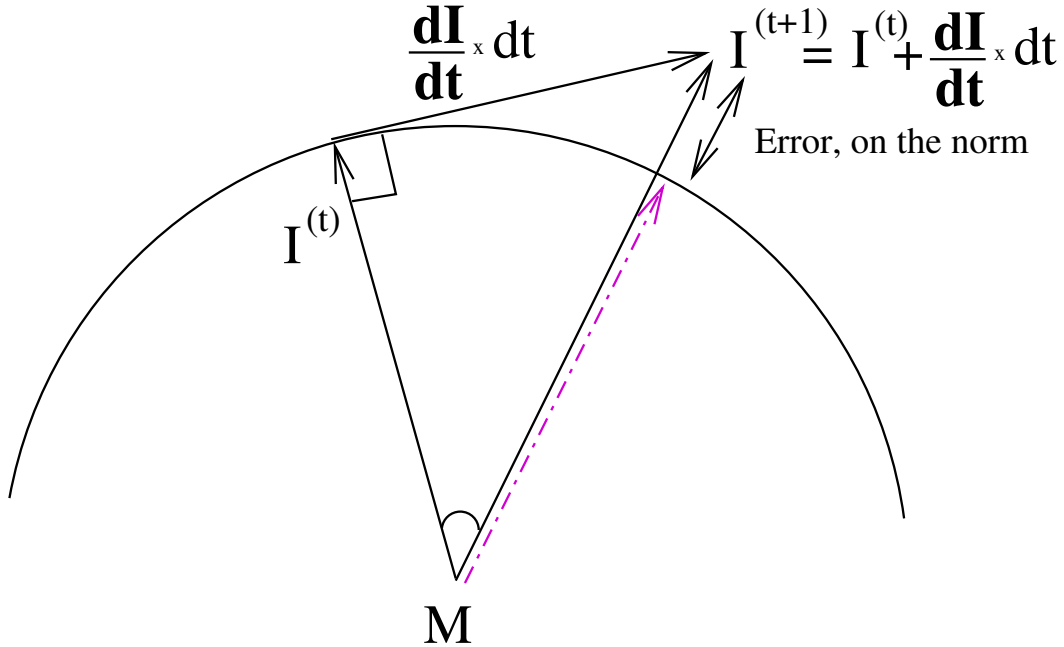


Figure 5.12: Numerical errors with classical explicit schemes.

In [27, 168] for direction vector diffusion purposes, this problem is avoided thanks to a re-normalization step of the vector $\mathbf{I}^{[k]}$ after few PDE iterations (see also recent developments in [182] for an interesting alternative solution). Anyway, this method cannot be applied when dealing with orthonormal vector sets, because the orthogonal angles between vectors may not be preserved by this way.

The mechanical interpretation of the 3D case (section 5.3.3) provides us a simple and accurate solution to this re-normalization problem : we apply at each time step the *instant rotation* corresponding to the evolution equation (5.13) on the orthonormal 3D basis $\mathcal{B}$. This rotation is the same for all the vectors $\mathbf{I}, \mathbf{J}, \mathbf{K}$ (section 5.3.3) and is given by the rotation vector $\boldsymbol{\omega}$, defined by the mechanic momentum :

$$\boldsymbol{\omega} = (\mathbf{I} \times \mathcal{L}^{\mathbf{I}}) + (\mathbf{J} \times \mathcal{L}^{\mathbf{J}}) + (\mathbf{K} \times \mathcal{L}^{\mathbf{K}})$$

where the $\mathcal{L}$ are defined by (5.7) and are discretized with classic finite-difference schemes [113] (their discretizations will be independent of the constraint preservation). The *infinitesimal rotation* matrix Γ corresponding to the instant rotation $\boldsymbol{\omega}$ is computed thanks to the Rodriguez' formula [71] :

$$\Gamma = e^{Hdt} = \mathbb{I} + \frac{\sin \|\boldsymbol{\omega}dt\|}{\|\boldsymbol{\omega}dt\|} Hdt + \frac{1 - \cos \|\boldsymbol{\omega}dt\|}{\|\boldsymbol{\omega}dt\|^2} H^2 dt^2$$

where $dt > 0$ is the time step and H is the skew-symmetric matrix associated to the rotation vector *omega* :

$$H = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

Then, the evolution scheme for the matrix-valued PDE (5.13) in the 3D case is simply :

$$\mathbf{R}_{(t+dt)} = \Gamma \mathbf{R}_{(t)}$$

where

$$\mathbf{R} = (\mathbf{I} \mid \mathbf{J} \mid \mathbf{K})$$

It provides a numerical way to preserve the unitary norm, as well as the orthogonal angles (the column vectors $\mathbf{I}, \mathbf{J}, \mathbf{K}$ of $\mathbf{R}$ perform *the same infinitesimal rotation* Γ at each time step t). The numerical error, due to ($dt \neq 0$), is only present in the rotation angle $\|\boldsymbol{\omega}\|$, but doesn't affect the orthonormal vector bases configuration.

Note also that this scheme naturally *preserves the determinants* of the corresponding orthogonal matrices $\mathbf{R}$, during the flow. Indeed,

$$\begin{aligned} \det(\mathbf{R}_{(t+dt)}) &= \det(\Gamma \mathbf{R}_{(t)}) \\ &= \det(\Gamma) \det(\mathbf{R}_{(t)}) \\ &= \det(\mathbf{R}_{(t)}) \end{aligned} \tag{5.19}$$

since by construction $\Gamma \in \text{SO}(3)$ and then $\det(\Gamma) = +1$. More particularly, it means that a rotation $\mathbf{R} \in \text{SO}(3)$ *cannot be transformed to a rotoinversion* during the orthogonal PDE flow (5.13).

It validates then our proposed method for rotation field regularization.

5.6 Extension to general matrix-constrained flows

Preliminary note : The main theory presented here has been mainly developed by Christophe Chefd'hotel / ODYSSEE, INRIA Sophia-Antipolis. Working on matrix-valued flows, he crossed some of the general issues presented in this thesis. We naturally started a collaboration leading to the improvement of our initial diffusion tensor regularization method, described in section 5.4. We will illustrate the general ideas of this interesting framework, and the reader is requested to refer to [48] and to the outgoing thesis of Christophe for more informations about the calculus leading to the presented equations.

5.6.1 Matrix constraints and exponential maps

The topic of this work is finding matrix-valued PDE flows (and more particularly *regularization PDE's*), that preserve some *nonlinear* constraints, as for instance orthogonal, positive-definite or spectral constraints.

Let us consider the manifold $\mathcal{M} = \mathcal{F}(\Omega, \mathcal{N})$ embedded in an infinite dimensional function space $\mathcal{F}(\Omega, \mathbb{R}^{n \times n})$. Each point X of $\mathcal{M}$ is a mapping $X : \Omega \rightarrow \mathcal{N}$, defined on an open subset Ω of the Euclidean space, which takes values in a *constrained matrix manifold* $\mathcal{N}$. The constraints are considered as *nonlinear*, which means that $\mathcal{M}$ is a *non-flat* manifold.

In order to find a constraint-preserving matrix flow, we naturally want to find a PDE velocity that is anywhere *tangent* to the manifold $\mathcal{M}$ (Fig.5.13) ².

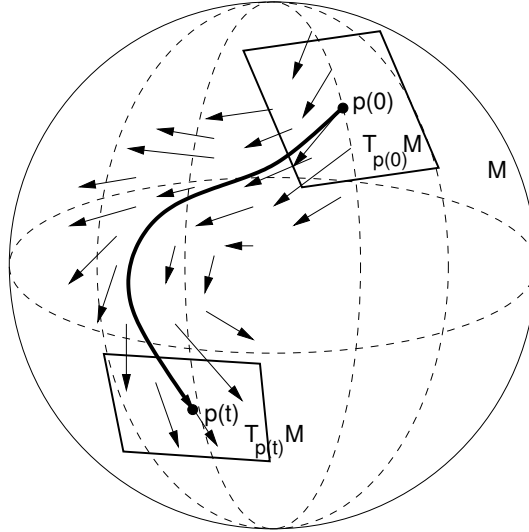


Figure 5.13: Matrix-constrained flow, seen as a tangent flow to a non-flat manifold $\mathcal{M}$.

This tangent space follows readily from the expression of the tangent space of $\mathcal{N}$.

Actually, Christophe Chefd'hotel noticed in [48] that a wide range of orthogonal and spectral constraints on matrices define *Lie Groups* and *homogeneous spaces*, which are submanifolds of

²This figure has been kindly provided by C. Chefd'hotel/ODYSSEE - INRIA Sophia-Antipolis.

$\mathbb{R}^{n \times n}$ that present nice algebraic properties. In particular, the tangent space at any given point can be formally found.

Moreover, the geodesics in such manifolds can be computed using *exponential maps*. It contributes to design explicit numerical schemes that *numerically preserve the matrix constraints*, with a discretization expression based on matrix exponentials. Note that a classical explicit Euler method, such as :

$$\frac{\partial M}{\partial t} = \beta_M \quad \Longrightarrow \quad M_{(t+dt)} = M_{(t)} + dt \beta_M$$

is not suitable for *non-flat manifolds*, since there is a risk of *stepping out from $\mathcal{M}$* for each (non infinitesimal) displacement (Fig.5.12). A first idea would be to project after each step the point $M_{(t+dt)}$ on the underlying constrained manifold. But this post-processing can be avoided, using a geometric integration step relying on *exponential maps*, following directly a geodesic path between each iteration (Fig.5.14)³.

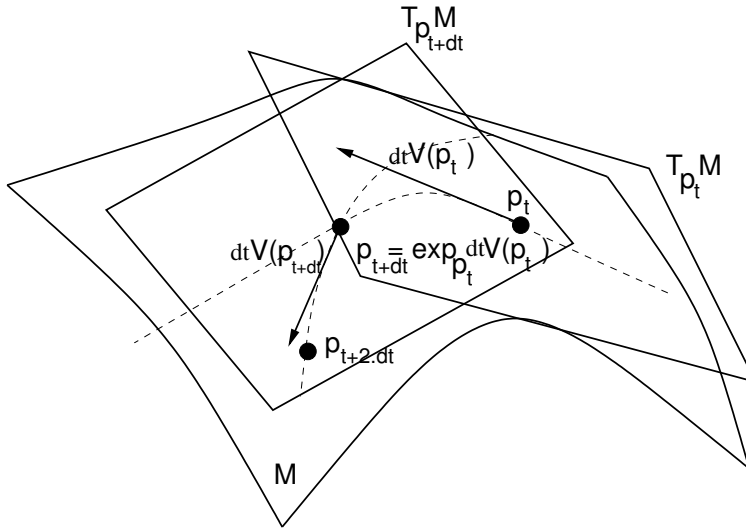


Figure 5.14: Using exponential maps to numerically preserve constraints.

In general, the equation of a geodesic satisfies a second order differential equation. But the fact is that for simple matrix manifolds, closed form solutions are often available (and thus the exponential map).

5.6.2 Three different matrix-valued flows for tensor regularization

In the followings, different choices of manifold $\mathcal{N}$ are considered, leading to different matrix constraints. For each case, the corresponding constraint-preserving flow is given, as well as the adapted numerical scheme, using matrix exponential. We will also show how these different constraints can be considered, in order to deal with the regularization of diffusion tensor field $\mathbf{T}$.

³This figure has been kindly provided by C. Chef d'hôtel/ODYSSEE - INRIA Sophia-Antipolis.

- **Positive-definite flow :**

We first consider the manifold $\mathcal{N} = \text{P}(n)$, of the symmetric and positive-definite matrices.

The corresponding constraint-preserving flow is then :

$$-\frac{\partial \mathbf{T}}{\partial t} = -((\mathcal{L} + \mathcal{L}^T) \mathbf{T}^2 + \mathbf{T}^2 (\mathcal{L} + \mathcal{L}^T)). \quad (5.20)$$

where $\mathcal{L}$ corresponds to an *unconstrained regularization term*, such as the ones proposed in chapter 3. Note that the PDE (5.20) allows to evolve directly the matrix coefficients while preserving the symmetric positive definiteness, and evolving the matrices with respect to the natural metric of $\mathcal{N}$.

The corresponding numerical scheme is

$$\mathbf{T}_{(t+dt)} = \exp(-\mathbf{T}_{(t)}(\mathcal{L} + \mathcal{L}^T) dt)^T \mathbf{T}_{(t)} \exp(-\mathbf{T}_{(t)}(\mathcal{L} + \mathcal{L}^T) dt), \quad (5.21)$$

where

$$\exp(\mathbf{T}) = \sum_{i=0}^{\infty} \frac{M_i}{i!}$$

denotes *the matrix exponential*, numerically implemented using a *Padé approximation*, as described in [86].

The interest behind this positive-definite PDE (5.20) mainly comes from the fact that the natural metric of the space $\text{P}(n)$ is taken into account for the design of the regularizing equation (see [48] for more details)

- **Orthogonal-preserving flow :**

As we noticed in section 5.4.2, eigenvalue over-smoothing can be avoided by regularizing the orientation part $\mathbf{U}$ and the diffusivity part $\mathbf{\Gamma}$ of a diffusion tensor field $\mathbf{T}$, with different coupled equations.

By considering the specific constrained manifold $\mathcal{N} = \text{O}(n)$, in order to regularize the tensor orientation $\mathbf{U}$, it ends up with the equation

$$\frac{\partial \mathbf{T}}{\partial t} = \mathbf{T} \mathcal{L}^T \mathbf{T} - \mathcal{L}, \quad (5.22)$$

We find again, with another formalism, the equation (5.13) already used to regularize orthogonal matrices, in section 5.3.2. Besides, an interesting numerical scheme extension comes naturally from the use of exponential maps :

$$\mathbf{T}_{(t+dt)} = \mathbf{T}_{(t)} \exp(-dt \{ \mathbf{T}_{(t)}, \mathcal{L} \}). \quad (5.23)$$

Note that for the case of 3×3 orthogonal matrices, we find again the proposed scheme in 5.5, since (5.23) reduces then to the well known *Rodriguez' formula*.

