


Springer Proceedings in Mathematics 5

Maitine Bergounioux *Editor*

Mathematical Image Processing

 Springer

Springer Proceedings in Mathematics

Volume 5

For further volumes:
<http://www.springer.com/series/8806>

Springer Proceedings in Mathematics

The book series will feature volumes of selected contributions from workshops and conferences in all areas of current research activity in mathematics. Besides an overall evaluation, at the hands of the publisher, of the interest, scientific quality, and timeliness of each proposal, every individual contribution will be refereed to standards comparable to those of leading mathematics journals. It is hoped that this series will thus propose to the research community well-edited and authoritative reports on newest developments in the most interesting and promising areas of mathematical research today.

Maïtine Bergounioux

Editor

Mathematical Image Processing

University of Orléans, France,
March 29th - April 1st, 2010



Springer

Editor

Maitine Bergounioux
UMR 6628-MAPMO
Fédération Denis Poisson
Université d'Orléans BP 6759
45067 Orléans Cedex 2
France
maitine.bergounioux@univ-orleans.fr

ISSN 2190-5614

ISBN 978-3-642-19603-4

e-ISBN 978-3-642-19604-1

DOI 10.1007/978-3-642-19604-1

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011927924

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume collects a number of the contributions presented at the Second Conference “Mathematics and Image Processing” held at the University of Orléans in 2010.

During the period March 29th–April 1st, 2010, roughly 80 people attended the conference and 20 invited talks were delivered covering various areas of Mathematical Image Processing. The main topics discussed during this meeting were the following:

- H. Ammari: *Generalized Polarization Tensors: Recovery of Fine Shape Details*
- H. Biermé: *Modélisation et analyse fractale de textures anisotropes. Applications imagerie médicale*
- X. Bonnefond: *The back and forth nudging method applied to thermoacoustic tomography*
- A. Cohen: *Theory and algorithms for anisotropic triangulations with applications to image representations*
- P.L. Combettes: *Méthodes d’éclatement proximales en traitement de l’image*
- F. Dibos: *Invariants projectifs: une approche par le groupe des recalages*
- L. Duval: *Statistical estimators based on Stein’s principle for M-band wavelets and filter banks*
- O. Faugeras: *Hyperbolic planforms in relation to visual edges and textures perception*
- A. Glaunes: *Mesures et courants comme modèles des sous-variétés en analyse d’image. Applications*
- Y. Gousseau: *Modèles “germe-grain” et principes de formation des images*
- S. Jaffard: *Méthodes d’analyse multifractale pour la classification d’images*
- P. Kornprobst: *From algorithmic to bio-inspired vision: is it so different ?*
- J. Lefèvre: *Analyse et modélisation du développement cérébral*
- S. Mallat: *Sparse Mixture Models for Inverse Problems*
- V. Perrier: *Divergence-free and Curl-free Wavelets on the cube and related applications*
- L. Piffet: *BV2 models for image denoising and texture identification*

- F. Rossi: *Image Reconstruction via hypoelliptic diffusion on the bundle of directions of the plane*
- O. Scherzer: *Derivative Free Functionals for Imaging*
- D.Tschumperlé: *PDEs on the space of patches for image denoising and registration*
- P. Weiss: *On the design of gradient coils for MRI with minimax current density*

The conference was organized by the “Image Group” of the Research Federation “Denis-Poisson” of Orléans and Tours Universities (France). As the chair of the Organizing Committee, I would like to extend my thanks the following institutions for their financial and material support:

- Région Centre
- Conseil Général du Loiret
- Mairie d’Orléans
- Université d’Orléans
- Fédération Denis-Poisson (CNRS) et Laboratoire MAPMO d’Orléans
- Commissariat à l’Energie Atomique (CEA)

The contributions appearing in this volume are a snapshot of the different topics that were discussed during the conference. They mainly concern, image reconstruction, texture extraction and image classification and involve a variety of disparate methods and applications. Therefore it was impossible to split the papers into generic groups, which is why they are presented in alphabetic order. All contributions are peer-reviewed.

I would like to thank all the contributors for their effort and the anonymous referees for their comments and suggestions. The help provided by the staff of Springer-Verlag is also greatly appreciated.

Orléans
December 2010

Maitine Bergounioux

Contents

Function Spaces Vs. Scaling Functions: Tools for Image Classification . . .	1
Stéphane Jaffard, Patrice Abry, and Stéphane Roux	
A Second Order Model for 3D-Texture Extraction	41
Maïtine Bergounioux and Minh Phuong Tran	
Analysis of Texture Anisotropy Based on Some Gaussian Fields with Spectral Density	59
Hermine Biermé and Frédéric J.P. Richard	
Image Reconstruction Via Hypoelliptic Diffusion on the Bundle of Directions of the Plane	75
Ugo Boscain, Jean Duplaix, Jean-Paul Gauthier, and Francesco Rossi	
Projective Invariant Features Detection and the Registration Group	91
Françoise Dibos	
Morphological Component Analysis for Decomposing Dynamic Textures	107
Sloven Dubois, Renaud Péteri, and Michel Ménard	
Texture Enhancing Based on Variational Image Decomposition	127
Florian Frühau, Carsten Pontow, and Otmar Scherzer	
A Locally Anisotropic Model for Image Texture Extraction	141
Loïc Piffet	
A Neural Field Model for Motion Estimation	159
Émilien Tlapale, Pierre Kornprobst, Guillaume S. Masson, and Olivier Faugeras	
Non-Local Regularization and Registration of Multi-Valued Images By PDE's and Variational Methods on Higher Dimensional Spaces	181
David Tschumperlé and Luc Brun	

List of Contributors

Patrice Abry CNRS UMR 5672 Laboratoire de Physique, ENS de Lyon, 46, allée d'Italie, F-69364 Lyon cedex, France, patrice.abry@ens-lyon.fr

Maïtine Bergounioux Laboratoire MAPMO - UMR 6628, Fédération Denis-Poisson, Université d'Orléans, France, maitine.bergounioux@univ-orleans.fr

Hermine Biermé Université Paris Descartes, Laboratoire MAP5 (CNRS UMR 8145), 45 rue des Saints-Pères, 75270 Paris Cedex 06, France, hermine.bierme@parisdescartes.fr

Ugo Boscain CMAP, École Polytechnique CNRS, Palaiseau, France, boscain@cmap.polytechnique.fr

Luc Brun Image Team, Laboratoire GREYC, 6 Bd Maréchal Juin, 14050 Caen Cedex/France, Luc.Brun@greyc.ensicaen.fr

Françoise Dibos LAGA, Institut Galilée, Université Paris 13, 99 avenue J-B Cément, 93430 Villetaneuse, France, dibos@math.univ-paris13.fr

Sloven Dubois Laboratoire Mathématiques, Image et Applications (MIA) et Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France, sloven.dubois01@univ-lr.fr

Jean Duplaix Laboratoire LSIS, Université de Toulon, France, duplaix@univ-tln.fr

Olivier Faugeras NeuroMathComp Project Team, INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles, 06902 Sophia Antipolis, France, Olivier.Faugeras@inria.fr

Florian Frühauf Hauptstrasse 1c/1, 6074, Rimm, Austria, florian.fruehauf@uibk.ac.at

Jean-Paul Gauthier Laboratoire LSIS, Université de Toulon, France, gauthier@univ-tln.fr

Stéphane Jaffard Laboratoire d'Analyse et de Mathématiques Appliquées, UMR 8050 du CNRS, Université Paris Est Créteil Val de Marne, 61 Avenue du Général de Gaulle, 94010 Créteil Cedex, France, jaffard@u-pec.fr

Pierre Kornprobst NeuroMathComp Project Team, INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles, 06902 Sophia Antipolis, France, Pierre.Kornprobst@inria.fr

Guillaume S. Masson INCM, UMR 6193 CNRS-Université de la Méditerranée 31, chemin Joseph Aiguier 13402 Marseille cedex, France, guillaume.masson@incm.cnrs-mrs.fr

Michel Ménard Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France, michel.menard@univ-lr.fr

Renaud Péteri Laboratoire Mathématiques, Image et Applications (MIA), Avenue Michel Crépeau, 17042 La Rochelle, France, renaud.peteri@univ-lr.fr

Loïc Piffet Université d'Orléans, Laboratoire MAPMO - UMR 6628, Fédération Denis-Poisson, Université d'Orléans, France, loic.piffet@univ-orleans.fr

Carsten Pontow Computational Science Center, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria, carsten.pontow@umit.at

Frédéric Richard Université Paris Descartes, Laboratoire MAP5 (CNRS UMR 8145), 45 rue des Saints Pères, 75270 Paris Cedex 06, France, frederic.richard@parisdescartes.fr

Francesco Rossi Laboratoire LSIS, Université Paul Cézanne, Marseille, France, francesco.rossi@lsis.org

Stéphane Roux CNRS UMR 5672 Laboratoire de Physique, ENS de Lyon, 46, allée d'Italie, F-69364 Lyon cedex, France, stephane.roux@ens-lyon.fr

Otmar Scherzer Computational Science Center, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria, otmar.scherzer@uibk.ac.at

Emilien Tlapale NeuroMathComp Project Team, INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles, 06902 Sophia Antipolis, France, Emilien.Tlapale@inria.fr

David Tschumperlé Image Team, Laboratoire GREYC, 6 Bd Maréchal Juin, 14050 Caen Cedex, France, David.Tschumperle@greyc.ensicaen.fr

Minh Phuong Tran Université d'Orléans, Laboratoire MAPMO - UMR 6628, Fédération Denis-Poisson, Université d'Orléans, France, minh-phuong.tran@univ-orleans.fr

Function Spaces Vs. Scaling Functions: Tools for Image Classification

Stéphane Jaffard, Patrice Abry, and Stéphane Roux

Abstract We investigate the properties of several classes of new parameters issued from multifractal analysis and used in image analysis and classification. They share the following common characteristics: They are derived from local quantities based on wavelet coefficients; at each scale, l^p averages of these local quantities are performed and exponents are deduced from a regression through the scales on a log–log plot. This yields scaling functions, which depend on the parameter p , and are used for model selection and classification. We expose possible variants, and their pros and cons. We relate the values taken by these scaling functions with the determination of the regularity of the image in some classes of function spaces, and we show that looking for richer criteria naturally leads to the introduction of new classes of function spaces. We will show which type of additional information this information yields for the initial image.

1 Introduction

Tools supplied by fractal geometry have been widely used in image processing in order to derive parameters of fractal nature, which can be used for classification, model selection and parameter fitting. Fractal objects often present two related aspects: One is analytic, and consists in scale invariance properties, and the other is geometric and is expressed by a fractional dimension: For instance,

Stéphane Jaffard (✉)

Laboratoire d'Analyse et de Mathématiques Appliquées, UMR 8050 du CNRS, Université Paris Est Créteil Val de Marne, 61 Avenue du Général de Gaulle, 94010 Créteil Cedex, France
e-mail: jaffard@u-pec.fr

Patrice Abry and Stéphane Roux

CNRS UMR 5672 Laboratoire de Physique, ENS de Lyon, 46, allée d'Italie,
F-69364 Lyon cedex, France
e-mail: patrice.abry@ens-lyon.fr, stephane.roux@ens-lyon.fr

the one-dimensional Brownian motion is scale invariant: $B(ax)$ has the same law as $a^{1/2}B(x)$, and its sample paths have fractional dimension $3/2$.

Similarly, the *Weierstrass–Mandelbrot functions*

$$W_{a,b}(x) = \sum_{n=-\infty}^{+\infty} \frac{\sin(a^n x)}{b^n} \quad (a > b > 1)$$

are selfsimilar of exponent $\alpha = \log b / \log a$ since $W_{a,b}(ax) = a^\alpha W_{a,b}(x)$, and the box dimensions of their graphs (see definition below) is $2 - \alpha$.

Let us consider the geometric aspect, which is supplied by fractal dimensions. The simplest notion of dimension which can be used is the *box dimension*:

Definition 1. Let A be a bounded subset of \mathbb{R}^d ; if $\varepsilon > 0$, let $N_\varepsilon(A)$ denote the smallest number of balls of radius ε required to cover A .

The upper and lower box dimension of A are respectively given by

$$\overline{\dim}_B(A) = \limsup_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon(A)}{-\log \varepsilon}, \quad \text{and} \quad \underline{\dim}_B(A) = \liminf_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon(A)}{-\log \varepsilon}.$$

The following important feature makes this notion useful in practical applications: If both limits coincide, then the box dimension can be computed through a regression on a log–log plot ($\log N_\varepsilon(A)$ vs. $\log \varepsilon$):

$$\dim_B(A) = \lim_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon(A)}{-\log \varepsilon}. \quad (1)$$

As such, this tool has a rather narrow field of applications in image processing; indeed, it applies only when a particular “fractal” set can be isolated in the image, with a sufficient resolution. However, the fact that the limit exists in (1) points towards another possible feature, which is much more common: Some quantities display an approximate power-law behavior through the scales. Such a property may hold for many other quantities than $N_\varepsilon(A)$ and the associated power-law exponents can thus yield parameters which can be used in image classification; this points towards the “analytic aspect” that we mentioned. The study of such quantities is usually referred to as *multifractal analysis*; the purpose of this field is:

- To introduce new quantities which present such power-law behaviors
- To study their mathematical properties, and relate them with scales of function spaces
- To determine geometric implications, as regards the presence of “fractal features” in the image.

This paper is divided as follows: In Sect. 2 we review the different quantities which are related with function spaces, and have been used for signal or image classification. We explain their relationships with function space indicators, and

in particular with Sobolev and Besov spaces. Section 3 deals with new classes of parameters, which are built using quantities related with different types of pointwise regularity criteria: The local L^q behavior of the function (Sect. 3.1), the regularization performed by fractional integration (Sect. 3.2), or a mixture of both approaches (Sect. 3.3). Finally Sect. 4 deals with some general results concerning grand canonical formalisms, which are required when one takes into account simultaneously two different pointwise criteria. We will develop the case where one of the exponents is a regularity index, and the other measures the local oscillations of f near a point, see Sect. 4.1.

2 Scaling Functions and Function Spaces

Let us now briefly review the different quantities which have been used up to now as possible candidates for scale invariance features. First, these quantities usually depend on (at least) one auxiliary parameter p (see e.g. (2) below), and therefore the exponents which are derived are not given by one real number (or a few), but are functions of this parameter p , hence the term *scaling functions* used in order to characterize these collections of exponents. Note that the use of a whole function in order to perform classification yields a potentially much richer tool than the use of one single number.

Let us now be more specific and start with what was historically the first example of a scaling function. It was introduced by N. Kolmogorov in the context of fully developed turbulence, with a motivation which was quite similar to ours, see [42]: Take advantage of the (expected) scale invariance of fully developed turbulence in order to derive a collection of “universal” parameters which could be computed on experimental data, and used for the validation of turbulence models.

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$. The *Kolmogorov scaling function* of f is the function $\eta_f(p)$ which satisfies:, for h small,

$$\int |f(x+h) - f(x)|^p dx \sim |h|^{\eta_f(p)}. \quad (2)$$

This is mathematically formalized by

$$\eta_f(p) = \liminf_{|h| \rightarrow 0} -\frac{1}{\log |h|} \log \left(\int |f(x+h) - f(x)|^p dx \right). \quad (3)$$

As in the case of the box dimension, we have to draw a distinction between the mathematical definition, whose purpose is to make sense in a general setting (e.g. whenever f is locally bounded), and the numerical evaluation of $\eta_f(p)$ which requires that the liminf is a real limit and, in practice, that one can make a precise regression on the scales available in the data. Note that the mathematical hypotheses bearing on f such that such scaling invariance holds are far from being well understood.

This first scaling function has a function space interpretation which will serve several purposes. First, it allows to derive several of its mathematical properties, but its main advantage will be to point the way towards variants and extensions of this scaling function which yield sharp information on the singularities present in the signal. This last motivation had unexpected consequences: For instance such new scaling functions allow to show the presence of “oscillating singularities” in the data, which was an important open issue in several applications, see [5, 29].

The most straightforward function space interpretation of the scaling function is obtained through the use of the spaces $\text{Lip}(s, L^p)$ defined as follows.

Definition 2. Let $s \in (0, 1)$, and $p \in [1, \infty]$; $f \in \text{Lip}(s, L^p(\mathbb{R}^d))$ if

$$f \in L^p \quad \text{and} \quad \exists C > 0, \forall h > 0, \quad \int |f(x+h) - f(x)|^p dx \leq C|h|^{sp}. \quad (4)$$

It follows from (2) and this definition that, if $\eta_f(p) < p$, then

$$\eta_f(p) = \sup\{s : f \in \text{Lip}(s/p, L^p(\mathbb{R}^d))\}. \quad (5)$$

In other words, the scaling function allows to determine which spaces $\text{Lip}(s, L^p)$ contain the signal for $s \in (0, 1)$, and $p \in [1, \infty]$. This reformulation has several advantages:

- Using classical embeddings between function spaces, one can derive alternative formulations of the scaling function which, though they are mathematically equivalent to (2) or (5), allow a better numerical implementation
- It has extensions outside of the range $s \in (0, 1)$, and therefore applies to signals which are either smoother or rougher than allowed by this range
- Extensions of these function spaces to $p < 1$ lead to a scaling function defined for $p < 1$, and therefore supply a richer tool for classification

The simplest setting for these extension is supplied by Besov spaces; they offer the additional advantage of yielding a wavelet reformulation of the scaling function, which is simple and robust, see [1, 3, 4, 33]. In order to define these spaces, we need to recall the notion of wavelet bases.

2.1 Orthonormal and Biorthogonal Wavelet Bases

Orthonormal wavelet bases are a privileged tool to study multifractal functions for several reasons. A first one, exposed in this section, is that classical function spaces (such as Besov or Sobolev spaces) can often be characterized by conditions bearing on the wavelet coefficients, see Sect. 2.2. We will only recall the properties of orthonormal and biorthogonal wavelet bases that will be useful in the sequel. We refer the reader for instance to [14, 15, 35, 36] for detailed expositions.

Orthonormal wavelet bases are of the following form: There exists a function $\varphi(x)$ and $2^d - 1$ functions $\psi^{(i)}$ with the following properties: The functions $\varphi(x - k)$

($k \in \mathbb{Z}^d$) and the $2^{dj/2}\psi^{(i)}(2^jx - k)$ ($k \in \mathbb{Z}^d$, $j \in \mathbb{Z}$) form orthonormal wavelet basis of $L^2(\mathbb{R}^d)$. This basis is r -smooth if φ and the $\psi^{(i)}$ are C^r and if the $\partial^\alpha \varphi$, and the $\partial^\alpha \psi^{(i)}$, for $|\alpha| \leq r$, have fast decay.

Therefore, $\forall f \in L^2$,

$$f(x) = \sum_{k \in \mathbb{Z}^d} c_k^{(0)} \varphi(x - k) + \sum_{j=0}^{\infty} \sum_{k \in \mathbb{Z}^d} \sum_i c_{j,k}^i \psi^{(i)}(2^jx - k); \quad (6)$$

the $c_{j,k}^i$ and $c_k^{(0)}$ are called the wavelet coefficients of f and given by

$$c_{j,k}^i = 2^{dj} \int_{\mathbb{R}^d} f(x) \psi^{(i)}(2^jx - k) dx, \text{ and } c_k^{(0)} = \int_{\mathbb{R}^d} f(x) \varphi(x - k) dx.$$

Note that the computation of these coefficients makes sense with very little assumption on f (a wide mathematical setting is supplied by tempered distributions). A natural setting for functions is supplied by the space L^1 with slow growth which is defined as follows.

Definition 3. Let f be a locally integrable function defined over \mathbb{R}^d ; f belongs to $L_{SG}^1(\mathbb{R}^d)$ if

$$\exists C, N > 0 \text{ such that } \int_{\mathbb{R}^d} |f(x)| (1 + |x|)^{-N} dx \leq C.$$

The wavelet expansion of a function $f \in L_{SG}^1(\mathbb{R}^d)$ converges a.e.; in particular at Lebesgue points, it converges towards the Lebesgue value

$$\lim_{r \rightarrow 0} \frac{1}{\text{Vol}(B(x_0, r))} \int_{B(x_0, r)} f(x) dx.$$

Furthermore, let $C_{SG}(\mathbb{R}^d)$ be the set of locally bounded and continuous functions which satisfy

$$\exists C, N > 0 : \quad |f(x)| \leq C(1 + |x|)^N.$$

Then, if $f \in C_{SG}(\mathbb{R}^d)$, then its wavelet expansion converges uniformly on compact sets.

We will also need decompositions on *biorthogonal wavelet bases*, which are a useful extension of orthonormal wavelet bases. Recall that a *Riesz basis* of a separable Hilbert space H is a collection of vectors (e_n) such that the finite linear expansions $\sum_{n=1}^N a_n e_n$ are dense in H and

$$\exists C, C' > 0 : \quad \forall N, \quad \forall a_n, \quad C \sum_{n=1}^N |a_n|^2 \leq \left\| \sum_{n=1}^N a_n e_n \right\|_H^2 \leq C' \sum_{n=1}^N |a_n|^2. \quad (7)$$

Two collections of functions (e_n) and (f_n) form *biorthogonal bases* if each collection is a Riesz basis, and if $\langle e_n | f_m \rangle = \delta_{n,m}$. When this is the case, any element $f \in H$ can be written $f = \sum \langle f | f_n \rangle e_n$ (where the series converges in H). Biorthogonal wavelet

bases are couples of Riesz bases of L^2 which are of the following form: On one side,

$$\varphi(x-k), \quad (k \in \mathbb{Z}^d) \quad \text{and} \quad 2^{dj/2} \psi^{(i)}(2^j x - k), \quad (k \in \mathbb{Z}^d, j \in \mathbb{Z})$$

and, on the other side,

$$\tilde{\varphi}(x-k) \quad (k \in \mathbb{Z}^d) \quad \text{and} \quad 2^{dj/2} \tilde{\psi}^{(i)}(2^j x - k), \quad (k \in \mathbb{Z}^d, j \in \mathbb{Z}).$$

Therefore, $\forall f \in L^2$,

$$f(x) = \sum_{k \in \mathbb{Z}^d} c_k^{(0)} \varphi(x-k) + \sum_{j=0}^{\infty} \sum_{k \in \mathbb{Z}^d} \sum_i c_{j,k}^i \psi^{(i)}(2^j x - k); \quad (8)$$

where

$$c_{j,k}^i = 2^{dj} \int_{\mathbb{R}^d} f(x) \tilde{\psi}^{(i)}(2^j x - k) dx, \quad \text{and} \quad c_k^{(0)} = \int_{\mathbb{R}^d} f(x) \tilde{\varphi}(x-k) dx. \quad (9)$$

We will use more compact notations for indexing wavelets. Instead of using the three indices (i, j, k) , we will use dyadic cubes. Since i takes $2^d - 1$ values, we can assume that it takes values in $\{0, 1\}^d - (0, \dots, 0)$; we introduce

$$\lambda \quad (= \lambda(i, j, k)) = \frac{k}{2^j} + \frac{i}{2^{j+1}} + \left[0, \frac{1}{2^{j+1}}\right)^d,$$

and, accordingly: $c_\lambda = c_{j,k}^i$ and $\psi_\lambda(x) = \psi^{(i)}(2^j x - k)$. Indexing by dyadic cubes will be useful in the sequel because the cube λ indicates the localization of the corresponding wavelet. Note that this indexing is injective: if $(i, j, k) \neq (i', j', k')$, then $\lambda(i, j, k) \neq \lambda(i', j', k')$. Dyadic cubes have the remarkable property that two of them are either disjoint, or one is included in the other.

The wavelet ψ_λ is essentially localized near the cube λ ; more precisely, when the wavelets are compactly supported, then, $\exists C > 0$ such that when $\psi^{(i)} \subset [-C/2, C/2]^d$ then $\psi_\lambda \subset 2^{-j}k + 2^{-j}[-C/2, C/2] \subset 2C\lambda$.

Remark 1. Two classes of orthonormal wavelet bases play a prominent role: For theoretical purposes, the wavelets introduced by Y. Meyer and P.-G. Lemarié, which belong to the Schwartz class (but are not compactly supported), and Daubechies wavelets which are compactly supported, but of limited regularity (the size of the support essentially is a linear function of the regularity of the wavelet), see [16, 35, 36].

In order to have a common notation for wavelets and functions φ , when $j = 0$, we note ψ_λ the function $\varphi(x-k)$ (where λ is, in this case, the unit cube shifted by k), and by c_λ the corresponding coefficient.

Finally, Λ_j will denote the set of dyadic cubes λ which index a wavelet of scale j , i.e., wavelets of the form $\psi_\lambda(x) = \psi^{(i)}(2^j x - k)$ (note that Λ_j is a subset of the dyadic cubes of side 2^{j+1}), and Λ will denote the union of the Λ_j for $j \geq 0$.

2.2 The Wavelet Scaling Function

A remarkable property of wavelet bases is that they supply bases for most function spaces that are used in analysis. The case of Besov spaces is typical:

Definition 4. Let $s \in \mathbb{R}$ and $p, q \in (0, \infty]$. Let ψ_λ be an r -smooth wavelet basis with $r > \sup(|s|, |s| + d(\frac{1}{p} - 1))$. A distribution f belongs to the Besov space $B_p^{s,q}(\mathbb{R}^d)$ if and only if its wavelet coefficients satisfy

$$\sum_{j \geq 0} \left(\sum_{\lambda \in \Lambda_j} \left[2^{(s-d/p)j} |c_\lambda| \right]^p \right)^{q/p} < \infty \quad (10)$$

(using the usual convention for l^∞ when p or q is infinite).

Remark 2. Historically, this was not the first definition of Besov spaces, since they were initially introduced as interpolation spaces between Sobolev spaces. However, this very simple characterization has opened the way to an extensive use of Besov spaces in many fields, including image processing and statistics, so that it is now rather taken as a definition, see [36]. Informally, it essentially means that f and its fractional derivative of order s belong to L^p (and, indeed, $B_2^{s,2}$ coincides with the Sobolev space H^s), see [36] for precise embeddings with Sobolev spaces. Let $s \in (0, 1)$, and $p \in [1, \infty]$; then the following embeddings hold:

$$B_p^{s,1} \hookrightarrow \text{Lip}(s, L^p(\mathbb{R}^d)) \hookrightarrow B_p^{s,\infty}. \quad (11)$$

Furthermore, if $s \geq 0$, $p > 0$ and $0 < q_1 < q_2$, then we have the embeddings

$$\forall \varepsilon > 0, \quad B_p^{s+\varepsilon,\infty} \hookrightarrow B_p^{s,q_1} \hookrightarrow B_p^{s,q_2} \hookrightarrow B_p^{s,\infty}, \quad (12)$$

which allow to redefine Kolmogorov's scaling function for $p \geq 1$ by,

$$\eta_f(p) = \sup\{s : f \in B_p^{s/p,\infty}\}. \quad (13)$$

There are two advantages in doing so: on one hand, it extends the scaling function to all values of $p > 0$, on the other hand, it suggests an alternative way to compute it, through a regression based on wavelet coefficients. Indeed, it follows from (10) and (13) that the scaling function of f is

$$\forall p > 0, \quad \eta_f(p) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |c_\lambda|^p \right). \quad (14)$$

We will use the same notation for the Kolmogorov scaling function and the wavelet scaling function since they coincide for a range of values; this slight abuse of notations will lead to no confusion. From now on, we take (14) for definition of

the *wavelet scaling function*, and also, as a practical way to compute it through a regression on a log–log plot. Note that its interpretation in terms of function spaces implies that it is independent of the (smooth enough) wavelet basis. A similar formula was previously introduced by A. Arneodo, E. Bacry and J.-F. Muzy, using the continuous wavelet transform, see for instance [8]. An additional advantage of using (14) as a definition is that the scaling function is well defined even if f is not a function (in the most general case, it can be a tempered distribution); a large degree of flexibility may prove necessary: Indeed, a picture is a discretization of the light intensity, which is a non-negative quantity. Therefore the most general mathematical modeling which takes into account only this a priori information amounts to make the assumption that f is a measure (indeed non-negative distributions necessarily are measures, by a famous theorem of L. Schwartz); a posteriori estimation of the smoothness of images using the wavelet scaling function showed that, indeed, some types of natural images are not smoother than measures are, see [2, 47].

2.3 The Uniform Hölder Exponent

We consider now the particular case of Besov spaces with $p = q = +\infty$, which will play an important role in the following: f belongs to $B_{\infty}^{s,\infty}(\mathbb{R}^d)$ if and only if $(c_k^{(0)}) \in l^{\infty}$ and

$$\exists C, \forall \lambda \in \Lambda, \quad |c_{\lambda}| \leq C 2^{-sj}. \quad (15)$$

The spaces $B_{\infty}^{s,\infty}$ coincide with the Lipschitz spaces $C^s(\mathbb{R}^d)$ when $s \notin \mathbb{N}$ (for instance, if $s = 1$, then (15) characterizes function in the Zygmund class, see [36]); it follows that, if $0 < s < 1$, $f \in B_{\infty}^{s,\infty}(\mathbb{R}^d)$ if f is bounded and

$$\exists C, \forall x, y, |f(x) - f(y)| \leq C|x - y|^s.$$

The *uniform Hölder exponent* of f is

$$H_f^{\min} = \sup\{s : f \in C^s(\mathbb{R}^d)\}. \quad (16)$$

It yields an additional parameter for image processing and classification that will prove important for additional reasons: Its determination will be a mandatory step for multifractal analysis; note that H_f^{\min} is related with the scaling function according to $H_f^{\min} = \lim_{p \rightarrow +\infty} \eta'_f(p)$, see [25]. In practice, it is derived directly from the wavelet coefficients of f through a simple regression on a log–log plot ; indeed, it follows from (16) and the wavelet characterization of the Besov spaces $B_{\infty}^{s,\infty}$ that

$$H_f^{\min} = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(\sup_{\lambda \in \Lambda_j} |c_{\lambda}| \right). \quad (17)$$

Note that the determination of H_f^{min} does not require any a priori assumption on f and that, in practice, it allows to settle if f is bounded or not. Indeed, it follows from (16) that:

- If $H_f^{min} > 0$, then f is locally bounded
- If $H_f^{min} < 0$, then $f \notin L_{loc}^\infty$

We will meet several situations where this a priori information is needed.

Let us consider now two simple models of random fields. The first one will show no bias in the estimation of H_f^{min} , and the other will show one.

The first model is supplied by Lacunary Wavelet Series (LWS) $X_{\alpha,\gamma}$ of type (α, γ) , see [23]; they are random fields defined on $[0, 1]^d$ (for $\alpha > 0$ and $\gamma < d$) as follows: A biorthogonal wavelet basis in the Schwartz class is used for the construction. One draws at random (uniformly) $2^{\gamma j}$ locations λ among the 2^{dj} dyadic cubes of width 2^{-j} included in $[0, 1]^d$, and the corresponding wavelet coefficients are set to the value $2^{-\alpha j}$, whereas the others are set to 0. In order to define LWS on the whole \mathbb{R}^d , one repeats this construction on all cubes of width 1. In this case it follows from (17) that $H_f^{min} = \alpha$ and (17) yields α exactly at each scale.

The second example falls in the general model of *Random Wavelet Series*. Let us describe the particular case that we consider now.

Definition 5. Let ψ_λ be a biorthogonal wavelet basis in the Schwartz class. A Uniform Random Wavelet Series (URWS) of type (α, β) is a random field of the form

$$X = \sum 2^{-\alpha j} X_\lambda \psi_\lambda,$$

where the X_λ are IID with common law, which is a non-vanishing random variable satisfying a tail estimate of the form

$$\mathbb{P}(|X| \geq A) \sim C \exp(-B|A|^\beta),$$

and C, B and β are positive constants.

Note that this model includes the Fractional Brownian Motion (FBM) of exponent $\gamma \in (0, 1)$. Indeed, if $s \in \mathbb{R}$, let $\hat{\psi}_s(\xi) = |\xi|^{-s} \hat{\psi}(\xi)$, where ψ is generating an orthonormal wavelet basis (ψ_s is the fractional integral of ψ of order s). If $\psi \in \mathcal{S}(\mathbb{R})$, then ψ_s is a wavelet and the $2^{j/2} \psi_s(2^j x - k)$ and the $2^{j/2} \psi_{-s}(2^j x - k)$ form biorthogonal bases, see e.g. [25]. If $t \in [0, 1]$ then

$$B_\gamma(t) = \sum_{j=0}^{\infty} \sum_{k \in \mathbb{Z}} 2^{-\gamma j} \xi_{j,k} \psi_{\gamma+1/2}(2^j t - k) + R(t) \quad (18)$$

where $R(t)$ is a C^∞ random process, and the $\xi_{j,k}$ are I.I.D. standard centered Gaussians, see [6, 39].

If $\beta > 1$, then the model supplied by URWS yields a random field, with a constant Hölder exponent α , see [10]. Since the X_λ are independent, one obtains that

$$\mathbb{P}\left(\sup_{\lambda \in \Lambda_j} |X_\lambda| \leq A\right) \sim 1 - C \exp(j \log 2 - BA^\alpha);$$

therefore, the quantity $\sup_{\lambda \in \Lambda_j} |X_\lambda|$ is asymptotically equivalent to $Cj^{1/\alpha}$. It follows that

$$-\frac{1}{j} \log_2 \left(\sup_{\lambda \in \Lambda_j} |c_\lambda| \right) = \alpha - \frac{\log_2 j}{\alpha j} (1 + o(1));$$

therefore the statistical estimator of H_f^{min} supplied by $-\log_2(\sup |c_\lambda|)/j$ is biased by a term equivalent to $(\log_2 j)/j$.

Note that, from the point of view of statistical estimation, the case of the determination of H^{min} is quite different from the determination of the scaling function. Indeed, for URWS, if $C_p = \mathbb{E}(|X|^p)$, then

$$\mathbb{E}\left(2^{-dj} \sum |C_\lambda|^p\right) = C_p 2^{-\alpha j} \text{ and } \text{Var}\left(2^{-dj} \sum |C_\lambda|^p\right) = 2^{-dj} C_{2p} 2^{-2\alpha j}.$$

It follows that no such bias exists for the estimation of the scaling function of URWS: It is at most C/j , i.e., smaller than for the uniform Hölder exponent by a logarithmic term.

2.4 Scaling Functions for Image Model Validation

We start by recalling a general problem posed by function-space modeling when applied to real-life signals: Data are always available with a finite resolution; therefore, assuming that images are functions (or perhaps distributions) continuously defined on \mathbb{R}^2 (or a subset of \mathbb{R}^2 such as a square or a rectangle) is an idealization which may be convenient for mathematical modeling, but should not mask the fact that real-life images are sampled and given by a finite array of numbers. Therefore, the problem of finding which function spaces contain a particular image is ill-posed: Indeed, given any “classical” space of functions defined on a square, and such an array of numbers, one can usually find a function in this space that will have the preassigned values at the corresponding points of the grid. For instance, a commonly met pitfall is that an image is given by grey-levels, and thus takes values in $[0, 1]$. Therefore, it may seem appropriate to use a modeling by bounded functions. We will see that wavelet techniques allow to discuss this assumption, and show that it is not satisfied for large classes of natural textures.

The resolution of these problems requires the use of *multiscale techniques* as supplied by wavelet analysis. Let us consider for instance the last example we mentioned: Starting with a discrete image, given by an array of 1024×1024 numbers all lying between 0 and 1, how can we decide that it can be modeled or not by a bounded function? It is clear that, if we consider the image at only one scale

(the finest scale in order to lose no information), then the answer seems to be affirmative. One way to solve the difficulty is to consider the image simultaneously at all the scales available and inspect if certain quantities behave through this range of scales as bounded functions do. The practical recipe in this case is to use (17) in order to determine numerically the value of H_f^{min} , through a regression on a log–log plot; if $H_f^{min} < 0$, then the image is not bounded, and if $H_f^{min} > 0$, then the image is bounded. Of course, if the numerical value obtained for H_f^{min} is close to 0 (i.e., if 0 is contained in the confidence interval which can be obtained using statistical methods such as the bootstrap, see [45, 47]) then the issue remains unsettled.

The wavelet scaling function yields an extensive information on the function spaces which contain a particular signal. As an example of its use, we will discuss the assumption that real-life images can be modeled as a sum of a function $u \in BV$ and another term v which will model noise and texture parts. There is no consensus on which regularity should be assumed for the second term v . The first “ $u + v$ model” (introduced by Rudin, Osher and Fatemi [43]) assumed that $v \in L^2$; however, the very strong oscillations displayed by some textures suggested that such components do not have a small L^2 norm, but might have a small norm in spaces of negative regularity index (i.e., spaces of distributions). Therefore the use of spaces such as divergences of L^∞ functions (or divergences of L^2 functions) were proposed by Y. Meyer, see [38], followed by several other authors, see [11, 41] and references therein. More sophisticated models also aim at separating noise from texture, and therefore split the image into three components ($u + v + w$ models, see [11]).

The Rudin–Osher–Fatemi algorithm consists in obtaining the regular component u of an image f by minimizing the functional $J(u) = \|u\|_{BV} + t \|f - u\|_2^2$, where t is a scale parameter which has to be tuned. In 2001, Y. Meyer proposed to minimize the alternative functional $J(u) = \|u\|_{BV} + t \|f - u\|_G$, where $\|f\|_G = \inf_{g: f = \nabla \cdot g} \|g\|_\infty$. More recently, in 2003, Osher, Solé and Vese proposed another model which uses for the texture and noise component a space of distributions easier to handle, the Sobolev space H^{-1} . The corresponding functional is $J(u) = \|u\|_{BV} + t \|f - u\|_{H^{-1}}^2$.

More recently, several variants have been proposed, based on the same fundamental ideas, but using alternative function spaces. The relevance of one particular function space is usually advocated using either theoretical arguments derived from functional analysis, or practical arguments motivated by the algorithmic implementation. The fundamental problem of determining to which function spaces a given image (or a part of a given image) belongs to has been rarely considered (see however [19] where the authors question the fact that natural images belong to BV). A first motivation for this question rises implicitly from the short review we just performed: The function spaces used in modeling should fit the data. Another motivation is that, if these function spaces depend strongly on the image that is considered, then this information might prove useful in image classification. The knowledge of the scaling function allows to settle the issues we raised concerning the function spaces which contain a given image. The following result is a direct consequence of the embeddings (11) and (12) between Besov spaces and other classical function spaces, and allows to settle, for a given image, if the models mentioned above fit the data.

Corollary 1. *Let f be a tempered distribution defined on \mathbb{R}^2 . Then, the values taken by $\eta_f(p)$ and by H_f^{\min} have the following interpretation:*

- If $\eta_f(1) > 1$, then $f \in BV$, and if $\eta_f(1) < 1$, then $f \notin BV$.
- If f is a measure, then $\eta_f(1) \geq 0$, and, if $\eta_f(1) > 0$, then f belongs to L^1 .
- If $\eta_f(2) > 0$, then $f \in L^2$ and if $\eta_f(2) < 0$, then $f \notin L^2$.
- If $\eta_f(2) > -2$, then $f \in H^{-1}$ and if $\eta_f(2) < -2$, then $f \notin H^{-1}$.
- If $H_f^{\min} > 0$, then f is bounded and continuous, and if $H_f^{\min} < 0$, then $f \notin L^\infty$.
- If $H_f^{\min} > -1$, then $f \in G$ and if $H_f^{\min} < -1$, then $f \notin G$.
- If f is a measure, then $H_f^{\min} \geq -2$.

2.5 Multifractal Formalism

Later refinements and extensions of the wavelet scaling function were an indirect consequence of its interpretation in terms of fractal dimensions of Hölder singularities, proposed by G. Parisi and U. Frisch in their seminal paper [42]. In order to explain their argumentation, we first recall the definition associated with pointwise regularity. The most widely used one is supplied by *Hölder regularity*.

Definition 6. Let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ be a locally bounded function, $x_0 \in \mathbb{R}^d$ and let $\alpha \geq 0$; f belongs to $C^\alpha(x_0)$ if there exist $C > 0$, $R > 0$ and a polynomial P of degree less than α such that, if $|x - x_0| \leq R$, then $|f(x) - P(x - x_0)| \leq C|x - x_0|^\alpha$.

The Hölder exponent of f at x_0 is: $h_f(x_0) = \sup \{ \alpha : f \text{ is } C^\alpha(x_0) \}$.

The isohölder sets are: $E_H = \{x_0 : h_f(x_0) = H\}$.

Note that Hölder exponents met in signal processing often lie between 0 and 1, in which case the Taylor polynomial $P(x - x_0)$ boils down to $f(x_0)$ and the definition of the Hölder exponent means that, heuristically, $|f(x) - f(x_0)| \sim |x - x_0|^{h_f(x_0)}$. The loose idea which lies at the starting point of the derivation proposed in [42] is that, if f is not smooth on a large set of locations, then, at a given scale h , the increments $f(x + h) - f(x)$ will bring a large contribution to (2), and therefore the knowledge of the scaling function should yield some information of the size of the sets where f has a given Hölder regularity. A statistical physics argument leads to a precise statement, usually referred to as the *multifractal formalism* concerning the size of the sets of singularities of f . In order to recall it, we start by giving the notion of “size” which is adapted to this problem, see [17].

Definition 7. Let $A \subset \mathbb{R}^d$. If $\varepsilon > 0$ and $\delta \in [0, d]$, we denote

$$M_\varepsilon^\delta = \inf_R \left(\sum_i |A_i|^\delta \right),$$

where R is an ε -covering of A , i.e., a covering of A by bounded sets $\{A_i\}_{i \in \mathbb{N}}$ of diameters $|A_i| \leq \varepsilon$. The infimum is therefore taken on all ε -coverings.

For any $\delta \in [0, d]$, the δ -dimensional Hausdorff measure of A is $mes_\delta(A) = \lim_{\varepsilon \rightarrow 0} M_\varepsilon^\delta$. There exists $\delta_0 \in [0, d]$ such that $\forall \delta < \delta_0$, $mes_\delta(A) = +\infty$ and $\forall \delta > \delta_0$, $mes_\delta(A) = 0$. This critical δ_0 is called the Hausdorff dimension of A , and is denoted by $\dim(A)$ (by convention, $\dim(\emptyset) = 0$).

If f is locally bounded, then the function $H \rightarrow \dim(E_H)$ is called the *spectrum of singularities* of f . Note that, in distinction with the box dimension, the Hausdorff dimension cannot be computed via a regression on a log-log plot; therefore it can be estimated on experimental data only through indirect methods. We will see that the use of the multifractal formalism is one of them.

The formula proposed by Paris and Frisch is

$$\dim(E_H) = \inf_{p \in \mathbb{R}} (d + Hp - \eta_f(p)), \quad (19)$$

see [42]. However, it meets severe limitations: Many natural processes used in signal or image modeling, are counterexamples, see [1]; additionally, the only result relating the spectrum of singularities and the scaling function in all generality is partial, and stated below, see [22, 25].

Theorem 1. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be such that $H_f^{min} > 0$. Define p_0 by the condition:*

$$\eta_f(p_0) = dp_0,$$

then

$$\dim(E_H) \leq \inf_{p > p_0} (d + Hp - \eta_f(p)). \quad (20)$$

2.6 Wavelet Leaders

A natural line of research is to look for an “improved” scaling function, i.e., one such that (19) would have a wider range of validity, and for which the upper bound supplied by Theorem 1 would be sharper. This led to the construction of the *wavelet leader scaling function*, which we now recall. The “basic ingredients” in this formula are no more wavelet coefficients, but wavelet leaders, i.e., local suprema of wavelet coefficients. The reason is that pointwise smoothness can be expressed much more simply in terms of wavelet leaders than of wavelet coefficients.

Let λ be a dyadic cube; 3λ will denote the cube of same center and three times wider. If f is a bounded function, the *wavelet leaders* of f are the quantities

$$d_\lambda = \sup_{\lambda' \subset 3\lambda} |c_{\lambda'}|$$

It is important to require f to be bounded; otherwise, the wavelet leaders of f can be infinite; therefore checking that $H_f^{min} > 0$ is a prerequisite of the method.

Remark 3. Similar quantities were previously introduced by S. Mallat, in the *Wavelet Transform maxima method*, which can be seen as counterpart of the wavelet leader technique for the continuous wavelet transform; it was used in multifractal analysis by A. Arneodo, E. Bacry and J.-F. Muzy, see [8, 35] and references therein; however it does not enjoy the same simplicity of implementation and very few mathematical results apply to it.

The reason for introducing wavelet leaders is that they give an information on the pointwise Hölder regularity of the function. Indeed, let $x_0 \in \mathbb{R}^d$, and denote by $\lambda_j(x_0)$ is the dyadic cube of width 2^{-j} which contains x_0 . If $H_f^{\min} > 0$, then

$$h_f(x_0) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(d_{\lambda_j(x_0)} \right) \quad (21)$$

(see [25] and references therein). Therefore, constructing a scaling function with the help of wavelet leaders is a way to incorporate pointwise smoothness information. It is therefore natural to expect that (20) will be improved when using such a scaling function instead of $\eta_f(p)$.

The *leader scaling function* is defined by

$$\forall p \in \mathbb{R}, \quad \zeta_f(p) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |d_\lambda|^p \right). \quad (22)$$

An important property of the leader scaling function is that it is “well defined” for $p < 0$, although it can no more be subject to a function space interpretation. By “well defined”, we mean that it has the following robustness properties if the wavelets belong to the Schwartz class (partial results still hold otherwise), see [25, 30]:

- ζ_f is independent of the wavelet basis.
- ζ_f is invariant under the addition of a C^∞ perturbation.
- ζ_f is invariant under a C^∞ change of variable.

Note that the wavelet scaling function does not possess these properties when p is negative. The leader scaling function can also be given a function-space interpretation for $p > 0$: Let $p \in (0, \infty)$; a function f belongs to the *Oscillation space* $\mathcal{O}_p^s(\mathbb{R}^d)$ if and only if

$$\forall j \geq 0, \quad \sum_{\lambda \in \Lambda_j} \left[2^{(s-d/p)j} d_\lambda \right]^p < \infty.$$

Then

$$\zeta_f(p) = \sup \{s : f \in \mathcal{O}_p^{s/p}\}.$$

Properties of oscillation spaces are investigated in [25, 30]. Note that their definition mixes two quantities of different natures: a local l^∞ norm (the leader, which is a supremum of wavelet coefficients), and a global l^p norm. Thus they are reminiscent of *amalgam spaces*, which were introduced by N. Wiener, and thoroughly studied

by H. Feichtinger, see [18]: These spaces are defined by taking local L^p norms of the function, and then averaging them through a discrete l^q sum. Note however the strong difference with oscillation spaces, where the l^∞ local norms are taken at each scale. Thus oscillation spaces do not favor a particular scale, in contradistinction with amalgam spaces, where a specific scale for the amalgam is chosen.

Let us now prove the following result, which shows that the wavelet leader scaling function can be alternatively defined through the “restricted leaders”

$$e_\lambda = \sup_{\lambda' \subset \lambda} |c_{\lambda'}|.$$

Lemma 1. *Let f be a function satisfying $H_f^{min} > 0$; then*

$$\forall p \in \mathbb{R}, \quad \zeta_f(p) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |e_\lambda|^p \right).$$

Remark 4. This equivalence is important because the formulation of the scaling function using “extended leaders” (where the supremum is taken on 3λ) is required in order to prove upper bounds for spectra, see [25]; on the other hand, the formulation using “restricted leaders” is more suited to numerical implementation, and in order to prove properties of the scaling function (since the supremums at a given scale are taken on non-overlapping cubes).

Proof. Let

$$S_f(p, j) = 2^{-dj} \sum_{\lambda \in \Lambda_j} |c_\lambda|^p \quad \text{and} \quad T_f(p, j) = 2^{-dj} \sum_{\lambda \in \Lambda_j} |e_\lambda|^p.$$

Since

$$\sup_{\lambda' \subset \lambda} |c_{\lambda'}| \leq \sup_{\lambda' \subset 3\lambda} |c_{\lambda'}|$$

it follows that, if $p \geq 0$, then $S_f(p, j) \leq T_f(p, j)$, and, if $p \leq 0$, then $T_f(p, j) \leq S_f(p, j)$.

On the other hand, denote by μ the “father” of the cube λ (i.e., the cube twice as wide which contains λ), and denote by $N(\mu)$ the 3^d “neighbours” of λ (i.e., the cubes of same width, whose boundary intersects the boundary of λ). Then

$$\sup_{\lambda' \subset 3\lambda} |c_{\lambda'}| \leq \sup_{v \in N(\mu)} \sup_{\lambda' \subset v} |c_{\lambda'}|.$$

It follows that, if $p \geq 0$, then $T_f(p, j) \leq 3^d S_f(p, j-1)$.

Finally, for any dyadic cube λ , there exists a “grandson” λ'' of λ such that $3\lambda'' \subset \lambda$. Therefore

$$\sup_{\lambda' \subset 3\lambda''} |c_{\lambda'}| \leq \sup_{\lambda' \subset \lambda} |c_{\lambda'}|;$$

therefore, if $p \leq 0$, then $S_f(p, j) \leq T_f(p, j+2)$. The result follows from these four estimates. \square

The *leader spectrum* of f is defined through a Legendre transform of the leader scaling function as follows

$$L_f(H) = \inf_{p \in \mathbb{R}} (d + Hp - \zeta_f(p)). \quad (23)$$

The following result of [25] shows the improvement obtained when using wavelet leaders: The upper bound is sharpened since, on one hand $\eta_f(p) = \zeta_f(p)$ if $p > p_0$, and, on the other hand, the infimum in (23) is taken for all $p \in \mathbb{R}$.

Theorem 2. *If $H_f^{\min} > 0$, then, $\forall H$, $\dim(E_H) \leq L_f(H)$.*

Furthermore, equality holds for large classes of models used in signal and image processing, such as Fractional Brownian Motions, lacunary and random wavelet series, cascade models,... see [1, 30] and references therein.

2.7 Scaling Function for Measures

Since nonnegative measures supply a natural setting for the modeling of images, we now expose the tools related with the multifractal analysis of nonnegative measures (see e.g. [2, 12, 20]), and we will show that they can also be related with function space interpretations.

Definition 8. Let $x_0 \in \mathbb{R}^d$ and let $\alpha \geq 0$. A nonnegative measure μ defined on \mathbb{R}^d belongs to $C^\alpha(x_0)$ if there exists a constant $C > 0$ such that, in a neighbourhood of x_0 , $\mu(B(x_0, r)) \leq Cr^\alpha$, where $B(x_0, r)$ denotes the open ball of center x_0 and radius r . Let x_0 belong to the support of μ ; then the Hölder exponent of μ at x_0 (also called the “local dimension”) is

$$h_\mu(x_0) = \sup\{\alpha : \mu \in C^\alpha(x_0)\}.$$

Let $E_\mu(H) = \{x_0 : h_\mu(x_0) = H\}$. The *spectrum of singularities* of μ (denoted by $d_\mu(H)$) is the Hausdorff dimension of $E_\mu(H)$.

We will need to deduce the Hölder exponent at every point from a “discretized version” of μ , i.e., from the values of μ on a countable collection of sets. A possible choice for this collection of sets is supplied by the dyadic cubes. The following lemma is a key ingredient in the derivation of the multifractal formalism for measures.

Lemma 2. *Let μ be a nonnegative measure defined on \mathbb{R}^d . Then*

$$h_\mu(x_0) = -\frac{1}{j} \liminf_{j \rightarrow +\infty} (\log_2 (\mu[3\lambda_j(x_0)])) . \quad (24)$$

Proof. By definition of the Hölder exponent,

$$\forall \varepsilon > 0, \exists r > 0, \forall r \leq R, \quad \mu(B(x, r)) \leq r^{H-\varepsilon};$$

but $3\lambda_j(x_0) \subset B(x_0, 3\sqrt{d}2^{-j})$, so that

$$\mu(3\lambda_j(x_0)) \leq (3\sqrt{d})^{H-\varepsilon} 2^{-j(H-\varepsilon)},$$

and it follows that $h_\mu(x_0)$ is bounded by the right hand-side of (24).

On the other hand, if $h_\mu(x_0) = H$, then there exist balls $B_n = B(x_0, r_n)$ and $\varepsilon_n > 0$ such that $r_n \rightarrow 0$, $\varepsilon_n \rightarrow 0$ and $r_n^{H+\varepsilon_n} \leq \mu(B_n) \leq r_n^{H-\varepsilon_n}$. Let j_n be such that

$$\frac{1}{2} 2^{-j_n} < r_n \leq 2^{-j_n};$$

then $B_n \subset 3\lambda_{j_n}(x_0)$ so that $\mu(B_n) \leq \mu(3\lambda_{j_n}(x_0))$, which implies the lower bound. \square

Let μ be a probability measure on \mathbb{R} . It follows from Lemma 2 that it is natural to define a scaling function associated with μ by

$$\tau_\mu(p) = \limsup_{n \rightarrow \infty} -\frac{1}{j} \log_2 \left(\sum_{\lambda \in \Lambda_j} (\mu(3\lambda))^p \right).$$

Let us now show why the spectrum of singularities is expected to be recovered from the scaling function. The definition of the scaling function roughly means that $\sum \mu(3\lambda) \sim 2^{-\eta_\mu(p)j}$. Let us estimate the contribution to this sum of the cubes λ that cover the points of $E_\mu(H)$. Lemma 2 asserts that they satisfy $\mu(3\lambda) \sim 2^{-Hj}$. By definition of the fractal dimension, we need about $2^{-d_\mu(H)j}$ such cubes to cover $E_\mu(H)$; thus, the corresponding contribution can be estimated by $2^{-dj} 2^{d_\mu(H)j} 2^{-Hpj} = 2^{-(d-d_\mu(H)+Hp)j}$. When $j \rightarrow +\infty$, the dominant contribution comes from the smallest exponent, so that $\tau_\mu(p) = \inf_H (d - d_\mu(H) + Hp)$. Inverting this Legendre duality relationship, and assuming that $d_\mu(H)$ is concave, we obtain

$$d_\mu(H) = \inf_H (d - \tau_\mu(p) + Hp). \quad (25)$$

Let us now show the relationship between the scaling function for measures, and the function spaces which contain the distribution function of this measure. Recall that the distribution function F_μ of μ is defined by $F_\mu(x) = \mu((-\infty, x])$.

Let $s \in (0, 1)$ and $p > 1$. A function F belongs to the Sobolev spaces $W^{s,p}(\mathbb{R})$ if it satisfies

$$\exists C > 0 \text{ such that} \quad \int \int \frac{|F(x) - F(y)|^p}{|x - y|^{sp+1}} dx dy \leq C. \quad (26)$$

We take $F = F_\mu$, and we split the integral (26) as a sum on the subdomains

$$A_j = \{(x, y) : 2^{-j} \leq |x - y| < 2 \cdot 2^{-j}\}.$$

The lefthand side of (26) is equivalent to

$$\sum_j 2^{j(sp+1)} \int \int_{A_j} |F_\mu(x) - F_\mu(y)|^p dx dy,$$

which, because F_μ is increasing, is equivalent to

$$\sum_j 2^{j(sp+1)} 2^{-2j} \sum_{\lambda \in \Lambda_j} |F_\mu(\lambda^+) - F_\mu(\lambda^-)|^p,$$

where $\lambda^+ = \frac{k+2}{2^j}$ and $\lambda^- = \frac{k-1}{2^j}$. But $F_\mu(\lambda^+) - F_\mu(\lambda^-) = \mu(3\lambda)$, and therefore the double integral in (26) is equivalent to

$$\sum_j 2^{j(sp-1)} \sum_{\lambda \in \Lambda_j} \mu(3\lambda)^p.$$

Coming back to the definition of $\tau_\mu(p)$ yields the function space interpretation

$$\tau_\mu(p) = \sup\{s : F_\mu \in W^{(s+1)/p,p}(\mathbb{R})\}, \quad (27)$$

which, using the embeddings between Sobolev and Besov spaces, implies that

$$\tau_\mu(p) = \eta_\mu(p) + 1.$$

3 Extensions of the Scaling Function: q -Leaders and Fractional Integration

Advances concerning the construction of new scaling functions beyond (22) were motivated by the following restriction: In order to be used, the wavelet leader method requires the data to be locally bounded. We saw a practical procedure in order to decide if this assumption is valid, namely the determination of the uniform Hölder exponent H_f^{min} . Experimental investigations showed that H_f^{min} is negative for large classes of natural “texture type” images, see [45–47], and therefore the method cannot be used as such.

In order to circumvent this problem, one can either change the data or change the method. By changing the data, we will mean “smoothing” them, in order to obtain a new signal, with a positive index H_f^{min} , on which the previous analysis may be safely performed. On the other hand, by changing the method, we will mean replacing the wavelet leaders by alternative quantities, which measure pointwise regularity (for another definition of regularity) and make sense even if the data are no more locally bounded. Let us start by exploring this second possibility.

3.1 q -Leaders

We will use the following extension of pointwise smoothness, which was introduced by Calderón and Zygmund in 1961, see [13].

Definition 9. Let $B(x_0, r)$ denote the open ball centered at x_0 and of radius r ; let $q \in [1, +\infty)$ and $\alpha > -d/q$. Let f be function which locally belongs to $L^q(\mathbb{R}^d)$. Then f belongs to $T_\alpha^q(x_0)$ if there exist $C, R > 0$ and a polynomial P such that

$$\forall r \leq R, \quad \left(\frac{1}{r^d} \int_{B(x_0, r)} |f(x) - P(x - x_0)|^q dx \right)^{1/q} \leq Cr^\alpha. \quad (28)$$

The q -exponent of f at x_0 is

$$h_f^q(x_0) = \sup\{\alpha : f \in T_\alpha^q(x_0)\}.$$

Note that the Hölder exponent corresponds to the case $q = +\infty$. This definition is a natural substitute for pointwise Hölder regularity when dealing with functions which are not locally bounded, but locally belong to L^q . In particular, the q -exponent can take values down to $-d/q$, and therefore it allows to model behaviors which locally are of the form $1/|x - x_0|^\alpha$ for $\alpha < d/q$, i.e., to deal with negative regularity exponents.

Let us now show how the notion of T_α^q regularity can be related to local l^q norms of wavelet coefficients. This will be done with the help of the following theorem, which yields a characterization of this pointwise smoothness. First, recall that the *local square functions at x_0* are

$$S_f(j, x_0)(x) = \left(\sum_{\lambda \in 3^j \lambda_j(x_0)} |c_\lambda|^2 1_\lambda(x) \right)^{1/2}.$$

The following theorem is proved in [27].

Theorem 3. Let $q \in (0, \infty)$, $\alpha > -d/q$ and assume that the wavelet basis used is r -smooth with $r > \sup(2\alpha, 2\alpha + 2d(\frac{1}{q} - 1))$; if $f \in T_\alpha^q(x_0)$, then $\exists C \geq 0$ such that $\forall j \geq 0$,

$$\|S_f(j, x_0)\|_q \leq C 2^{-j(\alpha + d/q)} \quad (29)$$

(if $q < 1$, then $\|\cdot\|_q$ denotes the L^q quasi-norm).

Conversely, if (29) holds and if $\alpha \notin \mathbb{N}$, then $f \in T_\alpha^q(x_0)$.

Remark 5. When α is an integer, then the wavelet condition (29) characterizes a slightly different space, which implies a $T_\alpha^q(x_0)$ with a logarithmic loss on the modulus of continuity. This is reminiscent of the case of uniform Hölder spaces for $s = 1$, in which case the wavelet condition (15) characterizes the Zygmund class instead of the usual C^1 space.

We now relate local square functions and local l^q norms of wavelet coefficients. (The derivation that we propose slightly differs from the one of [31], since it is in the spirit of “wavelet leaders”, whereas the one performed in [31] relies on extensions of two-microlocal spaces which were proposed by Y. Meyer and H. Xu in [40].)

Note that the condition $\|S_f(j, x_0)\|_q \leq C2^{-j(\alpha+d/q)}$ implies that, at each scale $j' \geq j$,

$$\left\| \left(\sum_{\lambda' \in 3\lambda_j(x_0), |\lambda'|=j'} |c_{\lambda'}|^2 1_{\lambda'}(x) \right)^{1/2} \right\|_q \leq C2^{-j(\alpha+d/q)},$$

where, by definition, $|\lambda'|$ denotes the scale j' of the cube λ' . Since dyadic cubes at a given scale do not overlap, the left-hand side can be computed exactly and yields

$$\left(\sum_{\lambda' \in 3\lambda_j(x_0), |\lambda'|=j'} |c_{\lambda'}|^q 2^{-dj'} \right)^{1/q}.$$

It follows that the pointwise q -exponent can be expressed by a regression on a log-log plot of the q -leaders

$$d_\lambda^q = \left(\sum_{\lambda' \in 3\lambda_j(x_0)} |c_{\lambda'}|^q 2^{-d(j'-j)} \right)^{1/q}. \quad (30)$$

Note that q -leaders are nothing but local Besov norms. Therefore, besides the fact that they are much easier to estimate than L^q norms of the local square function (as proposed in [28, 30]), they also offer the additional possibility of taking $q < 1$. In that case, the relationship with pointwise regularity is more intricate, see [28]. However, for classification purposes, one does not necessarily require an interpretation in terms of pointwise exponents.

As in the case of the usual wavelet leaders, one can associate function spaces with q -leaders: Let $p, q \in (0, \infty)$; a function f belongs to the *Oscillation space* $\mathcal{O}_{p,q}^s(\mathbb{R}^d)$ if and only if

$$\exists C, \forall j \geq 0, \quad \sum_{\lambda \in \Lambda_j} \left[2^{(s-d/p)j} d_\lambda^q \right]^p \leq C.$$

These function spaces are closely related with the spaces $S_{q,r,m}^\alpha$ and $\mathcal{D}_{q,r,m}^\alpha$ introduced by A. Seeger in [44]. Indeed these spaces are also defined through local L^q norms of the function. However, the motivation of Seeger was different from ours: He wished to obtain a new characterization of Triebel–Lizorkin spaces, and therefore only considered the cases where the parameters α, q, r, m lead to these spaces, whereas our motivation is at the opposite, since we are interested in these spaces when they strongly differ from the classical ones, and lead to new scaling functions.

The q -scaling function is defined by

$$\forall p \in \mathbb{R}, \quad \zeta_f(p, q) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |d_\lambda^q|^p \right). \quad (31)$$

In that case too, the scaling function is “well defined” for $p < 0$ (it satisfies the same robustness properties as the leader scaling function). When p is positive, the q -scaling function has the following function-space interpretation:

$$\zeta_f(p, q) = \sup\{s : f \in \mathcal{O}_{p,q}^{s/p}\}.$$

Note that, for a given q , this analysis is possible only if the data locally belong to L^q . This can be checked on the wavelet scaling function, since this condition will be satisfied if $\eta_f(q) > 0$. On the opposite, if $\eta_f(q) < 0$, this analysis is not relevant. We see here another use of the wavelet scaling function, as a preliminary quantity which is required to be computed before estimating the q -scaling function. Therefore it plays a similar role as the computation of H_f^{min} when dealing with the multifractal analysis based on wavelet leaders. Note that, as above, a spectrum can be attached to the q -exponent, and a multifractal formalism worked out, using the usual procedure; this spectrum is obtained as a Legendre transform of the q -scaling function, see [27, 28, 31]. An important open question is to understand, for a given function f , the relationships that exist between these q -spectra, as a function of the parameter q .

