Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat, Isabelle Herlin

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat — Isabelle Herlin

N° 6879

Mars 2009

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6879--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
PARIS - ROCQUENCOURT

Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat* , Isabelle Herlin†

Thème NUM — Systèmes numériques
Équipes-Projets Clime

Rapport de recherche n° 6879 — Mars 2009 — 32 pages

Abstract:

Data Assimilation is a mathematical framework used in environmental sciences to improve forecasts performed by meteorological, oceanographic or air quality simulation models. Data Assimilation techniques require the resolution of a system with three components: one describing the temporal evolution of a state vector, one coupling the observations to this state vector, and one defining the initial condition. In this report we use this framework to study a class of ill-posed Image Processing problems, usually solved by spatial and temporal regularization techniques. A generic approach is proposed to convert an ill-posed Image Processing problem in terms of a Data Assimilation system. This method is illustrated on the determination of optical flow from an image sequence. The main advantage of the resulting software is the use of a quality criteria on observations for weighting their contribution in the estimation process and of a dynamic model to ensure a relevant temporal regularity of the result.

Key-words: data assimilation, ill-posed problems, regularization, image assimilation, optical flow, motion.

* Université Pierre et Marie Curie, LIP6, 4 place Jussieu, 75005 Paris, France
† INRIA, CEREA, Joint Laboratory ENPC-EDF R&D, Université Paris-Est

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Résolution des problèmes mal posés du traitement d'image par Assimilation de Données

Résumé : L'assimilation de données est un cadre mathématique utilisé en science de l'environnement pour améliorer les prévisions météorologiques, océanographique ou en qualité de l'air. Cette technique nécessite la résolution d'un système d'équations : une première équation décrit l'évolution temporelle d'un vecteur d'état, une seconde décrit le couplage entre l'observation et le vecteur d'état et une dernière définie la condition initiale du vecteur d'état. Dans ce rapport, l'assimilation de donnée est utilisée pour résoudre une classe de problèmes mal posés en traitement de l'image, habituellement résolus par des techniques de régularisation spatio-temporelle. Une approche générique est décrite pour résoudre le problème mal posé dans le cadre de l'assimilation de données. Cette méthode est illustrée sur le problème de l'estimation du flot optique. Un avantage de cette méthode est de pouvoir utiliser un critère de qualité des observations pour pondérer la contribution des observations dans le calcul de la solution et l'équation d'évolution permet alors d'obtenir une régularité temporelle pertinente pour la solution.

Mots-clés : assimilation de données, problèmes mal posés, régularisation, assimilation d'images, flot optique, mouvement.

1 Introduction

In the research field of Image Processing, most problems are ill-posed in the sense that it is not possible to provide a unique solution (Hadamard, 1923). A first cause of ill-posedness is that the equations used to model image properties are under-determined. An example is given by the famous "aperture problem" occurring in the estimation of optical flow: a further constraint is required to compute a unique field of velocity vectors. As an image processing problem is usually modelled by a system of equations to be solved, the so-called *Image Model*, this type of ill-posedness means that the Image Model is not invertible. A second cause of ill-posedness occurs when the computation of image features can be obtained by different algorithms. For example, determining image gradient requires to approximate a differential operator by a discrete one among several possible finite difference formulations; each one providing a different result.

A common strategy to solve ill-posed problems is to provide the Image Model with additional information. Two options may be considered. 1) Providing explicit information: additional images are used to enlarge the set of input data. However, this is generally not possible because other acquisitions having the requested properties are not available. 2) Providing implicit information: either hypotheses on image properties or constraints on the solution can be used. A usual constraint is to restrict the dimension of the space of admissible solutions. For instance, the result may be searched among the functions with bounded spatial variations: this is called a "Tikhonov regularization method" in the literature (Tikhonov, 1963). In the general case, these additional image properties or constraints are expressed as equations, which, combined to the Image Model, lead to a new invertible Image Model.

Assuming an image sequence is available, enhancement may be obtained by taking into account the temporal information. Let us illustrate this on image segmentation, which is a pure spatial problem. A spatial regularization method, such as Shah-Mumford's functional (Mumford and Shah, 1989), produces a segmentation which is a compromise between a smooth solution and the fidelity to the input data. The segmentation process can be performed directly on the whole sequence and not on a frame-by-frame basis: following (Weickert and Schnörr, 2001), the solution is seek as a function depending on the spatial and temporal coordinates. This approach has however several drawbacks. First, it imposes an arbitrary temporal regularity which can not deal with complex dynamics. Second, missing data are taken into account in the process and introduce errors in the final solution. By "missing data", we refer to pixels' values displaying a wrong information due to either a failure of the acquisition system or noise. Third, as the solution is looked for in the spatio-temporal domain, this leads to an high computing complexity compared to a pure spatial model, which is a limiting factor for operational applications.

An alternative to this spatio-temporal approach, particularly interesting in case of applications with a high probability of missing data, is to consider the temporal evolution of images over the sequence. The challenge becomes to write an effective and pertinent dynamics model and to include it in the solution computation. Such information can be inferred, for instance, from *a priori* knowledge on images and on the observed phenomena. Moreover, data quality has also to be evaluated in order to ignore missing data in the process.

In this paper, we propose to use the *Data Assimilation* framework as a generic tool to solve ill-posed Image Processing problems. A Data Assimilation method solves a system of three components with respect to a state vector:

- an evolution equation describes the evolution of the state vector over time, using an operator called the "evolution model";
- an "observation equation" models the link between the state vector and the observations provided by the image sequence;
- the initial condition of the state vector.

Each component is inaccurate and a description of the error is stated in terms of a Gaussian noise characterized by a covariance matrix. This matrix depicts the dependencies existing between the components of the state vector on the one hand, and between two different locations in the space-time domain on the other hand.

As it will be proven in the paper, Data Assimilation is a suitable approach to solve the three drawbacks of Weickert's method because: the evolution model describes the images' temporal dynamics; the covariance associated to the observation equation error overcomes the problem of missing data by weighting, for each pixel, the importance of the observation equation in the computation of the solution; Data Assimilation is a frame-by-frame process which allows large image sequences to be processed.

This article is organized as follow. Section 2 introduces the concept and difficulties of ill-posed problems in the Image Processing research field. We give some typical examples and present a short start-of-the-art of Tikhonov regularization methods. Section 3 describes the variational Data Assimilation method known as the 4D-Var algorithm. How Data Assimilation can be used to solve ill-posed problems by assimilating image data within an appropriate evolution model is explained in Section 4. Section 5 is then a direct application and describes how to compute optical flow in this framework. Section 5 also presents and discusses experimental results. We conclude in Section 6 and give some scientific perspectives to this research study.

2 Ill-posed problems in Image Processing

Hadamard gives the following definition: a problem is well-posed if 1) it has an unique solution, 2) the solution depends continuously on the input (Hadamard, 1923). A problem which does not meet these conditions is called ill-posed. Using this definition, optical flow estimation, image registration, curves or surfaces matching, tracking of multiple objects, segmentation, restoration, deconvolution, denoising and shape from shading are ill-posed problems because the equations used for modeling these problems are under-constrained (the aperture problem of optical flow for instance).

Links between image properties and the solution are modeled as a set of equations constituting the Image Model. The image, input data, is denoted $\mathbf{Y}$ and depends on the spatial coordinate $\mathbf{x}$ in a bounded domain denoted Ω. The solution of the problem, denoted $\mathbf{X}$, is not necessarily an image: it can be a velocity field, a curve, *etc.* To be general, the Image Model is described by:

$$\mathbb{I}(\mathbf{X}, \mathbf{Y})(\mathbf{x}) = \vec{0} \quad \forall \mathbf{x} \in \Omega \tag{1}$$

with $\mathbb{I}$ a differentiable image operator that may be:

- linear: $\mathbb{I}(\mathbf{X}, \mathbf{Y}) = \mathbf{Y} - A(\mathbf{X})$ with A linear. This is a typically the case in segmentation, restoration, denoising, deconvolution: $\mathbb{I}$ measures the discrepancy between the input image and the solution filtered by the operator A.
- non linear. A common situation in image processing is the following: $\mathbb{I}(\mathbf{X}, \mathbf{Y}) = B(\mathbf{Y}(C(\mathbf{X}))$. The input data $\mathbf{Y}$ is only evaluated on the pixels $C(\mathbf{X})$ depending on the solution. For example, $\mathbb{I}(\mathbf{X}, \mathbf{Y})(\mathbf{x}) = -\|\nabla \mathbf{Y}(\mathbf{X}(\mathbf{x}))\|$ for the well known active contours or $\mathbb{I}(\mathbf{X}, \mathbf{Y})(\mathbf{x}) = Y_2(\mathbf{x} + \mathbf{X}(\mathbf{x})) - Y_1(\mathbf{x})$ for estimating optical flow between two frames Y_1 and Y_2.

In this section, attention is focused on variational methods: instead of directly solving (1), an optimization problem is formulated and the solution is obtained by minimizing $E(\mathbf{X}) = \int_\Omega \Psi(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) d\mathbf{x}$ with Ψ a convex function such as $\Psi(0) = 0$. If Equation (1) is under-constrained, a possible method to obtain a unique solution is to use a Tikhonov regularization. This is performed by adding a second term to the functional E which becomes:

$$E(\mathbf{X}) = \int_\Omega \left(\Psi(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) + \sum_{n \geq 0} \alpha_n \Psi \left(\left\| \frac{\partial^n \mathbf{X}}{\partial \mathbf{x}^n} \right\| \right) \right) d\mathbf{x} \tag{2}$$

An usual choice is to set $\alpha_i = 0$ for $i \neq 1$ ensuring a first order regularization or $\alpha_i = 0$ for $i \neq 1, 2$ ensuring a second order regularization. The norm used in the regularizing term is often the Euclidean norm but other choices are possible such as $\|\mathbf{X}\|^2 = \mathbf{X}^T A \mathbf{X}$, A being a symmetric and definite positive matrix. It is even possible to choose A depending on the input image $\mathbf{Y}$ in order to perform a regularization driven by $\mathbf{Y}$. For instance, to determine optical flow, Nagel uses in (Nagel, 1983) a matrix A depending on the local image configuration: the algorithm performs a weak regularization on edges and strong otherwise.

The minimization of E is led in the calculus of variation framework: the solution is searched as the zero of the Euler-Lagrange equation associated to (2), $\frac{\partial E}{\partial \mathbf{X}} = 0$, with $\frac{\partial E}{\partial \mathbf{X}}$ denoting the differential of E with respect to $\mathbf{X}$. For a first order regularization, the general expression of this Euler-Lagrange equation is:

$$-\nabla . \left(\Psi'(\|\nabla \mathbf{X}\|) \frac{\nabla \mathbf{X}}{\|\nabla \mathbf{X}\|} \right) + \frac{1}{\alpha_1} \nabla . \left(\Psi'(\|\mathbb{I}(\mathbf{X}, \mathbf{Y})\|) \frac{\partial \mathbb{I}}{\partial \mathbf{X}} \right) = 0$$

It is discretized by finite differences and the solution is obtained using a Jacobi or Gauss-Seidel method in the linear case and a method of steepest descent or conjugate gradient otherwise.