The problem of local orientation alignment also occurs here (Fig.5.11) and is solved in the same way as in section 5.4.2.

- **Isospectral flow** :

To avoid the eigenvector alignment step needed by the previous method (5.22), a more simple method consists in considering the constrained manifold $\mathcal{N}$ of the matrices with given sets of eigenvalues. The resulting constrained equation consists then in applying an *isospectral flow* on the initial tensor field, i.e. a regularizing flow that preserves the tensor diffusivities, while regularizing the tensor orientations :

$$\frac{\partial \mathbf{T}}{\partial t} = [\mathbf{T}, [\mathbf{T}, -(\mathcal{L} + \mathcal{L}^T)]] . \quad (5.24)$$

One can also obtain a suitable integration scheme for eq. (5.24), derived from the expression of the exponential map on the orthogonal group, and such that

$$\begin{aligned} \forall x \in \Omega, \mathbf{T}_{(t+dt)}(x) &= \mathbf{A}_{(t)}(x)^T \mathbf{T}_{(t)}(x) \mathbf{A}_{(t)}(x), \text{ with} \\ \mathbf{A}_{(t)}(x) &= e^{-dt[\mathcal{L}^T(x) + \mathcal{L}(x), \mathbf{T}_{(t)}(x)]}. \end{aligned} \quad (5.25)$$

The results are similar to the ones obtained with the decomposition and the orthogonal constraints. However, the computation cost is significantly reduced (no local alignment steps are needed anymore for regularizing tensor orientations). The PDE (5.24) applies directly on the original matrix coefficients $\mathbf{T}_{i,j}$.

- **Implementation considerations** :

The implementation is simple and follows strictly the proposed numerical schemes (5.21), (5.23) and (5.25). Besides, we have to mention that the vector-valued regularization term $\mathcal{L}$, corresponding to the unconstrained part of the equation is indeed spatially discretized using the techniques proposed in section 3.5. As for the exponential maps, they are numerically computed using a *Padé approximation*, as proposed in [85].

5.6.3 Comparative figures

We illustrate in Fig.5.15, the different behavior of our proposed diffusion tensor regularizing flows. The equations have been applied on a synthetic 2D image of noisy 3×3 diffusion tensors $\mathbf{T}$.

Some interesting remarks can be made on these different regularization methods :

- The semi-positive preserving method (Fig.5.15c) is not well suitable to handle diffusion tensor field, since no real control on the regularization of the spectral elements is possible.

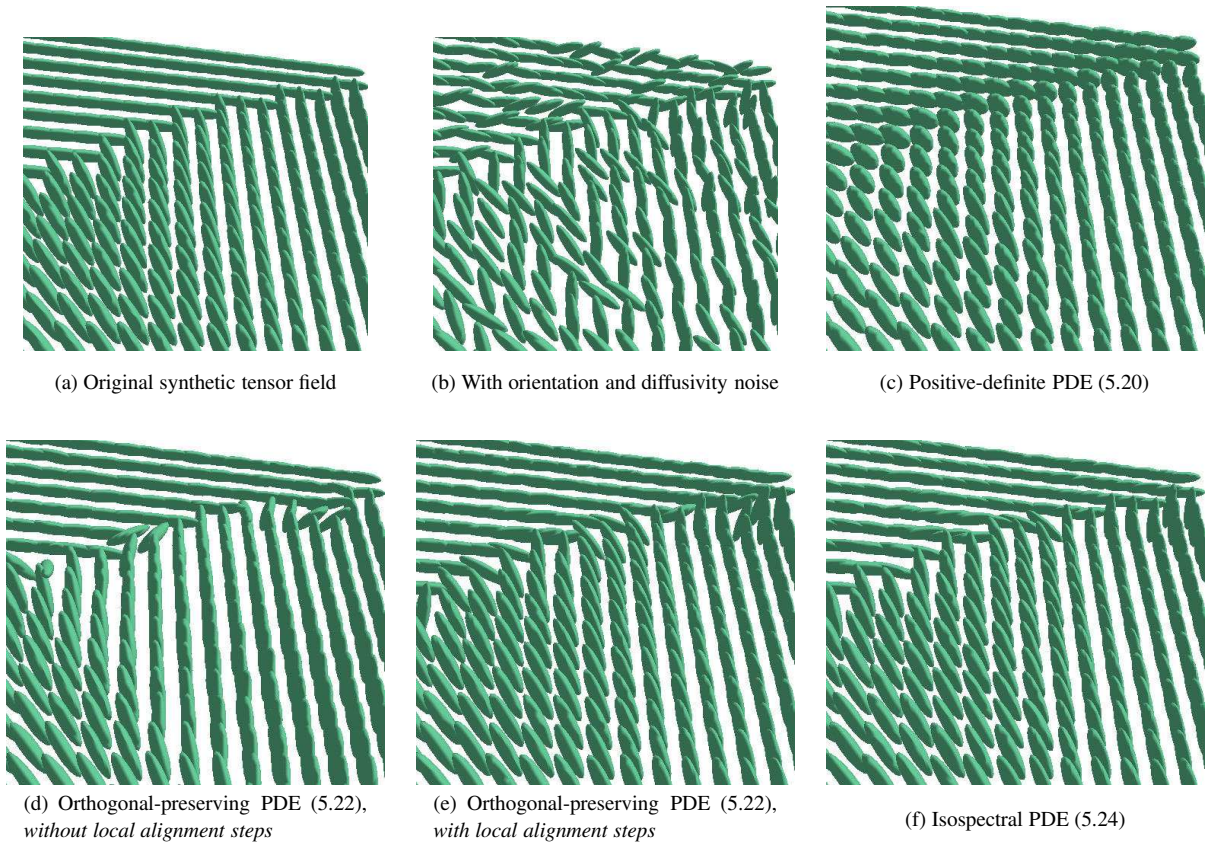


Figure 5.15: Comparison of different approaches for diffusion tensor regularization.

- The orthogonal-preserving flow (Fig.5.15d,e) leads to good denoising results, but it needs a spectral decomposition of the tensor field, as well as computationally expensive re-alignment processes.
- The isospectral flow (Fig.5.15f) gives similar results, but *without any spectral decomposition of the tensor field $\mathbf{T}$* . The high interest of this equation is that it works directly on the tensor coefficients, and doesn't need any realignment process.

In this chapter, we investigated a new and original variational framework that can be used to deal with a large range of constrained data representing orientations : unit vectors, rotation matrices and diffusion tensors. At the same time, we designed efficient numerical schemes, avoiding the classical re-projection problem, inherent to the previously proposed constrained regularization methods. We propose now to apply all these equations, in order to deal with applications of interest.

Si le temps est continu, il y a un paradoxe de la raison.

G. Hermosillo.

Chapter 6

Applications of Constrained PDE's



This chapter illustrates how the orthonormal vector sets framework may deal with various orientation and direction regularization problems. We first handle the problem of unit vector diffusion and propose examples of direction field regularization, as well as chromaticity denoising in color images. Then, we go one step further by applying an orthogonal matrix-preserving flow for two different applications : The regularization of estimated camera motions, allowing smooth reprojections of 3D virtual objects into real movies, and the restoration of noisy Diffusion Tensor MRI images, in order to retrieve more coherent fiber bundles in the white matter of the brain.

6.1 Direction field regularization

Regularization of unit vectors field, also called *direction regularization*, consists in anisotropically smoothing fields $\mathbf{I} : \Omega \rightarrow S^{n-1}$ of *unit vectors* belonging to the unit sphere $S^{n-1} \subset \mathbb{R}^n$, such that the orientation noise disappears first while the important discontinuities of the field are preserved. As we noticed in chapter 5, the direction diffusion is a simple and particular case of our general framework of *orthonormal vector sets regularization*. Considering an orthonormal set reduced to a single vector leads to a regularization equation preserving the *unit norm constraint* :

$$\forall \mathbf{x} = (x, y) \in \Omega, \quad \|\mathbf{I}(\mathbf{x})\| = 1 \quad (6.1)$$

In this section, we illustrate some regularization results obtained with our proposed norm constrained multi-valued diffusion PDE :

$$\begin{cases} \mathbf{I}_{(t=0)} &= \mathbf{I}_{\text{noisy}} \\ \frac{\partial \mathbf{I}}{\partial t} &= \mathcal{L} - (\mathcal{L} \cdot \mathbf{I}) \mathbf{I} \end{cases} \quad (6.2)$$

where we chose $\mathcal{L}$ to be the *unconstrained* vector regularization term, proposed in chapter 3 :

$$\forall i = 1 \dots n, \quad \mathcal{L}_i = \text{trace} \left(\left[\frac{1}{\sqrt{\lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T} + \frac{1}{\lambda_+^* + \lambda_-^*} \theta_+^* \theta_+^{*T} \right] \mathbf{H}_i \right)$$

We applied the norm constrained flow (6.2), in order to deal with two different cases :

- **Direction fields :**

Starting from noisy fields $\mathbf{I}_{\text{noisy}}$ of unit vectors representing directions, we can retrieve more smooth and coherent direction informations. The clear separation between the unit norm constraint (6.1) and the regularization term in the PDE (6.2) allows especially to adapt the regularization behavior to the considered problem (as we did in chapter 4 for image restoration, inpainting, etc.).

First, we illustrate in Fig.6.1, the importance of the norm constraint preservation when dealing with direction fields : The field Fig.6.1b has been obtained from the application of an *unconstrained* regularization PDE such as (4.1), while the Fig.6.1c results from the *norm-constrained* regularization PDE (6.2). These two anisotropic regularization schemes have of course similar comportments and have been carried out with the same set of parameters, starting from a 256×256 direction field and evolving it during 200 PDE iterations.

Note how the vector norms in the unconstrained result are decreasing, and even vanish when the vectors are located on discontinuities. As for the constrained result, the directions are anisotropically denoised without losing the important direction structures or the unitary norm.

We also used the interesting “scale-space” property of our regularizing flows, in order to construct *orientation scale-spaces* of direction fields. This is illustrated on Fig.6.2, with an initially noisy field (size 128×128), where different levels of structure details are obtained with a norm-preserving PDE such as (6.2). The different field structures are smoothly disappearing as the PDE evolution time goes by. Note that contrary to our flow visualization method proposed in chapter 4, we act here directly on the direction field, not in its representation.

- **Chromaticity denoising in color images :**

As described in the section 2.3, many authors proposed to use norm constrained regularizing flows, in order to deal with *chromaticity denoising* in color images [25, 39, 107, 142, 167, 182]. We quickly remind the principle : Each color vector $\mathbf{I} = (R, G, B)$ of a color image can be split into its norm $\|\mathbf{I}\|$ and its unit direction vector $\vec{i}$ (such that $\mathbf{I} = \|\mathbf{I}\| \vec{i}$).

It is a common way to separate the *chromaticity feature* and the *brightness feature* of a color pixel (see Fig.2.15b).

Acting separately on these two different color characteristics allows to adapt the algorithms to several types of noise, in a more precise way than unconstrained schemes. This is illustrated on Fig.6.3 : the two presented noisy images (respective size are 210×196 and

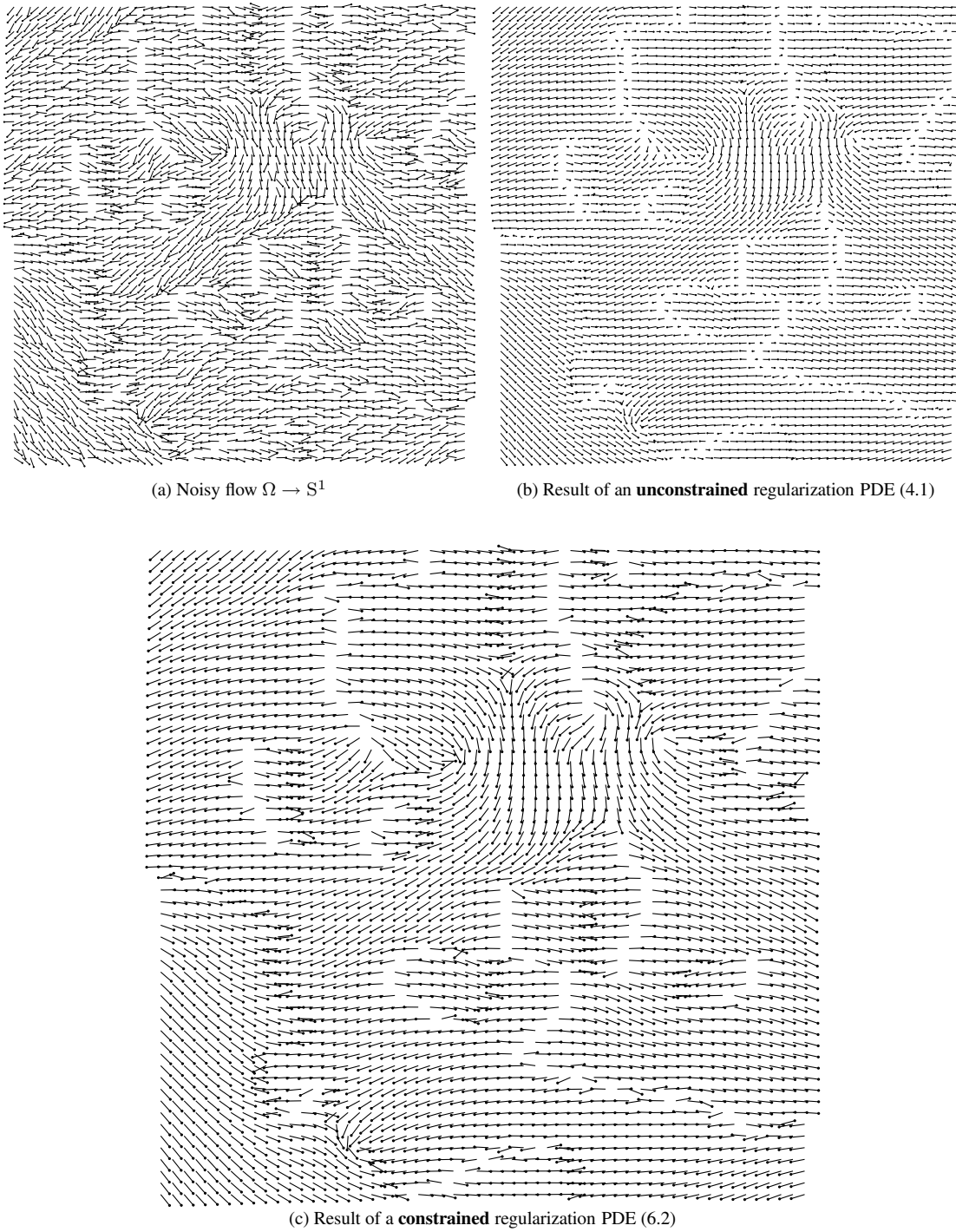


Figure 6.1: Constrained vs. unconstrained PDE's for direction regularization.