3.2 Fractional Integration

Let us come back to the problem raised at the beginning of Sect. 3, namely the introduction of tools adapted to situations where wavelet leaders cannot be used because the data analyzed are not locally bounded. We mentioned that an alternative solution is to “smooth” the data, so that the exponent H_f^{min} becomes positive. This can be done through the use of a *fractional integration*. We now expose this procedure, both from a theoretical and practical point of view.

Definition 10. Let f be a tempered distribution; the fractional integral of order s of f is the operator $(Id - \Delta)^{-s/2}$ defined as the convolution operator which amounts to multiply the Fourier transform of f with $(1 + |\xi|^2)^{-s/2}$.

Let ϕ be a C^∞ compactly supported function satisfying $\phi(x) = 1$ for x in a neighbourhood of x_0 . If $f \in L_{SG}^1$, its local fractional integral of order s is

$$f^{(-s)} = (Id - \Delta)^{-s/2}(\phi f). \quad (32)$$

Note that the function spaces which can be locally defined through the notion of local fractional integral do not depend on the function ϕ which is chosen. Indeed, denote by $f_1^{(-s)}$ and $f_2^{(-s)}$ the local fractional integrals of f corresponding to two different functions ϕ_1 and ϕ_2 ; $f\phi_1 - f\phi_2$ vanishes in a neighbourhood of x_0 , and

therefore is C^∞ in a neighbourhood of x_0 . The local regularity properties of the operator $(Id - \Delta)^{-s/2}$ imply that $(Id - \Delta)^{-s/2}(f\phi_1 - f\phi_2)$ also is C^∞ in a neighbourhood of x_0 ; therefore $f_1^{(-s)}$ and $f_2^{(-s)}$ differ by a C^∞ function in a neighbourhood of x_0 .

Numerically, the fractional integration can be replaced by a much easier procedure which shares the same properties in terms of regularity, a *pseudo-fractional integration* defined as follows, see [36].

Definition 11. Let $s > 0$, let ψ_λ be an r -smooth wavelet basis with $r > s + 1$ and let f be a function, or a distribution, with wavelet coefficients c_λ . The pseudo-fractional integral (in the basis ψ_λ) of f of order s , denoted by $I^s(f)$, is the function whose wavelet coefficients (on the same wavelet basis) are

$$c_\lambda^s = 2^{-sj} c_\lambda.$$

The following result shows that this numerically straightforward operation retains the same properties as the fractional integration.

Proposition 1. Let f be a function satisfying $H_f^{min} > 0$. Then, for any $s \in \mathbb{R}$, the wavelet scaling functions of $I^s(f)$ and $f^{(-s)}$ coincide. It is also the case for Hölder exponents or q -exponents:

$$\forall s > 0, \forall x_0, \forall q \in (0, +\infty], \quad h_{I^s(f)}^q(x_0) = h_{f^{(-s)}}^q(x_0).$$

The proof of this result requires to introduce several tools; the first one is the algebras \mathcal{M}^γ , which are defined as follows.

Definition 12. An infinite matrix $A(\lambda, \lambda')$ indexed by the dyadic cubes of \mathbb{R}^d belongs to \mathcal{M}^γ if

$$|A(\lambda, \lambda')| \leq \frac{C 2^{-(\frac{d}{2} + \gamma)(j - j')}}{(1 + (j - j')^2)(1 + 2^{\inf(j, j')} \text{dist}(\lambda, \lambda'))^{d + \gamma}}.$$

Matrices of operators which map an r -smooth wavelet basis onto another one belong to these algebras, as soon as $\gamma > r$, and more generally matrices (on wavelet bases) of pseudodifferential operators of order 0, such as the Hilbert transform in dimension 1, or the Riesz transforms in higher dimensions, belong to these algebras, see [36]. We denote by $\mathcal{Op}(\mathcal{M}^\gamma)$ the space of operators whose matrix on a r -smooth wavelet basis (for $r > \gamma$) belongs to \mathcal{M}^γ . Note that this space does not depend on the (smooth enough) wavelet basis which is chosen; indeed, the matrix of coordinate change from one r -smooth wavelet basis to another belongs to \mathcal{M}^γ if $r > \gamma$, so that the independence is a consequence of the fact that the spaces \mathcal{M}^γ are algebras.

The second tool that we will need is the notion of *vaguelette* system.

Definition 13. A set of functions (θ_λ) indexed by the dyadic cubes of scale $j \geq 0$, forms a vaguelette system of order s if

- For any $j \geq 1$ the vaguelettes θ_λ of scale j have vanishing moment up to order $s+1$, i.e., if, for any multi-index α satisfying $|\alpha| \leq s+1$, then

$$\int \theta_\lambda(x) x^\alpha dx = 0,$$

- The θ_λ satisfy the following uniform decay estimates: For any multi-index α satisfying $|\alpha| \leq s+1$, then

$$\forall N \in \mathbb{N}, \quad \left| \frac{\partial^\alpha \theta_\lambda}{\partial x^\alpha} \right| \leq \frac{C_N 2^{(\alpha+d/2)j}}{(1+|2^j x - k|)^N}.$$

Biorthogonal vaguelette bases are couples of Riesz bases θ_λ^1 and θ_λ^2 which are both vaguelette systems and form biorthogonal bases. Therefore, $\forall f \in L^2$,

$$f(x) = \sum_\lambda c_\lambda^1 \theta_\lambda^2, \quad \text{where} \quad c_\lambda^1 = \int_{\mathbb{R}^d} f(x) \theta_\lambda^1 dx.$$

The notions we introduced are related by the following key property, see [36]:

Proposition 2. *Let \mathcal{M} be an operator which maps an r -smooth wavelet basis to a vaguelette system of order r ; then, for any $\gamma < r$, \mathcal{M} belongs to $\mathcal{O}p(\mathcal{M}^\gamma)$.*

We now turn to the proof of Proposition 1

Proof. It is performed using the wavelet techniques developed in [36], such as the function spaces characterizations; therefore, we won't give a complete detailed proof, but only mention the main lines.

The first point consists in noticing that the systems

$$\psi_\lambda^1 = 2^{sj} (Id - \Delta)^{-s/2} \psi_\lambda \quad \text{and} \quad \psi_\lambda^2 = 2^{-sj} (Id - \Delta)^{s/2} \psi_\lambda$$

are biorthogonal vaguelette systems. This property is straightforward to check on the Fourier transform ψ_λ^1 and ψ_λ^2 , which, in this case, are completely explicit.

Note that

$$I^s(f) = \sum 2^{-sj} c_\lambda \psi_\lambda \quad \text{and} \quad f^{(-s)} = \sum 2^{-sj} c_\lambda \psi_\lambda^1.$$

If f is locally bounded, then $I^s(f)$ belongs locally to C^s (because of the wavelet characterization of C^s);

Assume now that $I^s(f)$ belongs to $C^\alpha(x_0)$; the operator that maps ψ_λ to ψ_λ^1 belongs to $\mathcal{O}p(\mathcal{M}^\gamma)$, and therefore preserves the pointwise wavelet regularity criterion, see [21], and it also preserves the $T_\alpha^q(x_0)$ regularity, see [13]. Therefore, it is satisfied by $f^{(-s)}$ and, since the uniform Hölder exponent of $f^{(-s)}$ is positive, the converse part of the wavelet pointwise regularity criterion implies that it belongs to $C^\beta(x_0)$ for any $\beta < \alpha$, see [21, 25]. The proof of the converse part is similar, using the biorthogonality of the ψ_λ^1 and ψ_λ^2 . \square

As regards signals and images for which $H_f^{min} < 0$, a possibility for applying to them a multifractal analysis based on the Hölder exponent is to perform on the data a fractional integration of sufficiently large order t ; indeed, the uniform regularity exponent H_f^{min} is always shifted exactly by t , see [29]. This simple property shows a possible strategy in order to perform the multifractal analysis of a signal which is not locally bounded: First determine its exponent H_f^{min} ; then, if $H_f^{min} < 0$, perform a fractional integration of order $t > -H_f^{min}$; it follows that the uniform regularity exponent of $I^t(f)$ is positive, and therefore it is a bounded function. The t -leaders are

$$d_\lambda^t = \sup_{\lambda' \subset 3\lambda} |c_{\lambda'} 2^{-tj'}| \quad (33)$$

The strategy we sketched leads to the consideration of the following *fractional leader scaling function* :

$$\forall p \in \mathbb{R}, \forall t > 0, \quad \zeta_f(t, p) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |d_\lambda^t|^p \right). \quad (34)$$

It can also be given a function-space interpretation: Let $p \in (0, \infty)$; a function f belongs to the *Oscillation space* $\mathcal{O}_p^{s,t}(\mathbb{R}^d)$ if and only if $(c_k^{(0)}) \in l^p$ and

$$\exists C, \forall j, \quad \sum_{\lambda \in \Lambda_j} \left[2^{(s-d/p)j} d_\lambda^t \right]^p \leq C,$$

see [24, 26] for properties of these spaces. Then

$$\zeta_f(p, t) = \sup \{s : f \in \mathcal{O}_p^{s/p, t}\}.$$

The Legendre transform of this scaling function is expected to yield, for each t , the spectrum of singularities of $I^t(f)$, see [24].

The strategy developed in this paragraph raises an open question: There is no canonical choice for the order of fractional integration, (the only condition is that it has to be larger than $-H_f^{min}$) and the quantities considered may depend on this order. This is one of the motivations for understanding how multifractal properties are modified under fractional integration. We will give some clues on these questions in Sect. 4 (see also [5, 29]).

3.3 The q -Fractional Scaling Function

The two strategies developed in the last two subsections correspond to different answers of the following problem: How to perform a multifractal analysis of non locally bounded data. They led to the consideration of either q -exponents, or of fractional integrals. A third option consists in not making a choice but taking advantage

of both strategies, and considering q -exponents of fractional integrals. It follows from Proposition 1 and the definition of q -leaders (30) that the q -exponent of a fractional integral of order t is given by a regression on a log-log plot of the following *grand leaders*

$$d_{\lambda}^{t,q} = \left(\sum_{\lambda' \subset 3\lambda_j(x_0)} \left| 2^{-tj'} c_{\lambda'} 2^{-d(j'-j)} \right|^q 2^{-dj'} \right)^{1/q}. \quad (35)$$

Not surprisingly, one can associate function spaces to grand leaders: Let $p, q \in (0, \infty)$; a function f belongs to the *Oscillation space* $\mathcal{O}_{p,q}^{s,t}(\mathbb{R}^d)$ if and only if

$$\exists C, \forall j \geq 0, \quad \sum_{\lambda \in \Lambda_j} \left[2^{(s-d/p)j} d_{\lambda}^{t,q} \right]^p \leq C.$$

The grand scaling function is defined by

$$\forall p \in \mathbb{R}, \quad \zeta_f(t, p, q) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(2^{-dj} \sum_{\lambda \in \Lambda_j} |d_{\lambda}^{t,q}|^p \right).$$

Note that the computation of the grand leaders require that f^{-t} locally belongs to L^q . Once again, this can be checked on the wavelet scaling function of f : This requirement will be fulfilled as soon as

$$\eta_f(q) > -qt. \quad (36)$$

Note that, in that case too, $\zeta_f(t, p, q)$ is “well defined” for $p < 0$ (it satisfies the same robustness properties as $\zeta_f(p)$). When p is positive, the grand scaling function has the following function space interpretation:

$$\zeta_f(t, p, q) = \sup \{s : f \in \mathcal{O}_{p,q}^{s/p,t}\}.$$

The derivation proposed by Parisi and Frisch can be adapted to the previous settings; indeed, as soon as a scaling function is supplied, one can derive a multifractal formalism, with corresponding upper bounds, following a general abstract procedure which was developed in [30]. We mention what it yields only in the most general setting supplied by the grand scaling function (actually, the other settings developed in Sects. 3.1 and 3.2 can be considered as subcases). Let

$$E_f(H, q, t) = \{x_0 : h_{f^{-t}}^q(x_0) = H\}.$$

The corresponding *grand multifractal formalism* asserts that the *grand spectrum* $d_f(H, q, t) := \dim(E_f(H, q, t))$ can be deduced from the grand scaling function by a Legendre transform

$$\forall H, q, t, \quad d_f(H, q, t) = \inf_{p \in \mathbb{R}} (d + Hp - \zeta_f(t, p, q)).$$

Using either the scaling function $\zeta_f(t, p, q)$ or its Legendre transform $d_f(H, q, t)$ gives a very complete information on the nature of the pointwise singularities of f , and a rich tool for classification. However, by construction, it cannot yield information on the dimensions of the sets where two different exponents take given values. Such information requires the construction of *grand canonical formalisms*. Let us now explain in a general setting how they can be derived.

4 An Introduction to the Grand Canonical Formalism

A common feature shared by (21), (24), and (29) is that all the pointwise exponents we considered can be deduced from a countable number of quantities e_λ indexed by the dyadic cubes, and the derivation is performed on a log-log plot bearing on the cubes that contain the point x_0 . In the measure case, $e_\lambda = \mu(3\lambda)$, in the pointwise Hölder case, $e_\lambda = d_\lambda$, and in the $T_\alpha^p(x_0)$ case, $e_\lambda = d_\lambda^p$, so that the general setting in which we work in this section will cover all the previous cases we already considered. These examples all fit in the following general framework for collections of positive quantities e_λ indexed by the dyadic cubes:

- The e_λ are increasing i.e., if $\lambda' \subset \lambda$ then $e_{\lambda'} \leq C \cdot e_\lambda$.
- A pointwise exponent is associated with the (e_λ) :

$$h(x_0) = \liminf_{j \rightarrow +\infty} -\frac{1}{j} \log_2 \left(e_{\lambda_j(x_0)} \right) \quad (37)$$

Grand canonical formalisms are fitted to situations where a couple of such exponents is considered: For example, they may correspond to the wavelet leaders of two functions f and g , in which case we will be interested in elaborating a multifractal formalism which yields the dimensions of the sets of points where each exponent takes a given value, or they may correspond to two different exponents associated with the same function. One important example is the couple formed by the Hölder and the oscillation exponents. Before deriving the grand canonical formalism, we recall the motivation for studying oscillation exponents.

4.1 Oscillation Exponents: Discussion of Possible Definitions

All notions of pointwise singularity which have been considered are variants on the notion of “regularity exponent”, which, roughly speaking, associates the exponent γ to the singularity $|x - x_0|^\gamma$ at x_0 , at least if $\gamma > 0$ (and $\gamma \notin 2\mathbb{N}$), see [28, 37] for explicit general definitions of the notion of regularity exponent); such exponents include the (most widely used) Hölder exponent (see Definition 6), the q -exponent (see Definition 9), and the weak-scaling exponent, see [37]. However, one can wish

to have information on how the function considered oscillates near the singularity at x_0 : Consider for instance the “chirps”

$$F_{\gamma,\beta}(x) = |x - x_0|^\gamma \sin\left(\frac{1}{|x - x_0|^\beta}\right), \quad (38)$$

for a given regularity γ , their oscillatory behavior in the neighbourhood of x_0 increases with β ; in this example, β parametrizes a “degree of oscillation”. (Note that the case $\beta = 0$, i.e., $F_{\gamma,0}(x) = |x - x_0|^\gamma$ is usually referred to as a “cusp”). We will use a refined description of singularities by introducing an additional “oscillation” parameter, that allows to draw distinctions between functions which share the same Hölder exponent. Measuring such an additional exponent rises additional difficulties, one of them being that several mathematical definitions have been proposed, yielding different types of information.

One motivation for the detection of singularities such as (38) is that the existence of such behaviors has been conjectured in some physical data, such as fully developed turbulence. Another motivation is an internal mathematical requirement in multifractal analysis: We saw that one is often obliged to compute a fractional integral of the signal before performing its multifractal analysis. In that case, the singularity sets can be modified in a way which is difficult to predict if singularities such as (38) are present in the signal. Therefore understanding what multifractal analysis yields in this case requires the consideration of such behaviors.

We wish to describe strong local oscillations which display the same qualitative feature as in (38). A clue is supplied by the following remark. Let $\gamma > 0$, be given and let us estimate the integral of (38). Since

$$\int_0^x |t - x_0|^\gamma \sin\left(\frac{1}{|t - x_0|^\beta}\right) dt = \int_0^x \frac{|t - x_0|^{\gamma+\beta+1}}{\beta} \left(\cos\left(\frac{1}{|t - x_0|^\beta}\right)\right)' dt,$$

it follows that, in the neighbourhood of x_0 , this integral is the sum of

$$\frac{|x - x_0|^{\gamma+\beta+1}}{\beta} \cos\left(\frac{1}{|x - x_0|^\beta}\right)$$

and higher order terms; thus the Hölder exponent of $F_{\gamma,\beta}$ is γ , but the Hölder exponent of its integral is $\gamma + \beta + 1$; thus one can reasonably expect that the oscillation exponent can be recovered by comparing the regularity exponents of f and of its integral.

Before proposing a precise mathematical procedure which allows to recover β , let us mention a natural requirements that a notion of “oscillation exponent” should satisfy in order to be of practical use. The definition proposed should allow for possible superpositions and “mixtures”; indeed, in the spirit of multifractal analysis, we do not expect these local behaviors to appear only in an isolated, “perfect” form as in (38), but rather for a dense set of values of x_0 , and with possible corruptions by noise. Therefore one should find a key feature of (38) that characterizes the exponent β , and use it as a general definition of oscillating singularity. We noticed that β

should be recovered by comparing the regularity exponents of f and of its integral. We will need a slight extension of this remark in order to obtain a definition of oscillation exponent which fulfills this requirement. For that purpose, we will use the notion of *local fractional integral* already considered. The Hölder exponent of the local fractional integral of f of order s at x_0 is called the ***fractional Hölder exponent of f at x_0*** and denoted by

$$h_f^s(x_0) = h_{f^{(-s)}}(x_0).$$

Lemma 3. *The definition of $h_f^s(x_0)$ does not depend on the function ϕ which is chosen.*

The proof is similar to the proof of Proposition 1. Properties of the fractional Hölder exponent have been investigated in [7, 34]. In particular one can show that it is a concave function.

It follows from the concavity of the fractional Hölder exponent that

$$\text{either} \quad \forall s > 0, \quad h_f^s(x_0) = +\infty \quad \text{or} \quad \forall s > 0, \quad h_f^s(x_0) < +\infty;$$

an example of the first case is given by

$$|x - x_0|^\alpha \sin \left(\exp \left(\frac{1}{|x - x_0|} \right) \right),$$

where the Hölder exponent at x_0 is α , but, as soon as $s > 0$, then $h_f^s(x_0) = +\infty$. When this first occurrence happens, the oscillation exponent should clearly be set to $+\infty$. Therefore, in the sequel, we suppose that the second case occurs. The definition of the oscillation exponent that we will choose is motivated by the simple but important remark that follows.

Lemma 4. *Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a locally bounded function, and let s be a positive integer. Denote by I_f^s an iterated integral of order s of f ; then*

$$h_f^s(x_0) = h_{I_f^s}(x_0).$$

Proof. Since the result is clearly local, we can assume that f is supported in a neighbourhood of x_0 , and therefore belongs to L^2 , which allows to use the Fourier transform without any restriction, and also allows to assume that $\phi = 1$ in the definition of the local fractional integral. Up to a polynomial term, one derives I_f^s from f by multiplying \hat{f} by $(i\xi)^{-s}$; this iterated integral has the same Hölder exponent as the one obtained using $|\xi|^{-s}$ instead. Indeed, the corresponding operators are either the same (up to a multiplicative constant), or deduced from each other by an Hilbert transform (the Fourier multiplier by $\text{sign}(\xi)$). In addition, applying the Hilbert transform does not modify the pointwise Hölder exponent since $f^{(-s)}$ has an exponent H^{\min} which is positive, see [21]. The result follows by noticing that

$$\frac{1}{(1 + |\xi|^2)^{s/2}} - \frac{1}{|\xi|^s} \sim \frac{C}{|\xi|^{s+2}} \quad \text{when} \quad |\xi| \rightarrow +\infty,$$

and therefore the corresponding operator is a uniformly smoothing operator which, for any $\alpha \in \mathbb{R}$, maps $C^\alpha(\mathbb{R})$ to $C^{\alpha+s+2}(\mathbb{R})$. Therefore,

$$h_{f^{(-s)} - f^s}(x_0) \geq 2 + \sup(h_{f^{(-s)}}(x_0), h_{f^s}(x_0)),$$

and we get $h_{f^{(-s)}}(x_0) = h_{f^s}(x_0)$. \square

It follows from this lemma that results in dimension 1 which are checked by hand through the computation of integrals can then be extended to any dimension and to the non-integer case by using fractional integrals; for instance, let us check that

$$\forall s \geq 0, \quad h_{F_{\gamma,\beta}}^s(x_0) = \gamma + (1 + \beta)s, \quad (39)$$

where $F_{\gamma,\beta}$ was defined by (38). Indeed, a straightforward integration by parts shows that this result holds for iterated integrals, and thus, using Lemma 4, it also holds for fractional integrals of integer order; the result for all s follows from the concavity of the fractional Hölder exponent. Therefore, a natural definition for the oscillation exponent of an arbitrary function at x_0 should use the slope of the function

$$s \rightarrow h_f^s(x_0). \quad (40)$$

In the case of $F_{\gamma,\beta}$, this function is linear, and the definition is unambiguous; however, it is not always the case, and one can show (see [7, 34]) that, in general, functions defined by (40) only satisfy the following properties:

Proposition 3. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a locally bounded function, and let $x_0 \in \mathbb{R}^d$. The function $s \rightarrow h_f^s(x_0)$ is concave; therefore it has everywhere a left and a right derivative. These derivatives satisfy*

$$\forall s \geq 0, \quad \frac{\partial(h_f^s(x_0))}{\partial s} \geq 1.$$

Furthermore, these properties characterize fractional Hölder exponents.

It follows from this characterization that $h_f^s(x_0)$ is, in general, not a linear function of s , and therefore many choices are possible for its slope. In practice, only two choices have been used up to now, leading to two different exponents:

- The chirp exponent (choice of the slope “at infinity”)

$$\gamma_f(x_0) = \lim_{s \rightarrow +\infty} \frac{\partial}{\partial s}(h_f^s(x_0)) - 1, \quad (41)$$

- The oscillation exponent (choice of the slope “at the origin”)

$$\beta_f(x_0) = \lim_{s \rightarrow 0} \frac{\partial}{\partial s}(h_f^s(x_0)) - 1, \quad (42)$$

(see [32] for properties associated with chirp exponents, and [7] for properties associated with oscillation exponents). The drawback of using (41) is that this notion is very unstable: If g is an arbitrary smooth (but not C^∞) function, one can have

$$\gamma_f(x_0) = \Gamma > 0 \quad \text{but} \quad \gamma_{f+g}(x_0) = 0.$$

A simple example of this phenomenon is given by the functions: $f(x) = x \sin(1/x)$ and $g = |x|^a$ for an $a \notin 2\mathbb{N}$ and large enough. Thus, the chirp exponent does not satisfy the natural requirement of stability under the addition of smooth noise.

4.2 Properties and Wavelet Estimation of the Oscillation Exponent

The instabilities mentioned above cannot occur when using the oscillation exponent as shown by the following result, which states that, as soon as one imposes this stability requirement, then the choice of $\beta_f(x_0)$ is canonical.

Proposition 4. *Let f be a locally bounded function satisfying: $\forall s > 0, h_f^s(x_0) < +\infty$. Then $\beta_f(x_0)$ is the only quantity which satisfies the following properties:*

- *It is deduced from the function $s \rightarrow h_f^s(x_0)$.*
- *If $h_g(x_0) > h_f(x_0)$, then $\beta_{f+g}(x_0) = \beta_f(x_0)$ (the oscillation exponent of f is not altered under the addition of a function which is smoother than f).*
- *It yields the exponent β for the functions $F_{\gamma,\beta}$.*

Proof. Let us first check that the oscillation exponent satisfies these properties. We only have to check the second one; indeed, the first one follows from the definition, and the last one has already been proved. Because of the finiteness assumption for h_f^s , and its concavity, it is a continuous function; therefore, it remains strictly smaller than $h_g(x_0)$ in a small neighbourhood of x_0 , and thus

$$\exists \eta > 0 : \quad \forall s < \eta, \quad h_f^s(x_0) < h_g^s(x_0),$$

so that $h_{f+g}^s(x_0) = h_f^s(x_0)$.

Let us now prove the converse result. The stability requirement implies that the quantity considered cannot be a function of $h_f^s(x_0)$ for an $s > 0$; therefore it is a “germ property” at $s = 0$, and therefore, a function of the value at 0 of the function $s \rightarrow h_f^s(x_0)$ and its derivatives. Since $h_f^s(x_0)$ can be an arbitrary concave function, higher order derivatives do not exist in general; therefore, only $h_f(x_0)$ and the first derivative can be involved. Finally, the fact that the exponent takes the value β for the functions $F_{\gamma,\beta}$ implies that it is given precisely by (42). \square

Remark 6. In practice, one cannot directly measure the oscillation exponent since it involves the estimation of how the Hölder exponent evolves under a fractional integration of “infinitesimal” order, and one rather estimates the evolution under

a fractional integration of given fixed order s , thus obtaining the s -oscillation exponent:

$$\beta_f(s, x_0) = \frac{h_f^s(x_0) - h_f(x_0)}{s} - 1. \quad (43)$$

Because of the concavity of $h_f^s(x_0)$,

$$\forall s > 0, \quad \gamma_f(x_0) \leq \beta_f(s, x_0) \leq \beta_f(x_0)$$

and

$$\beta_f(s, x_0) = 0 \iff \beta_f(x_0) = 0. \quad (44)$$

Note that the general definition of a cusp singularity at x_0 is given by the condition: $\beta_f(x_0) = 0$.

Note that (44) is of practical importance for the following reason: If one is interested in a qualitative information such as the existence or absence of oscillating singularities (and not the precise values taken by β_f), then, (44) shows that, in practice, one may as well work with the s -oscillation exponent rather than the oscillation exponent in order to obtain the required information.

We now recall the characterization of oscillating singularities, that was discovered by J.-M. Aubry, cf [9]. It makes use of the following notion introduced in [32], which is a natural generalization (in particular to higher dimensions) of the oscillating behavior of the sine function.

Definition 14. A function $g \in L^\infty(\mathbb{R}^d)$ is called indefinitely oscillating if and only if $\exists \omega \in \mathcal{S}(\mathbb{R}^d)$ such that such $\forall N \in \mathbb{N}$, g can be written

$$g = \omega + \sum_{|\alpha|=N+1} \partial^\alpha g_\alpha$$

with each $g_\alpha \in L^\infty(\mathbb{R}^d)$.

J.-M. Aubry proved that, if $h < h_f(x_0)$ and if f has oscillating exponent β , then f can be written, in the neighbourhood of x_0 as

$$f(x) = |x - x_0|^h g \left(\frac{x - x_0}{|x - x_0|^{\beta+1}} \right) + r(x),$$

where g is indefinitely oscillating, and r is smoother than the first term at x_0 (see [9]) for a more precise statement).

4.3 Derivation of β -Leaders

Let us now show how, heuristically, one can derive quantities similar to leaders, that would yield the oscillation exponent on a log-log plot. Let us work out the general heuristic that backs the derivation of a multifractal formalism in the specific setting

of the oscillation exponent (for the derivation in the initial setting of the Hölder exponent, see the seminal paper [42], and for a derivation in a dyadic setting adapted to the wavelet framework, see [25, 30]). We wish to express the oscillation exponent by a condition bearing on wavelet leaders. Recall that the leaders associated with the fractional integral of order t are the t -leaders (33).

It follows from Proposition 1 that

$$h_f^s(x_0) = \liminf_{j \rightarrow +\infty} \left(\frac{\log(d_j^s(x_0))}{\log(2^{-j})} \right), \quad (45)$$

therefore, the oscillation exponent can be derived from the d_λ^s according to (42).

We will need the following notations. The first one is a weak form of the \mathcal{O} notation of Landau: If F and G are two functions which tend to 0, $F = \overline{\mathcal{O}}(G)$ if

$$\liminf \frac{\log|F|}{\log|G|} \geq 1,$$

and the second one expresses the fact that two functions are of the same order of magnitude, disregarding “logarithmic corrections”:

$$F \sim G \quad \text{if} \quad \lim \frac{\log|F|}{\log|G|} = 1.$$

The following proposition is a consequence of the previous remarks.

Proposition 5. *Let f be such that $H_f^{\min} > 0$. The oscillating singularity exponents of f at x_0 are (h, β) if and only if its wavelet leaders satisfy the following conditions:*

1. $d_j(x_0) = \overline{\mathcal{O}}(2^{-hj})$.
2. *There exists a sequence $j_n \rightarrow \infty$ such that*

$$d_{j_n}(x_0) \sim 2^{-hj_n}, \quad (46)$$

3. *There exists a sequence $\lambda'_n \subset 3\lambda_{j_n}(x_0)$ such that:*

$$\begin{cases} j'_n = (1 + \beta)j_n + o(j_n) \\ |c_{\lambda'_n}| \sim d_{j_n}(x_0), \end{cases} \quad (47)$$

4. β is the smallest number such that (47) holds.

Remark 7. The last condition means that the supremum in the definition of the wavelet leader $d_{j_n}(x_0)$ is “almost” attained at a scale close to $(1 + \beta)j$.

Let $s > 0$ small enough be given. If f has an oscillating singularity with exponents (H, β) at x_0 then its wavelet leaders satisfy $d_j(x_0) \sim 2^{-Hj}$ and its integrated wavelet leaders satisfy $d_j^s(x_0) \sim 2^{-(H+s(1+\beta))j}$. By elimination, this allows to construct quantities that will scale like $2^{-\beta j}$ in the limit of small scales: the β -leaders

$$B_\lambda^s = 2^j \left(\frac{d_j^s(x_0)}{d_j(x_0)} \right)^{1/s},$$

and we expect that, if β is the oscillation exponent at x_0 , then, for s small enough,

$$B_\lambda^s \sim 2^{-\beta j}. \quad (48)$$

4.4 Derivation of Grand Canonical Formalisms

We now come back to the general setting supplied by two pointwise exponents given by the rate of decay, on a log–log plot, of two quantities e_λ and f_λ indexed by dyadic cubes. In order to derive a multifractal formalism for this couple of exponents, following the path suggested by the grand canonical formalism in statistical physics, we consider the following structure function

$$S_j(p, q) = 2^{-dj} \sum_{\lambda \in \Lambda_j} (e_\lambda)^p (f_\lambda)^q.$$

The associated scaling function is

$$\eta(p, q) = \liminf_{j \rightarrow +\infty} \left(\frac{\log(S_j(p, q))}{\log(2^{-j})} \right) \quad (49)$$

We will reformulate the fundamental idea due to G. Parisi and U. Frisch in this general setting. These authors gave an interpretation of the nonlinearity of the scaling function as the signature of the presence of different pointwise exponents (see [42] and also [20] for applications, particularly in the setting of invariant measures of dynamical systems).

The following notion of spectrum is adapted to the simultaneous consideration of two exponents.

Definition 15. Let $(e_\lambda)_{\lambda \in \Lambda}$ and $(f_\lambda)_{\lambda \in \Lambda}$ be increasing dyadic functions, and let

$$E_{H_1, H_2} = \{x : h_e(x) = H_1 \text{ and } h_f(x) = H_2\}.$$

The spectrum of singularities associated with the $(e_\lambda)_{\lambda \in \Lambda}$ is the function $d(H_1, H_2)$ defined by

$$d(H_1, H_2) = \dim(E_{H_1, H_2}).$$

The support of the spectrum is the set of values (H_1, H_2) for which $E_{H_1, H_2} \neq \emptyset$.

Let us now show heuristically how the spectrum of singularities can be recovered from the scaling function. The definition of the scaling function (49) means that,

$$\text{when } j \rightarrow +\infty, \quad S_j(p, q) \sim 2^{-\eta(p, q)j}.$$

Let us estimate the contribution to $S_j(p, q)$ of the dyadic cubes λ that cover the points of E_{H_1, H_2} . By definition of E_{H_1, H_2} , they satisfy

$$e_\lambda \sim 2^{-H_1 j} \quad \text{and} \quad f_\lambda \sim 2^{-H_2 j};$$

by definition of $d(H_1, H_2)$, since we use cubes of the same width 2^{-j} to cover E_{H_1, H_2} , we need about $2^{-d(H_1, H_2)j}$ such cubes; therefore the corresponding contribution is of the order of magnitude of

$$2^{-dj} 2^{d(H_1, H_2)j} 2^{-(H_1 p + H_2 q)j} = 2^{-(d - d(H_1, H_2) + H_1 p + H_2 q)j}.$$

When $j \rightarrow +\infty$, the main contribution comes from the smallest exponent, so that

$$\eta(p, q) = \inf_H (d - d(H_1, H_2) + H_1 p + H_2 q). \quad (50)$$

In the next section, we will show that the scaling function $\eta(p, q)$ is a concave function on \mathbb{R} , which is in agreement with the fact that the right-hand side of (50) necessarily is a concave function (as an infimum of a family of linear functions) no matter whether $d(H_1, H_2)$ is concave or not. However, if $d(H_1, H_2)$ also is a concave function, then the Legendre transform in (50) can be inverted (as a consequence of the duality of convex functions), which justifies the following assertion:

Definition 16. A sequence (e_λ, f_λ) follows the multifractal formalism if its spectrum of singularities satisfies

$$d(H_1, H_2) = \inf_{(p, q) \in \mathbb{R}^2} (d - \eta(p, q) + H_1 p + H_2 q). \quad (51)$$

The derivation exposed above is not a mathematical proof, and we do not expect (51) to hold in complete generality. However, in applications, it often happens that the spectrum of singularities itself has no direct interpretation and multifractal analysis is only used as a classification tool in order to discriminate between several types of signals; when such is the case, one is no more concerned with the validity of (51) but only with having its right-hand side defined in a numerically precise way.

Note that, in the specific case where the couple of exponents considered is the Hölder exponent and the oscillation exponent, one applies the above general derivation using wavelet leaders and β -leaders. This justifies the grand-canonical multifractal formalism proposed in [5, 29].

4.5 Concavity of the Scaling Function

We will now prove that the function η is concave on \mathbb{R}^2 , a property that was used in the derivation of the multifractal formalism

Proposition 6. *The function η defined by (49) is concave on \mathbb{R}^2 .*

In order to prove Proposition 6, we will need the following lemma.

Lemma 5. *Let $(a_i)_{i \in \mathbb{N}}$ and $(b_i)_{i \in \mathbb{N}}$ be sequences of positive real numbers. The function $\omega : \mathbb{R}^2 \longrightarrow \overline{\mathbb{R}} (= \mathbb{R} \cup \{+\infty, -\infty\})$ defined by*

$$\omega(p, q) = \log \left(\sum_{i \in \mathbb{N}} a_i^p b_i^q \right)$$

is convex on \mathbb{R}^2 .

Proof. We need to check that $\forall (p, q), (p', q') \in \mathbb{R}^2, \forall \alpha \in]0, 1[$,

$$\omega(\alpha(p, q) + (1 - \alpha)(p', q')) \leq \alpha \omega(p, q) + (1 - \alpha) \omega(p', q'). \quad (52)$$

Consider the sequences

$$A = ((a_1^p b_1^q)^\alpha, \dots, (a_N^p b_N^q)^\alpha, \dots) \quad \text{and} \quad B = ((a_1^{p'} b_1^{q'})^{1-\alpha}, \dots, (a_N^{p'} b_N^{q'})^{1-\alpha}, \dots);$$

Hölder's inequality applied with the conjugate exponents $1/\alpha$ and $1/(1 - \alpha)$ yields

$$\sum_{i=1}^{\infty} a_i^{\alpha p + (1-\alpha)p'} b_i^{\alpha q + (1-\alpha)q'} \leq \left(\sum_{i=1}^{\infty} a_i^p b_i^q \right)^\alpha \left(\sum_{i=1}^{\infty} a_i^{p'} b_i^{q'} \right)^{1-\alpha}.$$

Taking logarithms on both sides of this inequality yields (52). \square

We will now show that η is concave on \mathbb{R}^2 . For each j , one applies Lemma 5 to the sequences (e_λ) and (f_λ) . We obtain that, for any $j \geq 0$, the function

$$(p, q) \rightarrow \log \left(2^{-dj} \sum_{\lambda \in \Lambda_j} d_\lambda^p f_\lambda^q \right)$$

is convex; therefore, after dividing by $\log(2^{-j}) < 0$, we obtain a concave function; since concavity is preserved by taking infimums and pointwise limits, the concavity of the scaling function follows.

5 Concluding Remarks

Let us summarize the whole strategy that we proposed in order to find the relevant multifractal parameters attached to an image. The first step consists in computing the wavelet scaling function $\eta_f(q)$ defined for $q > 0$ by (14), and the additional parameter supplied by the uniform Hölder exponent H_f^{min} defined by (17); note that these parameters can be computed in all cases, without any regularity assumption on the data. They can be used for classification: For instance preliminary studies

made on a collection of natural textures show that their exponents H_f^{min} may widely differ, and therefore that it is a pertinent classification parameter, see [45–47].

These parameters also serve another purpose as a preliminary step in order to go further: H_f^{min} allows to determine if the leader scaling function (defined by (22)) is well defined: It is the case if $H_f^{min} > 0$. This scaling function has already proved useful in many situations: A particularly striking example is supplied by deciding which models for fully developed turbulence are relevant, see [33]: This is a typical situation where H_f^{min} and $\eta_f(q)$ are not discriminatory (for $p > 0$, several models fit the experimental values measured on turbulent data); on the other hand, the fact that $\zeta_f(p)$ is defined for negative values of p yields a range of parameters on which discrimination can be efficiently performed (note that this is in sharp distinction with the wavelet scaling function, which is defined only for $p > 0$). This example also shows the importance of using the corresponding parameters despite the fact that they do not have a function space interpretation, which is the case of scaling functions associated with negative ps , see [3, 4].

If $H_f^{min} < 0$, then we have the choice between two strategies. One consists in performing a pseudo-fractional integration of order larger than $-H_f^{min}$, so that the new signal has a positive uniform Hölder exponent. This strategy has been successfully applied to images, and yields a set of parameters which are well defined, and also proved useful for classification, see [45–47].

On the other hand, this method is not universal, and one can easily check on the following toy examples why this strategy may have for consequence a loss of information on the nature of the singularities present in the signal. Consider for instance the function

$$F(x) = |x - x_1|^\alpha \sin\left(\frac{1}{|x - x_1|^\beta}\right) + |x - x_2|^\delta + B_\gamma(x),$$

where B_γ is a Fractional Brownian Motion (FBM) of order $\gamma \in (0, 1)$. We assume that $0 < \alpha < \gamma$, and $-1 < \delta < 0$ so that $F \in L_{loc}^p$ as soon as $\alpha > -1/p$. Since $\delta < 0$, then, the computation of wavelet leaders cannot be performed, and a fractional integration of order $t > -\delta$ is required so that $H_{f(f)}^{min} > 0$. But if $\alpha + t(1 + \beta) > \gamma$, then this fractional integration make the oscillating singularity disappear, since its exponent becomes larger than the exponent of the integrated FBM. It is easy to check that such situations do happen: If $\alpha > \gamma + \delta\beta$, then either the order of fractional integration is too small, and the wavelet leaders are not well defined, or it is too large, and the oscillating singularity has been too much smoothed, and cannot be detected. This example shows that performing a fractional (or pseudo-fractional) integration has the effect of smoothing the data, and can make the oscillating singularities disappear in the presence of noise. Therefore, a better strategy in order to detect the presence of oscillating singularities can be to deal with a q -scaling function of the signal; in the more favorable cases, we won't have to perform any fractional integration at all, because, for some values of q , $\eta_f(q) > 0$. It is the case in the example we picked, since $\delta > -1$; see also [45–47] for examples taken from natural textures where H_f^{min} is negative, but nonetheless, the wavelet scaling function takes positive

values. In the worst cases where $\eta_f(q)$ is negative for all values of q , then one has to perform a fractional integration; however, one can use the grand scaling function for values of the fractional integration parameter t which are smaller than when dealing with leaders, and thus less information concerning the presence of oscillating singularities will be lost. The previous example showed the relevance of the computation of the wavelet scaling function of f : If, for a given q , $\eta_f(q) > 0$, then the q -scaling function (defined by (31)) is well defined for this value of q . The determination of this scaling function has also been advocated as a useful tools in order to perform a multifractal analysis of fractal boundaries, see [31]. Therefore, this discussion shows situations where the q scaling function can be required in order to perform a multifractal analysis, and even situations where the grand scaling function proves necessary.

Acknowledgement

We are particularly thankful to the anonymous referee for many useful comments, and to Maïtine Bergounioux for her constant help and encouragements during the redaction of this text.

References

1. Abry, P., Jaffard, S., Lashermes, B.: Wavelet leaders in multifractal analysis. In: Qian, T., et al. (eds.) *Wavelet Analysis and Applications. Applied and Numerical Harmonic Analysis*, pp. 201–246. Birkhuser Verlag, Basel, Switzerland (2006)
2. Abry, P., Jaffard, S., Roux, S., Vedel, B., Wendt, H.: Wavelet decomposition of measures: Application to multifractal analysis of images. In: Byrnes, J. (ed.) *Unexploded Ordnance Detection and Mitigation. NATO Science for peace and security, Series B 1–20*. Springer, Berlin (2008)
3. Abry, P., Lashermes, B., Jaffard, S.: Revisiting scaling, multifractal and multiplicative cascades with the wavelet leader lens. In: *Optic East, Wavelet Applications in Industrial Applications II*, vol. 5607, pp. 103–117. Philadelphia, PA (2004)
4. Abry, P., Lashermes, B., Jaffard, S.: Wavelet leader based multifractal analysis. In: *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing* (2005)
5. Abry, P., S. Jaffard, S., Roux, S., Detecting oscillating singularities in multifractal analysis: Application to hydrodynamic turbulence, 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Praha, Cz, May 2011.
6. Abry, P., Sellan, F.: The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer: Remarks and fast implementation, *Appl. Comput. Harmon. Anal.* **3**(4), 377–383 (1996)
7. Arneodo, A., Bacry, E., Jaffard, S., Muzy, J.-F.: Singularity spectrum of multifractal functions involving oscillating singularities *J. Four. Anal. Appl.* **4**, 159–174 (1998)
8. Arneodo, A., Bacry, E., Muzy, J.-F.: The thermodynamics of fractals revisited with wavelets. *Phys. A* **213**, 232–275 (1995)
9. Aubry, J.-M.: Representation of the singularities of a function, *Appl. Comp. Harmon. Anal.* **6**, 282–286 (1999)

10. Aubry, J.-M., Jaffard, S.: Random wavelet series. *Comm. Math. Phys.* **227**, 483–514 (2002)
11. Aujol, J.-F., Chambolle, A.: Dual norms and image decomposition models, *Int. J. Comput. Vis.* **63**, 85–104 (2005)
12. Brown, G., Michon, G.: J. Peyrière, On the multifractal analysis of measures. *J. Statist. Phys.* **66**, 775–790 (1992)
13. Calderòn, A.P., Zygmund, A.: Singular integral operators and differential equations. *Am. J. Math.* **79**, 901–921 (1957)
14. Cohen, A., Daubechies, I., Fauveau, J.-C.: Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **45**, 485–560 (1992)
15. Cohen, A., Ryan, R.: *Wavelets and Multiscale Signal Processing*. Chapman and Hall, London (1995)
16. Daubechies, I.: *Ten Lectures on Wavelets*. S.I.A.M, Philadelphia, PA (1992)
17. Falconer, K.: *Fractal Geometry*. Wiley, Chichester (1990)
18. Feichtinger, H.: Wiener amalgams over Euclidean spaces and some of their applications. *Lecture Notes Pure and Applied Mathematics*, vol. 136, pp. 123–137. Springer, Berlin (1992)
19. Gousseau, Y., Morel, J.-M.: Are natural images of bounded variation? *SIAM J. Math. Anal.* **33**, 634–648 (2001)
20. Halsey, T., Jensen, M., Kadanoff, L., Procaccia, I., Shraiman, B.: Fractal measures and their singularities: The characterization of strange sets. *Phys. Rev. A* **33**, 1141–1151 (1986)
21. Jaffard, S.: Pointwise smoothness, two-microlocalization and wavelet coefficients. *Publ. Matem.* **35**, 155–168 (1991)
22. Jaffard, S.: Multifractal formalism for functions. *SIAM J. Math. Anal.* **28**, 944–998 (1997)
23. Jaffard, S.: Lacunary wavelet series. *Ann. App. Prob.* **10**(1), 313–329 (2000)
24. Jaffard, S.: Beyond Besov spaces Part 2: Oscillation spaces. *Const. Approx.* **21**(1), 29–61 (2005)
25. Jaffard, S.: Wavelet techniques in multifractal analysis. *Fractal Geometry and Applications: A Jubilee of Benoît Mandelbrot*. In: Lapidus et, M., van Frankenhuijsen, M. (eds.) *Proceedings of Symposia in Pure Mathematics*, vol.72, Part 2, pp. 91–152. AMS (2004)
26. Jaffard, S.: Oscillation spaces: Properties and applications to fractal and multifractal functions. *J. Math. Phys.* **39**, 4129–4141 (1998)
27. Jaffard, S.: Pointwise regularity associated with function spaces and multifractal analysis. *Banach Center Publ.* **72**, 93–110 (2006). Figiel, T., Kamont, A. (eds.) *Approximation and Probability*
28. Jaffard, S.: Wavelet Techniques for pointwise regularity. *Ann. Fac. Sci. Toul.* **15**(1), 3–33 (2006)
29. Jaffard, S., Abry, P., Roux, S.: Singularités oscillantes et coefficients d’ondelettes dominants. *Proceedings of the GRETSI Conference at Dijon, France* (2009)
30. Jaffard, S., Abry, P., Roux, S., Vedel, B., Wendt, H.: The contribution of wavelets in multifractal analysis. In: *Wavelet methods in Mathematical Analysis and Engineering. Series in Contemporay Applied Mathematics, CAM*, vol. 14, 51–98. Higher Education Press (2010)
31. Jaffard, S., Melot, C.: Wavelet analysis of fractal Boundaries, Part 1: Local regularity and Part 2: Multifractal formalism. *Comm. Math. Phys.* **258**(3), 513–539 (2005)
32. Jaffard, S., Meyer, Y.: Wavelet methods for pointwise regularity and local oscillations of functions. *Mem. Am. Math. Soc.* **123**(587) (1996)
33. Lashermes, B., Roux, S., Abry, P., Jaffard, S.: Comprehensive multifractal analysis of turbulent velocity using wavelet leaders. *Eur. Phys. J. B.* **61**(2), 201–215 (2008)
34. Lévy-Véhel, J., Seuret, S.: The local Hölder function of a continuous function. *Appl. Comp. Harm. Anal.* **13**(3), 263–276 (2002)
35. Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic, San Diego (1998)
36. Meyer, Y.: *Ondelettes et Opérateurs*. Hermann, Paris (1990)
37. Meyer, Y.: *Wavelets, Vibrations and Scalings*, CRM Ser. AMS vol. 9. Presses de l’Université de Montréal (1998)
38. Meyer, Y.: Oscillating patterns in image processing and nonlinear evolution equations. *University Lecture Series*, vol. 22, AMS, Providence, RI (2001)

39. Meyer, Y., Sellan, F., Taqqu, M.: Wavelets, generalized white noise and fractional integration: The synthesis of fractional Brownian motion. *J. Fourier Anal. Appl.* **5**(5), 465–494 (1999)
40. Meyer, Y., Xu, H.: Wavelet analysis and chirps. *Appl. Comput. Harmon. Anal.* **4**, 366–379 (1997)
41. Osher, S., Solé, A., Vese, L.: Image decomposition and restoration using total variation minimization and the L^1 norm. *Multiscale Model Simul.* **1**, 349–370 (2003)
42. Parisi, G., Frisch, U.: On the singularity structure of fully developed turbulence; appendix to Fully developed turbulence and intermittency, by U. Frisch. In: *Proceedings of International Summer school Phys. Enrico Fermi*, North Holland, pp. 84–88. (1985)
43. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**, 259–268 (1992)
44. Seeger, A.: A note on Triebel-Lizorkin spaces. *Approx. Funct. Spaces Banach Center Publ.* **22**, 391–400 (1989)
45. Wendt, H., Abry, P., Jaffard, S.: Bootstrap for empirical multifractal analysis. *IEEE Signal Proc. Mag.* **24**(4), 38–48 (2007)
46. Wendt, H., Abry, P., Roux, S., Jaffard, S., Vedel, B.: Analyse multifractale d’images: l’apport des coefficients dominants. *Traitement du Signal* **25**(4–5), 47–65 (2009)
47. Wendt, H., Roux, S., Abry, P., Jaffard, S.: Wavelet leaders and Bootstrap for multifractal analysis of images. *Signal Process.* **89**, 1100–1114 (2009)

A Second Order Model for 3D-Texture Extraction

Maïtine Bergounioux and Minh Phuong Tran

Abstract In this paper we present the 3D-implementation of a second-order model for texture extraction that has been fully described in (Bergounioux and Piffet, Set Valued Anal. Variational Anal. 18(3–4):277–306 (2010)). Numerical experimentation has been performed for 2D-images. We generalize the discrete model to the 3D case. In particular we describe the whole discretization process. In addition, we add an algorithmic modification that improves texture extraction using a modified Hessian matrix. We end with numerical examples arising in biomedical imaging.

1 Introduction

In this paper we present the 3D-implementation of a second-order model for texture extraction that has been fully described in [3]. Numerical experimentation was performed for 2D-images. We generalize the discrete model to the 3D case. In particular we describe the complete discretization scheme. In addition, we add an algorithmic modification that improves texture extraction significantly using a modified Hessian matrix. This is also a generalization of the 2D-case (see Piffet [9, 10]). First, we recall the main definitions and present the generic second order variational model. Section 2 is devoted to the 3D-discretization and implementation. Then we present an “anisotropic” improvement of the algorithm which takes into account the (local) contours to compute the second-order derivative. We end with numerical examples arising in biomedical imaging, namely angiography MRI images.

Maïtine Bergounioux (✉) and Minh Phuong Tran
Université d’Orléans, Laboratoire MAPMO – UMR 6628, Fédération Denis-Poisson,
BP 6759, F-45067 Orléans Cedex 2, France
e-mail: maitine.bergounioux@univ-orleans.fr, minh_phuong.tran@univ-orleans.fr

1.1 Bounded Variation Spaces of First and Second Order

Let $\Omega \subset \mathbb{R}^n$ ($n \geq 2$) be an open bounded set. The space of functions of bounded variation, $BV(\Omega)$ is well known. We refer to [1, 2, 4] for example. We denote by $TV(u)$ the total variation of $u \in BV(\Omega)$:

$$TV(u) = \sup \left\{ \int_{\Omega} u \operatorname{div} \varphi \, dx : \varphi \in \mathcal{C}_0^1(\Omega), \|\varphi\|_{\infty} \leq 1 \right\} \quad (1)$$

Following Demengel [7] and Piffet [9] we may define the space of functions of bounded second-order variation (or hessian bounded) as

$$BV^2(\Omega) := \{u \in W^{1,1}(\Omega) \mid TV2(u) < +\infty\}.$$

Here the second-order total variation is defined as

$$TV2(u) := \sup \left\{ \int_{\Omega} \langle \nabla u, \operatorname{div}(\varphi) \rangle_{\mathbb{R}^n} \mid \varphi \in \mathcal{C}_c^2(\Omega, \mathbb{R}^{n \times n}), \|\varphi\|_{\infty} \leq 1 \right\} \quad (2)$$

where

$$\operatorname{div}(\varphi) = (\operatorname{div}(\varphi_1), \operatorname{div}(\varphi_2), \dots, \operatorname{div}(\varphi_n)),$$

with

$$\forall i, \varphi_i = (\varphi_i^1, \varphi_i^2, \dots, \varphi_i^n) \in \mathbb{R}^n \text{ and } \operatorname{div}(\varphi_i) = \sum_{k=1}^n \frac{\partial \varphi_i^k}{\partial x_k}.$$

The space $BV^2(\Omega)$ endowed with the norm

$$\|u\|_{BV^2(\Omega)} = \|u\|_{W^{1,1}(\Omega)} + TV2(u)$$

is a Banach space. Moreover, it has been proved in [9] that

$$BV^2(\Omega) = \left\{ u \in W^{1,1}(\Omega) \mid \forall i \in \{1, 2, \dots, n\} : \frac{\partial u}{\partial x_i} \in BV(\Omega) \right\}.$$

1.2 The Abstract Second-Order Model

We recall the variational model described in [3]. We refer to this paper for a precise motivation of this second-order model. Let $\Omega \subset \mathbb{R}^n$ be an open bounded set (smooth enough, for example with Lipschitz boundary). We consider the following functional:

$$\begin{aligned} F : BV^2(\Omega) &\rightarrow \mathbb{R}^+ \\ (v) &\mapsto F(v) \end{aligned}$$

$$F(v) = \frac{1}{2} \|u_d - v\|_{L^2(\Omega)}^2 + \lambda TV2(v) + \delta \|v\|_{W^{1,1}(\Omega)}$$

where $u_d \in L^2(\Omega)$ and $\lambda, \delta \geq 0$ and we are looking for a solution to the optimization problem:

$$\inf_{v \in BV^2(\Omega)} F(v) \quad (3)$$

It has been proved in [3] that problem (3) has a unique solution for $\lambda > 0$ and $\delta > 0$. However, this result is still true for the discretized problem even with $\delta = 0$. Moreover, in [4] we prove that the existence result still holds true for the infinite dimensional problem if the function v satisfies $\frac{\partial v}{\partial n}|_{\partial\Omega} = 0$ and $\Omega = \prod_{i=1}^n]a_i, b_i[$ is a square subset of \mathbb{R}^n . In what follows, we investigate the finite-dimensional problem, so we assume that $\delta = 0$.

1.3 Discretization of the 3D: Problem

In [3] the problem has been discretized in the case of 2D images and numerical tests have been performed. Here we generalize this work to the 3D-case and extend the anisotropic correction of the algorithm of [10]. In the sequel, $n = 3$ and the image size is $N_1 \times N_2 \times N_3$. The generic component of u is $u_{i,j,k}$ and we denote similarly the continuous function (previous section) and the corresponding (discretized) tensor. We denote $X = \mathbb{R}^{N_1 \times N_2 \times N_3}$ endowed with inner product and norm

$$\langle u, v \rangle_X = \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} u_{i,j,k} v_{i,j,k} \text{ and } \|u\|_X = \sqrt{\sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} u_{i,j,k}^2}$$

and set $Y = X \times X \times X$.

(a) We first compute the discrete gradient $\nabla u \in Y$ of the image $u \in X$:

$$(\nabla u_{i,j,k}) = (\nabla u_{i,j,k}^1, \nabla u_{i,j,k}^2, \nabla u_{i,j,k}^3)$$

where

$$\begin{aligned} \nabla u_{i,j,k}^1 &= \begin{cases} u_{i+1,j,k} - u_{i,j,k} & i < N_1 \\ 0 & i = N_1 \end{cases} \\ \nabla u_{i,j,k}^2 &= \begin{cases} u_{i,j+1,k} - u_{i,j,k} & j < N_2 \\ 0 & j = N_2 \end{cases} \\ \nabla u_{i,j,k}^3 &= \begin{cases} u_{i,j,k+1} - u_{i,j,k} & k < N_3 \\ 0 & k = N_3 \end{cases} \end{aligned}$$

(b) Discretization of the term $TV2(v)$.

We have

$$\langle \nabla u, \text{div} \phi \rangle = - \langle \phi, \nabla^2 u \rangle.$$

Then,

$$TV2(v) \simeq \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} \|(Hv)_{i,j,k}\|_{\mathbb{R}^9}$$

where

$$(Hv)_{i,j,k} = (Hv_{i,j,k}^{11}, Hv_{i,j,k}^{12}, Hv_{i,j,k}^{13}, Hv_{i,j,k}^{21}, Hv_{i,j,k}^{22}, Hv_{i,j,k}^{23}, Hv_{i,j,k}^{31}, Hv_{i,j,k}^{32}, Hv_{i,j,k}^{33}).$$

For every $i = 1, \dots, N_1$, $j = 1, \dots, N_2$ and $k = 1, \dots, N_3$, the computation of Hv gives

$$\begin{aligned} (Hv)_{i,j,k}^{11} &= \begin{cases} v_{i+1,j,k} - v_{i,j,k} + v_{i-1,j,k} & 1 < i < N_1 \\ v_{i+1,j,k} - v_{i,j,k} & i = 1 \\ v_{i,j,k} - v_{i-1,j,k} & i = N_1 \end{cases} \\ (Hv)_{i,j,k}^{12} &= \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i-1,j+1,k} + v_{i-1,j,k} & 1 < i \leq N_1 \\ 0 & 1 \leq j < N_2 \\ 0 & j = N_2 \\ 0 & i = 1 \end{cases} \\ (Hv)_{i,j,k}^{13} &= \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i-1,j,k+1} + v_{i-1,j,k} & 1 < i \leq N_1 \\ 0 & 1 \leq k < N_3 \\ 0 & k = N_3 \\ 0 & i = 1 \end{cases} \\ (Hv)_{i,j,k}^{21} &= \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j-1,k} + v_{i,j-1,k} & 1 \leq i < N_1 \\ 0 & 1 < k \leq N_3 \\ 0 & i = N_1 \\ 0 & k = 1 \end{cases} \\ (Hv)_{i,j,k}^{22} &= \begin{cases} v_{i,j+1,k} - v_{i,j,k} + v_{i,j-1,k} & 1 < j < N_2 \\ v_{i,j+1,k} - v_{i,j,k} & j = 1 \\ v_{i,j,k} - v_{i,j-1,k} & j = N_2 \end{cases} \\ (Hv)_{i,j,k}^{23} &= \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i,j-1,k+1} + v_{i,j-1,k} & 1 < j \leq N \\ 0 & 1 \leq k < N_3 \\ 0 & j = 1 \\ 0 & k = N_3 \end{cases} \\ (Hv)_{i,j,k}^{31} &= \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j,k-1} + v_{i,j,k-1} & 1 < k \leq N_3 \\ 0 & 1 \leq i < N_1 \\ 0 & k = 1 \\ 0 & i = N_1 \end{cases} \\ (Hv)_{i,j,k}^{32} &= \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i,j+1,k-1} + v_{i,j,k-1} & 1 \leq j < N \\ 0 & 1 < k \leq N_3 \\ 0 & j = N_2 \\ 0 & k = 1 \end{cases} \end{aligned}$$

$$(Hv)_{i,j,k}^{33} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} + v_{i,j,k-1} & 1 < k < N_3 \\ v_{i,j,k+1} - v_{i,j,k} & k = 1 \\ v_{i,j,k} - v_{i,j,k-1} & k = N_3 \end{cases}$$

1.4 Numerical Computation of the Solution of (3)

Let us consider $H^* : X^9 \rightarrow X$ defined as follows (H^* is the adjoint of operator H): for every $p = (p^{11}, p^{12}, p^{13}, p^{21}, p^{22}, p^{23}, p^{31}, p^{32}, p^{33}) \in X^9$,

$$(H^*p)_{i,j,k} = \sigma_{i,j,k}^{11} + \sigma_{i,j,k}^{12} + \sigma_{i,j,k}^{13} + \sigma_{i,j,k}^{21} + \sigma_{i,j,k}^{22} + \sigma_{i,j,k}^{23} + \sigma_{i,j,k}^{31} + \sigma_{i,j,k}^{32} + \sigma_{i,j,k}^{33}$$

where

$$\begin{aligned} \sigma_{i,j,k}^{11} &= \begin{cases} p_{i+1,j,k}^{11} - 2p_{i,j,k}^{11} + p_{i-1,j,k}^{11} & 1 < i < N_1 \\ p_{i+1,j,k}^{11} - p_{i,j,k}^{11} & i = 1 \\ p_{i-1,j,k}^{11} - p_{i,j,k}^{11} & i = N_1 \end{cases} \\ \sigma_{i,j,k}^{22} &= \begin{cases} p_{i,j+1,k}^{22} - 2p_{i,j,k}^{22} + p_{i,j-1,k}^{22} & 1 < j < N_2 \\ p_{i,j+1,k}^{22} - p_{i,j,k}^{22} & j = 1 \\ p_{i,j-1,k}^{22} - p_{i,j,k}^{22} & j = N_2 \end{cases} \\ \sigma_{i,j,k}^{33} &= \begin{cases} p_{i,j,k+1}^{33} - 2p_{i,j,k}^{33} + p_{i,j,k-1}^{33} & 1 < k < N_3 \\ p_{i,j,k+1}^{33} - p_{i,j,k}^{33} & k = 1 \\ p_{i,j,k-1}^{33} - p_{i,j,k}^{33} & k = N_3 \end{cases} \\ \sigma_{i,j,k}^{12} &= \begin{cases} p_{i+1,j,k}^{12} & i = 1, j = 1 \\ -p_{i+1,j-1,k}^{12} & i = 1, j = N_2 \\ p_{i+1,j,k}^{12} - p_{i+1,j-1,k}^{12} & i = 1, 1 < j < N_2 \\ -p_{i,j,k}^{12} & i = N_1, j = 1 \\ p_{i,j-1,k}^{12} & i = N_1, j = N_2 \\ p_{i,j-1,k}^{12} - p_{i,j,k}^{12} & i = N_1, 1 < j < N_2 \\ p_{i+1,j,k}^{12} - p_{i,j,k}^{12} & 1 < i < N_1, j = 1 \\ p_{i,j-1,k}^{12} - p_{i+1,j-1,k}^{12} & 1 < i < N_1, j = N_2 \\ p_{i,j-1,k}^{12} - p_{i,j,k}^{12} - p_{i+1,j-1,k}^{12} + p_{i+1,j,k}^{12} & 1 < i < N_1, 1 < j < N_2 \end{cases} \end{aligned}$$

$$\sigma_{i,j,k}^{13} = \begin{cases} p_{i+1,j,k}^{13} & i = 1, k = 1 \\ -p_{i+1,j,k-1}^{13} & i = 1, k = N_3 \\ p_{i+1,j,k}^{13} - p_{i+1,j,k-1}^{13} & i = 1, 1 < j < N_3 \\ -p_{i,j,k}^{13} & i = N_1, k = 1 \\ p_{i,j,k-1}^{13} & i = N_1, k = N_3 \\ p_{i,j,k-1}^{13} - p_{i,j,k}^{13} & i = N_1, 1 < k < N_3 \\ p_{i+1,j,k}^{13} - p_{i,j,k}^{13} & 1 < i < N_1, k = 1 \\ p_{i,j,k-1}^{13} - p_{i+1,j,k-1}^{13} & 1 < i < N_1, k = N_3 \\ p_{i,j,k-1}^{13} - p_{i,j,k}^{13} - p_{i+1,j,k-1}^{13} + p_{i+1,j,k}^{13} & 1 < i < N_1, 1 < k < N_3 \end{cases}$$

$$\sigma_{i,j,k}^{21} = \begin{cases} p_{i,j+1,k}^{21} & j = 1, i = 1 \\ -p_{i-1,j+1,k}^{21} & j = 1, i = N_1 \\ p_{i,j+1,k}^{21} - p_{i-1,j+1,k}^{21} & j = 1, 1 < i < N_1 \\ -p_{i,j,k}^{21} & j = N_2, i = 1 \\ p_{i-1,j,k}^{21} & j = N_2, i = N_1 \\ p_{i-1,j,k}^{21} - p_{i,j,k}^{21} & j = N_2, 1 < i < N_1 \\ p_{i,j+1,k}^{21} - p_{i,j,k}^{21} & 1 < j < N_2, i = 1 \\ p_{i-1,j,k}^{21} - p_{i-1,j+1,k}^{21} & 1 < j < N_2, i = N_1 \\ p_{i-1,j,k}^{21} - p_{i,j,k}^{21} - p_{i-1,j+1,k}^{21} + p_{i,j+1,k}^{21} & 1 < j < N_2, 1 < i < N_1 \end{cases}$$

$$\sigma_{i,j,k}^{23} = \begin{cases} p_{i,j+1,k}^{23} & j = 1, k = 1 \\ -p_{i,j+1,k-1}^{23} & j = 1, k = N_3 \\ p_{i,j+1,k}^{23} - p_{i,j+1,k-1}^{23} & j = 1, 1 < k < N_3 \\ -p_{i,j,k}^{23} & j = N_2, k = 1 \\ p_{i,j,k-1}^{23} & j = N_2, k = N_3 \\ p_{i,j,k-1}^{23} - p_{i,j,k}^{23} & j = N_2, 1 < k < N_3 \\ p_{i,j+1,k}^{23} - p_{i,j,k}^{23} & 1 < j < N_2, k = 1 \\ p_{i,j,k-1}^{23} - p_{i,j+1,k-1}^{23} & 1 < j < N_2, k = N_3 \\ p_{i,j,k-1}^{23} - p_{i,j,k}^{23} - p_{i,j+1,k-1}^{23} + p_{i,j+1,k}^{23} & 1 < j < N_2, 1 < k < N_3 \end{cases}$$

$$\sigma_{i,j,k}^{31} = \begin{cases} p_{i,j,k+1}^{31} & k = 1, i = 1 \\ -p_{i-1,j,k+1}^{31} & k = 1, i = N_1 \\ p_{i,j,k+1}^{31} - p_{i-1,j,k+1}^{31} & k = 1, 1 < i < N_1 \\ -p_{i,j,k}^{31} & k = N_3, i = 1 \\ p_{i-1,j,k}^{31} & k = N_3, i = N_1 \\ p_{i-1,j,k}^{31} - p_{i,j,k}^{31} & k = N_3, 1 < i < N_1 \\ p_{i,j,k+1}^{31} - p_{i,j,k}^{31} & 1 < k < N_3, i = 1 \\ p_{i-1,j,k}^{31} - p_{i-1,j,k+1}^{31} & 1 < k < N_3, i = N_1 \\ p_{i-1,j,k}^{31} - p_{i,j,k}^{31} - p_{i-1,j,k+1}^{31} + p_{i,j,k+1}^{31} & 1 < k < N_3, 1 < i < N_1 \end{cases}$$

$$\sigma_{i,j,k}^{32} = \begin{cases} p_{i,j,k+1}^{32} & k = 1, j = 1 \\ -p_{i,j-1,k+1}^{32} & k = 1, j = N_2 \\ p_{i,j,k+1}^{32} - p_{i,j-1,k+1}^{32} & k = 1, 1 < j < N_2 \\ -p_{i,j,k}^{32} & k = N_3, j = 1 \\ p_{i,j-1,k}^{32} & k = N_3, j = N_2 \\ p_{i,j-1,k}^{32} - p_{i,j,k}^{32} & k = N_3, 1 < j < N_2 \\ p_{i,j,k+1}^{32} - p_{i,j,k}^{32} & 1 < k < N_3, j = 1 \\ p_{i,j-1,k}^{32} - p_{i,j-1,k+1}^{32} & 1 < k < N_3, j = N_2 \\ p_{i,j-1,k}^{32} - p_{i,j,k}^{32} - p_{i,j-1,k+1}^{32} + p_{i,j,k+1}^{32} & 1 < k < N_3, 1 < j < N_2 \end{cases}$$

It is straightforward to prove that

Theorem 1. *The solution to problem (3) verifies:*

$$v = ud - P_{\lambda K}(u_d)$$

where $P_{\lambda K}$ is the orthogonal projector operator on λK and

$$K := \{H^*p \mid p \in X^9, \|p_{i,j,k}\|_{\mathbb{R}^9} \leq 1, 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3\}.$$

Proof. It is quite similar to the 2D-case proof. We refer to [3]. \square

To compute $P_{\lambda K}(u_d)$ we have to solve the following problem:

$$\begin{cases} \min \|\lambda H^*p - u_d\|_X^2 \\ p \in X^9 \\ \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \leq 1, 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \end{cases}$$

Following [6] and [3] we use the following algorithm to compute $P_{\lambda K}(u_d)$

Algorithm

Choose $\tau > 0$

1. Let $p^0 = 0, n = 0$.
2. Suppose p^n is known, we compute p^{n+1} as follows:

$$p_{i,j,k}^n = p_{i,j,k}^{n+1} + \tau \left[(H \left[H^*p - \frac{u_d}{\lambda} \right])_{i,j,k} + \left\| (H \left[H^*p^n - \frac{u_d}{\lambda} \right])_{i,j,k} \right\|_{\mathbb{R}^9} p_{i,j,k}^{n+1} \right]$$

which implies:

$$p_{i,j,k}^{n+1} = \frac{p_{i,j,k}^n - \tau (H \left[H^*p^n - \frac{u_d}{\lambda} \right])_{i,j,k}}{1 + \tau \left\| (H \left[H^*p^n - \frac{u_d}{\lambda} \right])_{i,j,k} \right\|_{\mathbb{R}^9}}$$

Theorem 2. *Let $\tau \leq 1/8^3$, then $\lambda(H^*p^n)_n$ converges to $P_{\lambda K_2}(u_d)$ as $n \rightarrow \infty$.*

Proof. Once again the proof is quite technical but similar to the 2D-case proof [3].

