Biologically plausible regularization mechanisms

Thierry Viéville

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Biologically plausible regularization mechanisms

T. Viéville

N° 4625

Novembre 2002

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4625--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Biologically plausible regularization mechanisms

T. Viéville

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

Rapport de recherche n° 4625 — Novembre 2002 — 42 pages

Abstract: This study aims at proposing an implementation of regularization mechanisms compatible with biological operators. More precisely, cortical maps code vectorial parametric quantities, computed by network of neurons. In computer vision, similar quantities are efficiently computed using implementations of partial differential equations which define regularization processes allowing to obtain well-defined estimations of these quantities.

One of these methods is based on an integral approximation of the diffusion operator used in regularization mechanisms. Following this formulation, the present development defines a somehow optimal implementation of such an integral operator with two interesting properties:

(i) when used on sampled data such as image pixels or 3D data voxels, it provides an unbiased discrete implementation of such an operator;

(ii) when used as a model of biological plausible mechanisms, it corresponds to a simple local feedback defined over a small bounded region of any shape inside the parametric space.

As such it may be linked to what is processed in a cortical column of the brain and provides an interesting model of general operators corresponding to such a neuronal structure.

The present development is illustrated by some experiments of visual motion estimation.

Key-words: Regularization methods, Diffusion operator, Biological model

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

Mécanismes de régularisation biologiquement plausibles

Résumé : Cette étude propose une implémentation de mécanismes de régularisation compatibles avec des opérateurs biologiques. Plus précisément, les cartes corticales du cerveau encodent des quantités vectorielles calculées par le réseau des neurones. En vision par ordinateur, des quantités similaires sont calculées efficacement en implémentant des équations aux dérivées partielles qui définissent des processus de régularisation permettant d'obtenir des estimations bien définies de ces quantités.

L'une de ces méthodes, introduite par Raviat puis développée par Degond et Mas-Gallic, est basée sur une approximation intégrale de l'opérateur de diffusion utilisé par le mécanisme de régularisation. En suivant cette formulation, le présent développement défini une implémentation en quelque sorte optimale d'un tel opérateur intégral ayant deux propriétés intéressantes:

(i) dans le cas de données échantillonnées (pixels d'une image ou voxels d'une image volumique), ce procédé permet d'obtenir une implémentation discrète non biaisée de l'opérateur,

(ii) dans le cas de la modélisation de mécanismes biologiquement plausibles, cela correspond simplement à un calcul itératif local défini dans un petit voisinage de forme quelconque à l'intérieur de l'espace des paramètres,

ce qui en tant que tel, peut-être relié à ce qui est calculé au sein d'une colonne corticale du cerveau, fournissant là un modèle intéressant d'opérateurs assez généraux et correspondant à une structure neuronale.

Le présent mécanisme est illustré par quelques expériences d'estimation visuelle du mouvement.

Mots-clés : Méthode de régularisation, Opérateur de diffusion, Modèles biologiques

1 Introduction

Cortical maps

Perceptual processes, in computer or biological vision, require the computation of "maps" of quantitative values. The retinal image itself is a "retinotopic map": for each cell of the retina or each pixel of the image there is have a value corresponding to the image intensity at this location. This value is a vectorial value for color images. A step further, in early-vision, the retinal image contrast is computed at each location, allowing to detect image edges related to boundaries between image areas. There are edge detectors in both artificial visual systems (see e.g. [10] for a general introduction) and in the brain neuronal structures involved in vision perception (e.g. [14] for a simple overview). In both cases, such maps encode not only the contrast magnitude, but several other cues: contrast orientation related to edge orientation, shape curvature, binocular disparity related to the visual depth, color cues, temporal disparity between two consecutive images in relation with visual motion detection, etc.. Such maps are not only parameterized by retinotopic locations, but also using 3D locations, or parameterized by other parameters such as orientation, retinal velocity, etc..

The Partial Differential Equation (PDE) approach

In computer vision, a relevant and efficient theory is now available regarding the definition and computation of such maps of quantitative values (see e.g. [1] for a didactic introduction about the "axiomatization" of this part of computer vision). This formalism not only provides a clear basic of "what is to be done" (i.e. requirements in order to have coherent and consistent definitions) but also of "how to do it" since the theory is effective in the sense that efficient implementations may be derived (see e.g. [2] for a recent treatise on this subject). The "what is to be done" level is formalized in terms of a criterion to minimize and the "how to do it" level is related to the, so called, Euler-Lagrange equations which allow to improve an initial guess of the solution and get closer to the optimal solution. At a technical level, as revisited in this paper, these quantities are efficiently computed using implementations of partial differential equations which define regularization processes allowing to

obtain well-defined estimations of these quantities. This powerful methodology is also very general in the sense that a large variety of computational problems are solvable within this framework (see e.g. [7, 6] for a review).

Using Neuronal Network (NN) models

In biological vision modeling, the situation is dominated by the idea that "cortical maps code vectorial parametric quantities are computed by network of neurons", mainly neuronal networks à-la Hopfield (e.g. [11] for a recent review). Such models allow to analyze in details the biological substrate of such computations (e.g. [8]), in other words to raise efficient hypotheses about "how it is done". Alternatives to these models also exist (e.g. [30] where a spike computational model is considered). Such models are either based on some "ad-hoc" mechanisms, more precisely specific descriptions of neuronal connections which are "plausible" regarding one or another operation (e.g. [14] for edge and disparity calculations) or more abstract mechanisms (e.g. [21, 28] where a computational description of early-vision and motion computations in terms of adaptive linear filtering is proposed). Although these approaches already allow a deep understanding of what is processed in important areas of the brain, there is no clear link with the computer vision algorithms mentioned before.

Building a link between PDE and NN

Such a link, would indeed have several advantages, providing a methodology to relate the "what is to be done" and the "how to do it" levels and also providing a very general computational framework for such calculations. A step further, PDE calculations regularize their input, i.e. provide a well-defined and stable estimation even in the case of noisy or partially defined data. Thanks to this property, algorithms have performances closer to those of biological systems than artificial neuronal networks. Furthermore, the link we want to build would provide a "common framework" between computational and biological vision in order to develop common models and biological plausible algorithms. However, building such a link is not immediate because there is yet no clear understanding of how such complex differential non-linear methods could cor-

respond to neuronal mechanisms although the question has been already raised [18, 24].

Implementing PDE as NN

In order to build this link, we must have a method which allows to implement the computer vision partial differential equations, using networks of neuronal units. As it will made explicit in the sequel, the difficult point is the anisotropic diffusion operators implementation. Concretely, such units must integrate the information in a bounded neighborhood so that the cooperation between these units allow a global computation of the quantitative map. Furthermore, to be biologically plausible, such mechanisms must be based on simple local linear feedback [8, 11]. In other words, we must provide a "particle implementation" of such numerical computations. Such a method, used in fluid dynamics computation has been introduced by Leonard [20] followed by Raviat and Mas-Gallic [25] and developed by Degond and Mas-Gallic [5]. It is based on an integral approximation of the diffusion operator used in the regularization mechanism.

The goal of this paper is to describe how computer vision partial differential equations can be implemented using networks of neuronal units.

What is the paper about

Following this track, in the next section, we are going to revisit the computer vision partial differential equation methodology in the case of vectorial maps and also discuss how it may be generalized to non-linear cortical maps computations.

We then are going to re-derive an improved version of the Degond and Mas-Gallic method, considering a somehow optimal implementation of such an integral operator with two interesting properties:

(i) when used on sampled data such as image pixels or 3D data voxels, it provides an unbiased discrete implementation of such an operator;

(ii) when used as a model of biological plausible mechanisms, it corresponds to a simple local feedback defined over a small bounded region inside the parametric space.

We finally will illustrate our discussion by some experiments of visual motion estimation.

Notations We write *vectors* and *matrices* in bold letters, matrices being written with capital letters and scalars in italic. The dual of a quantity $\mathbf{x}$ is represented as its transpose $\mathbf{x}^T$, the dot-product between vectors being written as $\mathbf{x}^T\mathbf{y}$. We represent the components of a matrix or a vector using superscripts, e.g.: $\mathbf{x} = (x^0, x^1, x^2)^T$.

Similarly we write *tensors* in bold letters with covariant and contravariant indexes written as indices and exponent, respectively. For instance, the following *Kronecker* symbol $\delta_i^j =$ if $i = j$ then 1 else 0, has covariant index i and contravariant index j. In this context, a $m \times n$ matrix $M_i^j, i = 1..n, j = 1..m$ is a 1-covariant / 1 -contravariant tensor. Although, the "Einstein" notation allows implicit summations of tensor indexes, it appears clearer, here, to explicit these summations.

For vector of integer indices $\alpha = (\alpha_1 \cdots \alpha_n) \in \mathcal{N}^n$ we write:

$$|\alpha| = \alpha_1 + \cdots + \alpha_n \text{ and } \alpha! = \alpha_1! \cdots \alpha_n!$$

so that we can write concisely:

$$\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \text{ and } \partial_{\mathbf{x}}^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}$$

while the Taylor expansion of a function $h : \mathcal{R}^n \to \mathcal{R}$ becomes:

$$h(\mathbf{x}) = h(\mathbf{x}_0) + \sum_{|\alpha|=1}^{r} \frac{\partial^\alpha h}{\alpha!} (\mathbf{x} - \mathbf{x}_0)^\alpha + R^r h$$

where the remainder $R^\epsilon h$ may be written using an integral form:

$$R^r h = \sum_{|\alpha|=r+1} \frac{r+1}{\alpha!} \int_{[0,1]} (\mathbf{x} - \mathbf{x}_0)^\alpha (1-u)^r \partial^\alpha h(\mathbf{x}_0 + u(\mathbf{x} - \mathbf{x}_0)) du$$

We also consider the Sobolev function space $W^{s,\infty}(\mathcal{R}^n)$ provided with the norm:

$$||f||_{s,\infty} = \sup_{0 \geq k \geq s} \left[\text{sup ess}_{|\alpha|=k, \mathbf{x} \in \mathcal{R}^n} |\partial^\alpha f(\mathbf{x})| \right]$$

where $\text{sup ess}_{\mathbf{x}} u(\mathbf{x})$ is the smallest constant, if any, for which $u(\mathbf{x})$ is bounded except in a negligible set of measure zero.

2 Using anisotropic diffusion operators.

Let us briefly revisit the computer vision partial differential equation methodology in the case of vectorial maps and discuss how it may be generalized to non-Euclidean cortical maps computations. The goal of this section is to define the problem and point out the importance of a correct implementation of diffusion operators.

A: Implementing regularization mechanisms

Here, as required in some situations (e.g. [1, 6, 12]), we consider functions which values are not scalars but vectors.

From functional criterion to Euler-Lagrange equation. For a vectorial map:

$$\mathbf{h} : \mathcal{R}^n \rightarrow \mathcal{R}^m$$

we consider the following variational problem:

$$\mathbf{h} = arg\ min \int l\left(\mathbf{x}, \mathbf{h}(\mathbf{x}), \nabla\mathbf{h}(\mathbf{x})\right) d\mathbf{x}$$

where $\nabla^k_j \mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}^k(\mathbf{x})}{\partial x^j}$ is a $m \times n$ matrix which corresponds to the Jacobian of the function $\mathbf{h}$ an is called its *gradient*.

In words, the *specification* of this map of values corresponds to an "objective" or a "criterion" to attain, as it will be illustrated in the sequel.

A necessary condition for a function $\mathbf{h}$ to be an extremum of this criterion is that the so called "normal" or Euler-Lagrange equation is verified. This equation simply states that, at an extremum, the criterion is locally "flat" i.e. its 1st order variation vanishes.

Here, the related Euler equation is obvious to derive (e.g. [9] for an introduction) and may be written:

$$\frac{\partial l}{\partial \mathbf{h}} - \mathbf{div}\left(\frac{\partial l}{\partial \nabla \mathbf{h}}\right) = 0$$

where $\frac{\partial l}{\partial \nabla \mathbf{h}}$ is a $m \times n$ matrix while the divergence is to be understood as a vectorial expansion, i.e. for a $m \times n$ matrix $\mathbf{M}(\mathbf{x})$:

$$\mathbf{div}_k(\mathbf{M}) = \sum_{j=1}^{n} \frac{\partial \mathbf{M}^j_k(\mathbf{x})}{\partial x^j} \quad , \quad k = 1..m$$

Defining regularization as an optimization problem. Let us now consider a particular case: the *regularization* $\mathbf{h}(\mathbf{x})$ of an "input" function $\bar{\mathbf{h}}(\mathbf{x})$. It is defined, as discussed e.g. in [6], by the following optimization problem:

$$\mathbf{h} = arg\ min \frac{1}{2} \int ||\mathbf{h} - \bar{\mathbf{h}}||^2_{\mathbf{\Lambda}} + \Phi\left(||\nabla\mathbf{h}||^2_{\mathbf{L}}\right)$$

with $||\nabla \mathbf{h}||^2_{\mathbf{L}} = \sum_{ijkl} \nabla \mathbf{h}^k_i(\mathbf{x})\, \mathbf{L}^{ij}_{kl}(\mathbf{x})\, \nabla \mathbf{h}^l_j(\mathbf{x})$ and $||\mathbf{h}-\bar{\mathbf{h}}||^2_{\mathbf{\Lambda}} = \left(\mathbf{h}-\bar{\mathbf{h}}\right)^T \mathbf{\Lambda} \left(\mathbf{h}-\bar{\mathbf{h}}\right)$.