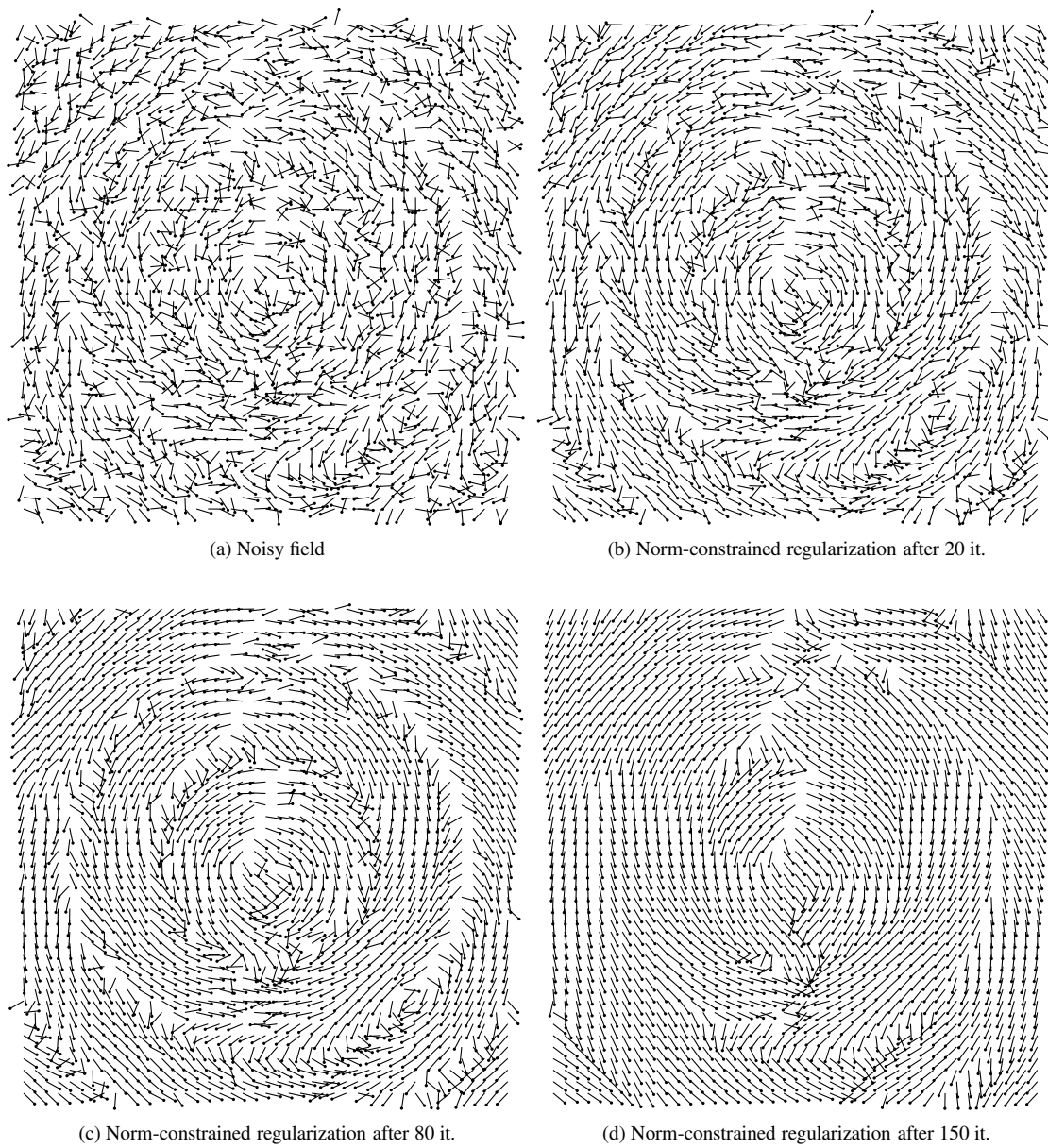


Figure 6.2: Constructing a scale-space of a direction field.



(a) Color images with pure chromaticity noise



(b) Regularized images, with unconstrained PDE's (4.1)

(c) Regularized images, with **norm constrained** PDE's (6.2)

Figure 6.3: Norm constrained regularization for chromaticity denoising in color images.

372×240) have been obtained by adding pure *chromaticity* noise on initially clear images, i.e we added Gaussian noise to the corresponding color direction field $\vec{i}$. We show on Fig.6.3b, the images obtained with unconstrained vector-valued regularization PDE's (4.1), and in Fig.6.3c, the regularized images using the norm-constrained regularization PDE (6.2). Despite the good results obtained by the unconstrained method, we are able to retrieve more detailed images if we act only on the chromaticity part of the color image, with norm constrained PDE's. This is perceptible for instance in the center of the orange and the flower in Fig.6.3 This is not surprising, since the unitary norm constraint is very well adapted to chromaticity noise.

More generally, if we have a prior knowledge about the type of noise that degraded the color image, we can add *constraints* to the regularization PDE's such that it adapts to the model of noise and performs better image restoration.

6.2 Regularization of estimated camera motions

In the computer vision field, a large number of *structure from motion* methods have been proposed in order to estimate video camera motions from real movies [71, 73]. More and more general market products now integrates these kind of algorithms [150].

The idea consists basically in locating correspondence points between video frames, then using them to autocalibrate and estimate the real motion of the video camera.

The output data can be decomposed into two different sequences :

- A temporal sequence $\mathbf{T}(s) : [0, nb] \rightarrow \mathbb{R}^3$ of *translation vectors*, corresponding to the translation motion of the camera, i.e the change of the viewpoint during the video sequence. nb is the total frames number of the input video sequence.
- A temporal sequence $\mathbf{R}(s) : [0, nb] \rightarrow \text{SO}(3)$ of *rotation matrices*, corresponding to the rotation motion of the camera, i.e the change of the view angle during the video sequence.

Estimating correspondence points is a very sensitive process which often reacts to image noise, and the resulting estimated temporal sequences $\mathbf{T}$ and $\mathbf{R}$ may be noisy. Defining a regularization algorithm adapted to these noisy data could be then sometimes needed.

For instance, this is the case if one wants to reproject 3D virtual objects on real movies, using the estimated camera motion information (this new technique attracts a growing interest in the domain of special effects for movies and advertising). Reprojecting a 3D object using a noisy estimated camera motion will result in non-realistic *shaking* effects.

Here, we propose to define a simple regularization scheme that solves this problem (Fig.6.4) :

- The translation part $\mathbf{T}$ of this camera motion can be easily restored, since it is equivalent to a classical unconstrained vector-valued 1D image, which can be regularized with adapted PDE's, as defined in chapter 3

$$\frac{\partial T_i}{\partial t} = \text{trace} \left(\left[\frac{1}{\sqrt{\lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T} + \frac{1}{\lambda_+^* + \lambda_-^*} \theta_+^* \theta_+^{*T} \right] \mathbf{H}_i \right)$$

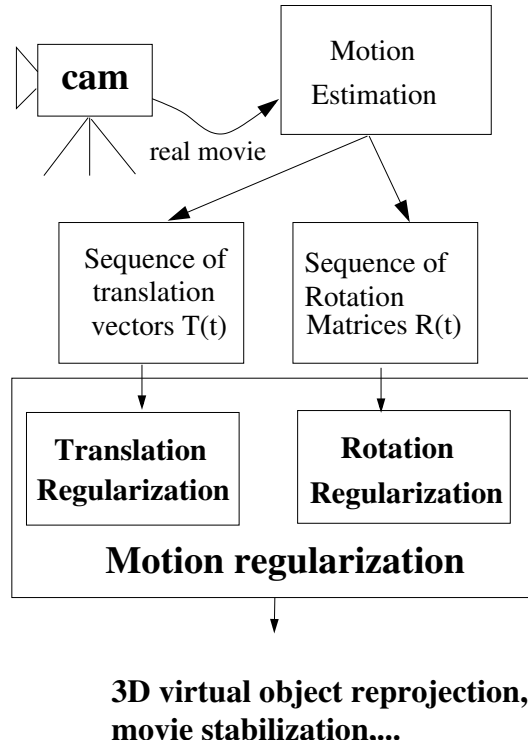


Figure 6.4: Principle of our camera motion regularization.

where $\mathbf{H}_i$ is the Hessian matrix of the vector channel T_i , while $\lambda_{\pm}^*$ and $\theta_{\pm}^*$ are respectively the eigenvectors and eigenvalues of the smoothed structure tensor $\mathbf{G}_{\sigma} = \left(\sum_{j=1}^n \nabla T_i \nabla T_i^T \right) * G_{\sigma}$.

Note that we use an *anisotropic* regularization here, since the translation motion of the camera may contains discontinuities.

- The rotation part $\mathbf{R}$ is handled by our orthogonal-matrix preserving PDE, defined in chapter 5 :

$$\frac{\partial \mathbf{R}}{\partial t} = \mathcal{L} - \mathbf{R} \mathcal{L}^T \mathbf{R} \quad (6.3)$$

where $\mathcal{L} \in \mathbb{R}^{3 \times 3}$ is an unconstrained regularizing matrix term. We chose :

$$\mathcal{L}_{ij} = \text{trace}(\mathbf{Id} \mathbf{H}_{ij}) = \Delta R_{ij}$$

(the $\mathbf{H}_{ij}$ are the different Hessians of the scalar images R_{ij}).

Note that we perform *isotropic* regularization here, since the rotation part of the camera should not contain discontinuities to be preserved (unlike the translation part).

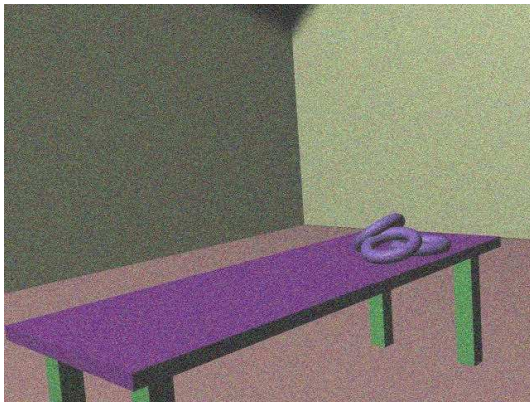
We illustrate this two-step regularization scheme with the reprojection of a 3D virtual teapot in the scene of a synthetic movie (Fig.6.5). This movie has been generated in a way that the original

camera motion is quite soft : of course, the smoothness hypothesis about the real camera motion is a necessary prerequisite for the algorithm.

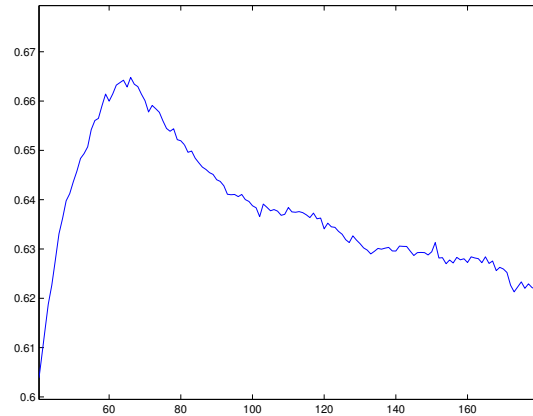
The figure is ordered as follows : the second column Fig.6.5b,d,f represents one of the three Euler angles of the estimated and regularizing motions. *Note that we didn't regularize the Euler-angles themselves*, but acted only on the coefficients of the rotation matrices $\mathbf{R}(s)$, for reasons explained in section 5.3.2. The first column Fig.6.5a,c,e respectively illustrates : (a) a frame of the original movie, (c) a frame with a teapot inserted therein, and finally (e) a superposition of the teapots corresponding to the estimated and regularized motion.

This figure 6.5 can hardly illustrate the difference between the 3D motions of the inserted virtual 3D object, in the two obtained video sequences (using the original and the restored camera motion estimation). Nevertheless, there is a clear *shaking effect* on the virtual teapot, when using the original estimation. This undesired effect is removed when using the regularized motion. Please see the presentation containing the corresponding MPEG files at :

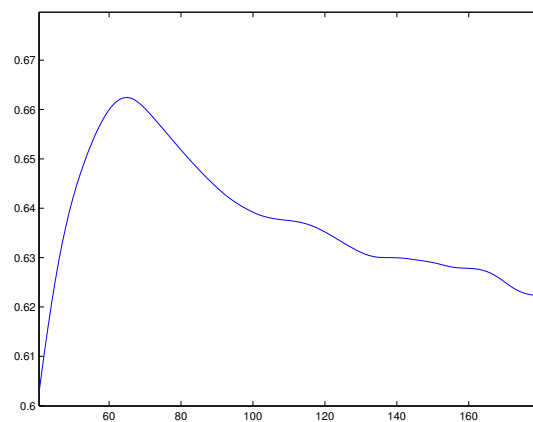
[http : //www - sop.inria.fr/robotvis/personnel/David.Tschumperle](http://www-sop.inria.fr/robotvis/personnel/David.Tschumperle)



(a) Synthetic movie with noisy frames

(b) Euler angle X of the estimated rotation motion $\mathbf{R}$ 

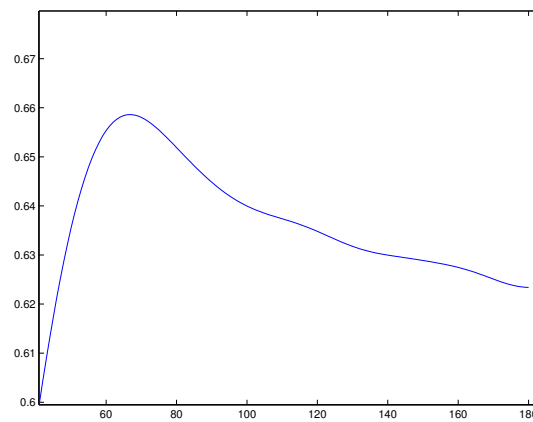
(c) Insertion of a virtual 3D teapot



(d) Euler angle X after motion regularization (10 it. of (6.3))



(e) Projections difference between regularized and original motion



(f) Euler angle X after motion regularization (20 it. of (6.3))

Figure 6.5: Regularization of video camera motions for 3D virtual object insertion.