\square

2 Introducing Anisotropy

L. Piffet [8–10] has observed (in the 2D-case) that cancelling one or more coefficients of the Hessian matrix permits to get rid of the contours along the corresponding direction.

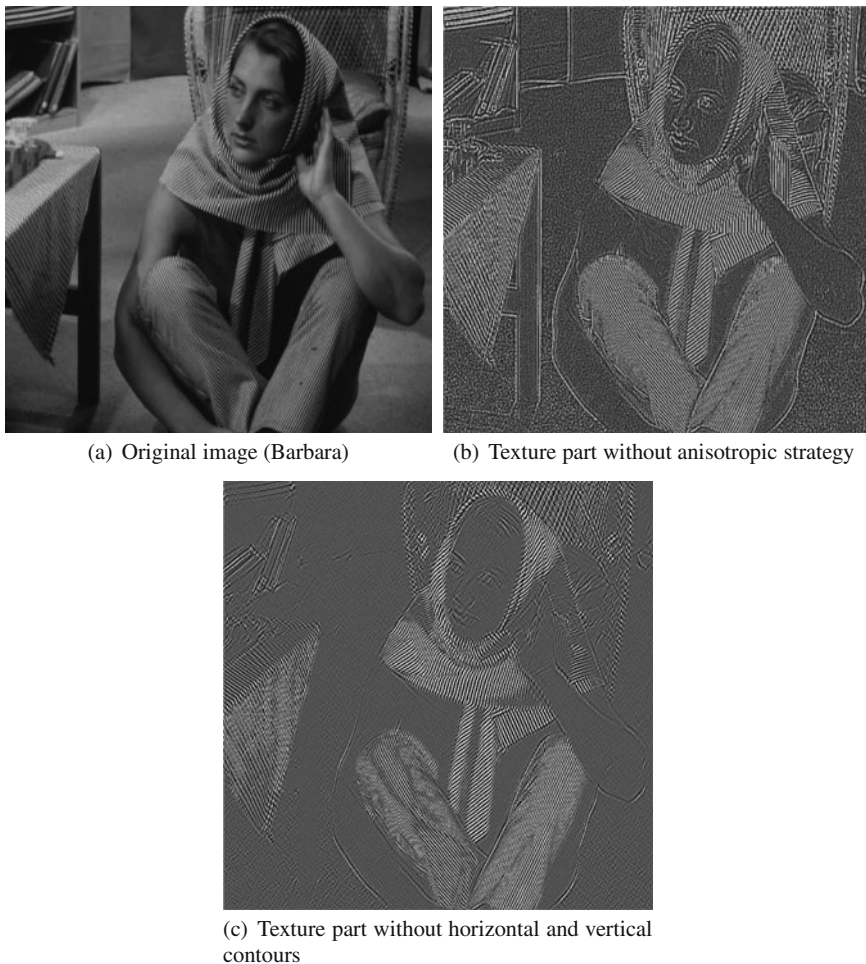


Fig. 1 Effects of anisotropic improvement strategy

We give a 2D-example in Fig. 1: here the coefficients $(H\nu)^{1,1}$ and $(H\nu)^{2,2} = 0$ have been globally set to 0. We can see that horizontal and vertical contours are not involved in the texture part any longer. This method has been improved since there were two major inconveniences:

- First, the same transform is performed at every pixel, so that the image is globally treated. All the vertical and horizontal lines are removed.
- Second, the transform is depended on the chosen (fixed) cartesian axis and it is not possible to remove contours that are not horizontal, vertical or diagonal.

Therefore, the Hessian matrix is now locally computed at every pixel. First, a rotation is performed so that the gradient direction is the new y -axis (or x -axis). The

corresponding Hessian matrix is computed and suitable coefficients are canceled. Then the inverse rotation is performed. For more details on can refer to [9, 10].

We compute the (local) 3D- Hessian matrix at a voxel (i, j, k) using this technique. We have to perform two rotations r_α and r_β to compute an modified hessian matrix H' . More precisely, we perform a change of variables (with the rotations) to compute the Hessian matrix and the adjoint matrix as in the previous section: the local axis (with the gradient vector as z -axis) are considered instead of the original fixed cartesian axis. Then, we may cancel the Hessian matrix terms corresponding to the gradient direction (for example), to get rid of the corresponding contour (if it is significant) in the extracted texture. Finally we go back to the original axis with the inverse rotations. Let us detail the process:

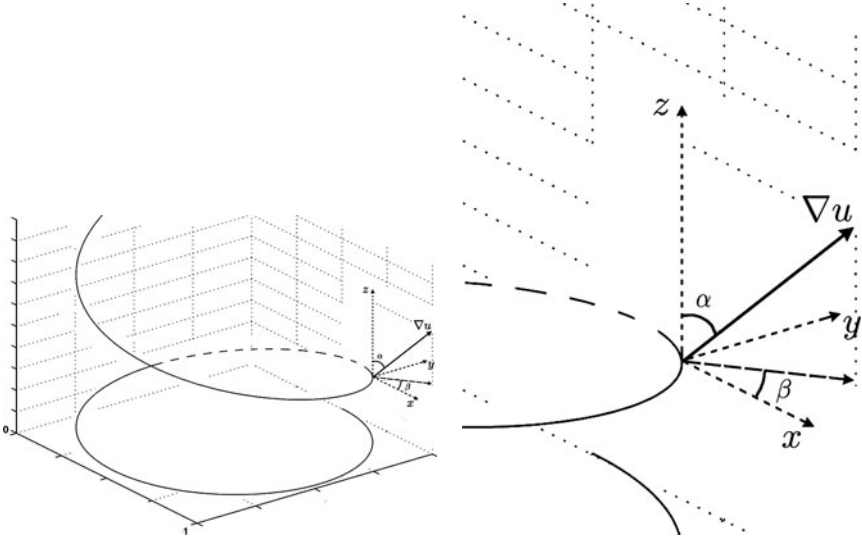


Fig. 2 Definition of local axis and angles α and β

The angles α and β are defined at point $X_o = (x_o, y_o, z_o)$ as follows: α is the (azimuthal) angle between the gradient $\nabla u(x_o, y_o, z_o)$ and the z -axis. β is the angle between the orthogonal projection of

$$\nabla u(x_o, y_o, z_o) := \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} (x_o, y_o, z_o)$$

(on the xOy plane) and the x -axis. Note that we can perform this transformation with axis Ox or Oy instead of Oz (Fig. 2). Let us define the two rotations: r_α and r_β which matrices are:

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \text{ and } R_\beta = \begin{pmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

with

$$\alpha = \text{atan} \left(\frac{u_z}{\sqrt{u_x^2 + u_y^2}} \right) (X_o), \quad \beta = \text{atan} \left(\frac{u_y}{u_x} \right) (X_o).$$

The change of variables from the fixed basis to the local one is given par

$$\tilde{X} = R_\beta R_\alpha X, \quad \text{with } X = (x, y, z) \in \mathbb{R}^3.$$

Moreover

$$X = (R_\beta R_\alpha)^{-1} \tilde{X} = R_\alpha^{-1} R_\beta^{-1} \tilde{X} = R_{-\alpha} R_{-\beta} \tilde{X}.$$

In the sequel, we set $\tilde{u}(\tilde{X}) := u(X)$ and $R_{\alpha,\beta} \stackrel{\text{def}}{=} R_{-\alpha} R_{-\beta}$ and we compute the first and second order derivative of \tilde{u} :

$$\nabla \tilde{u} = \begin{pmatrix} \frac{\partial \tilde{u}}{\partial \tilde{x}} \\ \frac{\partial \tilde{u}}{\partial \tilde{y}} \\ \frac{\partial \tilde{u}}{\partial \tilde{z}} \end{pmatrix} \quad \text{and} \quad \tilde{H} := \begin{pmatrix} \frac{\partial^2 \tilde{u}}{\partial \tilde{x}^2} & \frac{\partial^2 \tilde{u}}{\partial \tilde{x} \partial \tilde{y}} & \frac{\partial^2 \tilde{u}}{\partial \tilde{x} \partial \tilde{z}} \\ \frac{\partial^2 \tilde{u}}{\partial \tilde{x} \partial \tilde{y}} & \frac{\partial^2 \tilde{u}}{\partial \tilde{y}^2} & \frac{\partial^2 \tilde{u}}{\partial \tilde{y} \partial \tilde{z}} \\ \frac{\partial^2 \tilde{u}}{\partial \tilde{x} \partial \tilde{z}} & \frac{\partial^2 \tilde{u}}{\partial \tilde{y} \partial \tilde{z}} & \frac{\partial^2 \tilde{u}}{\partial \tilde{z}^2} \end{pmatrix}.$$

A short computation gives

$$\frac{\partial \tilde{u}}{\partial \tilde{x}} = \frac{\partial u}{\partial x} \frac{\partial \tilde{x}}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \tilde{y}}{\partial x} + \frac{\partial u}{\partial z} \frac{\partial \tilde{z}}{\partial x} = \nabla u \cdot \frac{\partial \tilde{X}}{\partial x} = \nabla u \cdot R(:, 1),$$

where \cdot denotes the \mathbb{R}^3 scalar product and $R(:, 1)$ is the first column of R . Finally, we get

$$\nabla \tilde{u} = R_{\alpha,\beta} \nabla u. \quad (4)$$

Now we compute \tilde{H} ; we set $\tilde{v} = \frac{\partial \tilde{u}}{\partial \tilde{x}}$ and estimate $\nabla \tilde{v}$ as above: this will be the first column of \tilde{H} .

$$\nabla \tilde{v} = R_{\alpha,\beta} \nabla v = R_{\alpha,\beta} \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial y \partial x} \\ \frac{\partial^2 u}{\partial z \partial x} \end{pmatrix}.$$

Finally

$$\tilde{H} = R_{\alpha,\beta} H. \quad (5)$$

As already mentioned, the idea is to cancel some terms of the Hessian matrix to get rid of (or to keep) the contours. However, without performing the rotations, there would be only few possible directions, for example vertical, horizontal and diagonal

in the 2D-case so that many contours are not considered. Performing the change of variables allows to identify the gradient direction (that is the contour direction if the gradient is large enough) with the z -axis and then cancel corresponding terms of the matrix \tilde{H} . Of course, we have to get back to the original situation. Let us denote by \mathcal{L} the (linear) transformation that assigns 0 to some coefficients of \tilde{H} (this is a projection). The whole process is described by

$$H \rightarrow \tilde{H} = R_{-\alpha} R_{-\beta} H \rightarrow \mathcal{L}(\tilde{H}) := \tilde{H}' \rightarrow [R_{\alpha, \beta}]^{-1} \mathcal{L}(\tilde{H}) = R_{\beta} R_{\alpha} \mathcal{L}(\tilde{H}) ,$$

that is

$$H \rightarrow [R_{\beta} R_{\alpha} \mathcal{L} R_{-\alpha} R_{-\beta}] H . \quad (6)$$

So, algorithm p. 48 is modified as follows

Algorithm

Choose $\tau > 0, \mu > 0$ and compute ∇u . Use a threshold process to identify the contours ($\|\nabla u\| \geq \mu$). Set I_{μ} the set of voxels corresponding to these “significant contours”.

1. Let $p^0 = 0, n = 0$.

For voxels in I_{μ} , modify H with the following rule

$$H \rightarrow \tilde{H} = R_{-\alpha} R_{-\beta} H \rightarrow \mathcal{L}(\tilde{H}) = [\mathcal{L} R_{-\alpha} R_{-\beta}] H := H'$$

and compute $(H')^*$

2. Same as before p.48 with H' instead of H .

3 Numerical Examples

Numerical experimentation has been done in the context of biomedical imaging. We consider a stack of 50 MRI images of the vessel network of brain mice. The challenge is to identify the network to get structural informations. Using 2D segmentation and interpolation methods is not possible, since the slices are not exploitable (see Fig. 3).

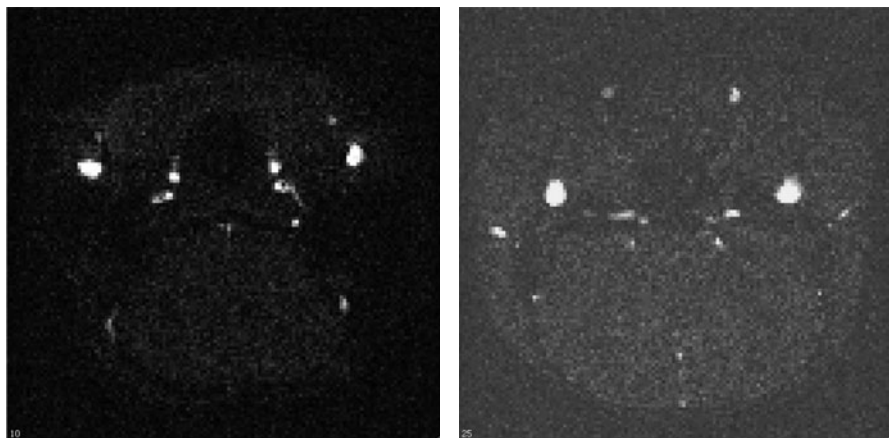
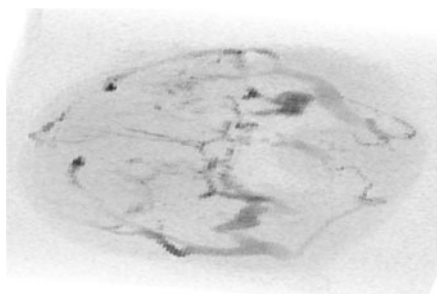


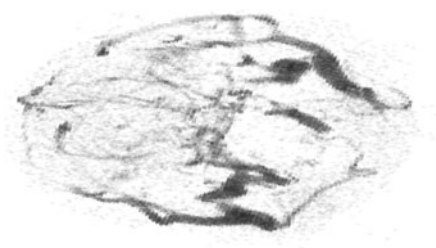
Fig. 3 2D slices example (slices 10 and 25)

Therefore we have to deal with the complete 3D information. We consider that noise and very small vessels effect is texture. Extracting texture gives the remainder part, the so-called “cartoon” (smooth part). We expect that the contours are kept in the cartoon part which in the cleaned image in some sense. Then classical segmentation methods (as threshold for example) can be used. The following results have been obtained without any anisotropic strategy. Indeed, computational time is large and we still have to improve the speed of algorithm. However, we present a comparison between the two methods with and without anisotropy strategy. The results show that the anisotropy technique is quite efficient and we have good hope to keep the whole contour information contour in the cartoon part (Fig. 4).

We have tested many values for λ and the maximum number of iterations. We present some results to show the influence of λ (images have been contrasted). We shall speed up the method in the future using (for example) Nesterov algorithms as in [11] (Figs. 5 and 6).



(a) Original Image



(b) Image with threshold at grey value = 210

Fig. 4 3D angiography image

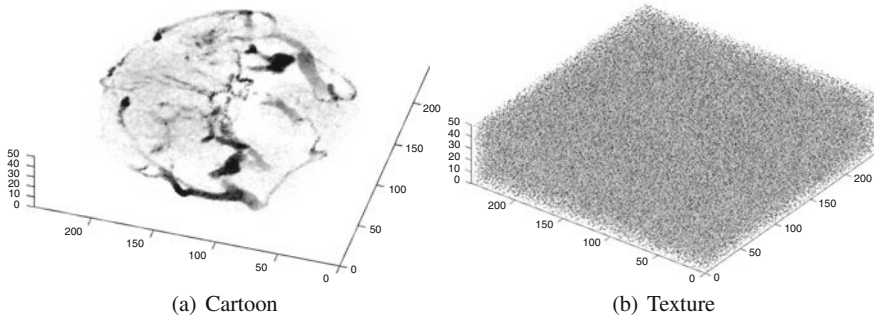


Fig. 5 No anisotropy strategy: $\lambda = 1$ and 5,000 iterations – The choice of small λ allows to denoising the image quite efficiently: here the texture is the noise and the cartoon the denoised image

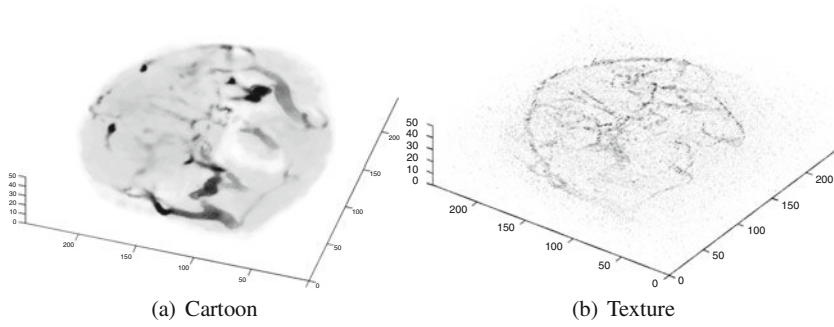


Fig. 6 No anisotropy strategy : $\lambda = 10$ and 5,000 iterations

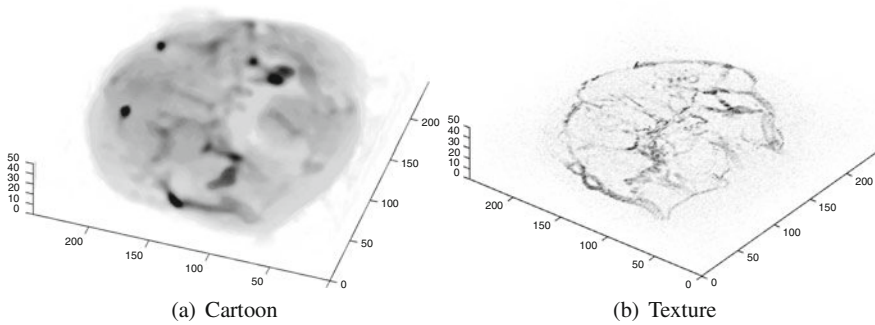


Fig. 7 No anisotropy strategy: $\lambda = 50$ and 10,000 iterations. The contours and the vessel network are recovered in the texture

We have tested the algorithm with and without anisotropy strategy. We give below results for $\lambda = 10$ and 5,000 iterations (Fig. 7). As the 3D cartoon and texture pictures are not easy to compare we give pictures of the difference as well (Figs. 8 and 9).

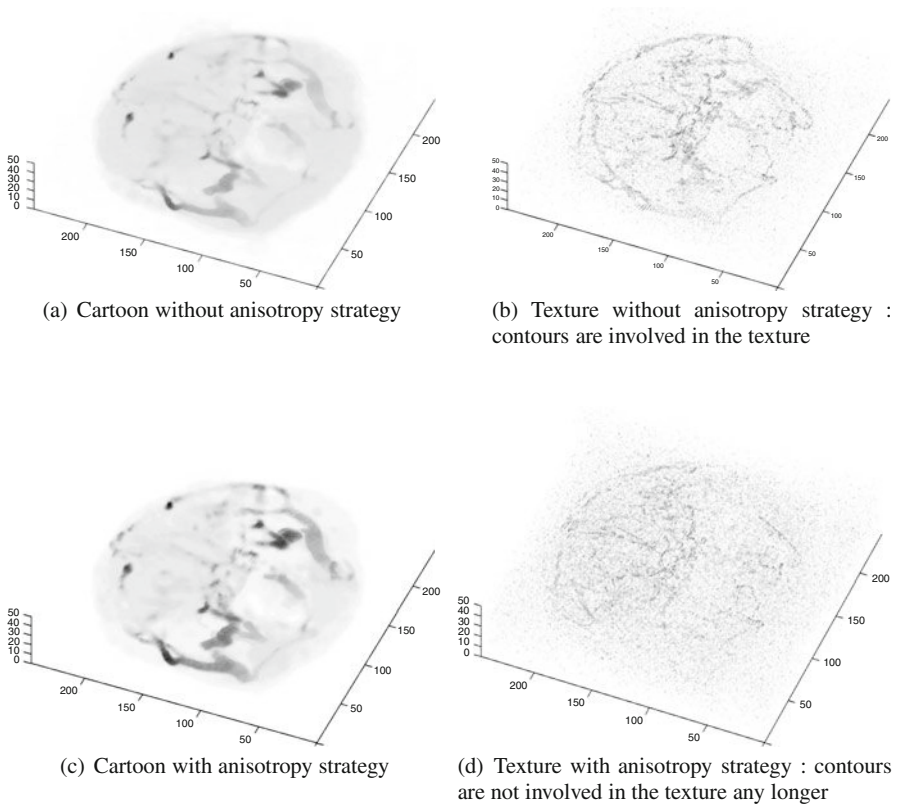
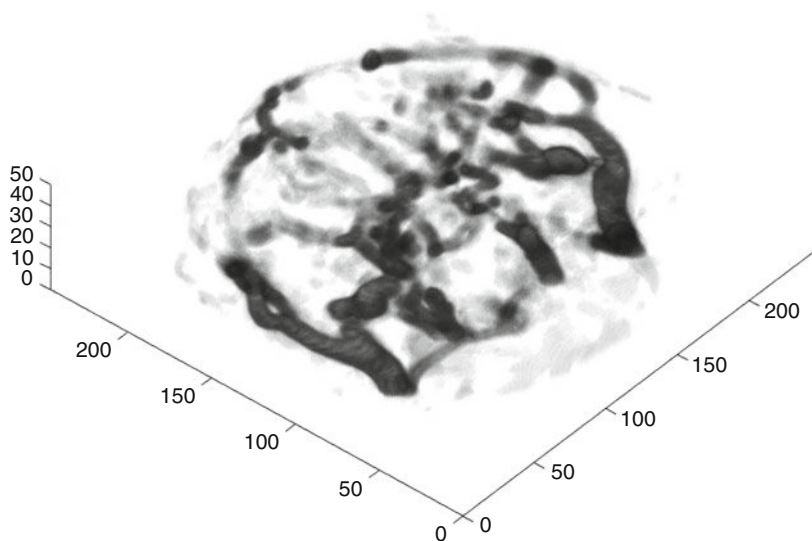
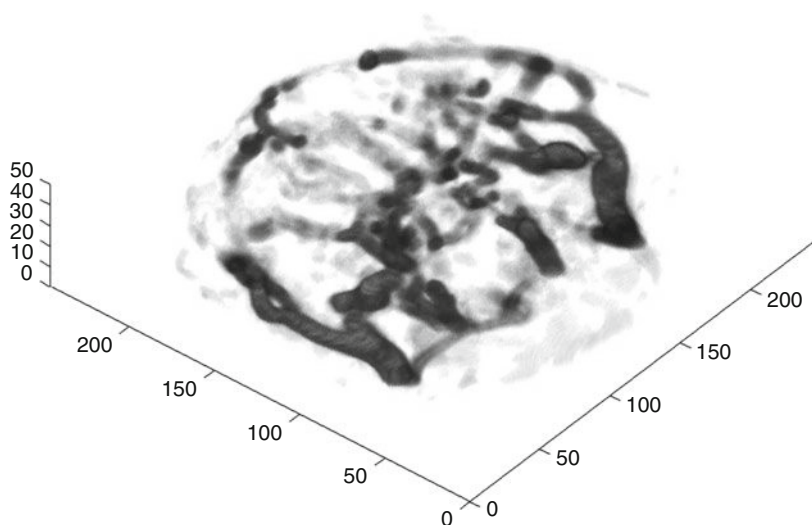


Fig. 8 Comparison between the two strategies for $\lambda = 10$ and 5,000 iterations



(a) Difference between cartoons



(b) Difference between textures

Fig. 9 Absolute value of the difference for $\lambda = 10$ and 5,000 iterations. The vessel-network which is alternatively included in the cartoon (when no anisotropy strategy is performed) or in the texture in the other case, so that the respective differences of cartoons and textures are the same and give the vessel network

Acknowledgements

We thank J.C. Belœil, S. Môme and F. Szeremeta, from CBM Laboratory in Orléans, for the use of these images, <http://cbm.cnrs-orleans.fr/>.

References

1. Ambrosio, L., Fusco, N., Pallara, D.: Functions of bounded variation and free discontinuity problems, Oxford mathematical monographs. Oxford University Press, Oxford (2000)
2. Attouch, H., Buttazzo, G., Michaille, G.: Variational analysis in Sobolev and BV spaces: applications to PDEs and optimization. MPS-SIAM series on optimization (2006)
3. Bergounioux, M., Piffet, L.: A second-order model for image denoising. Set Valued Anal. Variational Anal. **18**(3–4), 277–306 (2010)
4. Bergounioux, M.: On Poincaré-Wirtinger inequalities in spaces of functions of bounded variation, preprint (2010). <http://hal.archives-ouvertes.fr/hal-00515451/fr/>
5. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sci. **3** (2010), no. 3, 492–526.
6. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**, 89–97 (2004)
7. Demengel, F., Fonctions à hessien borné, Annales de l’institut Fourier **34**(2), 155–190 (1984)
8. Echegut, R., Piffet, L.: A variational model for image texture identification. In: Diehl, M., Glineur, F., Jarlebring, E., Michiels, W. (eds.) Recent Advances in Optimization and Its Applications in Engineering. Springer, Berlin (2010)
9. Piffet, L.: Modèles variationnels du second ordre pour l’extraction de textures 2D, Ph.D. Thesis, Orléans (2010)
10. Piffet, L.: A locally anisotropic model for image texture extraction. In: Bergounioux, M. (ed.) Mathematical Image Processing. Springer, Heidelberg (2011)
11. Weiss, P., Blanc-Fraud, L., Aubert, G.: Efficient schemes for total variation minimization under constraints in image processing. SIAM J. Sci. Comput. **31**(3), 2047–2080 (2009)

Analysis of Texture Anisotropy Based on Some Gaussian Fields with Spectral Density

Hermine Biermé and Frédéric J.P. Richard

Abstract In this paper, we describe a statistical framework for the analysis of anisotropy of image texture. This framework is based on the modeling of the image by two kinds of non-stationary anisotropic Gaussian field with stationary increments and spectral density: the extended fractional Brownian field (EFBF) and a specific Gaussian operator scaling field (GOSF), which both correspond to a generalization of the fractional Brownian field. In this framework, we tackle anisotropy analysis using some directional processes that are either defined as a restriction of the image on an oriented line or as a projection of the image along a direction. In the context of EFBF and GOSF, we specify links between the regularity of line and projection processes and model parameters, and explain how field anisotropy can be apprehended from the analysis of process regularity. Adapting generalized quadratic variations, we also define some estimators of the regularity of line and projection processes, and study their convergence to field model parameters. Estimators are also evaluated on simulated data, and applied for illustration to medical images of the breast and the bone.

1 Introduction

Texture analysis is an issue of particular topicality and practical importance in image processing [11, 23]. Texture, which can be roughly considered as a periodic aspect of images, has been modeled using various mathematical approaches ranging from statistical (e.g. Markov random fields [12]) to geometric (e.g. grouplets [24]). Model definitions have also varied depending on texture properties which are focused on. Among those properties, texture anisotropy is one of the most important. It is not

Hermine Biermé and Frédéric J.P. Richard (✉)
University Paris Descartes, Laboratory MAP5 (CNRS UMR 8145), 45 rue des Saints Pères,
75270 Paris Cedex 06, France
e-mail: hermine.bierme@parisdescartes.fr, frederic.richard@parisdescartes.fr

only common in natural images [24] but also critical for the analysis of images generated in a wide variety of domains, including biometry [16, 17] material science [14], hydro-geology [2], industry [13], or medical imaging [3, 10, 20, 27].

In this paper, we tackle texture analysis from a probabilistic point of view considering the image as a realization of a random field whose properties reflect those of the texture. From this point of view, we deal with the problem of estimating parameters of some anisotropic random field models defined below.

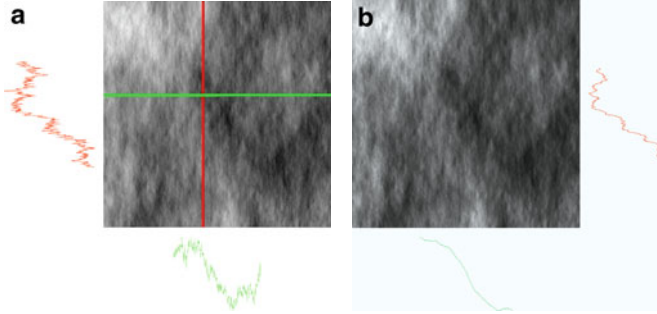


Fig. 1 Oriented analysis of a texture from (a) line processes and (b) projection processes

Texture anisotropy can be analyzed in a local way by computing oriented differential estimators at each image point [18, 26, 28]. Alternately, anisotropy can be apprehended from processes extracted from image [9, 13, 27], either *line-process* defined by restricting the image on oriented lines of the image domain, or *projection-process* obtained by projecting the image parallel to a given direction ; see the illustration on Fig. 1 and definitions in (5) and (7) of Sect. 2. From these processes, anisotropy can be analyzed by looking at regularity variations of processes when the extraction direction changes.

Here, our goal is to set up a statistical framework in which anisotropy can be consistently estimated using line and projection processes.

For defining this framework, we follow the set up of [9] describing d -dimensional Gaussian random fields (with stationary increments) for which the variogram v (see definition in (4) of Sect. 2) is of the form

$$\forall x \in \mathbb{R}^d, v(x) = \int_{\mathbb{R}^d} \left| e^{ix \cdot \zeta} - 1 \right|^2 f(\zeta) d\zeta, \quad (1)$$

where f is a positive even measurable function called spectral density and satisfying the condition $\int_{\mathbb{R}^d} (1 \wedge |\zeta|^2) f(\zeta) d\zeta < \infty$. In this set up, we focus on two models defined in dimension $d = 2$.

The first model, referred to as extended fractional Brownian field (EFBF), is defined with a spectral density

$$\forall \zeta \in \mathbb{R}^2, f(\zeta) = |\zeta|^{-2h(\arg(\zeta)) - 2}, \quad (2)$$

where $\arg(\zeta)$ is the direction of the frequency ζ and h is a measurable periodic function defined on the unit sphere \mathcal{S}^1 of \mathbb{R}^2 and ranging in $[H, M] \subset (0, 1)$ with $H = \operatorname{ess\,inf}_{\theta \in \mathcal{S}^1} h(\theta)$ and $M = \operatorname{ess\,sup}_{\theta \in \mathcal{S}^1} h(\theta)$. The fractional Brownian field, which is isotropic, is a particular case of EFBF obtained when function h is almost everywhere constant.

The second model, which can be seen as a Gaussian operator scaling field [2, 5] (GOSF), is defined with

$$\forall \zeta = (\zeta_1, \zeta_2) \in \mathbb{R}^2, f(\zeta) = (\zeta_1^2 + \zeta_2^{2a})^{-\beta}, \quad (3)$$

where $\beta = H_1 + (1 + 1/a)/2$ and $a = H_2/H_1$ for some $0 < H_1 \leq H_2 < 1$. The fractional Brownian field is a particular GOSF obtained when $H_1 = H_2$.

Since their spectral density depends on spectral direction, previous models are both anisotropic. Their anisotropy is characterized by model parameters, which are the directional Hurst index h for EFBF and constants H_1 and H_2 for GOSF. Hence, characterizing the anisotropy of these fields reduces to estimating their parameters. In the sequel, we specify links between the regularity of line and projection processes and model parameters, and explain how field anisotropy can be apprehended from the analysis of process regularity. Adapting generalized quadratic variations, we also define some estimators of the regularity of line and projection processes, and study their convergence to field model parameters.

2 Main Properties

2.1 Random Fields

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. A d -dimensional random field X is a map from $\Omega \times \mathbb{R}^d$ into \mathbb{R} such that $X(\cdot, y) := X(y)$ is a real random variable on Ω for all $y \in \mathbb{R}^d$. When $d = 1$, X is rather called a random process. A random field is Gaussian if any finite linear combination of its associated random variables is a Gaussian variable. A centered Gaussian field X is characterized by its covariance function: $(y, z) \mapsto \operatorname{Cov}(X(y), X(z))$. A field X has stationary increments if laws governing fields $X(\cdot + z) - X(z)$ and $X(\cdot) - X(0)$ are the same for all $z \in \mathbb{R}^d$. The law of a centered Gaussian field X with stationary increments is characterized by its variogram, defined by

$$\forall y \in \mathbb{R}^d, \gamma(y) = \mathbb{E}((X(y) - X(0))^2). \quad (4)$$

Gaussian fields with stationary increments and variogram of the form (1) will be referred to as Gaussian fields with spectral density. The EFBF and GOSF defined by (2) and (3) in introduction are two examples of these fields. From now on, we consider random fields for $d = 2$, corresponding to 2-dimensional images.

2.2 Line and Projection Processes

We now define line and projection processes which can be extracted from a two dimensional field X .

Definition 1. A line process $\mathcal{L}_{t_0, \theta} X$ is the restriction of a field X on a line identified by a point t_0 and a direction θ of \mathbb{R}^2 :

$$\forall t \in \mathbb{R}, \mathcal{L}_{t_0, \theta} X(t) = X(t_0 + t\theta). \quad (5)$$

We will specifically denote by $\mathcal{L}_\theta X$ the process on the line oriented in direction θ and passing through the origin.

Interestingly, if X is a Gaussian field with spectral density f , the process $\mathcal{L}_{t_0, \theta} X$ is also Gaussian with spectral density [3]

$$\forall v \in \mathbb{R}, f_{L, \theta}(v) = \int_{\mathbb{R}} f(\xi \theta^\perp + v\theta) d\xi. \quad (6)$$

Definition 2. A projection process $\mathcal{P}_\theta X$ of a field X is a windowed Radon transform of a field X in a direction θ of \mathbb{R}^2 . Given a window function ρ of the Schwartz class such that $\int_{\mathbb{R}} \rho(\gamma) d\gamma = 1$, it is defined as the projection of X along lines oriented in the direction θ^\perp through the window ρ

$$\forall t \in \mathbb{R}, \mathcal{P}_\theta X(t) = \int_{\mathbb{R}} X(s\theta^\perp + t\theta) \rho(s) ds, \quad (7)$$

If X is a Gaussian field with spectral density f , the projection process is also Gaussian with spectral density

$$\forall v \in \mathbb{R}, f_{P, \theta}(v) = \int_{\mathbb{R}} f(\xi \theta^\perp + v\theta) |\widehat{\rho}(\xi)|^2 d\xi, \quad (8)$$

where f is the spectral density of X .

2.3 Process Regularity

For the description of the regularity of a d -dimensional random field, we recall the usual definition of Hölder exponents. For $T > 0$, sample paths of X satisfy a uniform Hölder condition of order $\alpha \in (0, 1)$ on $[-T, T]^d$ if there exists a positive random variable A with $\mathbb{P}(A < +\infty) = 1$ such that a.s.

$$\forall y, z \in [-T, T]^d, |X(y) - X(z)| \leq A|y - z|^\alpha. \quad (9)$$

This equation gives a lower bound for the Hölder regularity of a field. The critical Hölder exponent β of a field (if it exists) is defined as the supremum of α for which

the Hölder condition (9) holds when it equals the infimum of α for which the Hölder condition (9) does not hold [9]. From an image point of view, the critical Hölder exponent of a field is related to the roughness of the texture: the rougher the texture is, the smaller the field regularity is.

For Gaussian fields with spectral density, the Hölder regularity can be characterized from the local behavior of the variogram around 0 or from the asymptotic behavior of the spectral density at high-frequencies [7, 9].

Proposition 1. *Let X be a d -dimensional Gaussian field with spectral density f and variogram v .*

(a) *Let $0 < \alpha \leq \gamma < 1$. If there exist $A, B_1, B_2 > 0$ and a positive-measure subset E of the unit sphere \mathcal{S}^{d-1} of \mathbb{R}^d such that for almost all $\xi \in \mathbb{R}^d$,*

$$\begin{aligned} (i) \quad & |\xi| \geq A \Rightarrow |f(\xi)| \leq B_1 |\xi|^{-2\alpha-d}; \\ (ii) \quad & |\xi| \geq A \text{ and } \frac{\xi}{|\xi|} \in E \Rightarrow |f(\xi)| \geq B_2 |\xi|^{-2\gamma-d}; \end{aligned}$$

then, there exist $\delta > 0$ and $C_1, C_2 > 0$ such that for all $y \in \mathbb{R}^d$,

$$(iii) \quad |y| \leq \delta \Rightarrow C_1 |y|^{2\gamma} \leq v(y) \leq C_2 |y|^{2\alpha}$$

(b) *If Condition (iii) holds for any α, γ with $0 < \alpha \leq \beta \leq \gamma < 1$ then β is the critical Hölder exponent of X .*

Let us notice that for random processes, we can use a definition of the Hölder regularity which holds for $\beta \geq 1$ [7, 9]. Similarly to Proposition 1, this extended Hölder regularity is also characterized by a behavior of the variogram around 0 or an asymptotic decay of the spectral density [7, 9].

We now state a theorem concerning the regularity of line processes associated to Gaussian fields with stationary increments.

Theorem 1. (a) *Let X be a 2-dimensional Gaussian field with stationary increments and assume that the line processes of X have Hölder exponents in all directions. Then, Hölder exponents of line processes are the same in all directions, except possibly in one direction where it is higher than in other directions.*

(b) *In the particular case of EFBF with spectral density defined as in (2), Hölder exponents of line processes are equal to $H = \text{essinf}_{\theta \in \mathcal{S}^1} h(\theta)$ whatever their direction.*

(c) *In the particular case of GOSF with spectral density defined as in (3), Hölder exponents of line processes are equal to H_1 in all direction except in direction $(0, 1)$ where it is equal to H_2 .*

Proof. Proofs of parts (a) of Theorem 1 can be found in [9, 13]. The part (b) and (c) were demonstrated in [3, 9] using Proposition 1. Essentially, in the case EFBF, the spectral density $f_{L,\theta}$ of a line process (see (5)) always satisfies conditions (i) and (ii) of Proposition 1 for $\beta = H = \text{essinf}_{\theta \in \mathcal{S}^1} h(\theta)$, whereas in the case of GOSF, it checks those conditions for $\beta = H_2$ in direction $(0, 1)$ and for $\beta = H_1$ in all other directions. \square

As stated by Theorem 1, the anisotropy of a Gaussian field with stationary increments has a specific display on the regularity of line processes. In some cases (e.g. EFBF), the anisotropy is simply not observable from line processes whose regularity does not vary according to direction. In other cases (e.g. GOSF), it is only observable in a single privileged direction where the regularity of line process is different from the others. In any cases, there can be at most one direction where the regularity of the line process varies.

Theorem 1 also implies that the anisotropy of EFBF and GOSF cannot be analyzed in the same way: whereas the line processes are useless to capture the anisotropy of EFBF, they can help enhancing a privileged direction of GOSF.

Next, we state a theorem about the regularity of the projection processes of EFBF.

Theorem 2. *Let X be an EFBF with spectral density defined as in (2). The Hölder regularity of the projected field $\mathcal{P}_\theta X$ is equal to $h(\theta) + 1/2$ for all directions θ of \mathbb{R}^2 .*

Proof. It was proved in [9] by checking asymptotic conditions on the spectral density of projection processes. \square

According to this theorem, the anisotropy of an EFBF is observable in projection processes since their regularity is directly related to the directional function h . Hence, projection processes are better suited than line processes for the characterization of EFBF anisotropy.

3 Parameter Estimation

In the previous section, we state several theorems showing that parameters of EFBF and GOSF are linked to the regularity of either line or projection processes extracted from the field. Therefore, the problem of estimating the parameters of these fields eventually reduces to the problem of estimating the regularity of several processes. Moreover, the line and projection processes of EFBF and GOSF are Gaussian with stationary increments and can be seen as generalizations of fractional Brownian motions (fBm). Hence, the regularity of these processes can be estimated using techniques developed for the estimation of the regularity parameter of a fBm (Hurst index).

There are numerous estimators of the Hurst index in the literature (see [1] for a survey). Here, we focus on generalized quadratic variations [7, 15, 19] and show how they can be adapted to the estimation of EFBF and GOSF parameters.

3.1 Generalized Quadratic Variations

Let Y be a Gaussian process with stationary increments and spectral density f , and $\{Y(\frac{k}{N}); 0 \leq k \leq N\}$ be an observed sequence of Y . Let us consider the stationary

sequence formed by second-order increments of Y with step $u \in \mathbb{N} \setminus \{0\}$

$$\forall p \in \mathbb{Z}, Z_{N,u}(Y)(p) = Y\left(\frac{p+2u}{N}\right) - 2Y\left(\frac{p+u}{N}\right) + Y\left(\frac{p}{N}\right). \quad (10)$$

Generalized quadratic variations of Y of order 2 are then given by

$$V_{N,u}(Y) = \frac{1}{N-2u+1} \sum_{p=0}^{N-2u} (Z_{N,u}(Y)(p))^2. \quad (11)$$

By stationarity of increments, we have $\mathbb{E}(V_{N,u}(Y)) = \mathbb{E}((Z_{N,u}(Y)(0))^2)$. Besides, according to [7], as N tends to $+\infty$,

$$\mathbb{E}(V_{N,u}(Y)) \sim c_H N^{-2H} u^{2H},$$

for some $c_H > 0$, whenever the spectral density f satisfies $f(\xi) \sim_{|\xi| \rightarrow +\infty} c |\xi|^{-2H-1}$, with $H \in (0, 2)$ and $c > 0$. Thus, we can intuitively define an estimator of H as

$$\hat{H}_N = \frac{1}{2 \log(2)} \log \left(\frac{V_{N,2}(Y)}{V_{N,1}(Y)} \right). \quad (12)$$

In [7, 15], the convergence of this estimator to H with asymptotic normality was demonstrated for $H \in (0, \frac{7}{4})$ under some appropriate assumptions on either the variogram of Y or on its spectral density.

3.2 Convergence Results

For the estimation of parameters of EFBF or GOSF, we apply generalized quadratic variations to processes extracted from the field and define estimators by comparing variations at different scales as in (12). In this way, we set line-based estimators $h_N^L(\theta)$ and projection-based estimators $h_N^P(\theta)$

$$\forall \theta, h_N^L(\theta) = \frac{1}{2 \log(2)} \log \left(\frac{V_{N,2}(\mathcal{L}_\theta X)}{V_{N,1}(\mathcal{L}_\theta X)} \right) \quad (13)$$

$$\forall \theta, h_N^P(\theta) = \frac{1}{2 \log(2)} \log \left(\frac{V_{N,2}(\mathcal{P}_\theta X)}{V_{N,1}(\mathcal{P}_\theta X)} \right), \quad (14)$$

where $\mathcal{L}_\theta X$, $\mathcal{P}_\theta X$, and $V_{N,u}(Y)$ are defined by (5), (7) and (11), respectively. In what follows, we state several theorems concerning the convergence of these estimators.

Let us start with the estimation of parameters of an EFBF.

Theorem 3. *Let X be an EFBF with spectral density defined as in (2), and θ be a direction in the unit sphere \mathcal{S}^1 . Assume that h is Lipschitz of order $\alpha \in (1/2, 1]$ on \mathcal{S}^1 . Then, almost surely,*

$$\lim_{N \rightarrow +\infty} \left(\hat{h}_N^P(\theta) - \frac{1}{2} \right) = h(\theta), \quad (15)$$

and, in law,

$$\lim_{N \rightarrow +\infty} \sqrt{N} \left(\hat{h}_N^P(\theta) - \frac{1}{2} - h(\theta) \right) = \mathcal{N}(0, K), \quad (16)$$

for some positive constant K .

Proof. According to Proposition 2.1 of [7], the spectral density of the projection process $\mathcal{P}_\theta X$ satisfies assumptions of Theorem 4.1 of [4] and therefore Sect. 4.3 of [4] gives the result. \square

This theorem ensures the consistency of \hat{h}_N^P as an estimator of the directional function h of an EBBF, together with its asymptotic normality. Theorem 3 only concerns the convergence of a projection-based estimator obtained in a single direction. In [27], we showed another result on the convergence of combinations of estimators in different directions.

Theorem 4. *Let X be an EBBF with spectral density defined as in (2). Let $\theta_0 = (0, 1)$ and $\theta_0^\perp = (1, 0)$. Assume that h is Lipschitz of order $\alpha \in (1/2, 1]$ on \mathcal{S}^1 . Then, almost surely,*

$$\lim_{N \rightarrow +\infty} \left(\hat{h}_N^P(\theta_0) - \hat{h}_N^P(\theta_0^\perp) \right) = h(\theta_0) - h(\theta_0^\perp), \quad (17)$$

and in law,

$$\lim_{N \rightarrow +\infty} \sqrt{N} \left(\hat{h}_N^P(\theta_0) - \hat{h}_N^P(\theta_0^\perp) - (h(\theta_0) - h(\theta_0^\perp)) \right) = \mathcal{N}(0, K), \quad (18)$$

for some positive constant K .

Proof. This result follows from Theorem 3.1 of [27] under the weaker assumption that h is Lipschitz of order $\alpha \in (1/2, 1]$ on \mathcal{S}^1 and not continuously differentiable. The proof may be adapted using Theorem 3 since it only takes into account correlations between combined estimators, which remain unchanged under this assumption. \square

This theorem is useful to construct hypothesis tests of anisotropy [27]. In these tests, the hypothesis of field isotropy will be rejected if the statistic

$$\delta_1 = |\hat{h}_N^P(\theta_0) - \hat{h}_N^P(\theta_0^\perp)| \quad (19)$$

is above a bound (see Sect. 4). This bound can be computed for any level of confidence using simulated data.

We have also defined another anisotropy test which is based on the statistic

$$\delta_2 = |\max(\hat{h}_N^P(\theta_0), \hat{h}_N^P(\theta_0^\perp)) - \min(\hat{h}_N^L(\theta_0), \hat{h}_N^L(\theta_0^\perp))|. \quad (20)$$

For this test, the isotropy assumption is rejected if δ_2 is above a bound.

Now, let us turn to the estimation of parameters of a GOSF.

Theorem 5. *Let X be a GOSF with spectral density defined as in (3). Let $\theta = (\theta_1, \theta_2)$ be a direction in the unit sphere \mathcal{S}^1 .*

(a) If $\theta_1 = 0$ (privileged direction), then almost surely

$$\lim_{N \rightarrow +\infty} \hat{h}_N^L(\theta) = H_2. \quad (21)$$

otherwise, almost surely,

$$\lim_{N \rightarrow +\infty} \hat{h}_N^L(\theta) = H_1. \quad (22)$$

(b) The convergence in law

$$\lim_{N \rightarrow +\infty} \sqrt{N} (\hat{h}_N^L(\theta) - H) = \mathcal{N}(0, K), \quad (23)$$

occurs when $\theta_1 = 0$ with $H = H_2$, when $\theta_2 = 0$ with $H = H_1$, and when $\theta_1 \neq 0$, $\theta_2 \neq 0$ and $a = H_2/H_1 > 2$ with $H = H_1$.

(c) When $a \leq 2$ and $\theta_1 \neq 0$, we only have

$$\mathbb{E} \left((\hat{h}_N^L(\theta) - H_1)^2 \right) = O_{N \rightarrow +\infty} \left(N^{-2(1-1/a)} \right). \quad (24)$$

Proof. See [3]. \square

This theorem ensures the consistency of \hat{h}_N^L as an estimator of the directional constants H_1 and H_2 of an GOSF. In this case, the asymptotic normality of estimators is guaranteed in all directions provided that the rate H_2/H_1 is large enough, i.e. the anisotropy is marked enough.

4 Numerical Study

The numerical study of estimators is usually done on a set of synthetic realizations of the random field with known parameter values. Therefore, it requires a good (ideally exact) method for the simulation of field realizations. But, the exact simulation of EFBF or GOSF is an open issue we are still working on. Hence, the numerical evaluation of estimators are not completely feasible.

To our knowledge, simulations methods for Gaussian fields with spectral density come down to Fourier methods which generates approximate realizations with properties that may differ from the theoretical ones [7]. There is however an exact method which was proposed by Stein [29] for the simulation of (isotropic) fractional Brownian fields. But it can be hardly extended to the simulation of anisotropic fields such as EFBF or GOSF.

Some simulation examples shown on Fig. 2 illustrate some visual features of different fields under studies. Realizations of fractional Brownian fields were obtained using the Stein's method. EFBF were simulated using a turning-band method adapted from [22], which is still under investigation [6]. In these simulations, the

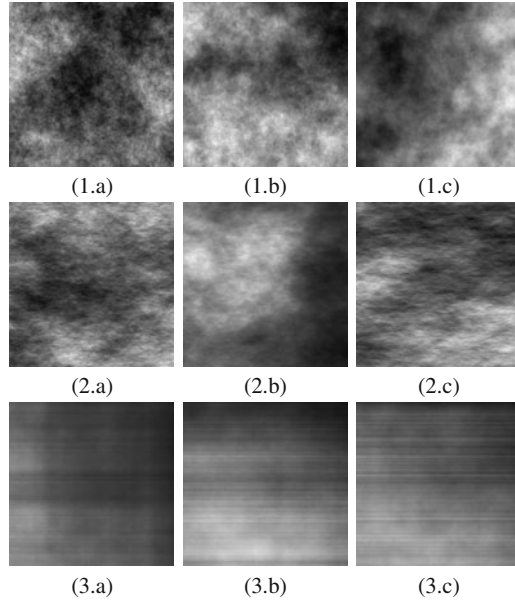


Fig. 2 Exact simulations of fractional Brownian field of increasing Hurst index H (Stein's method): **(1.a)** $H = 0.3$, **(1.b)** $H = 0.5$, **(1.c)** $H = 0.7$. Approximate simulations of extended fractional Brownian field (Turning-Band method): **(2.a)** $(H_1, H_2) = (0.2, 0.5)$, **(2.b)** $(H_1, H_2) = (0.5, 0.7)$, **(2.c)** $(H_1, H_2) = (0.2, 0.7)$. Ad hoc simulations of Gaussian operator scaling fields: **(3.a)** $(H_1, H_2) = (0.7, 0.8)$, **(3.b)** $(H_1, H_2) = (0.5, 0.8)$, **(3.c)** $(H_1, H_2) = (0.3, 0.8)$

spectral density was specified with

$$\forall \theta = (\theta_1, \theta_2) \in \mathcal{S}^1, h(\theta) = |\theta_1|H_1 + (1 - |\theta_1|)H_2, \quad (25)$$

for some selected pairs (H_1, H_2) in $(0, 1)^2$. The third series of fields are not precisely GOSF. There are just simulations of fields having a variogram of the form $v(x) = c_1|x_1|^{H_1} + c_2|x_2|^{H_2}$ and showing a privileged direction.

Our early numerical studies were based on Fourier simulations of EFBF [7, 27]. Here we give some preliminary results based on a turning-band simulations. We simulated sets of EFBF of size (512×512) with spectral density of the form (25), applied line- and projection-based estimators in two directions (horizontal and vertical) to simulations, and then computed biases and standard deviations of estimators. Results are reported in Table 1.

Applied to these simulations, both line-based estimators approximate the regularity $H = \min(H_1, H_2)$, whereas projection-based estimators give an approximation of H_1 or H_2 , depending of their direction. Standard deviations of line-based estimators are very small (below 10^{-2}) compared to those of projection-based estimators

Table 1 Evaluation of line- and projection-based estimators on sets of 100 extended fractional Brownian fields simulations (Turning-Band method) with different Hurst indices (bias \pm std)

H_1	H_2	$\hat{h}_L^N(\theta_0)$	$\hat{h}_L^N(\theta_0^\perp)$	$\hat{h}_P^N(\theta_0)$	$\hat{h}_P^N(\theta_0^\perp)$
0.2	0.2	0.003 \pm 0.003	0.003 \pm 0.003	-0.08 \pm 0.13	-0.05 \pm 0.11
0.5	0.5	0.003 \pm 0.003	0.003 \pm 0.003	-0.01 \pm 0.15	-0.04 \pm 0.14
0.7	0.7	0.003 \pm 0.003	0.003 \pm 0.003	-0.02 \pm 0.13	-0.04 \pm 0.13
0.2	0.5	0.053 \pm 0.003	0.083 \pm 0.003	-0.02 \pm 0.16	-0.08 \pm 0.13
0.2	0.7	0.060 \pm 0.005	0.088 \pm 0.003	-0.06 \pm 0.17	-0.29 \pm 0.15
0.5	0.7	0.044 \pm 0.004	0.096 \pm 0.003	-0.03 \pm 0.13	-0.04 \pm 0.12

(above 10^{-1}). This is due to a sub-sampling procedure we apply to projections so as to reduce the bias of projection-based estimators [7]. Biases of line-based estimators are also very small, especially when simulated fields are isotropic ($H_1 = H_2$). However, as the field anisotropy increases, H tends to be overestimated. This might be due either to a simulation artifact or a property of the estimator. Biases of projection-based estimators are reasonably small; most of them are below 10^{-1} . When the anisotropy is large ($H_1 = 0.2$ and $H_2 = 0.7$), the highest directional regularity is however underestimated.

Table 2 Evaluation anisotropy tests on sets of 100 extended fractional Brownian fields simulations (Turning-Band method) with different Hurst indices (bias \pm std ; p percent of realizations considered isotropic)

		First test		Second test	
$h(\theta_0)$	$h(\theta_0^\perp)$	δ_1	p	δ_2	p
0.2	0.2	0.01 \pm 0.16	94	0.08 \pm 0.06	93
0.5	0.5	0.01 \pm 0.14	98	0.07 \pm 0.05	97
0.7	0.7	0. \pm 0.13	99	0.07 \pm 0.05	99
0.2	0.5	0. \pm 0.10	53	-0.08 \pm 0.06	39
0.2	0.7	-0.1 \pm 0.10	36	-0.28 \pm 0.06	32
0.5	0.7	-0.1 \pm 0.06	100	-0.12 \pm 0.04	100

In Table 2, we present an evaluation of isotropy tests defined with statistics δ_1 and δ_2 given by (19) and (20). Observing biases and variances of δ_1 and δ_2 on isotropic simulations, we set rejection bounds for isotropy tests so as to get a level of confidence of 95%. We applied isotropy tests to all simulations and compute the percentage of images considered isotropic in each simulation sets (value p). On isotropic simulations, we observe that tests effectively detect the isotropic cases with a high level of confidence. On anisotropic cases, results show that tests can detect anisotropy only if the anisotropy is marked ($|h(\theta_0) - h(\theta_0^\perp)| > 0.2$). In simulation cases when anisotropy is the largest ($h(\theta_0) = 0.2$ and $h(\theta_0^\perp) = 0.7$), the risk of second species (i.e. the probability of considering that an anisotropic realization is isotropic) is still high for both tests (around 35%). The lack of sensitivity of tests

to anisotropy is mainly due to the large variance of projection-based estimators. We are currently working on the improvement of estimator precision.

5 Applications

We present two applications of the statistical framework described above. In these applications, images are either breast radiographs (mammograms) or bone radiographs, both being used by physicians in clinical routine [21, 27]; see Fig. 3 for illustration.

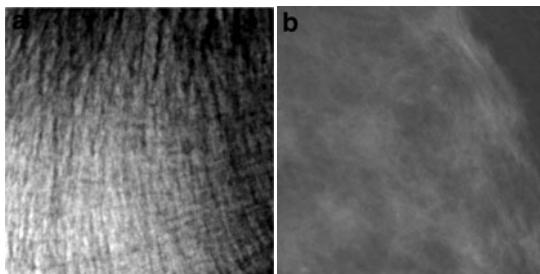


Fig. 3 Regions extracted from (a) a radiography of the calcaneum (a heel bone) and (b) a mammogram (breast radiography)

In [3, 27], we could show on some database that both images are anisotropic. Assuming mammograms to be realizations of EFBF and applying anisotropy tests on them, we obtained that 41% of mammograms could be considered as anisotropic with a level of confidence of 95% [27]. This anisotropy can be seen both, on Fig. 4b, where horizontal and vertical projection-based estimates are plotted for each mammograms, and on Fig. 4c, where the maximum of horizontal and vertical projection-based estimates is plotted against the minimum of horizontal and vertical line-based estimates.

The anisotropy of bone radiographs can be observed on Fig. 5, where couples of line-based estimates in different directions (horizontal, vertical, or diagonals) are plotted for each images. It can be seen that in the vertical direction θ_0^\perp , the regularity tends to be higher than in the three other directions; this direction corresponds to the axis of longitudinal trabeculae of the bone. Hence, according to Theorem 1, GOSF would be better suited than EFBF for the modeling of bone radiographs.

The anisotropy of mammograms is different from the one of bone radiographs. According to Figs. 4a, b, it seems not to be detected on lines. Contrarily to bone radiographs, mammograms would be better described by EFBF.

