Vector field regularization by generalized diffusion

Innocent Souopgui, François-Xavier Le Dimet, Arthur Vidard

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Vector field regularization by generalized diffusion

I. Souopgui — F.-X. Le Dimet — A. Vidard

N° 6844

February 2009

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6844--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Vector field regularization by generalized diffusion

I. Souopgui* , F.-X. Le Dimet* , A. Vidard*

Thème NUM — Systèmes numériques
Équipes-Projets MOISE

Rapport de recherche n° 6844 — February 2009 — 18 pages

Abstract: Regularization is a common procedure when dealing with inverse problems. Because of the ill-posedness of many inverse problems, one needs to add some constraints as regularization to the problem in order to get a satisfactory solution. A difficulty when using multiple constraints is to properly choose a weighting parameter for each constraint. We propose here a vector field regularization method that combines in a single constraint the two well-known regularization methods namely Tikhonov regularization and smoothing regularization. The particularity of this new method is that one have only one balance parameter to determine. We also suggest a robust implementation of the proposed method based on the equivalent generalized diffusion equation in some particular cases. This implementation is illustrated on a set of vector fields of fluid motion

Key-words: Regularization, generalized diffusion, vector field

* INRIA, Lab. Jean-Kuntzmann, BP 53, 38041 Grenoble Cedex 9 France. E-mail : Innocent.Souopgui@imag.fr, Francois-Xavier.Le-Dimet@imag.fr, Arthur.Vidard@imag.fr

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Régularisation de champs de vecteurs par diffusion généralisée

Résumé : La régularisation est un processus courant dans la résolution des problèmes inverses. Son usage est lié à la nature mal posée de la plupart des problèmes inverses. La régularisation consiste à ajouter des contraintes supplémentaires au problèmes à résoudre en vue d'obtenir une solution satisfaisante. La qualité et la quantité des contraintes ajoutées permet d'avoir de meilleurs résultats. Le choix des paramètres de pondération des différentes contraintes dans le cas ou il y en a plusieurs est un problème délicat. Nous proposons ici une méthode de régularisation des champs de vecteurs qui combine les deux principales méthodes de la litérature à savoir la régularisation de Tikhonov et la régularisation par lissage. La particularité de cette méthode est qu'il y a un unique paramètre de poids à déterminer. Nous suggérons aussi une implémentation robuste de cette méthode dans certains cas particuliers. Cette implémentation est basée sur l'équation de diffusion généralisée. Cette implémentation est illustrée par des résultats sur des champs de vitesse d'écoulement fluide.

Mots-clés : Régularisation, diffusion généralisé, champ de vecteurs

1 Introduction

Given a complete description of a physical system, prediction can be made by integration of the mathematical model; this is known as the forward (modelization or simulation) problem. The inverse problem consists in using given measurements of the system's state to infer the values of the parameters caracterising the model.

Inverse problems arise for example in geophysics, remote sensing and computer vision. In geophysics particulary for ocean and atmosphere, in order to predict the system state, one needs to reconstruct the initial state of the system from observation of some state variables; this is done using improved techniques known as data assimilation [7]. In remote sensing, the ability to retrieve meaningful information from degraded signal measurement by remote sensors is a crucial need. Information retrieval here include but is not limited to image restoration. In computer vision, one usualy needs to infer motion between successive frames of an image sequence.

In the deterministic case, the forward problem has a unique solution while the inverse one dosn't necessarily. The inverse problem has a unique solution only for idealistic situations that may not hold in practice [10]; this is due to the fact that inverse problems are generally underdetermined or overdetermined. Actual models have infinite degrees of freedom while in realistic experiments, the amount of data is usualy finite; in this case, the problem is said to be underdetermined. The underdetermined characteristic also true for discretized models where the amount of observed variables can be small in comparison with the number of variables of the model; for example, meteorological's system state are of the order of 10^8 variables while one have only 10^6 variables as observation. In some cases, there are more observations than parameters to infer, but these observations are redundant and sometime inconsistent leading to overdetermined system with no solution. The system is often partly underdetermined and partly overdetermined : this is known as ill-posedness which is a common characteristic of inverse problems. Another common characteristic of inverse problems is the ill-conditionning (small errors in the data causes large variations in the infered parameters)

Because of the ill-posedness and the ill-conditionning, one needs for solving an inverse problem to take into account any a priori knowledge of the properties of the solution : this is known as regularization. According to the properties imposed to the searched solution, one will speak of:

- regularization by closeness to a background (this case includes the minimal norm solution by considering the background as zero)
- regularization by smoothness of the solution (in a sense to be defined according to the problem)

We introduce here a mixed method, the regularization effect being a tradeoff between the closeness to background and the smoothness of the solution.

The equilibrum between both effects is insured by the a priori knowledge of the quality of the background. This method is based on the formulation of gradient vector flow [15] for snake [6] algorithms. Snakes or active contours algorithms are techniques developed in computer vision for object boundaries localization in image. The newly introduced method is tried out on vector field recovery from gaussian noisy field and on vector field coming from geophysical flow fields reconstruction.

The present document is organised as followed : Sections 2.1 and 2.2 briefly present the regularization by closeness to a background and the regularization by smoothness of the solution. Section 2.3 is a small review of the characteristics of the most used methods. The proposed method is described in sections 3.