6.3 DT-MRI image regularization

DT-MRI (Diffusion tensor magnetic resonance imaging) is a recent and non-invasive 3D medical image modality consisting in measuring the water molecule motions in the white matter tissues, using magnetic resonance techniques. We note $\mathbf{x} = (x, y, z)$. Each voxel $\mathbf{T}(\mathbf{x})$ of the acquired image $\mathbf{T}$.

$$\mathbf{T} : \Omega \rightarrow \mathcal{P}(3)$$

is a symmetric and positive definite 3×3 matrix that defines the local fiber structure of the tissues, as described in [89, 115, 145, 181] and section 1.2.4.

Here, we propose some results illustrating the high interest of DT-MRI regularization schemes, in order to compute fiber bundles in the white matter of the brain.

In the figure presented below, the representation of the DT-MRI tensors fields is done as follows :

- The left part of the images represents the diffusion tensors as ellipsoids, whose orientations and radiuses are respectively given by the eigenvectors and eigenvalues of the corresponding matrices $\mathbf{T}(\mathbf{x})$.
- The right part represents the estimated fibers, computed at each voxel of the volume $\mathbf{T}$ as the lines following the main eigenvector of the tensor $\mathbf{T}(\mathbf{x})$. Actually, it corresponds to a representation of the main fiber structures in the white matter of the brain.

The used regularization algorithms are the ones presented in the previous chapter 5, allowing to construct a scale-space of the brain fibers. It consists in a two-step algorithm :

- Regularization of the eigenvalues (tensor diffusivities), with classical unconstrained vector-valued diffusion PDE's, such as :

$$\frac{\partial \lambda_i}{\partial t} = \text{trace}(\mathbf{D} \mathbf{H}_i)$$

where $\mathbf{D}$ is an anisotropic diffusion tensor, conducting the regularization, as defined in chapter 3, and $\mathbf{H}_i$ is the Hessian of the eigenvalue λ_i .

- Regularization of the tensor orientations (an orthogonal matrix), with our rotation-constrained diffusion PDE :

$$\frac{\partial \mathbf{R}}{\partial t} = \mathcal{L} - \mathbf{R} \mathcal{L}^T \mathbf{R}$$

or, using the proposed *isospectral flow* directly on the tensor field $\mathbf{T}$:

$$\frac{\partial \mathbf{T}}{\partial t} = [\mathbf{T}, [\mathbf{T}, -(\mathcal{L} + \mathcal{L}^T)]] .$$

Please refer to the section 5.6 for the definition of the regularization term $\mathcal{L}$.

Fig.6.6 and Fig.6.7 show the application of our different diffusion tensor regularization scheme, for constructing smooth tissues fiber map in the white matter of the brain. We regularize a real $128 \times 128 \times 56$ DT-MRI dataset (courtesy of CEA-SHFJ [36])¹ Then, we follow at each voxel

¹We would like to thank J.-F. Mangin and J.-B. Poline (SHFJ-CEA) for providing us with the DT-MRI data (this work was partially supported by ARC MC2). We also thank R. Fournier for his visualization tool "TensView".

of the volume the main tensor directions which are representative of the fibers structures. Two regularization steps are shown (Fig.6.6.d,e). For each sub-figure, the tensor field is represented by ellipsoids on the left part of the image, and by the computed streamlines on the right. Note how the computed fibers are smoothed along the PDE flow (5.13). Here, the gradient functional may depend on physiological attributes, as for instance functions proposed in [60, 61].

The computations have been carried out with a Pentium 1Ghz processor, 1Gigabyte RAM, and the corresponding computation times are given for the presented sub-volume restoration process in the captions of Fig.6.7.

We noticed that the geodesic step approach allows time steps to be relatively high, and very few iterations are needed in practice. The results we obtained clearly illustrates the behavior of the different methods:

- The symmetric positive definite flow (5.20) tends to swell eigenvalues.
- The orthogonal constrained flow (5.22) works well with a local alignment step (Fig. 6.7e), but fails otherwise (Fig. 6.7d).
- The isospectral flow (5.24) has a quite similar behavior, but is more computationally efficient.

Even if the physiological validation of these results remains to be done, our methods seem to correct most artifacts (due to the image acquisition process) and retrieve the main global structures of the fiber network. It opens new perspectives to construct an accurate fiber map model of the brain.

Other interesting works on the particular problem of DT-MRI regularization have been proposed in the literature.

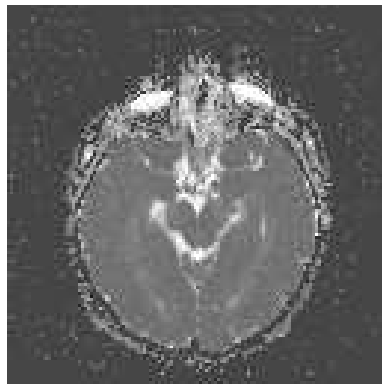
We can particularly mention the work of Coulon-etal [60, 61], who use a norm constrained PDE to regularize the principal direction $\mathbf{u}_1$ of the tensors, then compute the corresponding regularized tensors by projection the two other directions $\mathbf{u}_2$ and $\mathbf{u}_3$ into the orthogonal plane to the regularized $\mathbf{u}_1$. This is indeed a particular case of our rotation-preserving regularization : we can easily minimize a functional depending only on the main direction $\mathbf{u}_1$ when regularizing the orientation part of the tensor (rotation matrices). Anyway, our method doesn't require any reprojection step, thank to the use of specific numerical schemes.

Another interesting approach can be found in [181]. Here, the regularization is performed on the initial raw DT-MRI data, instead of doing it on the computed diffusion tensors. This obviously removes the problem of constrained regularization, but one has to define properly the coupling between all these raw images for the regularization process.

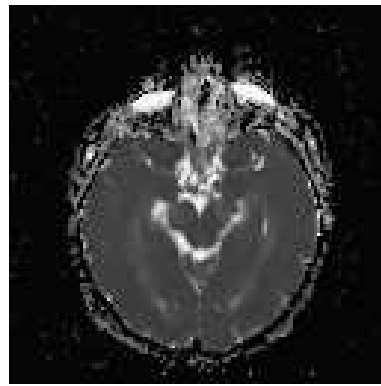
Other works of interest on this emerging issue can be found in [48, 146, 175].

Pas mal !

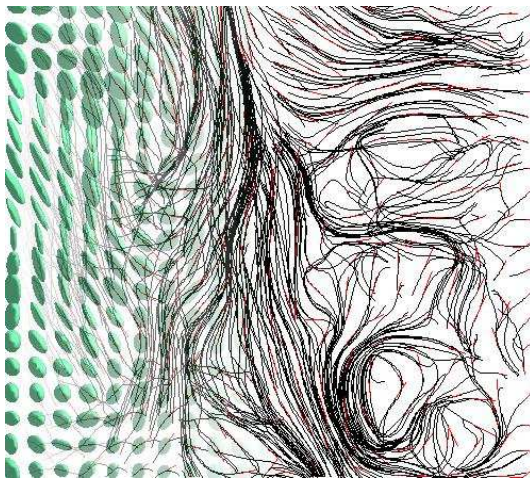
N.



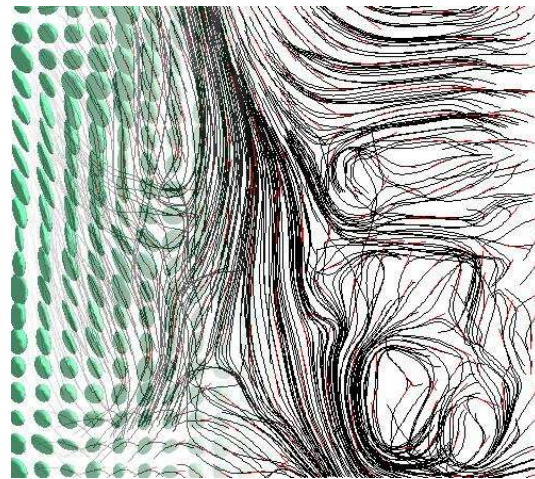
(a) Eigenvalues of an original DT-MRI volume



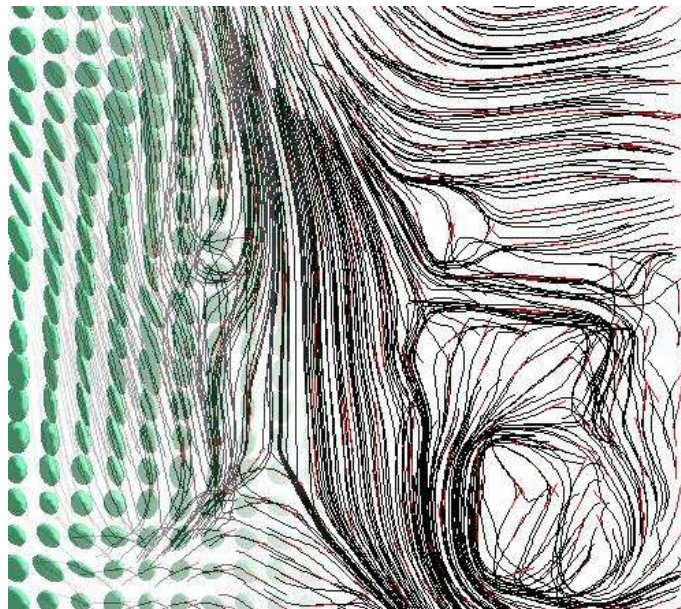
(b) Regularized eigenvalues with eq.(4.1)



(c) Initial tensors/fibers of a part of (a)



(d) Tensors/fibers after 10 it. of (5.13)



(e) Tensors/fibers after 30 it. of (5.13)

Figure 6.6: Constructing a structure preserving scale-space of the white matter fibers.

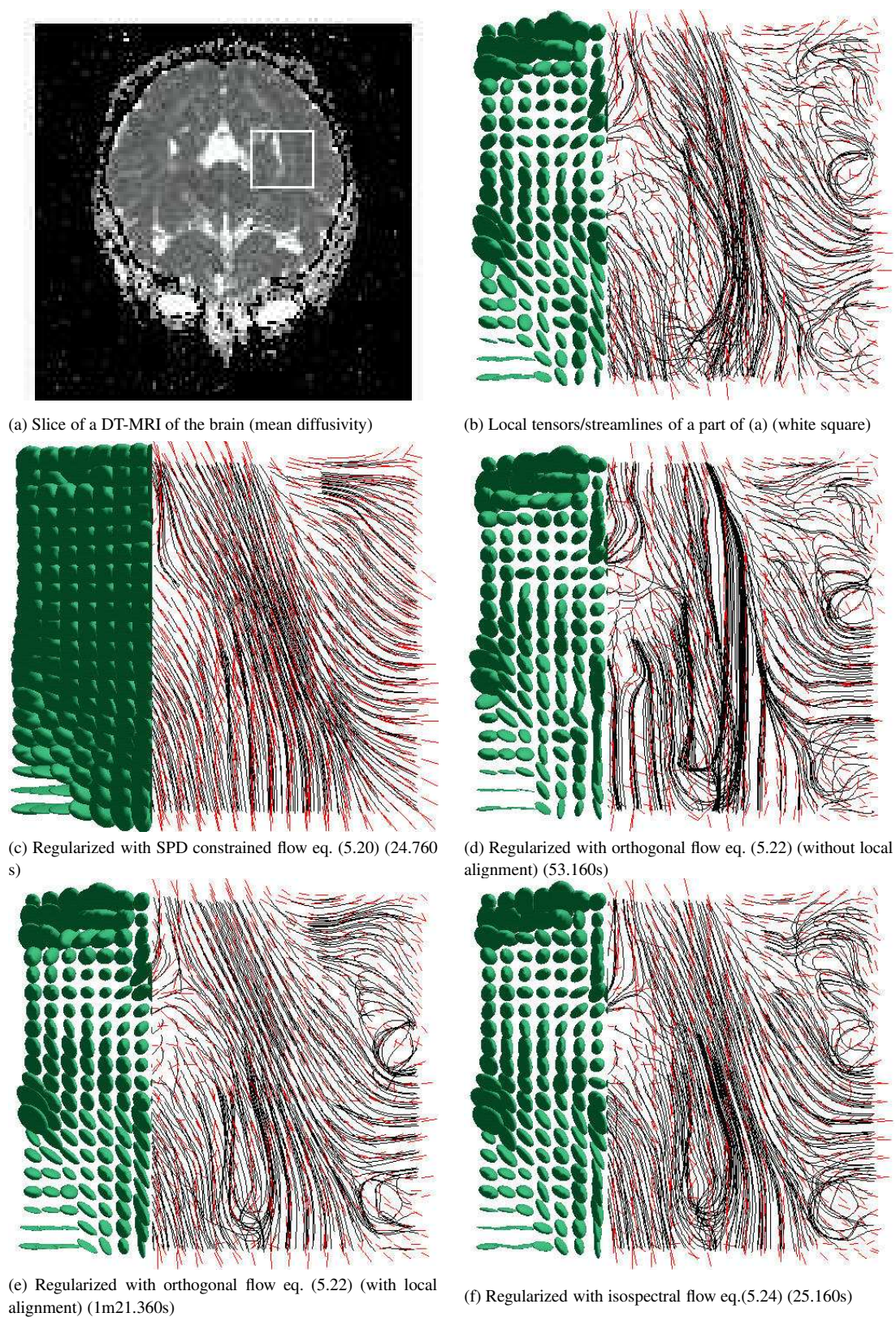


Figure 6.7: Comparison of 3 different methods for DT-MRI dataset regularization.