Here, on one hand :

- $\mathbf{\Lambda} : \mathcal{R}^n \rightarrow \mathcal{R}^{m\times m} \in W^{s,\infty}(\mathcal{R}^n)$ is a, so called, *measurement information metric*, which, for the previous definition to be coherent, is :
  - symmetric i.e. $\mathbf{\Lambda} = \mathbf{\Lambda}^T$ and
  - "positive" i.e. so that $\forall \mathbf{v} \in \mathcal{R}^m, \mathbf{v}^T\, \mathbf{\Lambda}\, \mathbf{v} \geq 0$.

  This metric allows to represent:
  (i) the precision of the input function: the higher this precision in a given direction, the higher the value of $\mathbf{\Lambda}$ in this direction (in a statistical framework, $\mathbf{\Lambda}$ corresponds to the inverse of a covariance matrix) but also
  (ii) partial observation, in the sense that if the input function $\bar{\mathbf{h}}(\mathbf{x})$ is not defined for some $\mathbf{x}$ we simply have to state $\mathbf{\Lambda} = 0$ at this location. Similarly, the fact that the input function $\bar{\mathbf{h}}(\mathbf{x})$ is only defined in some direction corresponds to matrix $\mathbf{\Lambda}$ positive but not definite in this direction (for instance, if $\bar{\mathbf{h}}(\mathbf{x})$ is only defined in the direction $\mathbf{u}$ at a given location we write $\mathbf{\Lambda} = k\, \mathbf{u}\, \mathbf{u}^T$ for some k),
  and more generally
  (iii) linear relations between measures and parameter estimation , say $\mathbf{M}\, \mathbf{h} = \mathbf{m}$, which is obviously equivalent to require $||\mathbf{h} - \bar{\mathbf{h}}||^2_{\mathbf{\Lambda}} = 0$ with $\mathbf{\Lambda} = \mathbf{M}^T\, \mathbf{M}$ and $\bar{\mathbf{h}} = \mathbf{M}^T\, \mathbf{m}$ (e.g. [36] for a development).

On the other hand:

- $\mathbf{L} : \mathcal{R}^n \rightarrow \mathcal{R}^{m\times n\times m\times n} \in W^{s,\infty}(\mathcal{R}^n)$ is a, so called, *diffusion tensor* $\mathbf{L}$, which is :
  - symmetric i.e. $\mathbf{L}^{ij}_{kl} = \mathbf{L}^{ji}_{lk}$ and
  - "positive" i.e. so that $\forall \mathbf{M} \in \mathcal{R}^{m\times n}, \mathbf{M}^T\, \mathbf{L}\, \mathbf{M} = \sum_{ijkl} \mathbf{M}^k_i\, \mathbf{L}^{ij}_{kl}\, \mathbf{M}^l_j \geq 0$ in order the previous definition to be coherent, while

- $\Phi : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ defines the *profile* of this regularization, it is a strictly increasing function with $\Phi(0) = 0$,

the combined influence of these two last elements being discussed in the next paragraph.

Briefly, the term $\Phi\left(||\nabla \mathbf{h}||^2_{\mathbf{L}}\right)$ is the *regularization* term [31] which allows to correctly defined ill-posed problems, as made explicit now.

In such a case, a few algebra leads to the related Euler equation which is a vectorial diffusion partial differential equation, i.e.:

$$\mathbf{\Lambda}\,\mathbf{h} - \mathbf{\Lambda}\,\bar{\mathbf{h}} - \Delta_{\mathbf{L},\Phi}\mathbf{h} = 0 \tag{1}$$

with

$$\Delta_{\mathbf{L},\Phi}\mathbf{h} = \mathbf{div}\left(\Phi'\left(||\nabla \mathbf{h}||^2_{\mathbf{L}}\right)\sum_{lj}\mathbf{L}^{ij}_{kl}\,\nabla\mathbf{h}^l_j\right)$$

the three terms being respectively the deformation (i.e. $\mathbf{\Lambda}\,\mathbf{h}$), the offset (i.e. $\mathbf{\Lambda}\,\bar{\mathbf{h}}$) and the diffusion (i.e. $\Delta_{\mathbf{L},\Phi}\mathbf{h}$).

A convection term of the form $+\mathbf{div}\,(\mathbf{A}\,\mathbf{h})$ may be also added, although in such a case, the related equation would not correspond to the Euler equation of a criterion, i.e. the solution would not be related to an optimization criterion.

Clearly, without the term of diffusion, the solution is $\mathbf{h} = \bar{\mathbf{h}}$ while the term of diffusion $\Delta_{\mathbf{L},\Phi}\mathbf{h}$ allows to "propagate" some information about $\mathbf{h}$ from one point $\mathbf{x}$ to another. The key idea is to have the information propagated:
(a) where the input function is partially or approximately defined; in such a case, since the matrix $\mathbf{\Lambda}$ vanishes or has a small contribution and $\mathbf{h}$ is entirely or mainly defined by the diffusion term,
(b) when the problem is ill-posed, i.e. if there are many and numerically unstable solutions; in such a case, we look for a solution which variations are minimized, as obtained by the regularization term.

Non-linear versus linear diffusion operators. Let us now review how Φ and $\mathbf{L}$ influence this diffusion of information.

From a straightforward derivation:

$$\begin{aligned}
\left[\Delta_{\mathbf{L},\Phi}\mathbf{h}\right]_k &= \textstyle\sum_{ijl}\frac{\partial}{\partial x^i}\left(\Phi'\left(||\nabla\mathbf{h}||^2_{\mathbf{L}}\right)\,\mathbf{L}^{ij}_{kl}\,\frac{\partial \mathbf{h}^l}{\partial x^j}\right)\\
&= \Phi'\left(||\nabla\mathbf{h}||^2_{\mathbf{L}}\right)\,\left[\textstyle\sum_{lj}\left[\sum_i\frac{\partial \mathbf{L}^{ij}_{kl}}{\partial x^i}\right]\,\frac{\partial\mathbf{h}^l}{\partial x^j} + \sum_{ijl}\mathbf{L}^{ij}_{kl}\,\frac{\partial^2\mathbf{h}^l}{\partial x^i\partial x^j}\right]\\
&+ \Phi''\left(||\nabla\mathbf{h}||^2_{\mathbf{L}}\right)\,\textstyle\sum_{ijl}\nabla\left[||\nabla\mathbf{h}||^2_{\mathbf{L}}\right]_i\;\mathbf{L}^{ij}_{kl}\,\frac{\partial\mathbf{h}^l}{\partial x^j}
\end{aligned}$$

and in the particular case where $\mathbf{L}$ is constant and without any influence, i.e. $\mathbf{L} = \delta^{ij}\,\delta_{kl}$ so that $||\nabla\mathbf{h}||^2_{\mathbf{L}} = ||\nabla\mathbf{h}||^2$ (we consider here the $\mathcal{L}^2$ norm) this

reduces to:

$$\begin{array}{rcl} [\Delta_{1,\Phi}\mathbf{h}]_k^T & = & \Phi'\left(||\nabla\mathbf{h}||^2\right)\,\sum_i \frac{\partial^2\mathbf{h}^k}{[\partial x^i]^2} \quad + \quad 2\,\Phi''\left(||\nabla\mathbf{h}||^2\right) \quad \sum_{ij}\frac{\partial^2\mathbf{h}^k}{\partial x^i\partial x^j}\,\frac{\partial\mathbf{h}^k}{\partial x^i}\,\frac{\partial\mathbf{h}^k}{\partial x^j} \\ & = & \Phi'\left(||\nabla\mathbf{h}||^2\right)\,\Delta\mathbf{h}^k \quad + \quad 2\,\frac{\Phi''\left(||\nabla\mathbf{h}||^2\right)}{||\nabla\mathbf{h}||^2} \quad \eta^{k\,T}\,\nabla^2\mathbf{h}^k\,\eta^k \end{array}$$

with $\eta^k = \frac{\nabla\mathbf{h}^{kT}}{||\nabla\mathbf{h}||}$, so that we obtain:
(i) a term related to Φ' which corresponds to an isotropic diffusion (here $\Delta\mathbf{h}^k$ is the Laplacian of the k-th component of $\mathbf{h}$) and
(ii) a term related to Φ'' which corresponds to an anisotropic diffusion process in the direction η of the gradient.

Furthermore, in this case all components of $\mathbf{h}$ are decoupled.

A natural requirement [1, 7] is to obtain:
(α) isotropic diffusion for small gradient magnitude because we want to maximize the propagation of information in uniform (thus information less) parts of the parametric space and
(β) cancel the propagation of information for high gradient magnitude in the direction of the gradient in order to preserve large (thus significant) variations of the input function.

In other words, we want to eliminate small, thus likely noisy, variations of the parametric map, but preserve large variations.

Adapting what has been proposed for instance in [2], the constraint (α) simply means that, for small gradient magnitudes, the 2nd term becomes negligible with respect to the 1st one, i.e.:

$$\lim_{||\nabla\mathbf{h}||\to 0}[2\,\Phi''(||\nabla\mathbf{h}||^2)/||\nabla\mathbf{h}||^2]/\Phi'(||\nabla\mathbf{h}||^2) = 0$$

while the constraint (β) means that, for high gradient magnitudes, the 2nd term must be the opposite of the 1st one, in order to cancel diffusion in the gradient direction. i.e.:

$$\lim_{||\nabla\mathbf{h}||\to+\infty}\Phi'(||\nabla\mathbf{h}||^2)/[2\,\Phi''(||\nabla\mathbf{h}||^2)/||\nabla\mathbf{h}||^2] = -1$$

This is the case, e.g. of $\Phi(u) = \sqrt{u}$ and several other profiles, as reviewed in [2].

As a conclusion, the use of the non-linear function Φ simply allows to introduce anisotropic diffusion with respect to the function gradient $\nabla\mathbf{h}$. Would it be possible to obtain the same property but preserving the linearity of the diffusion equation?

Thanks to what has been derived in the scalar case [22] (regarding the estimation of image motion) there is a positive answer to this question.

Let us consider that Φ is the identity function thus has no influence, i.e. $\Phi(u) = u$, so that, with a few algebra:

$$[\Delta_{\mathbf{L}}\mathbf{h}]_k = \sum_{lj} \mathbf{M}^j_{kl} \frac{\partial \mathbf{h}^l}{\partial x^j} + \sum_{ijl} \mathbf{L}^{ij}_{kl} \frac{\partial^2 \mathbf{h}^l}{\partial x^i \partial x^j} \text{ with } \mathbf{M}^j_{kl} = \sum_i \frac{\partial \mathbf{L}^{ij}_{kl}}{\partial x^i} \tag{2}$$

Following [22] let us consider that we have an initial or previous estimation $\nabla\widehat{\mathbf{h}} = \nabla\mathbf{h}$. This estimation may have several sources: the input function $\bar{\mathbf{h}}$ (i.e. $\nabla\widehat{\mathbf{h}} = \nabla\bar{\mathbf{h}}$), a priori information, the estimation obtained by previous steps of the algorithm, as detailed in the next sub-section. In other words, we *feedback* the initial estimation $\widehat{\mathbf{h}}$ to obtain an improved estimate, i.e. an estimate which closely verifies the diffusion equation.

From this estimate, let us choose:

$$\mathbf{L}^{ij}_{kl}\left(\nabla\widehat{\mathbf{h}}\right) = \lambda_\perp\left(||\nabla\widehat{\mathbf{h}}||\right)\delta^{ij}\delta_{kl} + \frac{\left[\lambda_{||}\left(||\nabla\widehat{\mathbf{h}}||\right) - \lambda_\perp\left(||\nabla\widehat{\mathbf{h}}||\right)\right]}{||\nabla\widehat{\mathbf{h}}||^2}[\nabla\widehat{\mathbf{h}}^k_i]^T[\nabla\widehat{\mathbf{h}}^l_j]^T \tag{3}$$

where $\lambda_\perp(||\nabla\widehat{\mathbf{h}}||)$ and $\lambda_{||}(||\nabla\widehat{\mathbf{h}}||)$ allows to adjust the smoothing factor in the normal and tangential directions with respect to the gradient.

More precisely, neglecting the spatial variation $\frac{\partial \mathbf{L}^{ij}_{kl}}{\partial x^i}$ of $\mathbf{L}$ we obtain:

$$[\Delta_{\mathbf{L}}\mathbf{h}]^T_k = \lambda_\perp \Delta\mathbf{h}^k + \left[\lambda_{||} - \lambda_\perp\right] \widehat{\eta}^{k\,T} \nabla^2\mathbf{h}^k \widehat{\eta}^k + o(||\partial\mathbf{L}||)$$

with $\widehat{\eta} = \nabla\widehat{\mathbf{h}}/||\nabla\widehat{\mathbf{h}}||$ which indeed corresponds to what has been obtained using the non-linear profile, with:

$$\lambda_\perp(u) = \Phi'(u) \text{ and } \lambda_{||}(u) = \lambda_\perp(u) + 2\,\lambda'_\perp(u)/u^2 \tag{4}$$

We may even choose a more general form for $\lambda_{||}(u)$ in the present case.

In the literature (e.g. as reviewed in [1, 2]), when considering a non-linear profile, at the implementation level, a similar *linear approximation* is used.

In this context, it is thus useless to consider non-linear profiles $\Phi()$ but simply use anisotropic diffusion parameterized by $\mathbf{L}$. Thanks to this derivation, we also obtain an example of coherent form for $\mathbf{L}$, from (3) and (4).

We thus will consider in the sequel that Φ is the identity function and will integrate its influence in the anisotropic diffusion operator parameterized by $\mathbf{L}$.