Previous experiments are preliminary steps of a multidisciplinary project which concern the evaluation of (a) breast cancer risk and (b) bone fracture risk due to

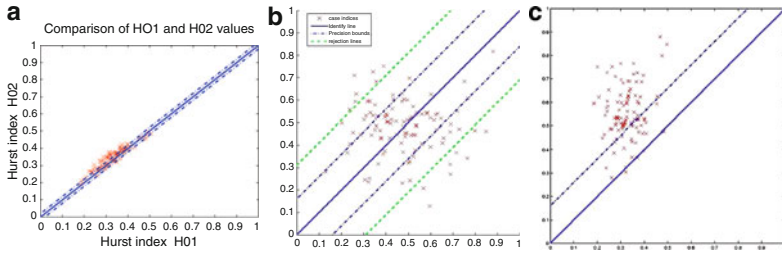


Fig. 4 Line- and projection-based estimates obtained on 116 full-field digital mammograms (private database, courtesy of P. Bakic and A. Maidment from Univ. of Pennsylvania): comparisons of (a) $h_N^L(\theta_0)$ (abscissa) and $h_N^L(\theta_1)$, (b) $h_N^P(\theta_0)$ and $h_N^P(\theta_1)$, and (c) $\min(h_N^L(\theta_0), h_N^L(\theta_1))$ and $\max(h_N^P(\theta_0), h_N^P(\theta_1))$. (directions: $\theta_0 = (1, 0)$ (horizontal), $\theta_0^\perp = (0, 1)$ (vertical))

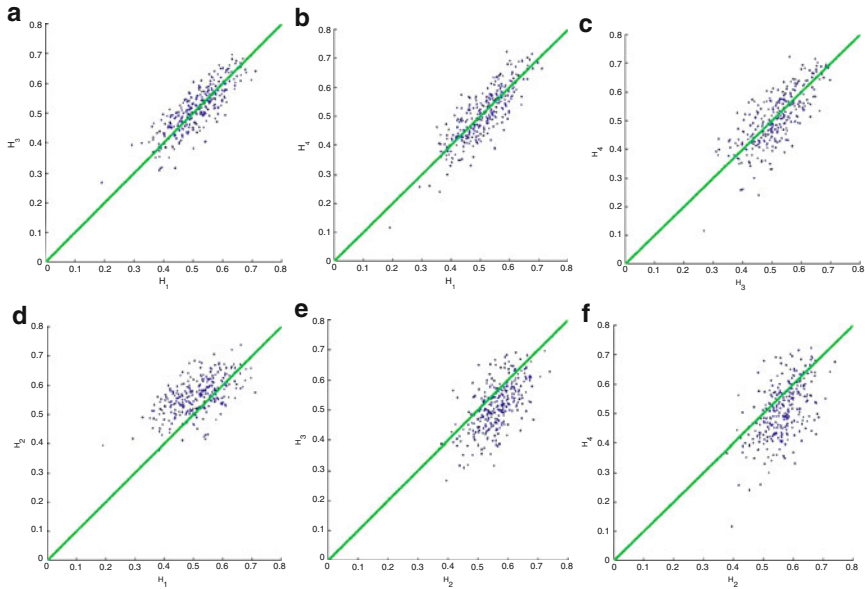


Fig. 5 Line-based estimates obtained on 211 calcaneum radiographs (database of Inserm unit U658 [21]): comparisons of (a) $h_N^L(\theta_0)$ (abscissa) and $h_N^L(\theta_1)$ (ordinate), (b) $h_N^L(\theta_0)$ and $h_N^L(\theta_1^\perp)$, (c) $h_N^L(\theta_1)$ and $h_N^L(\theta_1^\perp)$, (d) $h_N^L(\theta_0)$ and $h_N^L(\theta_0^\perp)$, (e) $h_N^L(\theta_0^\perp)$ and $h_N^L(\theta_1)$, and (f) $h_N^L(\theta_0^\perp)$ and $h_N^L(\theta_1^\perp)$. (directions: $\theta_0 = (1, 0)$ (horizontal), $\theta_0^\perp = (0, 1)$ (vertical), 3: $\theta_1 = (1, 1)/\sqrt{2}$ (diagonal), 4: $\theta_1^\perp = (-1, 1)/\sqrt{2}$ (diagonal))

osteoporosis [8, 25, 27]. In the context of this project, the line and projection estimates are used as descriptors of image aspect. By combining such descriptors to known clinical risk factors, we aim at improving the evaluation of both risks.

Acknowledgements

This work is part of the research program MATAIM, supported by the Agence Nationale pour la Recherche (ANR-09-BLAN-0029-01) and the Institut National du Cancer (INCA, 2009-1-SHS SP-01-UP5-1).

References

1. Bardet, J.M., Lang, G., Oppenheim, G., et al.: Semi-parametric estimation of the long-range dependence parameter: A survey. In: *Theory and Applications of Long-Range Dependence*, pp. 557–577. Birkhauser, Boston (2003)
2. Benson, D., Meerschaert, M.M., Bäumer, B., Scheffler, H.P.: Aquifer operator-scaling and the effect on solute mixing and dispersion. *Water Resour. Res.* **42**, 1–18 (2006)
3. Biermé, H., Benhamou, C.L., Richard, F.: Parametric estimation for gaussian operator scaling random fields and anisotropy analysis of bone radiograph textures. In Pohl, K. (ed.) *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'09), Workshop on Probabilistic Models for Medical Imaging*, pp. 13–24. London, UK, Sept 2009
4. Biermé, H., Bonami, A., León, J.R.: Central Limit Theorems and Quadratic Variation in terms of Spectral Density, *Electron. J. Probab.*, **16**, 362–395, (2011)
5. Biermé, H., Meerschaert, M.M., Scheffler, H.P.: Operator scaling stable random fields. *Stoch. Proc. Appl.* **117**(3), 312–332 (2007)
6. Biermé, H., Moisan, L., Richard, F.: A turning-band method for the simulation of anisotropic fractional brownian fields. in preparation (2010)
7. Biermé, H., Richard, F.: Estimation of anisotropic gaussian fields through radon transform. *ESAIM P S* **12**(1), 30–50 (2008)
8. Biermé, H., Richard, F., Rachidi, M., Benhamou, C.L.: Anisotropic texture modeling and applications to medical image analysis. In: *ESAIM Proceedings: Mathematical Methods for Imaging and Inverse Problems*, pp. 100–122. 2009
9. Bonami, A., Estrade, A.: Anisotropic analysis of some Gaussian models. *J. Fourier Anal. Appl.* **9**, 215–236 (2003)
10. Brunet-Imbault, B., Lemineur, G., Chappard, C., et al.: A new anisotropy index on trabecular bone radiographic images using the fast Fourier transform. *BMC Med. Imaging* **5**(4) (2005)
11. Chantler, M.J., Van Gool, L.J.: Editorial: Special issue on texture analysis and synthesis. *Int. J. Comput. Vis.* **62**(1–2), 5–5 (2005)
12. Cross, G., Jain, A.: Markov random field texture models. *IEEE Trans. PAMI* **5**(1), 25–39 (1983)
13. Davies, S., Hall, P.: Fractal analysis of surface roughness by using spatial data. *J. R. Stat. Soc. Ser. B* **61**, 3–37 (1999)
14. Germain, C., Da Costa, J.P., Lavialle, O., Baylou, P.: Multiscale estimation of vector field anisotropy application to texture characterization. *Signal Process.* **83**(7), 1487–1503 (2003)
15. Istas, J., Lang, G.: Quadratic variations and estimation of the local Holder index of a Gaussian process. *Ann. Inst. Henri Poincaré Prob. Stat.* **33**(4), 407–436 (1997)
16. Jain, A., Hong, L., Bolle, R.: On-line fingerprint verification. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 302–314 (1997)
17. Jiang, X.D.: On orientation and anisotropy estimation for online fingerprint authentication. *IEEE Trans. Signal. Process.* **53**(10), 4038–4049 (2005)
18. Kass, M., Witkin, A.: Analyzing oriented patterns. *Comput. Vis. Graph. Image Process.* **37**(362–385) (1987)
19. Kent, J.T., Wood, A.T.A.: Estimating the fractal dimension of a locally self-similar Gaussian process by using increments. *J. R. Statist. Soc. Ser. B* **59**(3), 679–699 (1997)

20. Lemineur, G., Harba, R., Jennane, R., et al.: Fractal anisotropy measurement of bone texture radiographs. In: First International Symposium on Control, Communications and Signal Processing, pp. 275–278. (2004)
21. Lespessailles, E., Gadois, C., Kousignian, I., Neveu, J.P., Fardellone, P., Kolta, S., Roux, C., Do-Huu, J.P., Benhamou, C.L.: Clinical interest of bone texture analysis in osteoporosis: A case control multicenter study. *Osteoporos. Int.* **19**, 1019–1028 (2008)
22. Matheron, G.: The intrinsic random functions and their applications. *Adv. Appl. Probab.* **5**(3), 439–468 (1973)
23. Nielsen, M., Hansen, L.K., Johansen, P., Sparring, J.: Guest editorial: Special issue on statistics of shapes and textures. *J. Math. Imaging Vis.* **17**(2), 87–87 (2002)
24. Peyré, G.: Texture synthesis with grouplets. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4), 733–746 (2010)
25. Rachidi, M., Richard, F., Biermé, H., Lespessailles, E., Chappard, C., Benhamou, C.L.: Conception of a composite osteoporosis fracture index including bone texture analysis and bone mineral density. In: Proceedings of the 18th International Bone Densitometry Workshop IBDW'08, Foggia, Italy, June 2008
26. Rao, A., Jain, R.: Computerized flow field analysis: Oriented texture fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 693–709 (1992)
27. Richard, F.J.P., Biermé, H.: Statistical tests of anisotropy for fractional brownian textures. application to full-field digital mammography. *J. Math. Imaging Vis.* **36**(3), 227–240 (2010)
28. Shu, C., Jain, R.: Vector field analysis for oriented patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 946–950 (1994)
29. Stein, M.L.: Fast and exact simulation of fractional Brownian surfaces. *J. Comput. Graph. Statist.* **11**(3), 587–599 (2002)

Image Reconstruction Via Hypoelliptic Diffusion on the Bundle of Directions of the Plane

Ugo Boscain, Jean Duplaix, Jean-Paul Gauthier, and Francesco Rossi

Abstract In this paper we present a model of geometry of vision which generalizes one due to Petitot, Citti and Sarti. One of its main features is that the primary visual cortex V1 lifts the image from \mathbb{R}^2 to the bundle of directions of the plane $PT\mathbb{R}^2 = \mathbb{R}^2 \times P^1$. Neurons are grouped into *orientation columns*, each of them corresponding to a point of the bundle $PT\mathbb{R}^2$.

In this model a corrupted image is reconstructed by minimizing the energy necessary for the activation of the orientation columns corresponding to regions in which the image is corrupted. The minimization process gives rise to an hypoelliptic heat equation on $PT\mathbb{R}^2$. The hypoelliptic heat equation is studied using the generalized Fourier transform. It transforms the hypoelliptic equation into a 1-d heat equation with Mathieu potential, which one can solve numerically.

Preliminary examples of image reconstruction are hereby provided.

1 Introduction

In this paper we provide a model of geometry of vision which generalizes one initially due to Petitot (see a complete description in [19]), then refined by Citti and Sarti [7], and in [5].

Ugo Boscain
CMAP, École Polytechnique CNRS, Palaiseau, France
e-mail: boscain@cmap.polytechnique.fr

Jean Duplaix and Jean-Paul Gauthier
Laboratoire LSIS, Université de Toulon, France
e-mail: duplaix@univ-tln.fr, gauthier@univ-tln.fr

Francesco Rossi (✉)
Laboratoire LSIS, Université Paul Cézanne, Marseille, France
e-mail: francesco.rossi@lsis.org

To start with, assume that the grey level of an image is represented by a function $\mathcal{J} \in L^2(\mathcal{D}, \mathbb{R})$, where \mathcal{D} is an open bounded domain of \mathbb{R}^2 . The algorithm that we present here is based on three crucial ideas coming from neurophysiology:

1. It is widely accepted that the retina approximately smoothes the images by making the convolution with a Gaussian function $G(\sigma_x, \sigma_y)$ (see for instance [12, 13, 18] and references therein). Then the smoothed image

$$f = \mathcal{J} * G(\sigma_x, \sigma_y) \in L^2(\mathbb{R}^2, \mathbb{R}) \cap d\mu_{\#}^{\infty}(\mathbb{R}^2, \mathbb{R}),$$

is generically a Morse function (i.e. a smooth function having as critical points only maxima, minima and saddles). This fact has interesting consequences, as explained in the following.

2. **The primary visual cortex V1 lifts the image from \mathbb{R}^2 to the bundle of directions of the plane $PT\mathbb{R}^2$.**

In a simplified model¹ (see [19, p. 79]), neurons of V1 are grouped into *orientation columns*, each of them being sensitive to visual stimuli at a given point a of the retina and for a given direction p on it. The retina is modeled by the real plane, i.e. each point is represented by $a \in \mathbb{R}^2$, while the directions at a given point are modeled by the projective line, i.e. $p \in P^1$. Hence, the primary visual cortex V1 is modeled by the so called *projective tangent bundle* $PT\mathbb{R}^2 := \mathbb{R}^2 \times P^1$. See Fig. 1.

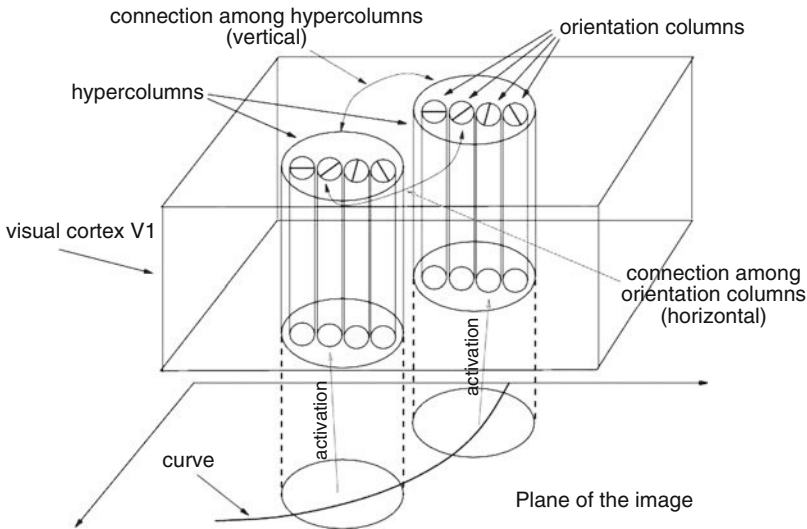


Fig. 1 A scheme of the primary visual cortex V1

¹ For example, in this model we do not take into account the fact that the continuous space of stimuli is implemented via a discrete set of neurons.

The smoothed image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is *lifted* to a function \tilde{f} defined as follows: $\tilde{f}(x, y, \theta) = f(x, y)$ if θ is the direction of the level set of f passing through (x, y) ; otherwise $\tilde{f}(x, y, \theta) = 0$.

3. If the image is corrupted or missing on a set $\Omega \subset \mathbb{R}^2$ (i.e. if f is defined on $\mathbb{R}^2 \setminus \Omega$), then **the reconstruction of f in Ω is made by minimizing a cost representing the energy** that the visual cortex should spend to excite orientation columns corresponding to points in Ω and hence that are not directly excited by the image. An orientation column is easily excited if it is close to another (already activated) orientation column sensitive to a similar direction in a close position (i.e. if the two are close in $PT\mathbb{R}^2$).

When the image to be reconstructed is a curve, this gives rise to a sub-Riemannian problem (i.e. an optimal control problem linear in the control and with quadratic cost) on $PT\mathbb{R}^2$.

When the image is not just a curve, the reconstruction is made by considering the diffusion process naturally associated to the sub-Riemannian problem on $PT\mathbb{R}^2$ (described by an hypoelliptic heat equation).

In this paper, we present the model described above with all mathematical details and apply it to simple problems of image reconstruction. We recall that the model by Petitot–Citti–Sarti has been already applied to different problems in imaging, like contour perception [7] and contour enhancement [9], image analysis and reconstruction [10]. The main differences and improvements are described in the following. Let us first recall that $PT\mathbb{R}^2$ can be seen as the quotient of the group of rototranslations of the plane $SE(2) \simeq \mathbb{R}^2 \times S^1$, with respect to \mathbb{Z}_2 , where the quotient is the identification of antipodal points in S^1 .

- We start by any function $\mathcal{J} : L^2(\mathcal{D}, \mathbb{R})$. After convolution with a Gaussian, generically, we are left with a Morse function.
- We lift the image on $PT\mathbb{R}^2$ and not on $SE(2)$. This improvement is crucial for the following reasons. First, on $PT\mathbb{R}^2$ the lift is unique, while on $SE(2)$ it is not, since level sets of the image are not oriented curves. Second, the domain S_f of the lift of a Morse function f is a manifold when the lift is made on $PT\mathbb{R}^2$, while it is a manifold with boundary if it is made on $SE(2)$ (for a continuous choice of the orientation of the level sets of f). The boundary appears on minima, maxima and saddles of f . Third, the problem of reconstruction of a curve is not well defined on $SE(2)$, see [5].
- The sub-Riemannian diffusion is not computed numerically starting from the hypoelliptic heat equation, but giving an approximation to the generalized (non-commutative) Fourier transform of this equation which is the Mathieu equation.

Our algorithm can be implemented numerically (some preliminary results are presented in Sect. 4) and the results can be compared with the ones coming from psychological experiments. Moreover, it can be useful to reconstruct the geometry of an image, as a preliminary step of exemplar-based methods (see [6]).

Our model, generalizing the approach by Petitot–Citti–Sarti, is based on hypoelliptic diffusion. Another approach has been proposed by Mumford and others, based on hypoelliptic direction process. It is used for several problems in imaging, see e.g. contour enhancement [17].

The structure of the paper is the following. In Sect. 2 we present in detail the sub-Riemannian structure defined on $PT\mathbb{R}^2$, according to the model by Petitot–Citti–Sarti. We then define the corresponding hypoelliptic diffusion and we find explicitly the corresponding kernel on $PT\mathbb{R}^2$. At the end, we present in detail the mathematical algorithm. Section 3 is devoted to various improvements of the mathematical algorithm to code it numerically. Finally, in Sect. 4 we provide preliminary examples of image reconstruction.

2 The Mathematical Model and the Algorithm

2.1 Reconstruction of a Curve

In this section we briefly describe an algorithm to reconstruct interrupted planar curves. The main interest of this section is the definition of the sub-Riemannian structure on $PT\mathbb{R}^2$, from which we are going to define the sub-elliptic diffusion equation.

Consider a smooth function $\gamma_0 : [a, b] \cup [c, d] \rightarrow \mathbb{R}^2$ (with $a < b < c < d$) representing a curve that is partially hidden or deleted in (b, c) . We assume that starting and ending points never coincide, i.e. $\gamma_0(b) \neq \gamma_0(c)$, and that initial and final velocities $\dot{\gamma}(b)$ and $\dot{\gamma}(c)$ are well defined and nonvanishing.

We want to find a curve $\gamma : [b, c] \rightarrow \mathbb{R}^2$ that completes γ_0 in the deleted part and that minimizes a cost depending both on the length and on the curvature K_γ of γ . Recall that $K_\gamma = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$ where (x, y) are the components of γ .

The fact that γ completes γ_0 means that $\gamma(b) = \gamma_0(b)$, $\gamma(c) = \gamma_0(c)$. It is also reasonable to require that the directions of tangent vectors coincide, i.e.

$$\dot{\gamma}(b) \approx \dot{\gamma}_0(b), \quad \dot{\gamma}(c) \approx \dot{\gamma}_0(c),$$

where

$$v_1 \approx v_2 \text{ if it exists } \alpha \in \mathbb{R} \setminus \{0\} \text{ such that } v_1 = \alpha v_2. \quad (1)$$

Remark 1. Notice that we have required boundary conditions on initial and final directions without orientation. Alternatively, the problem above can be formulated requiring boundary conditions with orientation, i.e. substituting in (2.1) the condition $\alpha \in \mathbb{R}^+$. However, this choice does not guarantee existence of minimizers for the cost we are interested in, see [5].

In this paper we are interested in the minimization of the following cost, defined for smooth curves γ in $[b, c]$:

$$J[\gamma] = \int_b^c \sqrt{\|\dot{\gamma}(t)\|^2 + \|\dot{\gamma}(t)\|^2 K_\gamma^2(t)} dt \quad (2)$$

This cost is interesting for several reasons:

- It depends on both length and curvature of γ . It is small for curves that are straight and short.
- It is invariant by rototranslation and by reparametrization of the curve.
- This cost has existence of minimizers, see [5]. Moreover, minimizers are analytic functions for which $\dot{\gamma} = 0$ at most in two isolated points. They are cusp points, see Fig. 2.

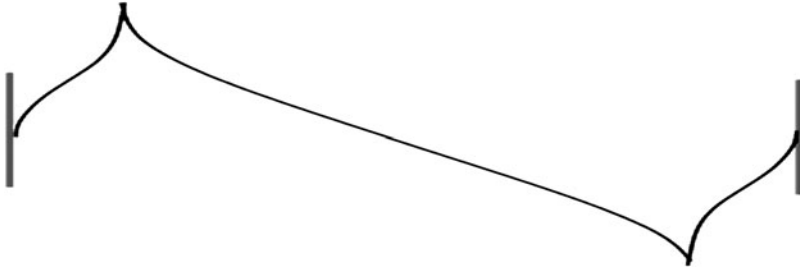


Fig. 2 A trajectory with two cusps

- Minimizing the cost $J[\gamma]$ is equivalent to minimize the energy-like cost

$$E[\gamma] = \int_b^c (\|\dot{\gamma}(t)\|^2 + \|\dot{\gamma}(t)\|^2 K_\gamma^2(t)) dt.$$

This is a good model for describing the energy necessary to excite orientation columns which are not directly excited by the image (because they correspond to the corrupted part of the image).

However, the most interesting aspect is that this cost is a Riemannian length for lift of planar curves on $PT\mathbb{R}^2$ (more precisely $J[\gamma]$ is a sub-Riemannian length, see below). As a consequence, we have a diffusion equation naturally associated with this cost.

Remark 2. One could argue that there is no reason to give the same weight to the length term $\|\dot{\gamma}\|$ and to the curvature term $\|\dot{\gamma}(t)\|^2 K_\gamma^2(t)$. However, define the cost

$$J_\beta[\gamma] := \int_b^c \sqrt{\|\dot{\gamma}(t)\|^2 + \beta^2 \|\dot{\gamma}(t)\|^2 K_\gamma^2(t)} dt$$

with a fixed $\beta \neq 0$. Then it is easy to prove that $J_\beta[\gamma_\beta] = \beta^2 J_1[\gamma] = \beta^2 J[\gamma]$. Therefore the problem of minimizing J_β is equivalent to the minimization of J with a suitable change of boundary conditions.

2.2 Left-Invariant Sub-Riemannian Manifolds

In this section we present a natural sub-Riemannian structure that can be defined on Lie groups. For a general definition of sub-Riemannian structures, see e.g. [3].

Definition 1. Let G be a Lie group with Lie algebra \mathfrak{L} and $\mathfrak{P} \subseteq \mathfrak{L}$ a subspace of \mathfrak{L} satisfying the **Lie bracket generating condition**

$$\text{Lie } \mathfrak{P} := \text{span}\{[\mathbf{p}_1, [\mathbf{p}_2, \dots, [\mathbf{p}_{k-1}, \mathbf{p}_k]]] \mid \mathbf{p}_i \in \mathfrak{P}\} = \mathfrak{L}.$$

Endow \mathfrak{P} with a positive definite quadratic form $\langle \cdot, \cdot \rangle$. Define a sub-Riemannian structure on G as follows:

- The distribution is the left-invariant distribution $\mathbf{\Delta}(g) := g\mathfrak{P}$
- The quadratic form \mathbf{g} on $\mathbf{\Delta}$ is given by $\mathbf{g}_g(v_1, v_2) := \langle g^{-1}v_1, g^{-1}v_2 \rangle$

In this case we say that $(G, \mathbf{\Delta}, \mathbf{g})$ is a left-invariant sub-Riemannian manifold.

A Lipschitz continuous curve $q(\cdot) : [0, T] \rightarrow G$ is said to be **horizontal** if

$$\dot{q}(t) \in \mathbf{\Delta}(q(t)) \text{ for almost every } t \in [0, T].$$

Given an horizontal curve $q(\cdot) : [0, T] \rightarrow G$, the its length is

$$l(q(\cdot)) = \int_0^T \sqrt{\mathbf{g}_{q(t)}(\dot{q}(t), \dot{q}(t))} dt.$$

The *distance* on $(G, \mathbf{\Delta}, \mathbf{g})$ is

$$d(q_0, q_1) = \inf\{l(q(\cdot)) \mid q(0) = q_0, q(T) = q_1, q(\cdot) \text{ horizontal}\}.$$

2.3 Lift of a Curve on $PT\mathbb{R}^2$ and the Sub-Riemannian Problem

Consider a smooth planar curve $\gamma : [b, c] \rightarrow \mathbb{R}^2$. This curve can be naturally lifted to a curve $\tilde{\gamma} : [b, c] \rightarrow PT\mathbb{R}^2$ in the following way. Let $(x(t), y(t))$ be the Euclidean coordinates of $\gamma(t)$. Then the coordinates of $\tilde{\gamma}(t)$ are $(x(t), y(t), \theta(t))$, where $\theta(t) \in \mathbb{R}/(\pi\mathbb{Z})$ is the direction of the vector $(x(t), y(t))$ measured with respect to the vector $(1, 0)$. In other words,

$$\theta(t) = \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right). \quad (3)$$

The requirement that a curve $(x(t), y(t), \theta(t))$ satisfies the constraint (3), can be slightly generalized by requiring that $(x(t), y(t), \theta(t))$ is an admissible trajectory of the control system on $PT\mathbb{R}^2$:

$$(\dot{x}, \dot{y}, \dot{\theta}) = u_1 (\cos(\theta), \sin(\theta), 0) + u_2 (0, 0, 1), \quad (4)$$

with $u_1, u_2 \in L^1([b, c], \mathbb{R})$. Indeed each smooth trajectory γ satisfying **[H]** is an admissible trajectory of (4).

Notice that $u_1(t)^2 = \|\dot{\gamma}(t)\|^2$, $u_2(t)^2 = \dot{\theta}^2 = \|\dot{\gamma}(t)\|^2 K_\gamma(t)^2$, thus

$$J[\gamma] = \int_b^c \sqrt{u_1(t)^2 + u_2(t)^2} dt.$$

Hence, the problem of minimizing the cost (2.1) on set of curves \mathcal{D} is (slightly) generalized into the optimal control problem (here $q(\cdot) = (x(\cdot), y(\cdot), \theta(\cdot))$)

$$\begin{aligned} \dot{q} &= u_1 (\cos(\theta), \sin(\theta), 0) + u_2 (0, 0, 1), \\ l(q(\cdot)) &= \int_b^c \sqrt{u_1(t)^2 + u_2(t)^2} dt \rightarrow \min, \\ q(b) &= (x_b, y_b, \theta_b), \quad q(c) = (x_c, y_c, \theta_c). \end{aligned} \quad (5)$$

This can be seen as a projective version of the control of the unicycle, since the dynamics for the unicycle is given by the same problem with $\theta \in \mathbb{R}/(2\pi)$, see below.

2.4 The Sub-Riemannian Problem on $SE(2)$

In what follows, it is convenient to lift the sub-Riemannian problem on $PT\mathbb{R}^2$ (5) on the group of rototranslation of the plane $SE(2)$, in order to take advantage of the group structure. This is the group of matrices of the form

$$SE(2) = \left\{ \begin{pmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{pmatrix} \mid \begin{array}{l} \theta \in \mathbb{R}/2\pi, \\ x, y \in \mathbb{R} \end{array} \right\}$$

In the following we often denote an element of $SE(2)$ as $g = (x, y, \theta)$.

A basis of the Lie algebra of $SE(2)$ is $\{p_1, p_2, p_3\}$, with

$$p_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad p_2 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad p_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

We define a trivializable sub-Riemannian structure on $SE(2)$ as presented in Sect. 2.2: consider the two left-invariant vector fields $X_i(g) = gp_i$ with $i = 1, 2$ and define

$$\mathbf{\Delta}(g) = \text{span}\{X_1(g), X_2(g)\}, \quad \mathbf{g}_g(X_i(g), X_j(g)) = \delta_{ij}.$$

In coordinates, the optimal control problem

$$\begin{aligned} \dot{g} &\in \mathbf{\Delta}(g), \quad l(g(\cdot)) = \int_b^c \sqrt{\mathbf{g}_{g(t)}(\dot{g}, \dot{g})} dt \rightarrow \min, \\ g(b) &= g_b, \quad g(c) = g_c, \end{aligned} \quad (6)$$

has the form (5), but with $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$.

The problem (6) is called the optimal control of the unicycle. It has been studied by Yuri Sachkov in the series of papers [14–16] (the first one in collaboration with I. Moiseev), where he computed the optimal synthesis. The solution of the minimization problem (5) on $PT\mathbb{R}^2$, can be obtained from the one of the problem on $SE(2)$. We will not go further on the problem of reconstruction of curves in this paper. For our purpose of image reconstruction, only the sub-Riemannian structure is important, since it permits to define intrinsically a nonisotropic diffusion process.

2.5 The Hypoelliptic Heat Kernel

When the image is not just a curve, one cannot use the algorithm described above in which curves are reconstructed by solving a sub-Riemannian problem with fixed boundary conditions. Indeed, even if a corrupted image is thought as a set of interrupted curves (the level sets), it is unclear how to connect the different components of the level set among them (see Fig. 3). Moreover, if the corrupted part contains the neighborhood of a maximum or minimum, then certain level sets are completely missing and cannot be reconstructed.

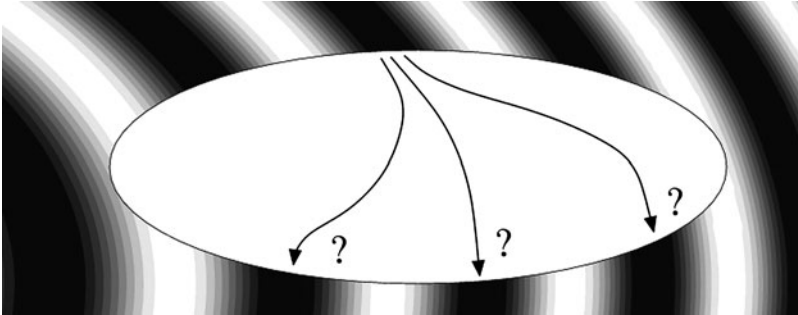


Fig. 3 The problem of connecting level sets

Next we use the original image as an initial condition for the nonisotropic diffusion equation associated to the sub-Riemannian structure.

Roughly speaking, this diffusion equation describes the density of probability of finding the system in the point (x, y, θ) at time t , while replacing the controls in (5) with independent Wiener processes.

This diffusion equation is

$$\partial_t \phi(x, y, \theta, t) = \Delta_H \phi(x, y, \theta, t) \quad (7)$$

where $\Delta_H = X_1^2 + X_2^2 = (\cos(\theta)\partial_x + \sin(\theta)\partial_y)^2 + \partial_\theta^2$.

The diffusion described by the (7) is highly non isotropic. Indeed one can estimate the heat kernel in terms of the sub-Riemannian distance (see for instance [2]), that is highly non isotropic as a consequence of the ball-box theorem (see for instance [4]). The heat kernel for the (7) on $SE(2)$ was computed in [2], see also [8]. More precisely, thanks to the left-invariance of X_1 and X_2 , there exists the kernel p_t such that

$$e^{t\Delta_H} \phi_0(g) = \phi_0 * p_t(g) = \int_G \phi_0(h) p_t(h^{-1}g) \mu(h)$$

is the solution for $t > 0$ to (7) with initial condition $\phi(0, g) = \phi_0(g) \in L^1(SE(2), \mathbb{R})$ with respect to the Haar measure μ .

We have computed p_t in [2]:

$$\begin{aligned} p_t(g) = & \int_0^{+\infty} \lambda \left(\sum_{n=0}^{+\infty} e^{a_n^\lambda t} \langle \text{ce}_n \left(\theta, \frac{\lambda^2}{4} \right), \mathfrak{X}^\lambda(g) \text{ce}_n \left(\theta, \frac{\lambda^2}{4} \right) \rangle + \right. \\ & \left. + \sum_{n=1}^{+\infty} e^{b_n^\lambda t} \langle \text{se}_n \left(\theta, \frac{\lambda^2}{4} \right), \mathfrak{X}^\lambda(g) \text{se}_n \left(\theta, \frac{\lambda^2}{4} \right) \rangle \right) d\lambda. \end{aligned} \quad (8)$$

Here

$$\begin{aligned} \mathfrak{X}^\lambda(g) : L^2(S^1, \mathbb{C}) &\rightarrow L^2(S^1, \mathbb{C}), \\ \mathfrak{X}^\lambda(g) \psi(\alpha) &= e^{i\lambda(x \cos(\alpha) - y \sin(\alpha))} \psi(\alpha + \theta) \end{aligned}$$

is the representation of the group element $g = (x, y, \theta)$ on $L^2(S^1, \mathbb{C})$. The functions se_n and ce_n are the 2π -periodic Mathieu cosines and sines, and $\langle \phi_1, \phi_2 \rangle := \int_{S^1} \phi_1(\alpha) \phi_2(\alpha) d\alpha$. The eigenvalues of the hypoelliptic Laplacian are

$$a_n^\lambda := -\frac{\lambda^2}{4} - a_n \left(\frac{\lambda^2}{4} \right) \quad \text{and} \quad b_n^\lambda := -\frac{\lambda^2}{4} - b_n \left(\frac{\lambda^2}{4} \right),$$

where a_n and b_n are characteristic values for the Mathieu equation. For details about Mathieu functions see for instance [1, Chap. 20].

Applying Hörmander theorem [11], we can prove that the operator $\partial_t - \Delta_H$ is hypoelliptic, then the kernel is a \mathcal{C}^∞ function of $(t, g) \in \mathbb{R}^+ \times G$.

$SE(2)$ is a double covering of $PT\mathbb{R}^2$. To a point $(x, y, \theta) \in PT\mathbb{R}^2$ correspond the two points (x, y, θ) and $(x, y, \theta + \pi)$ in $SE(2)$. From the next proposition it follows that we can interpret the hypoelliptic heat equation on $PT\mathbb{R}^2$ as the hypoelliptic heat equation on $SE(2)$ with a symmetric initial condition. It permits also to compute the heat kernel on $PT\mathbb{R}^2$ starting from the one on $SE(2)$.

Proposition 1. *Let $\phi_0 \in L^1(SE(2), \mathbb{R})$ and assume that $\phi_0(x, y, \theta) = \phi_0(x, y, \theta + \pi)$ a.e. Then the solution at time t of the hypoelliptic heat equation (7) on $SE(2)$, having ϕ_0 as initial condition at time zero, satisfies*

$$\phi(t, x, y, \theta) = \phi(t, x, y, \theta + \pi). \quad (9)$$

Moreover if $\phi_0 \in L^1(PT\mathbb{R}^2, \mathbb{R})$, then the solution at time t of the hypoelliptic heat equation on $PT\mathbb{R}^2$ (7) having ϕ_0 as initial condition at time zero is given by

$$\phi(t, q) = \int_{PT\mathbb{R}^2} \phi_0(\bar{q}) P_t(q, \bar{q}) d\bar{q} \quad (10)$$

where

$$P_t(x, y, \theta, \bar{x}, \bar{y}, \bar{\theta}) := p_t((\bar{x}, \bar{y}, \bar{\theta})^{-1} \circ (x, y, \theta + \pi)) + p_t((\bar{x}, \bar{y}, \bar{\theta})^{-1} \circ (x, y, \theta)). \quad (11)$$

In the right hand side of (11), the group operations are intended in $SE(2)$.

2.6 The Mathematical Algorithm

In this section we describe the mathematical algorithm for image reconstruction.

STEP 1: Smoothing of \mathcal{I}_c

Assume that the grey level of a corrupted image is described by a function $\mathcal{I}_c : \mathcal{D}_c := \mathcal{D} \setminus \Omega \rightarrow [0, \infty[$. The set Ω represents the region where the image is corrupted. The subscript “ c ” means “corrupted”. After making the convolution with a Gaussian of standard deviations $\sigma_x, \sigma_y > 0^2$, we get a smooth function defined on \mathbb{R}^2 , which generically is a Morse function:

$$f_c = \mathcal{I}_c * G(\sigma_x, \sigma_y).$$

Indeed, the following theorem holds:

Theorem 1. *Let Ω be a bounded domain of the plane \mathbb{R}^2 . Fix $\sigma_x, \sigma_y > 0$. Consider the convolution map³*

$$\rho : \begin{array}{ccc} L^2(\mathcal{D}, \mathbb{R}) & \rightarrow & d\mu_{\#}^{\infty}(\mathbb{R}^2) \\ \mathcal{I} & \mapsto & \mathcal{I} * G, \end{array}$$

where $G(\sigma_x, \sigma_y)$ is the Gaussian centred at $(0, 0)$ with standard deviations σ_x, σ_y . Let K be a compact subset of \mathbb{R}^2 with non-empty interior. Then the set

$$\mathbb{X}_K := \left\{ \mathcal{I} \in L^2(\mathcal{D}, \mathbb{R}) \text{ such that } \{\mathcal{I} * G\}_{|_K} \text{ is Morse} \right\}$$

is open and dense in $L^2(\mathcal{D}, \mathbb{R})$.

² \mathcal{I}_c is considered to be zero outside \mathcal{D}_c .

³ \mathcal{I} is considered to be zero outside \mathcal{D} .

STEP 2: The lift of $f_c : \mathbb{R}^2 \rightarrow \mathbb{R}$ to a function $\bar{f}_c : PT\mathbb{R}^2 \rightarrow \mathbb{R}$

This is made by associating to every point (x, y) of \mathbb{R}^2 the direction $\theta \in \mathbb{R}/(\pi\mathbb{Z})$ of the level set of f_c at the point (x, y) . This direction is well defined only at points where $\nabla f_c \neq 0$. At points where $\nabla f_c = 0$, we associate all possible directions. More precisely, we define the *lifted support* S_f , associated to the function f as follows,

$$S_f = \{(x, y, \theta) \in \mathbb{R}^2 \times P^1 \text{ s.t. } \nabla f_c(x, y) \cdot (\cos(\theta), \sin(\theta)) = 0\},$$

where the dot means the standard scalar product on \mathbb{R}^2 .

STEP 3: Lift of f_c to a distribution in $\mathbb{R}^2 \times P^1$ supported on S_f

Define the distribution on $\mathbb{R}^2 \times P^1$:

$$\bar{f}_c(x, y, \theta) = f_c(x, y) \delta(\mathbf{d}((x, y, \theta), S_f))$$

where δ is the Dirac delta distribution and $\mathbf{d}((x, y, \theta), S_f)$ is the standard Euclidean distance of the point (x, y, θ) from S_f .

Remark 3. This step is formally necessary for the following reason. The surface S_c is 2D in a 3D manifold, hence the real function f_c defined on it is vanishing a.e. as a function defined on $PT\mathbb{R}^2$. Thus the hypoelliptic evolution of f_c (that is, the next STEP 4) produces a vanishing function. Multiplying f_c by a Dirac delta is a natural way to obtain a nontrivial evolution.

STEP 4: Hypoelliptic evolution

Fix $T > 0$. Compute the solution at time T to the Cauchy problem,

$$\begin{cases} \partial_t \phi(g, t) = ((\cos(\theta) \partial_x + \sin(\theta) \partial_y)^2 + \beta^2 \partial_\theta^2) \phi(g, t) \\ \phi(g, 0) = \bar{f}_c(g). \end{cases} \quad (12)$$

Remark 4. In the formula above, the Laplacian is given by $X_1^2 + \beta^2 X_2^2$, thus it depends on the fixed parameter β . This means that we use evolution depending on the cost J_β rather than J . Tuning this parameter will provide better results of the reconstruction algorithm.

STEP 5: Projecting down

Compute the reconstructed image by choosing the maximum value on the fiber.

$$f_T(x, y) = \max_{\theta \in P^1} \phi(x, y, \theta, T).$$

3 Numerical Implementation

In this section we present the algorithm used in our simulations. We focus on the main difficulties of its numerical implementation and how to overcome them. STEP 1 is well implemented in several mathematical softwares.

3.1 STEPS 2-3: Lift of an Image

The formal definition of the lifted function is hard to realize numerically for two reasons: the discretization of the angle variable α and the presence of a Dirac delta.

Both issues are solved changing the definition of the lifted function:

$$\bar{f}_c(x, y, \theta) = f(x, y) \phi(\nabla f(x, y), \theta),$$

where $\phi(0, \theta) = 1/(2\varepsilon) \forall \theta \in \mathbb{R}/(\pi\mathbb{Z})$ and $\phi(v, \theta) = \phi^1(\arg(v) - \theta)$ where $\phi^1(\beta)$ is the π -periodic function assuming the following values over the interval $[0, \pi]$:

$$\phi^1(\beta) := \begin{cases} 1/(2\varepsilon) & \text{if } \beta \in [\frac{\pi}{2} - \varepsilon, \frac{\pi}{2} + \varepsilon], \\ 0 & \text{otherwise,} \end{cases}$$

for a fixed $\varepsilon > 0$.

Since the space is discretized, the non-zero values of \bar{f}_c are no longer defined over a set of null measure, hence the discretized hypoelliptic diffusion gives non vanishing function for all $T > 0$. Thus, it is not necessary to perform STEP 3.

3.2 STEP 4: Hypoelliptic Evolution

In this section we give a method to efficiently compute the hypoelliptic evolution (12). We first fix the notation for n -dimensional Fourier transform and Fourier series. Given $f(x)$ defined on \mathbb{R}^n (or on $[0, 2\pi]$), we have

$$\begin{aligned} \mathbf{FT}_x^\omega[f] &= \int_{\mathbb{R}^n} f(x) e^{-i\omega \cdot x} dx, \\ \mathbf{FS}_x^n[f] &= \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx} dx. \end{aligned}$$

Consider (12) and transform it into the corresponding problem on $SE(2)$. Then apply the following transformation

$$f(x, y, t) \Rightarrow \tilde{f}^k(\xi_x, \xi_y, t) := \mathbf{FT}_{x,y}^{\xi_x, \xi_y} \left[\mathbf{FS}_\theta^k[f(x, y, \theta)] \right].$$

Simple computations lead to prove that \tilde{f}^k satisfies the following Cauchy problem

$$\begin{cases} \partial_t \tilde{f}^k = A \tilde{f}^k + B \tilde{f}^k \\ \tilde{f}^k(t=0) = \tilde{f}_0^k := \mathbf{FT}_{x,y}^{\xi_x, \xi_y} [\mathbf{FS}_{\theta}^k[\tilde{f}_c(x, y, \theta)]] \end{cases} \quad (13)$$

where $A \tilde{f}^k = \alpha_k \tilde{f}^k$, $B \tilde{f}^k = b_1 \tilde{f}^{k+2} + b_2 \tilde{f}^{k-2}$, with

$$\alpha^k = -\frac{\xi_x^2}{2} - \frac{\xi_y^2}{2} - \beta^2 k^2, \quad b_1 = -\frac{\xi_x^2}{2} + \frac{\xi_y^2}{2} + i \frac{\xi_x \xi_y}{2}$$

and b_2 is the conjugate of b_1 . The solution of (13) satisfies

$$\tilde{f}^k(t) = \exp(At) \tilde{f}_0^k + \int_0^t \exp(A(t-s)) B \tilde{f}^k(s) ds,$$

that can be seen as a variation of constant formula. Applying it iteratively, we can find approximated solutions of (13) of a given order with respect to time. The first order approximation is

$$\tilde{f}^k(t) = e^{\alpha_k t} \tilde{f}_0^k + b_1 \int_0^t e^{\alpha_{k+2}(t-s)} e^{\alpha_k s} \tilde{f}_0^{k+2} ds + b_2 \int_0^t e^{\alpha_{k-2}(t-s)} e^{\alpha_k s} \tilde{f}_0^{k-2} ds + o(t).$$

The integrals can be computed explicitly, with special cases for $k = \pm 1$. Also for higher order approximations the integrals can be computed explicitly, but they present a higher number of special cases. In the examples provided below, we will use only the first or the second order approximations.

4 Results of Image Reconstruction

In this section we provide preliminary results of image reconstruction using the algorithm we presented above. For all these examples, we have manually tuned the parameters β , that is the relative weight, and T , the final time of evolution.

The key point of this algorithm is that the corrupted part is unknown. For this reason, one of the main properties we have to provide is that the evolution process does not modify the noncorrupted part. Empirical observations lead to notice that this goal is reached fixing a small value of β . Indeed, it means that the evolution on $PT\mathbb{R}^2$ along the direction θ has small velocity, so the noncorrupted parts do not change their behavior.

In the first example (Fig. 4), we provide the reconstruction of an image in which the level sets are corrupted horizontal lines (except for the corrupted part). In the second example, (Fig. 5) the level sets are corrupted ellipses.

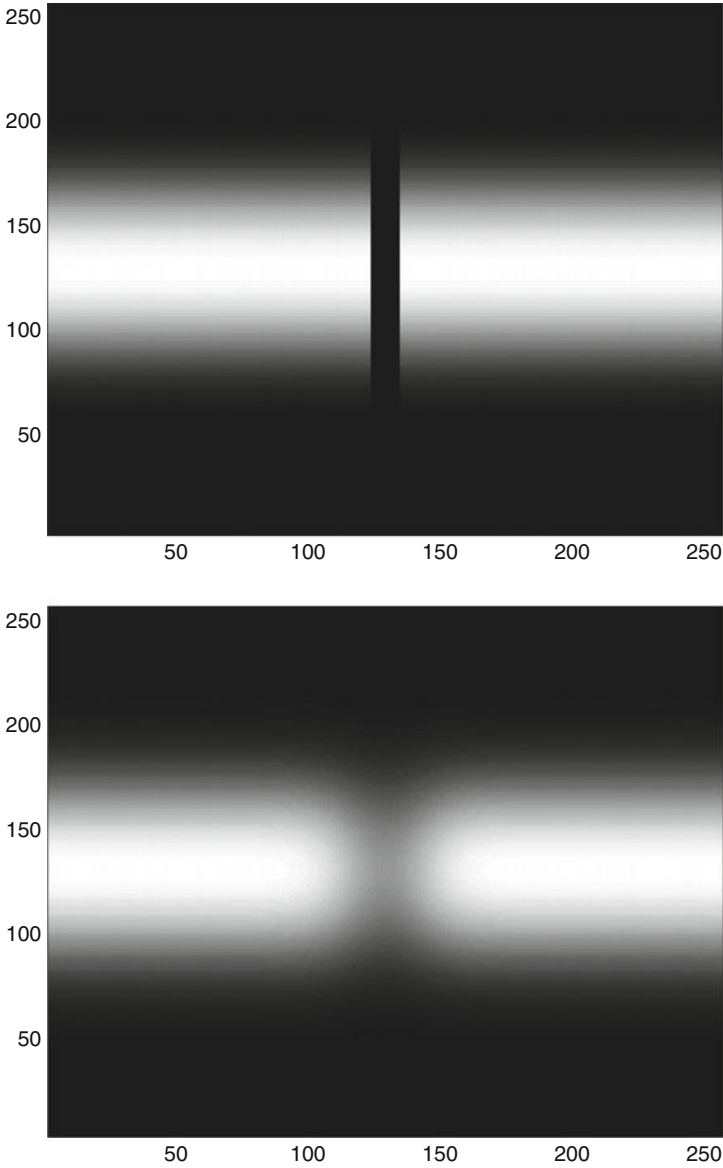


Fig. 4 Reconstruction of an image with horizontal level sets

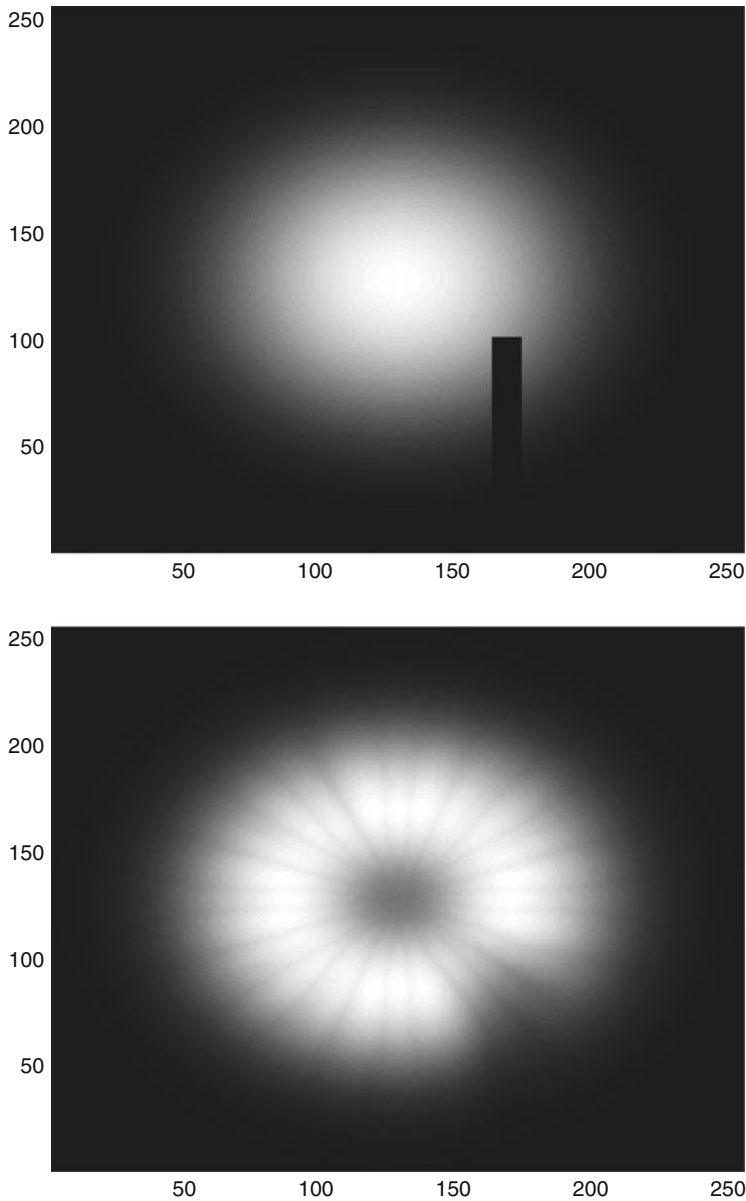


Fig. 5 Reconstruction of an image with elliptic level sets

Acknowledgments

This research has been supported by the European Research Council, ERC StG 2009 GeCoMethods, contract number 239748, by the ANR GCM, program “Blanc CSD 5”, and by the DIGITEO project “CONGEO”.

References

1. Abramowitz, M., Stegun, I.A. (eds.): Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. National Bureau of Standards Appl. Math., Washington (1964)
2. Agrachev, A., Boscain, U., Gauthier, J.-P., Rossi, F.: The intrinsic hypoelliptic Laplacian and its heat kernel on unimodular Lie groups. *J. Funct. Anal.* **256**, 2621–2655, (2009)
3. Agrachev, A.A., Sachkov, Yu.L.: Control Theory from the Geometric Viewpoint. Encyclopaedia of Mathematical Science, vol. 87. Springer, Berlin (2004)
4. Bellaïche, A.: The tangent space in sub-Riemannian geometry. In: Bellaïche, A., Risler, J.-J. (eds.) Sub-Riemannian Geometry. Progress in Mathematics, vol. 144, pp. 1–78. Birkhäuser, Basel (1996)
5. Boscain, U., Charlot, G., Rossi, F.: Existence of planar curves minimizing length and curvature, Proceedings of the Steklov Institute of Mathematics, vol. 270, no. 1, pp. 43–56, 2010
6. Cao, F., Gousseau, Y., Masnou, S., Pérez, P.: Geometrically guided exemplar-based inpainting, submitted
7. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. *J. Math. Imaging. Vis.* **24**(3), 307–326 (2006)
8. Duits, R., Franken, E.: Left-invariant Stochastic Evolution Equations on $SE(2)$ and its Applications to Contour Enhancement and Contour Completion via Invertible Orientation Scores, CASA report 35 – 2007, arxiv:0711.0951
9. Duits, R., Franken, E.: Left-invariant parabolic evolutions on $SE(2)$ and contour enhancement via invertible orientation scores, parts I and II. *Q. Appl. Math.* **68**, 255–331 (2010)
10. Duits, R., et al.: Image analysis and reconstruction using a wavelet transform.... *Int. J. Comput. Vis.* **72**, 79–102 (2007)
11. Hörmander, L.: Hypoelliptic second order differential equations. *Acta Math.* **119**, 147–171 (1967)
12. Langer, M.: Computational Perception, Lecture Notes. <http://www.cim.mcgill.ca/~langer/646.html> (2008)
13. Marr, D., Hildreth, E.: Theory of edge detection. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **207**(1167), 187–217 (1980)
14. Moiseev, I., Sachkov, Yu.L.: Maxwell strata in sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV 16, no. 2, pp. 380–399, 2010
15. Sachkov, Yu.L.: Conjugate and cut time in the sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV, Volume 16, Number 4, pp. 1018–1039
16. Sachkov, Yu.L.: Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV, to appear
17. Mumford, D.: Elastica and Computer Vision. In: Algebraic Geometry and its Applications, pp. 507–518. Springer, Berlin (1993)
18. Peichl, L., Wässle, H.: Size, scatter and coverage of ganglion cell receptive field centres in the cat retina. *J. Physiol.* **291**, 117–141 (1979)
19. Petitot, J.: Neurogéométrie de la vision, Les Éditions de l’École Polytechnique (2008)

Projective Invariant Features Detection and the Registration Group

Françoise Dibos

Abstract Affine invariant scale space analyses have been introduced in the 1990s and largely used in shape recognition applications. Nowadays they are also used for the determination of points of interests. Nevertheless, a projective analysis is necessary if information is to be gathered from images taken of the same scene but under different points of view. In previous work, we replaced the projective analysis due to the heat equation by a flow of second order equations. The registration group allows to model the deformations of an image due to camera motion by a 6 parameter group. We will show that it allows also to view in a new light the projective analysis and the Affine Morphological Scale Space of Alvarez, Lions, Guichard and Morel (Arch. Ration. Mechan. 16:200–257 (1993)). Moreover, the registration group gives directly a first approach for a Projective Scale Invariant Feature Tracking algorithm.

1 Introduction

The search for affine and projective differential invariants occupied a large place in image analysis the 1990s [1–3, 5, 9, 10, 12, 17], the aim being applications to shape recognition. These invariants are well known today and are mostly associated to other descriptors [6, 16]. Moreover, they can lead to the determination of interest points [13–15, 18]. These invariants were found within the framework of scale space analysis of images. A nice way of understanding these works is to refer to the evolution of a level line of an image under the heat-like equation.

$$\mathcal{C}_t = \mathcal{C}_{ss}$$

where s is the euclidean/ affine/projective arc length.

Françoise Dibos

LAGA, Institut Galilée, Université Paris 13, 99 avenue J-B Cément, 93430 Villetaneuse, France
e-mail: dibos@math.univ-paris13.fr

For example if s is the affine arc length, the evolution of \mathcal{C} is given by:

$$\mathcal{C}_t = (\kappa)^{\frac{1}{3}} \mathcal{N}$$

where κ is the euclidean curvature and \mathcal{N} the unit interior normal to curve \mathcal{C} . Then, the corresponding affine analysis for images, the so called Affine Morphological Scale Space (AMSS) [1], is given by

$$(AMSS) \quad \frac{\partial g}{\partial t} = t^{\frac{1}{3}} \|Dg\| \left(\operatorname{div} \frac{Dg}{\|Dg\|} \right)^{\frac{1}{3}} = t^{\frac{1}{3}} \|Dg\| (\operatorname{curv} g)^{\frac{1}{3}}.$$

Projective differential invariants were more difficult to finalize and to use. Indeed the heat-like projective equation is a fifth order equation developing singularities and not defined for lines and circles. More exactly this equation is

$$\frac{\partial \mathcal{C}}{\partial t}(m) = \frac{(\kappa)^{\frac{1}{3}}}{Z_m^2} \mathcal{N}$$

where Z_m is the Cartan's scale factor i.e.

$$Z_m = \left(\frac{\mu_s}{2} \right)^{\frac{1}{3}}$$

where μ is the affine curvature and s the affine arc length. Z_m is a fifth order differential element [9]. To resolve these difficulties, we proposed to replace this fifth order equation by a flow of second order equations [5]. Moreover, a modeling of the deformation of an image of a scene under various points of view, adapted to the camera displacement, conducted to replace the projective group by the registration group [4, 8].

This new modeling allows a simplification of the projective analysis previously proposed in the literature and shows also that the Affine Morphological Scale Space is a projective scale space in the center of the image.

Furthermore, the registration group gives directly an idea for a PSIFT (Projective Scale Invariant Feature Tracking) algorithm of which the ASIFT of Morel and Yu [15] is the affine approximation.

This paper is divided in three main parts. In Sect. 2, we recall some properties of the registration group. Section 3 is devoted to the simplification of the projective analysis previously proposed in the literature. In Sect. 4 we present a projective invariant feature tracking and show how the ASIFT algorithm is its affine approximation when the camera is far from the scene.

2 The Registration Group

2.1 Definition and Properties

The registration group has been introduced in [4, 8]. See [7, 11] for applications of this group in the determination of the movement of a camera from a sequence of images and a depth estimation of a scene.

The registration group is a 6 parameters group, instead of the 8 parameters for the projective group. It can be used for modeling the deformation of an image of a 3D scene under various points of view. The registration group is consistent with the displacement of a pinhole camera.

If Φ is a projective application in the registration group and f and g are two views of the same scene,

$$f = \Phi g \quad ; \quad g = \Phi^{-1} f$$

where Φ^{-1} , the inverse of Φ in the registration group, can be associated in a very natural way to the inverse displacement of the camera. This is not, in general, the case when the projective group is used.

A camera displacement may be decomposed in a 3D rotation and in a 3D translation. To these components may be associated two subgroups of the projective group. The special orthogonal group \mathcal{O}^+ for the rotations and the subgroup \mathcal{H} for the translations where

$$\mathcal{H} = \left\{ H = \lambda \begin{pmatrix} 1 & 0 & A \\ 0 & 1 & B \\ 0 & 0 & 1+C \end{pmatrix} \mid \det H = 1 \right\}$$

The homography associated to H is an homothety.

The use of the projective group deeply relies on the fact that these two subgroups generate the projective group. The registration group proposes an answer to the fact that these two subgroups do not commute.

Definition 1. The registration group \mathcal{G} is the set of 3D matrices Φ

$$\Phi = \begin{pmatrix} a & b & c+A \\ a' & b' & c'+B \\ a'' & b'' & c''+C \end{pmatrix} = RH$$

where $R = \begin{pmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{pmatrix}$ is the matrix of a 3D rotation.

Each matrix of \mathcal{G} is associated to the pair (R, T) with $T = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$

$$\Phi \longleftrightarrow (R, T)$$

The composition law is denoted by \star , if

$\Phi = \Phi_1 \star \Phi_2$, then the couple (R, T) associated to Φ is given by

$$\begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_1 & T_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_2 & T_2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_1 R_2 & R_1(T_2) + T_1 \\ 0 & 1 \end{pmatrix}.$$

If $\det \Phi \neq 0$, we can associate to Φ a projective transformation φ

$$\varphi(x, y) = \left(\frac{ax + by + c + A}{a''x + b''y + c'' + C}, \frac{a'x + b'y + c' + B}{a''x + b''y + c'' + C} \right)$$

From now on, when there is no ambiguity, we keep the same notation for φ and Φ .

Properties.

1. In \mathcal{G} the inverse of $\Phi \longleftrightarrow (R, T)$ is

$$\Phi^{-1, \star} \longleftrightarrow (R^{-1}, -R^{-1}(T))$$

2. Each matrix $\Phi \longleftrightarrow (R, T)$ in \mathcal{G} may be decomposed in

$$\Phi = R \begin{pmatrix} 1 & 0 & T \cdot R(i) \\ 0 & 1 & T \cdot R(j) \\ 0 & 0 & 1 + T \cdot R(k) \end{pmatrix} = R \begin{pmatrix} 1 & 0 & R^{-1}(T) \cdot i \\ 0 & 1 & R^{-1}(T) \cdot j \\ 0 & 0 & 1 + R^{-1}(T) \cdot k \end{pmatrix} = RH.$$

3. We have

$$\Phi = RH \iff \Phi = R \star H \quad \text{with} \quad H \longleftrightarrow (Idt, R^{-1}(T)).$$

4. If $\Phi = RH$ then

$$\Phi^{-1, \star} = R^{-1} \tilde{H} \quad \text{with} \quad \tilde{H} \longleftrightarrow (Idt, -T)$$

5. Decomposition of Φ :

Let $R_{\Delta, \alpha}$ be the 3D rotation with axis Δ in the plane (O, i, j) such that $\widehat{(i, \Delta)} = \theta$.

The angle of $R_{\Delta, \alpha}$ being α . Let R_{β}^k the 3D rotation with axis k and angle β .

Each rotation R may be written as

$$R = R_{\Delta, \alpha} R_{\beta}^k$$

so

$$\forall \Phi \in \mathcal{G} \quad \Phi = R_{\Delta, \alpha} R_{\beta}^k H = R_{\Delta, \alpha} S$$

where S is a similarity (Fig. 1).

The rotation $R_{\Delta, \alpha}$ changes the optical axis of the camera; $R_{\Delta, \alpha}$ is the “pure” projective homography.

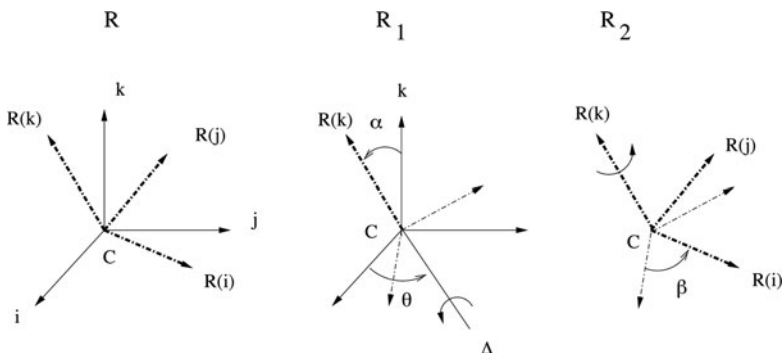


Fig. 1 Decomposition of a rotation

2.2 3D Homogeneous Image Associated to a Planar Image f and Deformations of f

Let \mathcal{E} be the affine space and (X, Y, Z) the coordinates of a point $M \in \mathcal{E}$ in the orthonormal basis $\mathcal{R} = (O, i, j, k)$. Let f be a planar image living in the affine plane ($Z = 1$). We associate to f the 3D homogeneous image F defined by

$$F(X, Y, Z) = f\left(\frac{X}{Z}, \frac{Y}{Z}\right) = f(x, y) \quad \text{if } Z \neq 0.$$

A projective deformation of f may be viewed as an image g obtained by cutting the 3D image F by a plane P (Fig. 2).

Let $\mathcal{R}' = (O', R(i), R(j), R(k))$ be a new orthonormal basis. We can consider this new basis as the position – orientation of a camera initially situated in O with optical axis k .