Appendix : How to implement vector-valued regularization PDE's ?



In this appendix, we detail some of the important part of the source code (using the C language) used to implement our proposed theoretical framework of unconstrained diffusion PDE's. This should help the interested reader to perform such vector-valued regularization flows in its own image processing algorithms. At first glance, diffusion PDE's and variational methods are often based on complex mathematical theories and seem hard to implement under a discrete form. Here, we want to underline simple numerical schemes and code tricks that eases the implementation step, speeds up the computation time and improves the quality of the obtained results.

This section is related to the particular topic of unconstrained PDE's, tackled in chapter 3. Here, we propose to analyze the simple implementation (in C) of the following vector-valued regularization equation :

$$\forall i = 1..n, \quad \frac{\partial I_i}{\partial t} = \text{trace} \left(\left[\frac{1}{\sqrt{1 + \lambda_+^* + \lambda_-^*}} \theta_-^* \theta_-^{*T} + \frac{1}{1 + \lambda_+^* + \lambda_-^*} \theta_+^* \theta_+^{*T} \right] \mathbf{H}_i \right) \quad (6.4)$$

Actually, its implementation is *very simple*. The PDE (6.4) is the one that has been used to obtain all the regularization results in chapter 4. In the proposed source code, we implement two different numerical schemes, as described in section 3.5 :

1. *Using the spatial discretization of the Hessian.* This is the faster method, equivalent to classical PDE discretization schemes using finite differences.
2. *Using local convolutions.* This is the slower but precise method we proposed in chapter 3. It consists in applying a spatially-adapting convolution kernel into the image, resulting in a highly non-linear image filtering.

Before all, we assume that the reader has at one's disposal libraries containing basic structures and corresponding processing functions allowing to manage these two different kind of objects :

1. **Images**, as 2D or 3D vector fields $\Omega \rightarrow \mathbb{R}^n$. Related library functions are image loading, gaussian blurring, etc. In our source code, we named this library `CImg` and the library function names are prefixed by the string `'cimg_'`.
2. **Matrices**. In our source code, a matrice $n \times n$ is represented with a simple *1D array* `matrix[n*n]`, where the coefficients are stored columns by columns (with a Matlab style). The related library function names are prefixed by `'m2x2_'` or `'m5x5_'`.

Here is the exact source code, used for our color image regularization experiments, using unconstrained vector-valued regularization PDE's (chapter 4). We insert comments when necessary, *after* the concerned source lines.

```
/*-----
mydiff :

Implementation of the multivalued diffusion PDE's

dI_i/dt = Trace(DH_i)      (i=1..n)

with two different numerical schemes.

(1) Using Hessian discretization or
(2) Using local convolutions

-----*/

#include "CImg.h"           // Image library with basic functions
#include "mlib.h"           // Matrix library with basic functions

float dt;                  // time step used for the PDE evolution
float zero;                // value of minimal eigenvalue
int msize;                 // size of the convolution mask

/*-----
Compute the gaussian kernel corresponding to a tensor D
-----*/
void get_gauss_mask(float D[4], float mask[25]) {
    static float S[2], V[4], u, v, n, s1, s2;
    static int i, j;
    t2x2_eigen(D, S, V); // spectral decomposition of the tensor D
    if (S[0] < zero) S[0] = zero;
    if (S[1] < zero) S[1] = zero;
    u = V[0];
    v = V[1];
    n = 0;
    mx1_zero(mask, 5, 5);
    for (j = -msize/2; j <= msize/2; j++)
        for (i = -msize/2; i <= msize/2; i++) {
            s1 = u*i + v*j;
            s2 = -v*i + u*j;
            s1 *= s1;
            s2 *= s2;
            n += (mask[5*(i+2) + (j+2)] = 1.0 / (2*M_PI*dt) * exp(-(s1/S[0] + s2/S[1]) / (4*dt)));
        }
    if (n > cimg_zero) for (i=0; i<25; i++) mask[i] /= n; // normalization of the convolution mask
}
```

The function `get_gauss_mask` computes an oriented 2D Gaussian kernel 3×3 or 5×5 (depending of the value of `msize`). This mask is oriented by a 2×2 diffusion tensor $\mathbf{D} = \mathbf{D}[4]$ given as an input parameter. Actually, the $\mathbf{D}$ corresponds to the tensors inside the trace operator in the equation (6.4). Computing this mask is the main time-consuming procedure of the numerical scheme based on local filtering, since it evaluates a lot of exponential functions. When using the other numerical scheme, this function is not used.

```
/*-----
Main procedure
-----*/
```

```

int main(int argc, char **argv) {
    CImg itmp, src, dest, velocity, structure, diffusion;
    CImgI hessian, gradient, visu;
    CImgDisplay disp;
    CImgStats stats;
    float xdt=0, lambda_1, lambda_2, ix, iy, G[4], H[4], eig_values[2], eig_vectors[4], I[25], kernel[25];
    int iteration, x, y, k, px, py, nx, ny, ax, ay, bx, by;

    // Read and check command line parameters
    cimg_usage("Implementation of Vector Diffusion PDE such as dI_i/dt = Trace(DH_i)");
    char *file_i = cimg_hparam("-i", NULL, "Input image");
    char *file_o = cimg_hparam("-o", NULL, "Output image");
    float noise = cimg_hparam_float("-noise", 0, "Add gaussian noise");
    float smooth = cimg_hparam_float("-smooth", 0.7, "Smooth diffusion tensors D and J");
    int nb_iter = cimg_hparam_int("-iter", 16000000, "Number of iterations");
    int save_iter = cimg_hparam_int("-save", 0, "Iteration sequence saving step");
    char statflag = cimg_hparam_bool("-stats", 0, "Display image statistics");
    int scheme = cimg_hparam_int("-scheme", 0, "Used numerical scheme (0=classic, 1=convolutions)");
    switch(scheme) {
    case 0:
        dt = cimg_hparam_float("-dt", 15, "Adaptative time step ratio");
        break;
    default:
        dt = cimg_hparam_float("-dt", 3, "Adaptative time step ratio");
        zero = cimg_hparam_float("-zero", 0.01, "Zero eigenvalue");
        msize = cimg_hparam_int("-msize", 5, "Size of the convolution mask (3 or 5)");
        xdt = dt;
        break;
    }
    cimg_err(!file_i, "Option -file_i is not specified");
    cimg_err(noise<0, "Option -noise = %g < 0", noise);
    cimg_err(smooth<0, "Option -smoothing = %g < 0", smooth);
    cimg_err(nb_iter<0, "Option -nb_iter = %d < 0", nb_iter);
    cimg_err(save_iter<0, "Option -save = %d < 0", save_iter);
    if (!save_iter) save_iter = nb_iter;

    // Data initialisation
    src = cimg_load(file_i); // define the source image I_0
    if (noise>0) cimg_add_gauss_noise(src, noise); // add synthetic gaussian noise if required
    dest = cimg_copy(src); // define the iterated image I(t)
    velocity = cimg_same(src); // define the PDE velocity
    structure = cimg_new(src->width, src->height, 4); // define the structure tensor field G
    diffusion = cimg_new(src->width, src->height, 3); // define the diffusion tensor field D
    hessian = cimgl_new_all(3, src->width, src->height, src->dim); // define the Hessians H_i (i=1..n)
    gradient = cimgl_new_all(2, src->width, src->height, src->dim); // define the gradient nabla I_i (i=1..n)

    if (cimg_max(src->width, src->height)<400) visu = cimgl_new_from(4, src, dest, structure, velocity);
    else visu = cimgl_new_from(1, dest);
    disp = cimgl_new_display(visu, cimg_fbasename(argv[0]), 1);
    cimg_stats(src, "Original image");
}

```

In the above lines, we read the input (noisy) image `src` and the variance parameter $\sigma = \text{smooth}$ for the smoothing of the structure tensor field $G_\sigma = G * G_\sigma$. We also initialize the needed images for the PDE flow computation : gradient image `gradient`, Hessian image `hessian`, iterated image `dest`, PDE velocity field `velocity`, structure tensor image `structure`, and diffusion tensor image `diffusion`. We can now start the main PDE iteration loop :

```

//-----
// PDE Iteration loop
//-----
for (iteration=1; iteration<=nb_iter; iteration++) {
    printf("\rIt. %d : dt = %g ", iteration, xdt); fflush(stdout);

    // Compute the Gradient and Hessian of each image channel I_i
    cimg_mapV(dest, k) cimg_m3x3map(dest, x, y, 0, k, I) {
        cimg_pix(gradient->pos[0], x, y, k) = 0.5*(I[7]-I[1]);
        cimg_pix(gradient->pos[1], x, y, k) = 0.5*(I[5]-I[3]);
        cimg_pix(hessian->pos[0], x, y, k) = I[7]+I[1]-2*I[4];
        cimg_pix(hessian->pos[1], x, y, k) = 0.25*(I[0]+I[8]-I[6]-I[2]);
        cimg_pix(hessian->pos[2], x, y, k) = I[5]+I[3]-2*I[4];
    }
}

```

The gradients and the Hessians of the images $I_i = \text{dest}(x, y, i)$ are computed with classical central finite differences. The trick here comes from the particular spatial loop `cimg_m3x3map` that makes a 3×3 mask `I[9]` (a matrix) cover the image `dest` with the indices `x` and `y`. For

each (x, y) , the mask $I[9]$ always corresponds to the 3×3 local neighborhood of the current image point $\text{dest}(x, y)$ (Fig.6.8). Note that the code of `cimg_m3x3map` is written in the `CImg` library as an *optimized macro*.

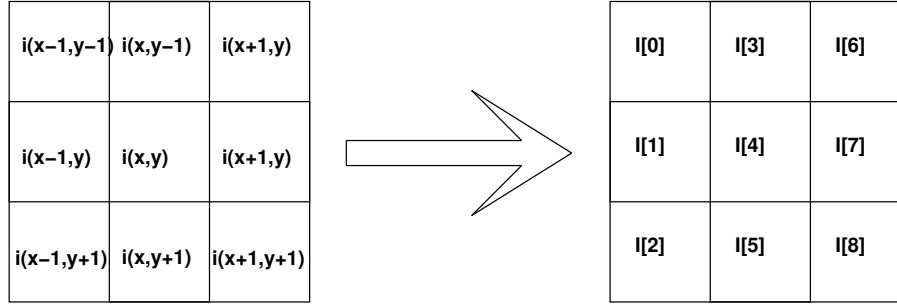


Figure 6.8: Principle of the loop `cimg_m3x3map`.

Knowing these gradients ∇I_i , we are now able to compute the field of the smoothed structure tensors $\mathbf{G}_\sigma$:

```
// Compute the smoothed structure Tensor G_sigma
cimg_mapXY(dest,x,y) { // compute the coefficients of each matrix G
    m2x2_zero(G);
    cimg_mapV(dest,k) {
        ix = cimg_pix(gradient->pos[0],x,y,k);
        iy = cimg_pix(gradient->pos[1],x,y,k);
        G[0] += ix*ix;
        G[1] += ix*iy;
        G[3] += iy*iy;
    }
    cimg_3set(structure,x,y,0,G[0],G[1],G[3]);
}
if (smooth>cimg_zero) { // smooth the tensor field G by a gaussian kernel
    itmp = cimg_deriche_blur(structure,smooth,1);
    cimg_swap(itmp,structure);
    cimg_free(itmp);
    visu->pos[2]=structure;
}
```

Note that the structure tensor field $\mathbf{G} = \text{structure}(x, y)$ is smoothed with a *Canny-Deriché filter* [62, 63, 64], in order to retrieve a more coherent local vector geometry of the structures in the processed image.

```
// Compute the spectral elements of G_sigma, and the diffusion tensor D
cimg_mapXY(dest,x,y) {
    cimg_3get(structure,x,y,0,G[0],G[1],G[3]); G[2] = G[1];
    t2x2_eigen(G,eig_values,eig_vectors);
    if (eig_values[0]<0) eig_values[0]=0; // positive constraints needed due to numerical
    if (eig_values[1]<0) eig_values[1]=0; // errors during eigenvalue decomposition
    cimg_4set(structure,x,y,0,eig_values[0],eig_values[1],eig_vectors[0],eig_vectors[1]);

    // define the diffusion tensor D, from its eigenvectors/eigenvalues
    lambda_1 = 1.0/sqrt(eig_values[0]+eig_values[1]+1); // eigenvalue of D, in the direction theta_- = eig_vectors[0]
    lambda_2 = 1.0/(eig_values[0]+eig_values[1]+1); // eigenvalue of D, in the direction theta_+ = eig_vectors[1]
    cimg_3set(diffusion,x,y,0,
        lambda_1*eig_vectors[0]*eig_vectors[0] + lambda_2*eig_vectors[1]*eig_vectors[1],
        (lambda_1 - lambda_2)*eig_vectors[0]*eig_vectors[1],
        lambda_1*eig_vectors[1]*eig_vectors[1] + lambda_2*eig_vectors[0]*eig_vectors[0]
    );
}
```

This part computes the diffusion tensor field $\mathbf{D}$ used in the trace operator of the PDE flow (6.4), from the eigenvalues and eigenvectors of the smoothed structure tensor $\mathbf{G}_\sigma$.

```

// Compute the PDE velocity (with 2 possible numerical schemes)
switch(scheme) {
case 0: // Using the discretization of the Hessian
    cimg_mapXY(dest,x,y) {
        cimg_3get(diffusion,x,y,0,G[0],G[1],G[3]); G[2]=G[1];
        cimg_mapV(dest,k) { // compute Trace(DH_i) for each image channel I_i
            H[0] = cimg_pix(hessian->pos[0],x,y,k);
            H[1] = H[2] = cimg_pix(hessian->pos[1],x,y,k);
            H[3] = cimg_pix(hessian->pos[2],x,y,k);
            m2x2_mul(G,H);
            cimg_pix(velocity,x,y,k) = m2x2_trace(H);
        }
    }
    stats = cimg_get_stats(velocity,0); // Compute the adaptative time step
    if (stats.max == stats.min) xdt = 0; else xdt = dt/cimg_max(fabs(stats.max),fabs(stats.min));
    cimg_mapXYV(dest,x,y,k) cimg_pix(dest,x,y,k) += xdt*cimg_pix(velocity,x,y,k);
    break;
}