From the beginning of this section, the image is considered as only depending on the spatial coordinate and the Image Model is purely spatial. In the case of optical flow computation, at least two frames are required, but the sequence is still processed on a frame by frame basis. As a consequence, the result obtained on one image has no link with those obtained on adjacent ones. An improved solution, when dealing with a temporal sequence, has been proposed by Weickert (Weickert and Schnörr, 2001) and consists in minimizing the functional:

$$E(\mathbf{X}) = \int_\Omega \int_0^\mathbf{T} \left(\Psi(\|\mathbb{M}(\mathbf{X},\mathbf{Y})\|) + \sum_{n\geq 0} \alpha_n \Psi \left(\left\| \frac{\partial^n \mathbf{X}}{\partial \mathbf{x}^n} \right\|, \left\| \frac{\partial \mathbf{X}}{\partial t} \right\| \right) \right) d\mathbf{x}dt \quad (3)$$

with $\mathbf{X}$ and $\mathbf{Y}$ becoming time dependent and the regularization term computed in the space-time domain. Such a functional describes accurately regular dynamics but becomes irrelevant for non linear ones. Moreover, missing data are taken into account: the method can only perform a smoothing of aberrant values using the spatio-temporal neighborhood. Last, the time is viewed as an additional dimension and a direct consequence is the proportional increase of the problem's size. A huge memory and computer time is then requested for processing a large sequence.

A partial answer to the problem of dealing with complex temporal dynamics and missing data is to add a model of the dynamic evolution. This implies to be able to solve simultaneously the Image Model and the evolution model and this can be achieved using the Data Assimilation framework described in the following Section.

3 The Data Assimilation framework

3.1 Modelisation of the problem

The Data Assimilation framework aims to solve the system (4,5,6) with respect to a state vector $\mathbf{X}(\mathbf{x},t)$, depending on the spatial coordinate $\mathbf{x}$ and time t:

$$\begin{aligned} \frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}(\mathbf{X})(\mathbf{x},t) &= \mathcal{E}_m(\mathbf{x},t) & (4)\\ \mathbf{X}(\mathbf{x},0) &= \mathbf{X}_b(\mathbf{x}) + \mathcal{E}_b(\mathbf{x}) & (5)\\ \mathbf{Y}(\mathbf{x},t) &= \mathbb{H}(\mathbf{X})(\mathbf{x},t) + \mathcal{E}_O(\mathbf{x},t) & (6) \end{aligned}$$

Equation (4) describes the temporal evolution of $\mathbf{X}$. $\mathbb{M}$, called *evolution model*, is supposed differentiable. As $\mathbb{M}$ may describe approximately the evolution of the state vector, a *model error* $\mathcal{E}_m$ is introduced to quantify the imperfections. Equation (5) establishes the initial condition of the state vector and the error is expressed by the *background error* $\mathcal{E}_b$. Equation (6), called *observation equation*, describes the links between the observation vector, $\mathbf{Y}(\mathbf{x},t)$, and the state vector. The observation may sometimes be a part of the state vector, in which case $\mathbb{H}$ is a projection operator. In other cases, the observation may be an indirect function of the state vector. Equation (6) is the standard form of the observation equation used in the Data Assimilation literature. However, this formulation is quite restrictive to describe the links, possibly complex, existing between the observation and the state vector. For instance, non-linear Image Models can not be expressed in this form. To be more general, the following will be used in this article:

$$\mathbb{H}(\mathbf{Y},\mathbf{X})(\mathbf{x},t) = \mathcal{E}_O(\mathbf{x},t) \quad (7)$$

which includes the previous formulation (6). The *observation error* $\mathcal{E}_O$ simultaneously represents the imperfection of $\mathbb{H}$ and the measurement errors. $\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be Gaussian and fully characterized by their covariance matrices Q, B and R.

For instance, let $\mathbf{X}$ denote a Gaussian stochastic vector depending on a space-time coordinate $(\mathbf{x}, t)$; $\mathbf{X} = \mathbf{X}(x, t)$ and $\mathbf{X}' = \mathbf{X}(x', t')$ are the values on two given locations. The covariance matrix Σ, computed for $\mathbf{X}$ and $\mathbf{X}'$, measures their dependency and is defined by:

$$\Sigma(\mathbf{x}, t, \mathbf{x}', t') = \iint (\mathbf{X} - \mathbb{E}\mathbf{X})^T (\mathbf{X}' - \mathbb{E}\mathbf{X}') dP_{\mathbf{X},\mathbf{X}'}$$

$P_{\mathbf{X},\mathbf{X}'}$ is the joint distribution of $(\mathbf{X}, \mathbf{X}')$ and $\mathbb{E}$ denotes the expectation.

3.2 Variational formulation

In order to solve equations (4), (5) and (7), the following functional, to be minimized, is defined:

$$\begin{aligned} E(\mathbf{X}) = &\int_A \int_A \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right)^T (\mathbf{x}, t) Q^{-1}(\mathbf{x}, t, \mathbf{x}', t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right) (\mathbf{x}', t') d\mathbf{x} dt d\mathbf{x}' dt' \\ &+ \int_A \int_A \mathbb{H}(\mathbf{X}, \mathbf{Y})^T (\mathbf{x}, t) R^{-1}(\mathbf{x}, t, \mathbf{x}', t') \mathbb{H}(\mathbf{X}, \mathbf{Y})(\mathbf{x}', t') d\mathbf{x} dt d\mathbf{x}' dt' \\ &+ \int_\Omega \int_\Omega \big(\mathbf{X}(\mathbf{x}, 0) - \mathbf{X}_b(\mathbf{x})\big)^T B^{-1}(\mathbf{x}, \mathbf{x}') \big(\mathbf{X}(\mathbf{x}', 0) - \mathbf{X}_b(\mathbf{x}')\big) d\mathbf{x} d\mathbf{x}' \end{aligned} \tag{8}$$

with $A = \Omega \times [0, \mathbf{T}]$, Ω being the spatial domain and $[0, \mathbf{T}]$ the temporal domain. If $\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be independent, the functional E represents the log-likelihood of $\mathbf{X}$. The minimization is carried out by solving the associated Euler-Lagrange equation. The differential $\dfrac{\partial E}{\partial \mathbf{X}}$ is obtained by computing the derivative of E with respect to $\mathbf{X}$ in direction η:

$$\frac{\partial E}{\partial \mathbf{X}}(\eta) = \lim_{\gamma \to 0} \frac{d}{d\gamma}(E(\mathbf{X} + \gamma\eta)) \tag{9}$$

and by introducing an auxiliary variable λ, called the *adjoint variable* in the literature of Data Assimilation:

$$\lambda(\mathbf{x}, t) = \int_A Q^{-1}(\mathbf{x}, t, \mathbf{x}', t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right) (\mathbf{x}', t') d\mathbf{x}' dt' \tag{10}$$

We detail in the appendix A the determination of the Euler-Lagrange equation associated to (8). This leads to the following system:

$$\lambda(\mathbf{x}, \mathbf{T}) = 0 \tag{11}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* (\mathbf{x}, t) R^{-1} \mathbb{H}(\mathbf{X}, \mathbf{Y})(\mathbf{x}', t') d\mathbf{x}' dt' \tag{12}$$

$$\mathbf{X}(\mathbf{x}, 0) = \int_\Omega B\lambda(\mathbf{x}', 0) d\mathbf{x}' + \mathbf{X}_b(\mathbf{x}) \tag{13}$$

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t') d\mathbf{x}' dt' \tag{14}$$

Equation (12) makes use of two *adjoint operators* denoted $\left(\dfrac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ and $\left(\dfrac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$. These operators are formally defined as the differential operators in the dual

space of $\mathbf{X}$. For a given operator $\mathbb{K}$ applied on $\mathbf{X}$, we have:

$$\int \left(\frac{\partial \mathbb{K}}{\partial \mathbf{X}}(\eta)\right)^T \lambda d\mathbf{x}dt = \int \eta^T \left(\frac{\partial \mathbb{K}}{\partial \mathbf{X}}\right)^* (\lambda) d\mathbf{x}dt \tag{15}$$

for all integrable functions η and λ. If $\dfrac{\partial \mathbb{K}}{\partial \mathbf{X}}$ is a differential operator, the adjoint operator represents a compact notation for integration by parts. Riesz's theorem ensures the existence and uniqueness of the adjoint operator. For clarifying the discussion, let us determine the adjoint operator of $\mathbb{K} = \dfrac{\partial}{\partial x}$ in an interval $[a, b]$:

$$\begin{aligned}
\int_a^b \frac{\partial f}{\partial x}(x) g(t) dx &= [f(x)g(x)]_{x=a}^{x=b} - \int_a^b f(x) \frac{\partial g}{\partial x}(x) dx \\
&= \int_a^b f(x) \left(\left(\delta(x-b) - \delta(x-a)\right) g(x) - \frac{\partial g}{\partial x}(x) \right) dx \\
&= \int_a^b f(x) \left(\frac{\partial g}{\partial x}\right)^* (x) dx
\end{aligned}$$

The adjoint operator of $\dfrac{\partial}{\partial x}$ in $[a, b]$ is then $\left(\dfrac{\partial}{\partial x}\right)^* = \delta(x-b) - \delta(x-a) - \dfrac{\partial}{\partial x}$.

3.3 Incremental algorithm

At this stage, a difficulty occurs: the state vector is determined by equations (13,14) using the adjoint variable and the adjoint variable is determined by equations (11,12) using the state vector. To break this deadlock, an incremental method is defined. The underlying idea comes from the following lemma:

$$\min_{w \in \mathcal{V}(w_0)} E(w) = \min_{\delta w \in \mathcal{V}(0)} E(w_0 + \delta w)$$

where w_0 denotes a local minimum of E and $\mathcal{V}(w_0)$ denotes one neighborhood of w_0. The state vector is therefore written as $\mathbf{X}_b + \delta\mathbf{X}$ where $\mathbf{X}_b$ is a mean term, called the *background variable* in the Data Assimilation literature, and $\delta\mathbf{X}$ is the incremental variable. $\mathbf{X}$ is replaced by $\mathbf{X}_b + \delta\mathbf{X}$ in equations (12), (13) and (14). If $\mathbb{M}$ and $\mathbb{H}$ are linear operators, we have:

$$\mathbb{M}(\mathbf{X}) = \mathbb{M}(\mathbf{X}_b) + \left.\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right|_{\mathbf{X}_b} (\delta\mathbf{X}) \tag{16}$$

$$\mathbb{H}(\mathbf{X}) = \mathbb{H}(\mathbf{X}_b) + \left.\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right|_{\mathbf{X}_b} (\delta\mathbf{X}) \tag{17}$$

In case of non linearity, they could however be approximated by a first order Taylor development resulting in (16,17). In both cases, this leads to the following

new system of equations:

$$\lambda(\mathbf{x},\mathbf{T}) = 0 \tag{18}$$

$$-\frac{\partial\lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}\right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}\right)^* R^{-1}\left(\mathbb{H}(\mathbf{X}_b,\mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X})\right) d\mathbf{x}'dt' \tag{19}$$

$$\mathbf{X}_b(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x}) \tag{20}$$

$$\frac{\partial \mathbf{X}_b}{\partial t} + \mathbb{M}(\mathbf{X}_b) = 0 \tag{21}$$

$$\delta\mathbf{X}(\mathbf{x},0) = \int_\Omega B\lambda(\mathbf{x}',0)d\mathbf{x}' \tag{22}$$

$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X}) = \int_A Q\lambda(\mathbf{x}',t')d\mathbf{x}'dt' \tag{23}$$