2 Regularization for inverse problems

In order to simplify the comprehension, we use two different notations; matrix notation wich is well suited for the first case (regularization by closeness to background), and functional notation for the rest of the document.

2.1 Regularization by closeness to a background

Also known as Tikhonov regularization [13], regularization by closeness to a background is achieved by forcing the solution to be closed to an initial guess. As mentionned above and only for this section, we are going to present the problem in $\mathbb{R}^n$.

Let $A \in \mathbb{R}^{m\times n}, m > n \; and \; \mathbf{b} \in \mathbb{R}^m$, a standard way of solving the overdetermined system $\mathbf{Ax} = \mathbf{b}$ is the linear least squares method which considers the minimization problem

$$find \; \mathbf{x}^* = \text{ArgMin } \mathbf{J}(\mathbf{x}).$$

where $J(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|^2$ is the so called cost function and $\|.\|$ is the euclidian norm in $\mathbb{R}^k$. If A^TA is nonsingular, then the solution is given by

$$\mathbf{x}^* = (\mathbf{A^T A})^{-1}\mathbf{A^T b}.$$

Since A^TA may be ill-conditionned or singular yielding a large number of solutions, one gives preference to a particular solution by adding the Tikhonov regularization term (1) to the cost function.

$$\varepsilon(\mathbf{x}) = \|\Gamma(\mathbf{x} - \mathbf{x_0})\|^2 \tag{1}$$

where Γ is known as the Tikhonov matrix to be defined; this matrix may be choose to ensure the nonsingularity and the well-conditionning of the

problem to solve. The cost function becomes

$$J(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma(\mathbf{x} - \mathbf{x_0})\|^2$$

and the solution is given by

$$\mathbf{x}^* = \mathbf{x_0} + (\mathbf{A^T A} + \mathbf{\Gamma^T \Gamma})^{-1}\mathbf{A^T}(\mathbf{b} - \mathbf{A x_0}).$$

The Tikhonov regularization forces the computed solution to be Γ-closed to the initial guess $\mathbf{x_0}$.

2.2 Regularization by smoothing

Another form of a priori knowledge is the smoothness of the solution. In this case, the regularization term is based on the derivative of the argument of the cost function instead of the argument itself.

Notations Let Ω be an open subset of $\mathbb{R}^m$ ($\Omega \subset \mathbb{R}^m$),
the inverse problem : find

$$ArgMin(J(\mathbf{v})), \mathbf{v} \in (L^2(\Omega))^n$$

with $\mathbf{v}(\mathbf{x}) = (v_i(\mathbf{x}))_{1 \leq i \leq n}$ and $\mathbf{x} = (x_i)_{1 \leq j \leq m} \in \Omega$ can be transformed to the regularised version

$$\text{find } ArgMin(J(\mathbf{v}) + \varepsilon(\mathbf{v}))$$

with the regularization term ε

2.2.1 First order methods

The first order regularizarion methods define ε as a function of the first order derivatives of $\mathbf{v}$:

$$\varepsilon(\mathbf{v}) = \Phi\left(\frac{\partial v_i}{\partial x_j}\right)_{1 \leq i,j \leq n,} \tag{2}$$

where Φ is called *regularization function* to be defined according to the application; a survey of regularization functions can be found in [4]. The most used of first order regularizarion methods is the gradient penalization. It has been used by Horn and Schunck in the formulation of optical flow [5] for motion estimation. The regularization function of Horn and Schunck is defined as follow:

$$\varepsilon(\mathbf{v}) = \int_\Omega \sum_{i=1}^{n} \|\nabla v_i\|^2 d\mathbf{x} \tag{3}$$

2.2.2 Second order methods

The goal of regularization shall be to discourage characteristics not allowed by the physical characteristics of the system. For fluid flow, these characteristics include for example vorticity, divergence and shearing : they are quantified by the first order derivatives of the vector field. So first order regularization can be used when physical system don't allow such a characteristic. However, fluid flows are not usualy free of this characteristics. So one may need to discourage not directly these characteristics but their spacial variation. This can be achieved by second order regularization methods which are based on the second order derivatives of the function on which they are applied; The regularization term in this case can be expressed as :

$$\varepsilon(\mathbf{v}) = \Phi\left(\frac{\partial^2 v_i}{\partial x_j \partial x_k}\right)_{1\leq i,j,k\leq n,} \tag{4}$$

An example based on the first order derivatives of *div* and *curl* is the regularization of Suter [11] defined as followed :

$$\varepsilon(\mathbf{v}) = \int_\Omega \alpha\|\nabla div(\mathbf{v})\|^2 + \beta\|\nabla curl(\mathbf{v})\|^2 d\mathbf{x} \tag{5}$$

where *div* and *curl* are respectively the divergence and vorticity of the vector field and are defined as follow:

$$\begin{aligned} div(\mathbf{v}) &= \nabla.\mathbf{v} \\ curl(\mathbf{v}) &= \nabla \wedge \mathbf{v} \end{aligned}$$

Remark 2.1 *Higher order derivatives of v can also be used for regularizarion; for example (5) has been generalized by Chen and Suter [2] using m-order derivatives of div and curl.*

$$\varepsilon(\mathbf{v}) = \int_\Omega \alpha\|\nabla^m div(\mathbf{v})\|^2 + \beta\|\nabla^m curl(\mathbf{v})\|^2 d\mathbf{x} \tag{6}$$