```

This describes the first possible numerical scheme, using the spatial discretization of the Hessian. As you may notice, the scheme is very direct and simple to implement. Moreover, we use an adaptative time step xdt allowing a maximum intensity variation dt of the image pixel values (usually $dt \in [10, 20]$). xdt is then used to update the iterated image `dest`.

```

default: // Using the local convolutions by oriented 2D gaussian kernels
    cimg_5mapXY(dest,x,y) {
        cimg_3get(diffusion,x,y,0,G[0],G[1],G[3]); G[2]=G[1];
        get_gauss_mask(G,kernel);
        cimg_mapV(dest,k) { // do the convolution for each image channel I_i
            cimg_m5x5get(dest,x,y,0,k,I);
            cimg_pix(dest,x,y,k) = cimg_m5x5dot(I,kernel);
            cimg_pix(velocity,x,y,k) = cimg_pix(dest,x,y,k)-I[12];
        }
    }
    break;
}

```

This is the second possible numerical scheme, using spatially-adapting local convolutions. Here, the convolutions are facilitated thanks to the special loop `cimg_m5x5map`, allowing an immediate access to the local neighborhood of the current pixel. This is basically the same macro as `cimg_m3x3map` (Fig.6.8), for a 5×5 local neighborhood.

```

// Prepare next iteration
if (file_o && !(iteration%save_iter)) {
    char tmpstring[1024];
    if (save_iter!=nb_iter) cimg_fnumber(file_o,iteration,5,tmpstring); else strcpy(tmpstring,file_o);
    cimg_save(dest,tmpstring);
}
if (statflag) cimg_stats(dest,"Image");
cimgl_refresh_display(visu,disp);
}

printf("> Done !\n");
exit(0);
return 0;
}

```

The iterations are saved during the PDE flow, in order to construct the entire scale-space of the image. The algorithm stops after a user-defined finite number of iterations.

Note that this source code is the one used for color image restoration experiments. We also point out that some adaptations are needed to deal with image inpainting, magnification and flow visualization, but they are quite simple and won't be detailed here. Nevertheless, this appendix pointed out the relative easiness of the proposed unconstrained algorithm.

□

Conclusion & Perspectives

Contributions :

In this document, we studied multivalued image regularization, which is a low-level process used in many image processing algorithms. We focused on variational and PDE-based approaches, acknowledged for many years as very efficient methods.

First of all, this thesis presented the present state of the art of this large domain, by comparing and analysing equations already proposed in the literature, through a local geometric viewpoint. Then, we extended these methods by proposing more general and unifying frameworks. On one hand, it gathered previous existing schemes. On the other hand, it allowed to design new regularization PDE's, more adapted to the local geometry of considered images.

Our main contributions take place in these two complementary fields :

- Unconstrained multivalued image regularization : We proposed a new formalism allowing to express previous approaches on unconstrained multivalued regularization within a common expression. This formulation is more adapted to the local understanding of the smoothing performed by these diffusion PDE's. Moreover, we defined a new regularization equation, based on the respect of a coherent vector image geometry, as well as numerical schemes adapted for the implementation of the whole range of multivalued regularization equations. The application to several problems related to color images illustrated the efficiency of our methods to deal with concrete cases.
- Constrained multivalued image regularization : We proposed a new constrained regularization approach, allowing to deal with a wide variety of direction and orientation features. This has been done by considering the preservation of specific orthonormal constraints during PDE regularizing flows. Thus, this new formalism naturally extended previous works on the diffusion of unit vector fields. But it also opened new ways of regularizing more complex orientation features, such as rotation matrices and diffusion tensors. It significantly enlarged the possible application field handled by constrained regularization algorithms. We illustrated it with two different problems of interest : estimated camera motion regularization, allowing realistic 3D virtual object insertions in real video sequences. Then, the reconstruction of more coherent fiber networks in the white matter of the brain, via DT-MRI images. We also proposed numerical schemes, well adapted to the preservation of orthonormal constraints.

Throughout this document, we tried hard to follow this scientific reasoning :

- First, the *study* and the *implementation* of existing methods.
- The *local geometric understanding* of these methods.
- The *generalization* in a common framework, leading to a formalization of new and more efficient equations.
- The *design of specific numerical schemes*, adapted for this kind of equations.
- The *application* to concrete problems, in order to illustrate the practical issues of our approaches.

Perspectives :

This thesis opened new questions that could be envisaged for future works. First, the aspects of convergence and uniqueness of our proposed schemes : our new unifying expressions could be adapted to study these important theoretical aspects.

We also want to validate the results obtained with DT-MRI image regularization, with the help of specialists. For instance, this may define physiological priors that could be easily incorporated into our regularization PDE's, leading to physiologically plausible models of white matter fibers. Finally, our new regularization equation could be integrated into other classical computer vision problems expressed with variational methods, such as optical flow, segmentation, etc. This could improve these algorithms thanks to the use of a specific local geometry adapted to the considered problem.

Conclusion & Perspectives

Contributions :

Dans ce document, nous avons étudié la régularisation d'images multivaluées, qui est un processus de base utilisé dans de nombreux algorithmes de traitement d'images. Nous nous sommes focalisés sur les formulations variationnelles et EDP, qui sont depuis plusieurs années des méthodes reconnues pour leur efficacité.

Cette thèse a d'abord présenté un état de l'art du domaine, en comparant et analysant par une approche géométrique locale, les équations déjà proposées dans la littérature. Puis nous avons étendu ces méthodes en proposant à chaque fois, un cadre plus général permettant à la fois d'unifier les approches précédentes, mais aussi de considérer de nouvelles équations adaptées à la géométrie locale des problèmes considérés.

Nos contributions principales se situent dans les deux domaines suivants :

- Régularisation *non-contrainte* d'images multivaluées : Nous avons proposé une nouvelle équation qui a permis d'exprimer sous une forme unificatrice, les précédents travaux portant sur la régularisation d'images multivaluées non contraintes. Cette formulation est mieux adaptée à la compréhension locale des processus de lissage effectués par les EDP de diffusion multivaluées. De plus, nous avons défini une nouvelle méthode de régularisation, basée sur le respect de propriétés géométriques intéressantes, ainsi que des schémas numériques adaptés pour implémenter toutes ces équations. L'application à plusieurs problèmes liés aux images couleurs a montré l'efficacité de nos méthodes de régularisation pour traiter des cas concrets.
- Régularisation *contrainte* d'images multivaluées : Nous avons proposés, une nouvelle approche de régularisation contrainte, qui permet de traiter une large variété de données de directions et d'orientations. Ceci a été possible en considérant la préservation de contraintes orthonormales dans l'évolution des EDP de régularisation. Ainsi, ce nouveau formalisme permet d'étendre les précédents travaux sur la régularisation de vecteurs unitaires, mais aussi de traiter des données d'orientations plus complexes, comme les matrices de rotation, ou les tenseurs de diffusion. Ceci a ouvert de façon significative le champ d'application possible de la régularisation d'orientations, et nous l'avons illustré avec deux problèmes concrets importants : la régularisation de mouvements estimés de caméra pour l'insertion réaliste d'objets virtuels 3D dans des séquences vidéos, et la reconstruction de réseaux de

fibres plus cohérents dans la matière blanche du cerveau via l'imagerie IRMd. Nous avons également proposé des schémas numériques bien adaptés à la préservation des contraintes orthonormales.

Tout au long de ce document, nous nous sommes donc efforcés de suivre la démarche scientifique suivante :

- D'abord l'*étude*, la *comparaison*, et l'*implémentation* des méthodes existantes.
- La *compréhension géométrique locale* de ces méthodes.
- La *généralisation* dans un cadre commun, amenant à la formalisation de nouvelles équations plus performantes.
- La *construction de schémas numériques adaptés* pour ces équations.
- L'*application à des problèmes concrets*, afin d'illustrer l'intérêt pratique de ces approches.

Perspectives :

De nouvelles voies d'études des EDP de régularisation ont été ouvertes par cette thèse, et pourraient être envisagées comme futurs axes de recherche. D'abord, les questions théoriques d'unicité et de convergence des algorithmes proposés : Nos nouvelles équations unificatrices pourraient être mieux adaptées pour étudier ces aspects théoriques importants.

Nous souhaitons ensuite faire valider par des spécialistes, les résultats obtenus sur la régularisation d'images IRMd. Cela permettrait en particulier de définir des à-priori physiologiques qui pourraient être facilement incorporées dans nos EDP de régularisation. Ceci, afin d'obtenir des modèles de réseaux de fibres de la matière blanche plausible d'un point de vue physiologique.

Finalement, notre nouvelle méthode de régularisation d'images vectorielles pourraient être intégrée dans des processus autres, comme par exemple le calcul de flot optiques ou les algorithmes de segmentation d'images, les rendant possiblement plus efficace grâce à la prise en compte des géométries locales inhérent à ces problèmes.

Author's publication list

International Journals and Magazines

- *D. Tschumperlé and R. Deriche.*
Diffusion PDE's on Vector-valued Images : Local Approach and Geometric Viewpoint.
In IEEE SPM (Signal Processing Magazine), September 2002, Vol 19. No 5. pp 16-25.
- *D. Tschumperlé and R. Deriche.*
Orthonormal Vector Sets Regularization with PDE's and Applications.
To appear in IJCV (International Journal of Computer Vision), December 2002.
- *C. Chef d'hôtel, D. Tschumperlé, R. Deriche and O. Faugeras.*
Manifold Constrained Partial Differential Equations and Geometric Integration Methods in Image Processing. Submitted to Journal of Mathematical Imaging and Vision.

International Conference Papers

- *D. Tschumperlé and R. Deriche.*
Constrained and Unconstrained PDEs for Vector Image Restoration.
In Proceedings of SCIA'2001 (Scandinavian Conference on Image Analysis), pp 153-160.
- *D. Tschumperlé and R. Deriche.*
Regularization of orthonormal vector sets using coupled PDE's.
In Proceedings of VLISM'2001 (IEEE Workshop on Variational and Level Sets Methods), pp 3-10.
- *D. Tschumperlé and R. Deriche.*
Diffusion Tensor Regularization with Constraints Preservation.
In Proceedings of CVPR'2001 (IEEE Conference on Computer Vision and Pattern Recognition).
- *D. Tschumperlé and R. Deriche.*
Constrained Flows of Matrix-Valued Functions : Application to Diffusion Tensor Regularization.
In Proceedings of ECCV'2002 (European Conference on Computer Vision).

- *D. Tschumperlé and R. Deriche.*
Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications. Submitted to CVPR'2003.
- *D. Tschumperlé and R. Deriche.*
DT-MRI Images : Estimation, Regularization and Application. To appear in EUROCAST'2003, Workshop "NI": Neuroinformatics and Neuroimaging, Las-Palmas, Spain, Feb. 2003.

National Conference Papers

- *D. Tschumperlé and R. Deriche.*
Restauration d'images vectorielles par EDP.
In Proceedings of RFIA'2000 (Conférence sur la Reconnaissance des Formes et l'Intelligence Artificielle).
- *D. Tschumperlé and R. Deriche.*
Régularisation par EDP de Champs de Vecteurs Orthonormes et Applications.
In Proceedings of RFIA'2002 (Conférence sur la Reconnaissance des Formes et l'Intelligence Artificielle).
- *D. Tschumperlé and R. Deriche.*
EDP, Images multivalues Contraintes et Applications.
In Proceedings of "Le traitement d' image à l'aube du XXIeme siecle".

Research Report

- *D. Tschumperlé and R. Deriche.*
Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications.
INRIA Research Report No 4657, Projet ODYSEE, Dcembre 2002.

Bibliography

- [1] R. Abraham, J.E Marsden, and T.S. Ratiu. *Manifolds, Tensor Analysis, and Applications*. Springer-Verlag, New York, 1991.
- [2] J.F. Abramatic and L.M. Silverman. Non linear restoration of noisy images. *IEEE Trans. Patern Analysis and Machine Intelligence*, 4(2):141–149, 1982.
- [3] L. Alvarez, R. Deriche, T. Papadopoulos, and J. Sanchez. Symmetrical dense optical flow estimation with occlusion detection. In proceedings of ECCV’02, pages 721–735, Copenhagen/Denmark.
- [4] L. Alvarez, R. Deriche, and F. Santana. Recursivity and PDE’s in image processing . In *Proceedings 15th International Conference on Pattern Recognition*, volume I, pages 242–248, September 2000.
- [5] L. Alvarez, R. Deriche, J. Weickert, and J. Sànchez. Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach. *International Journal of Visual Communication and Image Representation, Special Issue on Partial Differential Equations in Image Processing, Computer Vision and Computer Graphics*, 2000.
- [6] L. Alvarez, J. Esclarín, M. Lefebure, and J. Sánchez. A pde model for computing the optical flow. In *Proceedings of CEDYA XVI*. Universidad de Las Palmas de Gran Canaria, 1999.
- [7] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 123(3):199–257, 1993.
- [8] L. Alvarez, P.L. Lions, and J.M. Morel. Image selective smoothing and edge detection by nonlinear diffusion (II). *SIAM Journal of Numerical Analysis*, 29:845–866, 1992.
- [9] L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal of Numerical Analysis*, 31(2):590–605, 1994.
- [10] L. Alvarez, J. Weickert, and J. Sànchez. A scale-space approach to nonlocal optical flow calculations. In Mads Nielsen, P. Johansen, O.F. Olsen, and J. Weickert,

- editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 235–246. Springer–Verlag, 1999.
- [11] L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *The International Journal of Computer Vision*, 39(1):41–56, August 2000.
- [12] L. Alvarez. Images and PDE's. In M.O. Berger, R. Deriche, I. Herlin, J. Jaffre, and J.M. Morel, editors, *Images, Wavelets and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*. Springer, June 1996.
- [13] Y. Amit. A nonlinear variational problem for image matching. *SIAM Journal on Scientific Computing*, 15(1), January 1994.
- [14] H.C. Andrews and B.R. Hunt. *Digital Image Restoration*. Signal Processing. Prentice Hall, Englewood Cliffs, N.J., 1977.
- [15] G. Aubert, M. Barlaud, O. Faugeras, and J. Jehan Besson. Image segmentation using active contours: calculus of variations or shape optimization? *SIAM Journal on Applied Mathematics*, 2002. Submitted.
- [16] G. Aubert, L. Blanc-Féraud, M. Barlaud, and P. Charbonnier. A deterministic algorithm for edge-preserving computed imaging using Legendre transform. In *Proceedings of the International Conference on Pattern Recognition*, volume III, pages 188–191, Jerusalem, Israel, October 1994. Computer Society Press.
- [17] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal of Applied Mathematics*, 60(1):156–182, 1999.
- [18] G. Aubert and P. Kornprobst. A mathematical study of the relaxed optical flow problem in the space BV. *SIAM Journal on Mathematical Analysis*, 30(6):1282–1308, 1999.
- [19] 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.
- [20] D. Barash. Bilateral filtering and anisotropic diffusion : Towards a unified viewpoint. Technical report, HP Laboratories Israel, 2000.
- [21] F. Barbaresco. Spatial denoising of statistical parameters estimation by beltrami diffusion on embedding siegel space. *Submitted to PSIP 2003 (Physics in Signal and Image Processing)*, 2002.
- [22] M. Barlaud and C. Labit. *Compression et codage des images et des vidéos*. Ouvrage Collectif Hermes-Lavoisier, 2002.