The background variable $\mathbf{X}_b$ is first calculated from equations (20) and (21) independently of the adjoint variable and the incremental variable (direct run). The adjoint variable λ is then obtained from the background variable using equations (18) and (19) and the incremental variable $\delta\mathbf{X}$ is obtained from the adjoint variable using equations (22) and (23). If $\mathbb{M}$ and $\mathbb{H}$ are not linear, equations (18,19,22,23) produce an approximated solution $\mathbf{X}_b + \delta\mathbf{X}$ and an iterative algorithm is used to obtain an exact solution. The iterative algorithm is summarized in the following:

1. Initialization $(i = 0)$:

   (a) Compute the background variable $\mathbf{X}_b^0$ (providing the value of state vector at the first iteration):

$$\mathbf{X}_b^0(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x})$$
$$\frac{\partial \mathbf{X}_b^0}{\partial t} + \mathbb{M}(\mathbf{X}_b^0) = \vec{0}, \forall t \in [0,\mathbf{T}]$$

   (b) Initialize the incremental variable:

$$\delta\mathbf{X}(\mathbf{x},t) = \vec{0}, \forall t \in [0,\mathbf{T}]$$

2. Iteration $i > 0$:

   (a) Compute the adjoint variable λ:

$$\begin{aligned} \lambda(\mathbf{x},T) &= \vec{0} \\ -\frac{\partial\lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b^i}\right)^*(\lambda) &= -\int \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b^i,\mathbf{Y}}\right)^* R^{-1}\left[\mathbb{H}(\mathbf{X}_b^i,\mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b^i,\mathbf{Y}}(\delta\mathbf{X})\right] d\mathbf{x}'dt' \end{aligned}$$

   (b) Compute the incremental variable $\delta\mathbf{X}$:

$$\delta\mathbf{X}(\mathbf{x},0) = \int B(\mathbf{x},\mathbf{x}')\lambda(\mathbf{x}',0)d\mathbf{x}'$$
$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b^i}(\delta\mathbf{X}) = \int Q(\mathbf{x},t,\mathbf{x}',t')\lambda(\mathbf{x}',t')d\mathbf{x}'dt'$$

(c) Update the background value (providing the value of the state vector at iteration $i+1$):

$$\mathbf{X}_b^{i+1}(\mathbf{x},t) \Leftarrow \mathbf{X}_b^i(\mathbf{x},t) + \delta\mathbf{X}(\mathbf{x},t)$$

and back to step 2 until convergence.

4 Assimilation of images

This section explains how to solve the ill-posed Image Processing problems described in Section 2 using the framework of Data Assimilation and constitutes the core of this research and the main contribution of the paper. Using Data Assimilation to solve Image Processing Problem is an emerging domain. Studies have been done about curve tracking (Papadakis and Mémin, 2007) and determination of optical flow (Papadakis et al., 2007a,b). In *et al* (Huot et al., 2008), a method is proposed to estimate the ocean surface circulation from SST images: assimilating image data in a dynamic image model for producing pseudo-observations for the oceanographic model. In this paper, we consider the general case of ill-posed problems, solved using Tikhonov regularization methods, and we define a method to convert them, in a generic way, from Tikhonov regularization to the Data Assimilation framework. The proposed method rewrites the Image Model and the regularization term under the form of the system of three equations (4,5,7) and we study possible choices for the evolution model and the error's covariance, and their impact in term of spatio-temporal regularization.

First, state and observation vectors have to be defined. Obviously, the observations will be images or processed images, but the content of the state vector will strongly depends on the studied problem. For example, segmentation, denoising and restoration use a state vector which is an image. Tracking, image registration and motion estimation use a vector field. Active contours use a curve.
Sometimes, the link between the state vector and the observation is highly indirect. For instance, as it is not possible to deduce the ocean circulation from surface temperatures with a shallow-water model, an additional system has to be defined with relevant evolution and observation equations (Huot et al., 2008).

To solve an Image Processing problem using the 4D-Var algorithm as described in Section 3, a suitable equation describing the temporal evolution of the state vector has to be stated. Next, an observation equation is written expressing the links between the state vector and the images. Last, the initial condition and its background error should be defined. This is discussed in the next Subsections.

4.1 The evolution model

The evolution equation describes the temporal dynamics of the state vector and can be used, for instance, to impose a temporal regularity for recovering result in case of missing data. A first and simple possibility to specify it could be to assume $\mathbf{X}$ being constant over time, which can be expressed by:

$$\frac{d\mathbf{X}}{dt} = 0$$

This equation is developed using the chain rule:

$$\frac{\partial \mathbf{X}}{\partial t} + \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} = 0 \tag{24}$$

which is a transport equation. $\frac{\partial \mathbf{x}}{\partial t}$ is a velocity vector and the evolution model is $\mathbb{M}(\mathbf{X})(\mathbf{x}, t) = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t}$. An example of using (24) as evolution equation is given in Section 5 to determine optical flow.

Another possible specification is to express the transport of the state vector as a diffusion process, a physical law ruling the transport of chemical species or temperature. The general formulation is: $\frac{\partial \mathbf{X}}{\partial t} = \nabla^T (D \nabla \mathbf{X})$ and by identification the evolution model is: $\mathbb{M}(\mathbf{X}) = -\nabla^T (D \nabla \mathbf{X})$. The matrix D is a tensor characterizing simultaneously the direction and the intensity of the diffusion. If D does not depend on spatial coordinates, the diffusion is linear and equivalent to a smoothing process using a Gaussian convolution (Witkin, 1983). It is also possible to drive the diffusion according to image characteristics. A standard example is the Perona & Malik diffusion (Perona and Malik, 1990): the tensor matrix D is equal to $c(\|\nabla \mathbf{X}\|) Id$ with c a Gaussian function and Id the identity matrix with the result of smoothing the image on homogeneous regions and preserving contours. The tensor matrix can be chosen by taking into account the orientation of the image gradient (Weickert, 1998, 2001) resulting in spatial regularization properties similar to that of Nagel described in Section 2.

The two previous definitions of $\mathbb{M}$ remain restrictive for image sequences displaying complex temporal dynamics. In this case, the evolution model could be built using prior information on the scene. For example, dynamics can be approximated by piecewise linear functions whose parameters are estimated by analyzing the evolution of image probability density function with a particule filter method. However, this issue remains complex and widely open.

An evolution model based on a physical law can also be considered. In (Huot et al., 2008; Papadakis et al., 2007b) a shallow-water equation is chosen for modeling the temporal evolution of apparent motion. This is an advection-diffusion equation with additional forcing terms. However, this is specific to the ocean surface circulation and can not be applied to a generic framework. In particular, it is unsuitable to the application described in Section 5.

In addition to the evolution model, the matrix Q of equation (14) has to be specified. It is first important to understand the action of Q^{-1} in the functional (8). Let us consider three possible choices for Q and analyze their respective impact in a functional $\iint F^T(\mathbf{X}) Q^{-1} F(\mathbf{X}) d\mathbf{x} d\mathbf{x}'$ which has to be minimized.

As a first example, let Q be the Dirac covariance defined by $Q(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$. The Dirac function, $\delta()$, has the following property:

$$\int_\Omega \delta(x') \, \delta(x - x') \, dx' = \delta(x)$$

The inverse of a covariance matrix is formally and implicitly defined (Oliver, 1998) by:

$$\int \Sigma^{-1}(\mathbf{x}, \mathbf{x}'') \Sigma(\mathbf{x}'', \mathbf{x}') d\mathbf{x}'' = \delta(\mathbf{x} - \mathbf{x}') \tag{25}$$

By identification, we have $Q^{-1}(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$ and:

$$\begin{aligned} &\iint_{\Omega^2} F(\mathbf{X})^T(\mathbf{x}) Q^{-1}(\mathbf{x},\mathbf{x}') F(\mathbf{X})(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\ &= \int_\Omega F(\mathbf{X})^T(\mathbf{x}) F(\mathbf{X})(\mathbf{x}) d\mathbf{x} = \int_\Omega \|F(\mathbf{X})\|^2 d\mathbf{x} \end{aligned} \tag{26}$$

A Dirac covariance is therefore acting like a zero-order regularization.

More generally, we consider the case of an isotropic covariance which is written as $Q(\mathbf{x},\mathbf{x}') = q(\mathbf{x}-\mathbf{x}')$. In this case, determining its inverse from (25) is equivalent to determine its inverse convolution defining by:

$$q^{-1} \star q(\mathbf{x}) = \delta(\mathbf{x}) \tag{27}$$

This is more easily done in the Fourier domain as the latter equation is equivalent to:

$$\widehat{q^{-1} \star q}(\omega) = 1$$

Using the convolution theorem, we have:

$$\begin{aligned} \widehat{q^{-1}}(\omega)\widehat{q}(\omega) &= 1 \\ \widehat{q^{-1}}(\omega) &= \frac{1}{\widehat{q}(\omega)} \end{aligned}$$

and the inverse convolution is given by the inverse Fourier transform of $\frac{1}{\hat{q}}$.

Let us apply this with the exponential covariance defined by $q(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x}'|}{\sigma}\right)$. Its Fourier transform is $\frac{2\sigma}{1+\sigma^2\omega^2}$. We have:

$$\begin{aligned} \widehat{q^{-1}}(\omega) &= \frac{1+\sigma^2\omega^2}{2\sigma} \\ q^{-1}(\mathbf{x}) &= \frac{1}{2\sigma}\left(\delta(\mathbf{x}) - \sigma^2\delta''(\mathbf{x})\right) \end{aligned}$$

We replace the expression of $Q^{-1}(\mathbf{x},\mathbf{x}')$ by $q^{-1}(\mathbf{x}-\mathbf{x}')$ in the functional (26):

$$\begin{aligned} &\iint_{\Omega^2} F(\mathbf{X})^T(\mathbf{x}) Q^{-1}(\mathbf{x},\mathbf{x}') F(\mathbf{X})(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\ &= \frac{1}{2\sigma}\int_\Omega F(\mathbf{X})^T(\mathbf{x}) \left(F(\mathbf{X}) - \sigma^2 \frac{\partial^2 F(X)}{\partial \mathbf{x}^2}\right) d\mathbf{x} \\ &= \frac{1}{2\sigma}\int_\Omega \left(\|F(\mathbf{X})\|^2 + \sigma^2 \left\|\frac{\partial F(X)}{\partial \mathbf{x}}\right\|^2\right) d\mathbf{x} \end{aligned} \tag{28}$$

Line (28) is obtained from the previous using integration by parts. The exponential covariance is associated to a first-order regularization.