2.3 Improvement keys for regularization techniques

Using nonobvious and robust regularization functions makes it possible to take into account a priori knowledges of the system. Many developments based on (3) and (4) were suggested by various authors in particular contexts. For motion estimation, the community of computer vision proposed special regularization functions in order to preserve discontinuities; see [1, 4] for details. In the field of fluid dynamics, special regularization fonctions were developed in order to take into account the characteristics of the fluid flows such as vorticity, the divergence and shearing: [9] suggests a regularization function based on the fluid dynamics structures; [3] introduces intermediaries scalar functions which are approximation of divergence and curl

of the vector field to be regularised. Regularization by closeness to data and smoothing can be combined while solving a particular problem. In all cases it is necessary to properly choose the weighting parameters for the various terms or characteristics taken into account. The more regularization terms and characteristics taken into account, the more the weighting parameters to be chosen consequently.

3 Vector flow for vector field regularization

3.1 Gradient Vector Flow for snake algorithm

Snake algorithms [6] are techniques developped in computer vision for object boundaries localization in images. Here, the snake (also known as active contour) is a curve defined within the image domain which can move under the influence of internal forces and external forces. We are interrested by the external forces : these are forces computed from the image data that are used to attract or to push the snake towards object boundaries. The literature defines two types of snake model : parametric snakes [6] and geometric snakes [12, 8]. In it's original version, external forces for parametric snakes was defined as the negative gradient of a potential function (the simple case being the image function itself). Object boundaries are supposed to be located on high values of the gradient of the potential function. With this definition of external forces, parametric snakes algorithms have two key difficulties : the initial contour must be as closed as possible to the true object boundaries or else it shall converge to a wrong result; the second problem is the difficulty to progress into boundary concavities. Many authors proposed differents methods to adress these problems without complete success until [15, 14] suggested to define external forces by Gradient Vector Flow (7); These formulation of external forces has solved both problems encountered by parametric snake algorithms. The Gradient Vector Flow force for snake algorithms ic defined as follow:
let $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ be the edges map[1] of an image function, the gradient vector flow field is defined as the minimal argument of the energy functional

$$\varepsilon(\mathbf{v}) = \iint_{\Omega} \mu \sum_{1 \leq i,j \leq n} \left(\frac{\partial v_i}{\partial x_j} \right)^2 + \|\nabla f\|^2 \|\mathbf{v} - \nabla f\|^2 dxdy \qquad (7)$$

This energy functional is a trade off between the regularization term of Horn and Schunck and the closeness to the gradient of the edge map function. it aims at extending the edge map gradient from edges regions to homogeneous regions. The minimization is achieved by setting v to be close to ∇f when $\|\nabla f\|$ is large and to be smooth when $\|\nabla f\|$ is small.

[1] The edge map of an image function is the norm of the gradient of this function

3.2 Vector flow for regularization

Definition 3.1 *Let $\Omega \subset \mathbb{R}^m$ be an open subset of $\mathbb{R}^m$ with smooth boundaries, $\mathbf{v} \in (L^2(\Omega))^n$ a noisy vector field, $\mathbf{v}(\mathbf{x}) = (v_i(x))_{1\leq i\leq n}$ with $x = (x_i)_{1\leq i\leq m} \in \mathbb{R}^m$ and $\langle . \rangle_k$ and $\|.\|_k$ respectively the scalar dot product and the euclidian norm in $\mathbb{R}^k$. From the definition of gradient vector flow field for snake algorithm, we define the vector flow regularized field $\mathbf{w}^* \in (L^2(\Omega))^n$ to be the minumum argument of the regularization function:*

$$\varepsilon(\mathbf{w}) = \int_\Omega \mu \sum_{i=1}^{n} \|\nabla w_i\|_m^2 + \|\mathbf{v}\|_n^2 \|\mathbf{w} - \mathbf{v}\|_n^2 dx \tag{8}$$

The function (8) involves 2 terms : a smoothing term $\sum_{i=1}^{n} \|\nabla w_i\|_m^2$ and a data term $\|\mathbf{v}\|_n^2 \|\mathbf{w} - \mathbf{v}\|_n^2$ in reference to $\mathbf{v}$ which is the given field to be regularized. The scalar parameter μ controls the trade-off between the two terms. The setting of this parameter should take into account the amount of noise in data : for more noise in the given $\mathbf{v}$, increase the parameter μ.

- When $\|\mathbf{v}\|$ is large, the energy functional is dominated by the data term and it is minimized by setting $\mathbf{w}^* = \mathbf{v}$.
- When $\|\mathbf{v}\|$ is small, the energy functional is dominated by the smoothing part and it is minimized by smoothing $\mathbf{w}^*$.

The smoothing term is known as the first order regularization which penalizes the spatial deviations of the vector field.

The problem of finding

$$\mathbf{w}^* = \text{ArgMin } \varepsilon(\mathbf{w}) \tag{9}$$

is a problem of unconstrained optimization. This problem has a unique solution if ε is strictly convex, lower semi continuous and if

$$\lim_{\|\mathbf{w}\| \to +\infty} \varepsilon(\mathbf{w}) \to +\infty$$

It is well known that if such a function is differentiable, a necessary condition for $\mathbf{w}^*$ to be a solution of (9) is given by the Euler-Lagrange condition :

$$\nabla\varepsilon(\mathbf{w}^*) = 0 \tag{10}$$

where $\nabla\varepsilon$ is the gradient of ε with respect to $\mathbf{w}$. The difficulty with nonlinear problems is to express $\nabla\varepsilon$. When this gradient is expressed, one can use descent type algorithms to solve the minimization problem. We are going to show how to express $\nabla\varepsilon$ in restricted cases, and a way to solve the minimization problem without gradient descent algorithm.