Implementation of the regularization problem. From the previous Euler equation (2), at the implementation level, a recurrent equation is derived which allows the iterative numerical computation of the function $\mathbf{h}$ values for a mesh of samples (see e.g. [6]). Such an equation is usually of the form:

$$\widehat{\mathbf{h}} \leftarrow \mathbf{h} + \Upsilon \left[\mathbf{\Lambda}\, \bar{\mathbf{h}} + \Delta^*_{\mathbf{L}} \mathbf{h} - \mathbf{\Lambda}\, \mathbf{h} \right] \tag{5}$$

where the regular matrix Υ, with $1 \geq ||\Upsilon|| > 0$ allows to control the iteration convergence. In such a paradigm $\Delta^*_{\mathbf{L}}\mathbf{h}$ is an *implementable approximation* of the continuous second order differential operator $\Delta_{\mathbf{L}}\mathbf{h}$.

At each iteration, the new estimate $\widehat{\mathbf{h}}$ is computed from the previous estimate $\mathbf{h}$, usually using $\bar{\mathbf{h}}$ as initial value. Furthermore the diffusion operator is also recomputed at each step, since as made explicit for instance in (3), it may depend on the previous estimate of $\mathbf{h}$. In order this dependency not to affect the stability of the iteration mechanism, this previous estimate can be "smoothed" to obtain a sufficiently stationary value of $\mathbf{L}$. This aspect will not be further discussed here but we refer to [22] for a development.

As noticed for instance in [19], if $||\Upsilon|| > 0$ fixed points of this iterative equation are solution of the Euler equation, while if $||\Upsilon|| \to 0$ we obtain a stationary value for $\mathbf{h}$ which induces the convergence towards a sub-optimal value.

Furthermore, the matrix Υ is related to the fact that, at each step, we compute the new value of $\mathbf{h}$ form the previous value of $\mathbf{h}$ *solving a "simple" linear system only.* This includes Gauss-Seidel or Jacobi iteration methods and seems to be a general scheme.

The key problem here is thus to obtain a relevant approximation $\Delta^*_{\mathbf{L}}\mathbf{h}$ of the term of diffusion $\Delta_{\mathbf{L}}\mathbf{h}$ in order to correctly and efficiently implement the previous scheme. An estimation of $\nabla\mathbf{h}$ is also required but, as visible in (2), this second problem is particular case with $\mathbf{M}^j_{kl} = \delta_{kl}$ and $\mathbf{L}^{ij}_{kl} = 0$, so that we will focus on the 1st problem.

Can vectorial equations be decoupled ? A step further, we may analyze if we can "decouple" these equations, i.e. compute the new value $\widehat{\mathbf{h}}^k$ in equation (5) as a function of the previous $\mathbf{h}^k$ value only. To obtain this specific case, in the general case, the metric $\mathbf{\Lambda}$ must be diagonal. This also means that $[\Delta_{\mathbf{L}}\mathbf{h}]^k$ must depend on $\mathbf{h}^k$ only. As a consequence, in the general case again, $\mathbf{L}_{kl}^{ij}\frac{\partial \mathbf{h}^l}{\partial x^j}$ must not depend on $\mathbf{h}^{k'}$ for $k' \neq k$. This condition means that $\mathbf{L}_{kl}^{ij} = 0$ for $k \neq l$ i.e. $\mathbf{L}_{kl}^{ij} = \delta_{kl}\mathbf{L}_k^{ij}$ for some $\mathbf{L}_k^{ij}$.

For a generic tensor $n \times n \times m \times m$ tensor $\mathbf{L}_{kl}^{ij}$, we may look for a general linear transformation $\mathbf{h}^l = \sum_b \mathbf{P}_b^l \mathbf{h}^{'b}$ parameterized by the $m \times m$ matrix $\mathbf{P}_b^l$ with $\mathbf{L}_{ab}^{'ij} = \mathbf{P}_a^k \mathbf{L}_{kl}^{ij} \mathbf{P}_b^l = \delta_{ab}\mathbf{L}_a^{ij}$ for some $n \times n \times m$ tensor $\mathbf{L}_a^{ij}$. But since $n \times n \times m \times m > m \times m + n \times n \times m$ as soon as $m > 1$ the problem has no solutions in the general case.

We thus can not decouple diffusion equations in the general case.

This small discussion explains why it is important to consider the "vectorial" case when we regularize a non-scalar function, since we cannot reduce these equations to a set of independent scalar equations.

B: Computing harmonic maps on Riemannian manifold

Harmonic maps on Riemannian manifold. Another interesting problem is the following. Let us consider a differential Riemannian manifold N of dimension n (which may be the Euclidean space $\mathcal{R}^n$ with a differential atlas). Its metric is written $\mathbf{g}$, with $g = det(\mathbf{g})$. The (opposite[1] of the) Laplace-Beltrami operator (e.g. [15]) for a differentiable map $\mathbf{h} : N \to \mathcal{R}^m$ is defined by the formula:

$$\Delta \mathbf{h}^k = \sum_{ij} \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^j} \left(\sqrt{g}\, \mathbf{g}^{ij} \frac{\partial \mathbf{h}^k}{\partial x^j} \right)$$

i.e. when expanded:

$$\Delta \mathbf{h}^k = \sum_j \left[\sum_i \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^j} \left(\sqrt{g}\, \mathbf{g}^{ij} \right) \right] \frac{\partial \mathbf{h}^k}{\partial x^j} + \sum_{ij} \mathbf{g}^{ij} \frac{\partial^2 \mathbf{h}^k}{\partial x^i \partial x^j}$$

and a map is -in this context- harmonic iff $\Delta \mathbf{h} = 0$.

[1] There has been an historical sign error when the Laplacian has been generalized to non-linear spaces.

This operator is a direct generalization of anisotropic diffusion for non-Euclidean spaces (see [29] for a discussion and [17] for applications of this formalism).

If the metric is the identity, i.e. $\mathbf{g} = \mathbf{I}$, this corresponds to an Euclidean space and this operator reduces to the standard definition of the Laplacian. Otherwise, the introduction of a Riemannian metric is equivalent to anisotropic diffusion, since this equation is equivalent to (2) with:

$$\mathbf{L}_{kl}^{ij} = \mathbf{g}^{ij}\,\delta_{kl} \text{ and } \mathbf{M}_{kl}^{j} = \sum_i \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^j} \left(\sqrt{g}\,\mathbf{g}^{ij}\right)\,\delta_{kl} \tag{6}$$

This also shows that there is a direct correspondence between harmonic maps on manifold and non-isotropic regularize maps.

If $g = det(\mathbf{g}) = 1$, i.e. if the Riemannian metric is "isometric", it corresponds to the diffusion operator itself (just compare (6) with $\sqrt{g} = 1$ and (2)). Otherwise, anisotropic diffusion and Riemannian metric are in relation as made explicit here, but not strictly equivalent.

Generalization from $\mathcal{R}^n$ to non-linear manifolds, using the Laplace-Beltrami equation is to be considered for two reasons:
(1) for a robot or another mechanical system the physical parametric space is not always isomorphic to a unique $\mathcal{R}^n$ compact (e.g. for 3D rotations with angles which may be higher that π) but has a more complex mathematical structure;
(2) cortical maps are intrinsically "curved space" with the fact that parametric spaces of dimension $n > 2$ (e.g. the visual areas which code retinal localization, edges orientation, binocular disparity, color .. in a "interlaced" way) are represented onto 2D maps. This indeed means that deep deformations occur, which must be represented by a non-linear space with a variable metric, i.e. a Riemannian manifold.

The good news is that the implementation used to solve the problem **A** can be directly re-used to solve this problem **B**, despite the fact that when parameterizing a Riemannian manifold, several maps have to be taken into account depending the parameter range value, with are not discussed here (see [29] for the description of some additional mechanisms to manage the parameter transformation at the maps intersection).

Relation with general harmonic maps. More generally, for a differentiable map $\mathbf{h} : N \rightarrow M$ where M is a Riemannian manifold of dimension m which metric is written $\mathbf{f}$, we may consider the energy density of the function:

$$e(\mathbf{h})(\mathbf{x}) = \frac{1}{2}\, \mathbf{g}^{ij}(\mathbf{x})\, \mathbf{f}_{ab}(\mathbf{h}(\mathbf{x}))\, \frac{\partial \mathbf{h}^a(\mathbf{x})}{\partial x^i}\, \frac{\partial \mathbf{h}^b(\mathbf{x})}{\partial x^j}$$

and the function energy:

$$E(\mathbf{h}) = \int_N e(\mathbf{h})\, \sqrt{g}\, dx^1 \wedge .. \wedge dx^n$$

which related Euler equation is:

$$\Delta \mathbf{h}^k = \sum_{ij} \frac{1}{\sqrt{g}}\, \frac{\partial}{\partial x^j} \left(\sqrt{g}\, \mathbf{g}^{ij}\, \frac{\partial \mathbf{h}^k}{\partial x^j} \right) + \mathbf{g}^{ij} \sum_{ab} \mathbf{F}^k_{ab}(\mathbf{h})\, \frac{\partial \mathbf{h}^a}{\partial x^i}\, \frac{\partial \mathbf{h}^b}{\partial x^j} = 0$$

where $\mathbf{F}^a_{bc} = \frac{1}{2} \sum_d \mathbf{f}^{ad} \left(\frac{\partial \mathbf{f}_{bd}}{\partial x^c} + \frac{\partial \mathbf{f}_{cd}}{\partial x^b} - \frac{\partial \mathbf{f}_{bc}}{\partial x^d} \right)$ is the Christoffel symbols of the metric $\mathbf{f}$.

Solutions to these equations are called harmonic maps.

If $N = \mathcal{S}^1$ with its standard metric we obtain the geodesic curves of M as solutions, while if $M = \mathcal{R}^m$ we re-obtain the Laplace-Beltrami operator. Intrinsically, this equation means $trace(\nabla d\mathbf{h}) = 0$ where $d\mathbf{h}$ is the differential of $\mathbf{h}$ [15].

Although out of the scope of the present study, it will be an interesting perspective of the present work to generalize what will be developed in the sequel to this kind of equations, the key problem being to choose a relevant linearization of the non-linear term $\mathbf{g}^{ij} \sum_{ab} \mathbf{F}^k_{ab}(\mathbf{h})\, \frac{\partial \mathbf{h}^a}{\partial x^i}\, \frac{\partial \mathbf{h}^b}{\partial x^j}$, e.g. to consider:

$\mathbf{L}^{ij}_{kl} = \mathbf{g}^{ij}\, \delta_{kl}$

and

$$\mathbf{M}^j_{kl} = \sum_i \frac{1}{\sqrt{g}}\, \frac{\partial}{\partial x^j} \left(\sqrt{g}\, \mathbf{g}^{ij} \right) + \frac{1}{2} \left[\mathbf{g}^{ij} \sum_a \mathbf{F}^k_{al}(\widehat{\mathbf{h}})\, \frac{\partial \widehat{\mathbf{h}}^a}{\partial x^i} + \mathbf{g}^{ji} \sum_a \mathbf{F}^k_{la}(\widehat{\mathbf{h}})\, \frac{\partial \widehat{\mathbf{h}}^a}{\partial x^i} \right] \delta_{kl}$$

where $\widehat{\mathbf{h}}$ is a previous estimation of $\mathbf{h}$, as discussed previously.

We will not further develop this point here, but simply wanted to point out perspectives of the present approach.

3 Implementing diffusion operators

In the previous section the importance of a correct implementation, thus approximation, of diffusion operators. It appeared as the key problem, when considering so called "PDE" approaches for the computation of "parameter maps". Let us now discuss how to implement these operators in an efficient but also biologically plausible way.

Integral approximation of the diffusion operator

Introducing integral diffusion operators. As discussed previously, we consider a general second order differential operator:

$$[\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}]_k = \sum_j \mathbf{M}_{kl}^j \frac{\partial \mathbf{h}^l}{\partial x^j} + \sum_{ij} \mathbf{L}_{kl}^{ij} \frac{\partial^2 \mathbf{h}^l}{\partial x^i \partial x^j}$$

including 1st order operator implementation (i.e. with $\mathbf{L} = 0$) in our discussion.

Considering the Dirac distribution $\delta(\mathbf{x})$ (see [27] for a review) we may formally rewrite this equation:

$$[\Delta_{\mathbf{L}}\mathbf{h}]_k = \sum_k \sigma_{kl} * \mathbf{h}^l = \int_{\mathcal{R}^n} \sum_l \sigma_{kl}(\mathbf{x}-\mathbf{y})\, \mathbf{h}^l(\mathbf{y})\, d\mathbf{y}$$

with:

$$\sigma_{kl} = \sum_j \mathbf{M}_{kl}^j \frac{\partial \delta^l}{\partial x^j} + \sum_{ij} \mathbf{L}_{kl}^{ij} \frac{\partial^2 \delta^l}{\partial x^i \partial x^j}$$

where $*$ is the convolution product. In words, there is a direct mathematical and canonical link between differential and integral operator.

Based on this remark and following [5], we approximate this diffusion operator by an integral operator of the form:

$$\left[\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}(\mathbf{x})\right]_k = \int_{\mathcal{R}^n} \sum_l \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y}) \left[\mathbf{h}^l(\mathbf{y}) - \mathbf{h}^l(\mathbf{x})\right] d\mathbf{y}$$

for some distribution $\sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})$ to be discussed now.

Clearly, if $\sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y}) = \sigma_{kl}(\mathbf{x}-\mathbf{y})$, $\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}]_k = \Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}]_k$.