A more complex possibility is the Gaussian covariance defined by $q(\mathbf{x}) = \exp\left(-\frac{\mathbf{x}^2}{\sigma^2}\right)$. Using the previous technique, the Fourier transform of the Gaussian function is first established: $\widehat{q}(\omega) = \frac{\sigma}{\sqrt{\pi}}\exp(-\frac{\sigma^2}{4}\omega^2)$. The Fourier transform of the inverse covariance is thus $\widehat{q^{-1}}(\omega) = \frac{\sqrt{\pi}}{\sigma}\exp\left(\frac{\sigma^2}{4}\omega^2\right)$ but its inverse

Fourier transform can not be directly computed. Let us replace the exponential by its infinite series:

$$\begin{aligned}\widehat{q^{-1}}(\omega) &= \frac{\sqrt{\pi}}{\sigma}\sum_{n\geq 0}\frac{1}{n!}\left(\frac{\sigma^2}{4}\omega^2\right)^n \\ &= \frac{\sqrt{\pi}}{\sigma}\sum_{n\geq 0}a_n\omega^{2n} \end{aligned} \tag{29}$$

with $a_n = \dfrac{1}{n!}\left(\dfrac{\sigma}{4}\right)^{2n}$. Remembering that 1 is the inverse Fourier transform of the Dirac function and using the theorem of Fourier derivation, equation (29) leads to:

$$\begin{aligned} q^{-1}(\mathbf{x}) &= \frac{\sqrt{\pi}}{\sigma}\sum_{n\geq 0}a_n(i)^{2n}\frac{\partial^{2n}}{\partial x^{2n}}\delta(\mathbf{x}) \\ &= \frac{\sqrt{\pi}}{\sigma}\sum_{n\geq 0}a_n(-1)^n\delta^{(2n)}(\mathbf{x}) \end{aligned}$$

Let us examine the impact of such covariance in the functional (26):

$$\begin{aligned} &\iint F(\mathbf{X})^T(\mathbf{x})Q^{-1}(\mathbf{x},\mathbf{x}')F(\mathbf{X})(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \\ &= \sum_{n\geq 0}(-1)^n a_n\int F(\mathbf{X})^T(\mathbf{x})\left(\int \delta^{(2n)}(\mathbf{x}-\mathbf{x}')F(\mathbf{X})(\mathbf{x}')d\mathbf{x}'\right)d\mathbf{x} \\ &= \sum_{n\geq 0}(-1)^n a_n\int F(\mathbf{X})^T(\mathbf{x})F^{(2n)}(\mathbf{X})(\mathbf{x})d\mathbf{x} \end{aligned}$$

For the term of order n, integration by parts is applied n times:

$$(-1)^n\int F(\mathbf{X})^T(\mathbf{x})F^{(2n)}(\mathbf{X})(\mathbf{x})d\mathbf{x} = \int F^{(n)}(\mathbf{X})^T F^{(n)}(\mathbf{X})d\mathbf{x}$$

We finally obtain:

$$\iint F(\mathbf{X})^T(\mathbf{x})Q^{-1}(\mathbf{x},\mathbf{x}')F(\mathbf{X})(\mathbf{x}')d\mathbf{x}d\mathbf{x}' = \int\sum_{n\geq 0}a_n\left\|\frac{\partial^n F(\mathbf{X})}{\partial\mathbf{x}^n}\right\|^2(\mathbf{x})d\mathbf{x}$$

and the Gaussian exponential covariance corresponds to the Tikhonov regularization at any order.

The inversion of a covariance matrix is however non-trivial and usually inaccessible. Restrictive choices have to be made such as those previously described. In the general case, it still remains possible to approximate the matrix of covariance using finite difference operators and to inverse the matrix using numerical techniques. Unfortunately, if the discrete matrix is large, its inversion is costly and often numerically unstable. For further details, the reader is referred to (Oliver, 1998; Tarantola, 2005).

In conclusion, the type of regularization applied on the state vector – a zero-order, a first-order or any order – has to be chosen by selecting the suitable matrix of covariance (Dirac, exponential or Gaussian). A spatial regularization on $\mathbf{X}$ can not be included in the model error as Q describes the error on the evolution equation and not on the state vector itself. However, this regularization on $\mathbf{X}$ can be included in the observation error if required.

4.2 The observation equation

As previously pointed out, the observation equation describes the links between the state vector and the observation. In the standard framework of Image Processing, an image or a sequence of images provides the observation in the Image Model $\mathbb{I}$ expressed in equation (1). The operator $\mathbb{H}$, as it appears in equation (7) (to be more general than (6)), should then be directly defined as the image operator, i.e. $\mathbb{H} \equiv \mathbb{I}$.

The observation error, quantified by its matrix of covariance R, has also to be specified. This is a possible mean for not taking into account missing data: as R weights the observation influence in equation (19), its inverse should have values close to zero when observations should be discarded. Using the Dirac covariance (which expresses a null interaction between two space-time locations), R is then written as:

$$R(\mathbf{x}, t, \mathbf{x}', t') = r(\mathbf{x}, t)\delta\left(\mathbf{x} - \mathbf{x}'\right)\delta\left(t - t'\right)$$

with r a real matrix. The inverse is:

$$R^{-1}(\mathbf{x}, t, \mathbf{x}', t') = \delta\left(\mathbf{x} - \mathbf{x}'\right)\delta\left(t - t'\right) r^{-1}(\mathbf{x}, t) \tag{30}$$

The function r^{-1} characterizes the quality of the observation: a high value indicates that the observation value is relevant and a value close to zero indicates an irrelevant observation value, which should not included in the computation of the solution. Assuming the availability of a function f measuring the confidence in observation values ($f \in [0, 1]$, $f = 0$ for no confidence), one possible formulation of r^{-1} is:

$$r^{-1}(\mathbf{x}, t) = r_0(1 - f(\mathbf{x}, t)) + r_1 f(\mathbf{x}, t) \tag{31}$$

$r^{-1}(\mathbf{x}, t)$ will be equal to a "minimal value" r_0 where confidence is 0 and be equal to a "maximal value" r_1 where confidence is 1. Matrices r_0 and r_1 are chosen to be constant and invertible. For completeness, f is modeled as:

$$f(\mathbf{x}, t) = f_{\text{sensor}}(\mathbf{x}, t) f_{\text{noise}}(\mathbf{x}, t) f_{\mathbb{H}}(\mathbf{x}, t) \tag{32}$$

f_{sensor} indicates the availability of observation values: it is equal to 0 when data are not or wrongly acquired by the sensor. f_{noise} characterizes data quality: it is close to 0 for noisy data. $f_{\mathbb{H}}$ measures the confidence in the observation model; it is close to 0 if the observation equation is not valid.

Observation values with a low confidence will then not be considered for the computation of the state vector and solution of the Image Processing problem.

4.3 Conclusion

In this Section, we considered the class of ill-posed Image Processing problems which are usually solved by constraining spatially and temporally the solution. We proposed a generic method to convert the Image Model and the spatio-temporal regularity constraint in term of Data Assimilation components. The Image Model is taken as the observation model and the observation error weights the importance of the observation values in the computation of the state vector: when data are irrelevant, their values are no more used. Of course, this is only possible if the quality of the observation can be evaluated. The evolution

equation describes the temporal evolution of the state vector and we proposed two generic choices for the operator $\mathbb{M}$, depending on the application domain. However, both are too restrictive to deal with complex temporal dynamics. The spatial regularity of the state vector is ensured by the covariance matrix Q and we gave three examples of generic regularization. In the next Section, we show how to use Data Assimilation to determine optical flow by applying these general principles.

5 Application to optical flow estimation

Let I be a sequence of images on a bounded domain of $\mathbb{R}^2$, denoted Ω. Let $\mathbf{W}(\mathbf{x}, t)$ be the velocity vector of a point $\mathbf{x} \in \Omega$ between frame t and $t + \triangle t$ verifying:

$$I(\mathbf{x} + \mathbf{W}(\mathbf{x}, t)\triangle t, t + \triangle t) = I(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega \tag{33}$$

As this equation is non linear with respect to $\mathbf{W}$, the left member of equation (33) is often linearized using a first order Taylor development around $\triangle t = 0$. This provides the so-called optical flow constraint equation (Horn and Schunk, 1981):

$$\nabla I^T(\mathbf{x}, t)\mathbf{W}(\mathbf{x}, t) + \frac{\partial I}{\partial t}(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \Omega \tag{34}$$

Estimating apparent motion is an ill-posed problem: the velocity vector has two components and the optical flow equation is not sufficient to compute both. A solution could be obtained in the Image Processing context using a Tikhonov regularization as stated in Section 2 by constraining spatially (Horn and Schunk, 1981) or spatially and temporally (Weickert and Schnörr, 2001) the solution. Equation (34) is often preferred to (33) because it is linear and the associated Euler-Lagrange equation can be discretized using an explicit numerical scheme. However, it is possible to directly use the equation (33) to determine the optical flow. This has been described in (Brox et al., 2004) where the resulting Euler-Lagrange equation has been approximated using a semi-implicit scheme. The advantage of equation (33) is that W is correctly estimated even for high velocity values. Equation (34) is an approximation and only well suited for low velocity except if incremental algorithms (Odobez and Bouthemy, 1998; Proesmans et al., 1994) or space-scale methods (Alvarez et al., 2000) are considered.

In this paper, we choose to consider the optical flow constraint (34) in order to describe the application of the Data Assimilation framework to solve a standard Image Processing problem and to compare it with state-of-the-art methods.

5.1 Observation and evolution models

The optical flow constraint is used as the observation equation. With this choice of observation model, the field $\mathbf{W}(\mathbf{x}, t)$ of velocity vectors is now considered as the state vector $\mathbf{X}(\mathbf{x}, t)$ and the sequence of image gradients $(\nabla I(\mathbf{x}, t), I_t(\mathbf{x}, t))$ constitutes the observation vector $\mathbf{Y}(\mathbf{x}, t)$. By identification of (7) and (34), the observation model is:

$$\mathbb{H}(\mathbf{W}, I)(\mathbf{x}, t) = \nabla I(\mathbf{x}, t)^T \mathbf{W}(\mathbf{x}, t) + I_t(\mathbf{x}, t) \tag{35}$$

We have to define an appropriate observation error. Equations (30) and (31) are used to define the inverse of R and to locate the observation values which must not be assimilated. The observation model being scalar, the matrices r_0 and r_1 are scalars and respectively set to ϵ and $1-\epsilon$ with $\epsilon \simeq 10^{-6}$. Equation (32) is used as the observation confidence. Without any information, f_{noise} is assumed to be equal to 1. f_{sensor} is set to 0 where data are not acquired and to 1 otherwise. $f_{\mathbb{H}}$ is chosen from the following remark: the spatio-temporal gradient is null on regions of uniform grey level values and equation (34) is then degenerated. For avoiding considering these points, $f_{\mathbb{H}}$ is defined by:

$$f_{\mathbb{H}}(\mathbf{x},t) = 1 - \exp(-\|\nabla_3 I(\mathbf{x},t)\|^2) \tag{36}$$

where ∇_3 denotes the spatio-temporal gradient operator.

The transport of the velocity, equation (24), is chosen as the evolution equation:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \mathbf{W}^T \mathbf{W} = 0 \tag{37}$$

This equation is rewritten as the two-component system:

$$\frac{\partial U}{\partial t} + UU_x + VU_y = 0 \tag{38}$$

$$\frac{\partial V}{\partial t} + UV_x + VV_y = 0 \tag{39}$$

and the evolution model is:

$$\begin{aligned} \mathbb{M}(\mathbf{W}) &= \begin{pmatrix} \mathbb{M}_1(\mathbf{W}) & \mathbb{M}_2(\mathbf{W}) \end{pmatrix} \\ &= \begin{pmatrix} UU_x + VU_y & UV_x + VV_y \end{pmatrix} \end{aligned}$$

The Q matrix is chosen as:

$$Q(\mathbf{x},t,\mathbf{x}',t') = \exp\left(-\frac{1}{\sigma}\left(\|\mathbf{x}-\mathbf{x}'\| + |t-t'|\right)\right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{40}$$

The initial condition of the state vector has also to be provided. We make use of Horn and Schunck's algorithm (Horn and Schunk, 1981) to compute the velocity field on the two first frames of the sequence. We assume the initial condition to be without errors and we set $B(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$ as the background error.