Proposition 3.1 *For* $\mathbf{w} \in \left(H^2(\Omega)\right)^n$ *with* $\langle \nabla w_i(\mathbf{x}), \nu \rangle = 0,\ \forall \mathbf{x} \in \partial\Omega,\ 1 \leq i \leq n$
The gradient of ε *is given by*

$$\nabla\varepsilon(\mathbf{w}) = 2\int_\Omega \left(-\mu\nabla^2\mathbf{w} + \|\mathbf{v}\|^2(\mathbf{w}-\mathbf{v})\right) dx \tag{11}$$

Proof Let

$$\begin{aligned} \varepsilon(\mathbf{w}) &= \int_\Omega \mu \sum_{i=1}^n \|\nabla w_i\|_m^2 + \|\mathbf{v}\|_n^2 \|\mathbf{w}-\mathbf{v}\|_n^2 dx & (12)\\ &= \int_\Omega \mu \sum_{i=1}^n \langle \nabla w_i, \nabla w_i \rangle_m + \|\mathbf{v}\|_n^2 \langle \mathbf{w}-\mathbf{v}, \mathbf{w}-\mathbf{v} \rangle_n dx & (13) \end{aligned}$$

where $\langle ., . \rangle_k$ is the dot product in $\mathbb{R}^k$.
For $\mathbf{w} \in (H^2(\Omega))^n$, let $\mathbf{h} \in (L^2(\Omega))^n$, $\hat{\varepsilon}$ the gateaux-derivative of ε in the direction $\mathbf{h}$, the gradient $\nabla\varepsilon$ can be found by showing the linear dependance of $\hat{\varepsilon}$ with respect to $\mathbf{h}$.

$$\hat{\varepsilon}(\mathbf{w}, \mathbf{h}) = \lim_{\alpha \to 0} \frac{\varepsilon(\mathbf{w}+\alpha\mathbf{h}) - \varepsilon(\mathbf{w})}{\alpha}$$

Let write $\varepsilon = \varepsilon_s + \varepsilon_d$ (s and d standing for smoothing and data) where

$$\begin{aligned} \varepsilon_s(\mathbf{w}) &= \mu \int_\Omega \sum_{i=1}^n \langle \nabla w_i, \nabla w_i \rangle_m dx \\ \varepsilon_d(\mathbf{w}) &= \int_\Omega \|\mathbf{v}\|_n^2 \langle \mathbf{w}-\mathbf{v}, \mathbf{w}-\mathbf{v} \rangle_n dx \end{aligned}$$

$$\begin{aligned} \frac{\varepsilon_s(\mathbf{w}+\alpha\mathbf{h}) - \varepsilon_s(\mathbf{w})}{\alpha} &= \frac{\mu}{\alpha} \sum_{i=1}^n \int_\Omega \langle \nabla(w_i + \alpha h_i), \nabla(w_i + \alpha h_i) \rangle dx \\ &\quad -\frac{\mu}{\alpha} \sum_{i=1}^n \int_\Omega \langle \nabla w_i, \nabla w_i \rangle dx \end{aligned}$$

after development and the consideration that $\alpha \to 0$ we get

$$\begin{aligned} \hat{\varepsilon}_s(\mathbf{w}, \mathbf{h}) &= 2\mu \sum_{i=1}^n \int_\Omega \langle \nabla w_i, \nabla h_i \rangle dx \\ &= 2\mu \sum_{i=1}^n \sum_{j=1}^m \int_\Omega \frac{\partial w_i}{\partial x_j} \frac{\partial h_i}{\partial x_j} dx \end{aligned}$$

Green's formula lead to

$$\begin{aligned}\hat{\varepsilon}_s(\mathbf{w},\mathbf{h}) &= -2\mu\sum_{i=1}^{n}\sum_{j=1}^{m}\int_\Omega \frac{\partial^2 w_i}{\partial x_i^2}h_i dx + 2\mu\sum_{i=1}^{n}\sum_{j=1}^{m}\int_{\partial\Omega}\frac{\partial w_i}{\partial x_j}h_i\nu_j d\sigma \\ &= -2\mu\int_\Omega \left\langle\nabla^2\mathbf{w},\mathbf{h}\right\rangle dx + 2\mu\sum_{i=1}^{n}\int_{\partial\Omega}\left\langle\nabla w_i,\nu\right\rangle h_i d\sigma\end{aligned}$$

where $\nabla^2\mathbf{w} = (\nabla^2 w_i)_{1\leq i\leq n}$, ∇^2 the laplacian operator, $\partial\Omega$ is the boundary of Ω and ν the outward unit surface normal to $\partial\Omega$.