- [23] Bart M. ter Haar Romeny. *Geometry-driven diffusion in computer vision*. Computational imaging and vision. Kluwer Academic Publishers, 1994.
- [24] J. Becker, T. Preusser, and M. Rumpf. Pde methods in flow simulation post processing. *Computing and Visualization in Science*, 3(3):159–167, 2000.
- [25] 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.
- [26] 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.
- [27] M. Bertalmio, G. Sapiro, L.T. Cheng, and S. Osher. Variational problems and PDE's on implicit surfaces. In *IEEE Workshop on Variational and Level Set Methods VLSM'01* pages 186–193, Vancouver/Canada.
- [28] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. Imag. Proc.*, 7(3):421–432, 1998. Special Issue on Partial Differential Equations and Geometry-Driven Diffusion in Image Processing and Analysis.
- [29] L. Blanc-Feraud, P. Charbonnier, G. Aubert, and M. Barlaud. Nonlinear image processing : Modeling and fast algorithm for regularization with edge detection. In *Proceedings of the International Conference on Image Processing*, pages 474–477, Washington, USA, October 1995.
- [30] P. Blomgren. *Total Variation Methods for Restoration of Vector Valued Images*. PhD thesis, Department of Mathematics, University of California, Los Angeles, June 1998.
- [31] 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.
- [32] J. Bride and P. Meer. Global registration via direct methods: a statistical approach. In *Proceedings of CVPR'01*, December 2001.
- [33] D. Buerkle, T. Preusser, and M. Rumpf. Transport and diffusion in timedependent flow visualization. In *Proceedings IEEE Visualization*, 2001.
- [34] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *The International Journal of Computer Vision*, 22(1):61–79, 1997.

- [35] V. Caselles, J.M. Morel, G. Sapiro, and A. Tannenbaum. Introduction to the special issue on partial differential equations and geometry-driven diffusion in image processing and analysis. *IEEE Transactions on Image Processing*, 7(3):269–273, 1998.
- [36] SHFJ CEA. Web page : <http://www-dsv.cea.fr/thema/shfj>, 2000.
- [37] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.
- [38] T. Chan, S.H. Kang, and J. Shen. Euler’s elastica and curvature based inpainting. *SIAM J. Appl. Math.*, 2002.
- [39] T. Chan and J. Shen. Variational restoration of non-flat image features : Models and algorithms. *Research Report. Computational and applied mathematics department of mathematics Los Angeles.*, June 1999.
- [40] T. Chan and J. Shen. Mathematical models for local deterministic inpaintings. Technical Report 00-11, Department of Mathematics, UCLA, Los Angeles, March 2000.
- [41] T. Chan and J. Shen. Non-texture inpaintings by curvature-driven diffusions (cdd). Technical Report 00-35, Department of Mathematics, UCLA, Los Angeles, September 2000.
- [42] T. Chan and L. Vese. Image segmentation using level sets and the piecewise-constant Mumford–Shah model. Technical Report 00-14, UCLA CAM Report, 2000.
- [43] T.F. Chan, S.H. Kang, and J. Shen. Total variation denoising and enhancement color images based on the cb and hsv color models. *Journal of Visual Communication and Image Representation*, 12(4), June 2000.
- [44] T.F. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (cdd). *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001.
- [45] P. Charbonnier, G. Aubert, M. Blanc-Féraud, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of the International Conference on Image Processing*, volume II, pages 168–172, 1994.
- [46] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.
- [47] P. Charbonnier, L. Blanc-Feraud, and M. Barlaud. Noisy image restoration using multiresolution markov field. *JVCIR (Journal of Visual Communication and Image Representation)*, 3(14):338–346, December 1992.

- [48] C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Constrained flows on matrix-valued functions : application to diffusion tensor regularization. In *Proceedings of ECCV'02*, June 2002.
- [49] C. Chefd'hotel, G. Hermosillo, and O. Faugeras. A variational approach to multi-modal image matching. In *IEEE Workshop on Variational and Level Set Methods VLSM'01*, pages 21–28, Vancouver/Canada.
- [50] Chir. Web page, <http://www-sop.inria.fr/chir>.
- [51] M.T. Chu. Constructing symmetric nonnegative matrices with prescribed eigenvalues by differential equations. Technical report, Department of Mathematics, North Carolina State University, August 1990.
- [52] M.T. Chu. A list of matrix flows with applications. Technical report, Department of Mathematics, North Carolina State University, 1990.
- [53] M.T. Chu. Matrix differential equations : A continuous realization process for linear algebra problems. Technical report, Department of Mathematics, North Carolina State University, 1990.
- [54] M.T. Chu. On a differential equation approach to the weighted orthogonal procrustes problem. Technical report, Department of Mathematics, North Carolina State University, 1990.
- [55] M. Clerc, O. Faugeras, R. Keriven, J. Kybic, and T. Papadopoulos. A level set method for the inverse eeg/meg problem. In *Proceedings of the 8th International Conference on Functional Mapping of the Human Brain*, June 2002.
- [56] I. Cohen. Nonlinear variational method for optical flow computation. In *Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, 1993.
- [57] L.D. Cohen. On active contour models. Technical Report 1075, INRIA, August 1989.
- [58] C. Connolly and T. Fliess. A study of efficiency and accuracy in the transformation from rgb to cielab color space. *IEEE Transactions on Image Processing*, 6(7), July 97.
- [59] G.H. Cottet and L. Germain. Image processing through reaction combined with nonlinear diffusion. *Mathematics of Computation*, 61(204):659–673, October 1993.
- [60] O. Coulon, D.C. Alexander, and S.R. Arridge. A geometrical approach to 3d diffusion tensor magnetic resonance image regularisation. Technical report, Department of Computer Science, University College London., 2001.

- [61] O. Coulon, D.C. Alexander, and S.R. Arridge. A regularization scheme for diffusion tensor magnetic resonance images. In *XVIIth International Conference on Information Processing in Medical Imaging*, 2001.
- [62] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *The International Journal of Computer Vision*, 1(2):167–187, May 1987.
- [63] R. Deriche. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(12):78–88, January 1990.
- [64] R. Deriche. Recursively Implementing the Gaussian and Its Derivatives. In *Proc. Second International Conference On Image Processing*, pages 263–267, Singapore, September 7-11 1992.
- [65] R. Deriche. Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, Unité de Recherche Sophia-Antipolis, 1993.
- [66] R. Deriche and O. Faugeras. Les EDP en traitement des images et vision par ordinateur. Technical report, INRIA, November 1995. A more complete version of this Research Report has appeared in the French Revue "Traitement du Signal". Volume 13 - No 6 - Special 1996.
- [67] R. Deriche, P. Kornprobst, and G. Aubert. Optical flow estimation while preserving its discontinuities: A variational approach. In *Proceedings of the 2nd Asian Conference on Computer Vision ACCV'95* pages 71–80.
- [68] F. Devernay. *Vision stéréoscopique et propriétés différentielles des surfaces*. PhD thesis, École Polytechnique, Palaiseau, France, February 1997.
- [69] 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.
- [70] Epidaure. Web page : <http://www-sop.inria.fr/epidaure>, 2000.
- [71] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
- [72] O. Faugeras, F. Clément, R. Deriche, R. Keriven, T. Papadopoulos, J. Roberts, T. Viéville, F. Devernay, J. Gomes, G. Hermosillo, P. Kornprobst, and D. Lingrand. The inverse EEG and MEG problems: The adjoint space approach I: The continuous case. Technical Report 3673, INRIA, May 1999.
- [73] O. Faugeras, Q.T. Luong, and T. Papadopoulos. *The Geometry of Multiple Images*. MIT Press, 2001.

- [74] O. D. Faugeras and G. Hermosillo. Well-posedness of eight problems of multi-modal statistical image-matching. Technical Report 4235, INRIA, August 2001.
- [75] L. Florack. *Image Structure*. Kluwer Academic Publishers, 1997.
- [76] G.B. Folland. *Fourier Analysis and its Applications*. The Wadsworth & Brooks/Cole Mathematics Series. Brooks/Cole Advanced Books & Software, 1992.
- [77] M. A. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *Proceedings of the Intercommission Workshop of the International Society for Photogrammetry and Remote Sensing*, Interlaken, Switzerland, 1987.
- [78] B.R. Frieden. Restoring with maximum likelihood and maximum entropy. *Journal Opt. Soc. Amer.*, 62:511–518, 1972.
- [79] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [80] S. Geman and D.E. McClure. Bayesian image analysis: an application to single photon emission tomography. *Amer. Statist. Assoc.*, pages 12–18, 1985.
- [81] G. Gerig, O. Kubler, R. Kikinis, and F. Jolesz. Nonlinear anisotropic filtering of mri data. *IEEE TMI*, 11(2):221–231, 1992.
- [82] G. Gilboa, N. Sochen, and Y. Zeevi. Forward-and-backward diffusion processes for adaptative image enhancement and denoising. *IEEE Transactions on Image Processing*, 2002.
- [83] G. Gilboa, N. Sochen, and Y. Zeevi. Regularized shock filters and complex diffusion. In *Proceedings of ECCV'02*, pages 399–413, Copenhagen/Denmark.
- [84] G. Gilboa, N. Sochen, and Y.Y. Zeevi. Complex Diffusion Processes for Image Filtering. In *Proceedings of ECCV'02*, Copenhagen/Denmark.
- [85] G.H. Golub and C.F. Van Loan. *Matrix computations*. The John Hopkins University Press, Baltimore, Maryland, 1983.
- [86] G.H. Golub and C.F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, second edition, 1989.
- [87] J. Gomes. *Implicit representations of evolving manifolds in computer vision*. PhD thesis, Université de Nice Sophia-Antipolis, 2001.
- [88] A. Gouze, M. Antonini, and M. Barlaud. Quincunx filtering lifting scheme for image coding. In *SPIE, Visual Communications and Image Processing*, San Jose, USA, January 1999.

- [89] G.H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.
- [90] P.J. Green. Bayesian reconstruction from emission tomography data using a modified em algorithm. *IEEE Trans. Med. Imaging*, MI-9(1):84–93, March 1990.
- [91] J. Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, 1923.
- [92] A. Ben Hamza, G.R. Unal, and H. Krim. Towards a unified estimation theme: probabilistic vs. variational. *Signal Processing Magazine Special Issue*, 2002.
- [93] G. Hermosillo, O. Faugeras, and J. Gomes. Unfolding the cerebral cortex using level set methods. In Mads Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 58–69. Springer, September 1999.
- [94] G. Hermosillo. *Variational Methods for Multimodal Image Matching*. PhD thesis, INRIA, The document is accessible at <ftp://ftp-sop.inria.fr/robotvis/html/Papers/hermosillo:02.ps.gz>, 2002.
- [95] B.K. Horn and M.J. Brooks, editors. *Shape from Shading*. The MIT Press, 1989.
- [96] B.K.P. Horn. Height and Gradient from Shading. *The International Journal of Computer Vision*, 5(1):37–75, August 1990.
- [97] B.R. Hunt. The application of constrained least squares estimation to image restoration by digital computer. *IEEE Trans. Computers*, C-22(9):805–812, 1973.
- [98] S. Jehan-Besson, M. Barlaud, and G. Aubert. Video object segmentation using eulerian region based active contours. In *8th ICCV - Vancouver - BC - Canada*, volume I, pages 353–360, July 2001.
- [99] JPEG Consortium Web Page <http://www.jpeg.org>
- [100] K. Kanatani. Automatic singularity test for motion analysis by an information criterion. In Bernard Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, pages 697–708, Cambridge, UK, April 1996.
- [101] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, London, June 1987.
- [102] S. Kichenassamy. The Perona–Malik paradox. *SIAM Journal of Applied Mathematics*, 57(5):1328–1342, 1997.
- [103] R. Kimmel, R. Malladi, and N. Sochen. Image processing via the beltrami operator. In *Proceedings of the 3rd Asian Conference on Computer Vision*, volume 1, pages 574–581, Hong Kong, January 1998.