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} + \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

with

$$\overrightarrow{OO'} = Ai + Bj + Ck; \quad R = \begin{pmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{pmatrix}$$

Let P be the plane with orthonormal basis $(\omega, R(i), R(j))$ where $\overrightarrow{O'\omega} = R(k)$ and M a point in P . The coordinates of M are $(X' = x', Y' = y', Z' = 1)$ in \mathcal{R}' , so g may be written in the basis $(\omega, R(i), R(j))$

$$M \in P \quad g(x', y') = g(M) = F(M) = f(m) = f\left(\frac{X}{Z}, \frac{Y}{Z}\right).$$

$$\mathcal{R}_2 = (O_2, R_2 \circ R_1(i), R_2 \circ R_1(j), R_2 \circ R_1(k)) \quad \overrightarrow{O_1 O_2} = A_2 R_1(i) + B_2 R_1(j) + C_2 R_1(k)$$

Let g be the planar image obtained by cutting F by the plane $P = (\omega_1, R_1(i), R_1(j))$ where $\overrightarrow{O_1 \omega_1} = R_1(k)$. We have

$$g = g_f = \Phi_1 f \quad \Phi_1 \longleftrightarrow (R_1, T_1) \quad \text{with} \quad T_1 = \overrightarrow{O O_1}$$

The notation g_f emphasises that g is seen from f . The planar image g thus being defined, let G be the homogeneous 3D image associated to g .

Let Q be the plane $Q = (\omega_2, R_2 \circ R_1(i), R_2 \circ R_1(j))$ where $\overrightarrow{O_2 \omega_2} = R_2(k)$, Q cuts the homogeneous 3D images F and G . Now h may be connected to f or to g and there are two ways to parametrize h :

$$\forall M \in Q \quad h_f(M) = h_f(X'', Y'', 1) = h_f(x'', y'') = F(M)$$

$$\forall M \in Q \quad h_g(M) = h_g(X'', Y'', 1) = h_g(x'', y'') = G(M)$$

Proposition 1. *With the notations given above, we have*

$$h_g = \Phi_2 g \quad \Phi_2 \longleftrightarrow (R_2, T_2)$$

Proof. By construction. \square

Proposition 2. *With the notations given above, we have*

$$h_f = \Phi f = (\Phi_1 \star \Phi_2) f$$

with

$$\Phi_1 \longleftrightarrow (R_1, T_1) \quad T_1 = \begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix}$$

and

$$\Phi_2 \longleftrightarrow (R_2, T_2) \quad T_2 = \begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix}$$

Proof. The change of coordinates gives

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R_1 \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} + \begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix}$$

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R_2 \begin{pmatrix} X'' \\ Y'' \\ Z'' \end{pmatrix} + \begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix}.$$

So

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R_1 R_2 \begin{pmatrix} X'' \\ Y'' \\ Z'' \end{pmatrix} + R_1(T_2) + T_1$$

and

$$(x, y) = \left(\frac{X}{Z}, \frac{Y}{Z} \right) = \varphi(x'', y'')$$

where φ is the homography with matrix $\Phi \longleftrightarrow (R_1 R_2, R_1(T_2) + T_1)$. Thus

$$f(x, y) = f(\varphi(x'', y'')) = h_f(x'', y'')$$

and

$$h_f = \Phi f \quad \Phi = \Phi_1 \star \Phi_2 .$$

□

Proposition 3. *Let f, g, h three images of a same 3D scene, and*

$$h = h_f = \Phi f \quad \text{and} \quad g = g_f = \Phi_1 f ,$$

the relation between h and g is

$$h = \Phi \left(\Phi_1^{-1, \star} g \right) = (\Phi_1^{-1, \star} \star \Phi) g = \Phi_2 g = h_g .$$

Proof. Apply proposition 2 □

Remark 1. In the projective group, if $h = \Phi f$ and $g = \Phi_1 f$ one would have

$$h = (\Phi_1^{-1} \Phi) g$$

where Φ_1^{-1} is the projective application associated to the inverse matrix of Φ_1 and no movement of camera can be associated to Φ_1^{-1} .

Remark 2. In the registration group, we parametrize an image g which is the deformation of an image f under a movement of the camera thanks to the displacement of the camera. This modeling allows to ignore the initial position of the 3D scene. Accordingly, if f and g are two images of the same 3D scene associated respectively to the position-orientation \mathcal{R} and \mathcal{R}' of the camera; g is written by looking at f , i.e. g is the deformation of f due to the movement of the camera from \mathcal{R} to \mathcal{R}' and f is written by looking at g , i.e. f is the deformation of g due to the inverse movement. If $g = \Phi f$ then $f = \Phi^{-1, \star} g$ which is so called the reciprocity principle [4, 8].

In [5], we proposed a projective analysis thanks to a flow of second order partial differential equations. The proofs are nearly the same by replacing a general homography by a homography whose matrix belongs to the registration group. But the results can now be interpreted as being connected to camera motion. Moreover, it can be shown that the AMSS is a good approximation of this projective analysis in the center of the image when the change of the optical axis is small.

3 Projective Invariant Analysis

Let F be the 3D image associated to a planar image f . The projective invariant analysis proposed in [5] consists in finding connections between the affine invariant analysis of images obtained from sections of the 3D homogeneous image F . More precisely we have:

Theorem 1 ([5]). *Let $g = F_P$ be the image obtained by cutting the image F by a plane P . There exists a unique multiscale analysis (T_t^P) , $t \geq 0$, $P \in \mathbf{P}$, where \mathbf{P} is the set of tangent planes to the unit sphere such that:*

1. $\forall P \in \mathbf{P}$, (T_t^P) is the affine invariant analysis of $g = F_P$ proposed by Alvarez, Guichard, Lions, Morel (AMSS) [1].
2. $\forall (P, t)$, $\forall \lambda \neq 0$, $T_t^P F(\lambda X, \lambda Y, \lambda Z) = T_t^P F(X, Y, Z)$.
3. For every choice of (P, t) , for every projective transformation φ , with associated matrix Φ , there exist $Q = Q(\Phi, P)$ and $t' = t'(\Phi, t)$ such that

$$\varphi(T_{t'}^Q f) = T_t^P(\varphi f) = T_t^P g$$

with $T_t^P f$ defined by

$$(T_t^P f)(x, y) = (T_t^P F)(X, Y, Z)$$

where (X, Y, Z) are the homogeneous coordinates of $m = (x, y)$ and

$$(T_t^P F)(X, Y, Z) = F(P; t, M)$$

is the solution of the evolution equation

$$\begin{aligned} \frac{\partial F}{\partial t}(t, M) &= t^{\frac{1}{3}} (\overrightarrow{OM} \cdot (\alpha \wedge \beta))^{\frac{4}{3}} [D^2 F(t, X)(\mathbf{U}, \mathbf{U})]^{\frac{1}{3}} \\ F(0, M) &= F(X, Y, Z) = F(M) \end{aligned}$$

with

$$\mathbf{U} = (\beta \cdot DF(t, M))\alpha - (\alpha \cdot DF(t, M))\beta$$

and (α, β) an orthonormal basis of P .

The proof consists in writing the affine analysis of $g = F_P$ and the relations between the gradient DF_P and the Hessian matrix $D^2 F_P$ of the planar image $g = F_P$ and their F counterparts. Then to ensure that the evolution of the 3D image preserves its homogeneity.

3.1 The 2D Projective Analysis of f and the Registration Group

A rewriting of $[D^2 F(t, M)(\mathbf{U}, \mathbf{U})]$ shows that the analysis of F with parameter P is

$$\frac{\partial F}{\partial t}(t, M) = t^{\frac{1}{3}} \frac{(\overrightarrow{OM} \cdot (\alpha \wedge \beta))^2}{Z^2} Z^{\frac{4}{3}} [D^2 F(t, M)(k, k)]^{\frac{1}{3}}$$

$$F(0, M) = F(M) .$$

Then, if $(T_t^P f)(m) = (T_t^P f)(x, y) = f^P(t; x, y) = F(P; t, x, y, 1)$, the 2D analysis of f with parameter P is given by

$$\frac{\partial f^P}{\partial t}(m) = (t)^{\frac{1}{3}} (ax + by + c)^2 \|Df^P(m)\| (\text{curv} f^P(m))^{\frac{1}{3}}$$

$$f^P(0; m) = f(m)$$

where (a, b, c) are the coordinates of $\alpha \wedge \beta$, i.e. $u = (a, b, c)$ is a unit normal vector to P .

Let now Φ be a matrix of the registration group $\Phi = RH$ and φ the projective application associated to Φ . Direct calculations from [5] show that

$$(PI) \quad \varphi(T_{t'}^Q f) = T_t^P(\varphi f)$$

with $Q = R(P)$ and $t' = \frac{t}{|\det H|} = \frac{t}{d(O, \Phi(P))}$, where $\Phi(P)$ is the image of the affine plane P by the 3D affine application Φ .

This projective analysis may be simplified for the special orthogonal group \mathcal{O}^+ and used for shape recognition [6].

Let $\Phi = RH = R_{\Delta, \alpha} R_{\beta}^k$ be the decomposition of an element of the registration group, φ the projective transformation associated to Φ and P the plane $Z = 1$. Then an orthonormal vector to the plane $Q = R(P) = R_{\Delta, \alpha}(P)$ is

$R_{\Delta, \alpha}(k) = (\sin \theta \sin \alpha, -\cos \theta \sin \alpha, \cos \alpha)$ and the projective invariance (PI) leads to

$$\begin{aligned} & ((\sin \theta \sin \alpha)x' - (\cos \theta \sin \alpha)y' + \cos \alpha)^2 \|Df(x', y')\| (\text{curv} f(x', y'))^{\frac{1}{3}} \\ &= |\det H|^{\frac{4}{3}} \|D\varphi f(x, y)\| (\text{curv} \varphi f(x, y))^{\frac{1}{3}} . \end{aligned}$$

with

$$\varphi(x, y) = (x', y').$$

Remark 3. As in AMSS, $|\det H|^{\frac{4}{3}}$ is a scale factor. The constants in the equation (PI) are directly connected to the movement of the camera.

Moreover it can be shown, that for α small (i.e. a small change of the optical axis)

$$\left\{ \begin{array}{l} u = x' - x \approx -Cx + A - \beta y - \alpha x(y \cos \theta - x \sin \theta) + \alpha \sin \theta \\ v = y' - y \approx -Cy + B + \beta x - \alpha y(y \cos \theta - x \sin \theta) - \alpha \cos \theta \end{array} \right\}$$

Figure 3 is a simulation of the optical flow (u, v) generated by the transformation φ for $\theta = 3\pi/4, \alpha = 5 \cdot 10^{-4}, \beta = 5.1 \cdot 10^{-2}, A = 3, B = 2, C = -0.02$. The “pure” projective deformation due to the rotation $R_{\Delta, \alpha}$ is essentially visible on the border of

the image. In the center of the image the projective invariance may be approximated by the AMSS. In [4,8], the registration group was associated to a matching problem.

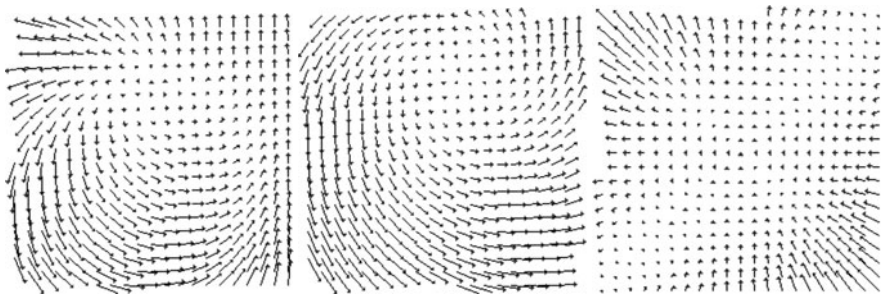


Fig. 3 Decomposition of the optical flow. From *left to right* : the complete flow, the flow due to the similarity, the flow due to $R_{\Delta, \alpha}$

Recently, Morel and Yu have proposed an Affine Scale Invariant Feature Tracking [15]. The aim of the last section is to revisit the ASIFT algorithm and, thanks to the registration group, propose a first step for a fully projective invariant feature tracking.

4 From ASIFT to PSIFT, a First Approach

Let us recall the standard decomposition of an element Φ of the registration group

$$\Phi = RH = R_{\Delta, \alpha} R_{\beta}^k H = R_{\Delta, \alpha} S$$

where $R_{\Delta, \alpha}$ is a rotation with axis in the plane (O, i, j) such that $\widehat{(i, \Delta)} = \theta$ and with angle α . S induces a deformation of the image through a direct similarity.

This approach allows a simple algorithm for the matches where, after elimination of the pure projective deformation (2 parameters), the matching may be reduced to finding a similarity (4 parameters).

Let f and g be two images. First, thanks to a well chosen discretization, two sequences (f_i) and (g_i) are built

$$f_i = R_{\Delta_i, \alpha_i} f, \quad g_i = R_{\Delta_i, \alpha_i} g.$$

Then all pairs (f_j, g_k) of images are compared. If f and g are two images of the same scene under different points of view, the modeling in the registration group allows to say that there exist (j, k) and a similarity S such that $f_j = Sg_k$.

This allows to propose a slight modification to the ASIFT algorithm in order to obtain a projective invariant image comparison. This will be done in Sect. 4.2.

Section 4.1 is devoted to the analysis of the matrix decomposition theorem of Morel and Yu and shows how this decomposition may be the result of an affine approximation of the projective matching.

4.1 The Matrix Decomposition Theorem of Morel and Yu

One of the main ideas of the ASIFT is the decomposition of an affine map with strictly positive determinant which is not a similarity [15]. Let us recall this theorem with the registration group notations.

Theorem 2 ([15]). *Any affine map $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with strictly positive determinant which is not a similarity may be written as*

$$A = \lambda \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \frac{1}{\cos \alpha} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix}.$$

Now, take $\Phi \in \mathcal{G}$, its standard decomposition introduced in Sect. 2.1 writes

$$\Phi = R_{\Delta, \alpha} R_{\beta}^k H$$

with

$$R_{\Delta, \alpha} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us denote R_{ASIFT} the 3D rotation

$$R_{ASIFT} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix},$$

thus

$$R_{ASIFT} = R_{\Delta, \alpha} R_{\theta}^k$$

and

$$\Phi = R_{ASIFT} R_{-\theta}^k R_{\beta}^k H = R_{ASIFT} R_{\psi}^k H = R_{ASIFT} S.$$

The only matrix for which one needs to do some work, in order to find an affine approximation for Φ , is the rotation with axis i , R_{α}^i .

Let f and g two planar images such that $f = R_{\alpha}^i g$, i.e.

$$f(x, y) = g\left(\frac{x}{y \sin \alpha + \cos \alpha}, \frac{y \cos \alpha - \sin \alpha}{y \sin \alpha + \cos \alpha}\right).$$

If f is a frontal view of a 3D scene in the plane ($Z = Z_0$), i.e.

$$f(x, y) = f\left(\frac{X}{Z_0}, \frac{Y}{Z_0}\right) = I(X, Y, Z_0),$$

then

$$I(X, Y, Z_0) = g\left(\frac{X}{Y \sin \alpha + Z_0 \cos \alpha}, \frac{Y \cos \alpha - Z_0 \sin \alpha}{Y \sin \alpha + Z_0 \cos \alpha}\right).$$

If the 3D scene is small ($|X|, |Y|$ small) and far from the camera (Z_0 large) then

$$\begin{aligned} \frac{X}{Y \sin \alpha + Z_0 \cos \alpha} &\sim \frac{X}{Z_0 \cos \alpha} \\ \frac{Y \cos \alpha - Z_0 \sin \alpha}{Y \sin \alpha + Z_0 \cos \alpha} &\sim \frac{Y \cos \alpha - Z_0 \sin \alpha}{Z_0 \cos \alpha}. \end{aligned}$$

Thus, the linear part of the affine approximation of the projective application R_α^i is

$$\frac{1}{Z_0} \begin{pmatrix} \frac{1}{\cos \alpha} & 0 \\ 0 & 1 \end{pmatrix}$$

and the linear part of the affine approximation of Φ may be written as

$$A = \lambda \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \frac{1}{\cos \alpha} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix};$$

which is the exact decomposition proved by Morel and Yu.

This remark allows a first step towards a PSIFT algorithm.

4.2 Towards a PSIFT Algorithm

For the moment, we just follow the work of Morel and Yu and propose a first idea for the algorithm.

1. First, a rotation with angle θ is applied on planar images. This is exactly the transformation

$$f \longrightarrow R_\theta^k f.$$

2. The tilt transformation is replaced by the projective transformation R_α^i

$$R_\theta^k f \longrightarrow R_\alpha^i(R_\theta^k f) = (R_\theta^k R_\alpha^i) f = R_{ASIFT} f.$$

3. The subsampling problem may occur in a different way. Indeed the projective transformation does not preserve parallelism.

For example the square $(A(0,0); B(1,0); C(1,1); D(0,1))$ is transformed in the rectangle $(A; B_1(\sqrt{2}, 0); C_1(\sqrt{2}, 1); D)$ by the affine application

$$\begin{pmatrix} \frac{1}{\cos \alpha} & 0 \\ 0 & 1 \end{pmatrix} \quad \alpha = \frac{\pi}{4}$$

and in the trapezoid $(A_2(0, -1); B_2(\sqrt{2}, -1); C_2(\frac{1}{\sqrt{2}}, 0); D_2(0, 0))$ by the projective application R_α^i . In a first time, we will nevertheless use the same antialiasing filter.

4. These transformations will be made on the two images with a suitable discretization of α and θ . Thus we obtain two sequences (f_n) and (g_n)

$$f_n = (R_{ASIFT})_n f \quad g_n = (R_{ASIFT})_n g$$

5. All simulated images are compared by a similarity invariant matching, for example with the SIFT algorithm [13]. Indeed, the modeling in the registration group allows then to write

$$\exists(j, k) \quad f_j = S g_k$$

where S is a similarity.

5 Conclusion

In this paper we present two ways to detect projective invariant features. The first one proposes differential projective invariants, the second one leads to a Projective Invariant Feature Tracking. Moreover these two approaches may be seen, thanks to a modeling in the registration group, as the natural extensions of the affine approaches [1, 15]. A future work will be to implement a PSIFT algorithm and to measure the obtained improvements.

Acknowledgement: The author thanks Georges Koepfler for valuable conversations on the use of the registration group for projective matching.

References

1. Alvarez, L., Guichard, F., Lions, P.-L., Morel, J.-M.: Axioms and fundamental equations of image processing. *Arch. Ration. Mechan.* **16**, 200–257 (1993)
2. Bruckstein, A., Shaked, D.: On projective invariant smoothing and evolutions of planar curves and polygons. *J. Math. Imaging Vis.* **7**(3), 181–192 (1998)
3. Cohignac, T., Lopez, C., More, J.-M.: Integral and local affine invariant parameter and application to shape recognition, *ICPR*, pp. 164–168. (1994)
4. Dibos, F.: Du groupe projectif au groupe des recalages: une nouvelle modélisation. *C.R. Acad. Sci. Paris Série I* **332**(9) (2001)
5. Dibos, F.: Projective analysis of 2D images. *IEEE Trans. Image Process.* **7**(3), 274–279 (1998)
6. Dibos, F., Frosini, P., Pasquignon, D.: The use of size functions for comparison of shapes through differential invariants. *J. Math. Imaging Vis.* **21**, 107–118 (2004)

7. Dibos, F., Jonchery, C., Koepfler, G.: Iterative camera motion and depth estimation in a video sequence. In: *Lecture Notes in Computer Science*, vol. 5702, pp. 1028–1035. *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*. Springer, Berlin (2009)
8. Dibos, F., Koepfler, G., Monasse, P.: Image alignment. In: *Geometric Level Set Methods in Imaging*. Springer, Heidelberg (2003)
9. Faugeras, O.: Sur l'évolution de courbes simples du plan projectif réel. *C.R. Acad. Sci. Paris Série I* **317**, 565–570 (1993)
10. Faugeras, O., Keriven, R.: Some recent results on the projective evolution of 2D curves. In: *Proceedings ICIP*, vol. 3, pp. 13–16. Washington (1995)
11. Jonchery, C., Dibos, F., Koepfler, G.: Camera motion estimation through planar deformation determination. *J. Math. Imaging Vis.* **32**(1) (2008)
12. Koepfler, G., Moisan, L.: Geometric Multiscale representation of numerical images. In: *Proceedings of the second International Conference on Scale Space Theories in Computer Vision*, Springer *Lecture Notes in Computer Science*, vol. 1682, pp. 339–350 (1999)
13. Lowe, D.: Distinctive image features from scale-invariant key points. *Int. J. Comput. Vis.* **60**, 91–110 (2004)
14. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. *Int. J. Comput. Vis.* **60**(1), 63–86 (2004)
15. Morel, J.-M., Yu, G.: ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Imaging Sci.* **2**(2), 438–469 (2009)
16. Muse, P., Sur, F., Cao, F., Gousseau, Y., Morel, J.-M.: An a contrario decision method for shape element recognition. *Int. J. Comput. Vis.* **69**, 295–315 (2006)
17. Sapiro, G., Tannenbaum, A.: Affine invariant scale space. *Int. J. Comput. Vis.* **11**, 25–44 (1993)
18. Sur, F.: Invariant image descriptors and affine morphological scale-space. *Rapport de Recherche*, no 6250, INRIA Lorraine, 2007

Morphological Component Analysis for Decomposing Dynamic Textures

Sloven Dubois, Renaud Péteri, and Michel Ménard

Abstract The research context of this work is dynamic texture analysis and characterization. A dynamic texture can be described as a time-varying phenomenon with a certain repetitiveness in both space and time.

Many dynamic textures can be modeled as a large scale propagating wavefront and local oscillating phenomena.

The Morphological Component Analysis approach with a well chosen dictionary is used to retrieve the components of dynamic textures. We define two new strategies for adaptive thresholding in the Morphological Component Analysis framework, which greatly reduce the computation time when applied on videos. These strategies are studied with different criteria. Finally, tests on real image sequences illustrate the efficiency of the proposed method.

1 Introduction

The study of dynamic textures, or temporal textures, is a recent research topic in the field of video processing. A dynamic texture can be described as a time-varying phenomenon with a certain repetitiveness in both space and time. A flag in the wind,

Sloven Dubois (✉)

Laboratoire Mathématiques, Image et Applications (MIA) et Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France
e-mail: sloven.dubois01@univ-lr.fr

Renaud Péteri

Laboratoire Mathématiques, Image et Applications (MIA), Avenue Michel Crépeau, 17042 La Rochelle, France
e-mail: renaud.peteri@univ-lr.fr

Michel Ménard

Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France
e-mail: michel.menard@univ-lr.fr

ripples at the surface of water, smoke or an escalator are all examples of dynamic textures. Figure 1 shows other examples of dynamic textures.

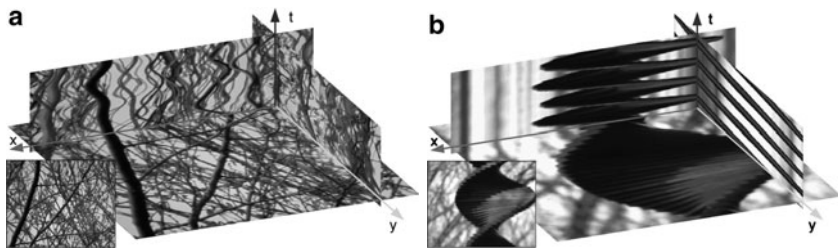


Fig. 1 2D+T slices of two dynamic textures. Here, a dynamic texture is seen as a data cube and is cut at pixel $O(x, y, t)$, giving three planes $(\vec{x}O\vec{y})$, $(\vec{x}O\vec{t})$ and $(\vec{y}O\vec{t})$

Without being exhaustive, the study of DT is an active topic, with many research areas such as synthesis [5], characterization [11, 12, 14] or segmentation [6, 8].

The context of our works is the characterization and the analysis of these dynamic textures, with the aim of being able to automatically retrieve video scenes with given dynamic textures [7].

Giving a proper definition of dynamic textures is a notoriously difficult problem. Dynamic textures are often defined as phenomena varying in both space and time with a certain spatio-temporal repetitiveness. They cannot only be considered as a simple extension of static textures to the time domain, but as a more complex phenomenon resulting from several dynamics. Each dynamic texture has its own characteristics, such as stationarity, regularity, repetitiveness, propagation speed, . . .

These characteristics are more or less difficult to extract depending on the complexity of the considered dynamic texture. For instance on Fig. 2a showing an image sequence of sea waves, two motions can be observed: the high-frequency motion of small waves (2), carried by the overall motion of the internal wave (1). It gets more complex when the two phenomena overlap with each other (3). This statement can also be made about an image sequence of trees on Fig. 2b.

Many dynamic textures can be decomposed into one or several local oscillating motions carried by far range waves. In order to better characterize these two sets of components, it is necessary to extract them separately.

In this article, the Morphological Component Analysis is used for decomposing and analyzing image sequences of natural scenes. To our knowledge, the only existing work using Morphological Component Analysis and video is recent and focuses on the inpainting of a cartoon sequence [17].

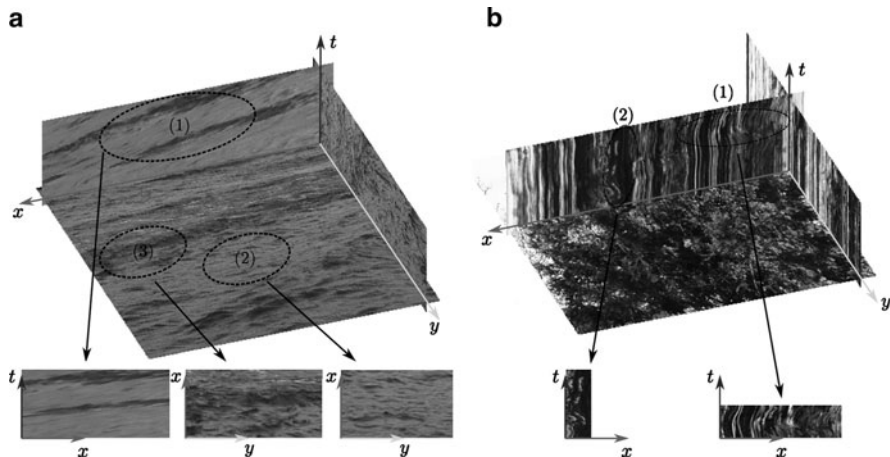


Fig. 2 2D+T slices of two dynamic textures: local oscillating phenomenon (2) and long range propagating wave (1) and a mixture of both of them (3)

In Sect. 2, the Morphological Component Analysis is briefly described. The dictionaries selected in the Morphological Component Analysis, adapted to dynamic textures, are presented.

A key issue is the computation time of Morphological Component Analysis that is related to the thresholding strategy. We propose in Sect. 3 two new adaptive thresholding strategies that reduced the computation time by a factor of four compared to the original algorithm. To evaluate these new strategies, different criteria are proposed.

Results on real sequences of dynamic textures are presented in Sect. 4 and future prospects are finally discussed.

2 Decomposing a Dynamic Texture Using Morphological Component Analysis

In order to perform correct characterization, it is necessary to well understand the nature of dynamic textures. In many works, a dynamic textures is often described as a time-varying phenomenon with a certain repetitiveness in both space and time. According to researches on synthesis [10] and observations made on a large dynamic textures database [13], a dynamic texture can be modelled as a sum of local oscillations carried by longer range waves.

Recent works for decomposing images and videos [1, 4, 16] seem relevant for extracting these components. We have chosen the Morphological Component Analysis because of the richness of the available dictionary, which is crucial, considering the complexity of dynamic textures.

2.1 Morphological Component Analysis Approach

The Morphological Component Analysis (*MCA*) approach finds an acceptable solution to the inverse problem of decomposing a signal onto a given vectorial basis, i.e. to extract components $(y_i)_{i=1,\dots,N}$ from a degraded observation y according to a sparsity constraint. This is obviously an ill-posed inverse problem.

The *MCA* approach assumes that each component y_i can be represented sparsely in the associated basis Φ_i :

$$\forall i = 1, \dots, N, \quad y_i = \Phi_i \alpha_i \quad \text{and} \quad \alpha_i = \Phi_i^T y_i \quad (1)$$

where α_i are the projection coefficients of y_i on basis Φ_i .

In this way, the obtained dictionary is composed of atoms built by associating several transforms $\Phi = [\Phi_1, \dots, \Phi_N]$ such that, for each i , y_i is well represented (sparse) in Φ_i and is not, or at least not as well, represented in Φ_j ($j \neq i$).

This induces that:

$$\forall i, j \neq i \quad \|\Phi_i^T y_i\|_0 < \|\Phi_j^T y_i\|_0 \quad (2)$$

$\|\dots\|_0$ being the pseudo-norm ℓ_0 (number of non-zero coefficients).

The choice of the basis is of course crucial. Each transform possesses its own characteristics and will be adapted to extract a particular phenomenon. This choice will be discussed in the next section.

Solving (2) implies to find a solution to the equation: $y = \Phi \alpha$. Starck et al. propose a solution for it in [16] and [15] by finding morphological components $(y_i)_{i=1,\dots,N}$ with the following optimization problem:

$$\min_{y_1, \dots, y_N} \sum_{i=1}^N \|\Phi_i^T y_i\|_p^p \quad \text{such that} \quad \left\| y - \sum_{i=1}^N y_i \right\|_2 \leq \sigma \quad (3)$$

where $\|\Phi_i^T y_i\|_p^p$ penalizes non-sparse solutions (usually $0 \leq p \leq 1$). σ is the noise standard deviation.

This optimization problem (3) is not easy to solve.

If all components y_j except the i th are fixed till iteration $k-1$, it is however proved that the solution $\alpha_i^{(k)}$ is given by hard thresholding the marginal residual $r_i^{(k)} = y - \sum_{j \neq i} y_j^{(k-1)}$:

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \left(\Phi_i^T \left(r_i^{(k)} \right) \right) \quad (4)$$

$\delta_{\lambda^{(k)}}$ being the thresholding function for threshold $\lambda^{(k)}$ at step k . These marginal residuals r_i are by construction likely to contain missing informations of y_i . This idea induces an iterative algorithm for thresholding the marginal residuals, the main steps of which are presented in Algorithm 1.

Algorithm 1 Morphological Component Analysis**Task** : Decompose a nD signal in dictionary Φ .**Parameters** :

- The signal y to be decomposed
- The dictionary $\Phi = [\Phi_1, \dots, \Phi_K]$
- The thresholding strategy strategy
- The stopping condition σ

Initialization :*// Components to be estimated are set to 0***for** $i = 1$ to N **do**

$$\tilde{y}_i^{(0)} = 0$$

end for*// Initialization of λ*

$$\lambda^{(1)} = \text{lambda_initialization}(\text{strategy})$$

// Initialization of the iteration number

$$k = 1$$

Main loop :**while** $\left\| y - \sum_{j=1}^N \tilde{y}_j^{(k-1)} \right\|_2 \leq \sigma$ **do***// For each component***for** $i = 1$ to N **do***// Compute the marginal residual*

$$\tilde{r}_i^{(k)} = y - \sum_{j \neq i} \tilde{y}_j^{(k-1)}$$

// Projection of $\tilde{r}_i^{(k)}$ on basis Φ_i

$$\tilde{\alpha}_i^{(k)} = \Phi_i^T \left(\tilde{r}_i^{(k)} \right)$$

// Hard thresholding of $\tilde{\alpha}_i^{(k)}$

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \tilde{\alpha}_i^{(k)}$$

// New estimation of \tilde{y}_i

$$\tilde{y}_i^{(k)} = \Phi_i \left(\alpha_i^{(k)} \right)$$

end for*// Update of threshold λ*

$$\lambda^{(k+1)} = \text{update}(\lambda^{(k)}, \text{strategy})$$

// Iterate

$$k = k + 1$$

end while

2.2 Choice of the Dictionary

A crucial point in the Morphological Component Analysis approach is the dictionary definition. An unsuitable choice of transformations will lead to non sparse and irrelevant decompositions of the different dynamical phenomena present in the sequence.

As mentioned previously, we model dynamic texture as a sum of local oscillations carried by long range waves. It is therefore necessary to associate each component with the most representative basis.

In [8], the authors show that the curvelet transform [3] is relevant for extracting non-local phenomena propagating temporally. It thus seems particularly interesting to model long range waves present in a dynamic texture.

The second part of the model is composed of locally oscillating phenomena that will be extracted using the local cosine transform.

The dictionary that we use in the Morphological Component Analysis algorithm is then composed of the 2D+T curvelet transform Φ_1 and the 2D+T local cosine transform Φ_2 .

2.3 Thresholding Strategy

The purpose of this work is the decomposition of natural dynamic textures, therefore our experiments have been conducted on sequences from the DynTex database [13], a large database of dynamic textures. The processed sequences have a duration of 5 s (128 images) and a size of 648 by 540 pixels.¹ On volumes of such a size, the computation time is non negligible, as some transforms require several minutes.

Let function $T()$ measures the execution time of a transform Φ_i during one cycle of the algorithm (analysis via Φ_i and synthesis via Φ_i^T). Two different platforms² have been used for the chosen dictionary, giving the computation time presented in Table 1.

Table 1 Computation time required for performing an analysis or a synthesis with the chosen dictionary on two different hardware configurations

	Platform 1 (32 bits)	Platform 2 (64 bits)
$T(\Phi_1) \approx T(\Phi_1^T)$ (s)	≈ 259	≈ 109
$T(\Phi_2) \approx T(\Phi_2^T)$ (s)	≈ 120	≈ 85

A recent work [2] has shown that a hundred of iterations is necessary to establish a good separation of the different components when a linear thresholding strategy (LTS) is used. In our case, the total computational time for a 5 s sequence is given by: $100 * (T(\Phi_1^T) + T(\Phi_1) + T(\Phi_2^T) + T(\Phi_2))$, which corresponds to 21 h on platform 1, and around 11 h on platform 2.

If we extend this result to the entire DynTex database, 605 days of calculation are required on a standard computer. This computation time can be reduced to 309 days on a dedicated server.

Recently, Bobin et al. [2] have proposed a thresholding strategy ‘Mean of Max’ (MoMS) that enables to obtain similar results but with fewer iterations (50 in average instead of 100). It requires a computation time of approximately 15 h45

¹ i.e. more than 44 million voxels.

² **Platform 1** : Processor 32 bits, 2.4 GHz, 4 GB of RAM.

Platform 2 : Processor 64 bits, 3.2 GHz, 24 GB of RAM.

(respectively 6 h on platform 2) for a 5 s video, resulting in approximately 453 days (respectively 232 days) for the whole database.

For indexing the whole DynTex database, the computation time of the MoMS is still acceptable, since it is always possible to divide the workload between several processors. In the case where one searches for a particular texture using a query sequence, these calculations are acceptable only on sequences with limited duration and low resolution. We propose to reduce these limitations by introducing two new thresholding strategies.

3 Two New Thresholding Strategies

Results of the decomposition using the MCA algorithm strongly depend on the evolution of the threshold $\lambda^{(k)}$ in one iteration of the main loop. Figure 3 shows two different evolutions of $\lambda^{(k)}$ corresponding to two strategies (S1) and (S2). Evolution of $\lambda^{(k)}$ is slower in case (S1) than in case (S2). In this example, evolution (S1), respectively (S2), leads to select 5% of the coefficients, respectively 25%, in the two bases. If we consider that evolution (S1) gives here an optimal threshold, a failure to control the value of $\lambda^{(k)}$ (case S2) will lead to a rapid allocation of too many coefficients in the two bases, degrading the final decomposition.

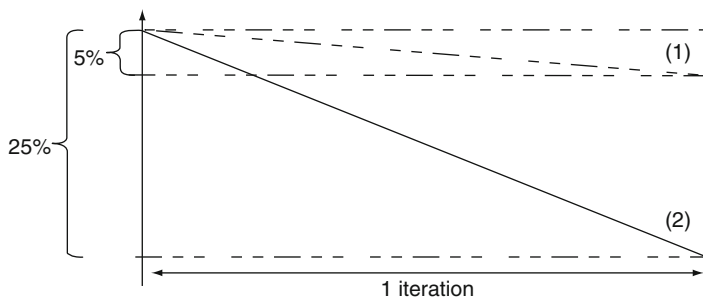


Fig. 3 Two thresholding strategies leading to different evolutions of the threshold value during one iteration of the main loop

The linear thresholding strategy (LTS) leads to the optimum $\lambda^{(k)}$ for 100 iterations [2]. For a large number of natural textures, this number of iterations can be greatly reduced, depending on the texture properties. LTS is then no longer optimum. However, the threshold evolution using LTS can be considered as a minimum slope below which the evolution of $\lambda^{(k)}$ is sub-optimal. A good strategy for computing $\lambda^{(k)}$ should lead to a slope greater than or equal to the one obtained using LTS.

The ‘Mean of Max’ strategy (MoMS) is interesting as it can adaptively change the evolution of $\lambda^{(k)}$. However, on natural texture sequences, this strategy tends to reduce too drastically this slope, or even almost cancel it.

3.1 Adaptive Thresholding Strategy with Linear Correction (ATSLc)

We propose to combine these two strategies into a new so-called adaptive thresholding strategy with linear correction (ATSLc), which defines $\lambda^{(k+1)}$ as the minimum value of $\lambda^{(k+1)}$ calculated using strategies LTS and MoMS.

The $\lambda^{(k)}$ update using ATSLc is formalized as follows:

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100} \right) \quad (5)$$

where:

$$\begin{aligned} m_1 &= \max_{\forall i} \left\| \Phi_i^T r^{(k)} \right\|_{\infty} \\ m_2 &= \max_{\forall j, j \neq i_0} \left\| \Phi_j^T r^{(k)} \right\|_{\infty} \text{ with } i_0 = \operatorname{argmax}_{\forall i} \left\| \Phi_i^T r^{(k)} \right\|_{\infty} \\ r^{(k)} &= y - \sum_{j=1}^K \hat{y}_j^{(k)} \text{ being the total residual} \end{aligned}$$

Using this strategy, we are sure to change the value of $\lambda^{(k+1)}$ corresponding to the steepest slope. In other words, when MoMS leads to values of $\lambda^{(k+1)}$ evolving slowly, $\lambda^{(k+1)}$ follows the LTS $\lambda^{(k+1)} = \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100}$. Otherwise, $\lambda^{(k+1)}$ follows the MoMS, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, reducing the number of main loops in Algorithm 1.

3.2 Adaptive Thresholding Strategy with Exponential Correction (ATSEc)

In some cases, the distribution of the coefficients is concentrated around the origin. This phenomenon can occur for several reasons, for instance an unsuitable choice of the decomposition bases leading to a similar non-sparse representation in the different bases.

In these cases, the LTS is no longer optimum. Close to the origin, the threshold range will indeed be too large compared to the number of coefficients to select in this interval. Figure 4 shows that about 80% of these coefficients are contained in the last interval: they will be all assigned at once, leading to unsuitable decompositions.

To overcome this problem, [15] use a thresholding strategy with exponential decay, ETS. This strategy enables to threshold on a large range of coefficients at

the first iterations of the algorithm, and on small intervals at the last iterations (see Fig. 4). This strategy leads to a better assignment of the coefficients when concentrated around this origin. However, as for the LTS strategy, the number of iterations has to be set to a large value.

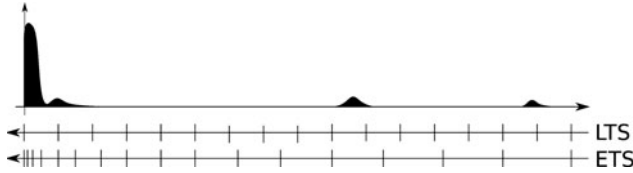


Fig. 4 Threshold intervals for the two strategies LTS and ETS on an illustrative distribution

Similarly to the ATSLc, we propose a second thresholding strategy combining the ETS and the MoMS approaches, so-called adaptive thresholding strategy with exponential correction, ATSEc.

It can be formalized as follows:

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min} \right)^{-\frac{1}{99}} \right) \quad (6)$$

In other words, when MoMS leads to values of $\lambda^{(k+1)}$ evolving too slowly, $\lambda^{(k+1)}$ follows the ETS strategy, $\lambda^{(k+1)} = \lambda^{(k)} * (\lambda^{(1)} - \lambda_{\min})^{-\frac{1}{99}}$. Otherwise, $\lambda^{(k+1)}$ follows the MoMS strategy, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, which also enables to decrease the iteration number in the main loop of Algorithm 1, while significantly decreasing the slope close to the origin.

3.3 Evaluation

In a first step the criterion of the gain in computing time is studied, then in a second step, several criteria for evaluating the decomposition quality are presented.

3.3.1 Computation Time Consideration

In both cases, the number of iterations required for the decomposition of an image sequence of DynTex decreases sharply. Indeed, it takes an average of 12 iterations for ATSLc and of 17 iterations for ATSEc to achieve decomposition.

The gain in terms of computing time is not proportional to the number of iterations removed. Indeed, as in the case of MoMS strategy, ATSLc and ATSEc strategies require an additional projection on all bases to calculate m_1 and m_2 .

The computing time for a sequence of images corresponds to the relationship: (Number of iterations) $\times (2 \times T(\Phi_1) + T(\Phi_1^T) + 2 \times T(\Phi_2^T) + T(\Phi_2))$. The average performance of the two strategies, ATSLc and ATSEc, is estimated in Table 2 on the whole DynTex database. The computation time strategies of the literature are also presented.

Table 2 Estimated computation times required to perform the decomposition using the MCA algorithm for different thresholding strategies

	LTS	ETS	MoMS	ATSLc	ATSEc	
Platform 1	$\approx 21h$	$\approx 21h$	$\approx 15h45$	$\approx 3h45$	$\approx 5h20$	Estimated time for a video
Platform 2	$\approx 10h45$	$\approx 10h45$	$\approx 8h$	$\approx 1h55$	$\approx 2h45$	
Platform 1	$\approx 605d$	$\approx 605d$	$\approx 453d$	$\approx 108d$	$\approx 154d$	Estimated time for the whole DynTex database
Platform 2	$\approx 309d$	$\approx 309d$	$\approx 232d$	$\approx 56d$	$\approx 79d$	

This is the average time estimated on the DynTex database for an image sequence. Moreover, the number of days for complete decomposition of the database is indicated, for both platforms.

In the case of the use of platform 2, about 2 h are needed for the decomposition of an image sequence with a size of $720 \times 576 \times 128$ voxels using the ATSLc strategy. It is a reduction by a factor of 5 of the computing time compared to the original LTS and ETS strategies.

If this result is extended to the whole of DynTex database, it takes about 60 days to perform a decomposition using ATSLc and ATSEc strategies. Moreover, this time can be divided by the number of cores used. In our case, the decomposition of the whole image sequences of dynamic textures, i.e. about 700, was carried out in a week.

In terms of computation time, the gain achieved by using ATSLc and ATSEc strategies seems very promising. An evaluation of the quality of the results is now presented.

3.3.2 ℓ_0 -Norm Criteria

At first, the quality of the decomposition using ATSLc and ATSEc strategies through the ℓ_0 -norm, as compared to LTS and ETS original strategies, is studied.

The number of coefficients selected by the MCA algorithm over the iterations k after the thresholding step is:

$$N_{coef}^{(k)} = \sum_{i=1}^N \left\| \alpha_i^{(k)} \right\|_0 \quad (7)$$

This indicator is used to evaluate the quality of the thresholding strategy. Indeed, for a poorly suited strategy, the number of coefficients selected will be irregular during iterations. This phenomenon is observed through a non-regular growth curve. In other words, during iterations, the more $N_{coef}^{(k)}$ evolves regularly, the more the thresholding strategy can be considered as being good.

On plots of Fig. 5, the evolution of $N_{coef}^{(k)}$ was computed for LTS, ETS, ATSLc and ATSEc strategies for four different sequences; they are displayed here in logarithmic scale. Observations made on these four videos can apply to all the sequences that we have tested (about 200 image sequences):

- The ATSLc and ATSEc strategies have similar behavior during the first iterations and diverge thereafter. During these first iterations, the two strategies choose the maximum slope that is $\frac{1}{2}(m_1 + m_2)$, then each of them behaves differently according to their respective corrections.
- For strategy LTS, most coefficients are selected in the latest iterations of the algorithm (the last 20 approximately). This is observed through the rapid growth curve $N_{coef}^{(k)}$ in the last steps. Thus, this strategy can lead to carrying out many iterations without selecting coefficients, then to a selection that is too fast at the end.
- By observing the growth of $N_{coef}^{(k)}$ for the ATSLc and ATSEc strategies, we can observe that the ATSEc is more regular. Indeed, the number of coefficients selected by the ATSLc strategy increases rapidly in the last iterations, contrary to ATSEc.

Now let us observe, for each basis Φ_i , the error on the choice of coefficients after the decomposition process. To this aim, the selected coefficients for two different threshold strategies are studied and the differences counted.

This criterion is formalized as follows:

$$\forall i, \zeta_i = \frac{1}{M} \left\| \Gamma \left(\alpha_i^{(S1)} \right) - \Gamma \left(\alpha_i^{(S2)} \right) \right\|_0 \quad (8)$$

whereby S1 and S2 are the two strategies in competition, $\alpha_i^{(S1)}$ (respectively $\alpha_i^{(S2)}$) coefficients of the basis Φ_i selected by the MCA algorithm using S1 strategy (respectively S2), M the total number of coefficients α_i , and $\Gamma(A)$ the operator setting the value to 1 if the coefficients are different from 0.

Table 3 presents the means and standard deviations of the errors on the coefficients poorly selected between two competing strategies (for example LTS and ATSLc) for each basis of the dictionary. These data were obtained using 200 videos. ζ_1 (respectively ζ_2) represents the error on the component of the 2D+T curvelet transform (respectively the component of the 2D+T local cosine transform).

We note, as for criterion (8), that:

- Both ATSLc and ATSEc strategies are approaching the solutions obtained by LTS and ETS strategies. Indeed, the greatest average error is 16.66%. However, this

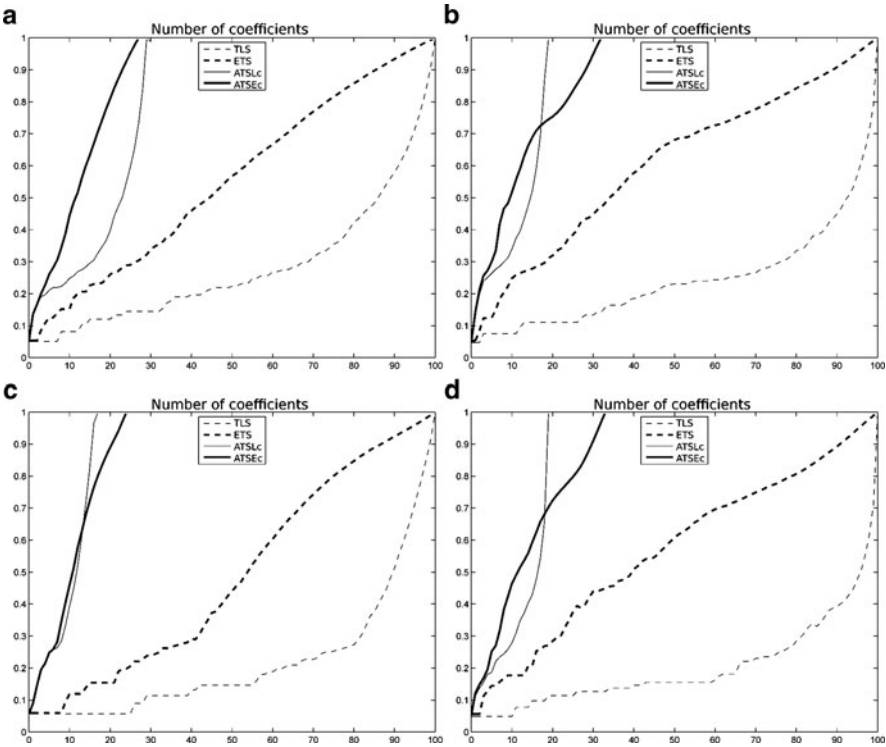


Fig. 5 Study of LTS, ETS, ATSLc and ATSEc strategies. The plots represent the number of thresholded coefficients (logarithmic scale) during the iterations on four different image sequences

Table 3 Means and standard deviations of the error on the coefficients poorly selected of S1 strategy (for example ATSLc) compared to a reference S2 strategy (for example LTS) for each basis of the dictionary (ζ_1 for the 2D+T curvelet transform and ζ_2 for the 2D+T local cosine transform)

	LTS				ETS			
	ζ_1		ζ_2		ζ_1		ζ_2	
	$\mu(\%)$	σ	$\mu(\%)$	σ	$\mu(\%)$	σ	$\mu(\%)$	σ
ATSLc	6.01	3.12	0.24	0.16	15.93	11.12	1.66	1.48
ATSEc	16.66	11.56	0.90	0.70	2.46	1.67	0.98	1.03

- is the comparison between ATSEc and LTS strategies, which is not the easiest to compare. Even if the final components for the two strategies are similar (this will be discussed later), the position of selected coefficients may differ.
- The adaptive thresholding strategy best approaching original strategies is the ATSEc strategy. This is observed through the means and standard deviations of the errors: ζ_1 and ζ_2 which are weak.
 - As observed using criterion (7), ATSLc strategy tends to select quickly many coefficients which can lead to decomposition errors. This is observed by an average error higher than in the case of the use of ATSEc. This strategy implied many

more iterations to divide the coefficients around the origin, thus leading to a small error. These statements can be made for other criteria.

3.3.3 ℓ_2 -Norm Criteria

After studying the influence of strategies on the selection of coefficients, the ℓ_2 -norm reconstruction error of the algorithm over the iterations is computed. It is formalized by:

$$\xi_T^{(k)} = \left\| y - \sum_{i=1}^N \tilde{y}_i^{(k)} \right\|_2 \quad (9)$$

As for criterion (7), the regularity of the plot reflects the performance of the thresholding strategy: the more rapidly and steadily the reconstruction error decreases over the iterations, the more the selected coefficients are relevant and representative of the original signal. On plots of Fig. 6, the evolution of $\xi_T^{(k)}$ was calculated for the LTS, ETS, ATSLc and ATSEc strategies for four image sequences. As for criterion (8), the results obtained on these four videos are representative of all tests (200 videos).

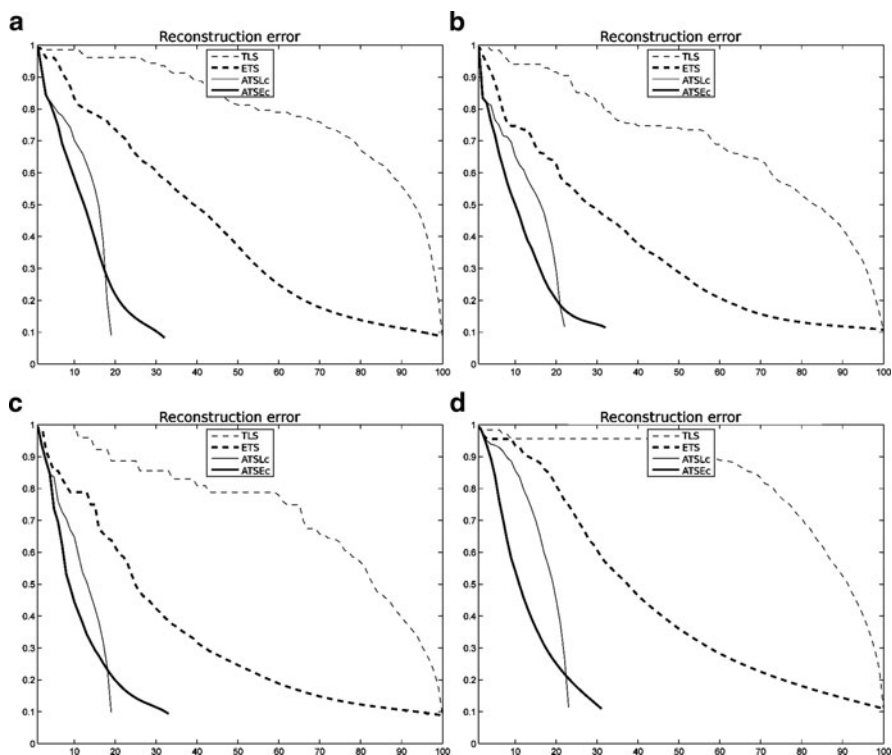


Fig. 6 Study of LTS, ETS, ATSLc and ATSEc strategies. The plots represent the reconstruction error in ℓ_2 -norm during the iterations

Several observations can be made:

- In the case of the LTS strategy, the most representative coefficients of the original signal are selected during the latest iteration of the algorithm. This is observed through the rapid decrease of the plot $\xi_T^{(k)}$.
- The ETS strategy selects many relevant coefficients during the first iterations: a rapid decline of $\xi_T^{(k)}$ is observed. Then it decreases for the remaining coefficients to be allocated. This phenomenon can be explained by the fact that the size ranges of selected coefficients decrease gradually, as a result of the iterations of the MCA algorithm.
- In the case of the two adaptive strategies, ATSLc and ATSEc, the reconstruction error decreases very quickly. Both strategies regularly select pertinent coefficients and quickly approach the original signal.
- The adaptive ATSEc strategy seems more efficient than ATSLc. Indeed, a sudden drop in the reconstruction error is observed in the case of ATSLc. This is because the MCA algorithm selects many relevant coefficients within few iterations. However, the coefficient allocation can be problematic, unlike the ATSEc strategy for which the allocation is made on a larger number of iterations.

The previous study focused on the ℓ_2 -norm error between the reconstruction and the original image sequence. However, we observe that the last iteration of the algorithm (cf. plots on Fig. 6) is identical for all strategies. This is explained by the construction of different strategies: each led the threshold $\lambda^{(k)}$ to a minimum value λ_{min} determined by the user.

Criterion (9) thus makes it possible to observe the performance of strategies over the iterations of the algorithm, it does not make it possible to evaluate the quality of the different components extracted.

For this purpose, a criterion that compares the morphological components obtained by the competing strategies is proposed:

$$\forall i, \xi_i = \frac{1}{\xi_{max}} \left\| \tilde{y}_i^{(S1)} - \tilde{y}_i^{(S2)} \right\|_2 \quad (10)$$

where $\tilde{y}_i^{(S1)}$ (respectively $\tilde{y}_i^{(S2)}$) is the morphological component i estimated with the S1 strategy (respectively S2). ξ_{max} is computed for each component $\tilde{y}_i^{(S2)}$ as the biggest possible mistake we can commit.

Table 4 presents the means and standard deviations of errors in ℓ_2 -norm on each base of the dictionary with respect to the chosen strategies. These results were obtained from 200 image sequences. ξ_1 (respectively ξ_2 and ξ_T) represents the error on the component of the 2D+T curvelet transform (respectively the component of the 2D+T local cosine transform and the reconstruction).

Observations are summarized here:

Table 4 Means and standard deviations of the error in ℓ_2 -norm between two strategies $S1$ (for example ATSLc) and $S2$ (for example LTS) for each base of dictionary (ξ_1 for the 2D+T curvelet transform, ξ_2 for the 2D+T local cosine transform) and its reconstruction ξ_T

	LTS					
	ξ_T		ξ_1		ξ_2	
	$\mu(\%)$	σ	$\mu(\%)$	σ	$\mu(\%)$	σ
ATSLc	0.27	0.10	1.56	1.08	1.56	0.92
ATSEc	0.48	0.20	2.30	1.82	2.18	1.50

	ETS					
	ξ_T		ξ_1		ξ_2	
	$\mu(\%)$	σ	$\mu(\%)$	σ	$\mu(\%)$	σ
ATSLc	0.39	0.15	2.67	1.90	2.43	1.58
ATSEc	0.38	0.17	1.55	1.51	1.50	1.22

- the maximum mean error between reconstructions obtained from two different strategies is only 0.48%. This indicates that, whatever the strategy used, the reconstruction will be almost identical.
- both adaptive strategies ATSLc and ATSEc, fairly approach the decompositions obtained using the original strategies. Indeed, for each component, the maximum average error observed between two different strategies is only 2.67%.

Considering the different criteria studied and the gain in terms of computation time, we can establish that the proposed strategies prove to be relevant for the processing of dynamic texture sequences. In the next section, we present the decomposition results of dynamic textures using the MCA algorithm using the adaptive thresholding strategy with exponential correction.

4 Experiments

4.1 Conditions

Results obtained using ATSLc and ATSEc strategies are promising and satisfactory. In this section, three of them are described in details.³

Decompositions of dynamic textures presented here were obtained using a dictionary composed of the 2D+T curvelet transform and the 2D+T local cosine transform.

The curvelet transform was calculated using 5 scales of decomposition, and 4 angular subdivisions at each change of scale. The local cosine transform was performed on windows of size $32 \times 32 \times 32$ voxels. These decompositions were

³ These three videos and other results are visible at:
http://mia.univ-larochelle.fr/demos/dynamic_textures/

calculated using the adaptive thresholding strategy with exponential correction. As in many experiments [9], we use $\lambda_{min} = \tau\sigma$ with $\tau = 3$.

4.2 Results

The first video shows a duck drifting slowly in a canal (Fig. 7). Reflections of trees in the rippling water and a static texture background are also observable. Figure 7 shows the decomposition results obtained on this video using the MCA algorithm with the ATSEc strategy.

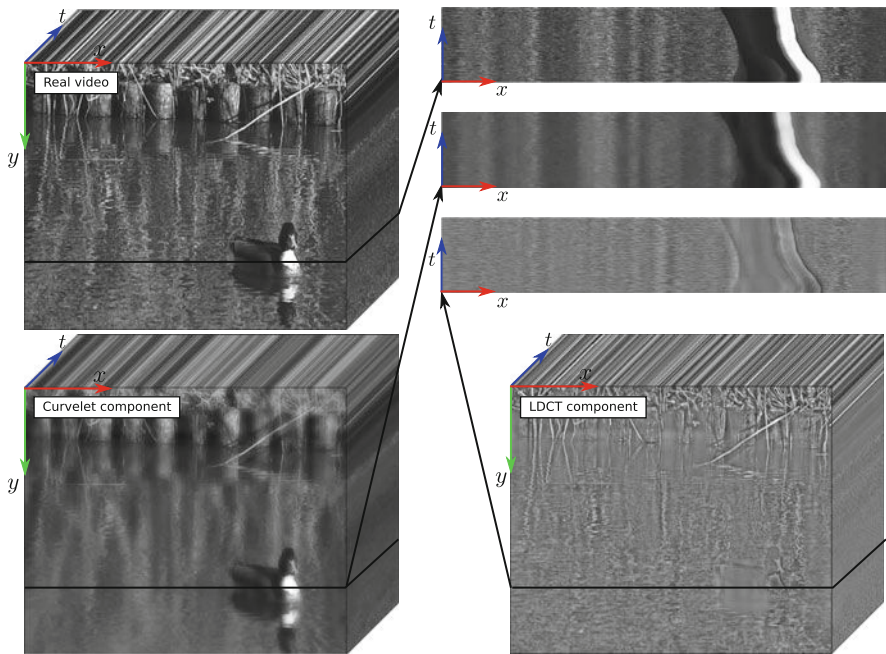


Fig. 7 Results of the MCA decomposition on a video using the ATSEc strategy. Spatio-temporal cuts xt enable to emphasize the temporal aspect of the decomposition

The geometric component is retrieved using the curvelet transform while the texture component is obtained by the local cosine transform: ripples, which are local phenomena, are well captured by the texture component, whereas reflections on the water surface are retrieved in the geometric component.

Spatiotemporal cuts along a xt plane make it possible to visualize the obtained decomposition. They show that the different objects in the scene (the duck, reflections of the trees on water, ...) are correctly considered as geometry. Reflections of the trees are not present in the texture component anymore. One can also observe that oscillations have been strongly attenuated in the geometric component.

This decomposition can bring us information that was not discernible on the original image sequence. For example, the texture of the duck's plumage under its neck can be observed in the texture component, and is not visible in the original video.

The second sequence presented is the surface of the sea near a beach. It consists of a carrier wave (the waves roll) and local phenomena (the foam). Figure 8 shows the decomposition results.

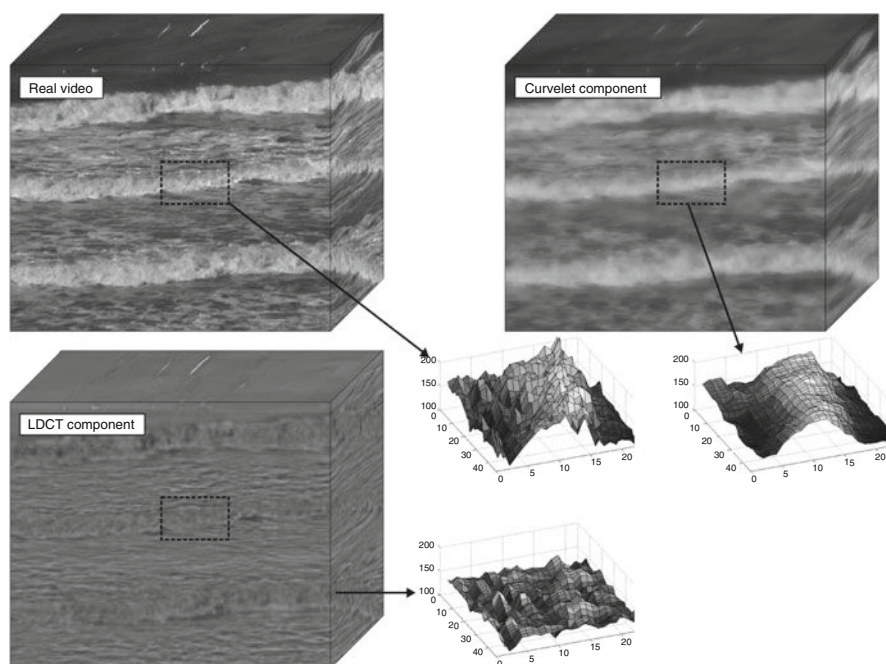


Fig. 8 Decomposition results of a video using the MCA algorithm and the ATSec strategy. Regions of Interest are plotted as surfaces in order to better visualize the algorithm behaviour

The part corresponding to the curvelet transform consists of the roller waves while the local foam and small ripples are captured by the local cosine transform. On areas represented as surfaces, the texture component does not contain the rolls of the waves, completely present in the geometry. The separation between the waves roll and the foam is clearly observable.

The next image sequence represents a fountain (Fig. 9). This fountain is composed of a jet, which once expelled, creates ripples at the water surface. Results of the 2D+T decomposition are shown on Fig. 9.

The two obtained components seem relevant: in the geometric part, the central column of the fountain and the bell shape caused by the jet are visible, whereas almost absent in the texture component.