> because $\int_{\mathcal{R}^n} \partial\delta^\alpha = 0, |\alpha| > 0$, so that $\int_{\mathcal{R}^n} \sigma_{kl} = 0$ since σ_{kl} is a linear combination of Dirac distribution derivatives, a straightforward derivation:
>
> $$\int_{\mathcal{R}^n} \sum_l \sigma_{kl}(\mathbf{x}-\mathbf{y}) \left[\mathbf{h}^l(\mathbf{y}) - \mathbf{h}^l(\mathbf{x})\right] d\mathbf{y} =$$
> $$\int_{\mathcal{R}^n} \sum_l \sigma_{kl}(\mathbf{x}-\mathbf{y})\, \mathbf{h}^l(\mathbf{y})\, d\mathbf{y} - \sum_l \underbrace{\int_{\mathcal{R}^n} \sigma_{kl}(\mathbf{x}-\mathbf{y})\, d\mathbf{y}}_{=0} \mathbf{h}^l(\mathbf{x}) =$$
> $$\int_{\mathcal{R}^n} \sum_l \sigma_{kl}(\mathbf{x}-\mathbf{y})\, \mathbf{h}^l(\mathbf{y})\, d\mathbf{y}$$
>
> shows that the two previous integral forms are equal.

As a consequence, the "exact" operator $\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}$ is a particular case of the "approximate" operator $\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}$.

However, it is not possible in practice to implement such a "punctual" operator because what is given at the implementation level is set of "samples". More precisely, we must consider a set of measures, defines as integral values of the continuous function. This legitimates the fact that the discrete version of a differential operator is approximated as an integral operator. Furthermore, numerical values contains uncertainties, it is a reasonable choice to "average" several values in order to smooth these uncertainties. These arguments explain the choice proposed by [5].

Basic properties of the integral operator. For such an approximation to be well-defined, the integral must be convergent. Here, to obtain this property, we assume that $\sigma^\epsilon(\mathbf{x},\mathbf{y})$ has a bounded support $\mathcal{S}$, thus *included in a ball* $\mathcal{B}(\mathbf{x},\epsilon)$ *of radius* ϵ, i.e. $\mathcal{S} \subset \mathcal{B}(\mathbf{x},\epsilon)$.

In addition, this operator must belong to a well-defined functional space. Here following [5] again, we consider a subset of the distributions: the Sobolev function space $W^{s,\infty}(\mathcal{R}^n)$. In fact, we are going to obtain solutions which are ordinary differentiable functions.

A step further, the present integral operator is very easy to interpret if we write:

$$[\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}(\mathbf{x})]_k = \sum_l \left[\int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})\,\mathbf{h}^l(\mathbf{y})\,d\mathbf{y}\right] - \bar{\sigma}^\epsilon_{kl}(\mathbf{x})\,\mathbf{h}^l(\mathbf{x})$$

with $\bar{\sigma}^\epsilon_{kl}(\mathbf{x}) = \int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})\,d\mathbf{y}$. It now appears as an operator which *computes a weighted mean value around* $\mathbf{x}$ *minus the balanced value at* $\mathbf{x}$.

This simply corresponds to the usual implementation of the Laplacian operator, i.e $\Delta\mathbf{h}$. But we work on a generalized form in order to implement anisotropic diffusion mechanisms.

Yet another step further, in coherence with the physical law of conservation property for diffusion processes, we must introduce a "conservation property" in order to guaranty the stability of the numerical computations. In words, we want to balance the $\mathbf{h}$ values in the information diffusion process, i.e. guaranty that if we reduce the value at one location it will increase elsewhere accordingly, as in a fluid which particles are neither created nor deleted. This balance is obtained when the average value of the diffusion operator is equal to zero.

In order to verify this conservation property:

$$\int_{\mathcal{R}^n} \Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}(\mathbf{x})\, d\mathbf{x} = 0$$

we consider that the operator has the following average symmetry:

$$\int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{z})\, d\mathbf{x} = \int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{z},\mathbf{y})\, d\mathbf{y}$$

which is equivalent to the conservation property, as the reader can very easily verify. In the literature [25, 5], authors choose a symmetric operator: $\sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y}) = \sigma^\epsilon_{kl}(\mathbf{y},\mathbf{x})$, although the "exact" operator σ_{kl} verifies the average symmetry property but is not symmetric. It is a more general choice to introduce a-priori a weaker constraint, in order to have a maximal number of degrees of freedom to optimize our solutions. We however will obtain symmetric operators in the sequel.

Relations between integral and differential operators. In order to relate the differential operator $\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}$ with its integral approximation $\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}$, following [5] again, let us identify $\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}$ with the Taylor expansion of $\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}$ at some order r.

Here, we consider the Taylor expansion of $\mathbf{g}(\mathbf{y},\mathbf{x}) = \mathbf{h}(\mathbf{y}) - \mathbf{h}(\mathbf{x})$ with respect to $\mathbf{y}-\mathbf{x}$.

In other words, we consider the following change of variables $\mathbf{d} = \mathbf{y}-\mathbf{x}$ and, say, $\mathbf{s} = (\mathbf{y}+\mathbf{x})/2$ in order to write the expansion with respect to $\mathbf{d}$:

$$\mathbf{g}(\mathbf{y},\mathbf{x}) = \textstyle\sum_{|\alpha|=1}^{r} \frac{\partial^\alpha \mathbf{h}}{\alpha!}(\mathbf{x})\Big|_{\mathbf{y}=\mathbf{x}} (\mathbf{y}-\mathbf{x})^\alpha + O(||\mathbf{y}-\mathbf{x}||^r)$$

and a few algebra yields:

$$\left[\Delta^*_{\mathbf{M},\mathbf{L}}\mathbf{h}\right]_k = \sum_{l=1}^{m} \sum_{|\alpha|=1}^{r} \frac{\partial^\alpha \mathbf{h}^l}{\alpha!} \int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})\, (\mathbf{y}-\mathbf{x})^\alpha\, d\mathbf{y} + [R^r\mathbf{h}]_k$$

where the remainder $R^\epsilon_{kl}\mathbf{h}$ of this expansion may be written using an integral form:

$$[R^r\,\mathbf{h}]_k = \sum_{l=1}^{m} \sum_{|\alpha|=r+1} \frac{r+1}{\alpha!} \int_{\mathcal{R}^n\times[0,1]} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})\, (\mathbf{y}-\mathbf{x})^\alpha\, (1-u)^r\, \partial^\alpha \mathbf{h}^l(\mathbf{x}+u\,(\mathbf{y}-\mathbf{x}))\, d\mathbf{y}\, du$$

From the assumptions we have raised, because the support is included in a ball of radius ϵ, the remainder is bounded by the standard condition:

$$||R^r \mathbf{h}||_{0,\infty} < C\,\epsilon^{r-1}||\mathbf{h}||_{r+1,\infty} \tag{7}$$

where C is a fixed quantity [5].

If we rewrite the diffusion operator with the same notations:

$$[\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}]_k = \sum_{l=1}^{m}\left[\sum_{\mathbf{e}_j}\mathbf{M}_{kl}^{j}\,\partial^{\mathbf{e}_j}\mathbf{h}^l + \sum_{\mathbf{e}_i+\mathbf{e}_j}\mathbf{L}_{kl}^{ij}\,\partial^{\mathbf{e}_i+\mathbf{e}_j}\mathbf{h}^l\right]$$

we easily identify the two expressions and obtain:

$$\begin{array}{rlcll}
r \geq |\alpha| > 2 & & \int_{\mathcal{R}^n}\sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y})\,(\mathbf{y}-\mathbf{x})^{\alpha}\,d\mathbf{y} & = & 0_r \quad [\text{C0}] \\
|\alpha| = 2 & \mathbf{L}_{kl}^{ij}(\mathbf{x}) & - \ \frac{1}{2}\int_{\mathcal{R}^n}\sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y})\,(\mathbf{y}-\mathbf{x})^{\mathbf{e}_i+\mathbf{e}_j}\,d\mathbf{y} & = & 0_r \quad [\text{C2}] \\
|\alpha| = 1 & \mathbf{M}_{kl}^{j}(\mathbf{x}) & - \ \int_{\mathcal{R}^n}\sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y})\,(\mathbf{y}-\mathbf{x})^{\mathbf{e}_j}\,d\mathbf{y} & = & 0_r \quad [\text{C1}]
\end{array}$$

where $0_r = \int_{\mathcal{S}} o(\mathbf{y}-\mathbf{x})^r d\,\mathbf{y}$ is a negligible quantity for sufficiently small ϵ.

These conditions [C0] [C2] and [C1] which allow to identify up to the rth order the two expressions, are used to derive different solutions as detailed in the sequel.

They have the interesting property to be "decoupled" in the sense that they allow, for a given (k,l) to relate each σ_{kl}^{ϵ} to the corresponding $\mathbf{M}_{kl}$ and $\mathbf{L}_{kl}$ independently.

Furthermore, from (7), we have a bound on the error for our approximation, as a function of ϵ^{r-1} i.e.:

$$||\Delta_{\mathbf{M},\mathbf{L}}\mathbf{h} - \Delta^{*}_{\mathbf{M},\mathbf{L}}\mathbf{h}||_{0,\infty} \leq C\,\epsilon^{r-1}||\mathbf{h}||_{r+1,\infty}$$

for a fixed constant C, since $\mathcal{S} \subset B(\mathbf{x},\epsilon)$.

This, for a covering of the parameter space by supports $\mathcal{S} \subset B(\mathbf{x},\epsilon)$ organized in some mesh, indeed allows to conclude that $\lim_{\epsilon\to 0}\Delta^{*}_{\mathbf{M},\mathbf{L}}\mathbf{h} = \Delta_{\mathbf{M},\mathbf{L}}\mathbf{h}$. In words, the approximation converges towards the exact solution when the size of the mesh elements vanishes. In practice, this means that *the better the mesh resolution, the better the approximation.* Fair enough, but in practice the mesh resolution is indeed limited. Considering an image for instance, it means that the highest the number of pixels the better the precision. This is a good

advertising for high resolution sensors. But it does not tell us what to do at a given resolution. This is going to be discussed in the sequel.

In the literature [25, 5], two solutions have being derived for this integral operator, in the case where $\mathbf{M} = \sum_i \frac{\partial \mathbf{L}}{\partial x^i}$ and for scalar functions only. Contrary to the present approach, these solutions do not correspond to a bounded support $\mathcal{S}$ although, indeed, the related integral is still convergent. On the contrary, they are related to cut-off functions defined and integrable in $\mathcal{R}^n$. In other words, the choice of the support is implicitly realized when choosing this cut-off function.

As a comparison with the solution on a bounded support derived in the sequel, let us briefly review these solutions. Here we consider only a scalar function h and omit the related indices.

The Raviat solution. This solution [25], at the origin of the work of [5], is of the form:

$$\sigma^\epsilon(\mathbf{x},\mathbf{y}) = \sum_{ij} \int_{\mathcal{R}^n} L_{ij}(\mathbf{z}) \, \frac{\partial \xi(\mathbf{x}-\mathbf{z})}{\partial x_i} \, \frac{\partial \xi(\mathbf{z}-\mathbf{y})}{\partial x_j} \, d\mathbf{z}$$

where the cut-off function ξ verifies $\int_{\mathcal{R}^n} \mathbf{x}^\alpha \, \xi(\mathbf{x}) \, d\mathbf{x} = \begin{cases} 0 & 1 \leq |\alpha| < r \\ 1 & \alpha = 0 \end{cases}$

This solution is the "more general" in the sense that it has been derived from conditions [C0] [C2] and [C1] (here convergence is now due to the cut-off function profile, not the fact that $\mathcal{S}$ is bounded) with a minimum of additional constraints.

Although formally simple, as readable on the formula, this solution is very heavy to implement because a numerical integration is required at each step in order to obtain $\sigma^\epsilon(\mathbf{x},\mathbf{y})$.

The Degond solutions. These are solutions [5] of the form:

$$\sigma^\epsilon(\mathbf{x},\mathbf{y}) = \frac{1}{\epsilon^{n+2}} \sum_{i,j} m_{ij}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \, \psi_{ij}\left(\frac{\mathbf{y}-\mathbf{x}}{\epsilon}\right)$$

(here equivalently we can choose $\frac{1}{2}\left(m_{ij}(\mathbf{x}) + m_{ij}(\mathbf{y})\right)$ instead of $m_{ij}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)$) with:

$$\mathbf{m} = \mathbf{L} - \frac{v}{n+2} trace(\mathbf{L}) \, \mathbf{I}_{n\times n} \text{ with } v \in \{0,1\}$$

If $\upsilon = 0$, we re-obtain $\psi_{ij} = \frac{\partial^2 \xi}{\partial x_i \partial x_j}$ with a cut-off function ξ which corresponds to the previous case

If $\upsilon = 1$, we obtain $\psi_{ij} = x_i\, x_i\, \theta(||\mathbf{x}||)$ where the cut-off function θ verifies:

$$\int_{\mathcal{R}^+} \rho^{n+1+2\,\alpha}\, \theta(\rho)\, d\rho = \begin{cases} \frac{\sigma_{n-1}\,(n-1)}{n\,(n+2)} & \alpha = 1 \\ 0 & 1 < 2\,\alpha \leq r \end{cases}$$

writing[2] $\sigma_n = \frac{n\,\pi^{n/2}}{\Gamma(n/2+1)}$

Such cut-off functions are calculated for instance considering a profile of the form: $\theta(\rho) = e^{-\beta\,\rho^2} \sum_{d=0}^{2\,d=r} a_d\, \rho^{2\,d}$ with $\beta > 0$ where the constants a_d are easily obtain the linear equations.

This solution is much efficient to implement but has three drawbacks for our purpose:
(i) it does not consider a bounded support and
(ii) among existing solutions this approach does not allow to derive optimal solutions as we are going to perform now,
(iii) it does not consider vectorial maps and any 2nd order differential operators.