5.2 Adjoint operators

In order to determine the adjoint operators for $\mathbb{M}$ and $\mathbb{H}$, the directional derivatives must first be established.
Using the definition (9), we obtain:

$$\begin{aligned} \frac{\partial \mathbb{M}_1}{\partial \mathbf{W}}(\eta) &= \frac{\partial \mathbb{M}_1}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_1}{\partial V}(\eta^2) = U\eta^1_x + U_x\eta^1 + V\eta^1_y + U_y\eta^2 \\ \frac{\partial \mathbb{M}_2}{\partial \mathbf{W}}(\eta) &= \frac{\partial \mathbb{M}_2}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_2}{\partial V}(\eta^2) = U\eta^2_x + V_y\eta^2 + V\eta^2_y + V_x\eta^1 \end{aligned}$$

with $\eta = \begin{pmatrix} \eta^1 & \eta^2 \end{pmatrix}^T$ and η_x and η_y standing for partial derivate with respect to x and y. The reader is referred to appendix B.1 for more details. Using (15),

integration by parts and considering border terms to be equal to zero, the adjoint operator of $\mathbb{M}$ is:

$$\left(\frac{\partial \mathbb{M}_1}{\partial \mathbf{W}}\right)^* (\lambda) = -U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + U_y\lambda^2$$
$$\left(\frac{\partial \mathbb{M}_2}{\partial \mathbf{W}}\right)^* (\lambda) = -U_x\lambda^2 - U\lambda_x^2 - V\lambda_y^2 + V_x\lambda^1$$

with $\lambda = \begin{pmatrix}\lambda^1 & \lambda^2\end{pmatrix}^T$. Details are reported in appendix B.2.
The directional derivative of the observation operator is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}}(\eta)(\mathbf{x},t) = \nabla I^T(\mathbf{x},t)\eta(\mathbf{x},t)$$

and determining the adjoint operator is direct, as described in appendix B.3:

$$\left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* (\lambda)(\mathbf{x},t) = \nabla I(\mathbf{x},t)\lambda(\mathbf{x},t)$$

5.3 Discretization

Using the choices made in Subsection 5.1, the three PDEs (19,21,23) become:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{W}^T\nabla\mathbf{W} = 0 \quad (41)$$
$$-\frac{\partial \lambda}{\partial t} - \nabla\lambda^T\mathbf{W} - \lambda^T\nabla^\perp\mathbf{W} = -\frac{1}{2\sigma}\nabla I R^{-1} \star L \quad (42)$$
$$\frac{\partial \delta\mathbf{W}}{\partial t} + \mathbf{W}^T\nabla\delta\mathbf{W} + \nabla\mathbf{W}^T\delta\mathbf{W} = Q \star \lambda \quad (43)$$

with $\nabla^\perp\mathbf{W} = \begin{pmatrix}\nabla^\perp U & \nabla^\perp V\end{pmatrix}$, $\nabla^\perp U = \begin{pmatrix}U_y & -U_x\end{pmatrix}^T$ and $L = I_t + \nabla I^T(\mathbf{W} + \delta\mathbf{W})$. The covariance matrices Q and R being chosen isotropic, they depend on coordinates $(\mathbf{x}-\mathbf{x}', t-t')$, and the right members of equations (42) and (43) can then be expressed as a convolution product. These three equations are discretized using the finite difference technique.

Let us first examine equation (41): it is a 2D non linear advection equation. The advection term corresponds to the velocity transport. Its direct approximation by a standard Euler scheme is numerically unstable. This instability has several origins: the non-linearity of the equation, the multi-dimensionality of the state vector, the simultaneous occurrence of linear and non linear terms in the equation. For stabilizing the numerical scheme, a diffusive term may be introduced into the equation:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla\mathbf{W}^T\mathbf{W} = \kappa\nabla^2\mathbf{W}$$

with $\kappa > 0$ having a small value as done in (Papadakis et al., 2007b). This is known under the name of Lax-Wendroff method. Such an equation can be approximated using an explicit Euler scheme (with Courant-Friedrich-Levy condition) or implicit Euler scheme. This has the drawback to smooth the solution of the equation and not preserving discontinuities. We propose, in the following, a stable numerical scheme for the advection equation without adding a diffusive

term. As $\mathbf{W}(\mathbf{x},t)$ is a vector of $\mathbb{R}^2$, equation (41) has two components. The first component combines a term of linear advection in direction y and non linear advection in direction x. It can be expressed as a two-equation system using a splitting method (Verwer and Sportisse, 1998):

$$\frac{\partial U}{\partial t} + UU_x = 0 \quad (44)$$

$$\frac{\partial U}{\partial t} + VU_y = 0 \quad (45)$$

Equation (44) is rewritten with the Lax-Friedrich method (Sethian, 1996) as $\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0$ with $F(U) = \frac{1}{2}U^2$. This new equation is discretized by the following explicit scheme:

$$U_{i,j}^{k+1} = \frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k) - \frac{\triangle t}{2}(F_{i+1,j}^k - F_{i-1,j}^k)$$

with $U_{i,j}^k = U(x_i, y_i, t_k)$, $F_{i,j}^k = F(U(x_i, y_i, t_k))$ and $\triangle t$ the time step. The term $\frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k)$ stabilizes the scheme (by adding a diffusive effect) when $\triangle t$ has a low value (the Courant-Friedrich-Levy condition). The linear advection (45) is discretized using an explicit shock scheme (Sethian, 1996):

$$U_{i,j}^{k+1} = U_{i,j}^k - \triangle t \left(\max(V_{i,j}^k, 0)\left(U_{i,j}^k - U_{i-1,j}^k\right) + \min(V_{i,j}^k, 0)\left(U_{i+1,j}^k - U_{i,j}^k\right)\right)$$

It can be seen that the second component of (41) contains a linear advection term in direction x and a non linear one in direction y. The same strategy is then applied for discretization.

Equation (42) combines a linear advection, a term of reaction and a forcing term. It also has two components. The first is $-\frac{\partial \lambda^1}{\partial t} - U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + U_y\lambda^2 = \frac{\partial I}{\partial x}A$ with $A = -\frac{1}{2\sigma}R^{-1} \star L$. It is split into two parts. The first part contains the linear advection in direction x and the reaction term $-\frac{\partial \lambda^1}{\partial t} - U\lambda_x^1 - V_y\lambda^1 = 0$ and is discretized in the same way as (45) with an explicit shock scheme. However, the equation is retrograde as the initial condition is given at time T:

$$\begin{aligned}(\lambda^1)_{i,j}^{k-1} = &\left(1 + \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)\right)(\lambda^1)_{i,j}^k + \\ &\triangle t \left(\max(U_{i,j}^k, 0)((\lambda^1)_{i,j}^k - (\lambda^1)_{i-1,j}^k) + \min(U_{i,j}^k, 0)((\lambda^1)_{i+1,j}^k - (\lambda^1)_{i,j}^k)\right)\end{aligned}$$

The second part contains the linear advection term in direction y and the forcing term: $-\frac{\partial \lambda^1}{\partial t} - V\lambda_y^1 + U_y\lambda^2 = \frac{\partial I}{\partial x}A$. Again, an explicit shock scheme is used:

$$\begin{aligned}(\lambda^1)_{i,j}^{k-1} = &(\lambda^1)_{i,j}^k - \frac{\triangle t}{2}\left(U_{i,j+1}^k - U_{i,j-1}^k\right)(\lambda^2)_{i,j}^k - \triangle t(I_x B)_{i,j}^k - \\ &\triangle t \left(\max(V_{i,j}^k, 0)((\lambda^1)_{i,j}^k - (\lambda^1)_{i,j-1}^k) + \min(V_{i,j}^k, 0)((\lambda^1)_{i,j+1}^k - (\lambda^1)_{i,j}^k)\right)\end{aligned}$$

Having the same structure, the second component of (42) is discretized with the same method. The complete numerical scheme is described in Appendix C.2.

The last equation, (43), is similar to equation (42): a linear advection with a reaction term and a forcing term. We therefore use the same discretization technique. The full numerical scheme is shown in Appendix C.3.

5.4 Results

The "taxi" sequence and a synthetic sequence have been chosen for discussing results. For both cases, image gradients are computed with a convolution method and a derivative Gaussian kernel, the parameter σ of the matrix Q – Eq (40) – and the Gaussian kernel variance are set to 1, the number of iterations is set to 5.

The taxi sequence displays several cars moving with a slow and quasi uniform motion.
In a first experiment we compute the optical flow using the Data Assimilation method with image gradients as observations. Horn & Schunk's method is also applied on the sequence and both results are displayed for comparison purposes. Figures 1, 2 and 3 show the results obtained on three frames of the sequence. Results are very similar illustrating that both methods are equivalent when observation values are available on the whole sequence. Having chosen Horn & Schunck or another method does not really matter. The paper's issue is not to compare optical flow methods but to prove the efficiently of Data Assimilation for dealing with missing data and complex dynamics. Then, a second experiment is designed in case of missing data. A large region around the white car, denoted $\Re$, is set to zero on one frame of the sequence to simulate a sensor failure. To indicate to the algorithm the irrelevance of pixel values inside this region, the function f_{sensor} returns the value 0 inside $\Re$ and the value 1 outside. The spatio-temporal gradient is then computed on the modified sequence and provides the observations. Figure 4 shows the results with Data Assimilation and Horn & Schunk methods. This latter obviously fails to provide acceptable velocity vectors over $\Re$ while Data Assimilation provides a correct result thanks to the eviction of missing observation in the computation. A similar experimentation is performed using several small regions set to zero on one frame and flagged as not acquired by f_{sensor}. This lack of observation also disturbs Horn & Schunk's algorithm while Data Assimilation provides a correct result as illustrated by Figure 5. Even a whole frame of the observation sequence can be missing: we force image gradients to zero on the fifth frame of the taxi sequence resulting to $f_{\mathbb{H}} = 0$ on this frame. In this case, Horn & Schunk's method can not provide any result on this frame. Figure 6 is then comparing results obtained by Data Assimilation with and without observation on frame 5. Results remain similar, due to the fact that the evolution model correctly describes the temporal dynamics of the sequence and compensates the missing data.

A third experiment is dedicated to the demonstration that the evolution model allows a better use of the dynamics in the estimation process. For that purpose, we built a synthetic sequence showing one square moving horizontally and one moving vertically. At the end of the sequence, the two squares meet each other. Figures 7(b), 8(b) and 9(b) show the result with Horn & Schunk's method which fails to estimate a correct velocity direction when the squares meet. This is due to an over-regularization. Figures 7(a), 8(a) and 9(a) show the result with Data Assimilation: directions are better estimated because the evolution model correctly describes the dynamics and avoids the effects of spatial regularization.

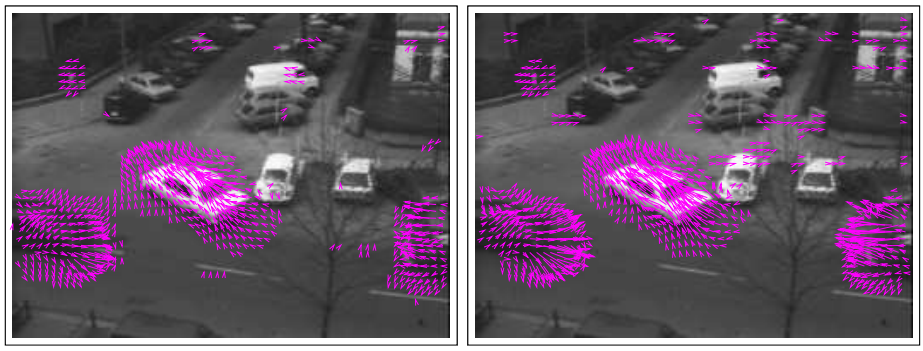


Figure 1: Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 3.