$$\begin{aligned}\frac{\varepsilon_d(\mathbf{w}+\alpha\mathbf{h})-\varepsilon_d(\mathbf{w})}{\alpha} &= \frac{1}{\alpha}\int_\Omega \|\mathbf{v}\|^2\left(\left\langle\mathbf{w}+\alpha\mathbf{h}-\mathbf{v},\mathbf{w}+\alpha\mathbf{h}-\mathbf{v}\right\rangle\right) \\ &\quad + \left(\left\langle\mathbf{w}-\mathbf{v},\mathbf{w}-\mathbf{v}\right\rangle\right)dx\end{aligned}$$

after development and the consideration that $\alpha\to 0$ we have

$$\hat{\varepsilon}_d = 2\int_\Omega \|\mathbf{v}\|^2\left\langle\mathbf{w}-\mathbf{v},\mathbf{h}\right\rangle dx$$

Looking for $\mathbf{w}$ such that ∇w_i is orthogonal to ν on $\partial\Omega$,

$$\left\langle\nabla w_i^*(\mathbf{x}),\nu\right\rangle = 0,\ \forall\mathbf{x}\in\partial\Omega,\quad 1\leq i\leq n \tag{14}$$

we get

$$\begin{aligned}\hat{\varepsilon}(w,h) &= 2\int_\Omega\left\langle-\mu\nabla^2\mathbf{w},\mathbf{h}\right\rangle dx + \|\mathbf{v}\|^2\left\langle\mathbf{w}-\mathbf{v},\mathbf{h}\right\rangle dx \\ &= 2\int_\Omega\left\langle-\mu\nabla^2\mathbf{w}+\|\mathbf{v}\|^{\mathbf{2}}(\mathbf{w}-\mathbf{v}),\mathbf{h}\right\rangle dx \\ &= \left\langle\nabla\varepsilon,\mathbf{h}\right\rangle\end{aligned}$$

■

3.3 Numerical implementation

3.3.1 Classical implementation

A classical way of solving (8) is to use gradient descent optimization algorithms. Given a first guess w_0 and a way of computing $\nabla\varepsilon(\mathbf{w})$, the following algorithm converge to the minimum w^* of ε.

$$w_k = w_{k-1} + \rho_k D_k$$

where D_k is the direction of descent, and ρ_k the step size of descent.

$$\rho_k = ArgMin\ \varepsilon\left(w_{k-1}+\rho D_k\right)$$

The process is stopped when a predefined convergence criterion is reached. The algorithm converges to the solution assuming it exists and is unique meaning

- ε is strictly convex and lower semi continuous
- $\lim_{\|\mathbf{w}\|\to+\infty} \varepsilon(\mathbf{w}) \to +\infty$

If the solution is not unique, the convergence to a local minimum depends on the initial guest and the properties of ε in the vicinity of this minimum.

3.3.2 generalized diffusion implementation

The classical implementation requires a way to compute $\nabla\varepsilon(\mathbf{w})$, the goal of the generalized diffusion equations implementation is to find a solution in a restricted subspace in which $\nabla\varepsilon(\mathbf{w})$ is easily expressed. In this subspace, we solved an equivalent problem to the euler-lagrange condition (10). Given the gradient of $\varepsilon(\mathbf{w})$ as expressed by (11) numerical methods for finding local minima of ε can be developed by solving $\nabla\varepsilon(\mathbf{w}) = 0$ which is equivalent to

$$\mu\nabla^2 w_i - \|\mathbf{v}\|^2(w_i - v_i) = 0, \quad 1 \leq i \leq n \tag{15}$$

According to development in [15], (15) can be solved by considering w_i as a funtion of time and solving (16)

$$\frac{\partial}{\partial t} w_i(x,t) \quad = \quad \mu\nabla^2 w_i(x,t) - \|\mathbf{v}\|^2(w_i(x,t) - v_i(x))) \tag{16}$$

the set of equations (16) are known as the generalized diffusion equations. The boundary conditions (∇w_i orthogonal to ν on $\partial\Omega$) are managed in numerical algorithm by adding extra grid cells out of the actuel domain (Ω).

3.4 Experimental result

In this section, we present the result of some experimentation of the proposed method. Figure (1) shows the true vector field used for our experimentation. Figure (2) and (3) show the results of the proposed method on noisy vector field obtained by addition of Gaussian white noise to the original vector field of figure (1). Figures (4) and (5) present the result on a vector field coming from a 4D variational data assimilation process. Figure (6) shows the robustness of the algorithm on the boundaries; here we tested the method on a swirling vector field with a part of it out of the domain.

Error analysis we did error analysis by using normalized Mean Squared Error MSE defined in image processing as follow : given a gray-level image

(or a one chanel image) I of size $M \times N$ and a noisy version $\tilde{I}$ of this image

$$MSE(\tilde{I}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\tilde{I}(i,j) - I(i,j)\right]^2}{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)^2} \tag{17}$$

One can extend (17) to K-chanels images as follow

$$MSE(\tilde{I}) = \frac{\sum_{c=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\tilde{I}_c(i,j) - I_c(i,j)\right]^2}{\sum_{c=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{N} I_c(i,j)^2} \tag{18}$$

Since a vector field can be considered as a multi-chanel image, (18) can be used for MSE analysis of a vector field. For the 2D vector field in our experiments, we used (19) which is a restriction of (18) to a 2-chanels image or a discretized 2D vector field in a rectangular domain.

$$MSE(\tilde{u},\tilde{v}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[(\tilde{u}(i,j) - u(i,j))^2 + (\tilde{v}(i,j) - v(i,j))^2\right]}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[u(i,j)^2 + v(i,j)^2\right]} \tag{19}$$

The MSE analysis of the experiment's results is summarised in table (1).