3.1 Derivation in the continuous case.

When constructing the solution, in [5], the previous constraints is rewritten using another Taylor expansion with respect to $\mathbf{d} = \mathbf{y} - \mathbf{x}$, writing again $\mathbf{s} = (\mathbf{y} + \mathbf{x})/2$ at some order $s > r$ (to be chosen later):

$$\sigma_{kl}^{\epsilon}(\mathbf{x}, \mathbf{y}) = \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\, \sigma_{kl}^{\epsilon\beta}(\mathbf{s})\Big|_{\mathbf{x}=\mathbf{y}}\, (\mathbf{y} - \mathbf{x})^{\beta} + O(||\mathbf{y} - \mathbf{x}||^s) \tag{8}$$

with $\sigma_{kl}^{\epsilon\beta}(\mathbf{s})\Big|_{\mathbf{x}=\mathbf{y}} = \partial_{\mathbf{y}}^{\beta} \sigma_{kl}^{\epsilon}(\mathbf{x}, \mathbf{y})\Big|_{(\mathbf{x},\mathbf{y})=(\mathbf{s},\mathbf{s})}$ the fact that $s > r$ allowing to have the remainder of the Taylor expansion "negligible" with respect to the previous one (we refer to [5] where the formal development is equivalent).

[2] This constant defines the surface of the boundaries $\partial B(\mathbf{x}, \rho)$ of a ball $B(\mathbf{x}, \rho) = \{\mathbf{y}, ||\mathbf{x} - \mathbf{y}|| < \rho\}$ of center $\mathbf{x}$ and radius ρ in $\mathcal{R}^n$, for $\rho = 1$.

Because, and this is a key point here, the support is bounded we can switch sum and integral and obtain:

$$\int_{\mathcal{R}^n} \sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y})\,(\mathbf{y}-\mathbf{x})^{\alpha}\,d\mathbf{y} = \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{x}) \int_{\mathcal{S}} (\mathbf{y}-\mathbf{x})^{\alpha+\beta}\,d\mathbf{y} + 0_r$$

with $\mathbf{s}=\mathbf{x}$ since $\mathbf{x}=\mathbf{y}$ at the point where $\sigma_{kl}^{\epsilon\beta}(\mathbf{x})$ is considered.

This allows to rewrite the previous conditions [C0], [C1] et [C2] as linear equations with respect to $\sigma_{\beta}^{\epsilon}(\mathbf{x})$ i.e.:

$$\sum_{|\beta|=0}^{s} \frac{\mu_{\alpha+\beta}(\mathbf{x})}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{x}) = 0_r + \left\{ \begin{array}{ll} 0 & 2<|\alpha|\leq r \\ 2\,\mathbf{L}_{kl}^{ij}(\mathbf{x}) & \alpha=\mathbf{e}_i+\mathbf{e}_j \\ \mathbf{M}_{kl}^{j}(\mathbf{x}) & \alpha=\mathbf{e}_j \end{array} \right. \tag{9}$$

using the notation: $\mu_{\alpha}(\mathbf{x}) = \int_{\mathcal{S}}(\mathbf{y}-\mathbf{x})^{\alpha}\,d\mathbf{y}$ so that $\mu_{\alpha}(\mathbf{x})$ is a polynomial of degree (less or equal but in fact) equal to $|\alpha|$.

Since there are $\frac{(n+d)!}{n!\,d!}$ monomial of degree less or equal d with n variables, we can immediately count the number of equations in (9) i.e. $p_{n,r} = \frac{(n+r)!}{r!\,n!} - 1$ in the general case.

On the other hand, the average symmetry of the operator yields:

$$\begin{array}{rcl} \int_{\mathcal{R}^n} \sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{z})\,d\mathbf{x} & = & \int_{\mathcal{R}^n} \sigma_{kl}^{\epsilon}(\mathbf{z},\mathbf{y})\,d\mathbf{y} \\ \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{s}) \int_{\mathcal{R}^n} (\mathbf{z}-\mathbf{x})^{\beta}\,d\mathbf{x} & = & \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{s}) \int_{\mathcal{R}^n} (\mathbf{y}-\mathbf{z})^{\beta}\,d\mathbf{y} \\ \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{s})\,(-1)^{|\beta|}\,\mu_{\beta}(\mathbf{z}) & = & \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\sigma_{kl}^{\epsilon\beta}(\mathbf{s})\,\mu_{\beta}(\mathbf{z}) \end{array}$$

this equalities being verified for all $\mathbf{z}$ if and only if $\sigma_{kl}^{\epsilon\beta}(\mathbf{s})=0$ when $|\beta|$ is odd. As a consequence, from (8), $\sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y})$ is going to be a symmetric operator.

Since there are $\frac{(n-1+d)!}{(n-1)!\,d!}$ monomial of degree d with n variables we finally have to consider $q_{n,s} = \sum_{d=0,d\ \text{even}}^{s} \frac{(n-1+d)!}{d!\,(n-1)!}$ functions $\sigma_{kl}^{\epsilon\beta}(\mathbf{s})$ as unknowns of this set of linear equations.

This allows to calculate, in the general case, the minimal order of expansion:

$$s(n,r) = \inf\{s, q_{n,s} \geq p_{n,r}\}$$

as reported in this table where $s(n,r)$ has been calculated:

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n = 1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
n = 2	2	4	4	6	8	10	10	12	14	16	16	18	20	20	22	24
n = 3	2	4	4	6	8	8	10	12	12	14	14	16	18	18	20	22

while calculating $q_{n,s} - p_{n,r}$:

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n = 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n = 2	2	4	0	2	5	9	1	5	10	16	4	10	17	2	9	17
n = 3	4	13	3	16	40	12	42	88	33	87	9	71	156	36	131	254

it appears that we obtain some redundancy, i.e. more unknowns than equation. This will be discussed in the next paragraph.

The linear system of polynomial equation, written in (9), yields solutions of the form:

$$\sigma_{kl}^{\epsilon\beta}(\mathbf{x}) = \sum_j \mathbf{A}_{j\,kl}^{\beta}(\mathbf{x})\,\mathbf{M}_{kl}^{j}(\mathbf{x}) + \sum_{ij} \mathbf{B}_{ij\,kl}^{\beta}(\mathbf{x})\,\mathbf{L}_{kl}^{ij}(\mathbf{x})$$

where $\mathbf{A}_{j\,kl}^{\beta}(\mathbf{x})$ and $\mathbf{B}_{ij\,kl}^{\beta}(\mathbf{x})$ are rational polynomials of the variables $\mathbf{x}$, which numerators are polynomials of degree less than $d_{\text{numer}} = (n-1)\,(r+s)$, with a common denominator which is polynomial of degree less than $d_{\text{denom}} = n\,(r+s)$ and corresponds to the determinant of the set of equations. As being rational polynomials, they are defined almost every-where and because $d_{\text{denom}} - d_{\text{numer}} = r + s \geq 2$, since $\mathbf{M}$ and $\mathbf{L}$ are bounded, $\sigma_{kl}^{\epsilon\beta}(\mathbf{x}) = o(1/||\mathbf{x} - \mathbf{x}_\bullet||^d), d \geq 2$, in the general case. It is thus integrable, so that the integral operator is well-defined. These "poles" likely approximate the punctual values of the Dirac distribution derivatives: when experimenting several solutions we have verified this assumption.

As a conclusion, the obtained solutions are of the form:

$$\sigma_{kl}^{\epsilon}(\mathbf{x},\mathbf{y}) = \sum_j \mathbf{A}_{j\,kl}(\mathbf{x},\mathbf{y})\,\mathbf{M}_{kl}^{j}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) + \sum_{ij} \mathbf{B}_{ij\,kl}(\mathbf{x},\mathbf{y})\,\mathbf{L}_{kl}^{ij}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)$$

with:

$$\begin{array}{rcl} \mathbf{A}_{j\,kl}(\mathbf{x},\mathbf{y}) & = & \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\mathbf{A}_{j\,kl}^{\beta}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\,(\mathbf{y}-\mathbf{x})^{\beta} \\ & \text{and} & \\ \mathbf{B}_{ij\,kl}(\mathbf{x},\mathbf{y}) & = & \sum_{|\beta|=0}^{s} \frac{1}{\beta!}\,\mathbf{B}_{ij\,kl}^{\beta}\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\,(\mathbf{y}-\mathbf{x})^{\beta} \end{array}$$

They thus are straightforward generalizations of the Degond solutions and have a computational cost which is of the same order of magnitude.

Derivation of an optimal operator. As mentioned previously, as soon as $s > s(n,r)$ we obtain an infinite set of possible solutions. How can we avoid

such an ambiguous situation but make profit out of it? Very simply, by looking for an *optimal* solution among these.

As mentioned before, ideally this operator should be "punctual" i.e. only compute second order derivatives at "one point". Since this is not implementable, we have accepted to consider an integral operator "around" this point. But the more information close to the point are used, the best. In other words, the "sharpest" the operator, the best. This means that we want the weights which are quantified by $\sigma(\mathbf{x},\mathbf{y})$ to be concentrated around $\mathbf{y}=\mathbf{x}$. Such a criterion is quantified, for instance, by minimizing the "inertia" or "variance" of the weights, i.e. their second order momenta. We thus choose:

$$\min_{\sigma^\epsilon} \int_{\mathcal{S}} \Xi(||\mathbf{y}||)\, \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})^2 d\mathbf{y} \tag{10}$$

considering a weight $\Xi(||\mathbf{y}||)$ (for simplicity we restrain to $\Xi(||\mathbf{y}||)=1$ for the present discussion, but we will use this weight in the discrete case) and where, from (8): $\int_{\mathcal{S}} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})^2 d\mathbf{y} = \sum_{\beta\beta'} \sigma^{\epsilon\beta}_{kl}(\mathbf{x})\, \sigma^{\epsilon\beta'}_{kl}(\mathbf{x})\, \mu_{\beta+\beta'}(\mathbf{x})$ in our case.

More formally, we may also require σ^ϵ_{kl} to be around each point $\mathbf{x}$ as closed as possible to the "exact" operator σ_{kl} defined previously, as a linear combination of Dirac distribution derivatives. Using weighted quadratic distances again, we may formalize this proximity by the following criterion :

$$\min_{\sigma^\epsilon} \int_{\mathcal{S}-B(\mathbf{x},\varepsilon)} \Xi(||\mathbf{y}||)\, [\sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y}) - \sigma_{kl}(\mathbf{x}-\mathbf{y})]^2\, d\mathbf{y}$$

where $B(\mathbf{x},\varepsilon)$ is a ball around $\mathbf{x}$ of radius ε which can be as small as possible. We must exclude a small neighborhood around $\mathbf{x}$ because this quantity contains a product of two distribution which is not defined at $\mathbf{x}$ [27]. Furthermore, because the support of σ_{kl} is $\{\mathbf{x}\}$, as being a linear combination of Dirac distribution derivatives, we immediately obtain:

$\int_{\mathcal{S}-B(\mathbf{x},\varepsilon)} a\Xi(||\mathbf{y}||)\, [\sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y}) - \sigma_{kl}(\mathbf{x}-\mathbf{y})]^2\, d\mathbf{y} = \int_{\mathcal{S}-B(\mathbf{x},\varepsilon)} \sigma^\epsilon_{kl}(\mathbf{x},\mathbf{y})^2\, d\mathbf{y}$

This criterion is, for sufficiently small ε, as close as possible to the chosen criterion (10), so that both approaches are in practice equivalent.

We thus have a quadratic criterion with the linear constraints (9), to be solved with respect to $\sigma^{\epsilon\beta}_{kl}(\mathbf{x})$ which are our unknowns. There is a unique solution in the general case, indeed rather heavy to write as a formula, but obvious to compute using a symbolic calculation.

Let us illustrate this result, for instance, considering a support $\mathcal{S} = [-1/2..1/2]^n$, i.e. an hyper-cube in $\mathcal{R}^n$. The following piece of maple code (see appendix A for some macros) returns the optimal solution:

```
# Define the kernel of the integral approximation of a diffusion operator,
sigma := proc(
 M, L, # .. defined by the 1st order coeffs M(u) and 2nd order coeffs L(u,v)
       #    as functions of x,
 Int,  # .. for the support integral Int(f(x), x),
 x :: vector, y :: vector, # .. returning an operator sigma(x, y),
 r :: integer,             # .. identify up to the r-th order,
 s :: integer              # .. using expansion up to the s-th order.
)
  option remember: local n, d, i, j, sigma, c: n := vectdim(x):
  # Define the sigma profile
  d := `union`(op(map(indexes, {$0..s}, n))):
  sigma := convert(map(d -> cat(sigma_, op(d)) * pow(evalm(y - x), d), d), `+`):
  # Return the optimal solution,
  c := d -> Int(sigma * pow(evalm(y - x), d), y):
  subs(leastsquare(
   # .. minimizing the required criterion,
   Int(sigma^2, y),
   # .. verifying the average symmetry property,
   {coeffs(
     subs(map((k,x,y) -> x[k]=y[k],{$1..n},x,y),Int(sigma, y)) - Int(sigma, x),
    convert(y, set)),
   # .. and the [C1] [C2] and [C0] conditions,
   'M(i)    - c(map(dirac, [$1..n], i))'$i=1..n,
   ''2*L(i,j) - c(map((a,b,c)->dirac(a,b)+dirac(a,c), [$1..n], i, j))'$j=i..n'$i=1..n
   } union map(c, `union`(op(map(indexes, {$3..r}, n)))),
   # .. for the sigma profile variabes.
   map(d -> cat(sigma_, op(d)), d)),
   map((k,x,y) -> x[k]=(x[k]+y[k])/2,{$1..n},x,y),
   sigma)
 end:
```

and we have been able to experiment that the derived solution verify the expected properties: symmetry of the operator, degrees of numerator and denominator (we always obtain degrees corresponding to the general case). As illustrated in Fig. 1, we have noticed the importance of using a reasonable order $r = 5..8$ for 1st or 2nd order operators with $s \simeq s(n, r) + 1..2$ in order to obtain a solution which is qualitatively correct, making profit of the optimality condition (10).