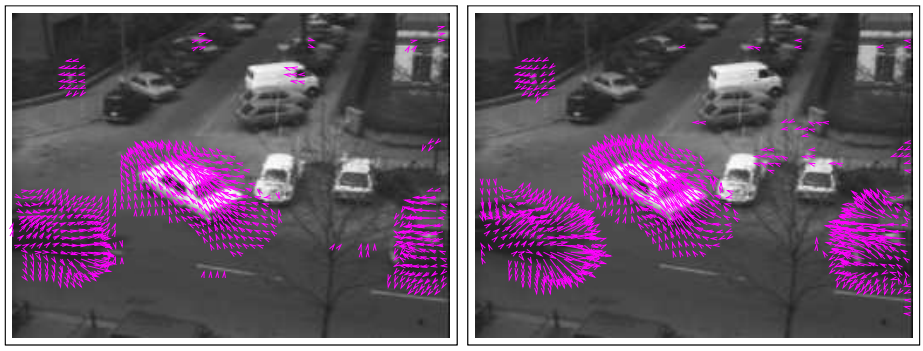


Figure 2: Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 6.

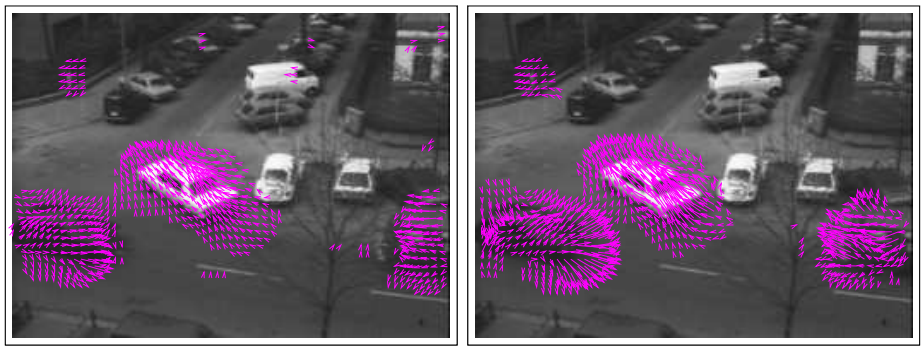
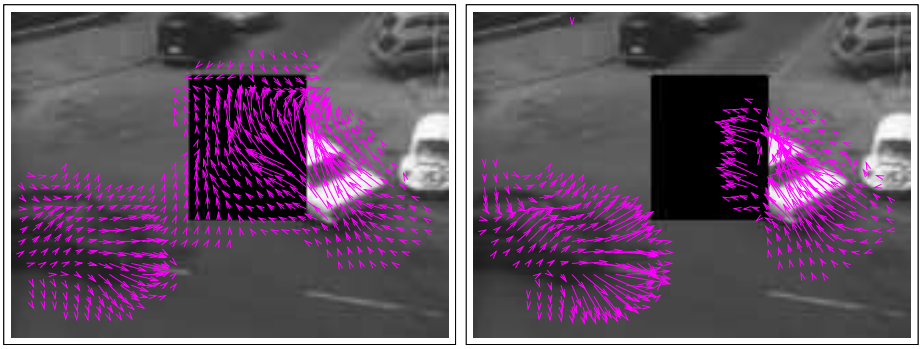
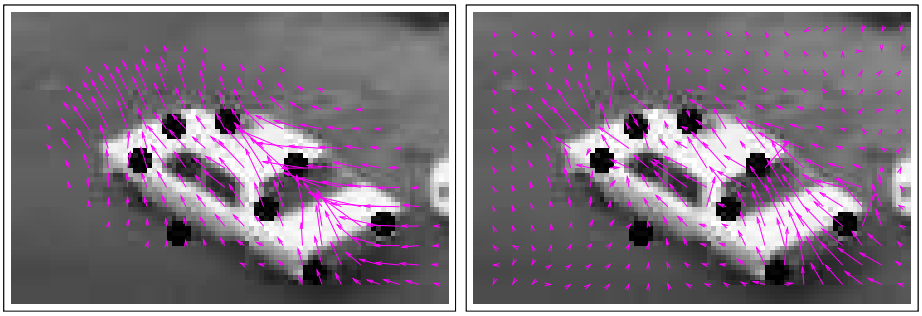


Figure 3: Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 9.



(a) Data assimilation (b) Horn-Schunk

Figure 4: Missing data on a large region in frame 5.

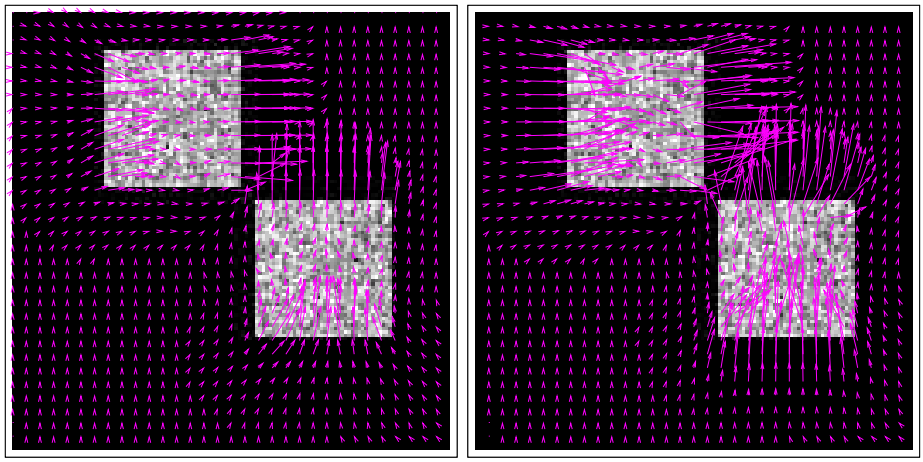


(a) Data assimilation (b) Horn-Schunk

Figure 5: Missing data on small regions in frame 5.

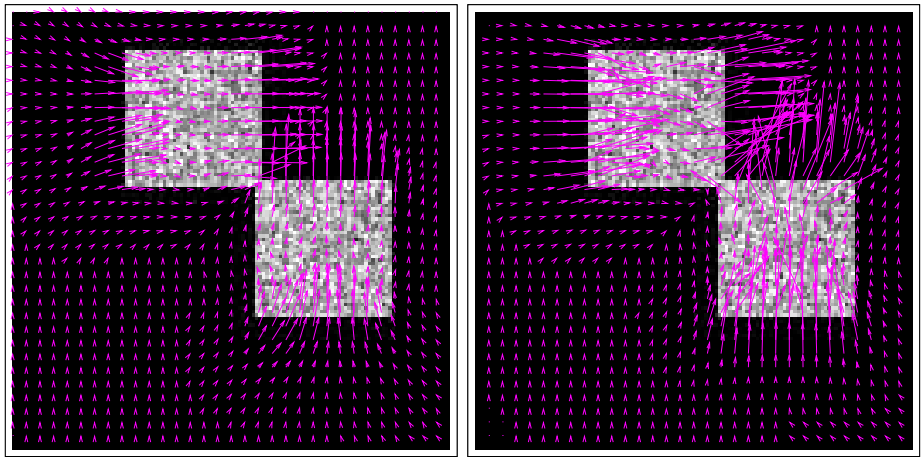
(a) Result with image gradients set to 0 on frame 5 (b) Result with image gradient available

Figure 6: Missing data on the whole frame 5.



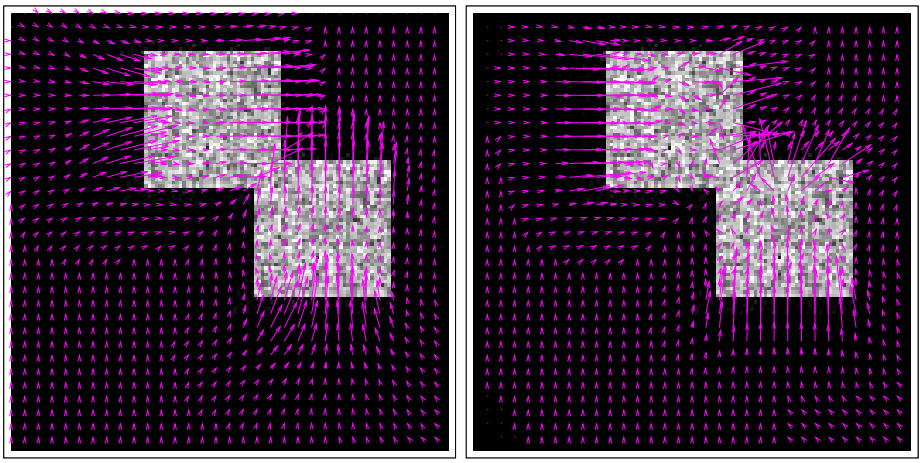
(a) Data Assimilation (b) Horn-Schunk

Figure 7: Results on synthetic sequence - frame 4.



(a) Data Assimilation (b) Horn-Schunk

Figure 8: Results on synthetic sequence - frame 7.



(a) Data Assimilation (b) Horn-Schunk

Figure 9: Results on synthetic sequence - frame 10.

6 Conclusion

In this paper we proposed a general framework to solve ill-posed Image Processing problems by Data Assimilation methods. This approach is an alternative to Weickert's method, which constrains the solution's variations in space and time.

In the case where the sequence dynamics are approximately known and can expressed as an evolution equation, we explained in the paper how this information can be used, simultaneously with the observation equation, in the framework of Data Assimilation to temporally constrain the solution and obtain a better result.

Moreover the knowledge of the temporal dynamics has also being proven useful to handle the problem of missing and noisy data. For that, the observation equation is associated to a specific observation covariance matrix, with high values on pixels of missing and noisy data, which are then not considered during the computation process. On these pixels, the solution is mainly obtained by the evolution equation. Obviously, this is only possible if a confidence measure on observation data is available.

The 4D-var algorithm described in the paper is a frame-by-frame process and has the advantage of low memory requirements compared to a Tikhonov spatio-temporal regularization method, which requires to include the whole sequence in the computation.

The three previous concerns have been illustrated on the issue of estimating the optical flow on a sequence of images by assimilating image gradient observations within a model describing the evolution of velocity. In this application, the evolution equation is the transport of velocity by itself. This is simple and correctly describes the dynamics, which are locally translational and uniform. As this is a non linear equation, the capability of Data Assimilation to solve non linear models is fully exploited. The chosen observation equation used in this application is the optical flow constraint equation, linearly linking velocity and image brightness gradients.

The discretization of the velocity transport equation leads to an unstable numerical scheme when standard techniques of finite differences are used. To overcome this problem, a robust numerical scheme, based on a splitting methods, has been proposed in the paper.