	Gaussian white noise					
	Add. noise					mult. noise
noisy vector field	0.15	0.28	0.34	0.70	3.36	0.83
GD reconstruction	0.023	0.026	0.024	0.023	0.045	0.021

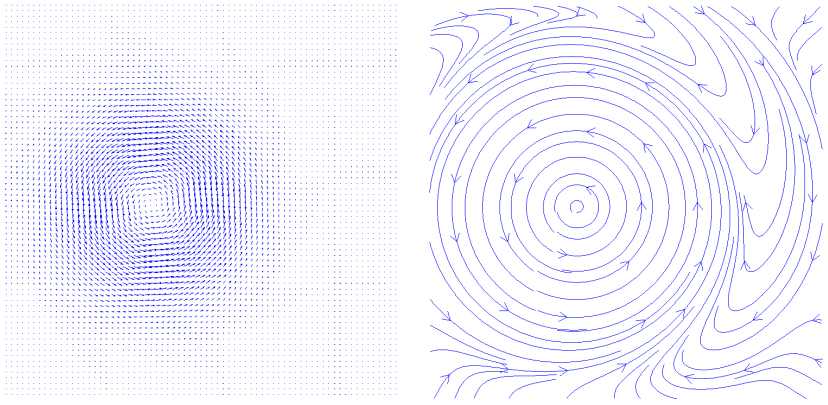
Table 1: MSE analysis

4 conlusion

We introduced a new vector field regularization method that combines the two well known methods of regularization :

- closeness to a background
- smoothness of the solution

This leads to a robust regularization method that can be extended to problems other than vector field regularization, for example images denoising. For Gaussian white noise, the first experiments shows that normalized MSE in the result remains less than 3% as long as the noise does not dominate the data. Experiment with vector field resulting from 4DVAR data assimilation process confirms the quality of the method. The first results show a



(a) (b)

Figure 1: True (experimental) vector field, isolated vortex; (a) vector field plotting, (b) streamlines plotting

promising method for problems of vector field reconstructions that can be embeded in data assimilation process for geophysical fluids and motion estimation algorithm. The smoothing term can be replace by an anisotropic or a high order derivative one in order to managed some particular situations.

References

[1] L. Alvarez, J. Esclarin, M. Lefbure, and J. Snchez. A pde model for computing the optical flow. In *XVI congreso de ecuaciones diferenciales y aplicaciones (C.E.D.Y.A. XVI)*, pages 1349–1356, Las Palmas de Gran Canaria, September 1999.

[2] Fang Chen and David Suter. Image coordinate transformation based on div-curl vector splines. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 1*, page 518, Washington, DC, USA, 1998. IEEE Computer Society.

[3] T. Corpetti, E. Mémin, and P. Pérez. Régularisation div-curl et Équation de continuité pour l'estimation du mouvement fluide. In *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, RFIA 2002*, volume 3, pages 887–898, Angers, France, January 2002.

[4] Thomas Corpetti. *Analyse d'écoulements fluides à partir de séquences d'images*. Hermes Science, 2004.

[5] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

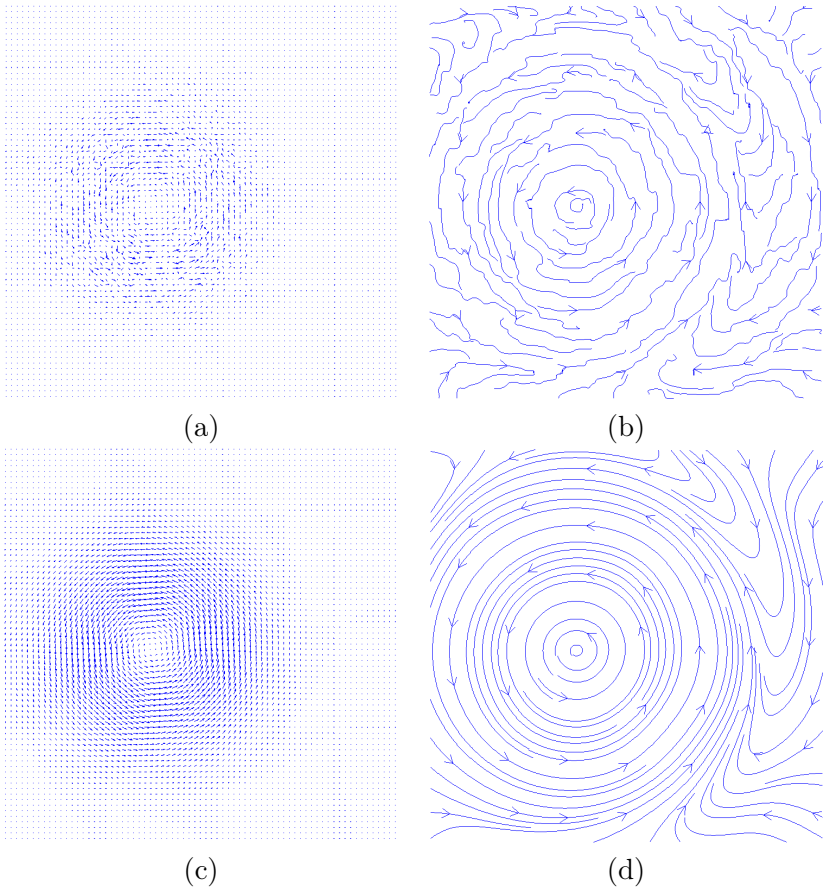


Figure 2: Vector field denoising using VF regularization, this is an example with low level multiplicative noise. The first row represents the noisy vector field and the associate streamlines; the second row represents the regularized version and and the associate streamlines; the true vector field is shown on figure (1)