Here, as a symbolic computation, solution with rational number is output. Although directly usable, `evalf()` functions (allowing to take floating point approximations of fractional integer) may be used for large derivations, in order to bound the derivation complexity.

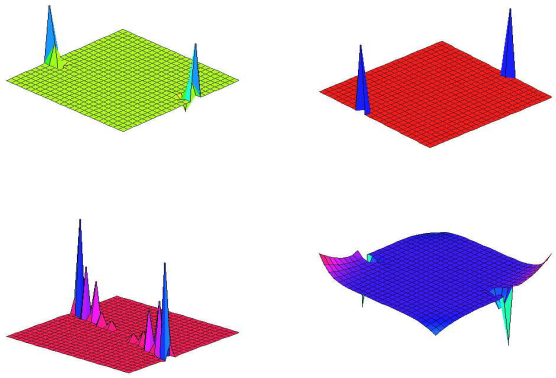


Figure 1: A few examples of operator 1D-profiles, considering an isotropic second-order derivative, represented in function of (x, y); *top-left view* r = 5, s = 10: we obtain a profile with two poles qualitatively equivalent to the δ'' distribution; *top-right view* r = 8, s = 20: increasing the order of correspondence, a profile closer to δ'' is obtained; *bottom-left view* r = 2, s = 3: when the correspondence is insufficient (r is too small) we obtain a profile which is still qualitatively correct but very "flat"; *bottom-right view* r = 6, s = 10: when considering without any redundancy, the approximation may be slightly biased with spurious effects.

A step further, this software tool allows to verify that the present mechanism generates coherent kernels for various differential operators as briefly illustrated in Fig. 2.

3.2 Derivation in the discrete case.

We consider now computer implementations on 2D or 3D dense data "images", i.e. a regular mesh of hyper-cubes.

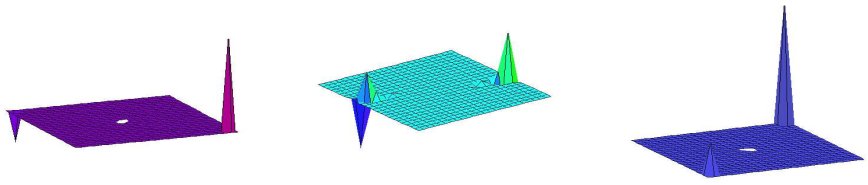


Figure 2: A few examples of operator 2D-profiles, with r = 3, s = 6, represented in the (x^0, x^1) plane; *left view* approximation of 1st order derivative isotropic operator $\partial^{(1,0)}$ qualitatively equivalent to the corresponding continuous operator $\delta^{(1,0)}$; *middle view* approximation of 2nd order non-isotropic operator $L^{ij}(\mathbf{x}) = \delta^{ij}\, x^0$ and *right view* a 2nd-order non-isotropic operator $L^{ij} = \delta^{ij} + i$, both illustrating how solutions adapt to such profiles.

The integral approximation must be sampled and becomes a summation:

$$\Delta^*_{\mathbf{L}}\mathbf{h}^k(\mathbf{x_u}) = \sum_l \sum_{\mathbf{y_v}=(v_1,\cdots,v_n)} \sigma^\epsilon_{kl}(\mathbf{x_u},\mathbf{y_v}) \left[\mathbf{h}^l(\mathbf{y_v}) - \mathbf{h}^l(\mathbf{x_u})\right] \text{ with } \mathbf{x_u} = (u_1,\cdots,u_n) \tag{11}$$

Here, we consider without loss of generality, that pixel/voxel hyper-cubes volume is 1.

In this context, a reasonable model [35, 19] for a "pixel" or "voxel" signal is to assume that:

$$\mathbf{h}(\mathbf{x_u}) = \int_{\left[u_1-\frac{1}{2}..u_1+\frac{1}{2},\cdots,u_n-\frac{1}{2}..u_n+\frac{1}{2}\right]} \mathbf{h}(\mathbf{x})\, d\mathbf{x}$$

i.e. that the *pixel/voxel value is the average value of the signal* distribution over its domain.

Furthermore, if we consider that $\sigma^\epsilon(\mathbf{x},\mathbf{y})$ *is piece-wise constant*, it appears that rewriting the integral operator as a summation is NOT an approximation but an *unbiased implementation* of these differential operators since:

$$\begin{array}{l}
\Delta^*_{\mathbf{L}} h(\mathbf{x_u}) = \sum_l \int_{\mathcal{R}^n} \sigma^\epsilon_{kl}(\mathbf{x_u},\mathbf{y}) \left[\mathbf{h}^l(\mathbf{y}) - \mathbf{h}^l(\mathbf{x_u})\right] d\mathbf{y} = \\
\sum_l \sum_{\mathbf{y_v}=(v_1,\cdots,v_n)} \sigma^\epsilon(\mathbf{x_u},\mathbf{y_v}) \int_{\mathbf{y}\in\left[v_1-\frac{1}{2}..v_1+\frac{1}{2},\cdots,v_n-\frac{1}{2}..v_n+\frac{1}{2}\right]} \left[\mathbf{h}^l(\mathbf{y}) - \mathbf{h}^l(\mathbf{x_u})\right] d\mathbf{y} = \\
\sum_l \sum_{\mathbf{y_v}=(v_1,\cdots,v_n)} \sigma^\epsilon(\mathbf{x_u},\mathbf{y_v}) \left[\mathbf{h}^l(\mathbf{y_v}) - \mathbf{h}^l(\mathbf{x_u})\right]
\end{array}$$

The key point here, is that we can directly compute the optimal values of the operator in the discrete case, without deriving an approximation from the continuous case.

More precisely, with a few algebra, conditions [C0], [C1] et [C2] reduce to:

$$\sum_{\mathbf{v}} \nu^{\alpha}_{\mathbf{v}}(\mathbf{x_u})\, \sigma^{\epsilon}_{kl}(\mathbf{x_u}, \mathbf{y_v}) = 0_r + \left\{ \begin{array}{ll} 0 & 2 < |\alpha| \leq r \\ 2\,\mathbf{L}^{ij}_{kl}(\mathbf{x_u}) & \alpha = \mathbf{e}_i + \mathbf{e}_j \\ \mathbf{M}^{j}_{kl}(\mathbf{x_u}) & \alpha = \mathbf{e}_j \end{array} \right. \tag{12}$$

using the notation: $\nu^{\alpha}_{\mathbf{v}}(\mathbf{x_u}) = \int_{\left[v_1-\frac{1}{2}..v_1+\frac{1}{2},\cdots,v_n-\frac{1}{2}..v_n+\frac{1}{2}\right]} (\mathbf{y}-\mathbf{x_u})^{\alpha}\, d\mathbf{y}$. Here, since $\mathbf{x_u}$ takes constant values, $\nu^{\alpha}_{\mathbf{v}}(\mathbf{x_u})$ is a constant.

Similarly, the average symmetry condition leads to:

$$\forall \mathbf{z_w}, \sum_{\mathbf{u}} \sigma^{\epsilon}_{kl}(\mathbf{x_u}, \mathbf{z_w}) = \sum_{\mathbf{v}} \sigma^{\epsilon}_{kl}(\mathbf{z_w}, \mathbf{y_v})$$

which again induces that the operator is symmetric, as the reader can easily verify.

Finally , the optimal criterion proposed in (10) is now of the form:

$$\min \sum_{\mathbf{v}} \Xi(||\mathbf{v}||) \sigma^{\epsilon}_{kl}(\mathbf{x_u}, \mathbf{y_v})^2$$

Here, we choose a weight of the form: $\Xi(u) = 1/(1+u^2)$.

It is immediate to count that in an hyper-cube $[-S..S]^n$ of size $(2\,S+1)^n$, for a given $\mathbf{x_u}$, the symmetric bi-variate operator has $q'_{n,S} = (2\,S+1)^n$ independent coefficients $\sigma^{\epsilon}_{kl}(\mathbf{x_u}, \mathbf{y_v})$. Since, as in the continuous case, there are $p_{n,r}$ equations, we calculate again, in the general case, the minimal value of S for a given order r and dimension n allowing to solve the linear equations:

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n = 1	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
n = 2	1	1	1	2	2	3	3	3	4	4	4	5	5	5	6	6
n = 3	1	1	1	2	2	2	2	3	3	3	4	4	4	4	5	5

Considering the vector:

$$\Sigma = \left[\sigma^{\epsilon}_{kl}(\mathbf{x_u}, \mathbf{y_v})_{\mathbf{v}\in[-S..S,..,-S..S]}\right]^T$$

containing the coefficients of this symmetric tensor, the optimal criterion corresponds to $\min||\Sigma||^2$, we thus have to find a vector of minimal magnitude,

verifying the linear equations (12). We may rewrite these equation:

$$\mathbf{A}\,\Sigma = \mathbf{b} \text{ with } \begin{array}{l} \mathbf{b} = \left[\,..\, \mathbf{M}^j_{kl}(\mathbf{x_u}) \,..\, 2\,\mathbf{L}^{ij}_{kl}(\mathbf{x_u}) \,..\, 0 \,..\, \right]^T \\ \mathbf{A} = (\,..\, \nu^\alpha_{\mathbf{v}}(\mathbf{x_u}) \,..\,) \end{array}$$

and from the S.V.D. of $\mathbf{A}$, written, say, $\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$, we immediately obtain the solution: $\Sigma = \mathbf{V}\,\mathbf{D}^{-1}\,\mathbf{U}^T\,\mathbf{b}$ (see e.g. [26] for a review).

We thus obtain an explicit solution of the form:

$$\sigma^\epsilon_{kl}(\mathbf{x_u}, \mathbf{y_v}) = \sum_j \mathbf{A}_{j\,kl}(\mathbf{x_u}, \mathbf{y_v})\,\mathbf{M}^j_{kl}\,(\mathbf{x_u}) + \sum_{ij} \mathbf{B}_{ij\,kl}(\mathbf{x_u}, \mathbf{y_v})\,\mathbf{L}^{ij}_{kl}\,(\mathbf{x_u})$$

which the same general properties as the continuous operator. It is not a surprise to obtain similar solutions, since we follow the same method, the only difference being to consider piece-wise constant profiles in this case.

Here $\mathbf{A}_{j\,kl}(\mathbf{x_u}, \mathbf{y_v})$ and $\mathbf{B}_{ij\,kl}(\mathbf{x_u}, \mathbf{y_v})$ are arrays of constant values, i.e. "masks" determined by the previous derivation.

As an illustration, for an isotropic 2D operator ($\mathbf{M}^i = 0$ and $\mathbf{L}^{ij} = \delta^{ij}$) we obtain the masks reported in Fig. 3. Although this does not bring much with respect to usual operators, it allows to verify the coherence of the obtained results.

$$\begin{bmatrix} .1048 & .1271 & .1048 \\ .1271 & .0725 & .1271 \\ .1048 & .1271 & .1048 \end{bmatrix} \qquad \begin{bmatrix} .02233 & .03182 & .03693 & .03182 & .02233 \\ .03182 & .05337 & .06462 & .05337 & .03182 \\ .03693 & .06462 & .03695 & .06462 & .03693 \\ .03182 & .05337 & .06462 & .05337 & .03182 \\ .02233 & .03182 & .03693 & .03182 & .02233 \end{bmatrix}$$

Figure 3: Isotropic 2D-masks obtained for $r = 2$ or 3 and $s = 1$ (*left array*) and $s = 2$ (*right array*). In both cases, we obtain a contribution decreasing with the eccentricity, and isotropic in the horizontal/vertical directions as expected. Since we have considered an integral over the domain without excluding the domain center, there is also a (small) contribution of the central pixel. These masks have been normalized in the sense that $\bar{\sigma} = 1$.

Similarly, an anisotropic 2D-operator is reported in Fig. 4 showing how we directly obtained the convolution mask from the "feedback" vector. Here, for

simplicity, we assume that the gradient $\nabla \widehat{h}_i^k$ is defined along the edge normal, i.e. in a unique direction $\mathbf{n}$, so that $\nabla \widehat{h}_i^k = \frac{\partial \widehat{h}^k}{\partial \mathbf{n}_i} \mathbf{n}_i$.

In order to understand the behavior of this mask, let us consider a normalized feedback vector, i.e. $||\hat{\mathbf{n}}||^2 = 1$, with e.g. $\hat{n}_x = 0$, yielding: $\begin{bmatrix} .1048 & .5053 & .1048 \\ -.2511 & .07255 & -.2511 \\ .1048 & .5053 & .1048 \end{bmatrix}$. It is obvious that this 2nd order anisotropic derivation mask corresponds to a second order derivative in the y direction and a smoothing in the x direction as expected.

$$\begin{bmatrix} .1048\,||\hat{\mathbf{n}}||^2 + .3782\,\hat{n}_x\,\hat{n}_y & .5053\,\hat{n}_y^2 - .2511\hat{n}_x^2 & .1048\,||\hat{\mathbf{n}}||^2 - .3782\,\hat{n}_x\,\hat{n}_y \\ .5053\,\hat{n}_x^2 - .2511\hat{n}_y^2 & .07255\,||\hat{\mathbf{n}}||^2 & .5053\,\hat{n}_x^2 - .2511\hat{n}_y^2 \\ .1048\,||\hat{\mathbf{n}}||^2 - .3782\,\hat{n}_x\,\hat{n}_y & .5053\,\hat{n}_y^2 - .2511\hat{n}_x^2 & .1048\,||\hat{\mathbf{n}}||^2 + .3782\,\hat{n}_x\,\hat{n}_y \end{bmatrix}$$

Figure 4: An example of anisotropic 2D-mask in the direction $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y)$ obtained for $r = 2$ or 3 and $s = 1$. See text for details.