A first perspective of this work is to investigate other evolution equations. For some applications it is possible to use an evolution equation which is physically justified: for instance, ocean surface motion is described with a shallow-water model. If we are concerned with video sequences of rigid objects, the transport of velocity by itself is relevant for locally translational displacements, but becomes inadequate for rotational or discontinuous displacements. A possible solution is to build a parametric model of the image dynamics: the image sequence's spatio-temporal domain is divided in several domains, each one describing a local evolution equation approximating correctly the image dynamics. Then, the transport of velocity by itself can be used for each domain but with specific initial conditions.
A second perspective is to investigate the spatial regularization of state vector $\mathbf{X}$ through the observation covariance matrix R as explained in Subsection 4.1.
A last perspective addresses the experimental application chosen in the paper: the estimation of optical flow. The optical flow constraint equation has been used as observation equation. This equation has the advantage to be linear but it is only an approximation of the transport of image brightness by velocity. The initial transport equation, even if non linear, can be used as observation equation: the differential and adjoint operator of the observation model can be determined if the image brightness (the observation vector) is supposed to be differentiable. The 4D-var algorithm described in Subsection 3.3 and designed to deal with non linear observation and evolution models, could then be used with such observation equation.

A The Euler-Lagrange equation of E (8)

Let us first compute the derivative of E in direction η using definition (9):

$$\begin{aligned}
E(\mathbf{X}+\gamma\eta) &= \int_A\int_A \left(\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta)\right)^T Q^{-1}\left(\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta)\right)^T d\mathbf{x}dtdx'dt' \\
&\quad + \int_A\int_A \mathbb{H}(\mathbf{X}+\gamma\eta)^T R^{-1}\mathbb{H}(\mathbf{X}+\gamma\eta)d\mathbf{x}dtd\mathbf{x}'dt' \\
&\quad + \int_\Omega\int_\Omega (\mathbf{X}+\gamma\eta-\mathbf{X}_b)^T B^{-1}(\mathbf{X}+\gamma\eta-\mathbf{X}_b)d\mathbf{x}d\mathbf{x}' \\
\frac{d}{d\gamma}E(\mathbf{X}+\gamma\eta) &= \int_A\int_A \left(\eta_t+\frac{d}{d\gamma}\mathbb{M}(\mathbf{X}+\gamma\eta)\right)^T Q^{-1}\left(\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta)\right)d\mathbf{x}dtd\mathbf{x}'dt' \\
&\quad + \int_A\int_A \frac{d}{d\gamma}\mathbb{H}(\mathbf{X}+\gamma\eta)^T R^{-1}\mathbb{H}(\mathbf{X}+\gamma\eta)d\mathbf{x}dtd\mathbf{x}'dt' \\
&\quad + \int_\Omega\int_\Omega \gamma^T B^{-1}(\mathbf{X}+\gamma\eta-\mathbf{X}_b)d\mathbf{x}d\mathbf{x}'
\end{aligned}$$

Let us γ tend to zero:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{X}}(\eta) &= \int_A\int_A \left(\eta_t + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}(\eta)\right)^T Q^{-1}\left(\mathbf{X}_t + \mathbb{M}(\mathbf{X})\right) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_A\int_A \frac{\partial \mathbb{H}}{\partial \mathbf{X}}(\eta)^T R^{-1}\mathbb{H}(\mathbf{X}) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_\Omega\int_\Omega \eta^T B^{-1}(\mathbf{X}-\mathbf{X}_b) d\mathbf{x}d\mathbf{x}'\end{aligned}$$

We use integration by parts in order to factorize each term with η^T:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{X}}(\eta) &= \int_A\int_A \eta^T \left(\delta(t=\mathbf{T}) - \delta(t=0) - \frac{\partial}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*\right) Q^{-1}\left(\mathbf{X}_t + \mathbb{M}(\mathbf{X})\right) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_A\int_A \eta^T \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X}) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_\Omega\int_\Omega \eta^T B^{-1}(\mathbf{X}-\mathbf{X}_b) d\mathbf{x}d\mathbf{x}'\end{aligned}$$

Let us introduce the adjoint variable λ in the previous expression and use Fubbini's theorem:

$$\begin{aligned}\frac{\partial \mathbb{M}}{\partial \mathbf{X}}(\eta) &= \int_\Omega \eta^T(\mathbf{x},\mathbf{T})\lambda(\mathbf{x},\mathbf{T})d\mathbf{x} - \int_\Omega \eta^T(\mathbf{x},0)\lambda(\mathbf{x},0)d\mathbf{x} \\ &+ \int_A \eta^T(\mathbf{x},t)\left(-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*(\lambda)\right) d\mathbf{x}dt \\ &+ \int_A \eta^T(\mathbf{x},t)\left(\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X}) d\mathbf{x}'dt'\right) d\mathbf{x}dt \\ &+ \int_\Omega \eta^T(\mathbf{x},0)\left(\int_\Omega B^{-1}(\mathbf{X}(\mathbf{x}',0) - \mathbf{X}_b(\mathbf{x}'))d\mathbf{x}'\right) d\mathbf{x}\end{aligned}$$

A solution of $\dfrac{\partial \mathbb{M}}{\partial \mathbf{X}}(\eta) = 0 \,\forall \eta$ w.r.t. $\mathbf{X}$ verifies the following system:

$$\lambda(\mathbf{x},\mathbf{T}) = 0 \tag{46}$$

$$-\lambda(\mathbf{x},0) + \int_\Omega B^{-1}(\mathbf{X}(\mathbf{x}',0) - \mathbf{X}_b(\mathbf{x}'))d\mathbf{x}' = 0 \tag{47}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*(\lambda) + \int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X}) d\mathbf{x}'dt' = 0 \tag{48}$$

Using the definition of inverse covariance (25), Equation (47) is rewritten as:

$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x}) + \int_\Omega B(\mathbf{x},\mathbf{x}')\lambda(\mathbf{x}',0)d\mathbf{x} \tag{49}$$

and the state vector is expressed from λ using (10):

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q(\mathbf{x},t,\mathbf{x}',t')\lambda(\mathbf{x}',t')d\mathbf{x}'dt' \tag{50}$$

Equations (46), (48), (49) and (50) provide the Euler-Lagrange equations.

B Determination of the adjoint operators

B.1 Differential of $\mathbb{M}$

The operator $\mathbb{M}$ is defined by:

$$\mathbb{M}(\mathbf{W}) = \mathbf{W}^T \nabla \mathbf{W} = \begin{pmatrix} \mathbb{M}_1(\mathbf{W}) \\ \mathbb{M}_2(\mathbf{W}) \end{pmatrix} = \begin{pmatrix} UU_x + VU_y \\ UV_x + VV_y \end{pmatrix}$$

The differential of $\mathbb{M}$ is formally equal to the following Jacobian matrix:

$$\frac{\partial \mathbb{M}}{\partial \mathbf{W}} = \begin{pmatrix} \dfrac{\partial \mathbb{M}_1}{\partial U} & \dfrac{\partial \mathbb{M}_1}{\partial V} \\ \dfrac{\partial \mathbb{M}_2}{\partial U} & \dfrac{\partial \mathbb{M}_2}{\partial V} \end{pmatrix} \tag{51}$$

Each element of this matrix is obtained using the definition of the directional derivative. Considering the first element of $\dfrac{\partial \mathbb{M}}{\partial \mathbf{W}}$, the directional derivative with respect to U in direction η^1 is given by:

$$\left(\frac{\partial \mathbb{M}_1}{\partial U}\right)^T \eta^1 = \lim_{\gamma \to 0} \frac{d}{d\gamma} \left(\mathbb{M}_1(U + \gamma\eta^1, V)\right)$$

and we obtain:

$$\begin{aligned}
\frac{\partial \mathbb{M}_1}{\partial U}^T \eta^1 &= U\eta^1_x + V\eta^1_y + U_x\eta^1 \\
\frac{\partial \mathbb{M}_1}{\partial V}^T \eta^2 &= U_y\eta^2 \\
\frac{\partial \mathbb{M}_2}{\partial U}^T \eta^1 &= V_x\eta^1 \\
\frac{\partial \mathbb{M}_2}{\partial V}^T \eta^2 &= U\eta^2_x + V\eta^2_y + V_y\eta^2
\end{aligned}$$

$$\frac{\partial \mathbb{M}}{\partial \mathbf{W}} = \begin{pmatrix} U\partial_x + V\partial_y + U_x & U_y \\ V_x & U\partial_x + V\partial_y + V_y \end{pmatrix}$$

In equation (23), the differential of $\mathbb{M}$ appears as a directional derivative in direction $\delta\mathbf{W}$ and it has been defined in such a way (see equation (51)) that the Jacobian matrix does not need to be transposed to compute the derivative in a given direction. It is therefore evaluated as:

$$\begin{aligned}
\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W}) &= \frac{\partial \mathbb{M}}{\partial \mathbf{W}}\delta\mathbf{W} = \begin{pmatrix} \dfrac{\partial \mathbb{M}_1}{\partial U}\delta U + \dfrac{\partial \mathbb{M}_1}{\partial V}\delta V \\ \dfrac{\partial \mathbb{M}_2}{\partial U}\delta U + \dfrac{\partial \mathbb{M}_2}{\partial V}\delta V \end{pmatrix} \\
&= \begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}
\end{aligned}$$

B.2 Adjoint operator of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$

The adjoint operator of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$ is formally defined by:

$$\begin{aligned}\left\langle \frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W}), \lambda \right\rangle_{L^2} &= \left\langle \delta\mathbf{W}, \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* (\lambda) \right\rangle_{L^2} \\ &= \int \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})\right)^T \lambda d\mathbf{x}\end{aligned}$$

The directional derivative $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})$ is a column vector and has to be transposed to perform the scalar product with $\lambda = \begin{pmatrix}\lambda^1 & \lambda^2\end{pmatrix}^T$:

$$\begin{aligned}\int \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})\right)^T \lambda d\mathbf{x} &= \int \begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}^T \begin{pmatrix}\lambda^1 \\ \lambda^2\end{pmatrix} d\mathbf{x} \\ &= \int \Big(-\delta U\partial_x(U\lambda^1) - \delta U\partial_y(V\lambda^1) + \delta U U_x\lambda^1 + \delta V U_y\lambda^1 \\ &\qquad \delta U V_x\lambda^2 - \delta V\partial_x(U\lambda^2) - \delta V\partial_y(V\lambda^2) + \delta V V_y\lambda^2 \Big) d\mathbf{x} \\ &= \int \begin{pmatrix}\delta U & \delta V\end{pmatrix} \begin{pmatrix} -U\lambda^1_x - V_y\lambda^1 - V\lambda^1_y + V_x\lambda^2 \\ -U_x\lambda^2 - U\lambda^2_x - V\lambda^2_y + U_y\lambda^1 \end{pmatrix} d\mathbf{x} \\ &= \int \begin{pmatrix}\delta U & \delta V\end{pmatrix} \begin{pmatrix} -U\partial_x - V\partial_y - V_y & V_x \\ U_y & -U\partial_x - V\partial_y - VU_x \end{pmatrix} \begin{pmatrix}\lambda^1 \\ \lambda^2\end{pmatrix} d\mathbf{x}\end{aligned}$$

The adjoint operator of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$ is:

$$\left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* = \begin{pmatrix} -U\partial_x - V\partial_y - V_y & V_x \\ U_y & -U\partial_x - V\partial_y - U_x \end{pmatrix}$$

and its expression in direction λ is:

$$\left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* (\lambda) = \begin{pmatrix} -U\lambda^1_x - V\lambda^1_y - V_y\lambda^1 + V_x\lambda^2 \\ U_y\lambda^1 - U\lambda^2_x - V\lambda^2_y - U_x\lambda^2 \end{pmatrix}$$