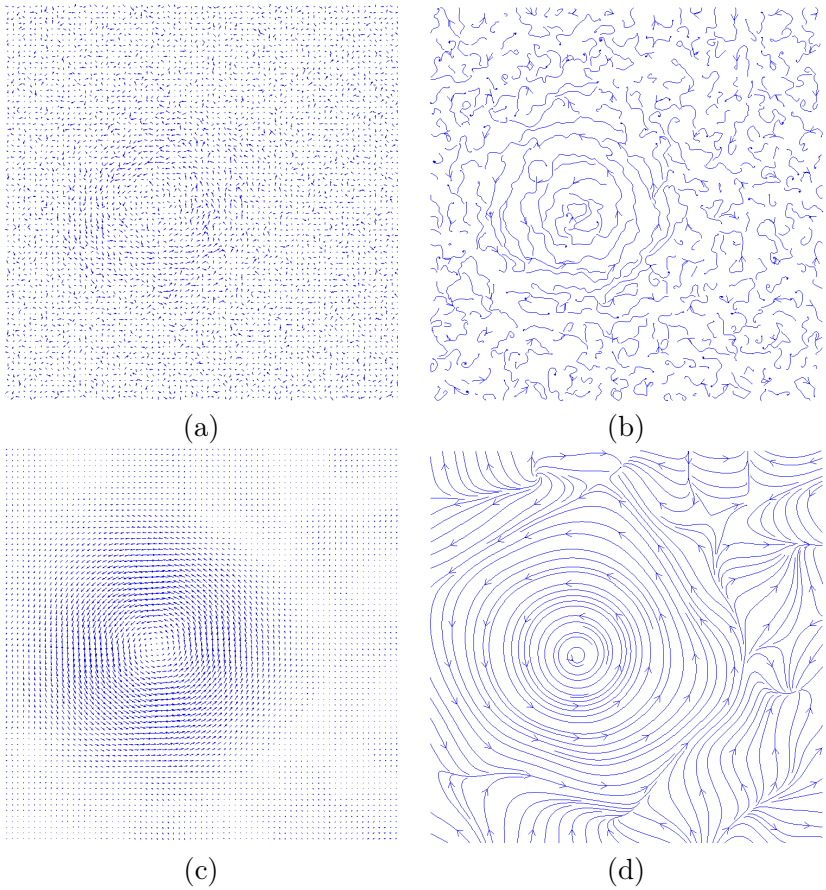


Figure 3: Vector field denoising using VF regularization, an example with hight level additive noise. The first row represents the noisy vector field and the associate streamlines; the second row represents the regularized version and and the associate streamlines; the true vector field is shown on figure (1)

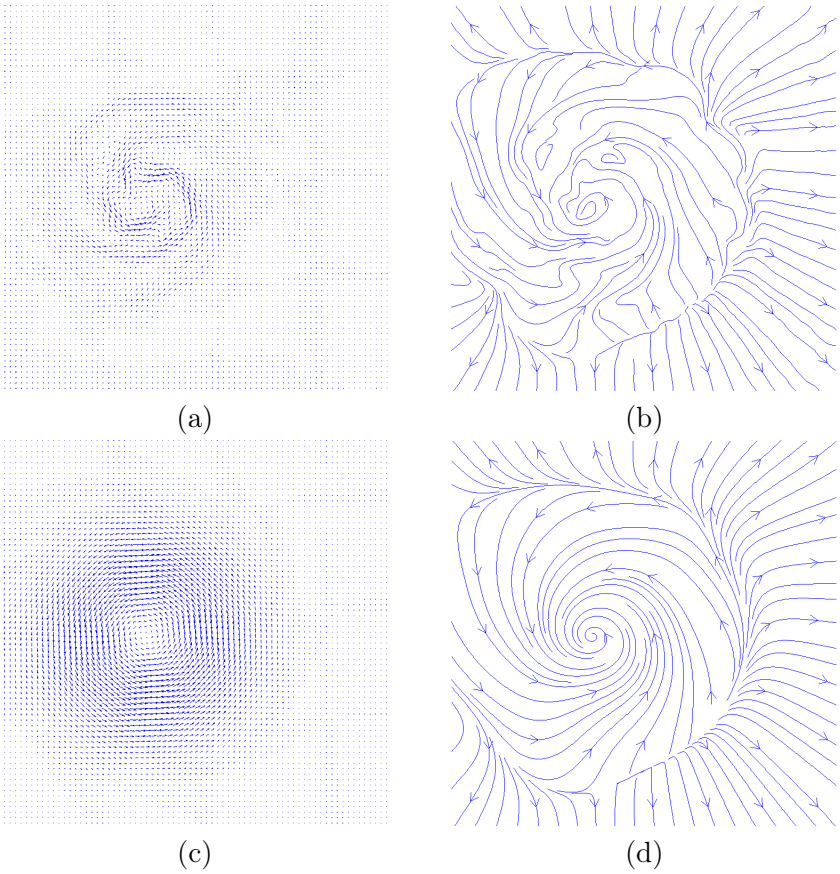


Figure 4: VF regularization of a vector field reconstructed by 4DVAR data assimilation; The first column corresponds to vector field plotting and the second column corresponds to the streamlines plotting. The first row is 4DVAR reconstruction without VF regularization and the second row the regularized version. The true vector field is the one on figure (1)