These discrete implementations of the present diffusion operator is going to be experimented in the next section.

Implementing the iteration scheme. As revisited in (5), for a control variable $\nu \in]0..1]$, we immediately obtain an iterative scheme to solve the regularization equation.

In the present context, from a few algebra left to the reader, the iterative scheme may be written:

$$\begin{aligned} \mathbf{h}^o(\mathbf{x_u}) \;=\; \textstyle\sum_{kl} \Upsilon_k^o \Big[\mathbf{h}^k(\mathbf{x_u}) + \nu \;\; & \Big[\mathbf{\Lambda}_l^k\, \bar{\mathbf{h}}^l(\mathbf{x_u}) + \\ & \Big[\textstyle\sum_{\mathbf{v},\mathbf{y_v} \in \mathcal{S}} \sigma_{kl}^\epsilon(\mathbf{x_u}, \mathbf{y_v})\, \mathbf{h}^l(\mathbf{y_v}) \Big]^T \Big] \Big] \end{aligned} \tag{13}$$

with $\begin{cases} \Upsilon &= [1 + \nu\,[\Lambda + \bar{\sigma}]]^{-1} \\ \bar{\sigma} &= \sum_{\mathbf{v},\mathbf{y_v} \in \mathcal{S}} \sigma(\mathbf{x_u}, \mathbf{y_v}) \end{cases}$ while the reader will easily verify that if $\nu > 0$ the fixed point of this iterative scheme is the Euler-Lagrange equation solution, while if $\nu \to 0$ the scheme converges (not necessarily towards the exact solution but to a sub-optimal solution) as used in [19].

We have derived this formula by simply solving the linear equation (5) with respect to $\mathbf{h}(\mathbf{x_u})$.

As far as biological plausibility is concerned, another "simpler" scheme is proposed:

$$\begin{aligned} \mathbf{h}^k(\mathbf{x_u}) \; += \; & \nu \sum_k \Big[\mathbf{\Lambda}_l^k \, \Big(\bar{\mathbf{h}}^l(\mathbf{x_u}) - \mathbf{h}^l(\mathbf{x_u}) \Big) \\ & + \Big[\sum_{\mathbf{v},\mathbf{y_v} \in \mathcal{S}} \sigma_{kl}^{\epsilon}(\mathbf{x_u}, \mathbf{y_v}) \, \mathbf{h}^l(\mathbf{y_v}) - \bar{\sigma}_{kl} \, \mathbf{h}^l(\mathbf{x_u}) \Big]^T \Big] \end{aligned} \tag{14}$$

i.e. considering $\Upsilon = \nu \, 1$, in (5). It is less efficient because we do not solve the linear part of the equation at each step, but it has the advantage to lead to a simpler iterative mechanism, without any "matrix inversion" and thus closer to the computing capability of neuronal network (see [8] for detailed discussions on biological plausible computations).

In fact, for small values of ν, the second scheme (14) is close to the first scheme (13) because (14) corresponds to the 1st expansion of (13) with respect to ν.

Experimenting considering a motion estimation problem

Video RGB image sequences

As an illustration we consider sequences of color RGB images (the color being represented using the red, green and blue channels) with the goal of computing the motion between two consecutive frames. Here, we have a standard video camera with two interlaced frames with a 20 msec delay between them. We choose to compute the inter-frame motion, as illustrated in Fig. 5. We do not compute the motion between two consecutive images because, with standard low-cost acquisition devices, the acquisition rate is not fixed (it depends on the computer load) and is usually slow (typically 1-5Hz with more than 10 pixels of disparity between two consecutive images for common mobile objects) so that occlusions, variation in object aspects, etc.. introduce a bias in the motion estimation. Computing the retinal motion with a 20 msec delay is thus preferable but requires to work with small quantities thus numerically not very stable, so that the present regularization framework is very useful in this case. This is thus a relevant application of our formalism.

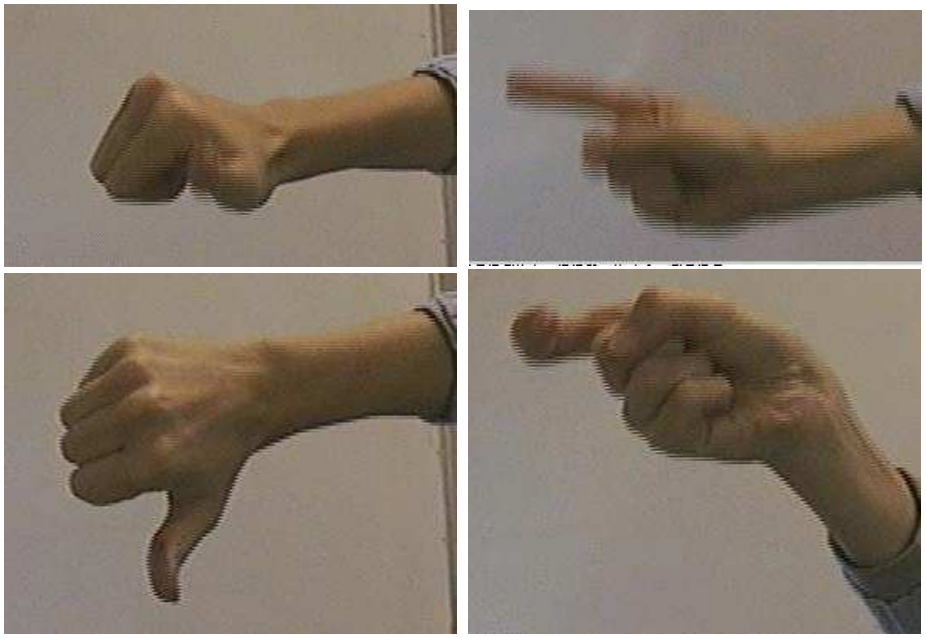


Figure 5: A few examples of interlaced images with a hand displacement, as used in this experimentation: the inter-frame displacement is visible at the hand boundary.

In order to compare our results with other motion estimation methods, we also consider the "Otte" [23] image sequence as shown in Fig. 6, which is of common use for benchmarking such algorithms.

Image gradient of RGB images

At a given image location $\mathbf{m} = (x, y)^T$ with a RGB image intensity $\mathbf{I} = (red, green, blue)^T$, the image contrast and gradient is decomposed using the following singular value decomposition:

$$\begin{array}{c} \mathbf{G} = \frac{\partial \mathbf{I}}{\partial \mathbf{m}} = c\, \mathbf{i_n}\, \mathbf{n}^T + c'\, \mathbf{i_t}\, \mathbf{t}^T \\ \text{with} \\ \mathbf{n} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \text{ and } \mathbf{t} = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix} \end{array} \tag{15}$$

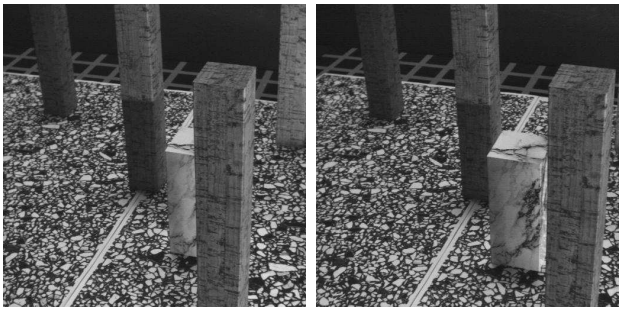


Figure 6: The "Otte" [23] image sequence, of common use to benchmark motion estimation algorithms.

where $c \geq c' \geq 0$ are the singular values, while $\mathbf{i_n} \in \mathcal{R}^3$, $\mathbf{i_t} \in \mathcal{R}^3$ are the orthonormal RGB singular vectors and $\mathbf{n}$ and $\mathbf{t}$ the retinal singular vectors.

More precisely, along an edge, we expect to have only one direction of intensity variation. This means that $c' \simeq 0$ thus $\mathbf{G} \simeq c\,\mathbf{i_n}\,\mathbf{n}^T$ and (i) θ corresponds to the edge orientation in the image, (ii) c corresponds to the edge magnitude and (iii) the unary vector $\mathbf{i_n}$ the edge color orientation in the RGB space. This is a straightforward generalization of edge orientation and magnitude definitions for monochromatic images.

This definition is interesting because it does not depend upon the color space, providing it is in linear relation with the RGB intensity. More precisely, any linear transformation of the RGB channel, say $\mathbf{I}' = \mathbf{M}\,\mathbf{I}$, yields $\mathbf{G}' = \mathbf{M}\,\mathbf{G}$ and, thanks to the SVD properties, the same orientation θ is calculated, while the edge magnitude and color orientation is now given in the new color space.

Initial motion map estimation

The retinal motion estimation $\mathbf{h} : \mathcal{R}^2 \rightarrow \mathcal{R}^2$ associates to a retinal point $\mathbf{x} = (u, v)^T$ a displacement $\mathbf{h}(u, v) = (\dot{u}, \dot{v})^T$. In our implementation, the input $\bar{\mathbf{h}}$ is obtained considering a standard multi-scale correlation operator. The sub-pixelic displacement and the displacement least-square precision (i.e. covariance in a statistical framework) is computed using a 2nd order approx-

imation of the inter-correlation profile[3]. This operator provides a relevant initial estimate but with several drawbacks: in uniform regions there is no local motion information, along edges this information is only available in a direction normal to the edge, in spurious (e.g. with pseudo-periodic patterns) regions erroneous estimation occurs. These are yet another reasons to use a regularization mechanism. Here we consider an affine approximation of the obtained displacement field as initial value.

As reviewed in [1] and well formalized e.g. in [2], it is also indeed possible to estimate the retinal motion directly using, in our equation, measures taken in the image intensity. The image intensity derivatives are in direct and linear relation with the retinal motion, considering the 1st order conservation equation (the so called Horn and Schunk equation, with several limitations, e.g. [34]) or other differential relation (e.g. [33] for 2nd order equations). We will not follow this track here because cortical motion computations[4] are indeed builded on low-level motion detector (represented here by the correlation operators and present in the LGN or V2 cortical areas) and higher-level motion estimator as observed in MT and MST. Here, we are concerned with biologically plausible implementation of such motion estimators.

[3]See, in `http://www-sop.inria.fr/odyssee/imp` the `imp.ima.InterlacedImages.Correl` class for more details.

[4]Here, we have in mind the dorsal cortical visual pathway, sometimes called magnocellular visual pathway extension, in which the neurons in layer 4B of V1 project to the thick stripes of V2 is considered. Area V2 then projects to V3, V5 (or MT, middle-temporal cortex), and MST (medial superior temporal cortex). This pathway is an extension of the magnocellular pathway from the retina and LGN, and continues the processing of visual detail leading to the perception of shape in area V3 and movement in areas V5 and MST. See [3] for a discussion.
As far as motion perception is concerned, simplifying the situation: cells in V5 are particularly sensitive to small moving objects or the moving edge of large objects; cells in dorsal MST respond to the movement of large scenes such as is caused with head movements; cells in ventral MST respond to the movement of small objects against their background. See for instance [8], Chap 10 for a discussion.
Such magnocellular neurons, deeply related to motion perception, show a high sensitivity to contrast, low spatial resolution, and high temporal resolution or fast transient responses to visual stimuli. These cellular characteristics make the magnocellular division of the visual system especially able to quickly detect novel or moving stimuli.

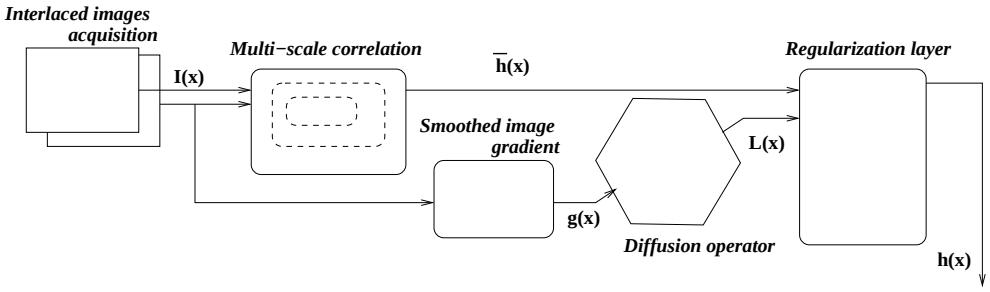


Figure 7: Schematic description of our implementation (see text for details): correlation based estimation of motion may be related to early layers of the visual pathways while the regularization processes are likely to correspond to cortical "motion areas".

Following the methodology proposed in [22], we refine this initial estimate using the present discrete implementation and considering the linearized diffusion operator proposed in (3). We thus simply choose $\nabla\widehat{\mathbf{h}} = c\,\mathbf{n}$ as obtain from (15), since we want to diffuse the information in uniform areas but not across edges. Furthermore we consider sharp $\lambda_\perp$ and$\lambda_{||}$ profiles i.e., for some contrast threshold τ and some regularization coefficient μ:

if $c < \tau$ then $\lambda_\perp = \lambda_{||} = \mu$ (isotropic diffusion in areas without edges)
if $c > \tau$ then $\lambda_\perp = \mu$ and $\lambda_{||} = 0$ (diffusion along but not across edges).

and have used the 3×3 masks shown in Fig. 3 and Fig. 4.