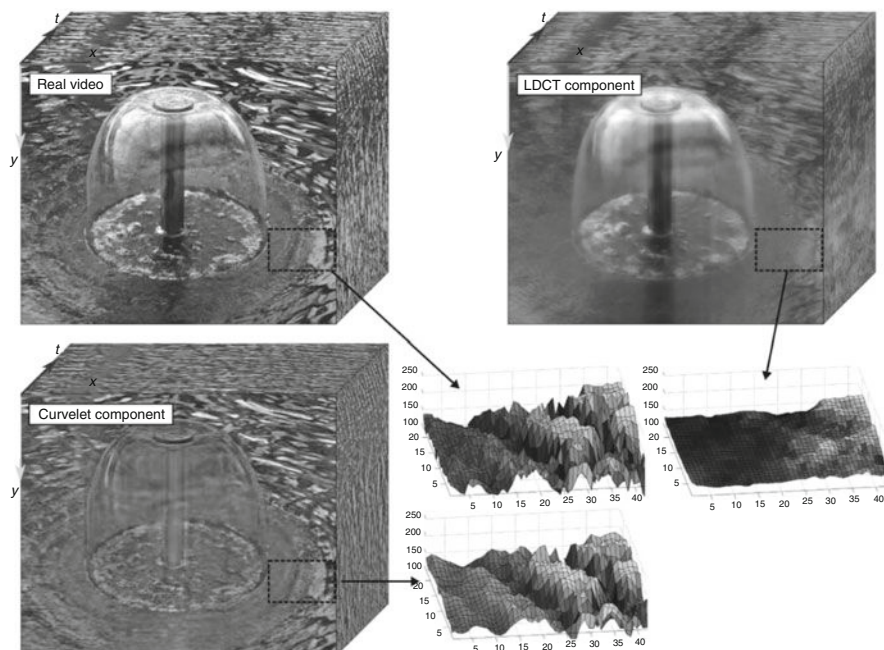


Fig. 9 Decomposition results of a video using the MCA algorithm and the ATSEc strategy. Regions of Interest are plotted as surfaces in order to better visualize the algorithm behaviour

One can also notice that the entire area in front of the water jet is free of ripples, observable contrariwise in the other component. These observations are also well noticeable on areas represented as surfaces where the geometric part is composed of a slight wave free of ripples.

5 Conclusion and Perspectives

This paper deals with the decomposition of dynamic textures in image sequences into different dynamical components. After considering the MCA algorithm and the different possible dictionaries for dynamic texture analysis, we propose two new adaptive thresholding strategies: the ATSLc and the ATSEc. These two new thresholding strategies lead to a significant gain in computation time. Compared to the original strategy, the necessary calculations are reduced by about five times, with equivalent quality of results. In our research context of dynamic texture indexing, it can eliminate the constraints of low resolution and duration on queries in a large video database.

Results on real videos from DynTex have been finally presented. These results confirm the relevance of the proposed model and make it possible to understand the different complex phenomena present in dynamic textures.

Other thresholding strategies are being studied to further improve this computation time. It is particularly necessary to develop strategies that better take into account our proposed model and the features of natural dynamic textures. The extracted components of dynamic textures can later be used as features for video retrieval applications.

In the context of video indexing, the different components obtained using the MCA algorithm can be used for extracting characteristic features: some related to the geometry of the dynamic texture (main motion direction, uniformity of the overall movement, ...) and some characterizing more local phenomena (speed, local vortex, ...).

References

1. Aujol, J.F., Chambolle, A.: Dual norms and image decomposition models. *Comput. Vis.* **63**, 85–104 (2005)
2. Bobin, J., Starck, J.L., Fadili, J.M., Moudden, Y., Donoho, D.L.: Morphological component analysis : An adaptive thresholding strategy. In: *IEEE Transactions on Image Processing*, vol. 16, pp. 2675–2681 (2007)
3. Candès, E., Demanet, L., Donoho, D.L., Ying, L.: Fast Discrete Curvelet Transforms. Tech. Rep. California Institute of Technology, Pasadena, Calif, USA (2005)
4. Chan, T.F., Osher, S., Shen, J.: The digital TV filter and nonlinear denoising. *IEEE Trans. Image Process.* **10**, 231–241 (2001)
5. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. *Int. J. Comput. Vis.* **51**, 91–109 (2003)
6. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: *IEEE International Conference on Computer Vision (ICCV 03)*, pp. 1236–1242, Beijing, China (2003)
7. Dubois, S., Péteri, R., Ménard, M.: A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In: *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 09)*, pp. 314–321, Povia de Varzim, Portugal (2009)
8. Dubois, S., Péteri, R., Ménard, M.: A 3D discrete curvelet based method for segmenting dynamic textures, In: *International Conference on Image Processing (ICIP 09)*, pp. 1373–1376, Cairo, Egypt (2009)
9. Fadili, J.M., Starck, J.L., Elad, M., Donoho, D.L.: MCALab: Reproducible research in signal and image decomposition and inpainting. *IEEE Comput. Sci. Eng.* **12**, 44–63 (2010)
10. Finch, M.: In: Fernando, R. (ed.) *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*, Chap. 1. http://http.developer.nvidia.com/GPUGems/gpugems_part01.html (2004)
11. Nelson, R.C., Polana, R.: Qualitative recognition of motion using temporal texture. *Comput. Vis. Image Underst.* **56**, 78–89 (1992)
12. Péteri, R., Chetverikov, D.: Qualitative characterization of dynamic textures for video retrieval. In: *International Conference on Computer Vision and Graphics (ICCVG 04)*, pp. 33–38, Warsaw, Poland (2004)
13. Péteri, R., Fazekas, S., Huiskes, M.J.: DynTex: A comprehensive database of dynamic textures. *Pattern Recognit. Lett.* **31**, 1627–1632 (2010)

14. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR 01), pp. 58–63, Kauai, USA (2001)
15. Starck, J.L., Elad, M., Donoho, D.L.: Image Decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Process.* **14**, 1570–1582 (2005)
16. Starck, J.L., Elad, M., Donoho, D.L.: Redundant multiscale transforms and their application for morphological component analysis. *Adv. Imaging Electron Phys.* **132** (2004)
17. Woielle, A., Starck, J.L., Fadili, J.M.: Inpainting with 3D sparse transforms. In: 22ème édition du colloque GRETSI, Dijon, France (2009)

Texture Enhancing Based on Variational Image Decomposition

Florian Frühauf, Carsten Pontow, and Otmar Scherzer

Abstract In this paper we consider the Augmented Lagrangian Method for image decomposition. We propose a method which decomposes an image into texture, which is characterized to have finite l^1 curvelet coefficients, a cartoon part, which has finite total variation norm, and noise and oscillating patterns, which have finite G -norm. In the second part of the paper we utilize the equivalence of the Augmented Lagrangian Method and the iterative Bregman distance regularization to show that the dual variables can be used for enhancing of particular components. We concentrate on the enhancing feature for the texture and propose two different variants of the Augmented Lagrangian Method for decomposition and enhancing.

Keywords: Image decomposition · Image enhancement · Anisotropic diffusion · texture · Curvelets · Total variation

Florian Frühauf
Hauptstrasse 1c/1, 6074 Rinn, Austria
e-mail: florian.fruehauf@uibk.ac.at

Carsten Pontow
Computational Science Center, University of Vienna, Nordbergstr. 15, 1090 Vienna, Austria
e-mail: carsten.pontow@umit.at

Otmar Scherzer (✉)
Radon Institute of Computational and Mathematics, Austrian Academy of Sciences,
Altenberger Str. 69, 4040 Linz, Austria
and
Computational Science Center, University of Vienna, Nordbergstr. 15, 1090 Vienna, Austria
e-mail: otmar.scherzer@uibk.ac.at

1 Introduction

The problem of simultaneous reduction of noise and enhancement of important information in image data is an active research area in image analysis. The motivation for this work comes from specifying texture as important image feature. We base our work on recently developed variational *image decomposition* models, which separate the image f into independent components.

Given image data f , *variational* decomposition models developed from *variational denoising* methods, are written in the form

$$\arg \min \{ \Phi_1(v_1) + \alpha_2 \Phi_2(v_2) : f = v_1 + v_2 \}. \quad (1)$$

Here Φ_1 and Φ_2 are appropriate functionals and $\alpha_2 > 0$ is a carefully chosen tuning parameter. The paradigm of such a method is the Rudin–Osher–Fatemi (ROF) model [12] which consists in calculating the minimizer of

$$\arg \min \left\{ J(v_1) + \alpha_2 \int_{\Omega} |v_2|^2 : f = v_1 + v_2 \right\}, \quad (2)$$

where $J(v_1)$ is the total variation of v_1 on Ω . Thus the ROF model is of the form (1) with $\Phi_1 = J$ and with Φ_2 the squared distance of the L^2 -norm. Typically, the minimizer \hat{v}_1 of the ROF functional reveals a blocky structure and looks like a cartoon image, and this is considered the essential image feature of f . Consequently $\hat{v}_2 := f - \hat{v}_1$ is considered noise.

In the recent years research has been devoted to further extract *fine* structures out of the image f (see [1, 2, 5, 7, 9, 11, 15] to mention but a few). This leads to *exact decomposition* models at several scales:

$$\begin{aligned} & \arg \min \mathcal{D}(v_1, \dots, v_n) \text{ with} \\ \mathcal{D}(v_1, \dots, v_n) &:= \left\{ \Phi_1(v_1) + \sum_{i=2}^n \alpha_i \Phi_i(v_i) : f = \sum_{i=1}^n v_i \right\}. \end{aligned} \quad (3)$$

In the terminology of optimization, this is a hard constraint. We also consider soft constraints, in which case we introduce a functional Φ_0 , typically the L^2 or Euclidean distance, and minimize the following unconstrained optimization problem:

$$\begin{aligned} & \arg \min \mathcal{R}(v_1, \dots, v_n) \text{ with} \\ \mathcal{R}(v_1, \dots, v_n) &:= \left\{ \Phi_0 \left(f - \sum_{i=1}^n v_i \right) + \sum_{i=1}^n \alpha_i \Phi_i(v_i) \right\}. \end{aligned} \quad (4)$$

Image regularization and decomposition, in general, use sophisticated convex functionals Φ_i , $i = 1, 2, \dots, n$.

For instance, an exact decomposition model with three components (i.e., $n = 3$) has been considered in [2]. There the functionals Φ_i , $i = 1, 2, 3$ used for

decomposition are the total variation semi-norm, the G -norm and the dual of a Besov-space norm, respectively. Y. Meyer [9], who introduced the G -norm to the image analysis community, characterized functions with finite G -norm as *texture*. For instance, oscillating functions have a finite G -norm and are paradigms of texture. However, also noise can have finite G -norm, and thus functions with finite G -norm contain oscillating patterns as well as noise. This becomes evident from our computations where the middle left image of Figs. 2 and 3, respectively, show oscillating structures from the silhouette of the Zebra and noise. We aim for further differentiating texture and noise components. In particular curvelets very well represent structures and patterns at multiple scales and in different orientations, and therefore are very well suited for making a further distinction of functions of finite G -norm.

In our work we investigate an exact decomposition into a cartoon part (using the total variation semi-norm), a texture component (which has a finite norm on a curvelet space) and a noise component (meaning finite E -norm, which has also been considered in [9] and is equivalent to the G -norm). The motivation for this work is to consider curvelet components as texture, because they characterize very well anisotropic structures of different lengths, which are inherent in many images (cf. [3, 8]). At this point we mention that already in [14] curvelet based functionals have been used, however, there to characterize the cartoon part.

The second part of paper is concerned with texture enhancing based on decomposition models. We consider a two step approach with several alternatives:

1. The first one is the *Augmented Lagrangian Method* (ALM), for finding a decomposition $(\hat{v}_i)_{i=1}^n$, i.e.,

$$f = \sum_{i=1}^n \hat{v}_i,$$

which minimizes the functional

$$(v_1, \dots, v_n) \rightarrow \Phi_1(v_1) + \sum_{i=2}^n \alpha_i \Phi_i(v_i).$$

In comparison with standard implementations of ALM, we apply alternating direction minimization with respect to particular components. In such a way it is possible to enhance for instance the texture component. Nowadays, ALM is also called iterative Bregman distance regularization [10], and in this modern terminology the algorithm is formulated below, because it makes transparent the enhancing features of the proposed algorithm. To our best knowledge, Bregman distance regularization appears first in [4]. As it was shown in [13], Bregman distance regularization with respect to the texture, which we always denote by v_2 in this paper, enhances with respect to the dual variable. In fact Bregman distance regularization contains an explicit step backward in time of a generalized diffusion process, which is a standard enhancing procedure.

2. The work of [14] has been picked up by [8] and iterative Bregman regularization has been used to solve for the compressed sensing problem. The focus of

our paper is different, however, because we focus on enhancing with respect to texture components. For enhancing we use the dual variables $\zeta_2^{(k)}$ of ALM for some iteration index k . In fact $\zeta_2^{(k)}$ is an element of the subgradient of $C_{\mathbf{a}}(v_2^{(k)})$. The term will be explained in detail in the algorithm. This approach provides a texture enhancement strategy:

$$v_2^{\text{enh}} = v_2^{(k)} + \tau \zeta_2^{(k)}, \quad (5)$$

where τ is chosen significantly larger than 2.

The core of the new enhancing procedure is the decomposition model, which is outlined below in detail.

2 Image Decomposition Model

In the following we consider a family of image decomposition models in a discrete setting. The notation and background information can be found in Sect. 5.

Consider a discrete image $f \in X$, the proposed image decomposition method is to calculate a minimizer of

$$\mathbf{J}(v_1, v_2, v_3) = J(v_1) + \alpha_2 C_{\mathbf{a}}(v_2) + \alpha_3 B^*(v_3/\delta), \quad (6)$$

subject to the constraint

$$f = \sum_{i=1}^3 v_i. \quad (7)$$

Here,

1. $\|\cdot\|_X$ is the Euclidean norm of the intensities of the noise part.
2. $J(v_1)$ is the discrete total variation of v_1 (cf. (15)).
3. $C_{\mathbf{a}}(v_2)$ is the weighted curvelet transform (cf. (20)).
4. B^* is an approximation of the dual-norm of the Besov space norm $\dot{B}_{1,1}^1$ (cf. (19)).

For evaluating the dual of the Besov space norm $B^*(v_3/\delta)$ and the curvelet functional $C_{\mathbf{a}}(v_2)$, respectively, we consider v_2 and v_3 as piecewise constant functions on the pixels and assume that they are extended by zero outside of Ω .

Note that in our setting the choice of the parameter α_3 is irrelevant, because B^* only attains the two values zero and infinity. Thus, without loss of generality, α_3 is set to one in the rest of the paper.

For the solution of the constrained minimization problem, we consider *two* variants of ALM, which are both *alternating direction algorithms*. To describe these two methods at once, we introduce a parameter β , which is 0 for the first method (denoted by ALM1) and 1 for the second method (denoted by ALM2).

- Initialize $k = 0$, $v_i^{(0)} = 0$, $i = 1, 2, 3$, $\zeta^{(0)} = 0$. Moreover, choose a sequence of positive parameters $(\tau^{(k)})_k$.¹
- Assign $k \rightarrow k + 1$. Check for an appropriate stopping criterion.
 - Given $v_i^{(k-1)}$, $i = 0, 1, 2, 3$.
 - Calculate

$$v_3^{(k)} := \arg \min_{v_3} \left\{ \frac{\tau^{(k)}}{2} \|f - v_1^{(k-1)} - v_2^{(k-1)} - v_3\|_X^2 + B^*(v_3/\delta) \right\}. \quad (8)$$

- Calculate

$$v_1^{(k)} := \arg \min_{v_1} \left\{ \frac{\tau^{(k)}}{2} \|f - v_1 - v_2^{(k-1)} - v_3^{(k)}\|_X^2 + DJ(v_1, v_1^{(k-1)}) \right\} \quad (9)$$

where we use the term

$$DJ(v_1, v_1^{(k-1)}) = J(v_1) - \beta \left(J(v_1^{(k-1)}) + \left(\zeta_1^{(k-1)}, v_1 - v_1^{(k-1)} \right) \right). \quad (10)$$

Here $\zeta_1^{(k-1)}$ denotes an element of the subgradient of J at $v_1^{(k-1)}$. That is, for $\beta = 1$, $DJ(v_1, v_1^{(k-1)})$ is the Bregman distance of the total variation semi-norm and for $\beta = 0$ it is the total variation semi-norm.

- Calculate

$$v_2^{(k)} := \arg \min_{v_2} \left\{ \frac{\tau^{(k)}}{2} \|f - v_1^{(k)} - v_2 - v_3^{(k)}\|_X^2 + \alpha_2 DC_{\mathbf{a}}(v_2, v_2^{(k-1)}) \right\}, \quad (11)$$

where

$$DC_{\mathbf{a}}(v_2, v_2^{(k-1)}) = C_{\mathbf{a}}(v_2) - C_{\mathbf{a}}(v_2^{(k-1)}) - \left(\zeta_2^{(k-1)}, v_2 - v_2^{(k-1)} \right) \quad (12)$$

denotes the Bregman distance of $C_{\mathbf{a}}$ and $\zeta_2^{(k-1)}$ denotes an element of the subgradient of $C_{\mathbf{a}}$ at $v_2^{(k-1)}$.

This step is identical for both ALM1 and ALM2.

- For ALM2 ($\beta = 1$), motivated from the classical Augmented Lagrangian method, we use the following update of the dual variable

$$\begin{aligned} \zeta_1^{(k)} &= \zeta_1^{(k-1)} + \tau^{(k)} (f - v_1^{(k)} - v_2^{(k-1)} - v_3^{(k)}), \\ \zeta_2^{(k)} &= \zeta_2^{(k-1)} + \tau^{(k)} (f - v_1^{(k)} - v_2^{(k)} - v_3^{(k)}). \end{aligned} \quad (13)$$

For ALM1 ($\beta = 0$), we use only the update for $\zeta_2^{(k)}$.

¹ In general one has to assume that $\sum_{k=0}^{\infty} \tau^{(k)} = \infty$, which, of course, is the case if one simply chooses $\tau^{(k)} = \tau > 0$.

Remark 1. Both algorithms are ad-hoc methods for the solution of the constrained minimization problem. Ad-hoc refers to the fact that in general the subgradient of the sum of the three terms is a superset of the sum of the single subgradients. In fact equality only holds if at least one of the subgradients $J(v_1)$ and $C_a(v_2)$ is continuous and $v_3/\delta \in \text{domain}(B^*)$ (see [6]). However, this, a priori is not guaranteed.

Since each Bregman distance minimization contains an enhancing step (see [13]) by ALM2 only the curvelet components are enhanced, while ALM1 also enhances the total variation component in addition.

For the standard ALM method $\zeta_2^{(k)}$ is an element of the subgradient of C_a at $v_2^{(k)}$ – actually, by adding it in an appropriate way to $v_2^{(k)}$ enhances it.

Convergence of the ALM algorithm has been proven in a very general setting. In our application, we have a simple situation of a finite dimensional Euclidean space and an l^2 comparison functional

$$(v_1, v_2, v_3) \rightarrow \mathcal{E}(v_1, v_2, v_3) = \frac{1}{2} \|f - v_1 - v_2 - v_3\|_X^2.$$

Thus the ALM method is convergent in the following sense.

Theorem 1. *Let f be admissible: That is, there exists a triple $(\tilde{v}_1, \tilde{v}_2, \tilde{v}_3)$, which is an element of the domain of \mathbf{J} , such that*

$$\tilde{v}_1 + \tilde{v}_2 + \tilde{v}_3 = f.$$

Then the ALM method consisting in minimization of

$$(v_1^{(k)}, v_2^{(k)}, v_3^{(k)}) := \arg \min_{(v_1, v_2, v_3)} \left\{ \frac{\tau^{(k)}}{2} \|f - v_1 - v_2 - v_3\|^2 + \right. \\ \left. D\mathbf{J}((v_1, v_2, v_3), (v_1^{(k-1)}, v_2^{(k-1)}, v_3^{(k-1)})) \right\} \quad (14)$$

satisfies

$$v_1^{(k)} + v_2^{(k)} + v_3^{(k)} \rightarrow f \text{ for } k \rightarrow \infty.$$

Remark 2. ALM2 is an alternating direction realization of the Augmented Lagrangian Method. To see this, we note that B^* is the characteristic function of the unit sphere with respect to the E -norm. Therefore, the gradient is zero on the domain and $+\infty$ outside. This means that for w in the domain of B^* we have

$$DB^*(w_3, w) = B^*(w_3).$$

Thus, to summarize: In each iteration of ALM2 an alternating direction of the Bregman distance is minimized. Taking into account the equivalence relation of the Bregman distance regularization and ALM, this also shows that our approach is an alternating direction minimization of the ALM method.

The second difference to the standard ALM, resulting from the alternating direction minimizations, consists in the use of two Lagrange parameters $\zeta_i^{(k)}$, $i = 1, 2$. The

standard ALM only uses one: According to (8) and (9) we have

$$\begin{aligned}\zeta_1^{(k)} - \zeta_1^{(k-1)} &\in \partial_{v_1} \mathcal{E}(v_1^{(k)}, v_2^{(k-1)}, v_3^{(k)}), \\ \zeta_2^{(k)} - \zeta_2^{(k-1)} &\in \partial_{v_2} \mathcal{E}(v_1^{(k)}, v_2^{(k)}, v_3^{(k)}).\end{aligned}$$

Note that the first and second subgradient are evaluated at two different points. ALM1 leaves out the update with respect to the total variation semi-norm (variable v_1) and enhances only the curvelets component (variable v_2). Therefore ALM1 pronounces the cartoon part more and the according decomposition into texture and noise appears more intuitive.

2.1 Numerical Results

The figures below show some decompositions calculated with the above algorithms. We used as input the noisy zebra image, Fig. 1 (right), which was calculated from Fig. 1 (left) by adding Gaußian random noise of variance 50.

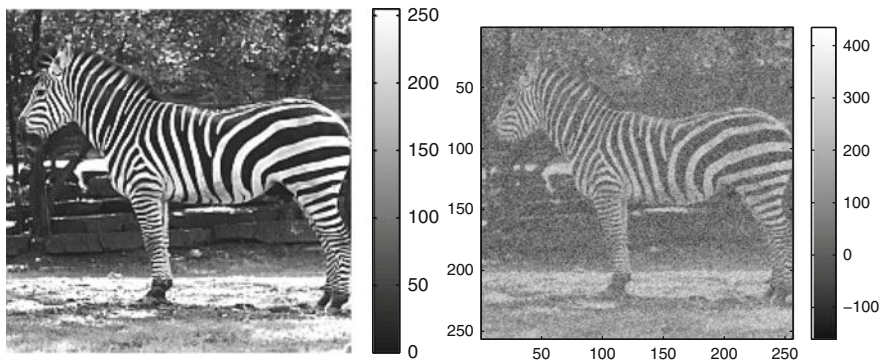


Fig. 1 *Left*: Ideal image. *Right*: Zebra data, which contains Gaußian random noise of variance 50

The following images are the output of the ALM1 algorithm after 40 iterations and of the ALM2 algorithm after 20 iterations. In both cases the input image f is decomposed into components v_1, v_2 , and v_3 . We emphasize that after a finite number of iterations the respective decomposition has only been calculated approximatively, but sufficiently well.

Figure 2 shows an approximative decomposition obtained with ALM1, where we used the parameters, $\alpha_2 = 0.6$, $\delta = 58.87$ and the curvelet weights $(a_1, a_2, a_3, a_4, a_5) = (8, 2, 0.02, 0.02, 0.02)$ of C_a (cf. (20)). Figure 3 shows an approximative decomposition obtained with ALM2, where we used the parameters, $\alpha_2 = 0.15$, $\delta = 58.87$ and the curvelet weights $(a_1, a_2, a_3, a_4, a_5) = (0.5, 0.5, 0.005, 0.005, 0.005)$ of C_a (cf. (20)).

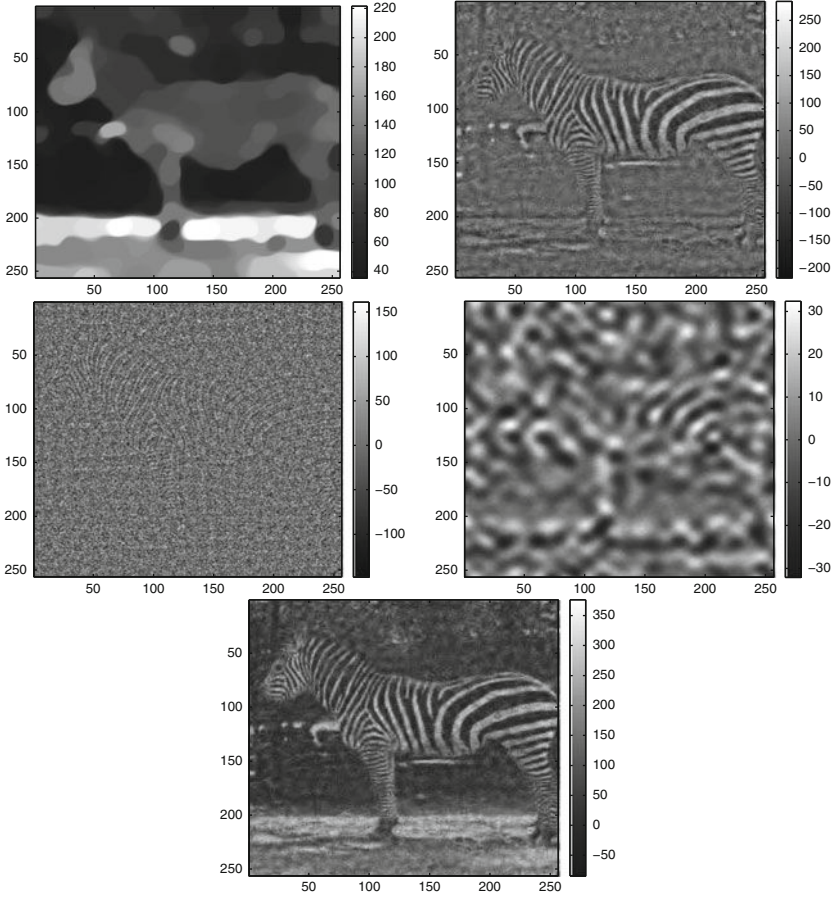


Fig. 2 ALM1: *Top Left*: Cartoon component v_1 , *Top Right*: Texture component v_2 , *Middle Left*: G -norm component v_3 . *Middle Right*: ζ_2 . *Bottom*: Cartoon + Texture: $v_1 + v_2$

A comparison of Figs. 2 and 3 shows that ALM1 finds a more distinct separation into components. Especially the cartoon part of ALM2 contains significant textured components.

3 Texture Enhancing Based on Image Decomposition

For *enhancing* data, a common procedure is to calculate a parabolic diffusion process backward in time. Here, we discuss an enhancing strategy based on dual variables of the ALM method.

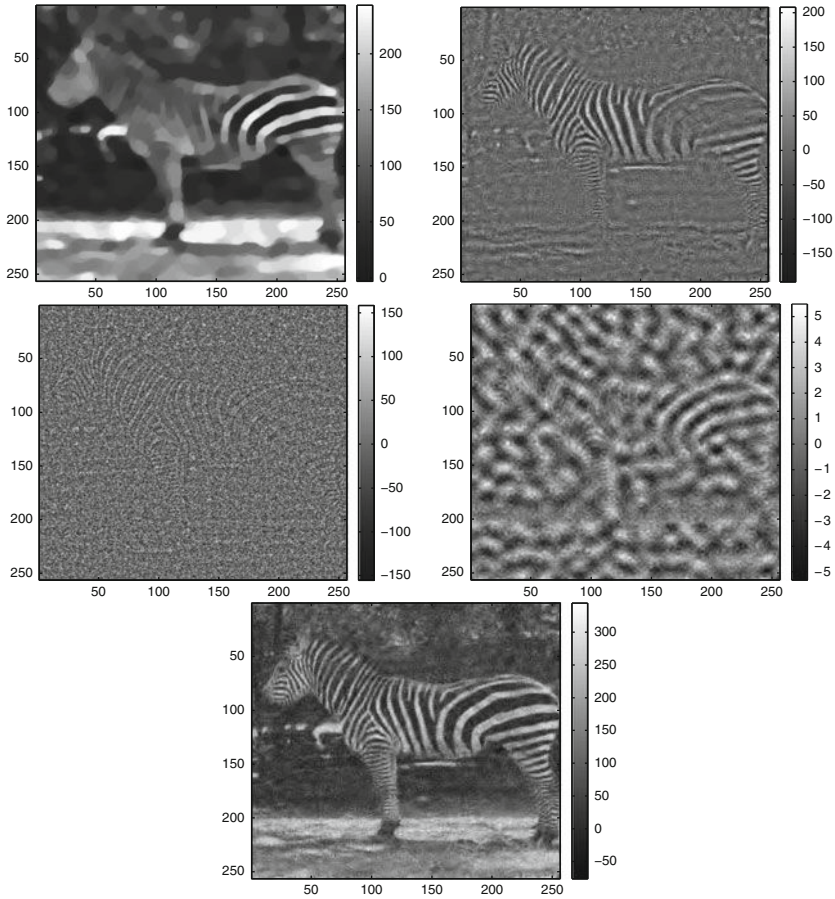


Fig. 3 ALM2: *Top Left*: Cartoon component v_1 , *Top Right*: Texture v_2 , *Middle Left*: G -norm component v_3 . *Middle Right*: ζ_2 . *Bottom*: Cartoon + Texture $v_1 + v_2$

3.1 Enhancing with Dual Variables

The proposed algorithm consists of two steps:

1. Image decomposition: By minimizing the functional \mathbf{F} defined in (6) we calculate three components v_i , $i = 1, 2, 3$ of $f = v_1 + v_2 + v_3$. We use both ALM1 and ALM2.
2. The second step is by enhancing the texture component. Here, we note that the dual variable $\lim_{k \rightarrow \infty} \zeta_2^{(k)} =: \zeta_2 \in \partial C_a(v_2)$ and calculate the enhanced image by

$$v_2^{\text{enh}} = v_2 + \tau \zeta_2$$

where τ is significantly larger than 1.

The following figures show cartoon enhanced images. In Fig. 4 we show enhanced texture components by adding ten times the texture component ζ_2 (dual variable of ALM1 and ALM2, respectively).

Then the image consisting of the sum of the cartoon, the enhanced texture, and the texture component is displayed.

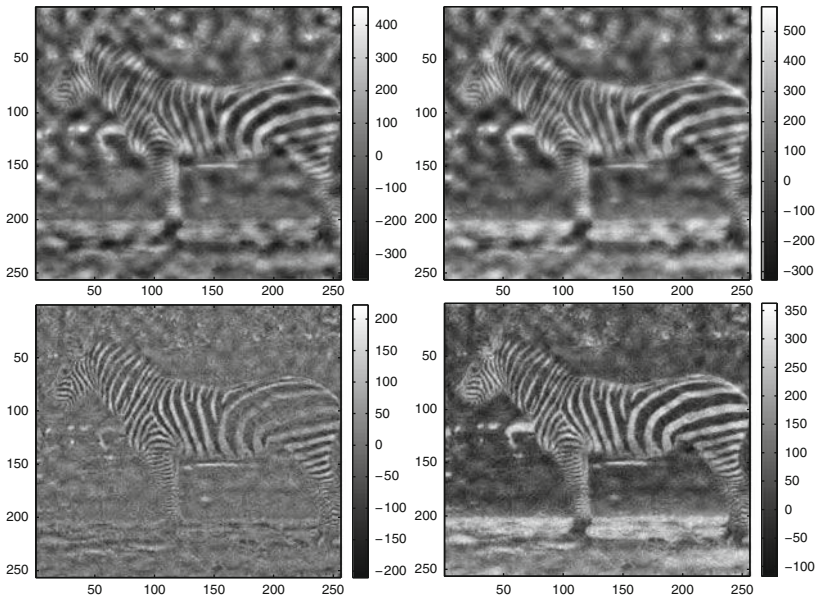


Fig. 4 *Top Left:* ALM2 enhanced texture component of v_2 with $\tau = 10$. *Top Right:* ALM2 enhanced image = $v_2^{\text{enh}} + v_1$. *Bottom Left:* ALM1 enhanced texture component of v_2 with $\tau = 10$. *Bottom Right:* ALM1 enhanced image = enhanced $v_2^{\text{enh}} + v_1$

4 Conclusion

In this paper we have considered two alternating direction variants of the Augmented Lagrangian Method for image decomposition. We have proposed a method which decomposes into texture, which is characterized to have finite l^1 -curvelet coefficients, a cartoon part, which has finite total variation norm, and noise and oscillating patterns, which have finite E -, G -norm, respectively. So far, the convergence analysis of the two algorithms is open. In the second part of the paper we have utilized the equivalence of the Augmented Lagrangian Method and the iterative Bregman distance to motivate some methods for texture enhancing. We concentrate on the enhancing feature for the texture.

5 Notations and Some Theoretical Background

5.1 Discrete Images

This paper is devoted to image decomposition and texture enhancing of discrete images.

- A discrete image consists of pixels $\Omega = \{1, \dots, N\}^2$ and intensity values $U := (u_{ij})_{(i,j) \in \Omega}$ associated with the pixels. Therefore every discrete image can be written as a matrix. Wavelets expansion require information on $\mathbb{Z} \times \mathbb{Z}$. We therefore associate with each image U the extended image $(u_{ij})_{(i,j) \in \mathbb{Z} \times \mathbb{Z}}$, where $u_{ij} = 0$ for $(i, j) \notin \Omega$.
- $X := \mathbb{R}^{N \times N}$ is the Euclidean space of images. It is associated with the scalar product $(u, v)_X = \sum_{1 \leq i, j \leq N} u_{i,j} v_{i,j}$.
- For $u \in X$ we define the *discrete gradient* ∇u via $(\nabla u)_{i,j} = \left((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2 \right)$, where

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{otherwise} \end{cases}$$

and

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{otherwise.} \end{cases}$$

- The *total variation* of the discrete image u is defined by

$$J(u) = \sum_{1 \leq i, j \leq N} |(\nabla u)_{i,j}|. \quad (15)$$

5.2 Besov Space

In this paper we use the homogeneous Besov spaces $\dot{B}_{1,1}^1$ of functions defined on \mathbb{R}^2 and its dual $E := \dot{B}_{-1,\infty}^\infty$. These spaces are defined as follows:

- $\Phi_{j,\mathbf{k}}$ denotes an orthonormal basis of smooth and compactly supported wavelets in $L^2(\mathbb{R}^2)$. The index $j \in \mathbb{Z}$ refers to scale and the index \mathbf{k} refers to position in \mathbb{R}^2 . That is, each function $u \in L^2(\mathbb{R}^2)$ can be written as

$$u(\mathbf{x}) = \sum_{(j,\mathbf{k}) \in \mathbb{Z}^3} c_{j,\mathbf{k}} \Phi_{j,\mathbf{k}}(\mathbf{x}) \text{ for } \mathbf{x} \in \mathbb{R}^2, \quad (16)$$

where

$$\|u\|_{L^2} = \sum_{(j,\mathbf{k}) \in \mathbb{Z}^3} |c_{j,\mathbf{k}}|^2 < \infty.$$

- It can be shown that $\dot{B}_{1,1}^1$ is the space of functions defined on \mathbb{R}^2 of the form (16) satisfying

$$\sum_{(j,\mathbf{k}) \in \mathbb{Z}^3} |c_{j,\mathbf{k}}| < \infty.$$

- The space E denotes the dual space of $\dot{B}_{1,1}^1$. It can be characterized by that the wavelet coefficients of functions in E satisfy

$$\sup_{(j,\mathbf{k}) \in \mathbb{Z}^3} |c_{j,\mathbf{k}}| < \infty.$$

We refer the interested reader for more information on this topic to [9].

5.3 Fenchel Duality

- Let \mathcal{X} be a locally convex space (such as \mathbb{R}^n with the Euclidean topology).
- Let $\Phi : \mathcal{X} \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$ be a convex, proper, and lower semi-continuous functional. We denote by Φ^* the dual or polar of Φ , i.e.,

$$\Phi^*(v) = \sup_u ((u, v)_{\mathcal{X}} - \Phi(u)).$$

- For a proper, convex, and lower semi-continuous functional $\Phi : \mathcal{X} \rightarrow \mathbb{R}$ and $f \in \mathcal{X}$, the following statements are equivalent (see e.g. [6]):

1. \hat{u} is a minimizer of the functional

$$u \rightarrow \Phi(u) + \frac{1}{2\lambda} \|f - u\|_{\mathcal{X}}^2. \quad (17)$$

2. $\hat{w} := f - \hat{u}$ is a minimizer of the functional

$$w \rightarrow \Phi^*\left(\frac{w}{\lambda}\right) + \frac{1}{2\lambda} \|f - w\|_{\mathcal{X}}^2. \quad (18)$$

- Let Φ be a semi-norm, then Φ^* is the indicator function of a closed convex set. For instance, let $\Phi(w) = \|w\|_{\dot{B}_{1,1}^1}$ be the norm of the homogeneous Besov space $\dot{B}_{1,1}^1$, then

$$B^*(w/\delta) = \chi_{\delta B_E}(w) = \begin{cases} 0 & \text{if } w \in \delta B_E := \{w : \|w\|_E \leq \delta\} \\ +\infty & \text{otherwise.} \end{cases} \quad (19)$$

5.4 Curvelets

- Curvelet functions $\psi_{j,\mathbf{k},l}$ form a tight frame of $L^2(\mathbb{R}^2)$. Here j denotes the scale index, l is a index of the angle, and \mathbf{k} is the index of the position. For more background on curvelets we refer to [3] (see also [8]) and references therein.

- Since the curvelet functions are constructed in such a way that they form a tight frame, every function $f \in L^2(\mathbb{R}^2)$ can be expanded in terms of these ansatz functions:

$$u(\mathbf{x}) = \sum_{(j,\mathbf{k},l) \in \mathbb{Z}^4} c_{j,\mathbf{k},l} \psi_{j,\mathbf{k},l}(\mathbf{x})$$

and satisfies

$$\|u\|_{L^2}^2 \leq \sum_{(j,\mathbf{k},l) \in \mathbb{Z}^4} |c_{j,\mathbf{k},l}|^2.$$

- Let $\mathbf{a} = (a_j)$ be a vector of numbers, which are uniformly bounded from below by a positive constant. The (scale) weighted discrete curvelet transform $C_{\mathbf{a}}$ of a function f is defined by

$$C_{\mathbf{a}}(u) := \sum_{(j,\mathbf{k},l) \in \mathbb{Z}^4} a_j |c_{j,\mathbf{k},l}|. \quad (20)$$

We call the functional $C_{\mathbf{a}}$ scale-weighted since the value of $C_{\mathbf{a}}$ is affected differently by the values of a_j .

The subgradient of $C_{\mathbf{a}}$ at u is the set of all functions with coefficients $u_{j,\mathbf{k},l} \in \text{sgn}(c_{j,\mathbf{k},l})$ for which

$$a_j |u_{j,\mathbf{k},l}| \in l^2.$$

Here $\text{sgn}(\rho)$ is an element of $[-1, 1]$ if $\rho = 0$ and $+1$ if $\rho > 0$ and -1 if $\rho < 0$.

Acknowledgement

The authors express are grateful to Andreas Obereder (Mathconsult Linz) for providing example files on anisotropic diffusion filtering. We also thank Kurt Rabitsch (CSC Vienna) for performing computations at the Vienna Scientific Cluster related. This work has been supported by the Austrian Science Fund (FWF) within the national research networks Industrial Geometry, project 9203-N12, and Photoacoustic Imaging in Biology and Medicine, project S10505-N20.

References

1. Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vis.* **22**(1), 71–88 (2005)
2. Aujol, J.-F., Chambolle, A.: Dual norms and image decomposition models. *Int. J. Comput. Vis.* **63**(1), 85–104 (2005)
3. Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. *Multiscale Model. Simul.* **5**, 861–899 (2006)
4. Cidade, G.A.G., Anteneodo, C., Roberty, N.C., Neto, A.J.S.: A generalized approach for atomic force microscopy image restoration with Bregman distances as Tikhonov regularization terms. *Inverse Probl. Sci. Eng.* **8**, 1068–2767 (2000)

5. Duval, V., Aujol, J.-F., Vese, L.A.: Mathematical modeling of textures: Application to color image decomposition with a projected gradient algorithm. *J. Math. Imaging Vis.* **37**, 232–248 (2010).
6. Ekeland, I., Temam, R.: *Convex Analysis and Variational Problems*. North-Holland, Amsterdam (1976)
7. Garnett, J., Le, T., Meyer, Y., Vese, L.: Image decompositions using bounded variation and generalized homogeneous Besov spaces. *Appl. Comput. Harmon. Anal.* **23**(1), 25–56 (2007)
8. Ma, J., Plonka, G.: Combined curvelet shrinkage and nonlinear anisotropic diffusion. *IEEE Trans. Image Process.* **16**, 2198–2206 (2007)
9. Meyer, Y.: *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22 of *University Lecture Series*. American Mathematical Society, Providence, RI (2001)
10. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation based image restoration. *Multiscale Model. Simul.* **4**(2), 460–489 (2005)
11. Osher, S., Solé, A., Vese, L.: Image decomposition and restoration using total variation minimization and the H^{-1} -norm. *Multiscale Model. Simul.* **1**(3), 349–370 (2003)
12. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**(1–4), 259–268 (1992)
13. Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: *Variational methods in imaging*, volume 167 of *Applied Mathematical Sciences*. Springer, New York (2009)
14. Starck, J.L., Elad, M., Donoho, D.L.: Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Process.* **14**, 1570–1582 (2005)
15. Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.* **19**(1–3), 553–572 (2003) Special issue in honor of the sixtieth birthday of Stanley Osher

A Locally Anisotropic Model for Image Texture Extraction

Loïc Piffet

Abstract We present a variational model for image texture identification. We first use a second order model introduced in Bergounioux and Piffet (Set Valued Variational Anal. 18(3–4):277–306 (2010)) for image denoising. The model involves a L^2 -data fitting term and a Tychonov-like regularization. We choose here the BV^2 norm, where BV^2 is the bounded hessian function space (see Bergounioux and Piffet, Set Valued Variational Anal. 18(3–4):277–306 (2010)). We observe that results are not satisfying since geometrical information appears in the oscillating component and should not. So we propose an anisotropic strategy, by setting components of the discrete hessian operator to 0 in order to focus on gradient directions. We precisely describe and illustrate the numerical methodology. Finally, we propose some numerical tests.

1 Introduction

This work is the continuation of [8], where we proposed a second order variational model in order to extract the texture from an image u_d . We assumed that u_d can be decomposed into two components : a cartoon part u and a part v containing strong oscillations (noise and texture). The model was the following :

$$\inf_{(u,v) \in BV(\Omega) \times BV^2(\Omega)} \frac{1}{2} \|u_d - u - v\|_{L^2(\Omega)}^2 + \lambda TV(u) + \mu TV^2(v), \quad (1)$$

with $\lambda, \mu \geq 0$; the term $\|u_d - u - v\|_{L^2(\Omega)}^2$ is the fitting data term $TV(u)$ and $TV^2(v)$ are respectively the first and second order total variation of u and v . We shall recall the definition in next section.

Loïc Piffet

Université d'Orléans, Laboratoire MAPMO – UMR 6628, Fédération Denis-Poisson, BP 6759,
F-45067 Orléans Cedex 2, France
e-mail: loic.piffet@univ-orleans.fr

Here, we do not consider this model any longer, but rather a Tychonov-like model introduced in [4] for image denoising. We assume that the image belongs to the space of bounded hessian functions $BV^2(\Omega)$ (that we recall in the sequel): this is equivalent to assert that $u = 0$ in the previous model. So, we consider the following:

$$\inf_{v \in BV^2(\Omega)} \frac{1}{2} \|u_d - v\|_{L^2(\Omega)}^2 + \mu TV2(v) \quad (2)$$

with $\mu \geq 0$. We shall investigate more precisely (1) in a forthcoming paper both from theoretical and numerical point of view. However, one can refer to [11]. In the present paper, we present a numerical improvement to solve problem (2) that has been introduced in [4].

The BV^2 -component v represents the regular part of the image, and the remainder term $w := u - u_d$ is the part containing noise, texture, and probably some contour lines. We can consider that a decomposition model is efficient for texture extraction if the whole oscillating information is contained in w , and v is only made of the smooth part as well as the geometrical information. Then, it turns out that (2), though it is a good model for image denoising, is not efficient for texture extraction. As a matter of fact, the component w contains the major part of the geometrical information of the original image. The goal of this work is to propose a method to remove contour lines from the component w . We use the idea introduced in [8] that consists in making the model become anisotropic. This permits to perform a better decomposition, but not really satisfactory. We propose here to improve this idea replacing the “global” anisotropy by a local anisotropy strategy, so that every contour line is specifically treated.

We first show the limits of the original (without any anisotropic strategy) and the globally anisotropic models. Then we introduce the locally anisotropic method. We shall focus on two images (Fig. 1) to present numerical tests before we give other examples.

2 The Second Order Variational Model

2.1 Infinite Dimensional Setting

Let us briefly recall the functional framework (for more details one can refer to [4, 7]) Let Ω be an open bounded subset of \mathbb{R}^2 smooth enough. We recall the definition of the space $BV(\Omega)$ of functions of bounded variation (see [1] for example)

$$BV(\Omega) = \{u \in L^1(\Omega) \mid TV(u) < +\infty\},$$

where

$$TV(u) := \sup \left\{ \int_{\Omega} u(x) \operatorname{div} \xi(x) dx \mid \xi \in \mathcal{C}_c^1(\Omega), \|\xi\|_{\infty} \leq 1 \right\}. \quad (3)$$

The space of bounded hessian functions $BV^2(\Omega)$ is defined as the space of $W^{1,1}(\Omega)$ functions such that $TV^2(u) < +\infty$ where

$$W^{1,1}(\Omega) = \{ u \in L^1(\Omega) \mid \nabla u \in L^1(\Omega) \}$$

is the classical Sobolev space and

$$TV^2(u) := \sup \left\{ \int_{\Omega} \langle \nabla u, \text{div}(\xi) \rangle_{\mathbb{R}^n} \mid \xi \in \mathcal{C}_c^2(\Omega, \mathbb{R}^{n \times n}), \|\xi\|_{\infty} \leq 1 \right\} < \infty, \quad (4)$$

where

$$\text{div}(\xi) = (\text{div}(\xi_1), \text{div}(\xi_2), \dots, \text{div}(\xi_n))$$

and

$$\forall i, \xi_i = (\xi_i^1, \xi_i^2, \dots, \xi_i^n) \in \mathbb{R}^n \text{ and } \text{div}(\xi_i) = \sum_{k=1}^n \frac{\partial \xi_i^k}{\partial x_k}.$$

The space $BV^2(\Omega)$ endowed with the following norm

$$\|f\|_{BV^2(\Omega)} := \|f\|_{W^{1,1}(\Omega)} + TV^2(f) = \|f\|_{L^1} + \|\nabla f\|_{L^1} + TV^2(f), \quad (5)$$

where TV^2 is given by (4), is a Banach space.

Now, we consider the following function, defined on $BV^2(\Omega)$:

$$\mathcal{F}(v) = \frac{1}{2} \|u_d - v\|_{L^2(\Omega)}^2 + \lambda TV^2(v, \Omega), \quad (6)$$

where $\lambda \geq 0$ and the optimization problem

$$\inf \left\{ \mathcal{F}(v) \mid v \in BV^2(\Omega), \frac{\partial v}{\partial n}|_{\partial\Omega} = 0 \right\}. \quad (7)$$

We first give a general existence and uniqueness result which slightly improves the one given in [4]:

Theorem 1. *Assume that $\lambda > 0$. Problem (7) has a unique solution v .*

Proof. Let $(v_n)_n$ be a minimizing sequence of (7). It is clear that $(TV^2(v_n))_n$ and $(\|v_n\|_{L^2(\Omega)})_n$ are bounded. The Poincaré–Wirtinger inequality proved in [3] allows to assert that $(v_n)_n$ is bounded in $W^{1,1}(\Omega)$, and thus bounded in $BV^2(\Omega)$. Embedding theorems given by F. Demengel [7] imply that $(v_n)_n$ converges (up to a subsequence) in $W^{1,1}(\Omega)$ to $v^* \in BV^2(\Omega)$. Then we may prove that v^* is a solution of (7) as in [4], what proves the existence. Uniqueness comes from the strict convexity of the cost functional \mathcal{F} . \square

If the image is noisy, the noise (and the texture) will appear in the oscillating part of the picture $w := u_d - v$. denoising Such an approach has already been used by Hinterberger and Scherzer in [9] for denoising process within $BV^2(\Omega)$ framework. However, their algorithm is different from the one we proposed in [4].

2.2 The Discretized Problem

We briefly recall the results of [4]. The problem (7) can be rewritten as

$$\inf_{v \in BV^2(\Omega)} \frac{\|u_d - v\|_{L^2(\Omega)}^2}{2\lambda} + TV^2(v, \Omega), \quad (8)$$

It is easy to see that problem (8) has a unique solution for every $\lambda > 0$.

In order to implement the solution, we first specify the discretization process. We consider here square images with size $N \times N$, for simplicity. To define a discrete version of the second total variation, we have to introduce a discrete version of the hessian operator. If $v \in X := \mathbb{R}^{N \times N}$, the “hessian”-matrix of v , that we denote (Hv) , is identified to a vector of $Z := X^4$, by

$$(Hv)_{i,j} = ((Hv)_{i,j}^{11}, (Hv)_{i,j}^{12}, (Hv)_{i,j}^{21}, (Hv)_{i,j}^{22}), \quad (9)$$

with, for $i, j = 1, \dots, N$,

$$\begin{aligned} (Hv)_{i,j}^{11} &= \begin{cases} v_{i+1,j} - 2v_{i,j} + v_{i-1,j} & \text{if } 1 < i < N, \\ v_{i+1,j} - v_{i,j} & \text{if } i = 1, \\ v_{i-1,j} - v_{i,j} & \text{if } i = N, \end{cases} \\ (Hv)_{i,j}^{12} &= \begin{cases} v_{i,j+1} - v_{i,j} - v_{i-1,j+1} + v_{i-1,j} & \text{if } 1 < i \leq N, 1 \leq j < N, \\ 0 & \text{if } i = 1, \\ 0 & \text{if } i = N, \end{cases} \\ (Hv)_{i,j}^{21} &= \begin{cases} v_{i+1,j} - v_{i,j} - v_{i+1,j-1} + v_{i,j-1} & \text{if } 1 \leq i < N, 1 < j \leq N, \\ 0 & \text{if } i = 1, \\ 0 & \text{if } i = N, \end{cases} \\ (Hv)_{i,j}^{22} &= \begin{cases} v_{i,j+1} - 2v_{i,j} + v_{i,j-1} & \text{if } 1 < j < N, \\ v_{i,j+1} - v_{i,j} & \text{if } j = 1, \\ v_{i,j-1} - v_{i,j} & \text{if } j = N. \end{cases} \end{aligned}$$

The discrete second order total variation of v reads

$$J_2(v) = \sum_{1 \leq i,j \leq N} \|(Hv)_{i,j}\|_{\mathbb{R}^4}. \quad (10)$$

We get the following discretized problem:

$$\inf_{v \in X} F(v) := \frac{\|u_d - v\|_X^2}{2\lambda} + J_2(v). \quad (11)$$

In the following, we adapt methods used by A. Chambolle for the Rudin–Osher–Fatemi model [10] to the second order case.

Optimality conditions for problem (11) use the Legendre–Fenchel conjugate J_2^* of J_2 . Theorem 2 gives the expression of J_2^* . The adjoint operator of $H, H^* : X^4 \rightarrow X$ is defined by :

$$\forall p \in X^4, \forall v \in X, \quad \langle H^* p, v \rangle_X = \langle p, H v \rangle_{X^4}.$$

The computation of H^* gives for every $p = (p^{11}, p^{12}, p^{21}, p^{22}) \in X^4$

$$\begin{aligned} (H^* p)_{i,j} = & \begin{cases} p_{i-1,j}^{11} - 2p_{i,j}^{11} + p_{i+1,j}^{11} & \text{if } 1 < i < N \\ p_{i+1,j}^{11} - p_{i,j}^{11} & \text{if } i = 1, \\ p_{i-1,j}^{11} - p_{i,j}^{11} & \text{if } i = N, \end{cases} \\ & + \begin{cases} p_{i,j-1}^{22} - 2p_{i,j}^{22} + p_{i,j+1}^{22} & \text{si } 1 < j < N, \\ p_{i,j+1}^{22} - p_{i,j}^{22} & \text{if } j = 1, \\ p_{i,j-1}^{22} - p_{i,j}^{22} & \text{if } j = N, \end{cases} \\ & + \begin{cases} p_{i,j-1}^{12} - p_{i,j}^{12} - p_{i+1,j-1}^{12} + p_{i+1,j}^{12} & \text{if } 1 < i, j < N, \\ p_{i+1,j}^{12} - p_{i+1,j-1}^{12} & \text{if } i = 1, 1 < j < N, \\ p_{i,j-1}^{12} - p_{i,j}^{12} & \text{if } i = N, 1 < j < N, \\ p_{i+1,j}^{12} - p_{i,j}^{12} & \text{if } 1 < i < N, j = 1, \\ p_{i,j-1}^{12} - p_{i+1,j-1}^{12} & \text{if } 1 < i < N, j = N, \\ p_{i+1,j}^{12} & \text{if } i = 1, j = 1, \\ -p_{i+1,j-1}^{12} & \text{if } i = 1, j = N, \\ -p_{i,j}^{12} & \text{if } i = N, j = 1, \\ p_{i,j-1}^{12} & \text{if } i = N, j = N, \end{cases} \\ & + \begin{cases} p_{i-1,j}^{21} - p_{i,j}^{21} - p_{i-1,j+1}^{21} + p_{i,j+1}^{21} & \text{if } 1 < i, j < N, \\ p_{i,j+1}^{21} - p_{i,j}^{21} & \text{if } i = 1, 1 < j < N, \\ p_{i-1,j}^{21} - p_{i-1,j+1}^{21} & \text{if } i = N, 1 < j < N, \\ p_{i,j+1}^{21} - p_{i-1,j+1}^{21} & \text{if } 1 < i < N, j = 1, \\ p_{i-1,j}^{21} - p_{i,j}^{21} & \text{if } 1 < i < N, j = N, \\ p_{i,j+1}^{21} & \text{if } i = 1, j = 1, \\ -p_{i,j}^{21} & \text{if } i = 1, j = N, \\ -p_{i-1,j+1}^{21} & \text{if } i = N, j = 1, \\ p_{i-1,j}^{21} & \text{if } i = N, j = N, \end{cases} \end{aligned}$$

We have proved in [4] the following existence result :

Theorem 2. *The Legendre–Fenchel conjugate of J_2 is the indicator function $\mathbf{1}_{K_2}$ of K_2 where*

$$K_2 := \{H^* p \mid p \in X^4, \|p_{i,j}\|_{\mathbb{R}^4} \leq 1, \forall i, j = 1, \dots, N\} \subset X. \quad (12)$$

and the solution v of (8) is characterized by : $v = u_d - P_{\lambda K_2}(u_d)$, where $P_{\lambda K_2}$ is the orthogonal projection operator on λK_2 .

We also proposed a fixed point algorithm similar to the one given by Chambolle in [6] that we have adapted to the second order framework. To compute $P_{\lambda K_2}(u_d)$ we have to solve

$$\min \left\{ \|\lambda H^* p - u_d\|_X^2 \mid p \in X^4, \|p_{i,j}\|_{\mathbb{R}^4}^2 - 1 \leq 0, i, j = 1, \dots, N \right\}.$$

The algorithm reads

$$p^0 = 0, \tag{13a}$$

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n - \tau (H[H^* p^n - f/\lambda])_{i,j}}{1 + \tau \|(H[H^* p^n - f/\lambda])_{i,j}\|_{\mathbb{R}^4}}. \tag{13b}$$

If $\tau \leq 1/64$ then $\lambda (H^* p^n)_n$ converges to $P_{\lambda K_2}(f)$.

3 Introducing Anisotropy

We observed in [4] that the second order model generates a blurring effect on the smooth BV^2 part. This means that contour lines are somehow too much “extracted” from the BV^2 component: they are involved in the oscillating component. As a result, this decomposition model is not efficient for texture extraction. This problem is illustrated by Fig. 2. We first present the two “academic” tests images.

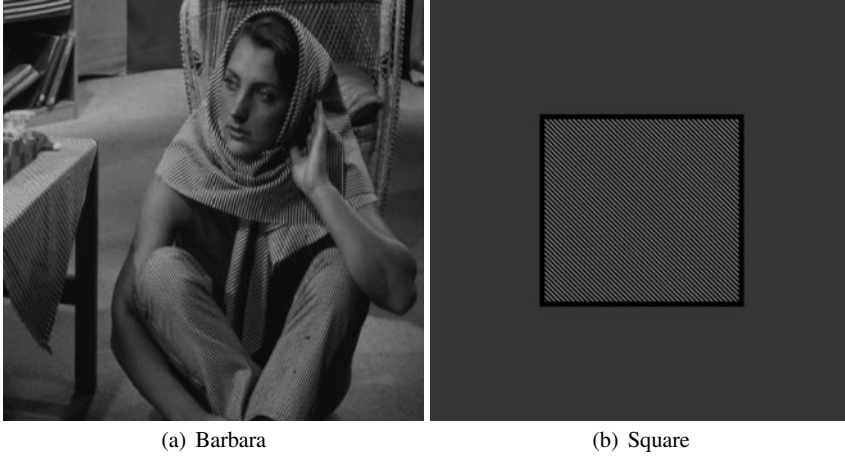


Fig. 1 Tests images

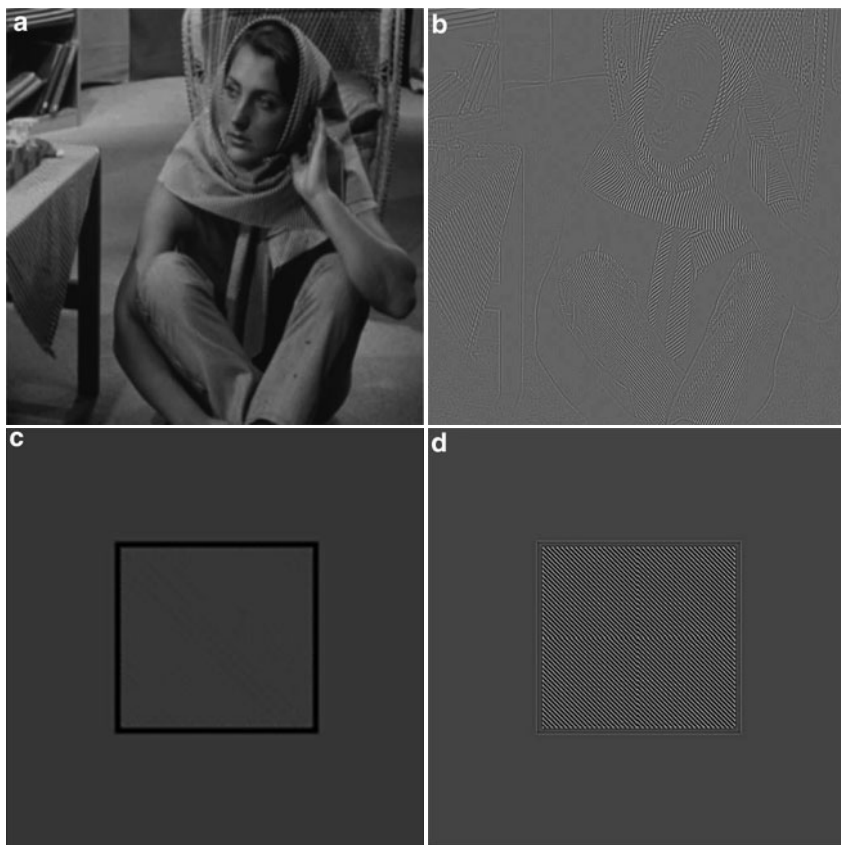


Fig. 2 (a) and (b) are the BV^2 - components from the test images. (c) and (d) are rescaled oscillating components

We can see that the whole texture is involved in the oscillating component (what we were expecting). However, this component contains geometrical information as well. Indeed, Fig. 2 shows that contour lines (arm, feet of the table in Fig. 1a, contour lines of the square in Fig. 1b...) appear in the texture part. We proposed in [8] a partial answer to this problem, with a “anisotropic” modification of the hessian operator, to give priority to chosen directions.

3.1 Uniform Anisotropic Model

We are going to uniformly modify the hessian operator. More precisely, the modification we perform is the same for all the pixels. The discretized hessian operator is given by (9) and formulas p. 144.

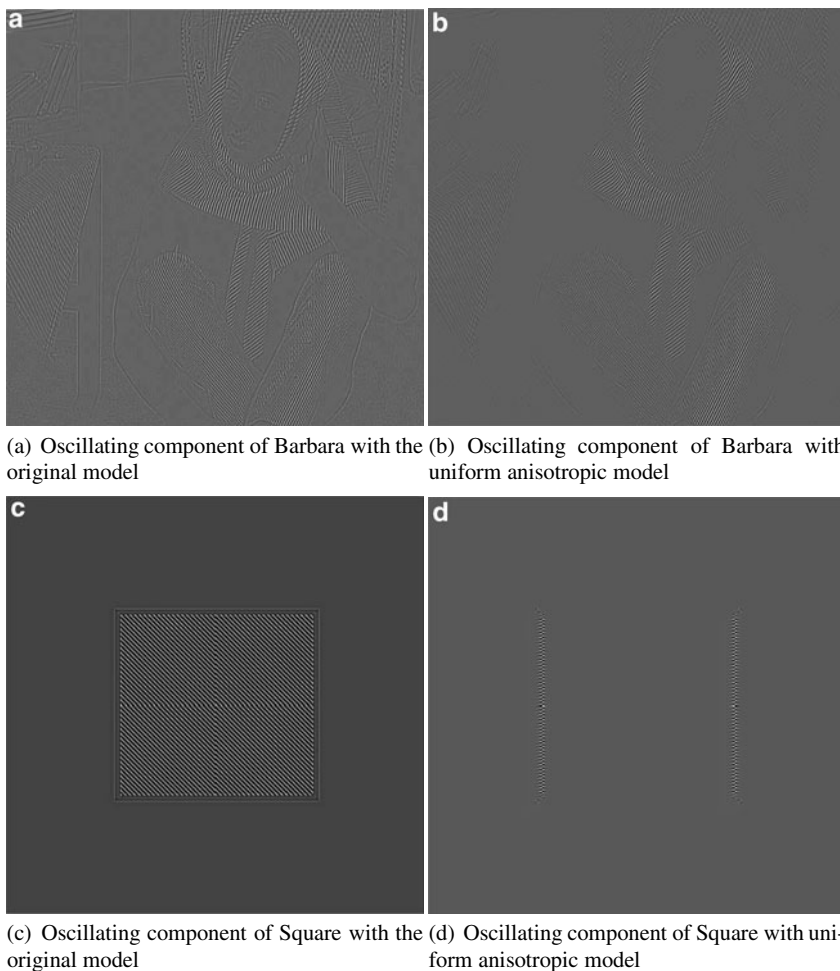


Fig. 3 Comparison between original and uniform anisotropic models – Oscillating components have been rescaled

Setting one of the Hessian components to 0 to cancels its effects during the processing. More precisely, we observe that if we set $(H\nu)^{1,1} = 0$, horizontal jumps

are not detected anymore, which means that vertical contour lines are not smoothed any longer. Thus, we keep these contours in the smooth part whereas they don't appear in the oscillating part anymore. Similarly, we can decide to act on horizontal contour lines ($(Hv)^{2,2} = 0$), or both vertical and horizontal contour lines ($(Hv)^{1,1} = (Hv)^{2,2} = 0$). We illustrate the last possibility in Fig. 3, where we compare results with and without uniform anisotropy. We notice that horizontal and vertical contours have disappeared from the texture part w that still contains almost all the oscillating information from the studied image. We notice that horizontal and vertical contour lines disappear. Nevertheless, diagonal contours still remain (Barbara's legs and arms). We also notice the loss of the oscillating information oriented horizontally and vertically (horizontal bands on Barbara's headscarf).