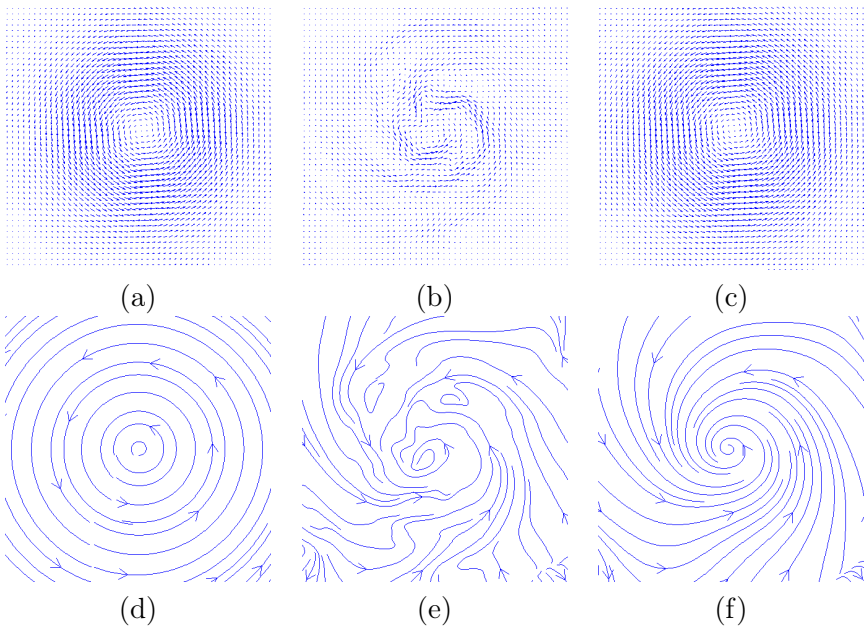


Figure 5: ZOOM on the vector field of figure (4); The first row represents vector field plotting; The second row represents streamlines plotting; Column 1 is the true field, column 2 the 4DVAR reconstructed field and column 3 the regularization of the 4DVAR reconstruction

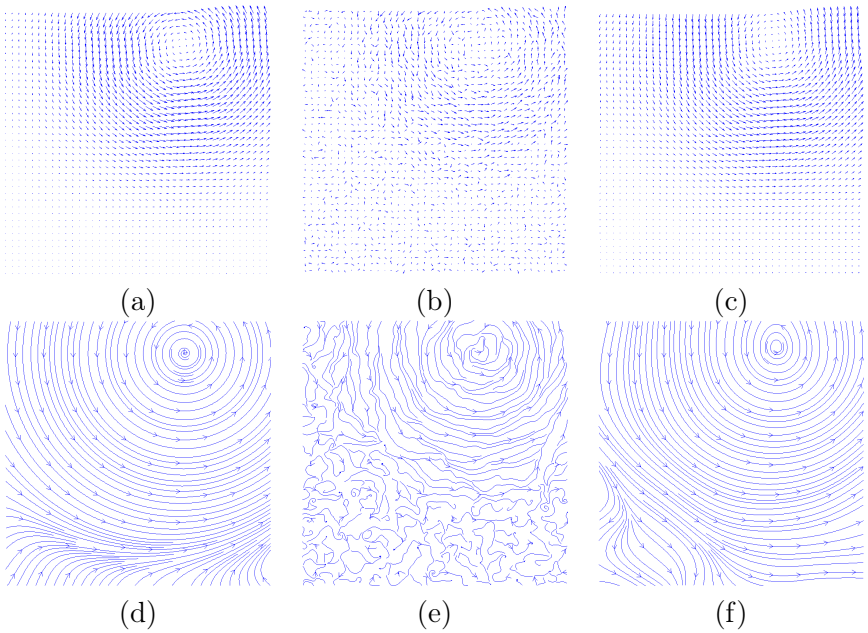


Figure 6: Vector field with intense activity on the boundaries. The first row represents vector field plotting; The second row represents streamlines plotting; Column 1 is the true field, column 2 the noisy vector field(gaussian noise) and column 3 the regularized version

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[7] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. *Tellus*, 38A:97 – 110, 1986.

[8] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95), Cambridge, MA, USA*, pages 840–845, June 1995.

[9] Etienne Mémin and Patrick Pérez. Fluid motion recovery by coupling dense and parametric vector fields. In *ICCV (1)*, pages 620–625, 1999.

[10] Roel Snieder and Jeannot Trampert. *Inverse Problems in Geophysics*, chapter Wavefield inversion, pages 119–190. Springer Verlag, New York, 2000.

[11] D. Suter. Motion estimation and vector splines. In *CVPR94*, pages 939–942, 1994.

[12] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4:306–331, 1988.

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

[14] C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing — An International Journal*, 71(2):131–139, 1998.

[15] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *TIP*, 7(3):359–369, 1998.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

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