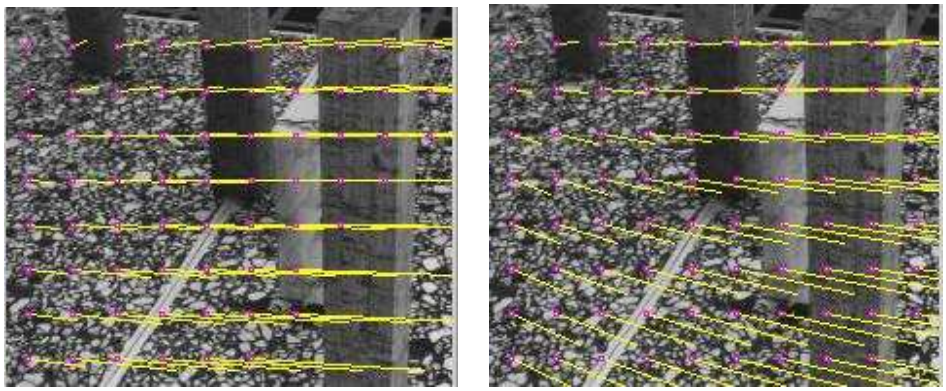
A typical experimental results is proposed in Fig. 8 for the "Otte" sequence.

Similarly, experimental results for interlaced images with a hand displacements are proposed in Figure. 9

Although very preliminary, this small illustrative experiment allowed to verify the coherence of our derivations on concrete examples.

4 Conclusion

An unbiased implementation of differential operators used in the implementation of diffusion processes required in regularization mechanisms has been



Isotropic *Anisotropic*

Figure 8: Result example obtained on the "Otte" sequence. *Isotropic* is, for comparison, what is obtained using an isotropic diffusion; *Anisotropic* is what is obtained by the present implementation. Although similar, in the last case we the regularization yields a less regular displacement field, taking local edge into account.

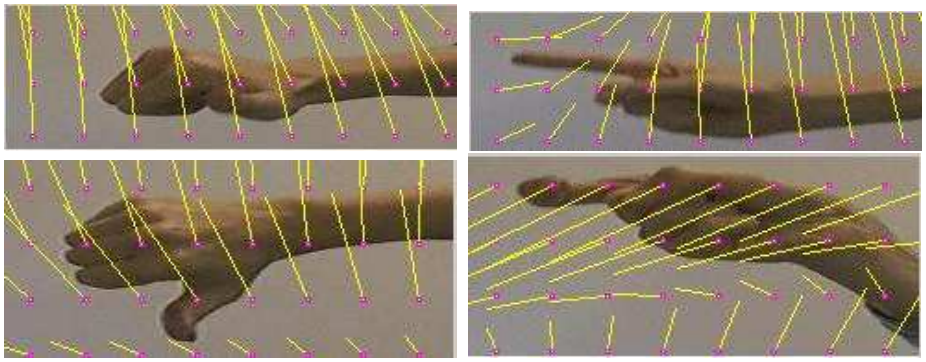


Figure 9: A few examples of hand displacement estimation using the regularized displacement field obtained from our experimentation.

proposed, using a method similar to [35] for ordinary image derivative operators.

Here, the proposed scheme seems to have the following advantages with respect to usual frameworks.

- It is designed to derive "vectorial" operators, i.e. to regularize vectorial maps. An interesting extension of the present work would be to consider not only vectorial but quantities with more complex properties such as transformation groups or diffusion tensors [32], may be following the track initiated by [13]. In our formalism this would be to represent the parameter space by implicit equations and minimize the same criterion but with constraints. Please refer to [13] for an extended discussion.

- It provides a way to "automatically" derived unbiased discrete implementation of the continuous equations derived by th regularization theory. Here "automatic" means not only a small computer tools which save some time (and some bugs :) when obtaining numerical schemes but also a direct track between the pixel/voxel formation model and the discrete operator. More precisely, our work relies on the fact that a pixel/voxel averages the "image" intensity over its surface/volume. This somehow restrictive assumption can be easily generalized if one is able to calibrate the image formation onto the sensor. For instance generalization to any linear convolution operator is straightforward.

- A step further, we have shown that if we consider bounded supports for the integral operator the so called Degond and Mas-Gallic approach reduces to a very simple scheme:
  (i) based on the identification of the Taylor expansion of the integral approximation with the original operator,
  (ii) finding an approximation as closed as possible to the original operator,
  defined from Dirac distributions derivatives. It is clear that what has been derived here is only a first outline of this double idea and that the "as-closed-as-possible" notion is to be worked out.

As far as biological plausibility is concerned, such operator open the door for considering *networks of biological neurons* (i.e. cortical maps computations) as something very different with respect to *neuronal networks*. From a computational point of view the present development shows that even rather

complex computations, if "correctly" implemented, may be obtained by a very simple computation map only dealing with an iterative local weighted averaging. The key idea is the notion of feedback, as pointed out recently by [3] in the visual cortical pathways. The "PDE" manifest may be caricatured as follows : if you feedback "correctly", you can compute a very complex non-linear global quantity using a simple local linear "suitable" diffusion operator. The formal derivations proposed here have used the so called Degond and Mas-Gallic approach in order to throw a bridge between the continuous well-defined PDE theory and what we are able to perform in practice when only considering a finite set of "particles" (this term being used in the original theory), e.g. an image of pixels/voxels or a set of neuronal units.

In the cortex, the "neuronal unit" is a cortical column. Let us end this discussion considering how our implementation may be mapped onto usual computational model of cortical columns processes (see [4] for a treatise on the subject, as already proposed by [18], although a "processing unit" may be more at the scale of maxi-columns [14] as discussed in [24]. Geometrically, these authors introduced the general notion of bundle while we restrict our formalization to the more specific notion of vectorial map. Although out of the scope of the present paper, such models are expected to explain several non-trivial properties of cortical computations, e.g. the local redundancy of neuronal signals [14, 24] providing an anisotropic process has been introduced in particular the fact that connections are not isotropic but, e.g. for edge detection in the diffusion direction [16] (i.e. in the direction which is coded by a neuron).

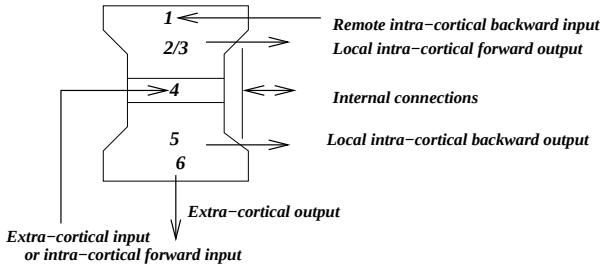


Figure 10: Schematic description of a cortical column, from [4].

Our mechanism may be linked to the cortical column architecture, represented in Fig. 10 as follows:
- *extra cortical input or intra-cortical input from previous layers* (layers IV of the cortex) corresponds to the input variable $\bar{\mathbf{h}}$,
- *extra cortical or backward intra-cortical output* (layers V of the cortex) corresponds to the output the computed variable $\mathbf{h}$,
- *local intra cortical connections* (layers II/III of the cortex) corresponds to the integration over the operator variable $\mathbf{y_v}$,
- *backward intra cortical connections* (layers I of the cortex) corresponds to the inter-parameter interaction, i.e. the different matrices indexed by k and l,
- *internal connections* correspond to the iterative operations in (14).

Indeed, this is a very "prospective" assumption, just thrown here as rough proposal to encourage further investigations about the links between recent computer vision PDE approaches and what has been observed in the primate cortex, the present work being a small step in this direction.

References

[1] L. Alvarez and J. Morel. Formalization and computational aspects of image analysis. *Acta Numerica*, pages 1–59, 1994.

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

[3] J. Bullier. Intergrated model of visual processing. *Brain Res. Reviews*, 36:96–107, 2001.

[4] Y. Burnod. *An adaptive neural network: the cerebral cortex*. Masson, Paris, 1993. 2nd edition.

[5] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. *Mathematics of Computation*, 53(188):485–525, 1989.

[6] R. Deriche and O. Faugeras. Les EDP en Traitement des Images et Vision par Ordinateur. *Traitement du Signal*, 13(6), 1996.

[7] R. Deriche, O. Faugeras, G. Giraudon, T. Papadopoulo, and R. Vaillant. Four applications of differential geometry to computer vision. In G. Orban and H.-H. Nagel, editors, *Artificial and Biological Vision Systems*, Basic Research Series, pages 93–141. Springer–Verlag, 1993.

[8] R. Durbin, C. Miall, and G. Mitchinson, editors. *The computing neuron*. Addison-Wesley, 1989.

[9] L. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. Proceedings of the American Mathematical Society, 1998.

[10] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[11] T. Gisiger, S. Dehaene, and J. P. Changeux. Computational models of association cortex. *Curr. Opin. Neurobiol.*, 10:250–259, 2000.

[12] G. Hermosillo. *Variational Methods for Multimodal Image Matching*. PhD thesis, INRIA, The document is accessible at ftp://ftp-sop.inria.fr /robotvis/html/Papers /hermosillo:02.ps.gz, 2002.

[13] G. Hermosillo, C. Chefd'hotel, and O. Faugeras. Variational methods for multimodal image matching. *ijcv*, 2002. to appear.

[14] D. Hubel. *L'oeil, le cerveau et la vision : les étapes cérébrales du traitement visuel*. L'univers des sciences. Pour la science, 1994.

[15] J. Jost. *Riemannian geometry and geometric analysis*. Springer Verlag, 2002.

[16] M. Kapadia, M. Ito, C. Gilbert, and G. Westheimer. Improvement in visual sensitivity by changes in local context: parallel studies in human observers and in v1 of alert monkeys. *Neuron*, 50:35–41, 1995.

[17] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, Sept. 2000.

[18] J. J. Koenderink. The brain as a geometry engine. *Psychol. Res.*, 52:122–127, 1990.

[19] P. Kornprobst, R. Peeters, T. Vieville, G. Malandain, S. Mierisova, S. Sunaert, O. Faugeras, and P. V. Hecke. Superresolution in MRI and its influence in statistical analysis. Technical report, INRIA, July 2002.

[20] A. Leonard. Vortex methods for flow simulations. *J. Comput. Phys*, 37:289–335, 1980.

[21] M.Carandini, D.J.Heeger, and J.A.Movshon. *Linearity and gain control in V1 simple cells*, chapter 13. New York:Plenum Press, Aug. 1998.

[22] H. Nagel and W. Enkelmann. An investigation of smoothness constraint for the estimation of displacement vector fiels from images sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[23] M. Otte and H. Nagel. Optical flow estimation: Advances and comparisons. In J.-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 51–70. Springer–Verlag, 1994.

[24] J. Petitot and Y. Tondut. Vers une neuro-géométrie. fibrations corticales, structures de contact et contours subjectifs modaux. *Mathématiques, Informatique et Sciences Humaines*, 145:5–101, 1999.

[25] P. A. Raviat. An analysis of particle methods. In F. Brezzi, editor, *Numerical Methods in Fluid Dynamics*, volume 1127 of *Lect. Notes in Math.*, pages 243–324. Springer Verlag, Berlin, 1985.

[26] L. Scharf. The SVD and reduced rank signal processing. *Signal Processing*, 25(2):113–133, 1991.

[27] L. Schwartz. *Théorie des distributions*. Hermann, 1957.

[28] E. P. Simoncelli and D. Heeger. A model of neuronal responses in visual area mt. *Vision Research*, 38:743–761, 1998.

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

[30] S. Thorpe, A. Delorme, and R. VanRullen. Spike based strategies for rapid processing. *Neural Networks*, 14:715–726, 2001.

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

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

[33] A. Verri, F. Girosi, and V. Torre. Differential techniques for optical flow. *Journal of the Optical Society of America A*, 7:912–922, 1990.

[34] A. Verri and T. Poggio. Against quantitative optical flow. In *Proceedings First International Conference on Computer Vision*, pages 171–180. IEEE Computer Society, 1987.

[35] T. Viéville and O. D. Faugeras. Robust and fast computation of unbiased intensity derivatives in images. In G. Sandini, editor, *Proceedings of the 2nd ECCV*, pages 203–211, Santa-Margherita, Italy, 1992. Springer–Verlag.

[36] T. Vieville, D. Lingrand, and F. Gaspard. Implementing a multi-model estimation method. *The International Journal of Computer Vision*, 44(1), 2001.

Acknowledgments: *Frédéric Alexandre* and *Olivier Faugeras* are gratefully acknowledged for some powerful ideas at the origin of this work.

A Maple macros used for symbolic derivations

The following macros have been used for the symbolic derivations proposed in this paper.

```
# Return the extremum of a quadratic criterion c with linear constraints ctr
leastsquare := proc(c, ctr :: set, vars :: set)
  local l, v, r:
  l := {'cat(_Z_,i)'$i=1..nops(ctr)}: v := vars union l:
  r := solve(
   convert(linalg[grad](
    c + sum('cat(_Z_,i)'*ctr[i],i=1..nops(ctr)), convert(v, list)), set),v):
  if r <> NULL then map((u, l) -> if not member(op(1,u), l) then u fi, r, l) fi
 end:

# Calculate the power of vector x with respect to multi-indices
pow := (x :: vector, d :: list(integer)) ->
```

```
 convert(map((i, x, d) -> x[i]^d[i], [$1..min(vectdim(x),nops(d))], x, d), `*`):

# Integrate an expression e over a canonic hyper-cube
intBox := proc(e, x :: vector)
 option remember: local r, i:
 r := e: for i to vectdim(x) do r := int(r, x[i]=-1/2..1/2) od
end:

# Return the set of multi-indexes d = [d_1 .. d_n] with r = sum(d_i, i=1..n)
indexes := proc(r :: integer, n :: integer)
  option remember:
  if r = 0 then
   {[0$i=1..n]}
  else
   map((d, n) -> op(
    map((i, n, d) ->
     map((j, i, d) ->
      if i = j then d[j] + 1 else d[j] fi,
     [$1..n], i, d),
    {$1..n}, n, d)),
   {op(procname(r-1, n))}, n)
  fi
end:

# Define the kronecker symbol
dirac := (a, b) -> if a = b then 1 else 0 fi:
```

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

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

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