- [104] 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.
- [105] R. Kimmel and N. Sochen. Geometric-variational approach for color image enhancement and segmentation. In Mads Nielsen, Peter Johansen, Ole F. Olsen, and Joachim Weickert, editors, *Scale-Space Theories in Computer Vision, Second International Conference, Scale-Space'99*, volume 1682 of *Lecture Note in Computer Science*, pages 295–305. Springer, 1999.
- [106] R. Kimmel and N. Sochen. Orientation diffusion or how to comb a porcupine. Technical Report 2000-02, CIS, 2000. Accepted to special issue on PDEs in Image Processing, Computer Vision, and Computer Graphics, Journal of Visual Communication and Image Representation, 2000.
- [107] R. Kimmel and N. Sochen. Orientation diffusion or how to comb a porcupine. *Journal of Visual Communication and Image Representation*, 2001.
- [108] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [109] P. Kornprobst. Calcul de flot optique avec préservation des discontinuités. Technical report, Rapport de stage de DEA. Proposé et encadré par R.Deriche et G. Aubert - Université Nice-Sophia Antipolis, France, June 1995.
- [110] 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.
- [111] P. Kornprobst, R. Deriche, and G. Aubert. Image restoration via PDE's. In *First Annual Symposium on Enabling Technologies for Law Enforcement and Security - SPIE Conference 2942 : Investigative Image Processing.*, Boston, Massachusetts, USA, November 1996.
- [112] 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.
- [113] P. Kornprobst, R. Deriche, and G. Aubert. EDP, débruitage et réhaussement en traitement d'image: Analyse et contributions. In *11ème Congrès RFIA*, volume 1, pages 277–286. AFCET, January 1998.
- [114] A. Koschan. A comparative study on color edge detection. In *Proceedings of the 2nd Asian Conference on Computer Vision, ACCV'95*, pages 574–578.

- [115] D. Le Bihan. Methods and applications of diffusion mri. In I.R. Young, editor, *Magnetic Resonance Imaging and Spectroscopy in Medicine and Biology*. John Wiley and Sons, 2000.
- [116] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: problems with bilinear constraint. *The International Journal of Computer Vision*, 37(2):1–24, June 2000.
- [117] M. Lefébure and L. Cohen. Optical flow and image registration : a new local rigidity approach for global minimization. In M. Figueiredo, J. Zerubia, and A. Jain, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'2001)*, volume 2134 of *Lecture Notes in Computer Science*, Sophia-Antipolis, France, 2001. Springer.
- [118] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [119] L.M. Lorigo, O.D. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin. CURVES: Curve evolution for vessel segmentation. *Medical Image Analysis*, 5:195–206, 2001.
- [120] 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.
- [121] G. Medioni, M.S. Lee, and C.K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.
- [122] J.M. Morel and S. Solimini. Segmentation of images by variational methods: A constructive approach. *Rev. Math. Univ. Complut. Madrid*, 1:169–182, 1988.
- [123] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, June 1985. IEEE.
- [124] H-H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence Journal*, 33:299–324, 1987.
- [125] M. Ng, R. Chan, and W. Tang. A fast algorithm for deblurring models with neumann boundary conditions. *SIAM J. Sci. Comput.*, 21:851–866, 2000.
- [126] M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7(4):291–308, 1997. Related ECCV:96 article available at <ftp://ftp-sop.inria.fr/robotvis/pub/html/Papers/nielsen-florack-et al:96.ps.gz>.
- [127] M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal of Applied Mathematics*, 61(2):633–658, 2000.

- [128] M. Nikolova and M. Ng. Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning. In *Proceedings of the International Conference on Image Processing*. IEEE Signal Processing Society, 2001.
- [129] S. Osher and R. Fedkiw. *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, October 2002.
- [130] S. Osher and N. Paragios, editors. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer-Verlag, August 2002.
- [131] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal of Numerical Analysis*, 27(4):919–940, August 1990.
- [132] T. Papadopoulos and M.I.A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. Research Report 3961, INRIA Sophia-Antipolis, June 2000.
- [133] N. Paragios and R. Deriche. ARTemis: A Geodesic Active Contour Tracker. In *IEEE Conference on Computer Vision and Pattern Recognition, the demo session*, Santa Barbara, USA, June 1998. IEEE Computer Society.
- [134] N. Paragios and R. Deriche. A PDE-based level set approach for detection and tracking of moving objects. In *Proceedings of the 6th International Conference on Computer Vision*, pages 1139–1145, Bombay, India, January 1998. IEEE Computer Society, IEEE Computer Society Press.
- [135] N. Paragios and R. Deriche. Geodesic active contours for supervised texture segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999. IEEE Computer Society.
- [136] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:266–280, March 2000.
- [137] N. Paragios and R. Deriche. Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. *International Journal of Visual Communication and Image Representation, Special Issue on Partial Differential Equations in Image Processing, Computer Vision and Computer Graphics*, 2001. To appear in 2001.
- [138] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *The International Journal of Computer Vision*, 46(3):223, 2002.
- [139] N. Paragios. *Geodesic Active Regions and Level Set Methods: Contributions and Applications on Artificial Vision*. PhD thesis, University of Nice Sophia-Antipolis, January 2000.

- [140] A. Pardo and G. Sapiro. Vector probability diffusion. In *Proceedings of the International Conference on Image Processing*. IEEE Signal Processing Society, September 2000.
- [141] X. Pennec. Registration of uncertain geometric features: Estimating the pose and its accuracy. In editor J. Le Moigne, editor, *Proceedings of the First Image Registration Workshop*, Greenbelt, USA, November 1997.
- [142] P. Perona. Orientation diffusions. *IEEE Transactions on Image Processing*, 7(3):457–467, March 1998.
- [143] 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.
- [144] Poseidon. The lifeguards third eye, web page : <http://www.poseidon-tech.com>,
- [145] C. Poupon. *Détection des faisceaux de fibres de la substance blanche pour l'étude de la connectivité anatomique cérébrale*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, December 1999.
- [146] C. Poupon, J.F Mangin, V. Frouin, J. Regis, F. Poupon, M. Pachot-Clouard, D. Le Bihan, and I. Bloch. Regularization of mr diffusion tensor maps for tracking brain white matter bundles. In W.M. Wells, A. Colchester, and S. Delp, editors, *Medical Image Computing and Computer-Assisted Intervention-MICCAI'98*, number 1496 in Lecture Notes in Computer Science, pages 489–498, Cambridge, MA, USA, October 1998. Springer.
- [147] C.A. Poynton. Poynton's colour faq (www.inforamp.net/poynton). Web page, 1995.
- [148] E. Prados, O. Faugeras, and E. Rouy. Shape from shading and viscosity solutions. Technical report, INRIA, 2002.
- [149] E. Prados, O. Faugeras, and E. Rouy. Shape from shading and viscosity solutions. In *Proceedings of ECCV'02*, June 2002.
- [150] RealviZ. Web-site : <http://www.realviz.com>.
- [151] A. Roche, G. Malandain, X. Pennec, and N. Ayache. Multimodal image registration by maximization of the correlation ratio. Technical Report 3378, INRIA, August 1998.
- [152] M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. In *Proc. IEEE Workshop on Motion and Video Computing*, Orlando, Florida, December 2002.

- [153] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [154] G. Sapiro. Vector-Valued Active Contours. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 680–685, San Francisco, CA, June 1996. IEEE.
- [155] G. Sapiro. Color snakes. *Computer Vision and Image Understanding*, 68(2), 1997.
- [156] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
- [157] 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.
- [158] H. Scharr and J. Weickert. An anisotropic diffusion algorithm with optimized rotation invariance. In *G. Sommer, N. Krüger, C. Perwass (Eds.), Mustererkennung 2000, Springer, Berlin*, pages 460–467, 2000.
- [159] J.A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
- [160] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. *International Conference on Computer Vision and Pattern Recognition*, 1996.
- [161] J. Shah. Curve evolution and segmentation functionals: Applications to color images. In *Proceedings of the International Conference on Image Processing*, pages 461–464, 1996.
- [162] N. Sochen. On affine invariance in the beltrami framework for vision. In *IEEE Workshop on Variational and Level Set Methods VLSM'01* pages 51–56, Vancouver/Canada.
- [163] 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.
- [164] N. Sochen, R. Kimmel, and R. Malladi. From high energy physics to low level vision. Technical Report 39243, LBNL report, UC Berkeley, 1996.
- [165] N. Sochen, R. Kimmel, and R. Malladi. A geometrical framework for low level vision. *IEEE Transaction on Image Processing, Special Issue on PDE based Image Processing*, 7(3):310–318, 1998.
- [166] I.S Sokolnikoff. *Tensor Analysis – Theory and Applications, 2n Ed.* Wiley, New York, 1964.

- [167] B. Tang, G. Sapiro, and V. Caselles. Direction diffusion. *International Conference on Computer Vision*, 1998.
- [168] 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.
- [169] S. Teboul, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Variational approach for edge-preserving regularization using coupled PDE's. *IEEE Transaction on Image Processing, Special Issue on PDE based Image Processing*, 7(3):387–397, 1998.
- [170] A.N. Tikhonov. Regularization of incorrectly posed problems. *Soviet. Math. Dokl.*, 4:1624–1627, 1963.
- [171] 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.
- [172] A. Trouné and L. Younes. Mise en correspondance par difféomorphismes en une dimension: définition et maximisation de fonctionnelles. In *12ème Congrès RFIA '00* Paris.
- [173] D. Tschumperlé and R. Deriche. Restauration d'images vectorielles par EDP. In *12ème Congrès RFIA '00*, pages 247–256. Paris.
- [174] 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.
- [175] D. Tschumperlé and R. Deriche. Diffusion tensor regularization with constraints preservation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai Marriott, Hawaii, December 2001.
- [176] D. Tschumperlé and R. Deriche. Regularization of orthonormal vector sets using coupled PDE's. In *IEEE Workshop on Variational and Level Set Methods VLSM'01* pages 3–10, Vancouver/Canada.
- [177] D. Tschumperlé and R. Deriche. Diffusion PDE's on Vector-Valued images. *IEEE Signal Processing Magazine*, 19(5):16–25, 2002.
- [178] D. Tschumperlé and R. Deriche. Orthonormal Vector Sets Regularization with PDE's and Applications. *International Journal of Computer Vision (IJCV, Special Issue VLSM)*, 2002.
- [179] D. Tschumperlé and R. Deriche. Régularisation par EDP de champs de vecteurs orthonormés et applications. In *13ème Congrès RFIA '02*, Angers, January 2002.

- [180] H. Urakawa. *Calculus of Variations and Harmonic Maps*. Translations of Mathematical Monographs 132, American Mathematical Society, 1993.
- [181] B. Vemuri, Y. Chen, M. Rao, T. McGraw, T. Mareci, and Z. Wang. Fiber tract mapping from diffusion tensor mri. In *1st IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM'01)*, July 2001.
- [182] L.A. Vese and S. Osher. Numerical methods for p-harmonic flows and applications to image processing. CAM Report 01-22, UCLA, August 2001.
- [183] P. Viola. *Alignment by Maximisation of Mutual Information*. PhD thesis, MIT, 1995.
- [184] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, University of Kaiserslautern, Germany, Laboratory of Technomathematics, January 1996.
- [185] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplement*, 11:221–236, 1996.
- [186] J. Weickert. Coherence-enhancing diffusion of colour images. *7th National Symposium on Pattern Recognition and Image Analysis*, April 1997.
- [187] J. Weickert. A review of nonlinear diffusion filtering. *Scale-Space Theory in Computer Vision, Lecture Notes in Comp. Science (Springer, Berlin)*, 1252:3–28, 1997. Invited Paper.
- [188] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, 1998.
- [189] J. Weickert and B. Benhamouda. *A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox*, pages 1–10. *Advances in Computer Vision*, Springer, Wien, 1997.
- [190] J. Weickert and B. Benhamouda. Why the peronamalik filter works. Technical Report 97/22, Department of Computer Science, University of Copenhagen, 1997.
- [191] J. Weickert and T. Brox. Diffusion and regularization of vector and matrix-valued images. Technical report, Universitat des Saarlandes, 2002.
- [192] 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.
- [193] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer-Verlag Berlin Heidelberg New York., 2000.
- [194] A.P. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.

- [195] C. Wurster. *Computers : An illustrated History*. Unknown, February 2002.
- [196] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33:116–125, 1986.
-

Résumé

Nous nous intéressons aux approches par EDP pour la régularisation d'images multivaluées, et leurs applications à une large classe de problèmes d'intérêts. L'étude et la comparaison des méthodes existantes nous permet à la fois de proposer un cadre mathématique commun mieux adapté aux interprétations géométriques locales de ces EDP, mais aussi de concevoir des schémas numériques efficaces pour leur mise en oeuvre. Nous développons de cette façon une nouvelle approche de régularisation multivaluée vérifiant certaines propriétés géométriques locales importantes, qui peut être utilisée dans de nombreuses applications différentes. Nous abordons ensuite le problème lié à la régularisation de données multivaluées contraintes. Un formalisme variationnel est proposé afin de traiter dans un cadre unifié, des données de direction comme les champs de vecteurs unitaires, de matrices de rotation, de tenseurs de diffusion etc. Les solutions apportées sont analysées et utilisées avec succès pour résoudre de nombreux problèmes, notamment la régularisation et l'interpolation d'images couleurs, la visualisation de flots, la régularisation de mouvements rigides estimés à partir de séquences vidéos, et l'aide à la reconstruction de réseaux cohérents de fibres dans la matière blanche du cerveau, à partir de la régularisation d'images d'IRM de diffusion.

Abstract

We are interested in PDE-based approaches for vector-valued image regularization, and its applications for a wide class of interesting image processing problems. The comparative study of existing methods allows us to propose a common mathematical framework, better adapted to understand the underlying diffusion geometry of the regularization processes, as well as design corresponding numerical schemes. Thus we develop a new multivalued image regularization approach that verifies important geometric properties. It can be used in a large range of regularization-related applications. We also tackle the problem of constrained regularization and propose a specific variational formalism unifying in a common framework, the equations acting on direction features : unit vectors, rotation matrices, diffusion tensors, etc. Proposed solutions are analyzed and used with success to solve applications of interest, such as color image regularization and interpolation, flow visualization, regularization of rigid motions estimated from video sequences, and aided reconstruction of coherent fibers network models in the white matter of the brain, using DT-MRI imaging.