B.3 Differential of $\mathbb{H}$ and adjoint operator

The operator $\mathbb{H}$ is defined by:

$$\mathbb{H}(\mathbf{W}, I)(\mathbf{x}, t) = \nabla I(\mathbf{x}, t)^T \mathbf{W}(\mathbf{x}, t) + I_t(\mathbf{x}, t)$$

The differential of $\mathbb{H}$ is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}} = \begin{pmatrix} \frac{\partial \mathbb{H}}{\partial U} & \frac{\partial \mathbb{H}}{\partial V} \end{pmatrix}^T$$

The first component is given by:

$$\begin{aligned}\frac{\partial \mathbb{H}}{\partial U}\eta^1 &= \lim_{\gamma\to 0} \frac{d}{d\gamma} \mathbb{H}(U + \gamma\eta^1, V) \\ &= \lim_{\gamma\to 0} \frac{d}{d\gamma} \left(I_x(U + \gamma\eta^1) + I_y V + I_t \right) \\ &= I_x\eta^1\end{aligned}$$

The same calculus leads for the second component to $\frac{\partial \mathbb{H}}{\partial V}\eta^2 = I_y\eta^2$ and finally the differential of $\mathbb{H}$ is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}} = \nabla I^T$$

The adjoint operator is defined by:

$$\begin{aligned}\left\langle \frac{\partial \mathbb{H}}{\partial \mathbf{W}}(\delta\mathbf{W}), \lambda \right\rangle_{L^2} &= \left\langle \delta\mathbf{W}, \left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* (\lambda) \right\rangle_{L^2} \\ &= \int \nabla I^T \delta\mathbf{W}\lambda d\mathbf{x} \\ &= \int \delta\mathbf{W}^T \nabla I \lambda d\mathbf{x}\end{aligned}$$

And finally, the adjoint operator is:

$$\left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* = \nabla I$$

C Numerical schemes

C.1 Evolution equation of the background variable

Equation (41) has two components and can be written as:

$$\frac{\partial U}{\partial t} + UU_x + VU_y = 0 \quad (52)$$

$$\frac{\partial V}{\partial t} + UV_x + VV_y = 0 \quad (53)$$

Equation (52) is rewritten and split into the system:

$$F = \frac{1}{2}U^2 \quad (54)$$

$$\frac{\partial U}{\partial t} + F_x = 0 \quad (55)$$

$$\frac{\partial U}{\partial t} + VU_y = 0 \quad (56)$$

Equations (55) and (56) are now linear advection equations. They can be approximated using the following schemes:

$$\begin{aligned}\frac{U_{i,j}^{k+1} - \frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k)}{\triangle t} &= -\frac{1}{2}(F_{i+1,j}^k - F_{i-1,j}^k) \\ \frac{U_{i,j}^{k+1} - U_{i,j}^k}{\triangle t} &= -S_y(V,U)_{i,j}^k\end{aligned}$$

S is the discrete operator approximating the advection operator using a shock filter, defined in the following in the y direction:

$$S_y(V,U)_{i,j} = \max(V_{i,j},0)\,(U_{i,j} - U_{i,j-1}) + \min(V_{i,j},0)\,(U_{i,j+1} - U_{i,j})$$

Equation (53) is rewritten and split into the system:

$$\begin{aligned} G &= \frac{1}{2}V^2 \\ \frac{\partial V}{\partial t} + G_y &= 0 \\ \frac{\partial V}{\partial t} + UV_x &= 0 \end{aligned}$$

and then approximated by:

$$\begin{aligned} \frac{V_{i,j}^{k+1} - \frac{1}{2}(V_{i+1,j}^k + V_{i-1,j}^k)}{\triangle t} &= -\frac{1}{2}(G_{i,j+1}^k - G_{i,j-1}^k) \\ \frac{V_{i,j}^{k+1} - V_{i,j}^k}{\triangle t} &= -S_x(U,V)_{i,j}^k \end{aligned}$$

C.2 Evolution equation of the adjoint variable

Equation (42) has two components and can be rewritten as follow:

$$\lambda_t^1 - U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2 = I_xB \quad (57)$$

$$\lambda_t^2 - U\lambda_x^2 - U_x\lambda^2 - V\lambda_y^2 + U_y\lambda^1 = I_yB \quad (58)$$

with $B = -\frac{1}{2\sigma}R^{-1} \star L$. Equation (57) is split into:

$$\begin{aligned} \lambda_t^1 &= U\lambda_x^1 + V_y\lambda^1 \\ \lambda_t^1 &= V\lambda_y^1 - V_x\lambda^2 + (I_xB) \end{aligned}$$

The numerical scheme is retrograde because the initial condition for λ^1 is given at time $t = \mathbf{T}$. We use an explicit Euler scheme:

$$\begin{aligned} \frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} &= S_x(U,\lambda^1)_{i,j}^k + \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)(\lambda^1)_{i,j}^k \\ \frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} &= S_y(V,(\lambda^1)^k)_{i,j} - \frac{1}{2}(V_{i+1,j}^k - V_{i-1,j}^k)(\lambda^2)_{i,j}^k + (I_xB)_{i,j}^k \end{aligned}$$

The numerical scheme is written:

$$\begin{aligned} (\lambda^1)_{i,j}^{k-1} &= (\lambda^1)_{i,j}^k - \triangle t\left(S_x(U,\lambda^1))_{i,j}^k + \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)(\lambda^1)_{i,j}^k\right) \\ (\lambda^1)_{i,j}^{k-1} &= (\lambda^1)_{i,j}^k - \triangle t\left(S_y(V,\lambda^1)_{i,j}^k - \frac{1}{2}(V_{i+1,j}^k - V_{i-1,j}^k)(\lambda^2)_{i,j}^k + (I_xB)_{i,j}^k\right) \end{aligned}$$

The equation (58) is split into:

$$\begin{aligned} \lambda_t^2 &= U\lambda_x^2 + U_x\lambda^2 \\ \lambda_t^2 &= V\lambda_y^2 - U_y\lambda^1 + (I_yB) \end{aligned}$$

and the numerical scheme is:

$$\begin{aligned} (\lambda^2)_{i,j}^{k-1} &= (\lambda^2)_{i,j}^k - \triangle t\left(S_x(U,(\lambda^2))_{i,j}^k + \frac{1}{2}(U_{i+1,j}^k - U_{i-1,j}^k)(\lambda^2)_{i,j}^k\right) \\ (\lambda^2)_{i,j}^{k-1} &= (\lambda^2)_{i,j}^k - \triangle t\left(S_y(V,\lambda^1)_{i,j}^k - \frac{1}{2}(U_{i,j+1}^k - U_{i,j-1}^k)(\lambda^1)_{i,j}^k + (I_yB)_{i,j}^k\right) \end{aligned}$$

C.3 Evolution equation of the incremental variable

Equation (43) has two components which are expressed as follow:

$$\delta U_t + U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V = Q \star \lambda^1 \quad (59)$$
$$\delta V_t + U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V = Q \star \lambda^2 \quad (60)$$

Equation (59) is split into:

$$\begin{aligned} \delta U_t + U\delta U_x + U_x\delta U &= 0 \\ \delta U_t + V\delta U_y + U_y\delta V &= Q \star \lambda^1 \end{aligned}$$

Again, linear advection terms are approximated using chock filter.

$$\begin{aligned} \frac{\delta U_{i,j}^{k+1} - \delta U_{i,j}^k}{\triangle t} &= -S_x(U, \delta U)_{i,j}^k - \frac{1}{2}(U_{i+1,j}^k - U_{i-1,j}^k)\delta U_{i,j}^k \\ \frac{\delta U_{i,j}^{k+1} - \delta U_{i,j}^k}{\triangle t} &= -S_y(V, \delta U)_{i,j}^k - \frac{1}{2}(U_{i,j+1}^k - U_{i,j-1}^k)\delta V_{i,j}^k + (Q \star \lambda^1)_{i,j}^k \end{aligned}$$

Equation (60) is split into:

$$\begin{aligned} \delta V_t + U\delta V_x + V_y\delta V &= 0 \\ \delta V_t + V\delta V_y + V_x\delta U &= Q \star \lambda^2 \end{aligned}$$

and approximated by:

$$\begin{aligned} \frac{\delta V_{i,j}^{k+1} - \delta V_{i,j}^k}{\triangle t} &= -S_x(U, \delta V)_{i,j}^k - \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)\delta V_{i,j}^k \\ \frac{\delta V_{i,j}^{k+1} - \delta V_{i,j}^k}{\triangle t} &= -S_y(V, \delta V)_{i,j}^k - \frac{1}{2}(V_{i+1,j}^k - V_{i-1,j}^k)\delta U_{i,j}^k + (Q \star \lambda^2)_{i,j}^k \end{aligned}$$

References

Alvarez, L., Weickert, J., and Sánchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In Springer-Verlag, editor, *Proceedings of European Conference on Computer Vision*, volume 4, pages 25–36, Prague, Czech Republic.

Hadamard, J. (1923). *Lecture on Cauchy's Problem in Linear Partial Differential Equations*. Yale University Press, New Haven.

Horn, B. and Schunk, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Huot, E., Herlin, I., and Korotaev, G. (2008). Assimilation of sst satellite images for estimation of ocean circulation velocity. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Boston, Massachusetts, U.S.A.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, XLII(577–685).

Nagel, H.-H. (1983). Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21:85–117.

Odobez, J.-M. and Bouthemy, P. (1998). Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155.

Oliver, D. (1998). Calculation of the inverse of the covariance. *Mathematical Geology*, 30(7):911–933.

Papadakis, N., Corpetti, T., and Mémin, E. (2007a). Dynamically consistent optical flow estimation. In *Proceedings of International Conference on Computer Vision*, Rio de Janeiro, Brazil.

Papadakis, N., Héas, P., and Mémin, E. (2007b). Image assimilation for motion estimation of atmospheric layers with shallow-water model. In *Proceedings of Asian Conference on Computer Vision*, pages 864–874, Tokyo, Japan.

Papadakis, N. and Mémin, E. (2007). Variational optimal control technique for the tracking of deformable objects. In *Proceedings of International Conference on Computer Vision*, Rio de Janeiro, Brazil.

Perona, P. and Malik, J. (1990). Space scale and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Proesmans, M., Van Gool, L., Pauwels, E., and Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *Proceedings of European Conference on Computer Vision*, volume 2, pages 295–304.

Sethian, J. (1996). *Level Set Methods*. Cambridge University Press.

Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics.

Tikhonov, A. N. (1963). Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4:1624–1627.

Verwer, J. and Sportisse, B. (1998). A note on operator splitting in a stiff linear case. Technical Report MAS-R9830, Center voor Wiskunde en Informatica.

Weickert, J. (1998). *Anisotropic diffusion in image processing*. ECMI Series. Teubner-Verlag, Stuttgart. ISBN:3-519-02606-6.

Weickert, J. (2001). Applications of nonlinear diffusion in image processing and computer vision. In *Acta Math. Univ. Comenianae. Proceeding of Algoritmy 2000*, volume LXX, pages 33–50.

Weickert, J. and Schnörr, C. (2001). Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255.

Witkin, A. P. (1983). Scale-space filtering. In *Proc. 8th Int. Joint Conf. Art. Intell.*, pages 1019–1022, Karlsruhe, Germany.

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
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 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