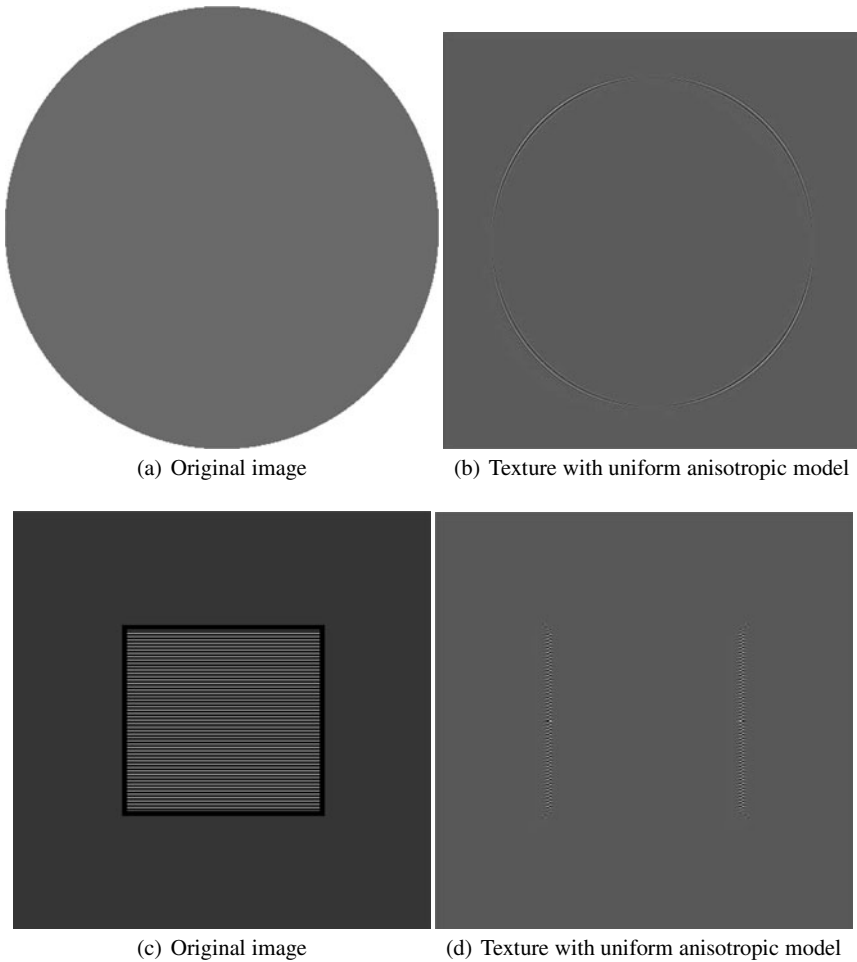


Fig. 4 One sees that the uniform anisotropic strategy is not good: the circle contours remain in the texture while the horizontal texture is lost in the second example

Unfortunately, even if we could deal with oblique contours (rotating the image), we can only consider two directions (the same for the whole image) which are necessarily orthogonal. Furthermore, once the direction is chosen, the (uniform) anisotropy strategy removes contour lines with respect to the direction and the texture oriented in the same direction as well. Figure 4 is an illustration of this point: we made tests with a circle. This is an image with contours along every direction and no texture. The second test image is very close to the one in Fig. 1b: texture is oriented in the same direction than contours we want to remove from the oscillating component. We can see the bad behaviour of the uniform anisotropic strategy in (Fig. 4).

To overcome these difficulties, we choose a local anisotropic strategy which depends on each pixel and is consistent with the contours.

3.2 Locally Anisotropic Model

3.2.1 Principle

We pointed out the bad behaviour of the uniform anisotropic model and especially the fact that we could only treat simultaneously two directions and that we necessarily lose the oscillating information in these two directions. So, we propose to work on contour lines locally. The different steps are the following :

- **Step 1 :** We first detect “interest” points (Fig. 5) which are pixels of contours that appear in the texture component and that we want to remove. The other pixels are treated with the original model (without any anisotropic strategy).
- **Step 2 :** We compute the gradient at every interest point: this gives us the angle α between the contour and (for example) the horizontal line.
- **Step 3 :** We extract a neighborhood (matrices of size $p \times p$) centered at every interest point, and rotate these submatrices with the angle α previously calculated. We get rotated thumbnails of a neighborhood of the considered pixel. Then, we apply the uniform anisotropic model to this thumbnail. We have to consider submatrices large enough to avoid side-effects (for example $p = 5$). This is illustrated Fig. 6, for a pixel detected at the first step.

Remark 1. We point at the fact that the process can be systematized introducing rotation matrices, that we locally apply to the discrete hessian operator (see [5]).

We hope that we remove significant contours from the oscillating component while preserving almost of all the texture. Figure 7 illustrates the result. Conversely, the BV^2 -component still contains a part of the texture, especially in a neighborhood of contours. We observe that we extract almost of all the texture information while

most contours have been removed. Practically, we had some difficulties that we mention in next section.

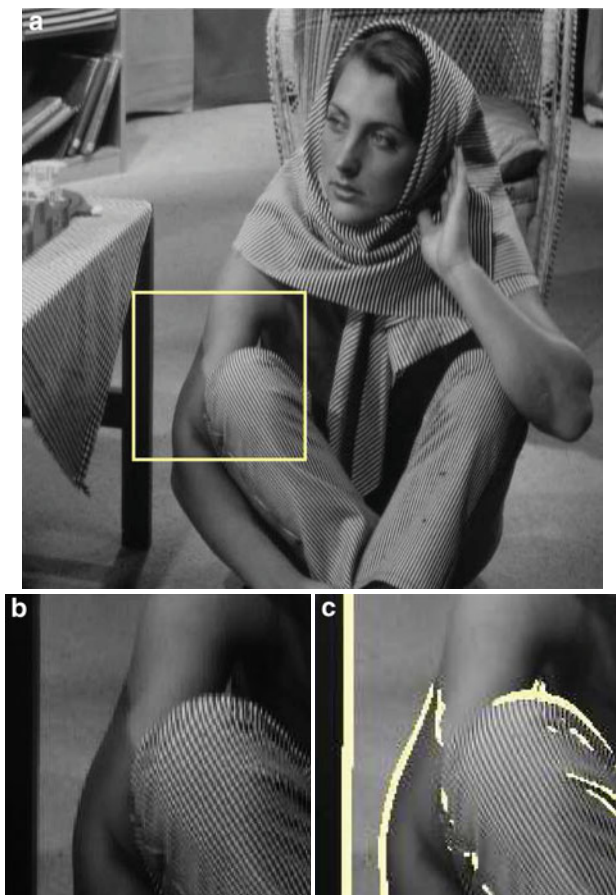


Fig. 5 (a) Choice of a pixel (b) zoom on the neighborhood (c) highlights pixels on which we apply the anisotropic model. Other points are treated with the original ROF2 model

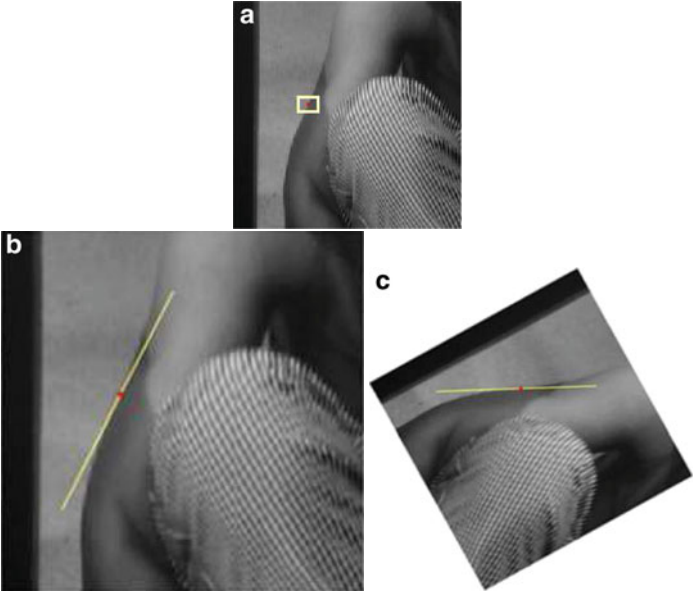
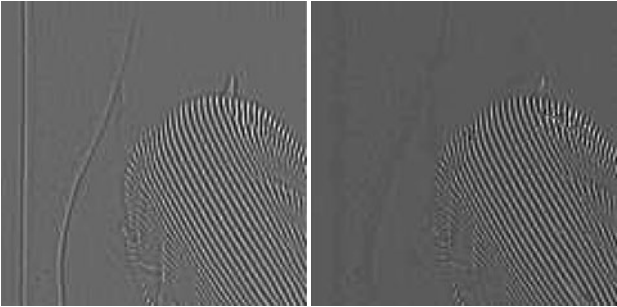


Fig. 6 (a) Choice of one of the pixels detected in Fig. 5 (b) Zoom on the pixel (c). After having calculated the angle α between the direction of the contour and the horizontal direction, we rotate the image to make the contour become horizontal. Then, we just have to apply the globally anisotropic model ($H^{2,2} = 0$) on a neighbourhood of the pixel



(a) Texture part obtained with the original ROF2 model (b) Texture part obtained with locally anisotropic strategy

Fig. 7 Comparison between original model (ROF2) and model involving local anisotropic strategy

3.2.2 Implementation

The first difficulty is to detect contour lines and “interest” points. As a matter of fact, we have to detect significant contours, without detecting points from texture. So we do not use classical contour detectors (as the gradient for example) which do not permit to make the difference between contours and texture (gradient is high in both cases).

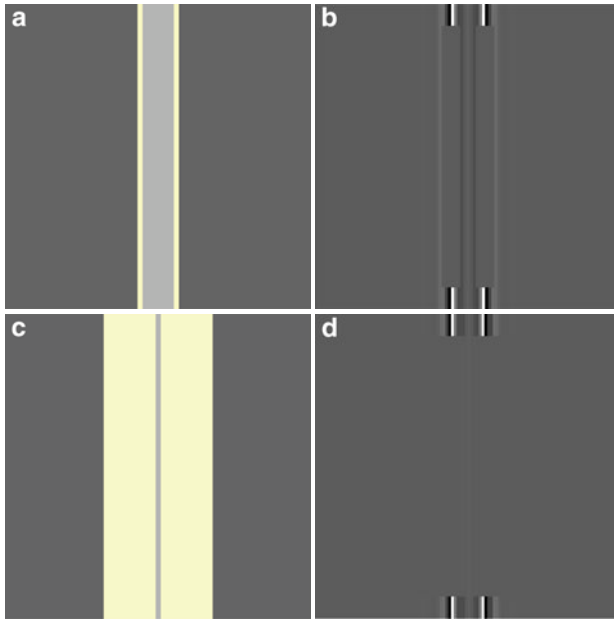


Fig. 8 Detection of points we have to treat with the locally anisotropic model on an image without texture. Images on the left put in evidence points locally treated. On the right, we propose the texture part of the associated solution. We don't apply the locally anisotropic model on the edges but the original ROF2 model to compare results. On (a), we detect contour lines with a gradient on the original image and treat these points with the locally anisotropic model; we observe on (b) that if a part of contour lines disappear, a lot of them remains in the oscillating component. We observe on (c) points that are detected thanks to a gradient on the BV^2 -component of the original image obtained with the original ROF2 model. We observe that we detect like this more points and on (d) that almost of all contour lines have disappeared from the texture component

Furthermore, a too much precise contour detector would not be appropriate as well. Indeed, we noticed in [4] that, even if the global model preserve contours, there is a small diffusion effect. The larger number of iterations or parameter λ value is, the more important is the diffusion effect. This means in particular that a point which is not a part of a contour but is close will sustain regularization effects. So, a

contour line detector, though it may quite efficient in a general case, will not detect all interest points, and, a contrario, will detect too many points from the texture. This phenomenon is illustrated Fig. 8 for an image without any texture. We can see that, in spite of the anisotropy, we still see vertical lines.

In order to overcome this problem, we first treat the image with the original algorithm (ROF2). We choose the parameter λ depending on the image and large enough to remove (almost) the whole texture. Contour lines are thereby diffused, and a classical contour detector (for example a gradient) applied on the smooth component permits to detect every interest point. Such a contour detector applied on the original image could not recover all interest points. Figure 8 highlights pixels that have to be treated with the anisotropic model and that do not appear in the original image. Then, we can notice that almost all contours are removed from the oscillating component.

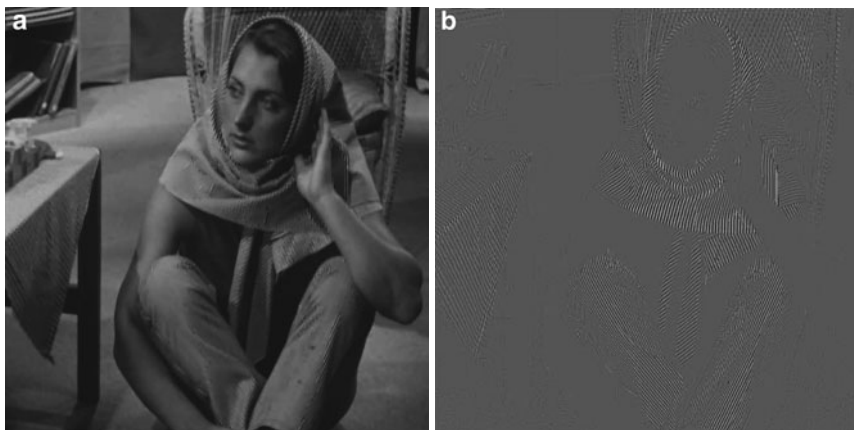


Fig. 9 Result obtained with the locally anisotropic model used on the full image. We see on (b) that the oscillating component almost of all the texture despite an acceptable loss of information near contour lines. The BV^2 component preserves in these places a part of the oscillating information

If the loss of useful information (texture) in the oscillating component is very acceptable compared to the geometrical information we make disappear, the “cartoon” part is not really satisfactory. Indeed, too much oscillating information is still involved in the smooth part. Therefore it is more consistent to call it texture extraction model rather than decomposition model.

4 Numerical Tests on a Natural Image

We propose to compare results obtained with the original ROF2 model and the locally anisotropic one on a natural image for different values of λ .

We observe that contour lines in the BV^2 -component are well preserved using the anisotropic model, so that we may use the concept of “cartoon” component as for the ROF model (see [2]). Moreover, we can see on the texture component that contours and edges visibly disappear with the add of anisotropy. Even if we can notice that the locally anisotropic model gives pretty good results for texture extraction, we still have to carefully understand what really happens and why simple modifications on the hessian operator gives such results. This will be covered in a forthcoming work.



Fig. 10 Original image

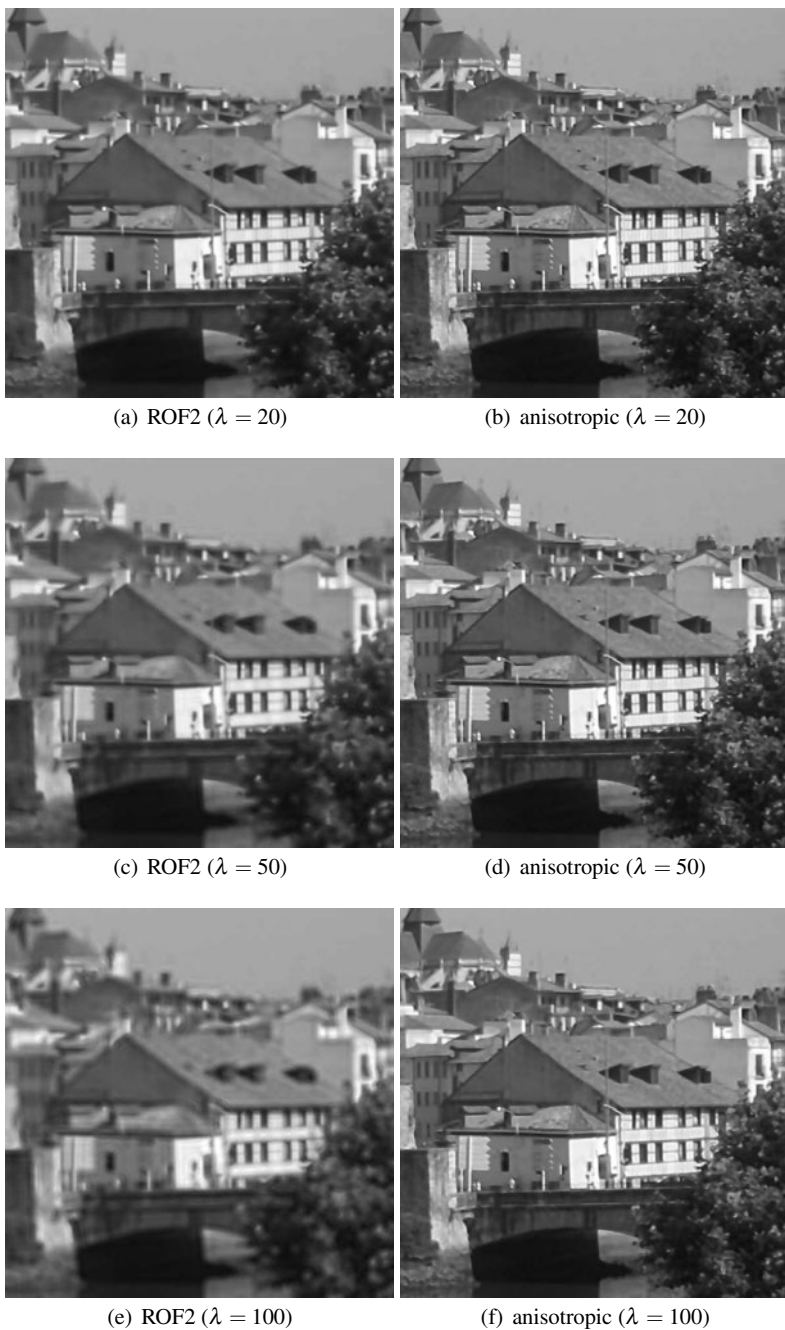


Fig. 11 BV^2 components of solutions obtained with the original ROF2 model (on the *left*) and the locally anisotropic model (on the *right*)

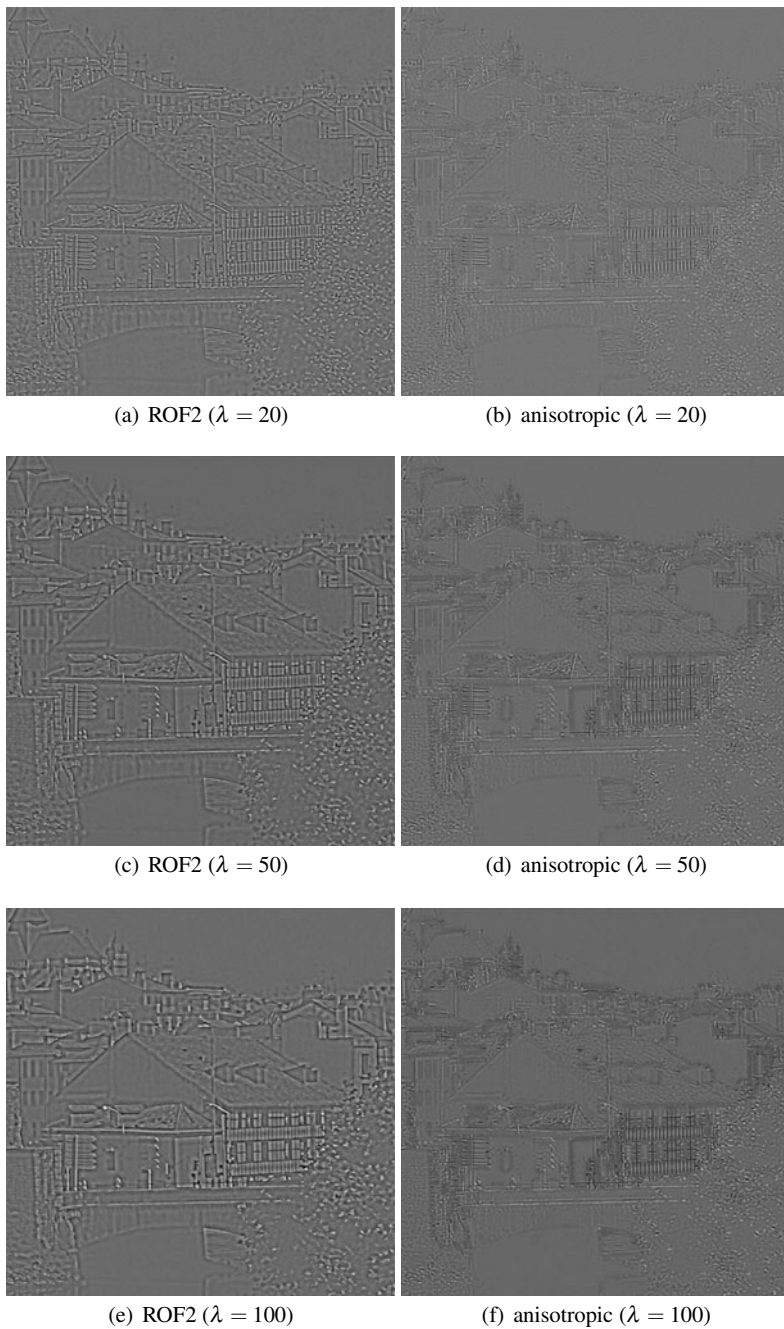


Fig. 12 texture components of solutions obtained with the original ROF2 model (on the *left*) and the locally anisotropic model (on the *right*)

References

1. Ambrosio, L., Fusco, N., Pallara, D.: Functions of Bounded Variation and Free Discontinuity Problems. Oxford Mathematical Monographs. Oxford University Press, Oxford (2000)
2. Aujol, J.-F.: Some First-order algorithms for total variation based image restoration. *J. Math. Imaging Vis.* **34**, 307–327 (2009)
3. Bergounioux, M.: On Poincaré-Wirtinger Inequalities in Spaces of Functions of Bounded Variation. <http://hal.archives-ouvertes.fr/hal-00515451/fr/> (2010)
4. Bergounioux, M., Piffet, L.: A second-order model for image denoising. *Set Valued Variational Anal.* **18**(3–4), 277–306 (2010)
5. Bergounioux, M., Tran, M.P.: A second order model for 3D-texture extraction. A second order model for 3D-texture extraction. In: Bergounioux, M. (ed.) *Mathematical Image Processing*. Springer, Heidelberg (2010) <http://hal.archives-ouvertes.fr/hal-00530816/fr>
6. Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* **20**, 89–97 (2004)
7. Demengel, F.: Fonctions à hessien borné. *Annales de l'Institut Fourier*, tome **34**(2), 155–190 (1984)
8. Echegut, R., Piffet, L.: A variational model for image texture identification. In: Diehl, M., Glineur, F., Jarlebring, E., Michiels, W. (eds.) *Recent Advances in Optimization and Its Applications in Engineering*, pp. 471–480. Springer, Berlin (2010)
9. Hinterberger, W., Scherzer, O.: Variational methods on the space of functions of bounded Hessian for convexification and denoising. *Computing* **76**(1–2), 109–133 (2006)
10. Osher, S., Fatemi, E., Rudin, L.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**, 259–268 (1992)
11. Piffet, L.: Modèles variationnels pour l'extraction de textures 2D, Ph.D. Thesis, Orléans (2010)

A Neural Field Model for Motion Estimation

Émilien Tlapale, Pierre Kornprobst, Guillaume S. Masson, and Olivier Faugeras

Abstract We propose a bio-inspired approach to motion estimation based on recent neuroscience findings concerning the motion pathway. Our goal is to identify the key biological features in order to reach a good compromise between bio-inspiration and computational efficiency. Here we choose the neural field formalism which provides a sound mathematical framework to describe the model at a macroscopic scale. Within this framework we define the cortical activity as coupled integro-differential equations and we prove the well-posedness of the model. We show how our model performs on some classical computer vision videos, and we compare its behaviour against the visual system on a simple classical video used in psychophysics. As a whole, this article contributes to bring new ideas from computational neuroscience in the domain of computer vision, concerning modelling principles and mathematical formalism.

1 Introduction

Biological vision shows intriguing characteristics in terms of performance and robustness. For example, the primate visual system is able to handle very large inputs, with around sixty-five millions cones and twice as many rods, whereas

Émilien Tlapale, Pierre Kornprobst (✉) and Olivier Faugeras
NeuroMathComp Project Team, INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles,
06902 Sophia Antipolis, France
e-mail: Emilien.Tlapale@inria.fr (Émilien Tlapale), pierre.kornprobst@inria.fr (Pierre Kornprobst), Olivier.Faugeras@inria.fr (Olivier Faugeras)

Guillaume S. Masson
INCM,UMR 6193 CNRS-Université de la Méditerranée 31, chemin Joseph Aiguier 13402
Marseille cedex, France
e-mail: guillaume.masson@incm.cnrs-mrs.fr

typical software input is two or three orders of magnitude smaller in terms of pixels. The luminance range and contrast sensitivity of the primate visual system are hardly comparable to the few bits used to represent grey scale as input to algorithmic software. A final noteworthy feature of the visual system is its capacity to respond correctly to a wide range of spatial and temporal scales, implying sensitivity to a large extent of velocities. This extraordinary biological machinery is moreover capable of achieving fast performance despite its intrinsic slow communication bandwidth.

Motion estimation is a task performed very well by the primate visual cortex. But motion estimation is also a key feature for many vision and robotic applications, and it is an active field of research in the computer vision community. Indeed, there is still a strong need for more accurate and efficient methods, and the enthusiasm around the estimation of optical flow is clear from the success of the recent benchmark proposed by [5].

In this article, our goal is to contribute towards bridging the gap between algorithmic and biological vision by proposing a bio-inspired motion estimation model based on a suitable mathematical framework. It is our conviction that a breakthrough can be reached by understanding which mechanisms make the visual system so robust when it deals with a large variety of data under changing environmental conditions.

We based our approach on three main ideas. First, the design of our model mimics the functional properties of the main visual cortex layers dedicated to motion. Second, the mathematical framework chosen is well posed and suitable to model cortical layers activity at a macroscopic scale. Third, we show how the evaluation of motion estimation models can be further extended, in the light of the available experimental data in neuroscience.

The article is organized as follows. In Sect. 2 we revisit the main properties of the visual system focusing on the motion pathway. In Sect. 3 we present our neural field model defined by a set of two coupled integral equations. We prove that our model is mathematically well-posed and we relate it with some classical approaches in computer vision (based on partial differential equations). In Sect. 4 we present our results, obtained not only on videos from computer vision but also on a classical stimulus from psychophysical experiments. Finally, in Sect. 5 we give the main perspectives related to this work.

2 The Primate Visual System

2.1 General Considerations

Two main functional streams of visual information processing are classically distinguished in the literature: the *form pathway*, which processes static features; the *motion pathway*, which concerns motion integration and segmentation. Both

pathways receive their input from the retina through the lateral geniculate nucleus (LGN) and the primary visual cortex (V1). This segregated organization is rooted in a similar dichotomy found at the neuronal level where parvocellular and magnocellular neurons exhibit different spatio-temporal bandwidths, color preferences and luminance contrast sensitivities as well as different conduction times [10, 43]. The focus of the present model is the cortical motion processing that depends primarily (but not exclusively, see [44]) upon the inputs for the motion pathway.

The primate visual cortex can be seen as set of densely connected aggregates of neurons known as *cortical maps*. Cortical maps form a highly connected hierarchy with forward and backward streams of information. Each of these cortical maps has been identified as processing a specific information such as texture, color, orientation or motion. Most of these cortical maps are retinotopically organized: adjacent neurons have receptive fields that cover slightly different, but overlapping portions of the visual field. This spatial organization of the neuronal responses to visual stimuli generally preserves the topology of the visual input, and cortical maps can be seen as a function of the visual input. The primary visual cortex (V1) feeds higher order maps distributed within several extra-striate areas spanning both parietal and temporal lobes of the neo-cortex. The information is transmitted to subsequent cortical maps by convergent forward connections. Note that higher order maps correspond to a coarser analysis of the visual input since they integrate more information by cumulative forward connections. Although much less is known about the properties and role of feedback projections, recent evidence has been accumulated to suggest that feedback indeed plays a crucial role in processing the visual information through context-dependent, modulatory effects or long distance diffusion [53].

2.2 The Motion Pathway

In the present paper we focus on two specific cortical maps known as V1 and MT. They both play a crucial role in estimating local motion and computing pattern motion properties. Moreover, their main characteristics are among the most examined in neuroscience, offering an enormous bulk of experimental data at different scales, from single neurons to behaviour. In this section, our goal is to briefly review some of the main features of these maps, emphasizing their functional properties and connectivities [11, 13, 15, 27].

The majority of inputs to MT comes from the V1 cortical map, particularly from its layer 4B [11]. While the exact computational rules are still disputed, there is a general view that V1 and MT implement the forward stream of a two stage motion integration [11]. Recently, it was found that some V1 complex cells exhibit some speed tuning, although these local estimates of target speeds are of large bandwidth. Lastly, V1 neurons are highly orientation selective. Both strong orientation selectivity and small receptive field size make V1 neurons particularly sensitive to the aperture problem.

Another important feature is that MT has many feedback connections to V1 [50]. There are experimental data supporting the view that such feedback play a crucial role in context-dependent processing by shaping center-surround interactions within V1 [3, 35, 53]. Feedback connections are also much faster than the horizontal ones (2–6 m/s versus 0.1–0.2 m/s) according to [31]. Thus, feedbacks from MT to V1 are a significant feature to take into account in models.

2.3 From Biology to Bio-Inspired Models?

Most models aiming to reproduce motion integration mechanisms are two-stage feed-forward models where V1 acts as a local-motion detector, and MT implements motion integration by pooling local motion cues. However, these models ignore two essential properties of motion integration.

The first property is that motion integration is fundamentally a spatial process. Various non-ambiguous motion cues need to be integrated and segregated to propagate motion information inside surfaces. This spatial property of motion integration has only been investigated in a few bio-inspired models [7, 28, 32, 63].

The second property is that motion integration is fundamentally a dynamical process. When presented with line drawings, plaids or barber poles, the perceived motion direction shifts over time [16, 52, 67]. Similar dynamics can be found in smooth pursuit eye movements, and reflect neural time courses. These dynamics however are only beginning to be investigated by modelers such as [42, 56].

In this article, our model has two recurrently connected stages. Since we want to consider the dynamics of the processing from a mathematical as well as behavioral perspective, the model is written as a dynamical system of neural field equations. The dynamics is the distinguishing feature without which comparisons to biology would be highly difficult. As a whole, in the context of motion estimation and up to our knowledge, this is the first contribution to propose a bio-inspired model based on the neural field formalism to handle real dynamical scenes.

3 A Neural Field Model for Motion Estimation

3.1 Biological Inspiration of the Model

Figure 1 describes the general structure of our model. Given an input k_1 , motion is estimated and integrated at two different spatial scales within two maps (p_1, p_2) that are recurrently interconnected. The connectivity rules between these two maps are written as a set of coupled integral equations (through forward and backward connections). This functional structure is inspired by the biology as discussed below.

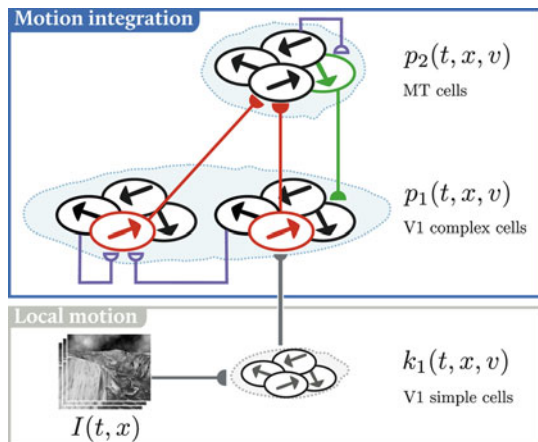


Fig. 1 A schematic view of the model showing the interactions of the different cortical layers. From a grey-level input, our model estimates local motion information in k_1 , which is then used in a feed-forward (red) / feedback (green) loop between p_1 and p_2 . There are also lateral connections in each layer (blue)

The input to our system (denoted by k_1) corresponds to a local motion estimation. Here it is based on a measure of correlations between frames (modified Reichardt detectors, see Sect. 3.2.2).

The first layer of our model (denoted by p_1) computes local direction and speed of motion. This corresponds to complex cells in primary visual cortex that have been shown to perform local velocity computation [48].

The second layer of our model (denoted by p_2) integrates motion over larger portions of the image and the information is propagated back to the first layer. This corresponds to MT cell properties. Our MT-like cells have larger receptive fields and are tuned to lower spatial frequencies and higher speeds than V1 cells. This fact is consistent with the view that V1 and MT stages operate at different scales [11]. Feed-forward models of motion integration are heavily rooted on such evidence [39, 51, 54, 66]. However, V1 and MT are recurrently interconnected [53] and existing models have shown that such a recurrent connectivity can play a role in solving the aperture problem in synthetic and natural sequences [7, 17], as well as implementing contextual effects observed in V1 and MT neurons [3].

Beyond the similarities concerning the functional structure, another major innovative aspect of our contribution is to propose a truly dynamical model. More precisely, we are interested in the evolution of motion estimation between frames so that comparison with neural and behavioral time courses becomes possible.

3.2 Description of Maps Interactions

3.2.1 The Neural Field Framework

One major difficulty in observing or modelling the brain is that its study involves a multi-scale and thus is a multi-disciplinary analysis. As noted by [18], there is a large variety of scales but there is no integrative model yet covering all of them at once. Thus, one has to choose a given scale and define a suitable mathematical framework for that scale.

In this article we consider the macroscopic scale, defining the cortical activity at the population level, for the number of neurons and synapses even in a small piece of cortical area is immense. In order to describe cortical activity at the population level, neural field models are proposed as a continuum approximation of the neural activity.

Since the seminal work by [64, 65] and [2], intensive research has been carried out to extend models and study them mathematically. The general mathematical study of such neural field equations can be very complex and it is still a challenging field of research [19, 26, 30, 58].

Our goal is to use this formalism for the problem of motion estimation. Following the general structure described in Sect. 3.1, the two maps describing the firing rate activity of a continuum of neuron populations are denoted by

$$p_i : (t, x, v) \in \mathbb{R}^+ \times \Omega \times \mathcal{V} \rightarrow p_i(t, x, v) \in [0, 1], \quad (1)$$

for $i \in \{1, 2\}$, where Ω is the spatial domain (a bounded open subset of \mathbb{R}^2) and $\mathcal{V} \subseteq \mathbb{R}^2$ is the velocity space (the space of possible velocities). $p_i(t, x, v)$ is the average activity of the population corresponding to position x and tuned to velocity v [47].

The general neural equation for an activity based model is:

$$\frac{\partial \mathbf{p}}{\partial t}(t, r) = -\Lambda \cdot \mathbf{p}(t, r) + \mathbf{S} \left(\int_{\Omega \times \mathcal{V}} \mathbf{W}(t, r, r') \mathbf{p}(t, r') dr' + \mathbf{K}(t, r) \right), \quad (2)$$

where $\mathbf{p} = (p_1, p_2)^T$, $r = (x, v)$ characterizes the population (position and velocity tuning), $\Lambda = \text{diag}(\lambda_1, \lambda_2)$ is a matrix describing the temporal dynamics of the membrane potential, and $\mathbf{S}(x) = (S_1(x_1), S_2(x_2))^T$ is a matrix of sigmoidal functions (defined by $S_i(s) = s/(1 + e^{-s})$). \mathbf{K} is an external current that models external sources of excitations (in our case, $\mathbf{K} = (\lambda_1^f k_1, 0)^T$ since there is no external input to map p_2). More importantly, $\mathbf{W}(t, r, r')$ describes how the population r' (at position x' and tuned to the velocity v') influences the population r at time t .

In the right-hand side of (2), the first term denotes the passive activity decay (with rate $\lambda_{1,2}$) when the input features to the target population is switched off. The second term denotes the cells activation functions ($S_{1,2}$), a non-linear transformation of the input.

Before giving more details on our model, let us mention three general interesting properties of the neural field formalism. First, the integral definition of the

weights allows large extent connectivity. Second, the sigmoid provides a tool to study contrast-gain effects. Finally, delays can be incorporated [22, 58] to constrain the connectivity.

3.2.2 Local Motion Estimation

The initial stage of every motion processing system is the computation of local motion cues as input to the system. Various models of motion detection have been proposed in the literature, with different degrees of biological plausibility [1, 49, 57, 61].

Starting from the input image sequence $I: (t, x) \in \mathbb{R}^+ \times \Omega \rightarrow I(t, x)$, we estimate the local motion k_1 using modified Reichardt detectors [7] enhanced to support sub-pixel velocities estimation. Two filtered images are correlated to estimate population activity: Directional derivatives are used to filter the input:

$$c_1(t, x, \alpha) = \frac{I(t, x) \overset{x}{*} \partial_\alpha^2 G_\sigma}{\varepsilon + \sum_{\beta \in \mathcal{O}} |I(t, x) \overset{x}{*} \partial_\beta^2 G_\sigma| \overset{x}{*} G_\sigma},$$

where ε avoids division by zero, G_σ denotes a Gaussian kernel, σ 's are scaling constants, $\overset{x}{*}$ denotes the convolution operator in space and ∂_α^2 denotes the second order directional derivative in the direction $\alpha \in \mathcal{O}$.

From these filtered outputs, we defined the half detectors by correlation with another frame:

$$\begin{aligned} c_2^+(t, x, v) &= \left(\sum_{\alpha \in \mathcal{O}} c_1(t, x, \alpha) c_1(t+1, x+v, \alpha) \right) \overset{x}{*} G_\sigma, \\ c_2^-(t, x, v) &= \left(\sum_{\alpha \in \mathcal{O}} c_1(t+1, x, \alpha) c_1(t, x+v, \alpha) \right) \overset{x}{*} G_\sigma, \end{aligned}$$

where σ 's are scaling constants. The half detectors are then combined by:

$$k_1(t, x, v) = \frac{|c_2^+(t, x, v)|_+ - \frac{1}{2}|c_2^-(t, x, v)|_+}{1 + |c_2^-(t, x, v)|_+},$$

where $|x|_+ = \max(0, x)$ is a positive rectification, for the activity of neurons is always positive.

3.2.3 Core Equations

The core of our model is defined by the interaction between the two populations, p_1 and p_2 , as described in (2). More precisely, we propose the following model

$$\frac{\partial p_1}{\partial t}(t, r) = -\lambda_1 p_1(t, r) + S_1 \left(k_1(t, r)(\lambda_1^f + \lambda^b p_2(t, r)) - \lambda_1^l G_{\sigma_1^l} \star \int_{\mathcal{V}} p_1(t, x, w) dw \right. \\ \left. + \lambda_1^d (G_{\sigma_1^d} \star^{x,v} p_1(t, r) - p_1(t, r)) \right), \quad (3)$$

$$\frac{\partial p_2}{\partial t}(t, r) = -\lambda_2 p_2(t, r) + S_2 \left(\lambda_2^f G_{\sigma_2^f} \star p_1(t, r) - \lambda_2^l G_{\sigma_2^l} \star \int_{\mathcal{V}} p_2(t, x, w) dw \right. \\ \left. + (G_{\sigma_2^d} \star^{x,v} p_2(t, r) - p_2(t, r)) \right), \quad (4)$$

where $r = (x, v)$ denotes the characteristic of the population (position and velocity tuning), λ 's and σ 's are constants and G_σ denote Gaussian kernels defined by

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{|x|^2}{2\sigma^2}},$$

when the convolution is in the spatial domain only, and

$$G_\sigma(x, v) = \frac{1}{4\pi^2\sigma_x^2\sigma_v^2} e^{-\frac{|x|^2}{2\sigma_x^2} - \frac{|v|^2}{2\sigma_v^2}}, \quad \text{with } \sigma = (\sigma_x, \sigma_v) \quad (5)$$

when the convolution is in the spatial and velocity domains.

One interesting property of this model is that it naturally performs a multi-scale analysis of motion through the exchanges between the two populations: (a) The feed-forward input from the previous layer (p_1) is integrated at the level of p_2 using a Gaussian weighting function, thus implementing the V1-to-MT convergence of connectivities ($\lambda_2^f G_{\sigma_2^f} \star p_1$), and (b) the feedback signals are modulatory to yield a correlative enhancement when feed-forward and feedback coincide, while the feed-forward input is left unchanged when no feedback is delivered (term $k_1(\lambda_1^f + \lambda^b p_2)$). Note that the feedback is written in a multiplicative way (as in [7]): We used a modulating feedback rather than driving feedback, similar to that found in studies of the motion processing system in primates [53].

For both equations a selection mechanism is defined by the term $-\lambda G_\sigma \star \int_{\mathcal{V}} p_i(t, x, w) dw$. Such short-range lateral inhibition, usually called recurrent inhibition, leads to a winner-take-all mechanism [21, 68]. Instead of a divisive inhibition found in some models [7, 46], we implemented a subtractive inhibition in order to fit into the neural field formalism.

Finally, both equations incorporate a diffusion term defined by $G_{\sigma_i^d} \star^{x,v} p_i - p_i$. As shown in Lemma (1) this term behaves asymptotically like a Laplacian operator.

Lemma 1. *Given a Gaussian kernel G_σ as defined in (5) with $\sigma = (\rho, \eta)$, let us denote*

$$A = G_\sigma \star^{x,v} p(x, v) - p(x, v), \quad (6)$$

then we have

$$A = \frac{\rho^2}{2} \sqrt{\pi} D_x^2 p(x, v) + \frac{\eta^2}{2} \sqrt{\pi} D_v^2 p(x, v) + o(\rho^2, \eta^2, \rho\eta). \quad (7)$$

where D_x^2, D_v^2 denote the Laplacian operator in the physical space and the velocity space.

Proof. Rescaling inside the integral in (6) we get

$$A = \frac{1}{\pi^2} \int_{\mathbb{R}^4} e^{-|y|^2 - |w|^2} (p(x - \rho y, v - \eta w) - p(x, v)) dy dw.$$

Then using a Taylor expansion of p (and assuming that $p \in \mathcal{C}^3(\mathbb{R}^4)$), and thanks to the following moment conditions,

$$\begin{aligned} \int_{\mathbb{R}^2} \exp(-|z|^2) dz &= \pi, \\ \int_{\Omega} z_i \exp(-|z|^2) dz &= 0, \\ \int_{\mathbb{R}^2} z_i z_j \exp(-|z|^2) dz &= 0 \quad (i, j = 1, 2, i \neq j), \\ \int_{\mathbb{R}^2} z_i^2 \exp(-|z|^2) dz &= \frac{\pi\sqrt{\pi}}{2}, \end{aligned}$$

we finally obtain (7). \square

3.3 Existence and Uniqueness of the Solution

In order to study the well-posedness of our model (3)–(4), let us consider the results presented in [29], for neural field equations. Note that in our case, the (3)–(4) do not exactly fit in the neural field formalism since the time-dependent input is used in a multiplicative way $k_1 p_2$ inside the sigmoid. As described in the previous section this term implements a modulating feedback diffusion. By applying the Cauchy–Lipschitz Theorem we show that the addition of such a multiplicative term to an activity-based neural field model maintains its well-posedness properties. First we check that the assumptions of the theorem are satisfied (Lemmas 2 and 3). Then, since the theorem proves existence and uniqueness of the solution on an open and bounded time interval, we show that this interval can be extended to the full half real line using a continuity argument (Theorem 2)

Let \mathcal{L} be the set $\mathbf{L}^2(\Omega \times \mathcal{V})$ of square integrable functions defined on the product set $\Omega \times \mathcal{V}$ and taking their values in \mathbb{R} , and $\mathcal{F} = \mathcal{L} \times \mathcal{L}$. The basic idea is to rewrite (3)–(4) as a differential equation defined on the set \mathcal{F} . With a slight abuse of notation we can write $p_i(t)(x, v) = p_i(t, x, v)$ and note $\mathbf{p} : \mathbb{R} \rightarrow \mathcal{F}$ the function defined by the following Cauchy problem:

$$\mathbf{p}(0) = \mathbf{p}_0 \in \mathcal{F}, \tag{8}$$

$$\mathbf{p}' = -\Lambda \mathbf{p} + \mathbf{S}(\mathbf{W}(t) \cdot \mathbf{p} + \mathbf{K}(t)), \tag{9}$$

with $\mathbf{p} = (p_1, p_2)^T$, $\mathbf{K} = (\lambda_1^f k_1, 0)^T$, and $\mathbf{S}(x_1, x_2) = (S_1(x_1), S_2(x_2))$. The operator \mathbf{W} is the 2×2 connectivity matrix function defined by the four linear mappings from \mathcal{L} to \mathcal{L} :

$$\begin{aligned}
W_{11} \cdot p &= -\lambda_1^l G_{\sigma_1^l}^{x,v} \star p + \lambda_1^d (G_{\sigma_1^d}^{x,v} \star p + p), \\
W_{12} \cdot p &= \lambda^b k_1 p, \\
W_{21} \cdot p &= \lambda_2^f G_{\sigma_2}^f \delta_v^x \star p, \\
W_{22} \cdot p &= -\lambda_2^l G_{\sigma_2^l}^{x,v} \star p + \lambda_2^d (G_{\sigma_2^d}^{x,v} \star p + p).
\end{aligned}$$

Functionally W_{11} and W_{22} correspond to lateral interactions in maps v1 and MT, W_{12} denotes the backward connection from MT to v1, and W_{21} denotes the forward integration from v1 to MT. In the following we note f the mapping defined by the right-hand side of (9):

$$f(t, \mathbf{p}) = -\Lambda \mathbf{p} + \mathbf{S}(\mathbf{W}(t) \cdot \mathbf{p} + \mathbf{K}(t)).$$

Note that the time dependence in the definition of f arises solely from the function k_1 that occurs in W_{12} and in \mathbf{K} . We prove the existence and uniqueness of a solution to (9) by proving (a) that f maps $I \times \mathcal{L}$ to \mathcal{L} where I is an open interval containing 0 and (b) that it is Lipschitz continuous with respect to the second variable. This allows us to apply the Cauchy–Lipschitz Theorem and to conclude that there is a unique maximal solution to (9), and that its interval of definition is an open interval $(-\alpha, \alpha)$ containing 0.

Lemma 2. *If $k_1(t)$ is measurable for all $t \in I$, f maps $I \times \mathcal{F}$ to \mathcal{F} .*

Proof. Let $\mathbf{p} = (p_1, p_2) \in \mathcal{F}$. If $k_1(t)$ is measurable for all $t \in I$, so is $W_{12} \cdot p_2$. All the other elements of $\mathbf{W} \cdot \mathbf{p}$ are simple or weighted sums (convolutions) of a measurable function \mathbf{p} and thus $\mathbf{W} \cdot \mathbf{p}$ is measurable. This implies that $\mathbf{S}(\mathbf{W}(t) \cdot \mathbf{p}(t) + \mathbf{K}(t))$ is in \mathcal{F} for all $t \in I$. \square

Lemma 3. *If $k_1(t)$ is measurable on $\Omega \times \mathcal{V}$ and bounded by \bar{k}_1 for all $t \in I$ the mapping f is Lipschitz continuous with respect to the second variable.*

Proof. We have

$$\begin{aligned}
\|f(t, \mathbf{p}) - f(t, \mathbf{q})\| &= \|-\Lambda(\mathbf{p} - \mathbf{q}) + \mathbf{S}(\mathbf{W}(t) \cdot \mathbf{p} + \mathbf{K}(t)) - \mathbf{S}(\mathbf{W}(t) \cdot \mathbf{q} + \mathbf{K}(t))\| \\
&\leq \max(\lambda_1, \lambda_2) \|\mathbf{p} - \mathbf{q}\| + S'_m \|\mathbf{W}(t) \cdot (\mathbf{p} - \mathbf{q})\|,
\end{aligned}$$

where S'_m is the maximum value taken by the derivatives of the sigmoids S_1 and S_2 . $\|\mathbf{W}(t) \cdot (\mathbf{p} - \mathbf{q})\|$ is upper-bounded by a constant times the sum of the four terms $\|W_{ij} \cdot (p_j - q_j)\|_{\mathcal{L}}$, $i, j = 1, 2$. Considering these terms we find two cases. The first case involves a convolution by a Gaussian is easily dealt with since:

$$\|G^{x,v} \star p\|_{\mathcal{L}} \leq k \|p\|_{\mathcal{L}} \quad \forall p \in \mathcal{L},$$

where the constant k depends on the Gaussian kernel. The second case concerns the multiplication by $k_1(t)$ in $W_{12}(t)$. Because of the hypothesis $K_1(t) p_2$ belongs to \mathcal{L} for all $t \in I$ and $\|k_1(t) p_2\| \leq \bar{k}_1 \|p_2\|_{\mathcal{L}}$. This completes the proof that f is Lipschitz continuous with respect to the second variable. \square

Theorem 1. *If $k_1(t)$ is measurable on $\Omega \times \mathcal{V}$ and bounded by \bar{k}_1 for all $t \in I$ there exists an open interval $J = (-\alpha, \alpha) \subset I$ centred at 0 such that the Cauchy problem (8–9) has a unique solution, hence is in $\mathcal{C}^1(J, \mathcal{F})$.*

Proof. Thanks to Lemmas 2 and 3 the conditions of the Cauchy–Lipschitz Theorem are satisfied. \square

Then, thanks to the sigmoids, it is easy to show that this solution is bounded.

Proposition 1. *The solution described in Theorem 1 is bounded for all $t \in J$*

Proof. The variation of constant formula yields:

$$\mathbf{p}(t) = e^{-\Lambda t} \mathbf{p}_0(t) + \int_0^t e^{-\Lambda(t-s)} \mathbf{S}(\mathbf{W}(s) \cdot \mathbf{p}(s) + \mathbf{K}(s)) ds,$$

for $t \in J$, from which it follows that

$$\begin{aligned} \|\mathbf{p}(t)\| &\leq \|e^{-\Lambda t}\| \|\mathbf{p}_0\| + \left\| \int_0^t e^{-\Lambda(t-s)} \mathbf{S}(\mathbf{W}(s) \cdot \mathbf{p}(s) + \mathbf{K}(s)) ds \right\| \\ &\leq e^{\max(\lambda_1, \lambda_2)\alpha} \|\mathbf{p}_0\| + \max\left(\frac{S_{1m}}{\lambda_1}, \frac{S_{2m}}{\lambda_2}\right) (e^{\max(\lambda_1, \lambda_2)\alpha} - 1) \\ &\leq e^{\max(\lambda_1, \lambda_2)\alpha} \left(\|\mathbf{p}_0\| + \max\left(\frac{S_{1m}}{\lambda_1}, \frac{S_{2m}}{\lambda_2}\right) \right), \end{aligned}$$

where S_{1m} and S_{2m} are the maximum values of the sigmoid functions S_1 and S_2 . \square

As in [29] we can extend this local result from $(-\alpha, +\alpha)$ to $(-\alpha, +\infty)$, assuming that the hypotheses on \mathbf{p}_0 in Theorem (1) are satisfied for $t \in (-\alpha, +\infty)$. Indeed, either $+\alpha = +\infty$ and the result is proved or there exists $0 < \beta < \alpha$ such that \mathbf{p} is not bounded for all $\beta \leq t < \alpha$, thereby obtaining a contradiction.

We summarize these results in the following theorem:

Theorem 2. *If $k_1(t)$ is measurable on $\Omega \times \mathcal{V}$ and bounded by \bar{k}_1 for all $t \in (-\alpha, +\infty)$ the Cauchy problem (8–9) has a unique bounded solution, hence in $\mathcal{C}^1((-\alpha, +\infty), \mathcal{F})$.*

3.4 Relations to the State of the Art in Computer Vision

One essential aspect of the neural field framework lies in the definition of interaction between populations through an integral form. Interestingly, under some assumptions, one can write relations between integral operators (acting in a neighbourhood) and differential operators (acting very locally). This question was investigated by [20, 23, 24] and further extended by [59]. In these papers, the authors show the correspondence between linear elliptic differential operators and their integral approximation. This idea has also been considered for nonlinear operators by

[4, 14]. Thus, one can see a direct relation between the neural field framework and PDE-based approaches.

As such, introducing the neural field framework for motion estimation can be related to the series of papers proposing PDE-based approaches for optical flow estimation, starting from [34]. In computer vision, this seminal work has been further improved by many authors such as [9, 25, 45, 62]. Improvements concern mainly the definition of the regularization term, which is how diffusion performs. In this class of approaches, since diffusion is defined by differential operators, the aperture problem is solved by local diffusion processes.

Here, using the neural field framework, we offer the possibility to define different kinds of connectivity patterns not necessarily corresponding to differential operators. More generally, for modelling in computer vision, the neural field formalism has two main advantages over PDE-based approaches: (a) The first advantage is that non-local interactions can be defined, which is not possible with classical PDE or variational approaches defining the interactions between neighbors through differential operators. (b) The second advantage is to naturally describe interactions between several maps. In our article, the two maps correspond to two scales of analysis, thus providing a *dynamical* multiscale analysis.

4 Results

4.1 Implementation Details

As far as implementation is concerned, spaces have to be discretized, including the velocity space \mathcal{V} . We chose $\mathcal{V} = \{-5, -4.5, \dots, 4.5, 5\}^2$ to sample the velocities on a grid of size 21×21 .

The model defined by (3)–(4) is fully specified by a set of fourteen parameters.¹ These parameters were tuned by matching the time scale dynamics of the simple translating bar stimulus presented latter in Fig. 6. Integration was performed using a 4th order Runge–Kutta method with at most ten iterations between two frames.

Since our distributed motion representation can be hard to analyse, and to facilitate comparisons with computer vision approaches, we estimate an optical flow m_i by averaging at each position the population response across all velocities [7]:

$$m_i(t, x) = \frac{\sum_{v \in \mathcal{V}} p_i(t, x, v) v}{\sum_{v \in \mathcal{V}} p_i(t, x, v)}, \quad i \in \{1, 2\}. \quad (10)$$

Then the optical flow is represented either by arrows or by a color coded image indicating speed and direction. We used the Middlebury color code [5], which emerged as the de facto standard in the optical flow computer vision community.

¹ Parameters chosen for the experiments: $\lambda_1 = 2$, $\lambda_1^f = 1$, $\lambda_1^b = 24$, $\lambda_1^l = 4$, $\sigma_1 = 2$, $\lambda_2 = 2$, $\lambda_1^d = 6$, $\lambda_2^f = 16$, $\lambda_2^l = 4$, $\sigma_2 = 2$, $\sigma_2^f = 8$, $\lambda_2^d = 10$, $\sigma_1^d = 2$, $\sigma_2^d = 10$.

The direction of the velocity corresponds to the hue, for instance yellow for downward velocities, while the speed of the velocity is encoded in the saturation, whiter (or less saturation) for slower speeds. The color code is illustrated in Fig. 2a.

4.2 Natural Scenes

We consider two classical videos from the computer vision (see Fig. 2): (a) The Hamburg taxi sequence, a recorded sequence where three cars and a pedestrian are moving; (b) The Yosemite sequence where the optical flow covers the whole spatial domain, and for which the ground truth is available.



Fig. 2 Experimental protocol. (a) Middlebury color disk mapping motion direction to hue and speed to saturation, and tested videos: (b) Hamburg taxis sequence with three cars moving; (c) Yosemite sequence with clouds [6]

Results for the Hamburg taxi sequence are shown in Fig. 3. We show how our model improves the initial noisy optical flow estimation obtained from the modified Reichardt detectors described in Sect. 3.2.2. The evolution of the optical flow between k_1 , p_1 and p_2 is shown in Figs. 3a, b, c. Note that since the output of the Reichardt motion detectors is unable to handle the borders, we replace the information on the border by a small identical activity on all velocities. When the optical flow is computed, the replacement leads to a zero velocity on the borders represented in Figs. 3a, b as a white frame. Due to the recurrent interactions in our model, those borders tend to be filled in when the motion is strong enough (such as the rightward moving car in Fig. 3b). Such a filling in mechanism is particularly useful for dense optical flows since it nicely reconstructs the motion at the borders.

Results for the Yosemite sequence are shown in Fig. 4. The optical flow m_2 estimated from p_2 is shown in Fig. 4a, with its associated angular error in Fig. 4b. The angular error is defined by:

$$\arccos \left(\frac{u_1 v_1 + u_2 v_2 + 1}{\sqrt{u_1^2 + u_2^2 + 1} \sqrt{v_1^2 + v_2^2 + 1}} \right)$$

where $u = (u_1, u_2)$ is the correct velocity and $v = (v_1, v_2)$ is the estimated velocity. In our case, the average error is 3.97° which is in the range of baseline results from [5]. Note that this error evolves in time (see Fig. 5) and that this average was estimated at convergence. Then it is important to mention that a large region of high angular error is located on subpixel velocities. One can explain such an error by the peculiar velocity space which offers poor angular resolution at low velocities. A coarser velocity space such as $\mathcal{V}' = \{-5, -4, \dots, 4, 5\}^2$ used in [56] would lead to a worse angular error as shown in Fig. 4d (average angular error 6.48°). To explain this, we show in Fig. 4c the norm of the ground truth for the Yosemite sequence: A subpixel definition of the velocity space is necessary because of the continuity of the optical flow. The influence of the diffusion term on the smoothness of the solution is illustrated in Fig. 4e where we set the diffusion to zero ($\lambda_{1,2}^d = 0$). Removal of the diffusion leads to spatial patches of selected velocities which increase the angular error. Motion within individual patches has the tendency to represent one selected motion direction. Consequently, the angular error is high inside those patches where only one velocity is activated, but low at their borders where multiple velocities simultaneously exist (see Fig. 4f).

In Fig. 5 we show the evolution of the average angular error (AAE) for each frame of the Yosemite sequence. We observe that the convergence is not reached between the first pair of frames (as in classical computer vision methods) simply because we limited the number of iterations between two frames. Indeed, we designed our model to reproduce motion integration dynamics and this exponential decay of the error is very important in psychophysics (see Sect. 4.3).

4.3 Psychophysical Stimulus

The dynamics of motion integration can be well characterized by a simple translating bar stimulus (see Fig. 6a). It has been shown experimentally that the perceived direction is initially biased towards the direction orthogonal to the bar orientation, and that this perceptual bias is corrected for longer durations [16]. This early bias was also shown in smooth pursuit for humans [41, 60] and monkeys [12].



Fig. 3 Result for the Hamburg taxi sequence. (a) Optical flow estimated from the Reichardt detectors k_1 . (b) Optical flow m_1 . (c) Optical flow m_2 . Note the filling in of the *left margin* due to the activity evoked by the leftward moving car

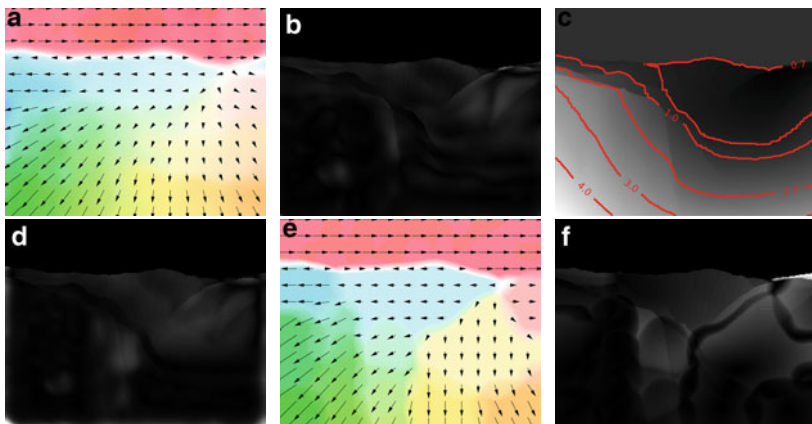


Fig. 4 Results for the Yosemite sequence. (a) Optical flow m_2 . (b) Associated angular error. (c) Norm of the ground truth with iso-contours plotted for some values. (d) Same as b but with the coarser discrete velocity space \mathcal{V}' . (e) Same as b but when diffusion terms are removed: patches appear. (f) Angular error corresponding to the optical flow shown in e : the error is low at the borders of the patches

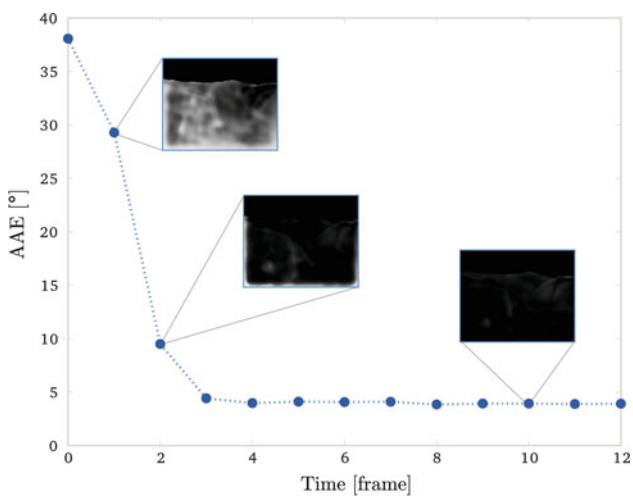


Fig. 5 Dynamical evolution of the average angular error (AAE) on the Yosemite sequence

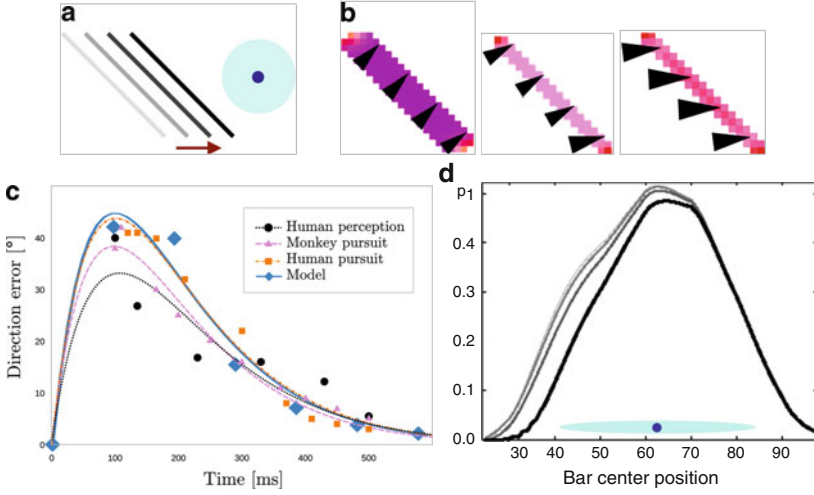


Fig. 6 Results for the translating bar. **(a)** The stimulus is a rightward translating bar, tilted at 45° . The *blue area* describes the classical receptive field of the p_1 population whose activity will be shown in **d** for several initial bar positions (indicated by different *gray levels*). **(b)** Optical flow m_2 at different times (100, 200 and 600 ms). **(c)** Temporal dynamics of the observed direction error for human perceived direction, human and macaque tracking direction, as well as our model. Data is reproduced from [12, 40, 60]. Both discrete measurements and best fit are shown (each data set was fitted by $f(t) = A \frac{x}{\tau} \exp(-\frac{x}{\tau})$). **(d)** Activity of the p_1 population indicated in **a** and tuned to the true bar velocity, for the different initial bar positions shown in **a**. This shows that when the bar is starting more far away, the activity starts earlier

In order to compare psychophysical dynamics to our results, let us define the perceived direction (from the population activity) as a read-out. The perceived direction $w(t) \in \mathbb{R}^2$ is defined as a unique velocity per frame and corresponding to the global motion. In this article, we defined it simply by averaging the activity of the population p_2 , with a temporal smoothing defined by the following dynamical equation:

$$\frac{dw}{dt}(t) = \lambda \left(\sum_{x \in \Omega} m_2(t, x) - w(t) \right), \quad (11)$$

where m_2 is defined by (10).

In Fig. 6b we show the optical flow m_2 that we obtained at different times (here we assume that the images are sampled every 100 ms). Then, using formula (11), we show in Fig. 6c the estimated direction error defined by the angular difference between the perceived direction and the true direction of the object. After a short period of time where the direction error reaches 45° , the perceived direction converges to the true direction with an exponential decay. The dynamics that we observe

in our model closely reproduce the experimental data measured for both pursuit and perception.

Note that the dynamics we observe here is not reproduced by any classical computer vision approach. Since their goal is generally to estimate the motion between two consecutive frames, the result is the same independently of the time at which the optical flow is estimated.

4.3.1 Motion Anticipation

In Fig. 6d we show the response of a V1 population in our model responding to a translating bar stimulus. We repeat the experiment several times, placing the initial bar position further away from the receptive field location of the recorded neuron. The further the initial bar position is from the V1 receptive field, the sooner its response starts. Note that this anticipation of the V1 neuron is equivalent to a spatial deformation of the receptive field towards the initial bar position. This change does not affect the late response of the population. The population stops firing at the same time in each experiment, that is when the bar leaves the receptive field.

Such behaviour has been observed at different levels of the mammals visual system. For instance, comparing moving with flashed bars, Berry et al. [8] found anticipatory responses in rabbit retinal ganglion cells when motion was present. Similar results were obtained in population reconstruction from spiking activity of cat area 17 neurons [38] as well as sub threshold responses as recorded with optical imaging [36, 37]. In monkeys, [33] found that most V1 neurons were strongly modulated by events prior to and distant from stimulation of their classical receptive fields. More recently, the group of one of us showed that spiking activity of V1 neurons exhibit similar anticipatory responses to moving bars in awake, fixating monkeys and that such anticipatory response was modulated by the distance travelled by the bar before entering inside the classical receptive field. These preliminary observations are well in line with the neural dynamics observed in our model. Thus, neural field models like ours can reproduce several keys aspects of multi-layered, densely interconnected biological systems which act more as active analyzer than mere representation of the inputs. Several other aspects such as motion extrapolation could be tested and the detailed dynamics of these neuronal activities could be used to tune both timing and spatial distributions of lateral and recurrent connectivities of the network.

5 Conclusion

In this article, we showed how biology can be a source of inspiration, focusing on the two following questions: (a) How to design a model taking into account biology? (b) What could be a suitable mathematical framework?

We showed how to start from the state-of-the-art in neuroscience concerning the motion pathway, as the source of inspiration to define our model. So far, it is not

possible to retain the full complexity of the cortical architecture, but our goal was to identify which key features should be incorporated into our model in order to reach a good compromise between bio-inspiration and computational efficiency.

Then, we chose a suitable mathematical formalism, namely the neural field formalism, in order to write the core equations of our model. Choosing this formalism has a biological interpretation: We focus on the macroscopic scale by defining the cortical activity at the population level. Since the number of neurons and synapses even in a small piece of cortical area is immense, we can assume that the relevant observable quantity is at the level of the population and not at the single cell level. From a computer vision point of view, the proposed neural field formalism, based on integral equations, has some interesting relationships with PDE-based approaches. Here we show that the neural field framework can successfully handle complex computer vision problems like motion estimation, and at the same time it offers a new well-posed framework, which can bring new ideas into the community.

Perspectives around this work are promising and cover (a) improving the model, and (b) using the power of the neural field formalism more effectively. (c) considering novel kinds of evaluation methodologies.

First, the proposed model can of course be extended in order to explain other biological phenomena. Among the possible extensions, we expect that the addition of delays [22, 30] will be essential to constrain the propagation speed and account for more experimental data.

Second, the theoretical analysis of such equations is expected to have profound impact on our knowledge of the human visual system. For instance, delay equations allow one to distinguish between intra-cortical and extra-cortical interactions, due to their different speeds, and allows one to gain new insights into the mechanisms of the primate cortex. Another example is the application of the bifurcation theory to the proposed models in order to explain multi-stable percepts.

Finally, evaluating motion models only on real scenes is certainly not enough if one claims that his model is bio-inspired. One has to take into account the human visual performance to characterize the bio-plausibility. Such a bio-inspired benchmark was initiated in [55] where scoring procedures and baseline results are proposed and it will be very interesting to investigate further experimental results to find criteria to evaluate models.

Acknowledgements This research work received funding from the Région PACA, the CNRS, the European Community (through FACETS, IST-FET, Sixth Framework, No 025213, and SEARISE, Seventh Framework, No 215866), the ERC grant No 227747 (NERVI) and the Agence Nationale de la Recherche (ANR, NATSTATS).

References

1. Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A* **2**, 284–299 (1985)
2. Amari, S.-I.: Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.* **27**(2), 77–87 (1977)

3. Angelucci, A., Bullier, J.: Reaching beyond the classical receptive field of v1 neurons: horizontal or feedback axons? *J. Phys. Paris* **97**(2–3), 141–154 (2003)
4. Aubert, G., Kornprobst, P.: Can the nonlocal characterization of sobolev spaces by bourgain et al. be useful to solve variational problems? *SIAM J. Numer. Anal.* **47**(2), 844–860 (2009)
5. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: *International Conference on Computer Vision, ICCV'07*, pp. 1–8. (2007)
6. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. *Int. J. Comput. Vis.* **12**(1), 43–77 (1994)
7. Bayerl, P., Neumann, H.: Disambiguating visual motion through contextual feedback modulation. *Neural Comput.* **16**(10), 2041–2066 (2004)
8. Berry, M.J., Brivanlou, I.H., Jordan, T.A., Meister, M.: Anticipation of moving stimuli by the retina. *Nature* **398**(6725), 334–338 (1999)
9. Black, M.J., Rangarajan, P.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vis.* **19**(1), 57–91 (1996)
10. Born, R.T.: Visual processing: parallel-er and parallel-er. *Curr. Biol.* **11**(14), R566–R568 (2001)
11. Born, R.T., Bradley, D.C.: Structure and function of visual area MT. *Annu. Rev. Neurosci.* **28**, 157–189 (2005)
12. Born, R.T., Pack, C.C., Ponce, C., Yi, S.: Temporal evolution of 2-dimensional direction signals used to guide eye movements. *J. Neurophysiol.* **95**, 284–300 (2006)
13. Bradley, D.C., Goyal, M.S.: Velocity computation in the primate visual system. *Nat. Rev. Neurosci.* **9**(9), 686–695 (2008)
14. Buades, A., Coll, B., Morel, J.-M.: Neighborhood filters and PDE's. *Numer. Math.* **105**(1), 1–34 (2006)
15. Carandini, M., Demb, J.B., Mante, V., Tollhurst, D.J., Dan, Y., Olshausen, B.A., Gallant, J.L., Rust, N.C.: Do we know what the early visual system does? *J. Neurosci.* **25**(46), 10577–10597 (2005)
16. Castet, E., Lorenceau, J., Shiffrar, M., Bonnet, C.: Perceived speed of moving lines depends on orientation, length, speed and luminance. *Vis. Res.* **33**, 1921–1921 (1993)
17. Chey, J., Grossberg, S., Mingolla, E.: Neural dynamics of motion processing and speed discrimination. *Vis. Res.* **38**, 2769–2786 (1997)
18. Churchland, P.S., Sejnowski, T.J.: *The Computational Brain*. MIT, Cambridge (1992)
19. Coombes, S.: Waves, bumps, and patterns in neural fields theories. *Biol. Cybern.* **93**(2), 91–108 (2005)
20. Cottet, G.-H., El Ayyadi, M.: A Volterra type model for image processing. *IEEE Trans. Image Process.* **7**(3) (1998)
21. Dayan, P., Abbott, L.F.: *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. MIT, Cambridge (2001)
22. Deco, G., Roland, P.: The role of multi-area interactions for the computation of apparent motion. *NeuroImage* **51**(3), 1018–1026 (2010)
23. Degond, P., Mas-Gallic, S.: The weighted particle method for convection-diffusion equations. *Math. Comput.* **53**(188), 485–525 (1989)
24. Edwards, R.: Approximation of neural network dynamics by reaction-diffusion equations. *Math. Methods Appl. Sci.* **19**, 651–677 (1996)
25. Enkelmann, W.: Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Comput. Vis. Graph. Image Process.* **43**, 150–177 (1988)
26. Ermentrout, B.: Neural networks as spatio-temporal pattern-forming systems. *Rep. Prog. Phys.* **61**, 353–430 (1998)
27. Escobar, M.-J.: *Bio-Inspired Models for Motion Estimation and Analysis: Human action recognition and motion integration*. Ph.D. thesis, Université de Nice Sophia-Antipolis (2009)
28. Escobar, M.-J., Masson, G.S., Viéville, T., Kornprobst, P.: Action recognition using a bio-inspired feedforward spiking network. *Int. J. Comput. Vis.* **82**(3), 284 (2009)
29. Faugeras, O., Grimbert, F., Slotine, J.-J.: Absolute stability and complete synchronization in a class of neural fields models. *SIAM J. Appl. Math.* **61**(1), 205–250 (2008)

30. Faye, G., Faugeras, O.: Some theoretical and numerical results for delayed neural field equations. *Phys. D* **239**(9), 561–578 (2010). Special issue on Mathematical Neuroscience
31. Grinvald, A., Lieke, E.E., Frostig, R.D., Hildesheim, R.: Cortical point-spread function and long-range lateral interactions revealed by real-time optical imaging of macaque monkey primary visual cortex. *J. Neurosci.* **14**(5), 2545–2568 (1994)
32. Grossberg, S., Mingolla, E.: Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading. *Psychol. Rev.* **92**(2), 173–211 (1985)
33. Guo, K., Robertson, R.G., Pulgarin, M., Nevado, A., Panzeri, S., Thiele, A., Young, P.: Spatio-temporal prediction and inference by v1 neurons. *Eur. J. Neurosci.* **26**, 1045–1054 (2007)
34. Horn, B.K., Schunck, B.G.: Determining Optical Flow. *Artif. Intell.* **17**, 185–203 (1981)
35. Hupé, J.M., James, A.C., Payne, B.R., Lomber, S.G., Girard, P., Bullier, J.: Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons. *Nature* **394**, 784–791 (1998)
36. Jancke, D., Chavane, F., Grinvald, A.: Stimulus Localization by Neuronal Populations in Early Visual Cortex: Linking Functional Architecture to Perception, chapter 5, pp. 95–116. *Neuronal, Behavioral, and Computational Approaches*, 1 edn. Springer, Berlin (2010)
37. Jancke, D., Chavane, F., Naaman, S., Grinvald, A.: Imaging cortical correlates of illusion in early visual cortex. *Nature* **428**, 423–426 (2004)
38. Jancke, D., Erlaghen, W., Schöner, G., Dinse, H.R.: Shorter latencies for motion trajectories than for flashes in population responses of primary visual cortex. *J. Physiol.* **556**, 971–982 (2004)
39. Löffler, G., Orbach, H.S.: Computing feature motion without feature detectors: A model for terminator motion without end-stopped cells. *Vis. Res.* **39**(4), 859–871 (1998)
40. Lorenceau, J., Shiffrar, M., Wells, N., Castet, E.: Different motion sensitive units are involved in recovering the direction of moving lines. *Vis. Res.* **33**, 1207–1207 (1993)
41. Masson, G.S., Stone, L.S.: From following edges to pursuing objects. *J. Neurophysiol.* **88**(5), 2869–2873 (2002)
42. Montagnini, A., Mamassian, P., Perrinet, L., Castet, E., Masson, G.S.: Bayesian modeling of dynamic motion integration. *J. Physiol. Paris* **101**(1–3), 64–77 (2007)
43. Nassi, J.J., Callaway, E.M.: Parallel processing strategies of the primate visual system. *Nat. Rev. Neurosci.* **10**(5), 360–372 (2009)
44. Nassi, J.J., Callaway, E.E.M.: Multiple circuits relaying primate parallel visual pathways to the middle temporal area. *J. Neurosci.* **26**(49), 12789–12798 (2006)
45. Nir, T., Bruckstein, A.M., Kimmel, R.: Over-parameterized variational optical flow. *Int. J. Comput. Vis.* **76**(2), 205–216 (2008)
46. Nowlan, S.J., Sejnowski, T.J.: Filter selection model for motion segmentation and velocity integration. *J. Opt. Soc. Am. A* **11**(12), 3177–3199 (1994)
47. Pinto, D.J., Brumberg, J.C., Simons, D.J., Ermentrout, G.B., Traub, R.: A quantitative population model of whisker barrels: Re-examining the wilson-cowan equations. *J. Comput. Neurosci.* **3**(3), 247–264 (1996)
48. Priebe, N.J., Lisberger, S.G., Movshon, A.J.: Tuning for spatiotemporal frequency and speed in directionally selective neurons of macaque striate cortex. *J. Neurosci.* **26**(11), 2941–2950 (2006)
49. Reichardt, W.: Autokorrelationsauswertung als Funktionsprinzip des Zentralnervensystems. *Z. Naturforsch.* **12**, 447–457 (1957)
50. Rockland, K.S., Knutson, T.: Feedback connections from area MT of the squirrel monkey to areas V1 and V2. *J. Comp. Neurol.* **425**(3), 345–368 (2000)
51. Rust, N.C., Mante, V., Simoncelli, E.P., Movshon, J.A.: How MT cells analyze the motion of visual patterns. *Nat. Neurosci.* **9**, 1421–1431 (2006)
52. Shiffrar, M., Lorenceau, J.: Increased motion linking across edges with decreased luminance contrast, edge width and duration. *Vis. Res.* **36**(14), 2061–2067 (1996)
53. Sillito, A.M., Cudeiro, J., Jones, H.E.: Always returning: feedback and sensory processing in visual cortex and thalamus. *Trends Neurosci.* **29**(6), 307–316 (2006)
54. Simoncelli, E.P., Heeger, D.J.: A model of neuronal responses in visual area MT. *Vis. Res.* **38**, 743–761 (1998)

55. Tlapale, É., Kornprobst, P., Bouecke, J.D., Neumann, H., Masson, G.S.: Evaluating motion estimation models from behavioral and psychophysical data. In: *Bionetics* (2010)
56. Tlapale, É., Masson, G.S., Kornprobst, P.: Modelling the dynamics of motion integration with a new luminance-gated diffusion mechanism. *Vis. Res.* **50**(17), 1676–1692 (2010)
57. Van Santen, J.P.H., Sperling, G.: Elaborated reichardt detectors. *J. Opt. Soc. Am. A* **2**(2), 300–320 (1985)
58. Veltz, R., Faugeras, O.: Local/global analysis of the stationary solutions of some neural field equations. *SIAM J. Appl. Dyn. Syst.* **9**(3), 954–998 (2010)
59. Viéville, T., Chemla, S., Kornprobst, P.: How do high-level specifications of the brain relate to variational approaches? *J. Physiol. Paris* **101**(1–3), 118–135 (2007)
60. Wallace, J.M., Stone, L.S., Masson, G.S.: Object motion computation for the initiation of smooth pursuit eye movements in humans. *J. Neurophysiol.* **93**(4), 2279–2293 (2005)
61. Watson, A.B., Ahumada, A.J.: Model of human visual-motion sensing. *J. Opt. Soc. Am. A* **2**(2), 322–342 (1985)
62. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging Vis.* **14**(3), 245–255 (2001)
63. Weiss, Y., Adelson, E.H.: Slow and smooth: A Bayesian theory for the combination of local motion signals in human vision. *Center for Biological and Computational Learning Paper*. (1998)
64. Wilson, H.R., Cowan, J.D.: Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* **12**, 1–24 (1972)
65. Wilson, H.R., Cowan, J.D.: A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Biol. Cybern.* **13**(2), 55–80 (1973)
66. Wilson, H.R., Ferrera, V.P., Yo, C.: A psychophysically motivated model for two-dimensional motion perception. *Vis. Neurosci.* **9**(1), 79–97 (1992)
67. Yo, C., Wilson, H.R.: Perceived direction of moving two-dimensional patterns depends on duration, contrast and eccentricity. *Vis. Res.* **32**(1), 135–147 (1992)
68. Yuille, A.L., Grzywacz, N.M.: A winner-take-all mechanism based on presynaptic inhibition feedback. *Neural Comput.* **1**(3), 334–347 (1989)

Non-Local Regularization and Registration of Multi-Valued Images By PDE's and Variational Methods on Higher Dimensional Spaces

David Tschumperlé and Luc Brun

Abstract We are interested in a simple transform that map any regular $2D$ multi-valued image into a high-dimensional Euler space of patches, so that each initial image patch is mapped as a single high-dimensional point. We propose a way to study the local geometry of this patch-projected image and we consider variational formulations and PDE's in this particular space. Hence, we define a way to find natural patch-based counterparts of some classical image processing techniques, including *Lucas–Kanade* registration, *Tikhonov* regularization and tensor-driven anisotropic diffusion PDE's. As a result, we end up with noteworthy variants of already known (non-variational) patch-based algorithms, namely the *Non Local Means* and *Block Matching* techniques. The interest of considering such variational or PDE approaches on high dimensional patch spaces is discussed and illustrated by comparison results with corresponding non-variational or non-patch methods.

1 Introduction

In the fields of Image Analysis, Processing and Synthesis, patch-based techniques generally meet with success. Defined as local square neighborhoods of image pixels, patches are very simple objects to work with, but they have the intrinsic ability to catch large-scale structures and textures present in natural images. Patches provide one of the simplest way to analyze, compare and copy textures, as soon as the considered patch sizes are higher than the so-called *texel* sizes (texture element, seen as the smallest significant unit of a texture). Moreover, patch-based methods are somehow intuitive: They indirectly reproduce the way humans are accomplishing some vision tasks by comparing semi-local image neighborhoods together. Proposed algorithms are often simple to implement, but they give surprisingly good results.

David Tschumperlé (✉) and Luc Brun
Image Team, Laboratoire GREYC, 6 Bd Maréchal Juin, 14050 Caen Cedex/France
e-mail: [David.Tschumperle,Luc.Brun]@greyc.ensicaen.fr

For instance, it has been long since patches have been used for solving the problem of estimating a displacement field between two images. This alignment problem can be classically (and partially) solved by the so-called *Block Matching* algorithm, which consists in comparing each neighborhood (patch) of one image with all of the other one, finding the best match for each pixel location. Unfortunately, the regularity of the resulting motion field is not ensured by such a crude method and regularized techniques [4, 19] are sometimes needed. More recently, patch-based algorithms have been proposed to tackle the problem of synthesizing a texture similar to an input model [1, 12, 28]. It has been shown that this quite complicated task can be very well achieved by a simple iterated copy-paste procedure of different patches coming from the model, found to fit the best with the local neighborhoods of the image to synthesize. Some variants of these techniques have been also applied for transferring textures from an image to another one [2, 15], and for textured inpainting [11, 14], i.e. the reconstruction of missing textured regions in an image. Despite their relative simplicity, one must admit that these algorithms give outstanding results. It is also worth to cite the set of recent patch-based denoising methods, initiated with the *Non Local Means* scheme [10] and continued with various derivatives [8, 9, 16]. Such methods are mainly based on an iterative weighted average of image patches. Here again, they have rapidly entered the hall of fame of image denoising techniques.

On the other side, variational methods for image processing [3, 23] are known to be mathematically well-posed, flexible and competitive: They have the great ability to put complex a priori constraints (such as regularity constraints) on the obtained solutions. While regularity is desired for many applications (e.g. for motion estimation), it may sometimes avoid the reconstruction of textured solutions (e.g. for denoising/inpainting), textures being oscillating patterns by nature. Few attempts have been made to incorporate textured features in specific variational denoising/inpainting formulations by putting explicit patches [17] or various image transforms [13] into the functionals to minimize, with few success. Mixing the performance of patches and the flexibility of variational methods is still a very exciting goal and an open challenge.

In this paper, we are looking toward a more general strategy to find patch-based counterparts of image processing techniques expressed as variational problems. Mainly, the idea lies on the construction of an alternate patch space on which the image to process/analyze is mapped. Thus, in this high-dimensional space, each existing image patch is represented by a single point. Variational formulations can be then defined and corresponding PDE's solved directly on the patch space, projecting back the obtained solution to the original image domain, if needed. So, instead of explicitly incorporating image patches into ad-hoc energy functionals, we rather consider the extension of existing classical (non-patch) functionals to a higher-dimensional space of patches. This is quite straightforward: Energy functionals and PDE's are generally expressed with terms that can be easily extended for an arbitrary number of dimensions (e.g. gradients). As a result, we obtain algorithms which are the natural patch-based counterparts of "pointwise" variational techniques.

This paper is organized as follows: First, we define the reversible mapping of a multi-valued image into its patch space (Sect. 2). Then, in this space, we consider the minimization of the *Tikhonov* regularization functional [7, 25] (Sect. 3). We show that the natural patch-based version of the resulting minimizing flow can be interpreted as a variant of the *Non Local Means* filter [10]. In a second attempt, we extend the corresponding high-dimensional heat flow to more sophisticated diffusion processes directed by fields of diffusion tensors. Finally, we tackle the problem of image alignment similarly, by minimizing an energy functional inspired by the *Lucas–Kanade* method [4, 19]. We end up with an interesting variational version of the *Block Matching* algorithm, implemented by the evolution of nonlinear PDE's. Application results and discussions on possible future applications of this general variational framework on patch spaces conclude this paper (Sects. 7–8).

2 Definition of the Patch Space

Let us consider a 2D noisy image $\mathbf{I} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$ (here $n = 3$, for color images). The i th component of a vector or image \mathbf{X} will be denoted as X_i . We define a patch $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ as the set of all image values belonging to a spatially discretized local $p \times p$ neighborhood of \mathbf{I} , centered at (x, y) . For simplification sake, the spatial discretization step of a patch is assumed to be 1. The dimension p is considered as odd, i.e. $p = 2q + 1$ ($q \in \mathbb{N}$). Actually, a patch $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ can be ordered as a np^2 -dimensional vector as:

$$\mathcal{P}_{(x,y)}^{\mathbf{I}} = (I_{1(x-q,y-q)}, \dots, I_{1(x+q,y+q)}, I_{2(x-q,y-q)}, \dots, I_{n(x+q,y+q)})$$

A multi-valued patch $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ is thus the concatenation of the scalar patches $\mathcal{P}_{(x,y)}^{I_i}$ for all image channels $i = 1 \dots n$. The mathematical study of the whole manifold geometry formed by these image patches has been initiated in [21]. Here we will rather consider this patch-based geometry of the image at a more local scale.

2.1 Mapping in the Patch Space

We define the $(np^2 + 2)$ -dimensional patch space $\Gamma = \Omega \times \mathbb{R}^{np^2}$. Each point \mathbf{p} of Γ is a high-dimensional vector which contains both informations of any (x, y) coordinates in Ω and of all values of any $p \times p$ patch \mathcal{P} in \mathbb{R}^{np^2} . Obviously, we want to highlight the points $\mathbf{p} = (x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}})$ in Γ , i.e. the locations which precisely correspond to existing patches in \mathbf{I} (a.k.a. the *located patches* of \mathbf{I}). We define the function $\tilde{\mathbf{I}}$ in Γ such that $\tilde{\mathbf{I}}(\mathbf{p})$ is non zero only for these located patches: $\tilde{\mathbf{I}} : \Gamma \rightarrow \mathbb{R}^{np^2+1}$, s.a. $\forall \mathbf{p} \in \Gamma$,

$$\tilde{\mathbf{I}}(\mathbf{p}) = \begin{cases} (\mathbf{I}_{(x,y)}, 1) & \text{if } \mathbf{p} = (x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}}) \\ \mathbf{0} & \text{elsewhere} \end{cases} \quad (1)$$

The application \mathcal{F} such that $\tilde{\mathbf{I}} = \mathcal{F}(\mathbf{I})$ computes a patch-based representation of \mathbf{I} . Note that the value space of $\tilde{\mathbf{I}}$ has an extra component set to 1 at the located patches of \mathbf{I} . This is comparable to what is usually done when dealing with projective spaces: This value plays a role of weighting action when inverting \mathcal{F} , i.e. retrieving back \mathbf{I} from $\tilde{\mathbf{I}}$. Intuitively, it defines how much a patch in Γ is *meaningful*, and by default all located patches of the initial image have the same importance. Note that $\tilde{\mathbf{I}}$ is a discontinuous multi-valued function. To avoid further derivation problems, we will in fact consider $\tilde{\mathbf{I}}_\varepsilon = \mathbf{I} * G_\varepsilon$, a continuous version of $\tilde{\mathbf{I}}$, where each patch $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ of \mathbf{I} is mapped as a normalized Gaussian function G_ε with a variance ε close to 0, instead of a single Dirac point in Γ .

2.2 Back-Projection on the Image Domain

Due to the high dimensionality of Γ , there are of course no unique ways to invert the patch transform $\tilde{\mathbf{I}} = \mathcal{F}(\mathbf{I})$. We define a back-projection method based on two steps : First, we retrieve the most significant patch $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ in Γ for every location (x,y) in Ω .

This is the one with the maximum projective weight, i.e.

$$\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}} = \operatorname{argmax}_{\mathbf{q} \in \mathcal{R}^{np^2}} \tilde{I}_{np^2+1}(x,y,\mathbf{q}) \quad (2)$$

Note that if one perturbs only slightly the patch representation $\tilde{\mathbf{I}}$ of \mathbf{I} , one will likely find that the significant patches $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ stays at the same locations $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ as the original ones, even though the pixel values of these most significant patches may have been modified. This is one of the main interest of this back-projection method : from a practical point of view, we really do not need to do this exhaustive search into the patch space (which would be really long to perform), since we can approximate the coordinates of $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ to be at the same location than the initial (non-filtered) corresponding patch.

In a second step, the back-projected image $\hat{\mathbf{I}}$ is reconstructed by combining the most significant patches together. Here, we use the simplest possible strategy, i.e. copy the normalized center pixel of each $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ at its location (x,y) :

$$\forall (x,y) \in \Omega, \hat{I}_{i(x,y)} = \frac{\tilde{I}_{ip^2 + \frac{p^2+1}{2}}(x,y, \mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}})}{\tilde{I}_{np^2+1}(x,y, \mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}})} \quad (3)$$

We could also have copied an entire sub-patch of $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ while overlapping neighborhood patches according to their relative weights. These patch copying/averaging considerations frequently appear in the patch-based algorithm literature. The best pick probably depends on the targeted application.

This simple patch transform allows to project at the same time the local and non-local structural informations of an image \mathbf{I} onto an Euclidean space Γ where nearby and non-zero points correspond to patches that are similar in values (with

respect to the L_2 norm) or are spatially grouped. Therefore, applying a local process (by nature) on Γ followed by a back-projection procedure is a quite elegant way of embedding it with non-local properties. This is what we aim to do in the followings, with the proposal of image smoothing and registration processes based on the evolution of PDE's on Γ .

3 Image Denoising by Patch-Based Tikhonov Regularization

Suppose we have a multi-valued image $\mathbf{I}^{noisy} : \Omega \rightarrow \mathbb{R}^n$ corrupted by some kind of noise. So will be its patch transform $\tilde{\mathbf{I}}^{noisy}$. We are looking for a patch-based minimizing flow able to regularize $\tilde{\mathbf{I}}^{noisy}$ rather than process directly \mathbf{I}^{noisy} . For this purpose, we minimize the following energy E_1 , classically denoted as the *Tikhonov* regularization functional, which have been simply extended to the high-dimensional space Γ :

$$E_1(\tilde{\mathbf{I}}) = \int_{\Gamma} \|\nabla \tilde{\mathbf{I}}_{(\mathbf{p})}\|^2 d\mathbf{p} \quad (4)$$

where $\|\nabla \tilde{\mathbf{I}}_{(\mathbf{p})}\| = \sqrt{\sum_{i=1}^{np^2+1} \|\nabla \tilde{I}_i(\mathbf{p})\|^2}$ is the habitual extension of the gradient norm for multi-valued datasets [30]. Note that this multi-valued gradient includes also the gradient $\|\nabla \tilde{I}_{np^2+1}\|$ of the patch weights.

3.1 Minimizing Flow

The PDE flow that minimizes (4) is found by the derivation of $E_1(\tilde{\mathbf{I}})$ using the Euler-Lagrange equations and by the expression of the corresponding gradient descent algorithm. It leads to the well known *heat flow* equation, which is in our case performed on the high-dimensional patch space Γ :

$$\begin{cases} \tilde{\mathbf{I}}_{[t=0]} = \tilde{\mathbf{I}}^{noisy} \\ \frac{\partial \tilde{I}_i}{\partial t} = \Delta \tilde{I}_i \end{cases} \quad (5)$$

where Δ is the Laplacian operator on Γ . Similarly to denoising techniques based on classical diffusion PDE's [3, 5, 6, 23, 24, 27], we are not particularly interested by the steady-state solution of (5), since it would roughly consist of a constant solution. We are rather looking for a solution of this multi-dimensional heat flow at a particular *finite time* t_1 . It has been already proven [18] that this solution is the convolution of the initial estimate $\tilde{\mathbf{I}}^{noisy}$ with a normalized Gaussian kernel G_{σ} with a standard deviation $\sigma = \sqrt{2} t_1$. Here, this convolution has to be done on the high-dimensional patch space Γ : $\tilde{\mathbf{I}} = \tilde{\mathbf{I}}^{noisy} * G_{\sigma}$, with

$$\forall \mathbf{p} \in \Gamma, \quad G_{\sigma}(\mathbf{p}) = \frac{1}{(2\pi\sigma^2)^{\frac{np^2+2}{2}}} e^{-\frac{\|\mathbf{p}\|^2}{2\sigma^2}} \quad (6)$$

This is a quite straightforward way of defining the patch-based counterpart of the *Tikhonov* regularization process.

3.2 Interpretation in the Image Domain

Interesting things arise when one tries to interpret the result of this patch-based heat flow (5) in the original image domain Ω . In fact, the patch transform (1) tells us that $\tilde{\mathbf{I}}^{noisy}$ vanishes almost everywhere in Γ , excepted on the points $\mathbf{p} = (x, y, \mathcal{P}_{(x,y)}^{noisy})^T$ corresponding to the locations of the original image patches in \mathbf{I}^{noisy} . So, the convolution (6) can be simplified as: $\forall (x, y, \mathcal{P}) \in \Gamma$,

$$\tilde{\mathbf{I}}_{(x,y,\mathcal{P})} = \int_{\Omega} \tilde{\mathbf{I}}_{(p,q,\mathcal{P}_{(p,q)}^{noisy})}^{noisy} G_{\sigma(p-x,q-y,\mathcal{P}_{(p,q)}^{noisy}-\mathcal{P})} dp dq$$

We also notice that the locations of the most significant patches in $\tilde{\mathbf{I}}$, as defined in (2), will be the same as the ones in $\tilde{\mathbf{I}}^{noisy}$, since the convolution (6) by a Gaussian kernel leaves the maxima of the patch weights \tilde{I}_{np^2+1} unchanged [18]. So, the inverse patch transform (3) of $\tilde{\mathbf{I}}$ has a simple closed-form expression in Ω , and can be written directly from \mathbf{I}^{noisy} , considering that we have

$\forall (x, y), \quad \mathcal{P}_{\max(x,y)}^{\tilde{\mathbf{I}}} = \mathcal{P}_{\max(x,y)}^{\tilde{\mathbf{I}}^{noisy}} = \mathcal{P}_{(x,y)}^{\mathbf{I}^{noisy}}$ as well as $\tilde{I}_{ip^2+p^2+1}^{noisy}(x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}^{noisy}}) = I_{i(x,y)}^{noisy}$, and $\tilde{I}_{np^2+1(x,y,\mathcal{P}_{(x,y)}^{\mathbf{I}^{noisy}})}^{noisy} = 1$. Using these relations together with (3), and after few calculus, we end up with:

$$\forall (x, y) \in \Omega, \quad \mathbf{I}_{(x,y)} = \frac{\int_{\Omega} \mathbf{I}_{(p,q)}^{noisy} w_{(x,y,p,q)} dp dq}{\int_{\Omega} w_{(x,y,p,q)} dp dq} \quad (7)$$

with $w_{(x,y,p,q)} = w_{(x,y,p,q)}^s w_{(x,y,p,q)}^p$, where $w_{(x,y,p,q)}^s = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-p)^2+(y-q)^2}{2\sigma^2}}$ and

$$w_{(x,y,p,q)}^p = \frac{1}{(2\pi\sigma^2)^{\frac{np^2}{2}}} e^{-\frac{\|\mathcal{P}_{(x,y)}^{noisy} - \mathcal{P}_{(p,q)}^{noisy}\|^2}{2\sigma^2}} \quad (8)$$

Thus, each pixel value $\mathbf{I}_{(x,y)}$ of the regularized image is the result of a weighted averaging of all noisy pixels $\mathbf{I}_{(p,q)}^{noisy}$, the weight depending both on the spatial distance between points (x, y) and (p, q) (first term in (8)), as well as the similarity between corresponding patches centered on (x, y) and (p, q) (second term in (8)). Of course, this interpretation as a filtering process in Ω greatly simplifies the implementation step since it naturally suggests an algorithm in a lower-dimensional space Ω . But we have to keep in mind that it is actually the outcome of a classical *Tikhonov* minimizing flow on the patch space Γ . Note that the Gaussian functions in the spatial and patch-based terms of the weighing function $w_{(x,y,p,q)}$ (8) have the same standard

deviation σ . Having different σ can be easily simulated by pre-multiplying the image \mathbf{I}^{noisy} by a ratio factor λ before processing, being then equivalent than having two different standard deviations $\sigma_{patch} = \sigma_{spatial}/\lambda$.

3.3 Link with Other Filtering Methods

The patch-based *Tikhonov* regularization method (7) is actually quite similar to the *Non Local Means* technique, as defined in [10]. Differences are twofold: First, it is naturally defined as the solution of a *minimizing flow* acting on multi-valued images, while the original formulation has been expressed as an explicit nonlinear filter acting on scalar images. Second, the averaging weights $w_{i,j}$ of the *Non Local Means* algorithm are only based on the similarity between patches. Here, the weighting function (8) also considers the spatial distances between patches to average. Also, in the extreme case when the patch size p is reduced to 1 (so, a patch is just one point), the regularization method (7) becomes the natural multi-valued version of another well know nonlinear image filter, namely the *Bilateral Filtering* method [5,6,20,26]. These filters are known to be anisotropic in the image domain Ω . The benefit of minimizing the *Tikhonov* functional on the patch space Γ rather than on the image domain Ω is obvious in terms of regularization quality: It avoids the typical isotropic smoothing behavior of the *Tikhonov* regularization that usually over-smoothes the important image structures, such as the edges, corners and textures. A comparative figure of all these regularization algorithms is shown and commented in Sect. 7.

4 Patch-Based Geometry in Γ

Suppose now we want to smooth $\tilde{\mathbf{I}}$ in an anisotropic way, instead of doing isotropic smoothing as the previous high-dimensional heat flow does. This indeed requires a preliminary analysis of its local geometry in Γ , by finding the directions and the contrast of the discontinuities in $\tilde{\mathbf{I}}$. For classical multi-valued images, this step is usually done by retrieving the spectral elements of the (possibly smoothed) structure tensor field \mathbf{J}_σ , as suggested in [29, 30]. But, as the expression of \mathbf{J}_σ is dimension-independent, we can easily consider its high-dimensional extension $\tilde{\mathbf{J}}_\sigma = \sum_{i=1}^n \nabla \tilde{I}_{i\sigma} \nabla \tilde{I}_{i\sigma}^T$ for our patch transform dataset $\tilde{\mathbf{I}}$, where $\tilde{I}_{i\sigma} = \tilde{I}_i * G_\sigma$ is a filtered version of \tilde{I}_i by a normalized $(np^2 + 2)$ -dimensional Gaussian kernel G_σ .

4.1 Estimation of $\nabla \tilde{\mathbf{I}}_\sigma$

Practically, the high dimensionality $(np^2 + 2)$ of Γ is a major obstacle for the estimation of $\tilde{\mathbf{J}}_\sigma$ since it is not possible to store all values of $\tilde{\mathbf{I}}$ in computer memory. Fortunately, these gradients can be estimated directly on the original image

domain Ω thanks to the derivation property of the convolution $\partial(\tilde{\mathbf{I}} * \mathcal{K}) = \tilde{\mathbf{I}} * \partial \mathcal{K}$, and to the fact that $\tilde{\mathbf{I}}$ vanishes almost everywhere except at $(x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}})$. So, we get: $\forall \mathbf{p} \in \Gamma$, $\nabla \tilde{I}_{i\sigma}(\mathbf{p}) = \int_{(k,l) \in \Omega} I_{i(k,l)} \nabla G_{\sigma(\mathbf{p}-\mathbf{q}_{(k,l)})} d_k d_l$, where $\mathbf{q}_{(k,l)} = (k, l, \mathcal{P}_{(k,l)}^{\mathbf{I}})$. Once the smoothed gradient $\nabla \tilde{I}_{i\sigma}$ is estimated, the computation of the symmetric $(np^2 + 2) \times (np^2 + 2)$ structure tensor $\tilde{\mathbf{J}}_{\sigma}$ becomes a straightforward operation.

4.2 Geometry Analysis

As demonstrated in [30], the spectral elements of the structure tensor $\tilde{\mathbf{J}}_{\sigma}$ are closely related to the local variations of the studied multi-valued function. In our case, $\mathcal{N}_{(\mathbf{p})} = \sqrt{\text{trace}(\tilde{\mathbf{J}}_{(\mathbf{p})})}$ gives precious informations about the local variation amplitudes considering at the same time both the spatial and patch-valued distributions of the neighboring points (i.e. the patch discontinuities). A high value for $\mathcal{N}_{(\mathbf{p})}$ means that the pixel is located on an edge of \mathbf{I} , but also that the patch to which it belongs is probably dissimilar to the other patches within its neighborhood.

The principal eigenvector $\tilde{\mathbf{u}}$ of $\tilde{\mathbf{J}}_{\sigma}$ is another valuable feature: it gives the orientation normal to the isophotes of $\tilde{\mathbf{I}}$, and here, is closely related to the normal vector to the manifold formed by the located patches of \mathbf{I} in Γ . Clearly, smoothing $\tilde{\mathbf{I}}$ along $\tilde{\mathbf{u}}$ should be avoided as much as possible in order to preserve the significant discontinuities and patch structures of \mathbf{I} . This is the usual edge-preserving heuristic which consists in favoring smoothing along the edge directions instead of across them. Here, the “edges” are simply considered as patch-based features. The variance σ used for the gaussian smoothing of $\tilde{\mathbf{J}}_{\sigma}$ tells about the regularity of the estimated spatial-patch geometry of $\tilde{\mathbf{I}}$. It ideally depends on the level of noise in \mathbf{I} . Note that the importance of the patch component versus the spatial part is directly proportional to the value range of the image values: In this setting, multiplying the values of \mathbf{I} with a factor $\lambda > 0$ allows to give more importance to the spatial part ($\lambda \rightarrow 0$) or to the patch-based part ($\lambda \gg 0$) of the analyzed image geometry.

5 Patch-Based Diffusion PDE's

Now, we have a continuous patch-based representation $\tilde{\mathbf{I}}$ of \mathbf{I} as well as local geometric indicators $\mathcal{N}_{(\mathbf{p})}$ and $\tilde{\mathbf{u}}_{(\mathbf{p})}$. We propose then to apply the following generic diffusion PDE on Γ :

$$\forall \mathbf{p} \in \Gamma, \quad \frac{\partial \tilde{\mathbf{I}}_{(\mathbf{p})}}{\partial t} = \text{trace}(\tilde{\mathbf{D}}_{(\mathbf{p})} \tilde{\mathbf{H}}_{i(\mathbf{p})}) \quad (9)$$

This is a high-dimensional version of the equation proposed in [27] in the context of classical 2D image regularization. $\tilde{\mathbf{H}}_{i(\mathbf{p})}$ is the Hessian matrix of \tilde{I}_i at \mathbf{p} . $\tilde{\mathbf{D}} : \Gamma \rightarrow \mathbf{P}_{(np^2+2)}$ is a field of diffusion tensors which direct the regularization process by

defining the orientations (tensors eigenvectors) and amplitudes (tensors eigenvalues) of the pixel diffusion at each point \mathbf{p} in Γ . The way $\tilde{\mathbf{D}}$ is chosen is discussed later, but it must obviously depend on $\mathcal{N}_{(\mathbf{p})}$ and $\tilde{\mathbf{u}}_{(\mathbf{p})}$.

5.1 Approximation By Local Convolutions

Tschumperlé and Deriche [27] shows that for a constant tensor field, the PDE (9) is a tensor-deformed heat flow whose solution $\tilde{\mathbf{I}}^{[t]}$ at time t is found to be the convolution of the initial data by a tensor-oriented Gaussian kernel. This result also holds in our patch space: $\tilde{\mathbf{I}}^{[t]} = \tilde{\mathbf{I}}^{[t=0]} * G_t^{\tilde{\mathbf{D}}}$, where $\forall \mathbf{p} \in \Gamma$, $G_t^{\tilde{\mathbf{D}}}(\mathbf{p}) = \frac{1}{(4\pi t)^{\frac{np^2+2}{2}}} e^{-\frac{\mathbf{p}^T \tilde{\mathbf{D}}^{-1} \mathbf{p}}{4t}}$.

When $\tilde{\mathbf{D}}$ is *not* constant (usual case), (9) becomes *nonlinear* and getting one explicit solution is no longer possible. To avoid the traditional resolution of (9) using a finite difference scheme, [27] proposed an alternative iterative approximation based on local convolutions by differently oriented Gaussian kernels. In our case, this approximation scheme can be transposed in Γ as:

$$\forall \mathbf{p} \in \Gamma, \quad \tilde{\mathbf{I}}_{(\mathbf{p})}^{[dt]} \approx \int_{\mathbf{q} \in \Gamma} \tilde{\mathbf{I}}_{(\mathbf{q})}^{[t=0]} G_{dt(\mathbf{p}-\mathbf{q})}^{\tilde{\mathbf{D}}(\mathbf{p})} d\mathbf{q} \quad (10)$$

where dt is the discretized time step of the PDE evolution. Note that as (10) is a local averaging process, the approximation remains stable even for large dt . The approximation is also better when $\tilde{\mathbf{D}}$ does not depend on the time t (i.e. is estimated from $\tilde{\mathbf{I}}^{[t=0]}$) and is not locally varying too much (it has smoothness properties). It is finally worth noting that (10) will not modify the locations of the most significant patches in $\tilde{\mathbf{I}}$. As a consequence, the back-projection procedure (3) is simplified since $\mathcal{P}_{sig(x,y)}^{\tilde{\mathbf{I}}}$ and $\mathcal{P}_{(x,y)}^{\mathbf{I}}$ stay in place. All this means that the solution of (9) can be actually estimated using a single (large) iteration, only at the original patches locations $(x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}})$, and as $\tilde{\mathbf{I}}^{[t=0]}$ is sparse, this can be directly expressed in the original image domain Ω : $\forall (x, y) \in \Omega$,

$$\tilde{\mathbf{I}}_{(\mathbf{p}_{(x,y)})}^{[t]} \approx \int_{(k,l) \in \Omega} \tilde{\mathbf{I}}_{(k,l)}^{[t=0]} G_{dt(\mathbf{p}_{(x,y)} - \mathbf{q}_{(k,l)})}^{\tilde{\mathbf{D}}(\mathbf{p}_{(x,y)})} d_k d_l \quad (11)$$

where $\mathbf{p}_{(x,y)} = (x, y, \mathcal{P}_{(x,y)}^{\mathbf{I}})$ and $\mathbf{q}_{(k,l)} = (k, l, \mathcal{P}_{(k,l)}^{\mathbf{I}})$.

5.2 Choice of the Diffusion Tensors

Note that if $\tilde{\mathbf{D}}$ is chosen to be the identity matrix, we get back to the high-dimensional heat flow, and the isotropic case from Sect. 3. But, isotropic diffusion tensors do not take care of the local geometry of the patch-projected dataset $\tilde{\mathbf{I}}$. The

diffusion is done in all patch-space orientations, including $\tilde{\mathbf{u}}_{(\mathbf{p})}$, the principal eigenvector of $\tilde{\mathbf{J}}_{\sigma(\mathbf{p})}$. In order to smooth the image while preserving the local spatial and patch structures of $\tilde{\mathbf{I}}$, we rather propose to consider the following field of anisotropic diffusion tensors:

$$\forall \mathbf{p} \in \Gamma, \quad \tilde{\mathbf{D}}_{(\mathbf{p})} = \frac{1}{\mathcal{N}_{(\mathbf{p})}} \left(\mathbf{I}_d - \tilde{\mathbf{u}}_{(\mathbf{p})} \tilde{\mathbf{u}}_{(\mathbf{p})}^T \right) \quad (12)$$

Our choice is inspired by the classical *Total Variation* framework for scalar images [22], corresponding to the anisotropic diffusion $\frac{\partial I}{\partial t} = \text{div}(\nabla I / \|\nabla I\|)$ which can be also expressed as $\frac{\partial I}{\partial t} = \text{trace}(\mathbf{D}\mathbf{H})$ with $\mathbf{D} = \frac{1}{\|\nabla I\|} (\mathbf{I}_d - \eta \eta^T)$ and $\eta = \nabla I / \|\nabla I\|$. Intuitively, this choice of diffusion tensors (12) reduce the amplitude of the patch-based diffusion process on locations having high patch-gradients and smoothes the patch structures always along the hyperplane locally tangent to the discontinuities in $\tilde{\mathbf{I}}$. Consequently, the anisotropic diffusion (9) is a *non-local patch-preserving smoothing procedure*. Note that for 1×1 patches ($q = 0$), our method also defines an anisotropic variant of the *Bilateral Filtering* method. Note that actually, any kind of anisotropic diffusion tensor field \mathbf{D} can be considered in our patch-based smoothing framework (9). The best choice is ideally application-dependent and \mathbf{D} may be designed from any other image features or external datasets if necessary.

6 Image Registration by Patch-Based Variational Method

Let us now consider the problem of estimating a displacement field $\mathbf{u} : \Omega \rightarrow \mathfrak{R}^2$ between two multi-valued images $\mathbf{I}^1 : \Omega \rightarrow \mathfrak{R}^n$ (the *reference* image) and \mathbf{I}^2 (the *target* image). This image alignment problem can be typically solved by a variational method, which aims at finding the vector field \mathbf{u} that minimizes the following energy:

$$E_2(\mathbf{u}) = \int_{\Omega} \alpha \|\nabla \mathbf{u}_{(\mathbf{p})}\|^2 + \|\mathcal{D}_{(\mathbf{p}, \mathbf{p}+\mathbf{u})}\|^2 d\mathbf{p} \quad (13)$$

The user-defined parameter $\alpha \in \mathfrak{R}^+$ imposes a regularity constraint on the estimated vector field \mathbf{u} , if it chosen to be non zero. $\mathcal{D}_{(\mathbf{p}, \mathbf{q})}$ is a measure of the *dissimilarity* between image pixels $\mathbf{I}_{(\mathbf{p})}^{t_1}$ and $\mathbf{I}_{(\mathbf{q})}^{t_2}$. Lot of different expressions for \mathcal{D} have been already proposed in the literature [3, 4, 19]. One of the most common choice makes the assumption that \mathbf{I}^1 and \mathbf{I}^2 are acquired under the same global illumination conditions (for instance, they are successive frames of a video sequence) and then, that the constraint of the *Brightness Consistency* holds.

In this case, \mathcal{D} can be reasonably chosen to be:

$$\mathcal{D}_{1(\mathbf{p}, \mathbf{q})} = \mathbf{I}_{\sigma(\mathbf{p})}^{t_1} - \mathbf{I}_{\sigma(\mathbf{q})}^{t_2} \quad (14)$$

where $\mathbf{I}_{\sigma}^{t_k} = \mathbf{I}^k * G_{\sigma}$ are filtered versions of the images \mathbf{I}^k , convolved by a normalized Gaussian kernel G_{σ} . It allows the consideration of regularized reference and

target images instead of possibly noisy ones. The choice $\mathcal{D} = \mathcal{D}_1$ in (13) leads to a variational extension of the *Lucas–Kanade* method [4, 19] for multi-valued images.

6.1 Dissimilarity Measure on the Patch Space

The dissimilarity measure \mathcal{D}_1 can be naturally extended to deal with patches, by expressing it with the patch transforms $\tilde{\mathbf{I}}^1$ and $\tilde{\mathbf{I}}^2$ instead of the original images \mathbf{I}^1 and \mathbf{I}^2 . This is quite straightforward: We just replace the pointwise intensities by the most significant patches (2) present in the regularized versions of the patch transforms $\tilde{\mathbf{I}}_\sigma^k = \tilde{\mathbf{I}}^k * G_\sigma$:

$$\mathcal{D}_2(\mathbf{p}, \mathbf{q}) = \tilde{\mathbf{I}}_{\sigma(\mathbf{p}, \mathcal{P}_{\max(\mathbf{p})}^1)}^{t_1} - \tilde{\mathbf{I}}_{\sigma(\mathbf{q}, \mathcal{P}_{\max(\mathbf{q})}^2)}^{t_2} \quad (15)$$

Here, the smoothed patch transforms $\tilde{\mathbf{I}}_\sigma^k$ represent edge-preserving filtered versions of the reference and target images \mathbf{I}^k , as mentioned in Sect. 3. It means also that the localization of the most significant patches is known to be $\mathcal{P}_{\sigma \max}^k = \mathcal{P}^k$.

In the particular case where $\sigma = 0$, the dissimilarity function \mathcal{D}_2 is the same as the one used for the classical *Block Matching* method. Moreover, when $\alpha = 0$ (no regularization constraints are considered on \mathbf{u}), the *Block Matching* is a global minimizer of (13).

6.2 Minimizing Flow

In a more general setting, the Euler–Lagrange derivation of (13) gives the set of coupled PDE’s which locally minimizes the energy functional E_2 :

$$\forall j \in [1, 2], \forall \mathbf{x} \in \Omega,$$

$$\begin{cases} \mathbf{u}_{[t=0]} = \mathbf{0} \\ \frac{\partial u_j(\mathbf{x})}{\partial t} = \alpha \Delta u_j + \\ \sum_{i=1}^{np^2+1} \left(\tilde{f}_{\sigma i(\mathbf{x}, \mathcal{P}_{(\mathbf{x})}^1)}^{t_1} - \tilde{f}_{\sigma i(\mathbf{x}+\mathbf{u}, \mathcal{P}_{(\mathbf{x}+\mathbf{u})}^2)}^{t_2} \right) [\nabla \mathcal{G}_i]_{j(\mathbf{x}+\mathbf{u})} \end{cases} \quad (16)$$

where $\mathcal{G}_i(\mathbf{x}) = \tilde{f}_{\sigma i(\mathbf{x}, \mathcal{P}_{(\mathbf{x})}^2)}^{t_2}$. Here again, this minimizing flow (16) can be implemented directly in the image domain Ω , without having to explicitly compute and store the patch transforms $\tilde{\mathbf{I}}_\sigma^k$. Indeed, it requires only the retrieval of the most significant patches in $\tilde{\mathbf{I}}_\sigma^k$, which can be proven to be $\mathcal{P}_{\sigma \text{sig}}^k = \mathcal{P}_{\text{regul}}^k$, where $\mathbf{I}_{\text{regul}}^k$ is the

nonlinear filtered version of \mathbf{I}^k by the *Tikhonov* regularization flow on the patch space.

This patch-based registration PDE (16) is a local minimizer of a *Block Matching*-like objective function. Differences are twofold: First, it *implicitly* considers edge-preserving filtered versions of the reference and target images, instead of isotropically smoothed ones. But most of all, it is able to put an important a priori smoothness constraint on the estimated field \mathbf{u} . This method combines then the significance of the patch description for the matching of image structures, while keeping the aptitude of the variational methods to impose useful constraints. These interesting properties are discussed and illustrated with a comparative figure in Sect. 7.

7 Application Results

We applied the different variational techniques on the patch space presented in this paper, on different color images considered in their original (R, G, B) color space (Fig. 1).

7.1 Color Image Denoising

The anisotropic regularization flow (16) on the patch space Γ can be used to enhance degraded color images (or other multi-valued datasets). Figure 2 compares it with the most connected algorithms, namely the *Non Local Means*, the *Bilateral Filtering* and the classical *Tikhonov* regularization performed on the image domain Ω . Synthetic white Gaussian noise ($\sigma_{noise} = 20$) has been added to the original color image *Barbara*. For the honesty of the comparison, we have not applied the scalar versions of these related filters as defined in the papers [10, 26], but their multi-valued extensions. It gives indeed better denoising results than applying them channel by channel. The PSNR between the noise-free and restored images, as well as the parameters used for the experiments are displayed. For each method, the parameters have been manually tuned to optimize the obtained PSNR. As our proposed flow (16) is actually very close in its final expression to the *Non Local Means* and *Bilateral* filters, the denoising results are of course very comparable in terms of quality. It seems to perform a little bit better on certain regions (some edges are sharper in the zoomed part). We believe it is due to the fact that our weighting function (8) considers both the similarity as well as the spatial distances between patches. The visual improvement is very subtle anyway, considering the huge raise of the processing time (around x20). Indeed, computing the N-D structure tensor alone already takes around 30 more time than the computation of a single iteration of the classical NL-means algorithm.

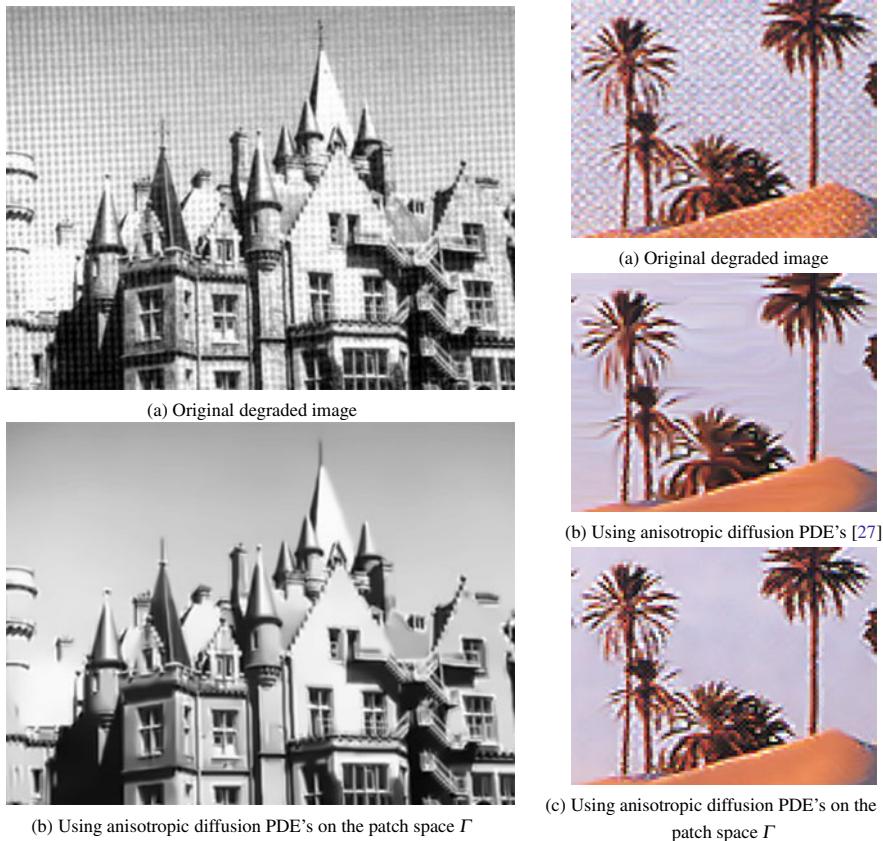


Fig. 1 Results of the proposed regularization flow on the patch space (7)

7.2 Color Image Registration

The patch-based registration PDE's (16) allow to estimate a smooth displacement field between two color images while considering a patch-based dissimilarity measure. Such a displacement field can be used then to generate a morphing video sequence, where new synthetic frames are added between the reference and target images (i.e. a *re-timing* process, where the reference frame corresponds to $t = 0$ and the target to $t = 1$). Figure 3 shows a comparison of the re-timed results obtained by our proposed registration method (16) and by the most related ones, i.e. the *Block Matching* algorithm and the pointwise version (14) of the variational registration



(a) Noisy image (std=20) (b) *Tikhonov* regularization on Ω (c) *Bilateral Filtering*



(d) *Non Local Means* (e) *Tikhonov* regularization on the patch space Γ

Result	PSNR	Parameters	Reference
(b)	25.33 dB	$\sigma = 0.6$	[7, 25]
(c)	26.97 dB	$\sigma_s = 10, \sigma_r = 25$	[5, 20, 26]
(d)	26.98 dB	$\sigma = 21$, patch 7×7	[10]
(e)	27.03 dB	$\sigma = 21$, patch 7×7	.

(f) PSNR(Original/Denoised) and parameters used for experiments

Fig. 2 Illustration of the *Tikhonov* regularization on the patch space (7), and comparisons with most related filtering techniques



Fig. 3 Illustration of the variational image registration method on the patch space (16), and comparisons with *Block Matching* and pointwise variational registration (13) methods

technique (13). For each method, the estimated displacement field \mathbf{u} is displayed as a colored image (first column). The mid-interpolated frame (i.e. reconstructed at $t = 0.5$) and a zoomed part of it are shown on the two last columns. This is indeed the hardest frame to reconstruct since it classically exhibits the largest ghosting artefacts due to motion estimation errors. To be able to handle large displacements, we considered a classical multiscale evolution scheme for the PDE's that minimize the pointwise and patch-based versions of the energy functional (13). Without any surprise, the original *Block Matching* algorithm fails in reconstructing a smooth motion field, resulting in a unpleasant mid-interpolated frame reconstruction. The pointwise version of the variational registration technique (13) is quite competitive when the smoothness constraint is high enough ($\alpha = 0.1$), but fails otherwise. Surprisingly, our proposed patch-based variational formulation (13) with the dissimilarity measure (15) outputs a quite good reconstructed frame, even when $\alpha = 0$. Considering a patch-based dissimilarity measure seems to have a natural tendency to intrinsically regularize the registration problem. Anyway, best visual results are obtained with our variational patch-based registration method while considering regularization constraints on the displacement field \mathbf{u} (last row).

8 Conclusions and Perspectives

In this paper, a reversible patch-based transform of multi-valued images has been presented. We illustrated the fact that considering variational methods in the corresponding patch space is a theoretical nice way to design natural patch-based counterparts of classical image processing techniques. The obtained experimental results seems to confirm this impression, even if the computation costs are still heavy to perform. This effort is a first step in a way to mix both patch and variational worlds and that our proposed framework can be used to translate many other “pointwise” variational formulations into their natural patch-based counterparts.

References

1. Ashikhmin, M.: Synthesizing natural textures. In: Symposium on Interactive 3D Graphics, pp. 217–226. (2001)
2. Ashikhmin, M.: Fast texture transfer. *IEEE Comput. Graph. Appl.* **23**(4), 38–43 (2003)
3. Aubert, G., Kornprobst, P.: Mathematical problems in image processing: PDE's and the calculus of variations. *Applied Mathematical Sciences*, vol. 147. Springer, Berlin (2006)
4. Bake, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **56**(3), 221–255 (2004)
5. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE PAMI* **24**(6), 844–847 (2002)
6. Barash, D., Comaniciu, D.: A common framework for Nonlinear Diffusion, Adaptive Smoothing, Bilateral Filtering and Mean Shift. *Image Vis. Comput.* **22**(1) (2004)
7. Bouhamidi, A., Jbilou, K.: Sylvester Tikhonov-regularization methods in image restoration. *J. Comput. Appl. Math.* **206**(1), 86–98 (2007)

8. Boulanger, J., Kervrann, C., Bouthemy, P.: Space-time adaptation for patch-based image sequence restoration. *IEEE PAMI*. **29**(6), 1096–1102 (2007)
9. Brox, T., Cremers, D.: Iterated nonlocal means for texture restoration. In: *SSVM*. vol. 4485 of *Lecture Notes in Computer Science*, pp. 13–24. Springer, Berlin (2007)
10. Buades, A., Col, B., Morel, J.M.: A non-local algorithm for image denoising. *IEEE CVPR* 60–65 (2005)
11. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **13**(9)
12. Efros, A.A., Leung, T.K.: Texture Synthesis by Non-Parametric Sampling. *IEEE ICCV* **2**, 1033–1038 (1999)
13. Fadili, M., Starck, J.L., Boubchir, L.: Morphological diversity and sparse image denoising. *IEEE ICASSP* **1**, 589–592 (2007)
14. Hays, J., Efros, A.A.: Scene completion using millions of photographs. *ACM Trans. Graph.* **26**(3) (2007)
15. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.: Image Analogies. In: *SIGGRAPH, Computer Graphics Proceedings*, pp. 327–340. (2001)
16. Kervrann, C., Boulanger, J.: Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. Image Process.* **15**(10), 2866–2878 (2006)
17. Kindermann, S., Osher, S., Jones, P.: Deblurring and denoising of images by nonlocal functionals. *SIAM Multiscale Model. Simul.* **4**, 1091–1115 (2005)
18. Koenderink, J.: The structures of images. *Biol. Cybern.* **50**, 363–370 (1984)
19. Lucas, B.D., Kanade, T.: In: Hayes, P.J. (ed.) *An Iterative Image Registration Technique with an Application to Stereo Vision*. *IJCAI*, pp. 674–679. William Kaufmann (1981)
20. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vis. (IJCV)* **81**(1), 24–52 (2009)
21. Peyré, G.: Manifold models for signals and images. *J. Comput. Vis. Image Underst.* 249–260 (2009)
22. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**, 259–268 (1992)
23. Sapiro, G.: *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge (2001)
24. Sochen, N.A., Zeevi, Y.Y.: Representation of Colored Images by Manifolds Embedded in Higher Dimensional Non-Euclidean Space. In: *ICIP*, vol. 1, pp. 166–170. (1998)
25. Tikhonov, A.: Regularization of incorrectly posed problems. *Sov. Math.* **4**, 1624–1627 (1963)
26. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *International Conference on Computer Vision (ICCV’98)*, pp. 839–846 (1998)
27. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with pde’s: A common framework for different applications. *IEEE PAMI* **27**(4), pp. 506–517 (2005)
28. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: *SIGGRAPH, Computer Graphics Proceedings*, pp. 479–488. (2000)
29. Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart (1998)
30. Zenzo, S.D.: A note on the gradient of a multi-image. *Comput. Vis. Graph. Image Process.* **33**(1), 116–125 (1986)